

OPTIMAL THREE-DIMENSIONAL
PROJECTION METHOD FOR SOLVING
LINEAR ALGEBRAIC EQUATIONS

Mok Tokko and Roy F. Keller

AMES LABORATORY, USAEC
IOWA STATE UNIVERSITY
AMES, IOWA

Date Transmitted: December 1972

PREPARED FOR THE U. S. ATOMIC ENERGY COMMISSION DIVISION OF RESEARCH
UNDER CONTRACT W-7405-eng-82



MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

OPTIMAL THREE-DIMENSIONAL PROJECTION METHOD FOR
SOLVING LINEAR ALGEBRAIC EQUATIONS

Mok Tokko and Roy F. Keller

Ames Laboratory, USAEC
Iowa State University
Ames, Iowa 50010

Date Transmitted: December 1972

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PREPARED FOR THE U. S. ATOMIC ENERGY COMMISSION
DIVISION OF RESEARCH UNDER CONTRACT NO. W-7405-eng-82

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

GG

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Available from: National Technical Information Service
Department A
Springfield, VA 22151

Price: Microfiche \$0.95

TABLE OF CONTENTS

ABSTRACT	iv
I. INTRODUCTION	1
II. ONE-DIMENSIONAL AND TWO-DIMENSIONAL PROJECTION METHOD	3
III. THREE-DIMENSIONAL PROJECTION METHOD	6
IV. OPTIMAL ORDERING OF COLUMNS OF A	12
V. EXAMPLES AND COMPARISONS	28
VI. CONCLUSIONS	41
VII. BIBLIOGRAPHY	43
VIII. ACKNOWLEDGMENTS	45
IX. APPENDIX: COMPUTER PROGRAM IMPLEMENTATION	46

ABSTRACT

This paper develops an optimal three-dimensional projection method for solving systems of linear algebraic equations. The method modifies three components of the approximate solution vector at each step by using three corresponding column vectors of the coefficient matrix.

The study involved demonstrates the effect of using the columns of the coefficient matrix in an order different than the usual cyclic one. The study also shows that the rate of convergence for a given system of equations is significantly affected by the order of the columns of the coefficient matrix. An optimal order is determined by the size of the angles between the three column vectors used at a given step. The two basic algorithms for ordering the columns of the coefficient matrix are given in Chapter IV.

The method developed is optimal in the sense that it makes a relatively large reduction in the Euclidean norm squared of the residue vector at a given step while maintaining computational simplicity. An ANGL matrix, whose elements are the angles between the columns of the coefficient matrix, is computed only once before the iteration begins. The amount of computation required for correcting one component of the approximate solution vector is essentially the same as the amount of computation required in a one-dimensional or two-dimensional projection method.

I. INTRODUCTION

The purpose of this paper is to determine an algorithm for an optimal three-dimensional projection method by taking a closer look at the coefficient matrix. The paper will illustrate how to reduce the number of iterative steps required to obtain the desired solution by reordering the columns of the coefficient matrix, while maintaining computational simplicity. The results of several test problems are shown in Chapter V and a sample FORTRAN program is implemented on an IBM/360/65 computer. A description and listing of the program are given in the Appendix.

There are numerous methods for solving systems of linear algebraic equations. They can be classified into two categories: direct methods and iterative methods. In direct methods, a solution is obtained by performing a fixed number of arithmetic operations. Although direct methods are known to be faster than the iterative ones, sometimes, because of accumulated round-off errors, one cannot obtain a solution that is accurate enough to be useful. In iterative methods, one starts with an approximate solution which can be either computed by a direct method or by simply guessing. A new improved solution, x^{k+1} , is obtained by operating on the old solution x^k . This process is repeated until a solution of desired accuracy is obtained.

Among the iterative methods is a class of methods called projection methods (6). The approximate solution is improved by projecting the residual vector onto a subspace determined by one or more columns of the coefficient matrix (3). Pyron (11), in his two-dimensional projection

methods, noted that the order by which the columns of the coefficient matrix were paired influenced the rate of convergence to a significant degree.

Some iterative methods impose restrictions on the systems to be solved; for example, Jacobi and Gauss-Seidel methods require that the spectral radius of the iteration matrix be less than one. The only requirement the projection method imposes for convergence is that the coefficient matrix be non-singular.

II. ONE-DIMENSIONAL AND TWO-DIMENSIONAL PROJECTION METHOD

For a system of linear equations

$$Ax = b \quad 2.1$$

it has been shown that the one-dimensional projection method converges if A is non-singular (7). The original approximation of the solution vector is improved one component at a time by an iterative scheme

$$x^{k+1} = x^k + dx_k w_k \quad 2.2$$

where dx_k is some scalar and w_k is a unit vector. The one-dimensional projection method uses one column of A at each step to change one component of the solution vector.

A change is obtained by minimizing the quadratic form (r^k, r^k) where r^k is the k^{th} residual vector defined by

$$r^k = b - Ax^k \quad 2.3$$

which results in

$$dx_k = \frac{(r^k, a_i)}{(a_i, a_i)} \quad 2.4$$

Here a_i is the i^{th} column of A when the change is made on the i^{th}

component of x^k . The new residual vector is computed after each change by

$$r^{k+1} = r^k - dx_{k,i} a_i. \quad 2.5$$

One-dimensional projection methods are in general slow in convergence. De la Garza (3) proposed methods in which the residual vector is projected onto a subspace consisting of two or more columns of A and Shen (12) proposed methods which accelerate any one-dimensional method. The acceleration is accomplished by projecting the residual vector onto two adjacent columns of A - thus correcting two adjacent components of the approximate solution vector as much as possible - and then taking another two components. Pyron (11) studied two-dimensional projection methods in the same way as Shen except that the pair of columns which form the subspace were chosen to obtain an optimal two-dimensional method.

The general equations for a two-dimensional projection method for changing the i^{th} and j^{th} components at the k^{th} step are:

$$\alpha_i = \frac{(r^k, a_i)}{(a_i, a_i)} \quad 2.6a$$

$$\alpha_j = \frac{(r^k - \alpha_i a_i, a_j)}{(a_j, a_j)} \quad 2.6b$$

$$dx_{k,i} = [\alpha_i - \alpha_j (a_i, a_j) / (a_i, a_i)] C_j^i \quad 2.6c$$

$$dx_{k,j} = [\alpha_j - dx_{k,i} (a_i, a_j)] C_j^i \quad 2.6d$$

where

$$C_j^i = \frac{1}{1 - \cos^2 \theta_{ij}} \cdot \quad 2.6e$$

The residual vector after each step is computed by,

$$r^{k+1} = r^k - dx_{k,i} a_i - dx_{k,j} a_j \cdot \quad 2.6f$$

As mentioned in the introduction, the only requirement for a two-dimensional projection method to converge is that A be non-singular. It has been shown that the two-dimensional method in general is more rapid in convergence than the one-dimensional method. Keller et al. (8) showed that for every one-dimensional projection method there exists a two-dimensional projection method which is an acceleration of the one-dimensional projection method.

III. THREE-DIMENSIONAL PROJECTION METHOD

We now turn to the consideration of changing three components of the solution vector at each step. For a system of linear equations

$$Ax = b$$

we define the residual vector to be,

$$r^k = b - Ax^k$$

as we did in Chapter II.

The three components of the approximate solution vector are modified by the following iterative scheme.

$$x_1^{k+1} = x_1^k + dx_{k,1}$$

$$x_2^{k+1} = x_2^k + dx_{k,2}$$

$$x_3^{k+1} = x_3^k + dx_{k,3} \quad 3.1$$

x_1 , x_2 , and x_3 are three arbitrary components of the solution vector.

The residual vector after each step is computed by,

$$r^{k+1} = r^k - dx_{k,1} a_1 - dx_{k,2} a_2 - dx_{k,3} a_3. \quad 3.2$$

where a_1 , a_2 , and a_3 are respectively the columns of A corresponding to x_1 , x_2 , and x_3 .

The computational algorithm for three-dimensional projection method can be based on the following three equations (6).

$$(r^{k+1}, a_1) = 0$$

$$(r^{k+1}, a_2) = 0$$

$$(r^{k+1}, a_3) = 0 \quad 3.3$$

These equations simply state that the residual vector, r^{k+1} , obtained after the k^{th} iteration step is orthogonal to each of the three columns used in the k^{th} step.

By substituting equation 3.2 into equations 3.3, we obtain a 3×3 system of linear equations in which the coefficient matrix is symmetric and consists of inner products of the three columns of A as shown by the following equations, 3.4 and 3.5.

$$(r^k - dx_{k,1} a_1 - dx_{k,2} a_2 - dx_{k,3} a_3, a_1) = 0$$

$$(r^k - dx_{k,1} a_1 - dx_{k,2} a_2 - dx_{k,3} a_3, a_2) = 0$$

$$(r^k - dx_{k,1} a_1 - dx_{k,2} a_2 - dx_{k,3} a_3, a_3) = 0 \quad 3.4$$

By expanding the inner product we get:

$$(a_1, a_1)dx_{k,1} + (a_1, a_2)dx_{k,2} + (a_1, a_3)dx_{k,3} = (r^k, a_1)$$

$$(a_1, a_2)dx_{k,1} + (a_2, a_2)dx_{k,2} + (a_2, a_3)dx_{k,3} = (r^k, a_2)$$

$$(a_1, a_3)dx_{k,1} + (a_2, a_3)dx_{k,2} + (a_3, a_3)dx_{k,3} = (r^k, a_3). \quad 3.5$$

The coefficient matrix is clearly symmetric and the three unknown variables are $dx_{k,1}$, $dx_{k,2}$, and $dx_{k,3}$. The subscripts 1, 2, and 3 in equations 3.5 are used instead of letters purely for convenience and they do not necessarily imply first, second, and third, respectively. For simplicity, we will write the inner product of two vectors, (a_i, a_j) as a_{ij} and henceforth omit the first subscript on $dx_{k,j}$.

To find an algorithm for a three-dimensional projection method we solve the system of equations 3.5 for dx_1 , dx_2 , and dx_3 , by applying Cramer's rule and obtain the following:

$$dx_1 = \frac{1}{\text{Det}} [(r^k, a_1)(a_{33}a_{22} - a_{23}^2) + (r^k, a_2)(a_{13}a_{23} - a_{12}a_{33}) \\ + (r^k, a_3)(a_{23}a_{12} - a_{13}a_{22})]$$

$$\begin{aligned}
dx_2 &= \frac{1}{\text{Det}} [(r^k, a_2)(a_{11}a_{33} - a_{13}^2) + (r^k, a_1)(a_{23}a_{13} - a_{12}a_{33}) \\
&\quad + (r^k, a_3)(a_{12}a_{13} - a_{23}a_{11})] \\
dx_3 &= \frac{1}{\text{Det}} [(r^k, a_3)(a_{11}a_{22} - a_{12}^2) + (r^k, a_1)(a_{12}a_{23} - a_{22}a_{13}) \\
&\quad + (r^k, a_2)(a_{12}a_{13} - a_{23}a_{11})] \tag{3.6}
\end{aligned}$$

where Det is the determinant of the coefficient matrix of the system of equations 3.5.

$$\text{Det} = a_{11}a_{22}a_{33} + 2(a_{23}a_{13}a_{12}) - a_{13}^2a_{22} - a_{23}^2a_{11} - a_{12}^2a_{33} \tag{3.7}$$

Divide both the numerator and the denominator of equations 3.6 by $a_{11}a_{22}a_{33}$ and apply the definition of cosine of the angle between two vectors,

$$\cos \theta_{ij} = \frac{(a_i, a_j)}{(a_i, a_i)^{\frac{1}{2}}(a_j, a_j)^{\frac{1}{2}}}$$

and equations 3.6 become

$$\begin{aligned}
dx_1 &= \frac{1}{D} \left[\frac{(r^k, a_1)}{a_{11}} (1 - \cos^2 \theta_{23}) + (r^k, a_2) (\cos \theta_{13} \cos \theta_{23} - \cos \theta_{12}) / (a_{11} a_{22})^{\frac{1}{2}} \right. \\
&\quad \left. + (r^k, a_3) (\cos \theta_{12} \cos \theta_{23} - \cos \theta_{13}) / (a_{11} a_{33})^{\frac{1}{2}} \right]
\end{aligned}$$

$$\begin{aligned}
dx_2 &= \frac{1}{D} \left[\frac{(r^k, a_2)}{a_{22}} (1 - \cos^2 \theta_{13}) + (r^k, a_1) (\cos \theta_{13} \cos \theta_{23} - \cos \theta_{12}) / (a_{11} a_{22})^{\frac{1}{2}} \right. \\
&\quad \left. + (r^k, a_3) (\cos \theta_{12} \cos \theta_{13} - \cos \theta_{23}) / (a_{22} a_{33})^{\frac{1}{2}} \right] \\
dx_3 &= \frac{1}{D} \left[\frac{(r^k, a_3)}{a_{33}} (1 - \cos^2 \theta_{12}) + (r^k, a_2) (\cos \theta_{12} \cos \theta_{13} - \cos \theta_{23}) / (a_{22} a_{33})^{\frac{1}{2}} \right. \\
&\quad \left. + (r^k, a_1) (\cos \theta_{12} \cos \theta_{23} - \cos \theta_{13}) / (a_{11} a_{33})^{\frac{1}{2}} \right] \tag{3.8}
\end{aligned}$$

where

$$\begin{aligned}
D &= \text{Det} / (a_{11} a_{22} a_{33}) \\
&= 1 + 2 \cos \theta_{12} \cos \theta_{13} \cos \theta_{23} - \cos^2 \theta_{12} - \cos^2 \theta_{13} - \cos^2 \theta_{23}. \tag{3.9}
\end{aligned}$$

We further simplify the equations 3.8 by letting

$$s_1 = (1 - \cos^2 \theta_{23}) / a_{11}$$

$$s_2 = (1 - \cos^2 \theta_{13}) / a_{22}$$

$$s_3 = (1 - \cos^2 \theta_{12}) / a_{33}$$

$$t_1 = (\cos \theta_{13} \cos \theta_{23} - \cos \theta_{12}) / (a_{11} a_{22})^{\frac{1}{2}}$$

$$t_2 = (\cos\theta_{12}\cos\theta_{23} - \cos\theta_{13}) / (a_{11}a_{33})^{\frac{1}{2}}$$

$$t_3 = (\cos\theta_{12}\cos\theta_{13} - \cos\theta_{23}) / (a_{22}a_{33})^{\frac{1}{2}}.$$

Finally, we have

$$dx_1 = \frac{1}{D} [(r^k, a_1)s_1 + (r^k, a_2)t_1 + (r^k, a_3)t_2]$$

$$dx_2 = \frac{1}{D} [(r^k, a_2)s_2 + (r^k, a_1)t_1 + (r^k, a_3)t_3]$$

$$dx_3 = \frac{1}{D} [(r^k, a_3)s_3 + (r^k, a_2)t_3 + (r^k, a_1)t_2]. \quad 3.10$$

Computationally, at any step, we need only to compute three inner products - (r^k, a_1) , (r^k, a_2) , and (r^k, a_3) for each triple $(1, 2, 3)$ of columns a_1 , a_2 , and a_3 . The constant terms, s_1 , s_2 , s_3 , t_1 , t_2 , t_3 , and D are computed strictly from the columns of A and thus they are independent of iteration steps. For an n by n system we need to compute $2n + n/3$ constant terms; s_i and t_i , where $i = 1, 2, \dots, n$, and one D for every triple. Very little time is consumed in computing these terms since the cosines of the angles between all columns of A are already computed and available before the iteration starts, as we will see later in Chapter IV.

IV. OPTIMAL ORDERING OF COLUMNS OF A

Pyron (11) noted in his two-dimensional projection method that the rate of convergence improved when the columns of the coefficient matrix were paired so that the cosine of the angle between two columns was large. As stated in Chapter III, a three-dimensional algorithm is chosen by selecting triples of columns of A.

Experimentation with test cases shows that when we choose different sets of triples, we change the rate of convergence significantly. In some cases, the rate of convergence improved by a factor of nearly 100. In this chapter we take a closer look at the coefficient matrix to determine an algorithm for selecting the "best" triples. An attempt is made to correct the components of the approximate solution vector as much as possible at each step by maximizing $(r^k, r^k) - (r^{k+1}, r^{k+1})$ which is the difference of the residual vector norm squared at any two successive iteration steps.

The following two lemmas will lead us to an appropriate expression for $(r^k, r^k) - (r^{k+1}, r^{k+1})$.

Lemma 4.1

The residual vector r^{k+1} is orthogonal to Δr^k , where $\Delta r^k = dx_1 a_1 + dx_2 a_2 + dx_3 a_3$ and $r^{k+1} = r^k - \Delta r^k$.

Proof Take the inner product of Δr^k with r^{k+1} and expand.

$$\begin{aligned} (r^{k+1}, \Delta r^k) &= (r^{k+1}, dx_1 a_1 + dx_2 a_2 + dx_3 a_3) \\ &= dx_1 (r^{k+1}, a_1) + dx_2 (r^{k+1}, a_2) + dx_3 (r^{k+1}, a_3) \end{aligned}$$

$$= dx_1(r^{k+1}, a_1) + dx_2(r^{k+1}, a_2) + dx_3(r^{k+1}, a_3).$$

Since r^{k+1} is orthogonal to a_1 , a_2 , and a_3 from Chapter III,

$$(r^{k+1}, \Delta r^k) = 0.$$

Lemma 4.2

Let Δr^k be defined as before and the following relationship holds.

$$(r^k, \Delta r^k) = (\Delta r^k, \Delta r^k)$$

Proof The equation 3.2 can be written as

$$r^{k+1} = r^k - \Delta r^k$$

By taking the inner product of both sides with Δr^k , we get

$$\begin{aligned} (r^{k+1}, \Delta r^k) &= (r^k - \Delta r^k, \Delta r^k) \\ &= (r^k, \Delta r^k) - (\Delta r^k, \Delta r^k). \end{aligned}$$

However, since $(r^{k+1}, \Delta r^k) = 0$ from lemma 4.1,

$$(r^k, \Delta r^k) = (\Delta r^k, \Delta r^k). \quad 4.1$$

Theorem 4.1

The difference of the norm squared of the residual vector at any two successive steps is always non-negative, i.e.,

$$(r^k, r^k) - (r^{k+1}, r^{k+1}) \geq 0$$

and at least once every cycle it is greater than zero, provided $r^k \neq 0$.

Proof $(r^k, r^k) - (r^{k+1}, r^{k+1}) = (r^k, r^k) - (r^k - \Delta r^k, r^k - \Delta r^k)$

$$= (r^k, r^k) - (r^k, r^k) + 2(r^k, \Delta r^k) - (\Delta r^k, \Delta r^k)$$

$$= 2(r^k, \Delta r^k) - (\Delta r^k, \Delta r^k).$$

However, $(r^k, \Delta r^k) = (\Delta r^k, \Delta r^k)$ from lemma 4.2, so that

$$(r^k, r^k) - (r^{k+1}, r^{k+1}) = (\Delta r^k, \Delta r^k).$$

Thus, $(r^k, r^k) - (r^{k+1}, r^{k+1})$ is always non-negative.

Now, we want to establish that $\Delta r^k \neq 0$ at least once during every cycle. We must recall that $\Delta r^k = dx_1 a_1 + dx_2 a_2 + dx_3 a_3$ and that the column vectors of A are linearly independent. The only way, then, for Δr^k to be zero at every step is for dx_1 , dx_2 , and dx_3 each to be zero at every step. But, this requires, from equations 3.5, that (r^k, a_i) be zero for $i = 1, 2, \dots, n$. In other words, r^k has to be orthogonal to all columns of A. This, however, is not possible because the only vector

orthogonal to all columns of A is the zero vector and this factor contradicts our assumption that $r^k \neq 0$.

We can now derive an expression for $(r^k, r^k) - (r^{k+1}, r^{k+1})$, so that from theorem 4.1 and lemma 4.2, we have,

$$(r^k, r^k) - (r^{k+1}, r^{k+1}) = (r^k, \Delta r^k).$$

Since $r^k = dx_1 a_1 + dx_2 a_2 + dx_3 a_3$, the equation above can be written as

$$\begin{aligned} (r^k, r^k) - (r^{k+1}, r^{k+1}) &= (r^k, dx_1 a_1 + dx_2 a_2 + dx_3 a_3) \\ &= dx_1 (r^k, a_1) + dx_2 (r^k, a_2) + dx_3 (r^k, a_3). \end{aligned}$$

Substituting equations 3.8 for dx_1 , dx_2 , and dx_3 , we have the following.

$$\begin{aligned} (r^k, r^k) - (r^{k+1}, r^{k+1}) &= \\ &= \frac{1}{D} \left[\frac{(r^k, a_1)}{a_{11}} (1 - \cos^2 \theta_{23}) + \frac{(r^k, a_2)}{(a_{11} a_{22})^{1/2}} (\cos \theta_{13} \cos \theta_{23} - \cos \theta_{12}) \right. \\ &\quad \left. + \frac{(r^k, a_3)}{(a_{11} a_{33})^{1/2}} (\cos \theta_{12} \cos \theta_{23} - \cos \theta_{13}) \right] (r^k, a_1) \\ &\quad + \frac{1}{D} \left[\frac{(r^k, a_2)}{a_{22}} (1 - \cos^2 \theta_{13}) + \frac{(r^k, a_1)}{(a_{11} a_{22})^{1/2}} (\cos \theta_{13} \cos \theta_{23} - \cos \theta_{12}) \right. \end{aligned}$$

$$\begin{aligned}
& + \frac{(r^k, a_3)}{(a_{22}a_{33})^{\frac{1}{2}}} (\cos\theta_{13}\cos\theta_{12} - \cos\theta_{23}) (r^k, a_2) \\
& + \frac{1}{D} \left[\frac{(r^k, a_3)}{a_{33}} (1 - \cos^2\theta_{12}) + \frac{(r^k, a_2)}{(a_{22}a_{33})^{\frac{1}{2}}} (\cos\theta_{12}\cos\theta_{13} - \cos\theta_{33}) \right. \\
& \left. + \frac{(r^k, a_1)}{(a_{11}a_{33})^{\frac{1}{2}}} (\cos\theta_{12}\cos\theta_{23} - \cos\theta_{13}) \right] (r^k, a_3)
\end{aligned}$$

$$(r^k, r^k) - (r^{k+1}, r^{k+1}) =$$

$$\begin{aligned}
& \frac{1}{D} \left[\frac{(r^k, a_1)^2}{a_{11}} (1 - \cos^2\theta_{23}) + \frac{(r^k, a_2)^2}{a_{22}} (1 - \cos^2\theta_{13}) + \frac{(r^k, a_3)^2}{a_{33}} (1 - \cos^2\theta_{12}) \right. \\
& + \frac{2(r^k, a_1)(r^k, a_2)}{(a_{11}a_{22})^{\frac{1}{2}}} (\cos\theta_{13}\cos\theta_{23} - \cos\theta_{12}) \\
& + \frac{2(r^k, a_1)(r^k, a_3)}{(a_{11}a_{33})^{\frac{1}{2}}} (\cos\theta_{12}\cos\theta_{23} - \cos\theta_{13}) \\
& \left. + \frac{2(r^k, a_2)(r^k, a_3)}{(a_{22}a_{33})^{\frac{1}{2}}} (\cos\theta_{13}\cos\theta_{12} - \cos\theta_{23}) \right].
\end{aligned}$$

4.2

Notice that the magnitude of the right side of equation 4.2 depends strictly on r^k and the columns of the coefficient matrix.

Generally, we expect that the larger the change in the residue (as represented by equation 4.2) at each successive step, the more rapid the convergence. Therefore, we want to choose a triple of column vectors which makes the right side of equation 4.2 as large as possible. The magnitude of the right side depends on both the angles between the three chosen column vectors and the three associated inner products, (r^k, a_1) , (r^k, a_2) , and (r^k, a_3) .

An effective way to attempt to maximize the right side is to maximize $\frac{1}{D}$. It usually takes only a few iterations for the magnitude of the inner products to become so small that they no longer affect the right side of equation 4.2 to any significant extent; $\frac{1}{D}$ is the dominant factor. Also, we note that D is strictly dependent on the size of the angles between the three column vectors used in each iteration step and can be computed initially.

In Chapter III, $\frac{1}{D}$ was defined as,

$$\frac{1}{D} = \frac{1}{1 + 2\cos\theta_{12}\cos\theta_{13}\cos\theta_{23} - \cos^2\theta_{12} - \cos^2\theta_{13} - \cos^2\theta_{23}} \quad 4.3$$

A computationally convenient way to maximize $\frac{1}{D}$ is to maximize the cosine squared terms in equation 4.3, which can be done by making the angles as small as possible. The effect the size of the angles has on the magnitude of $\frac{1}{D}$ is shown in Figure 1 on page 19. In this graph the three angles are arbitrarily assumed to be equal and the size of the

angles varies from zero to 90 degrees. When the measure of the three angles is 30 degrees, $\frac{1}{D}$ is 20 and becomes large very rapidly as the angles get smaller. For example, $\frac{1}{D}$ is 40 if the measure of the angles is 15.

The magnitude of $\frac{1}{D}$ is not only dependent on the size of the angles but is also affected by the size of one angle in relation to the other two. The magnitude of $\frac{1}{D}$ can be made larger by making the term $2\cos\theta_{12}\cos\theta_{13}\cos\theta_{23}$ in equation 3.9 smaller (depending, of course, on the sign). Figure 2 on page 20 shows the magnitude of $\frac{1}{D}$ as two of the three angles are arbitrarily assumed to be equal and the third angle is varied. For curve # 1, θ_{12} and θ_{13} are fixed at 30 degrees each and θ_{23} varies from 30 to 60 degrees. Similarly, for curve # 2, θ_{12} and θ_{13} are kept fixed at 40 degrees each and θ_{23} varies from 40 to 80 degrees, and for curve # 3, θ_{12} and θ_{13} are fixed at 50 degrees each and θ_{23} varies from 50 to 100 degrees. The graph shows that the magnitude of $\frac{1}{D}$ gets larger as one of the three angles approaches the sum of the other two. In other words, we would like to choose the three column vectors of A such that they are as nearly coplanar as possible. From now on, when we use the term coplanar we will mean that one of the three angles is equal to the sum of the other two.

Figure # 3 on page 21 illustrates how the magnitude of $\frac{1}{D}$ varies when the three angles are of different sizes and the sum of the three angles is kept constant. Initially, all three angles are 30 degrees. Keeping θ_{12} fixed at 30 we vary θ_{13} and θ_{23} . The magnitude of $\frac{1}{D}$ is plotted against θ_{13} , which is gradually made larger. We should remember,

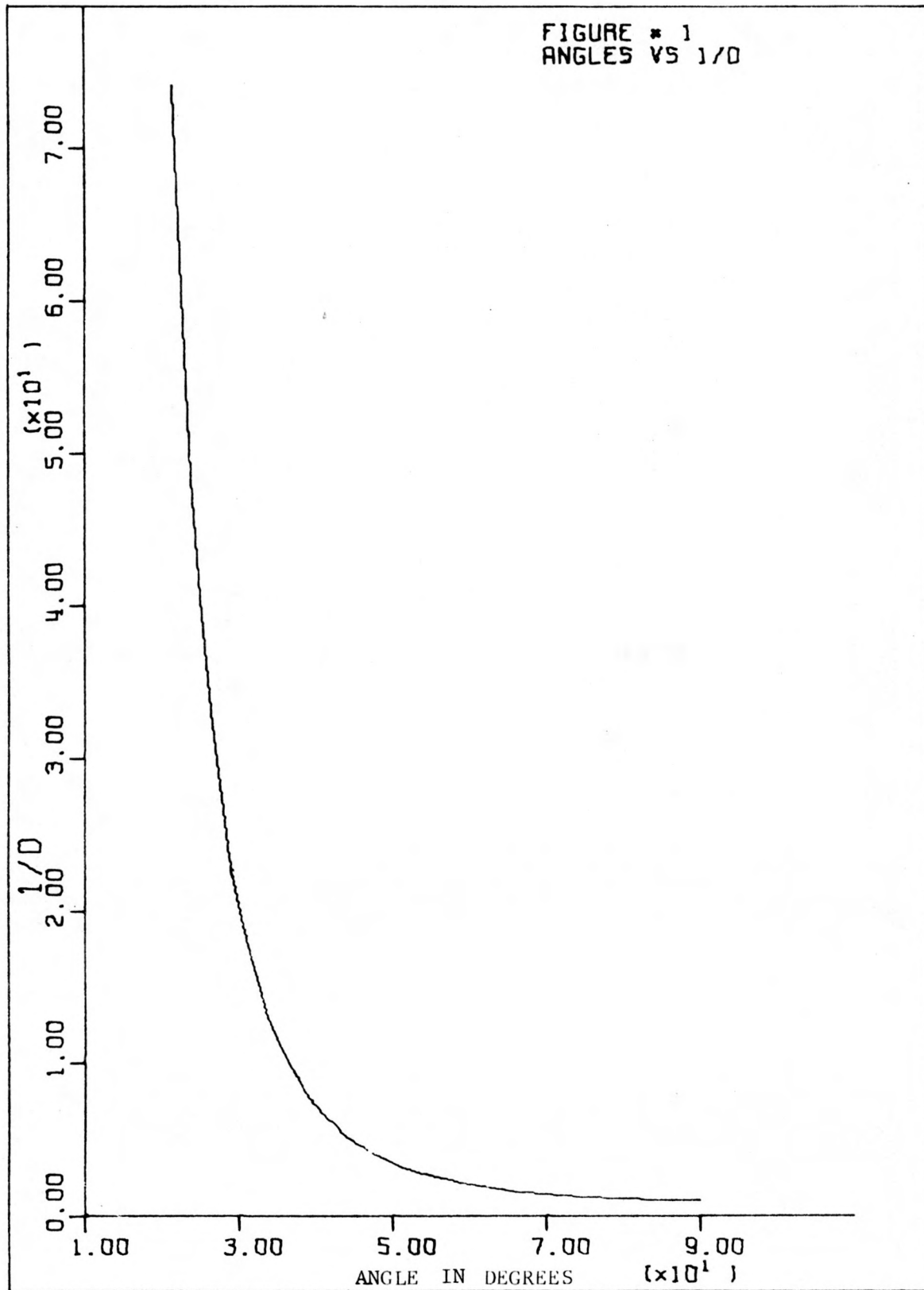


Figure 1. The magnitude of $1/D$ vs the three angles - the three angles are kept equal

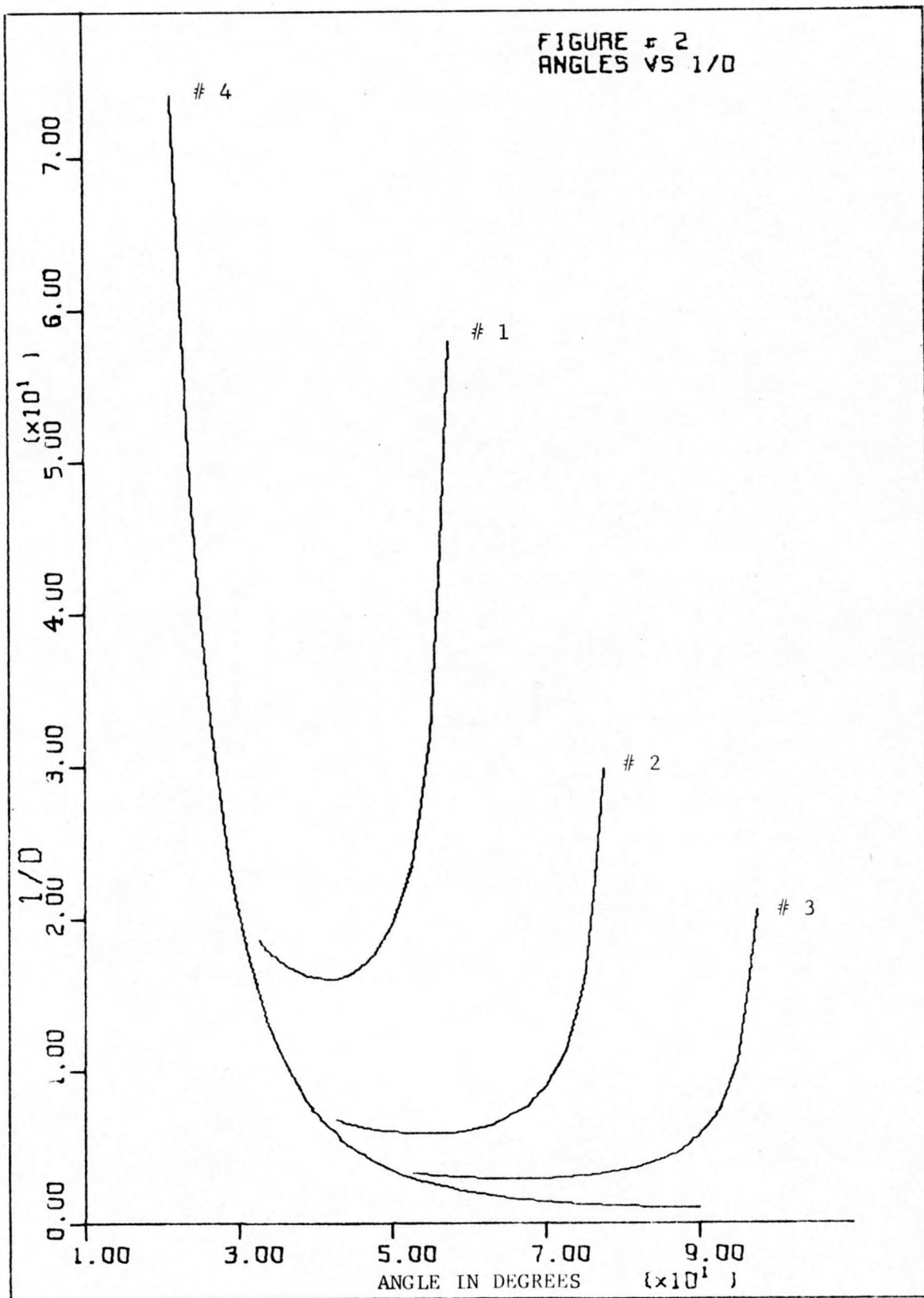


Figure 2. The magnitude of $1/D$ vs the three angles - two angles are kept constant and the third is varied

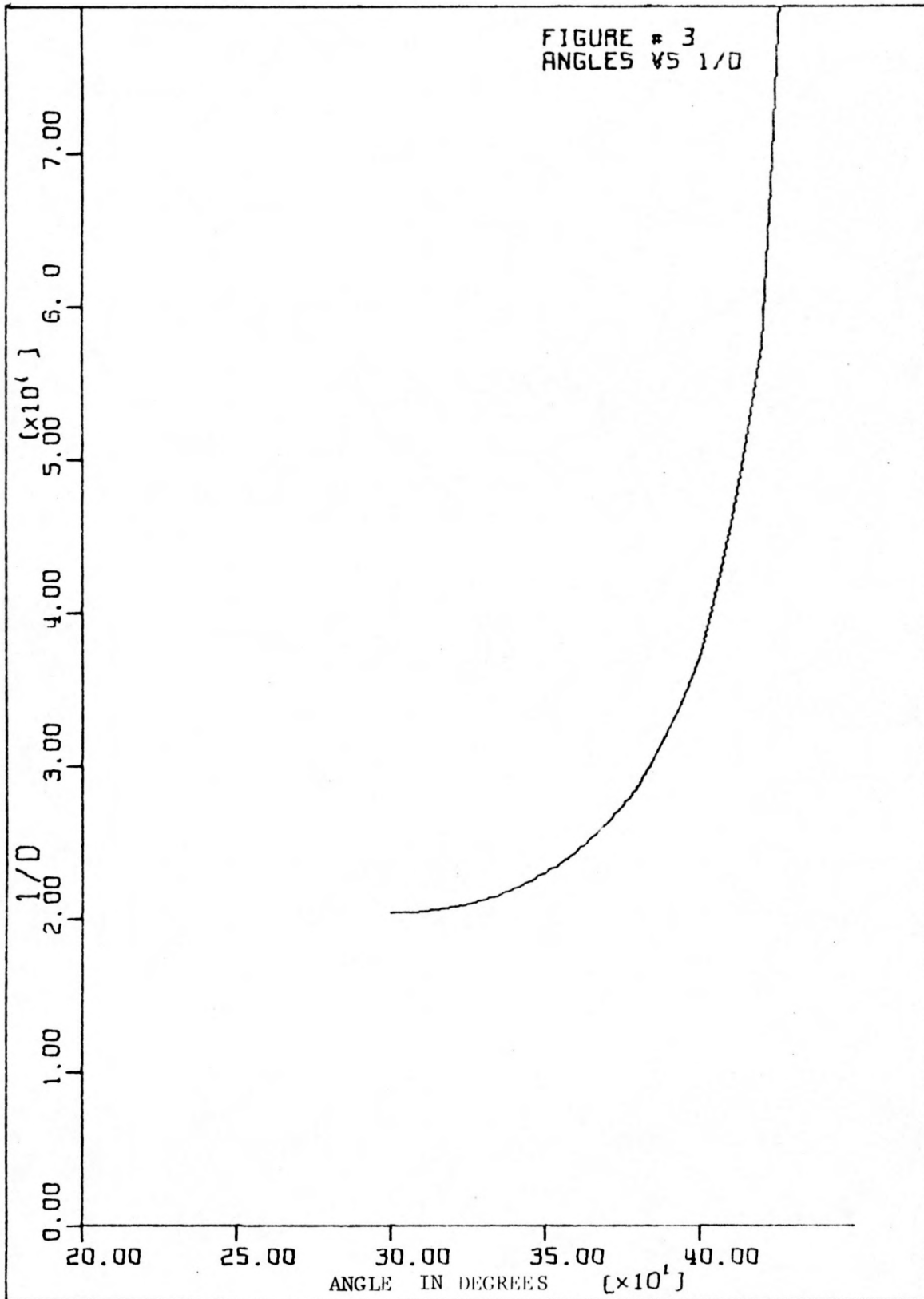


Figure 3. The magnitude of $1/D$ vs the three angles - the sum of the three angles is kept constant

too, that θ_{23} must be made smaller at the same rate in order to keep the sum of the three angles constant. Note that the value of $\frac{1}{D}$ is 20 when all three angles are 30 degrees (also see Figure 1). As the size of θ_{13} gets larger, the magnitude of $\frac{1}{D}$ becomes large. For example, when $\theta_{12} = 30$, $\theta_{13} = 40$, and $\theta_{23} = 20$, $\frac{1}{D}$ is about 48.

In short, we wish to establish an algorithm for ordering the columns of A - one which is based on the following two criteria:

1. Selection of the triples such that the angles between the three column vectors are as small as possible.
2. Selection of the triples such that the three column vectors are as nearly coplanar as possible.

These criteria necessitate the computation of the angles between all column vectors of A which is accomplished by computing the cosines of the angles between the column vectors by the familiar formula,

$$\cos\theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} .$$

We must then construct a matrix whose elements are the angles between the column vectors of A. This is done easily, for the reader's convenience, by converting the cosine of the angles into degrees. The matrix, denoted by ANGL, is used for ordering columns into triples.

Further study of Figure 2 indicates that the magnitude of $\frac{1}{D}$ is 20 when the three angles are 30 degrees and it gets larger as the angles get smaller (see curve # 4), if two angles are fixed at 30 degrees each. In order to obtain a magnitude of 20 or larger by the second criterion, the third angle must be between 50 and 60 degrees (see curve # 1). If the two angles are fixed at 40 degrees each, the third angle must be between 76 and 80 degrees. In other words, it is somewhat more difficult to get larger values of $\frac{1}{D}$ by the second criterion than it is by the first.

The algorithm below is primarily based on the first criterion. It assumes that the ANGL matrix has some small elements.

Algorithm 1.

1. Find the smallest element in the ANGL matrix and denote the row and the column subscripts as i and j , respectively.
2. Determine $K = k$ such that the following is a minimum.

$$\text{ANGL}(i,K) + \text{ANGL}(j,K) \quad \begin{cases} K = 1, 2, \dots, n. \\ k \neq i \\ k \neq j \end{cases}$$

If more than one column is found, select the one in which $|\text{ANGL}(i,K) - \text{ANGL}(j,K)|$ is a maximum. We now have the triple, (i, j, k) , which represents the columns of A .

3. Delete the i , j , and k^{th} rows and columns from consideration of the ANGL matrix and go to step 1.

In this algorithm we pick the triples of columns in a way such that the sum of the three angles associated with the three columns is as small as possible. Whenever we find more than one such triple we choose the one in which the three columns are as coplanar as possible. However, if the ANGL matrix has no relatively small element (or only a few), then algorithm 1 may not be of much help. In this case we can use an algorithm which is based on the second criterion and algorithm 2 given below accomodates this situation.

We start algorithm 2 by finding the smallest element in the ANGL matrix as we did in algorithm 1. We find the smallest element in the i^{th} row and j^{th} column which determines the first two members of the triple - namely, the i^{th} and j^{th} columns of A - and the element of the ANGL matrix thus found represents the angle between the two columns. We now select the third member of the triple, k , such that the three column vectors, i , j , and k are as coplanar as possible. This is accomplished by choosing k such that the absolute difference between $\text{ANGL}(i,k)$ and $\text{ANGL}(j,k)$ is a maximum. We can now formalize algorithm 2.

Algorithm 2.

1. Ascertain the smallest element in the ANGL matrix and denote the row and column subscripts as i and j , respectively.

2. Determine $K = k$ such that the following is a maximum.

$$| \text{ANGL}(i,K) - \text{ANGL}(j,K) | \begin{cases} K = 1, 2, \dots, n. \\ k \neq j \\ k \neq i \end{cases}$$

We now have the triple, (i,j,k) representing the columns of A.

3. Delete the i , j , and k^{th} rows and columns of ANGL matrix from consideration and go to step 1.

It should be noted that we can use either one or both algorithms in a given problem. We may use algorithm # 1 until the small elements in the ANGL matrix are exhausted and then switch to algorithm # 2.

A problem arises when the dimension of the coefficient matrix is not divisible by 3, but this problem is solved by overlapping some of the triples, i.e., some of the columns are used more than once in a cycle. Sometimes this can be helpful even though overlapping is not necessary. For instance, if we successfully ordered the columns of A except for three columns, we discover that these three columns do not form a good triple according to either of the above two criteria. We then search the ANGL matrix in an attempt to find columns which would form a better triple with any of the three remaining columns. This point is illustrated in test problem # 5 (Chapter V).

The right side of equation 4.2 is controlled by yet another factor which has six terms. The first three terms are:

$$(r^k, a_1)^2(1 - \cos^2\theta_{23}) + (r^k, a_2)^2(1 - \cos^2\theta_{13}) + (r^k, a_3)^2(1 - \cos^2\theta_{12})$$

4.4

and obviously these terms are always non-negative. The last three terms of equation 4.2 are:

$$\begin{aligned} & + \frac{2(r^k, a_1)(r^k, a_2)}{(a_{11}, a_{22})} (\cos_{13}\theta\cos\theta_{23} - \cos\theta_{12}) \\ & + \frac{2(r^k, a_1)(r^k, a_3)}{(a_{11}, a_{33})} (\cos\theta_{12}\cos\theta_{23} - \cos\theta_{13}) \\ & + \frac{2(r^k, a_2)(r^k, a_3)}{(a_{22}, a_{33})} (\cos\theta_{12}\cos\theta_{13} - \cos\theta_{23}) \end{aligned} \quad 4.5$$

and the sign of these terms is affected by the size of the angles. We can increase the right side of equation 4.2 by making the terms in 4.5 positive. We begin by making the expressions, containing cosines of the angles

$$\begin{aligned} & (\cos\theta_{13}\cos\theta_{23} - \cos\theta_{12}) \\ & (\cos\theta_{12}\cos\theta_{23} - \cos\theta_{13}) \\ & (\cos\theta_{13}\cos\theta_{12} - \cos\theta_{23}) \end{aligned} \quad 4.6$$

positive.

If all three angles are less than 90 degrees, then the three expressions in 4.6 are usually negative. However, if any one of the three angles is greater than 90 degrees, the cosine of this angle is negative and at least one of the expressions in 4.6 is positive - and when all three angles are greater than 90 degrees, all three expressions in 4.6 are positive. However, this does not assure us that all three terms in 4.5 are positive. The sign of these terms still depends on the sign of the three inner products (r^k, a_1) , (r^k, a_2) , and (r^k, a_3) .

We now assert that the sign of the three terms in 4.5 is most likely to be positive for the following reasons (provided, of course, 4.6 is all positive).

1. Each term in 4.5 contains a product of two inner products and the sign of each term is positive if the sign of the two inner products are both positive or both negative.
2. If the angles between the three column vectors are small, then it is most likely that the sign of all three inner products are the same.

Although we cannot guarantee that selecting triples consisting of angles greater than 90 degrees will always speed up the rate of convergence, we strongly propose that these angles should be taken into consideration. In most cases the rate of convergence will improve.

V. EXAMPLES AND COMPARISONS

In this chapter we illustrate the ordering of columns discussed in Chapter IV with examples and make some comparisons. We begin by comparing the arithmetic operations (addition and multiplication) required by various methods.

To change one component of the solution vector the three-dimensional algorithm requires n multiplications and $n - 1$ additions for computing (r^k, a_i) , n multiplications and $n + 1$ additions for computing the residual vector, $r^{k+1} = r^k - dx_i a_i$, and 4 multiplications and 2 additions for calculating dx_i in equation 3.10. Therefore, the three-dimensional algorithm requires a total of $2n + 4$ multiplications and $2n + 3$ additions and the two-dimensional algorithm requires about the same amount of computation. The difference is in computing dx_i ; the two-dimensional algorithm requires 3 multiplications and 2 additions.

The following is the comparison of the three methods.

Method	No. of Multiplication per component change	No. of Addition per component change
1. Gauss-Seidel	$n + 1$	$n + 1$
2. 2-dimensional projection	$2n + 3$	$2n + 2$
3. 3-dimensional projection	$2n + 4$	$2n + 3$

Note that each of the two projection methods requires approximately twice as much computation as the Gauss-Seidel method.

In addition to the figures shown above, the projection methods require some initial computation. In computing $\frac{1}{2}(n^2 - n)$ cosines, each

requires $n + 2$ multiplications and n additions - also, n multiplications and $n - 1$ additions are required for (a_i, a_i) . For the three-dimensional algorithm, $4n$ multiplications and $2n$ additions are needed to calculate s_i and t_i .

The following is the comparison of the total number of arithmetic operations required by various methods, including the Gauss elimination method, provided, of course, that the solution is obtained after c cycles.

Method	Total Number of Operations
1. Gauss-Seidel	$c(2n^2 + n)$
2. 2-dimensional projection	$c(4n^2 + 4n) + (n^3 + 2n)$
3. 3-dimensional projection	$c(4n^2 + 5n) + (n^3 + 8n)$
4. Gauss elimination	$2n^3 - 2n^2 + 2n - 1$

The execution time for multiplying two floating-point numbers is almost the same as the execution time for adding two floating-point numbers in most of the IBM 360 models including the model 65. Furthermore, we can compare various methods by considering the total number of additions and multiplications.

When comparing the three-dimensional method with the two-dimensional method, it is better to use the number of iterations because, as we have shown, the number of arithmetic operations performed per iteration step in the three-dimensional method is virtually equal to the number of operations performed in the two-dimensional method.

The three-dimensional algorithm, along with the two-dimensional algorithm, is implemented into a FORTRAN program, compiled by an H-level compiler, and tested on an IBM/360/65 computer. A program from the system library is used to test the Gauss elimination method - the program is also written in FORTRAN and compiled by an H-level compiler. This program utilizes complete pivoting and handles all matrices as vectors.

In all of the test problems, iteration stops when the norm of the residual vector is less than 0.001. A test for convergence is made at the end of each cycle. The execution time comparison is made among various methods. The time shown in the comparison table is the "task time", as measured over the entire program, without any other program in the computer. The timing unit in the computer is accurate only to one-sixtieth of a second.

The description of various methods used with test problems are as follows.

- | | | |
|----|--------------|--|
| 1. | 3-D(R) | 3-dimensional projection method with consecutive ordering of columns. |
| 2. | 3-D(O_1) | 3-dimensional projection method using algorithm 1. |
| 3. | 3-D(O_2) | 3-dimensional projection method using algorithm 2. |
| 4. | 3-D(O_3) | 3-dimensional projection method using algorithm 1 but using triples with angles greater than 90 degrees. |

- | | | |
|----|----------------------|---|
| 5. | 3-D(0) _{ov} | 3-dimensional projection method with overlapping of columns. Algorithm 1 and algorithm 2 are both used. |
| 6. | 3-D/2-D | Combination of optimal 2-dimensional method and 3-dimensional method. |
| 7. | 2-D Opt | Optimal 2-dimensional method. |
| 8. | G-S | Gauss-Seidel method. |
| 9. | G-E | Gauss elimination method. |

Test Problems

In test problem # 1 algorithm 2 gives the same triples as algorithm 1, however, the 3-D(0₃) method - which makes use of angles greater than 90 degrees - is far superior. The coefficient matrix is ill-conditioned, and both the Gauss-Seidel and the Gauss elimination methods fail to produce the solution.

For test problem # 2, 3-D(0₃) is not attempted since all the angles in the ANGL matrix are 90 degrees or less. The smallest element in the ANGL matrix for the test problem # 3 is 84 degrees; but 3-D(0₁) method compares favorably with the Gauss-Seidel method.

Test Problem # 1

$$A = \begin{bmatrix} .3 & -.5 & .5 & -.5 & .4 & -.5 \\ .2 & -.4 & .4 & -.5 & .3 & -.4 \\ .1 & -.3 & .1 & -.2 & .2 & -.4 \\ .1 & -.2 & .2 & -.1 & .2 & -.3 \\ -.2 & .3 & -.2 & .2 & -.4 & .2 \\ -.3 & .1 & -.1 & .1 & -.2 & .2 \end{bmatrix} \quad b = \begin{bmatrix} -.3 \\ -.4 \\ -.5 \\ -.1 \\ -.1 \\ -.2 \end{bmatrix}$$

	1	2	3	4	5	6
1	0	151	29	149	21	152
2	151	0	164	15	164	13
3	29	164	0	168	23	158
4	149	15	168	0	155	22
5	21	164	23	155	0	158
6	152	13	158	22	158	0

Method	Order of Columns	Cycles	Iter	Time
1. 3-D(R)	(1 2 3) (4 5 6)	2411	14466	2.61
2. 3-D(O_1)	(2 4 6) (1 3 5)	299	1794	.60
3. 3-D(O_2)	(2 4 6) (1 3 5)	299	1794	.60
4. 3-D(O_3)	(2 5 6) (1 3 4)	97	582	.21
5. 2-D Opt	(3 4) (2 6) (1 5)	691	4146	1.11
6. G-S	No convergence			
7. G-E	No solution			

Test Problem # 2

$$A = \begin{bmatrix} 8 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 8 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 8 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 8 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 8 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 8 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 8 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 8 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 8 \end{bmatrix} \quad b = \begin{bmatrix} 14 \\ 14 \\ 15 \\ 15 \\ 14 \\ 15 \\ 15 \\ 14 \\ 14 \end{bmatrix}$$

	1	2	3	4	5	6	7	8	9
1	0	46	46	88	90	90	90	90	90
2	46	0	46	88	90	90	90	90	90
3	46	46	0	79	88	88	90	90	90
4	88	88	79	0	46	47	88	90	90
5	90	90	88	46	0	46	88	90	90
6	90	90	88	47	46	0	79	88	88
7	90	90	90	88	88	79	0	46	46
8	90	90	90	90	90	88	46	0	46
9	90	90	90	90	90	89	46	46	0

Method	Order of Columns	Cycle	Iter	Time
1. 3-D(O_1)	(1 2 3) (4 5 6) (7 8 9)	6	54	.17
2. 3-D(O_2)	Same as 3-D(O_1)			
3. 3-D(O_3)	No angles larger than 90			
4. 2-D Opt	(1 2) (3 4) (5 6) (7 8) (8 9)	21	210	.20
5. G-S		53	477	.37
6. G-E				.12

Test Problem # 3

$$A = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \quad b = \begin{bmatrix} .5 \\ 1.0 \\ .5 \\ .0 \\ .0 \\ .0 \\ .0 \\ .0 \\ .0 \end{bmatrix}$$

	1	2	3	4	5	6	7	8	9
1	0	116	87	116	87	90	87	90	90
2	116	0	116	84	102	84	90	87	90
3	87	116	0	90	87	116	90	90	87
4	116	84	90	0	115	87	116	84	90
5	87	102	87	115	0	115	84	115	84
6	90	84	116	87	115	0	90	84	116
7	87	90	90	116	84	90	0	116	87
8	90	87	90	84	115	84	116	0	116
9	90	90	86	90	84	116	87	116	0

Method	Order of Columns	Cycle	Iter	Time
1. 3-D(O_1)	(2 4 6) (5 7 9) (1 3 8)	36	324	.27
2. 3-D(O_2)	(2 3 4) (5 6 7) (1 8 9)	24	216	.22
3. 3-D(O_3)	(1 2 3) (4 5 6) (7 8 9)	12	108	.14
4. 2-D O_{pt}	(1 2) (3 4) (5 6) (7 8) (7 9)	20	200	.17
5. 3-D(O_{ov})	(2 6 5) (4 8 9) (1 7 4) (3 5 2)	16	192	.22
6. G-S		17	153	.22
7. G-E				.10

The ANGL matrix for the test problem # 4 contains eight angles ranging from 10 to 24 degrees and algorithm 1 works well as expected. If we apply algorithm 2 we first choose ANGL(1,4) which is 10 degrees. We have several choices for selecting the third member of the first triple; any one of the columns 2, 5, 7, or 9 can be selected. Note that the maximum of $|\text{ANGL}(i,k) - \text{ANGL}(j,k)|$ is only 3 degrees. We select the column 7 as the third member because this choice gives the angle triple (10,15,18) which we believe to be very small. Thus, the triple chosen is identical to the triple chosen by algorithm 1.

In determining the second triple, we first select ANGL(2,5) which is 11 degrees. For the third member of the triple we have a choice of three columns, 3, 6, and 9. The maximum of $|\text{ANGL}(2,k) - \text{ANGL}(5,k)|$ is only one degree and the angle triples associated with each of the three columns are (11,87,88), (11,86,87), and (11,87,88), respectively. Note that each of the three angle triples contains two very large angles. If we pick the column 3, we are left with (6 8 9) as our third triple. The associated angle triple is (15,89,90) and this triple also contains two very large angles. Logically, we do not expect algorithm 2 to perform as well as algorithm 1.

Test Problem # 4

$$A = \begin{bmatrix} .5 & .2 & -.1 & .4 & .2 & -.1 & .3 & .2 & -.1 \\ .4 & -.1 & .2 & .5 & -.1 & .2 & .4 & -.1 & .2 \\ .4 & -.1 & -.2 & .4 & -.1 & -.2 & .5 & -.1 & -.2 \\ -.1 & .5 & .0 & -.1 & .4 & .0 & -.2 & .4 & .0 \\ .0 & .4 & .1 & .0 & .5 & .1 & .0 & .4 & .1 \\ .1 & .2 & -.2 & .1 & .2 & -.2 & .1 & .5 & -.2 \\ .2 & .0 & .5 & .2 & .0 & .4 & .2 & .1 & .2 \\ .0 & .1 & .4 & .0 & .1 & .5 & .0 & .1 & .4 \\ .1 & .0 & .2 & .1 & .0 & .3 & .1 & .0 & .5 \end{bmatrix} \quad b = \begin{bmatrix} 1.5 \\ 1.6 \\ .4 \\ .9 \\ 1.6 \\ .6 \\ .2 \\ 1.6 \\ 1.3 \end{bmatrix}$$

	1	2	3	4	5	6	7	8	9
1	0	91	85	10	90	86	18	86	85
2	91	0	88	94	11	87	101	24	88
3	85	88	0	83	87	13	85	89	22
4	10	94	83	0	93	84	15	88	82
5	90	11	87	93	0	86	99	24	87
6	86	87	13	84	86	0	86	89	15
7	18	101	85	15	99	86	0	94	85
8	86	24	89	88	24	89	94	0	90
9	85	88	22	82	87	15	85	90	0

Method	Order of Columns	Cycle	Iter	Time
1. 3-D(R)	(4 5 9) (1 3 7) (2 8 6)	351	3159	1.10
2. 3-D(O ₁)	(1 4 7) (2 5 8) (3 6 9)	11	99	.20
3. 3-D(O ₂)	(1 4 7) (2 5 3) (6 8 9)	143	1287	.55
4. 3-D(O ₃)	Not attempted			
5. 2-D Opt	(1 4) (2 5) (3 6) (7 9) (8 2)	149	1490	.67
6. G-S	No convergence			
7. G-E				.09

The smallest angle in the ANGL matrix for test problem # 5 is 37 degrees and there are six other angles ranging from 44 to 59 degrees. We do not believe these angles to be very small and we propose that algorithm 2 should assuredly be considered. The column triples obtained from algorithm 2 are, (1 6 9), (3 5 7), and (2 4 8) and the associated angle triples are, (27,63,90), (44,102,121), and (97,107,134). The three columns associated with the first triples are very nearly coplanar. The iteration count drops from 945 (using algorithm 1) to 585 (using algorithm 2).

Since there are several angles greater than 90 degrees in the ANGL matrix, we make an attempt to form triples consisting of angles greater than 90 degrees. The following is the comparison of the angle triples associated with two different sets of triples, method 4 and method 5.

Method	Angle Triples			
Method 4	(37,109,112)	(44,99,113)	(56,107,108)	(63,106,107)
Method 5	(37,63,90)	(53,99,134)	(56,95,121)	(44,99,113)

In method 4, all the angle triples contain angles of unlike signs whereas, in method 5, we retain the first triple, (1 6 9), since it is so nearly coplanar. After comparing the angle triples as shown above we can only conclude that the triples in method 5 are better than the triples in method 4.

Test Problem # 5

$$A = \begin{bmatrix} .5 & .2 & .1 & .3 & .4 & .0 & .8 & .7 & .1 \\ .4 & .0 & -.3 & .1 & -.3 & .2 & -.8 & .9 & .0 \\ .0 & -.2 & .1 & .0 & .2 & .1 & .3 & -.6 & .3 \\ .4 & .1 & .0 & -.7 & .0 & -.3 & .0 & .2 & .1 \\ .9 & .0 & -.1 & .0 & -.2 & .4 & -.6 & .0 & .8 \\ .0 & .1 & .0 & -.4 & .9 & .0 & .3 & -.6 & .0 \\ .8 & -.2 & .3 & .6 & .0 & .1 & .0 & .2 & .3 \\ .1 & .9 & -.8 & .8 & -.7 & .0 & -.4 & .3 & .1 \\ .2 & .0 & -.3 & .5 & -.2 & .2 & -.4 & .0 & .3 \end{bmatrix} \quad b = \begin{bmatrix} 3.1 \\ .2 \\ .2 \\ -.2 \\ 1.2 \\ .3 \\ 1.5 \\ -.5 \\ .3 \end{bmatrix}$$

	1	2	3	4	5	6	7	8	9
1	0	87	113	73	97	59	106	63	37
2	87	0	134	99	113	97	99	72	90
3	113	134	0	107	51	108	53	113	109
4	73	99	107	0	104	56	95	73	75
5	97	113	51	104	0	102	44	117	99
6	59	97	108	57	102	0	121	85	44
7	106	99	53	95	44	121	0	107	109
8	63	72	113	73	117	85	107	0	90
9	37	90	107	75	99	44	109	90	0

Method	Order of Columns	Cycle	Iter	Time
1. 3-D(R)	(1 2 3) (4 5 6) (7 8 9)	251	2259	.74
2. 3-D(O ₁)	(1 6 9) (3 5 7) (2 4 8)	105	945	.42
3. 3-D(O ₂)	(1 9 8) (5 6 7) (2 3 4)	65	585	.33
4. 3-D(O ₃)	(1 9 3) (5 7 2) (4 6 3) (8 1 7)	116	1396	.53
5. 3-D(O ₃)	(1 9 8) (3 7 2) (4 6 7) (5 7 2)	54	648	.33
6. 3-D(O ₃)	(1 9 8) (3 7 2) (4 6 9) (5 7 2)	47	564	.30
7. 2-D Opt	(1 9) (5 7) (2 3) (4 6) (8 1)	94	940	.45
8. G-S	No convergence			
9. G-E				.13

The ANGL matrix for test problem # 6 has ten very small angles ranging from 12 to 22 degrees. Algorithm 1 gives three triples (5 6 8), (1 4 9), and (2 3 7), and the desired solution is obtained after 7002 iterations. Note, however, that the angle triple associated with (2 3 7) is (18,63,64) while the angle between the columns 3 and 7 is 18 degrees and the angle between the columns 2 and 6 is 22 degrees. We solve the system by applying three-dimensional algorithm for the first two triples and two-dimensional algorithm for the two pairs, (3 7) and (2 6). The iteration count is reduced from 7002 to 1980 - a reduction well worth consideration!

We now replace the pair, (2 6), with a triple, (2 6 8), whose angle triple is (18,22,22). The iteration count is further reduced from 1980 to 1342. Replacing the other pair, (3 7) with a triple, (2 3 7), fails to reduce the iteration count. The angle triple associated with this triple is (18,63,64) and it contains two relatively large angles.

We conclude that it is far better to use a triple rather than a pair, if the size of the angles associated with the triple is reasonably close to the size of the angle associated with the pair.

Test Problem # 6

$$A = \begin{bmatrix} .1 & .5 & .2 & .1 & .5 & .5 & .1 & .4 & .0 \\ .1 & .4 & .0 & .0 & .5 & .5 & .1 & .5 & .1 \\ .0 & .1 & .1 & .1 & .3 & .3 & .0 & .2 & .1 \\ .5 & .1 & .1 & .4 & .0 & .1 & .1 & .0 & .4 \\ .4 & .0 & .0 & .5 & .1 & .1 & .0 & .1 & .4 \\ .4 & .1 & .0 & .4 & .0 & .0 & .0 & .1 & .5 \\ .0 & .0 & .1 & .1 & .1 & .1 & .1 & .1 & .1 \\ .1 & .1 & .3 & .0 & .0 & .1 & .2 & .0 & .1 \\ .1 & .1 & .5 & .1 & .1 & .0 & .5 & .1 & .0 \end{bmatrix} \quad b = \begin{bmatrix} 2.4 \\ 2.2 \\ 1.2 \\ 1.7 \\ 1.6 \\ 1.5 \\ .7 \\ .9 \\ 1.5 \end{bmatrix}$$

Test Problem # 6 (cont.)

	1	2	3	4	5	6	7	8	9
1	0	68	73	18	76	71	72	78	18
2	68	0	63	74	23	22	64	22	74
3	73	63	0	75	68	69	18	69	80
4	18	74	75	0	76	73	76	72	18
ANGL = 5	76	23	68	76	0	13	69	12	78
6	71	22	69	73	13	0	72	18	73
7	72	64	18	76	69	72	0	68	80
8	71	22	69	72	12	18	68	0	72
9	18	74	80	18	78	73	80	72	0

Method	Order of Columns	Cycle	Iter	Time
1. 2-D Opt	(5 8) (4 9) (3 7) (2 6) (9 1)	1175	11750	3.97
2. 3-D(R)	(1 2 3) (4 5 6) (7 8 9)	860	7740	2.32
3. 3-D(0 ₁)	(6 8 5) (1 9 4) (2 3 7)	678	7002	1.80
4. 3-S(0 ₂)	Same as 3-D(0 ₁)			
5. 3-D/2D	(6 8 5) (1 9 4) (2 6) (7 3)	191	1980	.72
6. 3-D(0) _{ov}	(6 8 5) (1 9 4) (2 8 5) (7 3)	122	1342	.53
7. 3-D(0) _{ov}	(6 8 5) (1 9 4) (2 8 5) (7 3 2)	128	1536	.57
8. 3-D(0) _{ov}	(6 8 5) (1 9 4) (2 6 5) (7 3)	195	1755	.78
9. 3-D(0) _{ov}	(6 8 5) (1 9 4) (2 6 5) (2 3 7)	198	2376	.83
10. G-S	No convergence			
11. G-E				.13

VI. CONCLUSIONS

Many iterative methods for solving systems of linear algebraic equations have been developed since the advent of high speed electronic computers. The Gauss-Seidel method is one of the most well known iterative methods; but to insure the convergence, the spectral radius of the iteration matrix must be less than one.

The most significant advantage of projection methods is that the only requirement for convergence is for the coefficient matrix to be non-singular. The regular projection method is usually slow in convergence but a significant improvement in the rate of convergence was obtained by the two-dimensional methods. We have demonstrated that the three-dimensional projection method, when the columns of the coefficient matrix are appropriately ordered, significantly improves the rate of convergence.

Direct methods (such as the Gauss elimination method) for solving systems of linear algebraic equations are very popular, and for a small system, these methods are generally faster than the three-dimensional projection method. For a large system, however, the Gauss elimination method may require as much computation as the three-dimensional projection method. If n is very large and the projection method converges in n cycles or less, the number of arithmetic operations required by the two methods is approximately of order n^3 .

If the coefficient matrix is sparse, the execution time for the projection method is reduced since, in most computers, multiplication

by zero takes less time than multiplication by a non-zero number. In fact, with a sparse matrix we need to multiply only the non-zero elements. This will not only shorten the execution time but it will also reduce the amount of storage needed to solve the system.

The algorithm developed in this paper for ordering the columns of the coefficient matrix seems best suited for an interactive system. The ANGL matrix could be displayed and the user could then select appropriate triples.

Future Research

We feel that the following items are worthy of future research:

1. In Chapter IV, algorithm 1 is recommended if the ANGL matrix contains some small elements and algorithm 2 is suggested if the ANGL matrix has only a few or no small elements. A more precise definition of a "small" angle is needed.
2. In test problem # 6, some of the columns of the coefficient matrix are used more than once in a cycle. In some cases this "overlapping" of columns improved the rate of convergence. A simple algorithm to determine when to overlap columns is desirable.
3. We have shown that it is better to use a triple rather than a pair if the size of the angles associated with the triple is reasonably close to the angle associated with the pair. Further study is necessary to provide a criterion for determining when a pair should be used rather than a triple.

VII. BIBLIOGRAPHY

1. G. E. Forsythe, Solving Linear Algebraic Equations Can Be Interesting, Bull. Amer. Math. Soc., 59 (1953) 299-329.
2. L. Fox, An Introduction to Numerical Linear Algebra, Oxford University Press, New York, 1956.
3. A. de la Garza, An Iterative Method for Solving Systems of Linear Equations, Union Carbide and Carbon Corp K-25 Plant, Oak Ridge, Tenn., Report K-731, 1951.
4. Peter Henrici, Elements of Numerical Analysis, John Wiley and Sons, New York, 1964.
5. A. S. Householder, Some Numerical Methods for Solving Systems of Linear Equations, Amer. Math. Monthly, 57 (1950) 453-459.
6. A. S. Householder, Principles of Numerical Analysis, McGraw-Hill Book Company, New York, 1953.
7. R. F. Keller, A Single-Step Gradient Method for Solving Systems of Linear Equations, University of Mo. at Columbia, Math. Sci. Tech. Report No. 8, 1964.
8. R. F. Keller, H. D. Pyron, and I-Ming Shen, On Determining an Optimal 2-Dimensional Projection Method, Document Library, Ames Laboratory, USAEC, Ames, Iowa, Report No. IS-2755, 1968.
9. Serge Lang, Linear Algebra, Addison-Wesley Publishing Company, New York, 1966.
10. D. C. Murdoch, Linear Algebra for Undergraduate, John Wiley and Sons, New York, 1957.
11. Howard D. Pyron, A Non-Stationary Two-Dimensional Acceleration for the One-Dimensional Projection Method, Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1970.
12. I-Ming Shen, Acceleration of the Projection Method for Solving Systems of Linear Equations, Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1970.

13. R. G. Stanton, Numerical Methods for Science and Engineering, Prentice-Hall, Englewood Cliffs, N.J., 1961.
14. R. S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1962.

VIII. ACKNOWLEDGMENTS

I wish to thank my graduate committee for its guidance and interest in my program. This committee was composed of the following professors: I. R. Hentzel, R. F. Keller (major professor), C. G. Maple, T. A. Smay, J. D. Stevens, and R. M. Stewart. Words cannot express the thanks due to Dr. Keller for his encouragement, patience and guidance during the course of this study.

Special thanks are due to my family for their willingness to sacrifice for my educational endeavors. Special thanks is also due to my wife, Catherine, for typing this thesis and for her love and support.

IX. APPENDIX: COMPUTER PROGRAM IMPLEMENTATION

The three-dimensional projection method described in Chapter III is implemented into a FORTRAN program and it has been tested on an IBM/360/65 computer. The convergence criterion is based on the length of the residual vector and a test for convergence is made after each cycle. The approximate solution vector is initially set to zero within the program.

This program can solve a system of linear algebraic equations by the three-dimensional algorithm or two-dimensional algorithm. It can also solve a system by a combination of both three-dimensional and two-dimensional algorithms. If the combination method is used, the three-dimensional algorithm should be executed before the two-dimensional one.

Variable Dictionary

- A: Coefficient matrix.
- C: Right side of equations.
- N: Dimensional of coefficient matrix.
- R: Residual vector.
- S: Constant terms S_i in equations 3.10.
- T: Constant terms t_i in equations 3.10.
- X: Solution vector.
- NT: Total number of columns used per cycle
- N2: Number of columns processed by 2-D method.
- N3: Number of columns processed by 3-D method.

DET : D in equations 3.10.

DX1 : dx_1 .

DX2 : dx_2 .

DX3 : dx_3 .

ERR : Accuracy.

ANGL : Angles between columns of coefficient matrix.

AXR1 : (r^k, a_1) .

AXR2 : (r^k, a_2) .

AXR3 : (r^k, a_3) .

COS1 : Cosines of angles between columns of coefficient matrix.

DOTP : Inner products of columns.

ICOL : Ordered columns.

ISTOP : If 1, program stops after the initial computation.

DPSQR : $(a_i, a_i)^{\frac{1}{2}}$.

KOUNT : Cycle counter.

LIMIT : Maximum number of cycles before iteration stops.

```

    DIMENSION ITITL(20),A(9,9),DOTP(9,9),C(9),DPSQR(9),
    *ICOL(20),S(20),T(20),DET(20),X(9),R(9),COS1(9,9)
100 READ(5,900,END=800) (ITITL(I),I=1,20)
    READ(5,901)N,LIMIT,ISTOP,ERR
    DO 110 I=1,N
    READ(5,902) (A(I,J),J=1,N),C(I)
110 R(I)=C(I)
    WRITE(6,903)(ITITL(I),I=1,20)
    WRITE (6,904) N,LIMIT,ERR
    CCN=180./3.1415
    M=N-1
C . . . . . COMPUTE INNER PRODUCTS OF COLUMNS OF A.
C
    DC 125 L=1,N
    DO 125 I=L,N
    SUM=0.
    DC 120 J=1,N
120 SUM=SUM+A(J,L)*A(J,I)
    DOTP(L,I)=SUM
125 DOTP(I,L)=SUM
    DO 135 I=1,N
    DPSQR(I)=SQRT(DOTP(I,I))
135 COS1(I,I)=0.
C . . . . . COMPUTE COSINES OF ANGLES BETWEEN
C . . . . . COLUMNS OF A
C
    DO 140 I=1,M
    K=I+1
    DO 140 J=K,N
    SX=DOTP(I,J)/(DPSQR(I)*DPSQR(J))
    COS1(I,J)=SX
    ANGL(I,J)=CON*ARCCOS(SX)
140 ANGL(J,I)=ANGL(I,J)
    WRITE (6,906)
    DO 150 I=1,N
150 WRITE(6,905) (A(I,J),J=1,N), C(I)
    WRITE (6,908)
    WRITE (6,907) (I,I=1,N)
    DC 155 I=1,N
    X(I)=0.
    R(I)=C(I)
155 WRITE (6,909) I, (COS1(I,J),J=1,N)
    IF (ISTOP.EQ.1) STOP
C . . . . . DISPLAY ANGL MATRIX
C . . . . . AND READ IN THE ORDER OF COLUMNS
    KOUNT=1
    READ(5,910,END=800) NT,N3,N2,(ICOL(I),I=1,NT)
    WRITE(6,911) N3,N2,(ICOL(I),I=1,NT)
C . . . . . COMPUTE CONSTANT TERMS T, S, DET.

```

```

IF (N3.EQ.0) GO TO 210
DO 160 I=1,N3,3
  I1=ICOL(I)
  I2=ICOL(I+1)
  I3=ICOL(I+2)
  IT=(I+3)/3
  C1=COS1(I1,I2)
  C2=COS1(I1,I3)
  C3=COS1(I2,I3)
  DET(IT)=1.+2.*C1*C2*C3-C1**2-C2**2-C3**2
  T(I)=(1.-C3**2)/DOTP(I1,I1)
  T(I+1)=(1.-C2**2)/DOTP(I2,I2)
  T(I+2)=(1.-C1**2)/DOTP(I3,I3)
  S(I)=(C2*C3-C1)/(DPSQR(I1)*DPSQR(I2))
  S(I+1)=(C1*C3-C2)/(DPSQR(I1)*DPSQR(I3))
160 S(I+2)=(C1*C2-C3)/(DPSQR(I2)*DPSQR(I3))
C
175 DO 190 IN=1,N3,3
C . . . . . BEGIN 3-D ITERATION LOOP.
  I1=ICOL(IN)
  I2=ICOL(IN+1)
  I3=ICOL(IN+2)
  IS=(IN+3)/3
  AXR1=0.
  AXR2=0.
  AXR3=0.
  DO 180 I=1,N
  AXR1=AXR1+R(I)*A(I,I1)
  AXR2=AXR2+R(I)*A(I,I2)
180 AXR3=AXR3+R(I)*A(I,I3)
  DET2=DET(IS)
  DX1=(AXR1*T(IN)+AXR2*S(IN)+AXR3*S(IN+1))/DET2
  DX2=(AXR2*T(IN+1)+AXR1*S(IN)+AXR3*S(IN+2))/DET2
  DX3=(AXR3*T(IN+2)+AXR2*S(IN+2)+AXR1*S(IN+1))/DET2
  X(I1)=X(I1)+DX1
  X(I2)=X(I2)+DX2
  X(I3)=X(I3)+DX3
  DO 188 I=1,N
188 R(I)=R(I)-DX1*A(I,I1)-DX2*A(I,I2)-DX3*A(I,I3)
190 CONTINUE
  IF(N2.EQ.0) GO TO 295
C . . . . . BEGIN 2-D ITERATION LOOP.
C
210 M2=N3+1
215 DO 218 IY=M2,NT,2
  I=ICOL(IY)
  J=ICOL(IY+1)
218 DET(IY)=DOTP(I,I)*DOTP(J,J)-DOTP(I,J)**2
  DO 225 IR=M2,NT,2
  I=ICOL(IR)

```

```

J=ICCL(IR+1)
AXR1=0.
AXR2=0.
DC 220 IW=1,N
AXR1=AXR1+R(IW)*A(IW,I)
220 AXR2=AXR2+R(IW)*A(IW,J)
DX1=(DOTP(J,J)*AXR1-DOTP(I,J)*AXR2)/DET(IR)
DX2=(DOTP(I,I)*AXR2-DOTP(I,J)*AXR1)/DET(IR)
X(I)=X(I)+DX1
X(J)=X(J)+DX2
DC 225 IP=1,N
R(IP)=R(IP)-A(IP,I)*DX1
225 R(IP)=R(IP)-A(IP,J)*DX2
295 DPR=0.
DC 297 I=1,N
297 DPR=DPR+R(I)**2
SQDPR=SQRT(DPR)
IF(SQDPR.LT.ERR) GO TO 300
IF(KOUNT.GE.LIMIT) GO TO 300
KOUNT=KOUNT+1
IF(N3.EQ.0) GO TO 215
GC TO 175
300 WRITE(6,912) (I,X(I),I=1,N)
WRITE(6,913) KOUNT,NT,SQDPR
GC TO 100
800 STCP
900 FCRMAT(20A4)
901 FCRMAT (3I5,F10.6)
902 FCRMAT(8F10.4)
903 FCRMAT('1',20A4)
904 FORMAT('0',10X,'N=',I3,'/11X,'MAX NO OF ITER=',I6,
*/11X,'ACCURACY=',F10.5)
905 FCRMAT('0',10F10.5)
906 FCRMAT('0A MATRIX')
907 FCRMAT('0',5X,10I8)
908 FORMAT('1','ANGLES BETWEEN COLUMNS OF A')
909 FCRMAT('0',15,10F8.2)
910 FCRMAT(20I4)
911 FORMAT('ON3=',I5,' N2=',I5,' ORDER OF COL.=',20I3)
912 FORMAT('0SOLUTION VECTOR',/,(10X,'X(',I3,')=',F10.5))
913 FORMAT('0NO. OF CYCLES=',I6,' TOT NO. OF COL USED=',I5,
*' THE NORM OF RES VECTOR=',F10.5)
914 FCRMAT ('0',15,9F10.5)
915 FCRMAT (' ',5X,9F10.5)
END

```