CONJUGACY AND GRADIENTS

by

MAGNUS R. HESTENES

I have been invited to describe my experiences in the field of numerical analysis and to describe how these experiences influenced me in my studies of mathematics. In particular, I was invited to tell the story of the development of the conjugate gradient method for solving linear systems. I was one of the originators of this method.

At the invitation of the Mathematical Association of America, John Todd and I have written a short history of the Institute for Numerical Analysis, 1947-1954, located on the campus of UCLA. This Institute, called INA, was a Section of the National Applied Mathematical Laboratories, which formed the Applied Mathematics Division of the National Bureau of Standards, a part of the Department of Commerce. In this brief history we were concerned mainly with the mathematical aspects of this program. In particular, we were concerned about who participated in the project, what did they do, and what was their University affiliation. It is not my intention to repeat the material presented in this history except perhaps for some special items of interest.

As many of you know my specialty in mathematics is Variational Theory and Optimal Control Theory. My experiences in these fields have greatly influenced my approach to problems in numerical analysis. I shall describe certain aspects of Variational Theory, which are not only of interest in themselves but which led to a method of attack of certain computational problems.

I received my doctorate at the University of Chicago in 1932. After remaining at Chicago for a year, I left for Harvard as a National Research Fellow to work with Marston Morse. Inspired by the works of George D. Birkhoff, his mentor, Morse had become famous by his development of the Calculus of Variations in the large. Early in 1934, G.D. Birkhoff invited me to join him in writing a chapter in the Calculus of Variations. He wished to develop a new approach to the Calculus of Variations in the large. His idea was simple. It

© 1987 ACM 089791-229-2/87/005/0071 75¢

came from the observation that every critical point of a function $F(x)$ satisfied constraints of the form

$$F'(x;h) = 0,$$

where h is held fast and x was allowed to vary. Here $F'(x,h)$ is the first variation of F, the differential of F. Unfortunately, in the general case, this procedure introduced too many singularities to be effective. However, it was very effective in the quadratic case. In quadratic case the condition $F'(x,h) = 0$ is a "conjugacy" condition although we did not use the term. As a result I wrote a long paper with Birkhoff on this subject developing these ideas for Calculus of Variations in the Small. Later, I wrote an extensive paper of the theory of "Quadratic Forms in Hilbert Space with Applications to the Calculus of Variations". In this paper the concept of conjugacy played a dominant role. I used the term "Q-orthogonality" instead of the term "conjugacy" in my writings. To see what conjugacy means in this context, may I remark that the extremals, the solutions of Euler-Lagrange equations, are the elements that are conjugate to the elements that vanish on the boundary. Thus, I was very familiar with the concept and use of conjugacy early in my career.

It is interesting to recall that, in 1936, I developed an algorithm for constructing a set of mutually conjugate directions in Euclidean Space for the purpose of studying quadric surfaces. I showed my results to Professor Graustein, a Geometer at Harvard University. His reaction was that it was too obvious to merit publication. This shows that Geometers were well versed in the concept of conjugacy. It suggests that perhaps hidden in the literature on geometry there is a method for finding the center of an ellipsoid which is equivalent to the method of conjugate gradients.

During the latter years of World War II, I was a member of the Applied Mathematics Group at Columbia University. Here I was concerned with the mathematical theory of aerial gunnery. We tested our theory with numerical computations. In one project, L.W. Cohen flew fighter planes on paper, duplicating with remarkable accuracy the results obtained by photography of actual paths of fighter planes, flying under certain gunnery rules for attacking bombers. Cohen succeeded where others had failed. He succeeded because he wrote his algorithm in a manner so as to decrease errors which one encounters in computations.

When World War II ended, I returned to the University of Chicago. Shortly thereafter I accepted a Professorship at UCLA. Here I was approached by E. Paxson of the RAND Corporation to study the problem of steering a fighter plane so that it reached a prescribed position and direction in minimum time. This was a complicated variational problem involving differential constraints. Such problems had various names, such as, the Problem of Bolza, the Problem of Lagrange, or the Problem of Mayer. I found that the classical formulation of these problems did not fit this time optimal problem in a natural manner. Accordingly, I reformulated the variational problem so as to be more easily applicable to this minimum time problem. In doing so I had formulated a variational problem which is now known as an Optimal Control Problem. I translated the known results to fit this new formulation. The results were written up in 1949 as a RAND Report and were not published in a standard journal at that time. Later in 1965 I published a book entitled *Calculus of Variations and Optimal Control Theory*, which included the theoretical basis for this time optimal control problem. You might be interested to know that Pontryagin too was invited by his government to study the problem of aerial combat. This led to his formulation of Optimal Control Theory and Differential Games. His first necessary condition for an optimal control problem is now called *Pontryagin's Maximum Principle.* It is an extension of the standard conditions of Euler, Lagrange and Weierstrass. He established his results under weaker hypothesis than had been used heretofore. Thus, the study of the theory of aerial combat led to the development of modern theory of optimal control both here and in Russia.

Return to the time optimal problem proposed by Paxson. We obtained the equation of motion for our fighter plane and attempted to solve these equations numerically on a REAC. The REAC was an electrical analogue computer with about 3% accuracy. We tried to solve our problem as an initial value problem hoping to obtain the prescribed terminal conditions by a suitable choice of initial conditions. The results were disastrous. It turned out that our equations were unstable in the forward direction. They were also unstable in the backward direction. However, by making many trials, we did obtain some notion of the nature of optimal paths. But this did not give us a sought after "Rule of Thumb" method for flying a plane in an optimal manner. Because of this experience I became convinced that we should look for an alternative approach to numerical solutions of variational problems of this type. In my considerations I restricted myself to simple variational problems. In particular, I chose to study the classical problem of finding surface of revolution of least area having prescribed circular boundary curves. The Euler equations to this problem normally has more than one solution satisfying prescribed boundary conditions. Only one of these solutions is minimizing. I tried two iterative methods, namely, Newton's Method and an Optimal Gradient Method. Our numerical experiments with these two methods were highly successful. In order to preserve these computations for future use, I wrote a second RAND Report in 1949 describing what we did. This report received a wide circulation among engineers and I received undo credit for devising these methods. Incidentally, with regard to the gradient method, I had to formulate an adequate definition of the concept of the gradient of an integral. To do so I introduced an inner product $\langle g, h\rangle$ on the space of variations. The gradient of $F(x)$ at a point x is a variation g such that

$$F'(x, h) = \langle g, h\rangle$$

for all admissible variations h. I found that the inner product usually used heretofore was unsatisfactory because elements of the form $x + ag$ were not admissible elements. However, I also found that there were a large class of inner products that were suitable for the problem at hand. These inner products need not be fixed but could vary with the element x with which were concerned. One such inner product is the inner product $F''(x; g, h)$ induced by the second variation of F. When this inner product is used, our gradient method becomes a version of Newton's method. Thus Newton's method can be viewed as a gradient method determined by a "preferred" inner product which varies at each step.

I also tried one other method, later called a penalty function method. In the simple case in which we minimize $f(x)$ subject to a constraint $g(x) = 0$ it proceeds as follows. Select a sequence $\{c_n\}$ converging to infinity. Obtain the minimizer x_n of the penalty function

$$F_n = f + c_n g^2.$$

Then, under favorable conditions, the sequence $\{x_n\}$ will converge to the minimizer x_0 of f subject to $g = 0$. Moreover, the sequence $\{2c_n g(x_n)\}$ converges to the Lagrange multiplier λ. In theory, this method is exellent and can be used effectively for theoretical purposes. Unfortunately, when I tried to solve a simple problem numerically by this method, I found that it had poor convergence properties due to round off errors and so I abandoned it for the time being. Besides, for variational problems with differential constraints, I knew that I would need to consider what is now known as relaxed controls and so would lead to a more complicated theory than I was willing to accept at that time. Later, at about 1969, I was invited to give a talk on computational procedures for solving optimization problems. It occurred to me at that time that a result in the folklore of Variational Theory could be used for this purpose. This result states that if x_0 minimizes $f(x)$ subject to $g(x) = 0$, then normally there is a multiplier λ and a constant c such that x_0 minimizes the function

$$F(x) = f(x) + \lambda g(x) + cg(x)^2$$

for all x near x_0 even when the constraint $g(x) = 0$ is not satisfied. Usually a relative small value of c is effective. Having chosen c, a suitable value of the multiplier λ can be found by an iterative procedure. The iteration that we shall use is obtained by observing that the gradient of F is given by the formula

$$F'(x) = f'(x) + [\lambda + 2cg(x)]g'(x).$$

This formula suggests the following iteration

Select an initial point x_1, an initial multiplier λ_1, and a suitable constant c. Having obtained x_1 and λ_1 find a minimizer x_{i+1} of the function

$$F_i(x) = f(x) + \lambda_i g(x) + cg(x)^2.$$

Then set

$$\lambda_{i+1} = \lambda_i + \alpha g(x_{i+1}) \qquad (\text{say } \alpha = 2c)$$

and repeat.

To obtain an initial estimate for x_1 and c, one can begin with the penalty function method. I called this method "a method of multipliers". This algorithm with some modifications proved to be an effective method for solving constrained minimum problems. An equivalent algorithm was also suggested by M.J.D. Powell at about the same time.

Return to the Summer of 1949. At that time I was invited to join the Institute for Numerical Analysis (INA) on a part time basis. In accepting this invitation I expected to pursue my studies of numerical methods in variational theory. However, I was diverted by Barkley Rosser who was the new director of INA. Rosser initiated a program of studying methods for solving linear equations and for finding eigenvalues and eigenvectors of matrices. He organized a seminar on this subject. The principal participants of this seminar were Barkley Rosser, George Forsythe, Cornelius Lanczos, Gertrude Blanch, Magnus Hestenes, William Karush, and Marvin Stein. Rosser and Forsythe specialized on finding solutions of linear equations. Forsythe, in particular, proceeded to classify known methods for such solutions. Hestenes, Karush, and Stein were chiefly responsible for the study of methods for finding eigenvalues and eigenvectors of matrices. Lanczos continued to refine his methods for solving eigenvalue problems. Blanch, who was in charge of numerical computations, acted as an advisor on numerical procedures. Of course, we did not limit ourselves to our specialties and participated actively on all the topics taken up in the seminar.

With regard to the study on solving linear equations, we specialized on iterative methods for solving linear equations. We did so in part because it appeared that they required less high speed storage than other methods. Besides we found them to be interesting. We surveyed the known methods both from a theoretical point of view and from a numerical point of view. In preparing the short history of INA, which I wrote with John Todd, I found a manuscript written by Rosser and myself developing a unified theory for a large class of methods. I had forgotten that we had written this article. A summary of the contents of this article is given in the history of INA which Todd and I wrote. In this paper we discussed various algorithms for solving a linear equation

$$Ax = h$$

where A is a nonsingular $n \times n$-matrix and h was a prescribed n-dimensional vector. We used the size of the residual

$$r = h - Ax = A(x_0 - x)$$

as a measure of the closeness of x to the solution x_0 of our equation. To measure the size of r, we sometimes used the largest component of r. At other times, with * denoting transpose, we used a function of the form,

$$f(x) = \tfrac{1}{2}r^*Kr = \tfrac{1}{2}x^*Bx - k^*x + c,$$

where K is a positive definite symmetric matrix and

$$B = A^*KA, \quad h = A^*Kh, \quad c = \tfrac{1}{2}h^*Kh.$$

The solution x_0 of $Ax = h$ minimizes f and solves the equation $Bx = k$. When A is a positive definite symmetric matrix we can choose $K = A^{-1}$. Then $B = A$, $k = h$, and

$$f(x) = \tfrac{1}{2}x^*Ax - h^*x + c$$

where c is an unknown constant which plays no role in our considerations. It should be noted that the minimizer x_0 of $f(x)$ is the center of the ellipsoid, $f(x) =$ constant. Thus, the problem of solving $Ax = h$ is equivalent to that of finding the center of an ellipsoid. We observed further that the minimum point x_2 of $f(x)$ on a line

$$x = x_1 + tp$$

was the midpoint of the chord in which this line intersected the ellipsoid $f(x) = f(x_1)$.

Although it was not immediately obvious, we found that the algorithms that we studied were equivalent to one of the following type

$$x_{i+1} = x_i - H_i(Ax_i - h) = x_i + H_i r_i \tag{1}$$

where r_i is the residual

$$r_i = h - Ax_i.$$

From this fact we concluded that, if

$$\mu = \limsup_{i \to \infty} \|I - H_i A\| < 1,$$

then the sequence $\{x_i\}$ converges linearly to the solution x_0 at the rate μ. In many cases the matrix H_i need not be constructed explicitly by the algorithm. For example, we can obtain x_{i+1} from x_i by a subroutine of the following type

> Choose m vectors $u_1, u_2, \ldots, u_m$ which span our space and selected vectors $v_1, v_2, \ldots, v_m$ such that $d_j = v_j^* A u_j$ is not zero for $j = 1, \ldots, m$. Select $y_1 = x_i$. Then, for $j = 1, \ldots, m$, set
>
> $$y_{j+1} = y_j + \alpha_j u_j, \quad \alpha_j = v_j^*(k - Ay_j)/d_j.$$
>
> Finally set $x_{i+1} = y_{m+1}$.

It can be shown that when x_{i+1} is obtained from x_i in this manner, then there is a matrix H_i such that equation (1) holds. In view of this result the Gauss Seidel method and a large number of other standard methods can be studied simultaneously by considering an algorithm of the form (1). A discussion of our considerations of this nature can be found in the History of INA which Todd and I wrote. We omit these considerations here. However, I would like to remark that in most of the numerical cases we considered convergence was very slow. We were therefore on the lookout for more rapidly convergent algorithms. We also considered the introduction of a relaxation constant β in our algorithm but did not develop an adequate theory for this case.

One of the algorithms that we tried was a gradient method for minimizing the error function

$$f(x) = \tfrac{1}{2}x^* Ax - h^* x$$

for the case when A is positive definite. The negative gradient of f is the residual $r = h - Ax$. Accordingly, the gradient algorithm is of the form

$$x_{i+1} = x_i + a_i r_i, \quad a_i = |r_i|^2/r_i + Ar_i,$$

where $r_i = h - Ax_i$ and $t = a_i$ is chosen to as to minimize $f(x_i + tr_i)$. We called this method, the optimal gradient method. Forsythe constructed a positive definite 6 × 6-matrix in a random fashion and proceeded to test the optimal gradient method numerically. He found that the method "bogged down" and that the solution could not be obtained using a reasonable number of steps. Accordingly he tried two different acceleration techniques. The first one used the relaxed equation

$$x_{i+1} = x_i + \beta a_i r_i,$$

where β is some number between 0 and 2. Values of β, such as 7, 8, and 1.2, were effective. Even $\beta = 0.2$ was better than $\beta = 1$. He also tried the following acceleration scheme suggested by Motzkin. When the algorithm bogged down he added an additional step of minimizing f along the line through x_{i-2} and x_i to obtain a new estimate x_{i+1}. This method was equally effective but somewhat more complicated to use. We discovered that

Aitken had used the second scheme earlier. Incidentally, this acceleration scheme yields one step of the conjugate gradient method described below.

Rosser returned to Cornell in the fall of 1950 and returned to INA for summer 1951 to pursue his studies of solutions of linear equations and to attend a Conference on "Solutions of Linear Equations and the Determination of Eigenvalues" to be held at INA in August 1951. In June or July 1951, after almost two years of studying algorithms for solving systems of linear equations, we finally "hit" upon a conjugate gradient method. I had the privilege of first formulating this new method. However, it was an outgrowth of my discussions with my colleagues at INA. In particular, my conversations with George Forsythe had a great influence on me. During the month of July 1951, I wrote an INA Report on this new development. When E. Stiefel arrived at INA in August to attend the conference on Solutions of Linear Equations, he was given a copy of my paper. Shortly thereafter he came to my office and said about the paper "this is my talk". It occurred that he too had invented the Conjugate Gradient Algorithm and had carried out successful experiments using this algorithm. Accordingly, I invited Stiefel to remain at UCLA and INA for one semester so that we could write an extensive paper on this subject. In the meantime C. Lanczos observed that the Conjugate Gradient Method could be derived from his algorithm for finding eigenvalues of matrices. In view of these remarks we see that there are three persons who are credited for inventing the Conjugate Gradient Method, namely, Stiefel, Hestenes, and Lanczos. However, as remarked above, this algorithm was an outgrowth of the program at INA on Solutions of Linear Equations originated by J.B. Rosser and participated upon by various members of INA, such as, G. Forsythe, W. Karush, T. Motzkin, L. Paige and M. Stein. Of these researchers, Forsythe was the most active in supplying numerical experiments for the algorithms discussed by the group. It was my privilege to invent the name "Conjugate Gradient Routine" for the new algorithm we had constructed.

The Conjugate Gradient Algorithm is based on the following property of ellipsoids:

The midpoints of parallel chords of an $(n-1)$-dimensional ellipsoid E_{n-1} lies on a $(n-1)$-plane π_{n-1} passing through the center x_0 of E_{n-1}. The $(n-1)$-plane π_{n-1} and the vectors in π_{n-1} are said to be conjugate to these chords.

Analytically, an ellipsoid E_{n-1} is the set of points x satisfying an equation of the form

$$f(x) = \tfrac{1}{2}x^*Ax - h^*x = \text{constant} \qquad (A^* = A > 0).$$

The minimizer x_0 of f is the center of E_{n-1} and solves the equation

$$Ax = h.$$

Parallel chords of E_{n-1} have a common direction vector p. A midpoint x of one of these chords minimizes f along this chord. It follows that the negative gradient

$$r = h - Ax = A(x_0 - x)$$

at such a midpoint x is orthogonal to p. That is,

$$p^*r = p^*(h - Ax) = p^*A(x_0 - x) = 0$$

or, equivalently,

$$p^*Ax = p^*h.$$

This equation represents an $(n-1)$-plane π_{n-1} through the center x_0 of E_{n-1}. Its normal is the vector Ap. Every vector q in π_{n-1} is orthogonal to Ap and is conjugate to p. The relation

$$p^*Aq = 0$$

therefore expresses the conjugacy of two vectors p and q.

Let us apply this result to the 2-dimensional case. We seek to find the center of an ellipse. Referring to Figure 1 let x be a point on an ellipse E. Let p be a vector tangent to E at x and let r be an inner normal of E at x. Through the tip y of r, draw a chord uv perpendicular to r. Let $z = y + bp = \frac{1}{2}(u + v)$ be the midpoint of this chord. Denote the vector joining x to z by p_c. Then $p_c = z - x = r + bp$. The vector p_c is conjugate to p. The midpoint x_c of the chord emanating from x in the direction p_c is the center of the ellipse E. The point x_c also minimizes the function

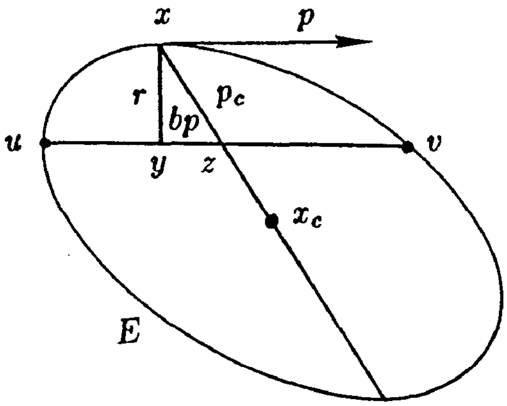

Figure 1

$$f(x) = \tfrac{1}{2}x^*Ax - h^*x \qquad (A^* = A > 0)$$

on this 2-dimensional space, where $f(x) =$ constant is an analytical representation of E. The geometric construction of the minimizer x_c of f can be carried out analytically as follows:

Choose a point x and compute $r = h - Ax$. Let p be a vector orthogonal to r. Compute

(2a) $$p_c = r + bp, \qquad b = -p^*Ar/p^*Ap$$

(2b) $$x_c = x + ap_c, \qquad a = p_c^*Ar/p_c^*Ap_c.$$

The point x_c minimizes $f(x)$ on our 2-plane.

This result leads us to the conjugate gradient routine. We shall give several versions of the conjugate gradient algorithm (cg-algorithm) for solving the equation

$$Ax = h,$$

where A is a positive definite symmetric matrix. The first of these is the formulation given independently by Stiefel and by Hestenes. It proceeds as follows.

Cg-algorithm I

Initial step. Select a point x_1 and compute

$$p_1 = r_1 = h - Ax_1 \tag{3a}$$

$$c_1 = p_1^* r_1, \quad d_1 = p_1^* A p_1, \quad a_1 = c_1/d_1 \tag{3b}$$

$$x_2 = x_1 + a_1 p_1, \quad r_2 = h - Ax_2 = r_1 - a_1 A p_1. \tag{3c}$$

Iterative steps. Having obtained p_{i-1}, d_{i-1}, x_i, r_i compute

$$p_i = r_i + b_{i-1} p_{i-1} \quad \text{with} \quad b_{i-1} = p_{i-1}^* A r_i / d_{i-1} \tag{3d}$$

$$c_i = p_i^* r_i, \quad d_i = p_i^* A p_i, \quad a_i = c_i/d_i \tag{3e}$$

$$x_{i+1} = x_i + a_i p_i, \quad r_{i+1} = h - Ax_{i+1} = r_i - a_i A p_i. \tag{3f}$$

Terminate when $r_{m+1} = 0$. Then $x_0 = x_{m+1}$ solves $Ax = h$.

In this algorithm the length of the vector p_i is not important. We can therefore, if we wish, introduce a scale factor σ_i for p_i. When this is done our formulas for these vectors take the form

$$p_1 = \sigma_1 r_1, \quad p_{i+1} = \sigma_{i+1}(r_{i+1} + b_i p_i). \tag{4}$$

The scaling $\sigma_1 = 1$, $\sigma_{i+1} = (1+b_i)^{-1}$ is particularly useful because then $p_{i+1} = h - Ay_{i+1}$ at a point y_{i+1} on the line segment joining x_i to x_{i+1}. Alternatively, we can use generalized gradients in which we have the formulas

$$(5) \qquad p_1 = Hr_1, \quad p_{k=1} = Hr_{i=1} + b_i p_i, \quad b_i = -p_i^* Hr_{i+1}/d_i,$$

where H is a positive definite symmetric matrix. When these equations are used we call our algorithm a generalized *cg*-algorithm. A discussion of these and other variants of the *cg*-algorithm can be found in my book on *Conjugate Direction Methods in Optimization.*

Return to the *cg*-algorithm (3a)-(3f). Observe that equations (3d)-(3f) can be obtained from equations (2a) and (2c) by setting $x = x_i$, $r = r_i = h - Ax_i$, $p = p_{i-1}$, $p_c = p_i$, and $x_c = x_{i+1}$. It follows that the point x_{i+1} minimizes $f(x)$ on the 2-plane

$$x = x + \alpha r_i + \beta p_{i-1}.$$

This 2-plane is also determined by the points x_{i-1}, x_i, and $y_{i+1} = x_i + r_i$. The point x_{i+1} therefore minimizes $f(x)$ on this 2-plane and is the center of the ellipse in which this 2-plane cuts the ellipsoid $f(x) = f(x_i)$. Stiefel considered the direction p_i to be a relaxation of the direction r_i.

In view of this result we have the following alternative description of *Cg*-algorithm I.

Cg-algorithm II

Initial step. Choose x_1 and compute $r_1 = h - Ax_1$. Then find the minimizer x_2 of $f(x)$ on the line through x_1 and $y_2 = x_1 + r_1$.

Iterative steps. For $i = 2, 3, \ldots$, compute $r_i = h - Ax_i$ and find the minimum point x_{i+1} of $f(x)$ on the 2-plane through the points x_{i-1}, x_i, and $y_{i+1} = x_i + r_i$. The point x_{i+1} is the center of the ellipse in which the 2-plane cuts the ellipsoid defined by $f(x) = f(x_i)$.

Terminate when $r_{m+1} = 0$. Then x_{m+1} solves $Ax = h$.

When this algorithm is put in analytic form we obtain the following set of equations with x_1 as the initial point.

$$r_1 = h - Ax_1, \quad x_2 = x_1 + d_i r_i, \quad r_2 = r_1 - \alpha_1 A r_1 \tag{6a}$$

$$x_{i+1} = (x_i + \alpha_i r_i - \beta_{i-1} x_{i-1})/(1 - \beta_{i-1}), \quad i > 1 \tag{6b}$$

$$r_{i+1} = (r_i - \alpha_i A r_i - \beta_{i-1} r_{i-1})/(1 - \beta_{i-1}), \quad i > 1 \tag{6c}$$

$$\alpha_i = |r_i|^2 / r_i^* A r_i, \quad \beta_{i-1} = r_{i-1}^* (r_i - \alpha_i A r_i)/|r_i|^2. \tag{6d}$$

The scalars α_i and β_{i-1} are chosen so that r_{i+1} is orthogonal to r_i and r_{i-1}. With x_1 as the initial point, algorithms (3a)-(3f) and (6a)-(6d) generate the same points $x_2, x_3, \ldots$. Algorithm (6a)-(6d) can be found in the original paper by Hestenes. It is sometimes called GRADIENT PARTAN.

It can be shown that the point x_{i+1} minimizes $f(x)$ on the i-plane determined by the points $x_1, x_2, \ldots, x_i$, and $y_{i+1} = x_i + r_i$. This i-plane can be represented parametrically by the equation

$$x = x_1 + \gamma_1(x_2 - x_1) + \cdots + \gamma_{i-1}(x_i - x_{i-1}) + \gamma_i r_i.$$

It can be shown that, for the minimizer x_{i+1}, we have $\gamma_1 = \gamma_2 = \cdots = \gamma_{i-2} = 1$. It follows that x_{i+1} lies in the 2-plane

$$x = x_{i-1} + \gamma_{i-1}(x_i - x_{i-1}) + \gamma_i r_i$$

and so minimizes $f(x)$ on this 2-plane. In view of this result, Cg-algorithm II is equivalent to the following

Cg-algorithm III

Initial step. Select x_1 and compute $r_1 = h - Ax_1$. Find the minimizer x_2 of $f(x)$ on the line through x_1 and $x_1 + r_1$.

Iterative steps. For $i = 2, 3, \ldots$, find the minimizer x_{i+1} of $f(x)$ on the i-plane determined by the points $x_1, x_2, \ldots, x_i$, $y_{i+1} = x_i + r_i$, where $r_i = h - Ax_i$.

Terminate when $r_{m+1} = 0$. The point x_{m+1} solves $Ax = h$.

The cg-algorithm can also be interpreted geometrically as described in the following

Cg-algorithm IV

We seek to find the center of the $(n-1)$-dimensional ellipsoid E_{n-1} defined by the equation $f(x) = f(x_1)$. The point x_1 is on E_{n-1}. Let C_1 be a chord of E_{n-1} emanating from x_1 in the direction of the inner normal of E_{n-1} at x_1. Find the midpoint x_2 of C_1. The $(n-1)$-plane π_{n-1} through x_2 conjugate to C_1 contains the center x_0 of E_{n-1}. If $x_2 - x_0$ we are done. Otherwise π_{n-1} intersects the ellipsoid $f(x) = f(x_2)$ in a $(n-2)$-dimensional ellipsoid E_{n-2} whose center is also x_0. Accordingly, we have reduced the dimension of our space of search by 1. We now repeat the process and select a chord C_2 of E_{n-2} emanating from x_2 in the direction of the inner normal of E_{n-2} at x_2. Find the midpoint x_3 of C_2. The $(n-2)$-plane π_{n-2} in π_{n-1} conjugate to C_2 contains the common center x_0 of E_{n-2} and E_{n-1}. If $x_3 = x_0$ we are done. Otherwise, π_{n-2} intersects the ellipsoid $f(x) = f(x_3)$ in a $(n-3)$-dimensional ellipsoid E_{n-3} whose center is also x_0. Again we have reduced the dimension of our space of search by 1. Proceeding in this manner we finally obtain a chord C_m of an $(n-m)$-dimensional ellipsoid E_{n-m} whose midpoint is x_0 thereby completing our search for the center of E_{n-1}.

The following analytic version of Cg-method IV led to the name *Conjugate gradient algorithm.*

Cg-algorithm V

Starting with a point x_1 find the direction p_1 of steepest descent of $f(x)$ at x_1. Proceed in the direction p_1 to the point x_2 at which $f(x)$ has a minimum value. Let π_{n-1} be the $(n-1)$-plane through x_2 conjugate to p_1. Find the direction p_2 of steepest descent at x_2 of $f(x)$ in π_{n-1}. Proceed from x_2 in the direction p_2 until a point x_3 is reached at which $f(x)$ has a minimum. Let π_{n-2} be the $(n-2)$-plane in π_{n-1} conjugate to p_2 (and hence also conjugate to p_1). Find the direction of steepest descent p_3 at x_3 of $f(x)$ in π_{n-2} and proceed to the minimum point x_4 of $f(x)$ in this direction. Proceeding in this manner we finally reach the minimum point x_0 of $f(x)$ in our original n-space. It is the solution of $Ax = h$.

We call $p_1, p_2, \ldots$ and their multiples "conjugate gradients" of $f(x)$. Except possibly for a positive scale factor, they are the vectors $p_1, p_2, \ldots$ generated by Cg-algorithm I.

There is another version of the cg-algorithm which is of interest. In this algorithm we alternate minimizations of the functions

$$f(x) = \tfrac{1}{2}x^*ax - h^*x, \quad g(x) = \tfrac{1}{2}|r|^2 = \tfrac{1}{2}|h - Ax|^2.$$

It proceeds in the manner described in the following

Cg-algorithm VI

Select a point x_1. Set $y_1 = x_1$ and compute $p_1 = -f'(y_1)$. Having obtained x_i, y_i, and $p_i = -f'(y_i)$, find the minimum point x_{i+1} of $f(x)$ on the line $x = x_i + tp_i$. Next determine the minimum point y_{i+1} of $g(x)$ on the line joining y_i to x_{i+1}. Compute $p_{i+1} = -f'(y_{i+1})$. Terminate when $x_{m+1} = y_{m+1}$ or equivalently when $f'(y_{m+1}) = 0$. The point $x_{m+1} = y_{m+1}$ is the minimum point of $f(x)$ and solves the equation $Ax = h$.

It is also of interest to note that the conjugate gradient algorithm can be put in the form (1) with H_i replaced by a_iH_i. We then have the iteration

$$x_{i+1} = x_i + a_iH_ir_i, \quad r_i = h - Ax_i$$

where H_i is a positive definite symmetric matrix. We adjoin to this an updating procedure for the matrix H_i. It has the property that $H_{n+1} = A^{-1}$. This form of the conjugate gradient algorithm is due to Davidon, who fashioned it so as to be applicable to nonlinear equations. It was modified later by Fletcher and Powell. It is now called the Davidon-Fletcher-Powell method or the variable metric method. There are several versions of this algorithm. The one that we shall present is the following

Cg-algorithm VII

Let H be a positive definite matrix. Set $H_1 = H$ and perform the following iteration with x_1 as the initial point and $r_1 = h - Ax_1$.

$$p_i = H_i r_i, \quad s_i = Ap_i, \quad q_i = H_i s_i, \quad d_i = p_i^* s_i, \tag{7a}$$

$$\delta_i = q_i^* s_i, \quad e_i = \delta_i / d_i, \quad c_i = p_i^* r_i, \quad a_i = c_i / d_i \tag{7b}$$

$$x_{i+1} = x_i + a_i p_i, \quad r_{r+1} = r_i - a_i s_i = h - Ax_{i+1} \tag{7c}$$

$$H_{i+1} = H_i - (p_i q_i^* + q_i p_i^*)/d_i + (e_i + 1) p_i p_i^* / d_i. \tag{7d}$$

Terminate when $r_{m+1} = 0$. Then x_{m+1} solves $Ax = h$. If $m = n$, we have $H_{n+1} = A^{-1}$.

Under perfect computations we have the relations

$$p_1 = Hr_1, \quad p_{i+1} = Hr_{i+1} + b_i p_i$$

so that Algorithm (7a)-(7d) is equivalent to the generalized cg-algorithm and is equivalent to Algorithm (3a)-(3f) when $H = I$. It involves more computations than the original algorithm. However, it is within it a built-in correction of roundoff errors and so usually gives better results than the original cg-algorithm when the matrix A is ill-conditioned. Extensions of this algorithm have been useful in the minimization of nonquadratic functions. There are many variations of the updating formula (7d) for H_i. For example, one can add nonnegative multiples of the matrix

$$(e_i p_i - q_i)(e_i p_i - q_i)^*$$

to H_{i+1} with altering its basic properties.

We have given seven versions of the cg-algorithm. Additional versions can be found in my book on *Conjugate Direction Methods in Optimization.* One of the first five versions given above was the original version of cg-algorithm developed at INA. I believe that it was either *Cg*-algorithm IV or *Cg*-algorithm V but I am not certain about this. It could have been *Cg*-algorithm III or II because, at that time, Forsythe and I were experimenting with algorithms for minimizing $f(x)$ on i-planes for $i = 2, 3, \ldots$.

In the application of the *cg*-algorithm, it is often desirable to precondition the matrix A before applying the *cg*-algorithm. Also the *cg*-algorithm is sometimes used in conjunction with other algorithms for solving linear equations.

Cg-algorithm I has within it an algorithm for computing the characteristic polynomial of A. One needs only replace A by λ. This algorithm is equivalent to one developed earlier by C. Lanczos. It follows that the algorithm of Lanczos for finding eigenvalues implicitly contains the *cg*-algorithm although none of us recognized this fact in the seminar we conducted. When Lanczos became aware of this feature of his algorithm, he formulated an alternative version of the *cg*-algorithm which he called a "Method of Minimized Iterations". The connections between his algorithm and the original *cg*-algorithm can be found in the historical account of INA which I wrote with J. Todd.

The *cg*-algorithm has some useful properties. At each step the value of the error function $f(x)$ is diminished. So also is the distance of our estimate x_i from the solution x_0. This latter property may fail when generalized gradients are used. If A has multiple eigenvalues, the algorithm will terminate in less than n steps. It follows that if A has clustered eigenvalues, a good estimate of the solution is obtained early. A discussion of these and other properties of the *cg*-algorithm can be found in the original paper by Stiefel and Hestenes and in my book entitled *Conjugate Direction Methods in Optimization* published by Springer in 1980. We also discussed the problem of finding least square solutions for a general equation $Ax = h$ in which A may be nonsymmetric and singular. There is a vast literature on *cg*-algorithms and Lanczos' algorithms. References can be found in my book and in a recent paper by Gene Golub and Dianne O'Learly entitled *Some History of the Conjugate Gradient and Lanczos Algorithms* 1948-1976. This excellent paper has been submitted to the SIAM Review.

As I remarked earlier, in our seminar I was responsible for studies of methods for obtaining eigenvalues of a matrix A. We developed a gradient method for finding the eigenvalues of a symmetric matrix. It turned out that this method could be viewed as a generalization of the power method. Of course, we studied the power method and the inverse power method. We also considered the Jacobi method but did not have the computing facilities for a serious study of this method numerically. In addition we considered the problem of finding singular values of matrices. Our studies complimented the studies of Lanczos for finding eigenvalues of matrices.