A Crude-Search Davidon-Type Technique with Application to Shuttle Optimization

William F. Powers*
The University of Michigan, Ann Arbor, Mich.

A parameter optimization scheme devised by Fletcher which does not require an elaborate one-dimensional search has been modified to make it an effective scheme for trajectory optimization. The method involves an updating matrix, H_{J+1}, which is defined by either the classical Davidon formula or a well-defined formula which satisfies $H_{J+1}\,\Delta x_J = \Delta g_J$ and is the basis of a new variable metric method due to Broyden. A rule involving a scalar product determines which formula should be used to avoid the tendency of the H-matrix to become singular or unbounded. The method is applied, along with the Davidon and Broyden methods, to various shuttle ascent optimization problems.

Introduction

IN the past decade, a large amount of research has been devoted to the development of parameter optimization schemes which blend the advantages of both the gradient and Newton methods, while minimizing their disadvantages. That is, attempts have been made to develop methods which require only first-order information and are stable, i.e., guarantee a decrease in the cost at each iteration (gradient method properties), and which have rapid convergence in a suitably small neighborhood of the solution (a Newton method property). Most of the schemes involve modifications or extensions of the pioneering efforts reported in Ref. 1 [the conjugate gradient (CG) method] and Refs. 2 and 3 [the Davidon-Fletcher-Powell (DFP) method or variable metric method].

Some of the notable extensions of the CG and DFP methods are contained in Refs. 4–11. References 1–5 require a one-dimensional (1-D) search subprogram to determine the length of the correction vector. Experience has shown that the development of an effective 1-D search in these algorithms is a very critical and costly part of the scheme. Because the one-dimensional curves may vary qualitatively from problem to problem (and iterate to iterate), the 1-D search subprogram requires considerable programing effort if it is to be applicable to a large class of problems.

In the last few years, iterative schemes have been developed which replace the elaborate 1-D search with a crude, easy-to-program 1-D search.[6–10] Of course, to give up the 1-D search, some desirable property must be forsaken. In Refs. 7–10, guaranteed convergence in at most n iterations ($n =$ number of parameters) for a quadratic function is lost; however, quadratic convergence is obtained in at most $n+1$ steps in Refs. 7. 8, and 10, and in at most $n+2$ steps in Ref. 9. Furthermore, the method of Ref. 9 will converge any homogeneous function in at most $n+2$ iterations.

In Table 1 a concise summary of the main methods which do not require an elaborate 1-D search is presented. The methods of Refs. 7, 8, and 10, and this paper resemble the DFP method in that if

$$f(x_1, \ldots x_n)$$

is to be minimized, then the iteration scheme is defined by the update formula

$$x^{(k+1)} = x^{(k)} - \alpha_k H_k g^{(k)}$$

where $x^{(k)} \equiv$ current value of the vector x, $x^{(k+1)} \equiv$ new value of x, $\alpha_k \equiv$ a scalar parameter (the 1-D search parameter), $H_k \equiv$ an $n \times n$ matrix which is updated on each iterate, $g^{(k)} \equiv f_x[x^{(k)}] \equiv$ the gradient of f evaluated at $x^{(k)}$. A particular scheme is defined by the way that it updates H_k and α_k.

As noted in Table 1, the method of this paper does not possess the finite convergence property for quadratic functions. At first glance this appears to make the scheme noncompetitive with the other methods. However, it is precisely this property which is sacrificed to eliminate the need for the 1-D search. The reasons for this decision will be discussed in a later section.

An aerospace optimization problem of current interest is the space shuttle trajectory optimization problem. Although this is a natural function space problem (i.e., the problem is described by differential equations with controls which are unknown functions of time to be determined), because of the many shuttle configurations it is probably best to represent the problem as a parameter optimization problem until a final design is decided upon. In this way, the vehicle may be changed without greatly affecting the optimization program, and in addition, important design parameters may be included in the optimization without modifying the iteration procedure. With a function space method, adjoint or Euler-Lagrange equations, which are strongly problem dependent, must be determined. This process is tedious and time consuming when aerodynamic forces are present, and when the problem is highly constrained, as is the shuttle trajectory optimization problem.

In Refs. 12–14, shuttle ascent trajectories are optimized by representing the steering angle rate as a sequence of straight line segments, and then applying the DFP parameter optimization technique. The technique performed better than most established parameter and function space methods on a particularly difficult shuttle test problem described in Ref. 15.

The initial goal of this research was to determine the relative performance capabilities of the DFP method described in Refs. 12–14 and one of the techniques which does not require a 1-D search. In addition to this comparison, a relatively new method due to Broyden[16] which requires a 1-D search was also considered since it is related to the technique of this paper.

Presented as Paper 72-907 at the AIAA/AAS Astrodynamics Meeting, Palo Alto, Calif., September 11–12, 1972; submitted December 14, 1972; revision received July 31, 1973. This research was supported by the National Science Foundation under Grant GK-30115. The work was initiated at the NASA Johnson Space Center, Mission Planning and Analysis Division. The author wishes to thank I. L. Johnson, NASA Johnson Space Center, and J. L. Kamm, TRW-Systems (Houston), for many valuable discussions concerning the Davidon-Fletcher-Powell method and their computer program, PEACE.

Index categories: LV/M Trajectories; Spacecraft Navigation, Guidance, and Flight-Path Control Systems.

[*] Associate Professor, Department of Aerospace Engineering. Member AIAA.

Table 1 Summary of parameter optimization methods without a 1-D search

	Property				
Method	No. of steps for quadratic convergence	Stability guaranteed $(f_{J+1} \leq f_J)$	Conjugacy for quadratic function $(p_i^T Apj = 0,\ i \neq j)$	Reset philosophy	Method for choosing step-length
Ref. 7	$n+1$	NO	NO[a]	Two well-defined tests	Choose α_J so that $f(x_J) - f(x_{J+1}) \geq \epsilon \alpha_J g_J^T H_J g_J > 0$ is satisfied $(0 < \epsilon < 1$ given)
Ref. 8 and method of this paper	∞	YES	NO[a]	None proposed	Crude search for function decrease; never go to 1-D search.[b] Sometimes to 1-D seach.[c]
Ref. 9	$n+2$	YES	NO	Well defined; based on too-small step or linear dependence problems	Well-defined; select from 2 values; if no function decrease go to 1-D search
Ref. 10	$n+1$	YES	YES	None proposed	Golden section bracketing procedure

[a] Yes if a 1-D search is employed.
[b] Method of this paper.
[c] Method of Ref. 8.

To insure a fair comparison, the basic computer program used in Refs. 12–14 was used with all techniques. The results of the comparisons are discussed in Sec. 4.

2. The Parameter Optimization Algorithm

As noted in the Introduction, the fact that Fletcher's Method[8] does not converge a quadratic function in a finite number of steps might cause one to assume that the method is not competitive with the other methods of Table 1. However, if one surveys the major parameter optimization research papers, one finds that there exist numerous opinions as to what is the most important element in the class of algorithms of Refs. 1–10 for general nonlinear functions. For example: the method should be stable on every iterate; a precise 1-D search is critical; the directions should be conjugate; the method should converge a quadratic function in a finite number of iterates; resetting is necessary for convergence. In any case, for general nonlinear functions, only numerical simulations have caused such statements and not mathematical proofs.

The purpose of this paper is not to compare the methods of Table 1, but instead to present a modification of the algorithm of Ref. 8 and to compare its performance on difficult nonlinear problems with two techniques that require 1-D searches, especially the DFP method. First the algorithm will be stated, and then the underlying theory will be discussed in Sec. 3. The notation $\Delta(\ )_J \equiv (\ )_{J+1} - (\ )_J$ is employed below.

1) Specify x_0, α_0, H_0, μ. Calculate $f_0 \equiv f[x_0]$, $g_0 \equiv g[x_0]$; set $J = 0$. (H_0 is an arbitrary symmetric, positive-definite matrix and $\alpha_0 > 0$.)

2) Calculate $f_{(J+1)1} \equiv f[x_J - \alpha_{J1} H_J g_J]$, where α_{J1} is the current estimate of α_J, and check $f_{(J+1)1} < f_J$. If yes, go to 3 if $\alpha_{J1} = 1$ or go to a crude step-size increase package if $\alpha_{J1} \neq 1$; if no, go to a crude stepsize decrease package. (The particular packages used for the algorithm of this paper are listed in the Appendix.) The resultant step-size is denoted by α_J.

3) If $J = 0$, go to 5. If $J > 0$, check $\Delta f_J/(g_J^T \Delta x_J) \geq \mu$. If yes, go to 4; if no, decrease α_J until the inequality is satisfied and then go to 4.

4) Check $10^2 g_J^T \Delta x_J \leq g_{J+1}^T \Delta x_{J-1}$. If yes, go to 5; if no, increase α_J until the inequality is satisfied and then go to 5. (The particular α_J increase package for this paper is described in the Appendix.)

5) Calculate $x_{J+1} = x_J - \alpha_J H_J g_J$ and g_{J+1}. Check $\Delta g_J^T \Delta x_J > 0$. If yes, go to 6. If no, increase α_J.

6) Check $\Delta g_J^T \Delta x_J \geq \Delta g_J^T H_J \Delta g_J$. If yes, go to 8; if no, go to 7.

7) Calculate:

$$H_{J+1} = H_J + \frac{\Delta x_J \Delta x_J^T}{\Delta x_J^T \Delta g_J} - \frac{H_J \Delta g_J \Delta g_J^T H_J}{\Delta g_J^T H_J \Delta g_J} \tag{1}$$

Go to 9.

8) Calculate:

$$H_{J+1} = H_J - \frac{\Delta x_J \Delta g_J^T H_J}{\Delta x_J^T \Delta g_J} - \frac{H_J \Delta g_J \Delta x_J^T}{\Delta x_J^T \Delta g_J} + \left(1 + \frac{\Delta g_J^T H_J \Delta g_J}{\Delta x_J^T \Delta g_J}\right)\left(\frac{\Delta x_J \Delta x_J^T}{\Delta x_J^T \Delta g_J}\right) \tag{2}$$

Go to 9.

9) If $\alpha_J < 1$, set $\alpha_{J+1} = \alpha_J$; otherwise, set $\alpha_{J+1} = 1$. Set $J = J + 1$ and go to 2.

From the algorithm above one can see that $\alpha_J = 1$ is the desired value of the stepsize, and in the terminal iterations of the scheme $\alpha_J \to 1$ if the scheme is behaving like Newton's method as desired. Fletcher[8] bases most of his discussion on the $\alpha_J = 1$ case and devotes little attention to the $\alpha_J \neq 1$ case. In the trajectory optimization problems of this paper, the $\alpha_J \neq 1$ case occurs more often than not because of the difficulty of the problem and the use of finite difference formulas for the gradient calculations. Thus, more details about the $\alpha_J \to 1$ case have been included in this section and the Appendix than in Ref. 8.

3. Theoretical Basis

The algorithm of Sec. 2 is basically a scheme for choosing between two formulas for the H_{J+1}-matrix while preserving a reasonable step-size. Either Eq. (1) or (2) is used to define H_{J+1}. Equation (1) is the classical DFP formula, a rank-two formula. Equation (2) is also a rank-two formula which has been studied in its own right in Refs. 16 and 17. The fact that Eq. (2) is rank-two may be seen by rewriting it as

$$H_{J+1} = \left(I - \frac{\Delta x_J \Delta g_J^T}{\Delta x_J^T \Delta g_J}\right) H_J \left(I - \frac{\Delta g_J \Delta x_J^T}{\Delta x_J^T \Delta g_J}\right) + \frac{\Delta x_J \Delta x_J^T}{\Delta x_J^T \Delta g_J} \tag{3}$$

In Ref. 16, Broyden shows that both Eqs. (1) and (2) are members of his one-parameter class of formulas introduced in Ref. 5, and that both satisfy the "quasi-Newton property," $H_{J+1}\Delta x_J = \Delta g_J$. Equation (1) results by choosing his β_J-parameter to be zero while Eq. (2) results if $\beta_J = 1/(\Delta g_J{}^T \Delta x_J)$. Broyden noted that in numerical experiments comparing the use of Eqs. (1) and (2) separately and with a 1-D search that the algorithms had similar characteristics in the early stages but quite different characteristics in the terminal stages. This behavior is explained by the fact that $\beta_J = 1/(\Delta g_J{}^T \Delta x_J)$ may be near zero in the early stages of the algorithm because the gradients may be relatively large (where $|\Delta g_J{}^T \Delta x_J| = |g_{J+1}{}^T \Delta x_J - g_J{}^T \Delta x_J| = |g_J{}^T \Delta x_J|$ in a quasi-Newton scheme which employs a 1-D search). Since Eq. (2) and a 1-D search for the DFP program were required in the simulations, it was an easy task to also obtain simulations of Broyden's new method, i.e., Eq. (2) with a 1-D search. These results are also presented in Sec. 4.

Before considering Fletcher's justification for the basic algorithm, mention should be made of the occurrence of Eqs. (1) and (2) in Ref. 17. Since numerous updating formulas for the H-matrix have been proposed in the past decade, Greenstadt[18] considered the problem of choosing the "best" update formulas subject to appropriate constraints (e.g., symmetry and finite-convergence for a quadratic with a 1-D search). After investigating a number of performance indices, Greenstadt found that the following optimization problem gave tractable results:

$$\text{Minimize: } F(\Delta H_J) = Tr(W\Delta H_J W \Delta H_J{}^T) \tag{4}$$

$$\text{Subject to: } \Delta H_J{}^T = \Delta H_J \qquad \text{(symmetry)} \tag{5}$$

$$\Delta H_J \Delta g_J = \Delta x_J - H_J \Delta g_J \qquad \text{(quasi-Newton)} \tag{6}$$

where $Tr(\ ) \equiv$ trace of ( ) and W is an arbitrary matrix to be specified. The expression obtained for ΔH_J by solving this minimization problem involves the arbitrary matrix W. Goldfarb[17] found that $W^{-1} = H_{J+1}$ results in Eq. (2) and $W^{-1} = H_{J+1} - (\Delta g_J{}^T \Delta x_J)(H_J \Delta g_J{}^T H_J)/(\Delta g_J{}^T H_J \Delta g_J)^{3/2}$ results in Eq. (1), the DFP formula. He also showed that $z^T(\Delta H_J|_1 - \Delta H_J|_0)z \geq 0$, where $z \equiv$ arbitrary n-vector, $\Delta H_J|_1 \equiv H_{J+1} - H_J$ in Eq. (2), and $\Delta H_J|_0 \equiv H_{J+1} - H_J$ in Eq. (1). This means that Eq. (2) is less likely to tend toward singularity while Eq. (1) is less likely to tend toward unboundedness. Fletcher[8] obtained a similar result by a different argument, and this forms the basis of his algorithm.

Let us now consider Fletcher's method.[8] Denote the formula of Eq. (1) by H_0 and the formula of Eq. (2) by H_1. Let ϕ be a scalar parameter and define the linear combination

$$H_\phi \equiv (1 - \phi)H_0 + \phi H_1 \tag{7}$$

It is shown in Ref. 8 that if $\phi \varepsilon [0,1]$, then H_ϕ possesses the following property: If $f(x)$ is a quadratic function with $G \equiv [f_{x_i x_j}]$ positive definite, then the eigenvalues of $G^{1/2} H_\phi G^{1/2}$ (arranged in order) tend monotonically to one for any sequence of vectors Δx (i.e., H_ϕ tends to the inverse Hessian G^{-1} in a certain sense). Note that the property does not require a 1-D search. In addition to this property, it is shown that if $\phi \notin [0,1]$, then H^{-1} may diverge from G.

Since Eq. (7) represents an infinity of formulas, if it is to be useful there must exist a rule for selecting which value of $\phi \varepsilon [0,1]$ to use on a given iterate. Fletcher presents such a scheme by noting that a typical pitfall in the classical Davidon method is the tendency of the updating matrix H to become either singular or unbounded. He shows that if $\phi > \phi'$, then the eigenvalues $(\lambda_1 \ldots, \lambda_n)$ of H_ϕ, $H_{\phi'}$ (arranged in ascending order) are such that $\lambda_i(\phi) \geq \lambda_i(\phi')$ $(i = 1, \ldots, n)$, which implies $H_1 \equiv H_\phi|_{\phi=1}$ is "less singular" than $H_0 \equiv H_\phi|_{\phi=0}$, and H_0 is "less unbounded" than H_1. Thus, a simple test for nearness to singularity would indicate whether to use H_0 or H_1, which are the extreme elements of the class H_ϕ, $\phi \varepsilon [0,1]$. Fletcher shows that

$$\phi = \hat{\phi} \equiv \Delta g^T \Delta x/(\Delta g^T \Delta x - \Delta g^T H \Delta g)$$

defines the "rank one" formula

$$H_{J+1} = H_J + \frac{(\Delta x_J - H_J \Delta g_J)(\Delta x_J - H_J \Delta g_J)^T}{\Delta g_J{}^T(\Delta x_J - H_J \Delta g_J)} \tag{8}$$

The interesting thing about this formula is that if $\Delta g^T \Delta x > 0$, then $\hat{\phi} \notin [0,1]$, and the formula does not restrict the eigenvalues of H in any way. Thus, one can use the rank one formula to indicate which value of $\phi \in \{0,1\}$ should be used by simply checking the sign of $\Delta g^T \Delta x - \Delta g^T H \Delta g$; that is, if $\Delta g^T \Delta x > 0$ is enforced, then $\Delta g^T \Delta x - \Delta g^T H \Delta g > 0$ implies $\phi > 1$ (which means H_1 should be used) and $\Delta g^T \Delta x - \Delta g^T H \Delta g < 0$ implies $\phi < 0$ (which means H_0 should be used). If $\Delta g^T \Delta x - \Delta g^T H \Delta g = 0$, then H_1 is used to avoid singularity. Note that this test is step 6 of Sec. 2.

The only other steps in the algorithm of Sec. 2 which need to be discussed are steps 3 and 4. Step 3 is a check to determine if the stepsize is so large that an unreasonably small decrease in the function is attained. That is,

$$f_{J+1} = f_J + g_J{}^T \Delta x_J + O(\Delta^2) \tag{9}$$

implies

$$\Delta f_J/(g_J{}^T \Delta x_J) = 1 + O(\Delta) \tag{10}$$

If $0 < \Delta f_J / g_J{}^T \Delta x_J \ll 1$, then the decrease in cost is unreasonably small with respect to the steepness of the gradient.

Step 4 is a "filter" for the test

$$\Delta g_J{}^T \Delta x_J > 0 \tag{11}$$

It was noted in a number of simulations before the insertion of step 4 that condition (11) was violated. It is well-known that if f is bounded from below, then there exists a larger value of α_J which will cause the inequality to be satisfied, and in Fletcher's paper a scheme for increasing α_J is presented. However, this scheme might result in numerous costly gradient evaluations. (In the problems of the next section, a single gradient is approximately as costly as fourteen to eighteen function evaluations.) Since gradient calculations are so costly, an approximate test had to be devised to avoid the calculation of more than one gradient per iteration, and step 4 is the result.

It was noted that whenever the $\Delta g_J{}^T \Delta x_J > 0$ test was violated, the value of $|g_J{}^T \Delta x_J|$ was appreciably smaller than the value $|g_J{}^T{}_{-1} \Delta x_{J-1}|$ (two to three orders of magnitude smaller). That is, on successive iterates on which $\Delta g^T \Delta x > 0$, the value of $g^T \Delta x$ was changing by zero to one-to-a-half orders of magnitude, whereas it changed by at least two to three orders of magnitude when the test was violated.

Note that to cause $|g_J{}^T \Delta x_J|$ to increase toward $|g_{J-1}{}^T \Delta x_{J-1}|$, one need only increase the value of the search parameter, which is the same remedy for the $\Delta g_J{}^T \Delta x_J > 0$ violation. Thus, the following test was employed before the computation of $g(x_{J+1})$:

$$100 g_J{}^T \Delta x_J \leq g_{J-1}{}^T \Delta x_{J-1} \tag{12}$$

i.e., if $g_J{}^T \Delta x_J$ is at least 100 times greater than $g_{J-1}{}^T \Delta x_{J-1}$, then the stepsize is increased and a decrease in $g_J{}^T \Delta x_J$ is guaranteed. (Note that $g^T \Delta x < 0$ is guaranteed on each iterate because of step 3.) For all the shuttle computations this test always detected the $\Delta g_J{}^T \Delta x_J > 0$ violation without computation of $g(x_{J+1})$ for an unacceptable x_{J+1}-value.

The test (12) has not been proved mathematically and it seems feasible that there exist cases when the test is satisfied by $\Delta g_J{}^T \Delta x_J < 0$ and/or the tolerance value of 100 is unsuitable for other physical situations. However, $\Delta g_J{}^T \Delta x_J$ must be computed in each iteration for the H-formulas, and thus, the $\Delta g_J{}^T \Delta x_J > 0$ inequality can always be checked and guaranteed.

In any case, no more computation is required than in the original Fletcher's method since $g_J{}^T \Delta x_J$ must be computed for other formulas in the method.

4. Space Shuttle Trajectory Optimization

A number of Space Shuttle trajectory optimization problems were simulated in the development of the algorithm, including three ascent problems and a re-entry problem. A comparison of numerous algorithms for the stage-and-half configuration ascent problem are presented here along with partial results for a pressure-fed booster configuration ascent problem.

The stage-and-half optimization problem involved 18 parameters (azimuth adjustment parameter, payload, pitch angle at the time when engines dropped, and 15 pitch rates), where payload is to be maximized. The optimization is from 10 sec after liftoff to orbital insertion (50 × 100 with inclination specified). The results for this optimization problem are presented in Tables 2 and 3.

In Table 2 a comparison of DFP and the Modified Fletcher's method is shown for the case of a reasonably good guess for the initial parameter vector. The relatively small value of $g^T Hg$ on the 52nd iterate of the DFP method indicates that the problem is reasonably converged on that iterate. Considering the 52nd iterate of the DFP method as the converged solution, four digit accuracy is obtained by DFP on the 26th iterate and by the Modified Fletcher's Method on the 31st iterate (with respect to payload and boundary condition satisfaction). Since DFP requires more function evaluations and since a single gradient calculation corresponds roughly to 18 function evaluations, the computing times to reach the 26th iterate in DFP and the 31st iterate in Modified Fletcher are approximately the same. As shown in Table 2, ten more iterates are obtained for the Modified Fletcher's method in the same amount of computer time. However, DFP gets a lower value for the cost in the same amount of computer time, thus exhibiting better terminal convergence.

In Table 3 a comparison of six algorithms is shown for a poor guess of the initial parameter vector. In the first column

Table 2 Stage-and-half configuration: good initial x_0

	Davidon-Fletcher-Powell				Modified Fletcher		
Iteration number	Cost	No. of function evaluations	Search parameter	$\left.\frac{df}{d\alpha}\right\|_{\alpha=0} = -g^T Hg$	Cost	No. of function evaluations	Search parameter
0	0.13201×10^1	1	···	-0.79×10^4	0.13201×10^1	1	···
1	0.10640×10^1	12	0.646×10^{-4}	-0.90×10^1	0.10913×10^1	5^a	0.125×10^{-4}
2	0.10322×10^1	4	0.701×10^{-2}	-0.68×10^1	0.10595×10^1	4^a	0.312×10^{-3}
3	0.98038	4	0.161×10^{-1}	-0.37×10^1	0.10480×10^1	5^a	0.156×10^{-2}
4	0.21987	6	0.440	-0.28×10^1	0.57441	2^a	0.195
5	0.14202	6	0.795×10^{-1}	-0.21×10^0	0.25564	2^a	0.195
10	0.84552×10^{-1}	6	0.124×10^2	-0.28×10^{-4}	0.85738×10^{-1}	4^b	0.488×10^{-1}
20	0.84198×10^{-1}	4	0.876×10^1	-0.61×10^{-6}	0.84427×10^{-1}	2^a	0.763×10^{-2}
30	0.84187×10^{-1}	3	0.326	-0.36×10^{-7}	0.84212×10^{-1}	2^a	0.477×10^{-3}
40	0.84182×10^{-1}	5	0.198×10^3	-0.48×10^{-9}	0.84206×10^{-1}	2^b	0.298×10^{-3}
52	0.84181×10^{-1}	3	0.280×10^1	-0.79×10^{-12}	0.84204×10^{-1}	2^a	0.596×10^{-3}
62	···	···	···	···	0.84202×10^{-1}	4^a	0.142×10^{-3}

(43 min Univac 1108; central differences)

[a] Used DFP formula [Eq. (1)].
[b] Used Broyden formula [Eq. (2)].

Table 3 Stage-and-half configuration; poor initial x_0

	Gradient method		DFP (with search)		Broyden (with search)		Modified Fletcher		DFP (crude search)		Broyden (crude search)	
Iteration number	Cost	No. of fn. eval.	Cost	No. of fn. eval.	Cost	No. of fn. eval.	Cost	No. of fn. eval.	Cost	No. of fn. eval.	Cost	No. of fn. eval.
0	1611	1	1611	1	1611	1	1611	1	1611	1	1611	1
1	271.1	7	271.1	7	271.1	7	397.7	10^a	397.7	10	397.7	10
2	54.65	5	13.75	4	13.75	4	34.64	5^b	34.64	4	63.07	5
3	13.04	5	2.792	4	2.792	4	32.24	7^a	32.27	7	30.72	6
4	10.83	4	1.311	4	1.311	4	30.95	3^a	28.31	3	16.58	4
5	8.667	4	0.8897	4	0.8897	4	8.795	3^b	7.948	3	7.395	3
6	7.736	4	0.2636	5	0.2636	5	6.003	3^b	5.571	3	4.426	3
7	6.922	4	0.1621	4	0.1621	4	5.968	3^b	4.809	3	3.422	3
8	6.438	4	0.1563	4	0.1563	4	3.123	3^a	3.429	3	3.241	3
9	6.123	4	0.1554	4	0.1554	4	2.631	3^b	2.850	3	2.388	4
10	6.038	4	0.1461	4	0.1461	4	2.420	3^b	2.749	3	2.123	5
12	5.143	17	0.1183	4	0.1183	4	2.205	3^a	1.815	4	0.4776	3
14	3.574	17	0.1046	4	0.1045	4	0.8923	4^b	1.528	5	0.4309	4
16	2.891	15	0.0972	4	0.0967	4	0.5323	3^a	0.4339	5	0.3682	4
18	2.834	17	0.0923	5	0.0921	5	0.3663	3^a	0.3741	3	0.3238	3
20	2.790	6	0.0888	4	0.0886	4	0.3252	3^b	0.3619	3	0.3104	3
22	2.718	4	0.0875	5	0.0873	5	0.3092	3^b	0.3485	4	···	···
30	2.457	4	0.08553	6	0.08551	5						
50	1.739	25	0.08516	6	0.08508	6						
60	1.727	4	0.08490	5	0.08485	4						

(10 min, Univac 1108; forward differences)

(Approx. 40 min, Univac 1108; forward differences)

[a] Used DFP formula [Eq. (1)].
[b] Used Broyden formula [Eq. (2)].

the gradient method (with a 1-D search) is included to show the difficulty of obtaining good terminal convergence in this problem. The next two methods, DFP and Broyden, were the best performers with Broyden slightly better than DFP. Note that DFP and Broyden give identical costs (to four digits) in the early iterates and then Broyden begins to get slightly lower costs; this is the same characteristic Broyden[16] noticed. The last three columns show three methods which use only a crude search: Modified Fletcher, DFP with a crude search, and Broyden with a crude search. All three methods gave comparable results with Modified Fletcher obtaining the lowest cost in 10 min computer time. All three H-formulas satisfy the main property of Fletcher's paper (i.e., $\phi \varepsilon [0,1]$), and for this particular problem probably give similar results because the H-matrix remains well-behaved. Although these three methods are not better than DFP or Broyden (with searches) on this problem, they are appreciably better than the gradient method and yet do not require extensive programming.

Table 4 shows an incomplete study of results obtained for the pressure-fed booster shuttle ascent problem. In this problem an element of the main diagonal of the H-matrix in the DFP method became appreciably smaller than the other elements of the main diagonal in the early iterates. This caused the 1-D search considerable trouble is obtaining a minimum, as noted by the large number of function evaluations (especially on the 3rd, 4th, and 5th iterates). In this particular problem the Modified Fletcher's method performed better than the DFP method in that it required a considerably less number of function evaluations and obtained a lower cost value in the same number of iterates. Note how the Modified Fletcher method uses both of the formulas on this problem [i.e., Eq. (1) is used 6 times, Eq. (2) is used 5 times].

During the course of the study a number of observations were made with respect to the performance of the algorithms and reports of their performance in the literature. These are summarized below.

1) The performance of the DFP method is strongly dependent upon the 1-D search used. In the early part of the study, the Modified Fletcher method required approximately the same amount of computer time as DFP to obtain the same cost on a number of different problems. Then a more sophisticated search was used in the DFP algorithm. The DFP method then became a much better performer. This explains how, in the literature, numerous algorithms are reported to outperform DFP when with an efficient search DFP is clearly the better performer. [In Ref. 8, Fletcher's method is reported to outperform DFP on a number of standard functions. However, when the two were compared with the NASA-JSC PEACE DFP program, DFP easily outperformed Fletcher's method.]

2) In a number of papers in the literature, little emphasis is given to the expense of computing gradients as opposed to function evaluations. For example, the IBM Scientific Subroutine[19] version of DFP calculates a gradient each time it evaluates the function. This calculation is not serious on low-dimension, test type problems, but it is extremely important when realistic problems are attacked (especially problems which require numerical integration for the function and gradient evaluations).

3) In the early stages of the study, the effect of resetting to a gradient step every so many iterates was investigated. On the problems considered herein it was not found to be helpful; in fact, it was found to be detrimental in the terminal stages of convergence because the H-matrix had to be rebuilt. Most of the example problems in the literature which get improved convergence with reset are of relatively low dimension. (One theoretical advantage of reset is if it is included in any stable H-matrix type algorithm, then convergence can be proved for the same class of functions for which convergence can be proved for the gradient method.)

5. Conclusions

A parameter optimization method devised by Fletcher has been modified to make it an effective scheme for trajectory optimization. The resultant scheme has the following desirable features. 1) An elaborate 1-D search is not required, and thus, the scheme is easy to program. 2) Davidon-type formulas are utilized. 3) Because of a filter for the $\Delta g_j^T \Delta x_j > 0$ test, no more than one gradient per iteration has been required in simulations to date. (Other schemes which do not involve a 1-D search sometimes suffer from requiring more than one gradient computation on a number of iterations.) A flowchart of the algorithm is given in Ref. 20.

In addition to the study of the Modified Fletcher's method, a new method due to Broyden[16] was simulated, but not extensively. It performed slightly better than DFP and thus merits further analysis.

Finally, with respect to the relative merits of DFP and the Modified Fletcher's method, if an efficient 1-D search is available it appears that DFP (or possibly the new Broyden formula) gives the best results, especially in the terminal stages

Table 4 Ascent problem with H-matrix tending to singularity in DFP method

		Davidon-Fletcher-Powell				Modified Fletcher	
Iteration number	Cost	No. of function evaluations	Search parameter	$\left.\frac{df}{d\alpha}\right\|_{\alpha=0} = -g^T H g$	Cost	No. of function evaluations	Search parameter
0	1.058	1		-0.8	1.058	1	···
1	1.058	4	0.25×10^{-4}	-0.2	1.058	5[a]	0.12×10^{-4}
2	1.057	5	0.95×10^{-2}	-0.3×10^{-1}	1.057	7[a]	0.78×10^{-2}
3	1.056	26	0.35×10^{-1}	-0.9×10^{-1}	1.057	3[b]	0.78×10^{-2}
4	1.056	37	0.68×10^{-4}	-0.1×10^{2}	1.056	6[a]	0.48×10^{-3}
5	1.053	30	0.10×10^{1}	-0.1×10^{-1}	1.056	11[a]	0.95×10^{-6}
6	1.050	5	0.53	-0.4×10^{-2}	1.033	4[a]	0.24×10^{-6}
7	1.024	9	0.93×10^{1}	-0.4×10^{-2}	1.029	3[a]	0.12×10^{-6}
8	1.001	6	0.84×10^{1}	-0.3×10	0.9816	3[b]	0.12×10^{-6}
9	0.9883	5	0.68×10^{1}	-0.3×10^{-2}	0.9751	3[b]	0.12×10^{-6}
10	0.9615	7	0.21×10^{2}	-0.3×10^{-2}	0.9607	3[b]	0.12×10^{-6}
11	0.9592	7	0.16×10^{1}	-0.2×10^{-2}	0.9554	3[b]	0.12×10^{-6}
15	0.9547	5	0.44×10^{1}	-0.1×10^{-3}	···	···	···
20	0.9378	5	0.94×10^{1}	-0.6×10^{-4}			
34	0.9360	22	0.10×10^{8}			(7 min, Univac 1108; central differences)	
		(22.5 min, Univac 1108; central differences)					

[a] Used DFP formula [Eq. (1)].
[b] Used Broyden formula [Eq. (2)].

of convergence. However if either an efficient 1-D search is not available or the H-matrix appears ill-conditioned in the DFP method or single-precision computations are desired, then the Modified Fletcher method appears to be an attractive alternative.

Appendix: Crude Searches

Stepsize Increase

Given x_J and α_{J1}, where α_{J1} is an estimate of α_J. Define

$$f[x_{Ji}] \equiv f[x_J - \alpha_{Ji} H_J g_J] \qquad (\mathrm{i} = 1,2,\ldots) \qquad (\mathrm{A1})$$

Evaluate $f[x_{Jk}](k = 1,2,\ldots)$ with

$$\alpha_{J(k+1)} = 5\alpha_{Jk}, \qquad (k = 1,2,\ldots,K) \qquad (\mathrm{A2})$$

where K is the least integer where a function increase occurs, i.e.,

$$f[x_{J1}] > f[x_{J2}] > \cdots > f[x_{J(K-1)}] < f[x_{JK}] \qquad (\mathrm{A3})$$

Define

$$\alpha_{J(K+1)} = 2.5\alpha_{J(K-1)} \qquad (\mathrm{A4})$$

and then define α_J to be either $\alpha_{J(K-1)}$ or $\alpha_{J(K+1)}$, whichever gives the least function evaluation.

Stepsize Decrease

Given x_J and α_{J1} where α_{J1} is an estimate of α_J. Evaluate $f[x_{Jk}](k = 1,2,\ldots)$ by Eq. (A1) with

$$\alpha_{J(k+1)} = \alpha_{Jk}/2, \qquad (k = 1,2,\ldots,K) \qquad (\mathrm{A5})$$

where K is the least integer where a function increase occurs [i.e., Eq. (A3)]. Define α_J to be $\alpha_{J(K-1)}$.

[Note: In Eqs. (A2) and (A5), the factors 5 and 2 were found to be convenient choices in the problems of this paper. However, any values greater than one may be used.]

Finally, the factors 5 and 2 were also used in the increase and decrease portions of Steps 3 and 4, Sec. 2.

References

[1] Hestenes, M. R. and Stiefel, E., "Method of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, Vol. 49, No. 6, 1952, pp. 409–436.

[2] Davidon, W. C., "Variable Metric Method for Minimization," Research and Development Rept. ANL-5990, Rev., Nov. 1959, Atomic Energy Commission, Washington, D.C.

[3] Fletcher, R. and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization," *Computer Journal*, Vol. 6, No. 2, 1963, pp. 163–168.

[4] Fletcher, R. and Reeves, C. M., "Function Minimization by Conjugate Gradients," *Computer Journal*, Vol. 7, No. 2, 1964, pp. 149–154.

[5] Pearson, J. D., "Variable Metric Methods of Minimization," *Computer Journal*, Vol. 12, No. 5, 1969, pp. 171–178.

[6] Broyden, C. G., "Quasi-Newton Methods and Their Application to Function Minimization," *Mathematics of Computation*, Vol. 21, 1967, pp. 368–381.

[7] Murtagh, B. A. and Sargent, R. W. H., "Computational Experience with Quadratically Convergent Minimization Methods," *Computer Journal*, Vol. 13, No. 2, 1970, pp. 185–194.

[8] Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer Journal*, Vol. 13, No. 3, 1970, pp. 317–322.

[9] Jacobson, D. H. and Oksman, W., "An Algorithm that Minimizes Homogeneous Functions of N Variables in $N+2$ Iterations and Rapidly Minimizes General Functions," TR 618, Oct. 1970, Harvard Univ., Cambridge, Mass.

[10] Kelley, H. J., Myers, G. E., and Johnson, I. L., "An Improved Conjugate Direction Minimization Procedure," *AIAA Journal*, Vol. 8, No. 11, Nov. 1970, pp. 2091–2093.

[11] McCormick, G. P. and Ritter, K., "Methods of Conjugate Directions Versus Quasi-Newton Methods," *Mathematical Programming*, Vol. 3, No. 1, 1972, pp. 101–129.

[12] Johnson, I. L. and Kamm, J. L., "Parameter Optimization and the Space Shuttle," *Proceedings of the 1971 JACC*, St. Louis, Mo., Aug. 1971, pp. 776–781.

[13] Johnson, I. L. and Kamm, J. L., "Near Optimal Shuttle Trajectories Using Accelerated Gradient Methods," AAS/AIAA Paper 328, Ft. Lauderdale, Fla., 1971.

[14] Kamm, J. L. and Johnson, I. L., "Optimal Shuttle Trajectory-Vehicle Design Using Parameter Optimization," AAS/AIAA Paper 329, Ft. Lauderdale, Fla., 1971.

[15] Johnson, I. L. and Kamm, J. L., "An MSC Guideline for Atmospheric Flight Optimization," MSC Internal Note 71-FM-200, May 20, 1971, Johnson Space Center, Houston, Texas.

[16] Broyden, C. G., "The Convergence of a Class of Double-Rank Minimization Algorithms: 2. The New Algorithm," *Journal of the Institute of Mathematics and its Applications*, Vol. 6, No. 3, 1970, pp. 222–231.

[17] Goldfarb, D., "A Family of Variable-Metric Methods Derived by Variational Means," *Mathematics of Computation*, Vol. 24, 1970, pp. 23–26.

[18] Greenstadt, J., "Variations on Variable-Metric Methods," *Mathematics of Computation*, Vol. 24, 1970, pp. 1–22.

[19] IBM System/360, Scientific Subroutine Package, (360A-CM-03X) Version III (1968), IBM Technical Publication Department, p. 221. (Corrected by IBM Memo, May 1969.)

[20] Powers, W. F., "A Crude-Search Davidon-Type Technique with Application to Shuttle Optimization," AIAA Paper 72-907, Palo Alto, Calif., 1972.