Numerical Methods for Non-linear Optimization

Conference sponsored by the

Science Research Council

University of Dundee, Scotland, 1971

Edited by

F. A. LOOTSMA

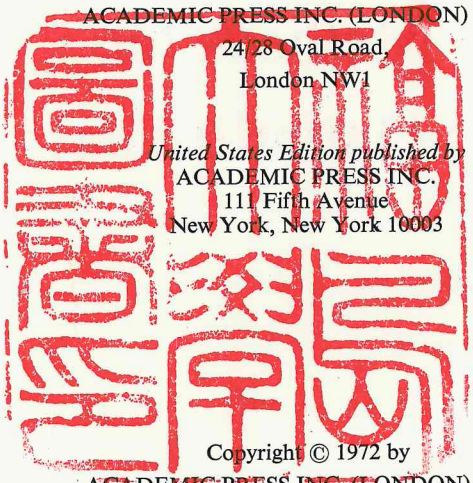
Philips Research Laboratories
Eindhoven, The Netherlands

1972

Academic Press **London · New York**

ACADEMIC PRESS INC. (LONDON) LTD.
24/28 Oval Road,
London NW1

United States Edition published by
ACADEMIC PRESS INC.
111 Fifth Avenue
New York, New York 10003



Library of Congress Catalog Card Number: 72-84446
ISBN: 0-12-455650-7

PRINTED IN GREAT BRITAIN BY
WILLIAM CLOWES & SONS LIMITED,
LONDON, COLCHESTER AND BECCLES

6. A Uniform Evaluation of Unconstrained Optimization Techniques

D. M. HIMMELBLAU
Department of Chemical Engineering, The University of Texas at Austin, Austin, Texas, U.S.A.

Summary

To compare the performance of 15 different unconstrained non-linear programming algorithms, 15 minimization test problems were executed on a CDC 6600 computer. In each test the same degree of precision was sought in the relative value of the objective function, the vector of variables, and the gradient of the objective function. The criteria used in the evaluation were robustness and relative overall ranking for time of execution. It was concluded that Fletcher's algorithm was superior to the others.

1. Introduction

The evaluation of the performance of unconstrained non-linear programming algorithms on an equitable basis requires a suitable combination of appropriate test problems, programming skill, and reasonable criteria of comparison. Algorithms can be examined from a theoretical viewpoint as well as being tested by experimentation. The former can only be applied to a rather restricted class of problems; hence we will be concerned here with the evaluation of the effectiveness of algorithms by experimentation, i.e., by solving test problems. Algorithms can be tested on problems with both a small and large number of variables, on problems with varying degrees of non-linearity, and on problems evolving from practical applications, such as least squares, solution of sets of non-linear equations, and the like. By examining the effectiveness of an algorithm in treating a variety of problems, one can hope to predict the general effectiveness of an algorithm in solving other problems of a like and also of a different nature.

For a test problem to be useful it is best that the problem have a single extremum or at least a restricted number of extrema. We cannot yet expect at the current stage of development that an algorithm will pick out the global minimum if a problem has more than one minimum, but it should at least

reach a local minimum to be considered successful. Nevertheless, if one considers the type of problem in which one local minimum exists at some x, the vector of the variables, and in addition the objective function, $f(x)$, has a global minimum at $-\infty$, if a particular minimization technique proceeds towards the global minimum, should it be deemed a success or a failure?

Any experimental comparison of algorithms depends to a considerable extent on how the algorithms are programmed for the computer. Small details of the programming can exert a considerable influence on the effectiveness of an algorithm. Slight changes in the termination criteria, the unidimensional search technique, test of matrices for singularity, matrix inversion procedures, reset, and the like make a big difference in the performance of an algorithm. Just a simple change in the initial step in the unidimensional search can be shown to exert quite an impact on the search trajectory in minimizing Rosenbrock's function. Many of these factors have been ignored by authors in reporting their test results for an algorithm because the factors were deemed to be unrelated to the basic algorithm, yet their contribution to the workability of an algorithm should never go unrecognized. Many algorithms to be successful also require that heuristic logic be introduced into the algorithm in addition to the bare skeleton of the procedure described in the literature. Such logic is devised from experimental experience with failure and has little to do with the fundamental concept underlying the algorithm, but makes it work.

Before evaluating the relative effectiveness of the various unconstrained algorithms, some remarks are appropriate concerning the criteria to use in evaluating their effectiveness.

2. Criteria for Evaluation

It is generally accepted that the primary criterion in evaluating general purpose algorithms must be whether or not the algorithm can solve most of the problems posed; that is, is the algorithm *robust*? Of course, any algorithm can be defeated by a suitably designed (pathological) problem, and even for other than pathological problems we cannot realistically ask that an algorithm solve *all* possible problems, for it is easy to pose an unconstrained non-linear programming problem that leads to negative arguments, division by zero, discontinuities, and the like. But the robustness of the algorithm is always of primary concern.

A second criterion is that of the desired degree of precision in the solution, that is, in the value of the objective function at the extremum, $f(x^*)$, and of the elements of the vector of variables x^* at the extremum. Usually the degree of precision in the solution depends upon the termination criteria used to end the computation. To provide for uniform criteria for termination, in the results described here the algorithms were adjusted so that the same relative precision in the optimal x-vector, x^*, in $f(x^*)$ and $\nabla f(x^*)$ were the joint bases

for stopping the search in each code. Figure 1 indicates why both of the first two criteria must be involved in the termination procedure. If the algorithm terminates solely on the fractional change in $f(x)$ being less than some small

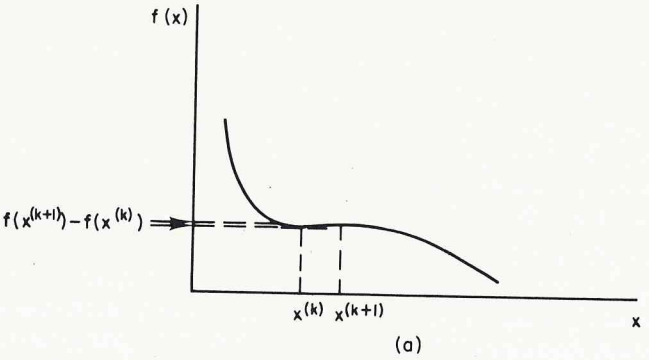


FIG. 1a. A criterion based solely on

$$\frac{f(x^{(k+1)}) - f(x^{(k)})}{f(x^{(k)})} < \epsilon$$

will terminate prematurely on a flat plateau.

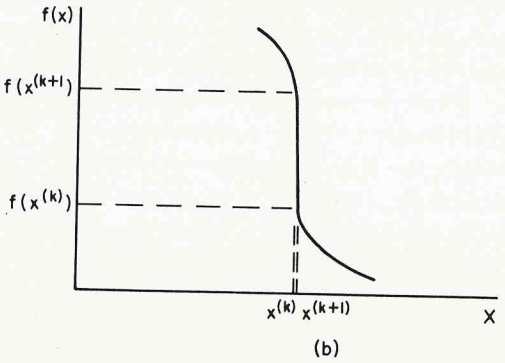


FIG. 1b. A criterion based solely on

$$\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}} < \epsilon$$

will terminate prematurely on a very steep slope.

number, a flat plateau can cause premature termination. Alternatively, if the algorithm is set up to terminate solely on the fractional change in the elements of x, a steep slope can cause premature termination. Use of only the components of the gradient can lead to termination at a saddle point. One feature that should be mentioned concerning the termination criteria is that when $f(x)$ and/or x approach 0, the termination criteria must be the fractional change

rather than relative change in $f(x)$ and/or x to avoid dividing by a very small number. Some authors have used as termination criteria the norm of $\nabla f(x)$, the norm of x, or the norm of the search direction d, and although these are perfectly adequate criteria under most circumstances, they do suffer from the same deficiencies identified in Fig. 1.

Assuming that uniform termination criteria are adopted, a third criterion of the effectiveness of an algorithm is the number of functional evaluations of $f(x)$ to reach the desired precision in $f(x)$ and x. Certainly this criterion is better than the number of stages, or iterations, for the number of stages varies quite widely from algorithm to algorithm and in many algorithms means something quite different than the selection of a new search direction. Nevertheless, the number of function evaluations itself is not too satisfactory a measure of efficiency for algorithms with widely differing strategies because the number of functional evaluations to devise a search direction relative to the number of functional evaluations to move in a given direction differ widely from strategy to strategy. Furthermore, how should the evaluation of the derivatives be weighted relative to the evaluation of the objective function itself in those algorithms that use derivatives so that the derivative methods include both objective function and derivative evaluations? Finally, one can reduce the number of function evaluations by all sorts of time-consuming tests, special heuristic operations, matrix operations, and so forth, so that a comparison based solely on function evaluations can easily be misleading.

Consequently, a fourth criterion, the computation time to execute an algorithm, is alternatively cited as a measure of the effectiveness of an algorithm. Although the relative time to termination is not a particularly desirable measure of the effectiveness of an unconstrained non-linear programming algorithm, in lieu of a better measure it often has to serve. Certain hazards exist in using time alone as a criterion. For simple test problems the time required to read the data and execute the print commands (we, of course, omit the printing time itself) in a code with a modestly detailed print out—say x and $f(x)$ for each stage—may prove to be two or three times the computation time experienced when these phases of the code are bypassed. In computers in which the central processing unit operates on several programs in a time-sharing recycling mode, the input–output times when added up may easily exceed by a factor of 2 or 3 the single pass time for execution. Thus, the type of computer, the care in the coding of the algorithm, and the character of the measured time all have an important bearing on the use of time of execution as a criterion. Such information is usually missing from reports in the literature describing the behaviour of a specific algorithm. In the work described below in Section 4 a CDC 6600 computer was used to determine the execution times. All printing, peripheral processing, and system manipulation times were specifically excluded from the times listed in Table 2, so that the times indeed

represent the number of seconds required to execute the algorithms without interruption of any sort.

In summary, the criteria to be considered in the evaluation of the unconstrained algorithms are.

1. robustness—success in obtaining an optimal solution (to within a certain precision);
2. number of function (and derivative) evaluations;
3. computer time to termination (to within the desired degree of precision).

3. Test Problems

A number of test problems have been used by various authors of non-linear programming algorithms. Some of the test problems have been used so often that they have assumed the role of 'classics', being repetitively used to compare the performance of algorithms, usually to demonstrate that a new algorithm is as good or better than its predecessors. Table 1 lists 15 functions that were used to evaluate the respective algorithms. The functions contained only a few variables, in contrast to some least squares test functions of a large number of variables that have appeared in the literature. The latter represent a rather special type of function, however.

TABLE 1
Comparison of 15 algorithms for 15 test problems

All functions were minimized; f = failed to reach solution in 15 sec either because progress was slow or termination occurred at incorrect solution. The times shown are the execution times on a Control Data 6600. This computer has a 'standard time' of 22 sec for the standard timing (matrix inversion) program described by A. R. Colville in Techn. Report Nr. 320-2949, IBM Corp., New York Scientific Center, June 1968.

PROBLEM 1 (Zangwill, 1967)

$$f(x) = (1/15)(16x_1^2 + 16x_2^2 - 8x_1x_2 - 56x_1 - 256x_2 + 991)$$

$$x^{(0)} = (3,8); \quad x^* = (4,9); \quad f(x^*) = -18.2$$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	1	5	4	0·006
-G	1	23	4	0·008
B-D	1	5	4	0·008
-G	1	23	4	0·009
P2-D	1	5	4	0·006
-G	1	23	4	0·008
P3-D	1	5	4	0·008
-G	1	23	4	0·008
PN-D	1	5	4	0·007
-G	1	23	4	0·007
FR-D	1	5	4	0·006
-G	1	23	4	0·010
CP-D	1	5	4	0·007
-G	1	23	4	0·009
IP-D	1	5	4	0·006
-G	1	23	4	0·005
GP-D	1	6	4	0·007
-G	1	24	4	0·007
F	2	3	6	0·004
HJ		80		0·009
NM		185		0·024
P-D	1	29		0·010
-G	1	218		0·024
R		62		0·018
S-D	1	16		0·005
-G	1	84		0·008

TABLE 1—*continued*

PROBLEM 2 (White and Holst, 1964)

$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$

$x^{(0)} = (-1{\cdot}2, 1{\cdot}0); \quad x^* = (1,1); \quad f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	13	219	28	0·030
-G	14	196	30	0·022
B-D	16	230	34	0·031
-G	14	182	30	0·021
P2-D	67	789	136	0·102
-G				f
P3-D	27	297	56	0·047
-G	37	480	76	0·044
PN-D	14	857	30	0·044
-G	20	492	42	0·043
FR-D	35	1304	72	0·152
-G	28	574	58	0·047
CP-D	44	515	90	0·063
-G	32	481	66	0·043
IP-D	35	462	72	0·061
-G	141	2092	284	0·171
GP-D	10	208	58	0·032
-G	9	126	52	0·021
F	37	53	106	0·022
HJ		651		0·056
NM		359		0·035
P-D	14	284		0·038
-G	2	156		0·016
R		294		0·053
S-D	14	256		0·040
-G	14	194		0·026

TABLE 1—*continued*

PROBLEM 3

$$f(x) = x_1^3 + x_2^2 - 3x_1 - 2x_2 + 2$$

$$x^{(0)} = (0,2); \qquad x^* = \begin{cases} (1,1), \\ (-\infty, x_2) \end{cases} \qquad f(x^*) = \begin{cases} -1 \\ -\infty \end{cases}$$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	5	16	12	0·015
-G	6	64	14	0·019
B-D	5	16	12	0·012
-G	6	64	14	0·021
P2-D	8	25	18	0·020
-G	5	64	12	0·016
P3-D	6	19	14	0·017
-G	6	19	14	0·016
PN-D				a
-G				a
FR-D				a
-G				a
CP-D				a
-G				a
IP-D				a
-G				a
GP-D				a
-G				a
F	5	7	7	0·015
HJ		640		0·305
NM		190		0·173
P-D	2	24		0·014
-G	5	220		0·050
R	160	163		0·052
S-D				a
-G				a

a = converges toward global minimum, $f(x) \to -\infty$

TABLE 1—*continued*

PROBLEM 4 (Beale, 1958)

$$f(x) = [1{\cdot}5 - x_1(1 - x_2)]^2 + (2{\cdot}25 - x_1(1 - x_2^2)]^2 + [2{\cdot}625 - x_1(1 - x_2^3)]^2$$

$$x^{(0)} = (1{\cdot}0, 0{\cdot}8); \qquad x^* = (3{\cdot}0, 0{\cdot}5); \qquad f(x^*) = 0$$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D				f
-G	10	158	22	0·022
B-D				f
-G	10	159	22	0·021
P2-D				f
-G	42	534	86	0·069
P3-D				f
-G	17	247	36	0·029
PN-D				f
-G	15	237	32	0·031
FR-D				f
-G	28	258	58	0·033
CP-D				f
-G	27	373	56	0·041
IP-D				f
-G	33	342	68	0·039
GP-D				f
-G	10	149	58	0·024
F	16	22	44	0·013
HJ		205		0·024
NM		230		0·029
P-D	27	134		0·021
-G	21	396		0·039
R		218		0·052
S-D				f
-G	9	161		0·024

TABLE 1—*continued*

PROBLEM 5 (Engvall, 1966)

$f(x) = x_1^4 + x_2^4 + 2x_1^2 x_2^2 - 4x_1 + 3$

Normalization of initial search vector used in the unidimensional search in all algorithms except DFP-D and B-D.

$x^{(0)} = (0{\cdot}5, 2{\cdot}0)$: $x^* = (1{\cdot}0, 0)$; $f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	7	82	16	0·016
-G	8	119	18	0·017
B-D	9	120	20	0·017
-G	7	84	16	0·020
P2-D	18	106	38	0·022
-G	20	227	42	0·027
P3-D	9	65	20	0·013
-G	10	119	22	0·017
PN-D	7	57	16	0·012
-G	12	138	26	0·017
FR-D	8	58	18	0·011
-G	18	134	38	0·019
CP-D	11	79	24	0·016
-G	13	142	28	0·020
IP-D	10	71	22	0·015
-G	10	144	22	0·019
GP-D	5	51	28	0·014
-G	8	105	46	0·020
F	10	15	30	0·008
HJ		64		0·008
NM		210		0·025
P-D	6	96		0·017
-G	5	264		0·024
R		119		0·026
S-D	7	119		0·017
-G	7	137		0·016

TABLE 1—*continued*

PROBLEM 6 (Box, 1966)

$$f(x) = \sum_{i=1}^{10} [\exp(-x_1 t_i) - \exp(-x_2 t_i) - \exp(-t_i) + \exp(-10t_i)]^2$$

$(t_i = 0{\cdot}1, 0{\cdot}2, \ldots, 1{\cdot}0)$

$x^{(0)} = (5, 0); \quad x^* = (1, 10); \quad f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	12	142	26	0·277
-G	11	207	24	0·370
B-D	9	112	20	0·222
-G	9	145	20	0·262
P2-D	14	128	30	0·256
-G				f
P3-D				f
-G				f
PN-D	8	114	18	0·217
-G	14	255	30	0·442
FR-D	10	95	22	0·192
-G	33	296	68	0·587
CP-D	167	824	336	1·81
-G	11	198	24	0·351
IP-D	29	212	60	0·434
-G	292	1503	586	3·26
GP-D	6	54	34	0·138
-G	6	96	34	0·212
F	19	25	50	0·076
HJ		498		0·799
NM		268		0·436
P-D	24	161		0·275
-G	18	278		0·464
R		314		0·527
S-D	12	177		0·304
-G	17	406		0·674

TABLE 1—*continued*

PROBLEM 7 (Zangwill, 1967)

$$f(x) = (x_1 - x_2 + x_3)^2 + (-x_1 + x_2 + x_3)^2 + (x_1 + x_2 - x_3)^2$$

$$x^{(0)} = (100, -1, 2{\cdot}5); \qquad x^* = (0, 0, 0); \qquad f(x^*) = 0$$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	2	9	9	0·010
-G	5	99	18	0·017
B-D	3	12	12	0·009
-G	5	99	18	0·018
P2-D	7	27	24	0·012
-G	18	210	57	0·037
P3-D	5	20	18	0·011
-G	6	128	21	0·018
PN-D	3	12	12	0·008
-G	5	175	18	0·032
FR-D	3	12	12	0·014
-G	7	106	24	0·021
CP-D	19	77	60	0·021
-G	20	321	81	0·017
IP-D	28	102	87	0·020
-G	26	356	81	0·004
GP-D	3	14	30	0·010
-G	5	100	54	0·019
F	4	6	18	0·004
HJ		130		0·014
NM		810		0·096
P-D	5	84		0·014
-G	7	502		0·057
R		297		0·067
S-D	5	37		0·010
-G	6	108		0·020

TABLE 1—*continued*

PROBLEM 8 (Schmidt and Vetters, 1970)

$$f(x) = \frac{1}{1+(x_1-x_2)^2} + \sin\left(\frac{\pi x_2 + x_3}{2}\right) + \exp\left[\left(\frac{x_1+x_2}{x_2} - 2\right)^2\right]$$

$x^{(0)} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$; $x^* = (0{\cdot}78547, 0{\cdot}78547, 0{\cdot}78547)$; $f(x) = 3$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	8	34	27	0·034
-G	9	110	30	0·029
B-D	8	34	27	0·022
-G	10	116	33	0·033
P2-D	45	201	138	0·068
-G	45	540	138	0·125
P3-D	8	32	27	0·019
-G	12	110	39	0·038
PN-D	8	34	27	0·017
-G	16	130	51	0·038
FR-D	16	63	51	0·025
-G	17	148	54	0·040
CP-D	51	205	156	0·064
-G	26	256	84	0·059
IP-D	63	221	192	0·073
-G	90	555	273	0·143
GP-D	6	26	66	0·022
-G	8	90	90	0·038
F	9	12	36	0·015
HJ		177		0·053
NM		279		0·057
P-D	9	86		0·021
-G	11	312		0·060
R		158		0·049
S-D	8	75		0·015
-G	9	127		0·024

TABLE 1—*continued*

PROBLEM 9 (Engvall, 1966)

$$f(x) = \sum_{i=1}^{5} f_i^2(x), \qquad f_1(x) = x_1^2 + x_2^2 + x_3^2 - 1$$

$$f_2(x) = x_1^2 + x_2^2 + (x_3 - 2)^2 - 1$$

$$f_3(x) = x_1 + x_2 + x_3 - 1$$

$$f_4(x) = x_1 + x_2 - x_3 + 1$$

$$f_5(x) = x_1^3 + 3x_2^2 + (5x_3 - x_1 + 1)^2 - 36$$

$$x^{(0)} = (1, 2, 0); \qquad x^* = (0, 0, 1); \qquad f(x^*) = 0$$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	17	99	54	0·028
-G	18	253	57	0·056
B-D	16	93	51	0·030
-G	18	255	57	0·044
P2-D	6317	41285	18954	7·55
-G				f
P3-D	48	275	147	0·065
-G	37	498	114	0·080
PN-D	25	169	78	0·036
-G	32	473	99	0·073
FR-D	41	283	126	0·054
-G	217	1481	654	0·154
CP-D	1219	6236	3660	1·08
-G	400	3155	1203	0·489
IP-D	2261	11374	6786	1·89
-G				f
GP-D	10	75	114	0·032
-G	12	173	138	0·044
F	22	31	93	0·022
HJ		81		0·012
NM		561		0·087
P-D	11	315		0·052
-G	13	652		0·079
R		457		0·114
S-D	16	150		0·032
-G	17	304		0·048

TABLE 1—*continued*

PROBLEM 10 (Fletcher and Powell, 1963)

$f(x) = 100\{(x_3 - 10\theta)^2 + [(x_1^2 + x_2^2)^{1/2} - 1]^2\} + x_3^2$,

where $2\pi\theta = \tan^{-1}(x_2/x_1)$, $-\pi/2 < 2\pi\theta < 3\pi/2$.

$x^{(0)} = (-1, 0, 0)$; $x^* = (1, 0, 0)$; $f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	20	141	63	0·047
-G	24	380	75	0·087
B-D	21	140	66	0·047
-G	25	384	78	0·085
P2-D	1097	5795	3294	1·59
-G	4378	20476	13137	6·35
P3-D	96	451	291	0·142
-G	133	1780	402	0·402
PN-D	36	221	111	0·068
-G	42	588	129	0·134
FR-D	36	202	111	0·054
-G	149	838	450	0·222
CP-D	57	318	174	0·080
-G	121	1171	366	0·256
IP-D	1945	6659	5838	1·945
-G	661	5013	1986	1·128
GP-D	10	119	114	0·048
-G	12	200	138	0·061
F	35	42	126	0·036
HJ		1230		0·214
NM		566		0·118
P-D	4	48		0·017
-G	12	277		0·057
R		513		0·146
S-D	21	191		0·048
-G	24	430		0·090

TABLE 1—*continued*

PROBLEM 11 (Bard, 1970)

$$f(x) = \sum_{i=1}^{15} \{y_i - [x_1 + a_{1i}/(x_2\, a_{2i} + x_3\, a_{3i})]\}^2$$

where y_i and a_{1i}, a_{2i}, and a_{3i} are tabulated data:

y_i	a_{1i}	a_{2i}	a_{3i}	y_i	a_{1i}	a_{2i}	a_{3i}
0·14	1	15	1	0·39	8	8	8
0·18	2	14	2	0·37	9	7	7
0·22	3	13	3	0·58	10	6	6
0·25	4	12	4	0·73	11	5	5
0·29	5	11	5	0·96	12	4	4
0·32	6	10	6	1·34	13	3	3
0·35	7	9	7	2·10	14	2	2
				4·39	15	1	1

$x^{(0)} = (1, 1, 1)$; $x^* = (0{\cdot}0824, 1{\cdot}1330, 2{\cdot}3437)$; $f(x^*) = 8{\cdot}215 \times 10^{-3}$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	16	78	51	0·041
-G	11	164	36	0·058
B-D	10	55	33	0·029
-G	11	163	36	0·055
P2-D	19	93	60	0·048
-G	19	269	60	0·089
P3-D	140	495	423	0·259
-G	162	1510	489	0·541
PN-D	20	180	63	0·073
-G	23	326	72	0·109
FR-D	20	114	63	0·047
-G	47	382	144	0·171
CP-D	38	209	117	0·078
-G	42	390	129	0·135
IP-D				f
-G				f
GP-D	7	62	78	0·040
-G	7	122	78	0·056
F	38	42	126	0·047
HJ				g
NM		711		0·226
P-D	6	102		0·061
-G	6	174		0·080
R				g
S-D	18	134		0·049
-G	13	198		0·063

g — not executed

TABLE 1—*continued*

PROBLEM 12 (Powell, 1964)

$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$

Note that the Hessian matrix of this function at its minimum is singular

$x^{(0)} = (3, 1, 0, -1)$; $x^* = (0, 0, 0, 0)$; $f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	36	434	148	0·079
-G	73	526	296	0·112
B-D	38	374	156	0·071
-G	36	358	148	0·064
P2-D	1655	10317	6624	2·22
-G	1173d	15688	4696	2·27
P3-D	312	1422	1252	0·853
-G	1699	22763	6800	3·43
PN-D	30	693	124	0·084
-G	47	971	182	0·125
FR-D	104	624	420	0·092
-G	912	3774	3652	0·691
CP-D	3226	10432	12908	1·66
-G	402	4158	1612	0·536
IP-D	140	705	564	0·098
-G	7045	72318	28184	9·24
GP-D	22	149	428	0·064
-G	17	219	329	0·065
F	60	68	272	0·062
HJ		77		0·015
NM		1022		0·153
P-D	25	966		0·114
-G	37	1783		0·197
R		801		0·183
S-D	41	622		0·111
-G	112	1117		0·230

d = converged with reset only.

TABLE 1—*continued*

PROBLEM 13 (Cragg and Levy, 1969)

$f(x) = [\exp(x_1) - x_2]^4 + 100(x_2 - x_3)^6 + \tan^4(x_3 - x_4) + x_1^8 + (x_4 - 1)^2$

$x^{(0)} = (1, 2, 2, 2); \quad x^* = (0, 1, 1, 1); \quad f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	96	424	388	0·180
-G				f
B-D	84	350	340	0·138
-G				f
P2-D	153	625	616	0·245
-G				f
P3-D	501	1753	2008	0·751
-G				f
PN-D	26	165	108	0·060
-G	31	439	128	0·110
FR-D	39	221	160	0·062
-G	72	680	292	0·156
CP-D	148	783	596	0·199
-G	157	1433	632	0·338
IP-D	121	634	488	0·164
-G	85	795	344	0·182
GP-D				f
-G	35	366	688	0·189
F	82	91	364	0·099
HJ		9283		1·52
NM		563		0·147
P-D	36	3480		0·980
-G	45	3103		0·728
R		955		0·585
S-D	128	1662		1·10
-G	339	3749		3·02

TABLE 1—*continued*

PROBLEM 14 (Wood)

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^3 + 10{\cdot}1(x_2 - 1)^2 + (x_4 - 1)^2 + 19{\cdot}8(x_2 - 1)(x_4 - 1).$$

The function has local minima that can cause premature termination.

$$x^{(0)} = (-3, -1, -3, -1); \qquad x^* = (1, 1, 1, 1); \qquad f(x^*) = 0$$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	57	475	232	0·085
-G	40	721	164	0·113
B-D	42	310	164	0·089
-G	40	664	172	0·092
P2-D				f
-G				f
P3-D	260	805	890	0·314e
-G				1·20e
PN-D				f
-G				f
FR-D				f
-G	189	3288	760	0·330
CP-D	136	704	548	0·188
-G	186	2745	748	0·410
IP-D	231	9202	928	2·42
-G	639	4264	2560	1·59
GP-D	16	367	250	0·079
-G	15	268	364	0·089
F	60	61	244	0·024
HJ		836		0·152
NM		797		0·154
P-D	25	276		0·041
-G	39	850		0·098
R		1043		0·378
S-D	38	715		0·171
-G	54	905		0·244

e = with reset gave best time.

TABLE 1—*continued*

PROBLEM 15

$$f(x) = \sum_{i=1}^{5} \frac{a_i + x_i^2}{1 + x_i^2}$$

where a_i is a random number in the interval (0·1, 1·0) *on each iteration.*

$x^{(0)} = (1, -1, 1, -1, 1)$; $x^* = (0, 0, 0, 0, 0)$; $f(x^*) = 0$

Algorithms	Search directions	Functional evaluations	Derivative evaluations	Time
DFP-D	21	800	110	0·308
-G	17	294	90	0·127
B-D	27	4511	140	1·56
-G	26	361	135	0·155
P2-D	15	207	80	0·092
-G	12	179	65	0·077
P3-D	18	1744	95	0·636
-G	16	162	85	0·076
PN-D				f
-G	27	450	140	0·195
FR-D				f
-G				f
CP-D	20	242	105	0·097
-G	18	228	95	0·086
IP-D	14	216	75	0·084
-G	10	123	55	0·051
GP-D				f
-G				f
F				f
HJ				f
NM				f
P-D				f
-G				f
R				f
S-D				f
-G				f

4. Evaluation of Unconstrained Algorithms

A number of comparisons of the effectiveness of algorithms have appeared in the literature (Bard, 1970; Box, 1965, 1966; Curtin, 1968; Fletcher, 1965; Huang and Levy, 1970; Jones, 1970; Leon, 1966; Murtagh and Sargent, 1970; Pearson, 1969; Pierson and Rajtora, 1970; Wortman, 1969). However, examination of the test results presented so far can only give a fragmentary picture of the relative effectiveness of the unconstrained algorithms because each study used different unidimensional search methods, different termination criteria, and different methods of counting equivalent function evaluations. A more desirable state of affairs would be to conduct an evaluation using a uniform set of standards and test problems.

To this end the following algorithms were tested on the problems listed in Table 1.

Analytical derivatives

(1) DFP: Davidon-Fletcher-Powell, rank 2 (Fletcher and Powell, 1963).
(2) B: Broyden (rank 1) (1965).
(3) P2: Pearson No. 2 (1969) without reset.
(4) P3: Pearson No. 3 (1969) without reset.
(5) PN: Projected Newton (Pearson, 1969).
(6) FR: Fletcher-Reeves (1964), reset each n + 1 iterations.
(7) CP: Continued Partan (Shah *et al.*, 1964).
(8) IP: Iterated Partan (Shah *et al.*, 1964).
(9) GP: Goldstein-Price (1967).
(10) F: Fletcher (1970).

Derivative free

(11) HJ: Hooke-Jeeves (1961).
(12) NM: Nelder-Mead (1965).
(13) P: Powell (1964).
(14) R: Rosenbrock (1960).
(15) S: Stewart (DFP with numerical derivatives) (1967).

Details of the logic of these algorithms can be found in the cited references, and also in Himmelblau (1972). The first 10 algorithms use analytical first derivatives, whereas the last five algorithms do not require analytical derivatives.

Two unidimensional searches were employed for each algorithm in which a unidimensional search was required: the Golden Section (G) and the DSC-Powell (designated D) search (Box *et al.*, 1967). Consequently, in effective 26

different algorithms were tested. The termination criteria for the unidimensional searches were all the same and the termination criteria for the algorithms were all the same being as follows:

1. termination of the unidimensional searches: The square of the norm of the vector between the current point to the middle point was than 10^{-6};
2. termination of the algorithm itself:
   (a) Relative change in $f(x) \leqslant 10^{-5}$ [or absolute change in $f(x) \leqslant 10^{-5}$ if $f(x) \to 0$];
   (b) relative change in $x_i \leqslant 10^{-5}$, $i = 1, \ldots, n$ (or absolute change in $x_i \leqslant 10^{-5}$ if $x_i \to 0$);
   (c) absolute change in elements of $\nabla f(x) \leqslant 10^{-4}$.

Of particular interest in the study are answers to the following questions:

1. which are the better and which are the poorer algorithms?;
2. how does the nature of the problem, that is the degree of non-linearity, the number of variables, and so forth affect the performance of an algorithm?;
3. how do the derivative-free algorithms compare with those that use derivatives?

Table 1 lists the times in seconds to execute each algorithm on a CDC 6600 computer together with the corresponding number of function evaluations, derivative evaluations, and iterations for the 15 test problems. Footnotes to the table designate the instances in which an incorrect answer was obtained, excessive time was required, and so forth. The blank spaces for the search algorithms exist because the derivative evaluation and search direction counts sometimes have no meaning comparable to the other algorithms. The term 'derivative evaluation' refers not to the number of gradient calls but to the number of derivatives evaluated so that if a gradient had three elements, one gradient call would yield three derivative evaluations. A derivative count is a better measure than a gradient call count because derivatives can vary widely in complexity. The term 'search directions' is equivalent to number of iterations in most variable metric and conjugate methods, but means little in the many of the derivative free methods, and hence the count has been omitted for some of the algorithms.

Each algorithm can be examined in relation to the character of the minimization problem by scanning the headings of Table 1. Not enough problems with many variables are included in this study to warrant an evaluation of the effect of dimensionality on the algorithms, but the effect of non-linearity can be observed. General problems such as 6 and 13 require more time for all the algorithms than the lower order functions, but nothing startling is observed.

Table 2, which lists the number of failures by algorithm in order of increasing number, shows the order of reliability or robustness with respect to the test problems. Because problem 15 was a stochastic problem, a count excluding problem 15 is also listed. For either case, a large group of algorithms exists of the same degree of robustness.

TABLE 2
Number of failures on 15 problems

No. of failures excluding Problem 15	Algorithm	No. of failures including Problem 15	Algorithm
0	FR-G	0	CP-G
	CP-G	1	DFP-D
	GP-G		DFP-G
	F		B-D
	HJ		B-G
	NM		PN-G
	P-D		FR-G
	P-G		CP-D
	R		GP-G
	S-G		F
1	DFP-D		HJ
	DFP-G		NM
	B-D		P-D
	B-G		P-G
	PN-G		R
	CP-D		S-G
	S-D	2	P2-D
2	P2-D		P3-D
	P3-D		P3-G
	P3-G		IP-D
	PN-D		IP-G
	FR-D		S-D
	IP-D	3	PN-D
	IP-G		FR-D
	GP-D		GP-D
>2	P2-G	>3	P2-G

To provide some insight as to the relative performance of the group of algorithms with less than two failures, the algorithms were ranked by their respective execution times. Table 3 gives the relative performance of the algorithms for the 14 deterministic problems. A value of 1·00 designates the algorithm with the smallest execution time, and the other entries in one column represent the respective times relative to 1·00 in increasing order.

TABLE 3
Ranking by Relative times

PROBLEM 1		PROBLEM 2		PROBLEM 3	
Relative time	Algorithm	Relative time	Algorithm	Relative time	Algorithm
1·00	F	1·00	P-G	1·00	B-D
1·50	DFP-D	1·31	B-G	1·17	P-D
1·75	PN-G	1·31	GP-G	1·25	DFP-D
1·75	CP-D	1·37	DFP-G	1·25	F
1·75	GP-G	1·37	F	1·58	DFP-G
2·00	DFP-G	1·63	S-G	1·75	B-G
2·00	B-D	1·87	DFP-D		
2·00	S-G	1·94	B-D		
2·25	B-G	2·19	NM		
2·25	CP-G	2·50	S-D		
2·25	HJ	2·69	PN-G		
2·50	FR-G	2·69	CP-G		
2·50	P-D	2·88	P-D		
4·50	R	2·94	FR-G		
6·00	NM	3·31	R		
6·00	P-G	3·50	HJ		
PROBLEM 4		PROBLEM 5		PROBLEM 6	
1·00	F	1·00	F	1·00	F
1·62	B-G	1·00	HJ	2·79	GP-G
1·62	P-G	2·00	DFP-D	2·92	B-D
1·69	DFP-G	2·00	CP-D	3·43	B-G
1·85	GP-G	2·00	S-G	3·62	P-D
1·85	HJ	2·13	DFP-G	3·65	DFP-D
1·85	S-G	2·13	B-D	4·63	CP-G
2·23	NM	2·13	PN-G	4·87	DFP-G
2·39	PN-G	2·13	P-D	5·74	NM
2·54	FR-G	2·38	FR-G	5·82	PN-G
3·00	P-G	2·50	B-G	6·10	P-G
3·15	CP-G	2·50	CP-G	6·95	R
4·00	R	2·50	GP-G	7·25	FR-G
		3·00	P-G	8·88	S-G
		3·13	NM	10·05	HJ
		3·25	R	23·8	CP-D

TABLE 3—*continued*

Relative time	Algorithm	Relative time	Algorithm	Relative time	Algorithm
PROBLEM 7		PROBLEM 8		PROBLEM 9	
1·00	F	1·00	F	1·00	HJ
2·25	B-D	1·40	P-D	1·83	F
2·50	DFP-D	1·47	B-D	2·33	DFP-D
3·50	HJ	1·60	S-G	2·50	B-D
3·50	P-D	1·93	DFP-G	3·67	B-G
4·25	DFP-G	2·20	B-G	3·67	GP-G
4·25	CP-G	2·26	DFP-D	4·00	S-G
4·50	B-G	2·53	PN-G	4·33	P-D
4·75	GP-G	2·53	GP-G	4·67	DFP-G
5·00	S-G	2·67	FR-G	6·08	PN-G
5·25	FR-G	3·27	R	6·58	P-G
5·25	CP-D	3·53	HJ	7·25	NM
8·00	PN-G	3·80	NM	9·50	R
14·2	P-G	3·93	CP-G	12·8	FR-G
16·7	R	4·00	P-G	40·7	CP-G
24·0	NM	4·27	CP-D	90·0	CP-D
PROBLEM 10		PROBLEM 11		PROBLEM 12	
1·00	P-D	1·00	B-D	1·00	HJ
2·12	F	1·41	DFP-D	4·13	F
2·76	DFP-D	1·62	F	4·27	B-G
2·76	B-D	1·90	B-G	4·33	GP-G
3·55	P-G	1·93	GP-G	4·73	B-D
3·59	GP-G	2·00	DFP-G	5·27	DFP-D
4·71	CP-D	2·10	P-D	7·47	DFP-G
5·00	B-G	2·17	S-G	7·60	P-D
5·12	DFP-G	2·69	CP-D	8·33	PN-G
5·29	S-G	2·76	P-G	10·2	NM
6·94	NM	3·76	PN-G	12·2	R
7·88	PN-G	4·66	CP-G	13·1	P-G
8·59	R	5·90	FR-G	15·3	S-G
12·6	HJ	7·79	NM	35·7	CP-G
13·1	FR-G			46·0	FRG
15·1	CP-G			110	CP-D

TABLE 3—*continued*

Relative time	Algorithm	Relative time	Algorithm
PROBLEM 13		PROBLEM 14	
1·00	F	1·00	F
1·11	PN-G	1·71	P-D
1·39	B-D	3·54	DFP-D
1·48	NM	3·71	B-D
1·57	FR-G	3·71	GP-G
1·82	DFP-D	3·83	B-G
1·91	GP-D	4·08	P-G
2·01	CP-D	4·71	DFP-G
3·41	CP-G	6·33	HJ
5·91	R	6·42	NM
7·34	P-G	7·83	CP-D
9·90	P-D	10·7	S-G
15·4	HJ	13·8	FR-G
30·5	S-G	15·8	R
		17·1	CP-G

In order to reduce the mass of data and reach some reasonably comprehensible conclusions, the algorithms were ranked according to their relative execution times, and the rankings averaged. If the algorithm failed, it was assigned a ranking of 14. Table 4 gives the overall ranking for the algorithms, each problem being weighted equally. It could be argued that the more difficult problems (in some sense) should be weighted more heavily than the easy problems, but this was not done. Although the absolute values of the function evaluations or execution times may have little meaning, the relative ranking of the algorithms should be a quite reasonable quantitative mechanism for evaluation. It was difficult to decide how to rank the algorithms that failed to solve a problem and those that failed to solve problem 15, but it was decided to include all the problems but No. 15, and to list the algorithms not solving a problem at the bottom of the ranking.

A surprising result of the ranking was that the algorithms tended to cluster into groups. Consequently, Table 4 lists the algorithms in Table 3 in increasing rank order according to execution times classed by the qualitative terms superior, good, fair, etc., a procedure that seems to be more meaningful for the limited set of test problems used than a continuous classification. Within each classification the order is that of the computed ranking.

Fletcher's algorithm was significantly better than all the others. As expected, the search algorithms were slower than many of the algorithms that used

derivatives, but what is of interest is the high ranking of Powell's algorithm. Problem 15 seemed to cause the derivative-free algorithms, and Fletcher's algorithms to fail, the significance of which is not fully understood. The algorithms not included in Table 3 or 4 are not generally recommended because they could terminate prematurely or could be ineffective because of unduly excessive use of computer time. In the derivative methods, slow oscillations in the search indicated that the direction matrix becomes nearly singular. Incorporation of an appropriate restart procedure restoring the direction

TABLE 4
Evaluation of unconstrained algorithms from execution times
(Numbers in parenthesis indicate average rank based on Table 3)

Classification	Algorithm
Superior	Fletcher (1·8)
Very Good	DFP-D (4·7)
	B-D (4·7)
Good	GP-G (5·6)
	B-G (6·1)
	P-D (6·1)
	DFP-G (6·8
Fair	S-G (8·8)
	HJ (9·6)
	PN-G (10·0)
	P-G (10·0)
	NM (10·0)
	FR-G (10·1)
	CP-G (11·2)
	R (11·9)
	CP-D (12·0)

matrix to a positive definite form can improve some of the variable metric methods, but this step will not lift the algorithm into the superior or very good class.

Attempts to compare quite different algorithms on the basis of the number of function and derivative evaluations can be less satisfactory than the use of execution times, particularly if the times are to be determined on the same computer using common subroutines and the problems are solved to the same degree of precision. The main obstacle to the use of the number of function and derivative evaluations is that these two quantities need to somehow be combined into a suitable single measure of 'equivalent evaluations'. It would be possible to evaluate the time to compute each derivative relative to each

function, and use the relative times as the weighting factors for the amalgamation, but even this procedure is subject to question. For example, the gradient components are evaluated much more often in the Goldstein-Price algorithm than in the Fletcher-Reeves algorithm leading to some relative distortion between these two algorithms. Consequently, ranking by 'equivalent function' evaluation count has not been carried out in this study but can be done by the interested reader from the data in Table 1. If the analysis were carried out it is fairly clear that the ranking of superior and very good algorithms would remian about the same as in Table 4.

5. Conclusions

Fifteen algorithms (26 when two different unidimensional searches are included) have been ranked according to execution time and robustness in solving fourteen test problems. The Fletcher algorithm was clearly superior to all the others, followed by the Davidon-Fletcher-Powell (rank 2) and Broyden (rank 1) algorithms using the DCS-Powell unidimensional search. Of particular interest to the user from the viewpoint of the program preparation time is the high ranking of the derivative free algorithm by Powell (1964) with the DCS-Powell unidimensional search.

Acknowledgement

The author is indebted to Michael Andenberg for his helpful and illuminating discussions and for his significant contribution in the preparation of the computer codes used in this study.

References

Bard, Y. (1970). Comparison of gradient methods for the solution of nonlinear parametric estimation problems. *SIAM J. Numer. Anal.* **7**, 159–186.

Beale, E. M. L. (1958). 'On an Iterative Method of Finding a Local Minimum of a Function of More than One Variable'. Technical Report No. 25, Statistical Techniques Research Group, Princeton University.

Box, M. J. (1965). A new method of constrained optimization and a comparison with other methods. *Comput. J.* **8**, 42–52.

Box, M. J. (1966). A comparison of several current optimization methods, and the use of transformations in constrained problems. *Comput. J.* **9**, 67–77.

Box, M. J., Davies, D., and Swann, W. H. (1969). 'Nonlinear Optimization Techniques'. ICI Monograph No. 5, Oliver and Boyd, London.

Broyden, C. G. (1965). A class of methods for solving nonlinear equations. *Math. Comput.* **19**, 577–584.

Cragg, E. E., and Levy, A. V. (1969). Study on a supermemory gradient method for the minimization of functions. *J. Optim. Theory Applns* **4**, 191–205.

Curtin, J. F. (1968). 'The Application of Direct Search and Descent Methods of Function Minimization to Parameter Estimation'. Dept. of Supply, Weapons Res. Establishment, Salisbury, Australia.

Engvall, J. L. (1966). 'Numerical Algorithm for Solving Over-determined Systems of Nonlinear Equations'. NASA Document N70-35600.

Fletcher, R. (1965). Function minimization without evaluating derivatives—a review. *Comput. J.* **8**, 33–41.

Fletcher, R. (1970). A new approach to variable metric algorithms. *Comput. J.* **13**, 317–322.

Fletcher, R., and Powell, M. J. D. (1963). A rapidly convergent descent method for minimization. *Comput. J.* **6**, 163–168.

Fletcher, R., and Reeves, C. M. (1964). Function minimization by conjugate gradients. *Comput. J.* **7**, 149–154.

Goldstein, A. A., and Price, J. F. (1967). An effective algorithm for minimization. *Num. Math.* **10**, 184–189.

Himmelblau, D. M. (1972). 'Applied Nonlinear Programming'. McGraw-Hill, New York.

Hooke, R., and Jeeves, T. A. (1961). 'Direct search' solution of numerical and statistical problems. *J. Ass. comput. Mach.* **8**, 212–221.

Huang, H. Y., and Levy, A. V. (1970). Numerical experiments on quadratically convergent algorithms for function minimization. *J. Optim. Theory Applns* **6**, 269–282.

Jones, A. (1970). Spiral—A new algorithm for nonlinear parameter estimation using least squares. *Comput. J.* **13**, 301–308.

Leon, A. (1966). A Comparison among eight known optimizing procedures. *In*: 'Recent Advances in Optimization Techniques' (A. Lavi and T. P. Vogl, eds.), pp. 23–42. John Wiley, New York.

Murtagh, B. A., and Sargent, R. W. H. (1970). Computational experience with quadratically convergent minimization methods. *Comput. J.* **13**, 185–194.

Nelder, J. A., and Mead, R. (1965). A simplex method for function minimization. *Comput. J.* **7**, 308–313.

Pearson, J. D. (1969). Variable metric methods of minimization. *Comput. J.* **12**, 171–178.

Pierson, B. L., and Rajtora, S. G. (1970). Computation experience with the Davidon method applied to optional control problems. *IEEE Trans.* **SSC4**, 240–242.

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7**, 155–162.

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. *Comput. J.* **3**, 175–184.

Schmidt, J. W., and Vetters, K. (1970). Ableitungsfreie Verfahren für nichtlineare Optimierungsprobleme. *Numer. Math.* **15**, 263–282.

Shah, B. V., Beuhler, R. J., and Kempthorne, O. (1964). Some algorithms for minimizing a function of several variables. *J. Soc. ind. appl. Math.* **12**, 74–92.

Stewart, G. W. (1967). A modification of Davidon's minimization method to accept difference approximations to derivatives. *J. Ass. comput. Mach.* **14**, 72–83.

White, B. F., and Holst, W. R. (1964). Paper submitted at the 1946 Spring Joint Computer Conf., Washington, D.C.

Wood, C. F., Westinghouse Research Laboratories, Pittsburg, Pa.

Wortman, J. D. (1969). NLPROG, Ballistic Research Laboratories Memorandum Report No. 1958, Aberdeen Proving Ground, Md.

Zangwill, W. I. (1967). Minimizing a function without derivatives. *Comput. J.* **10**, 293–296.

Zwart, P. (1970). Unimodal functions. *J. Optim. Theory Applns* **6**, 155–156.

10. The Choice of Step Length, a Crucial Factor in the Performance of Variable Metric Algorithms

L. C. W. DIXON
The Numerical Optimization Centre, The Hatfield Polytechnic, Hatfield, Hertfordshire, England

Summary

A number of different variable metric algorithms have recently been proposed and many reports containing numerical examples of their performance are now available. In many of these reports different choices of step length have been used, thus complicating the comparison of their relative performance. At first sight this might appear to be an effect of secondary importance. It is the aim of this paper to show that it is crucial.

Many of these algorithms belong to a family introduced by Huang (1970), who showed that, provided the step length is determined by an accurate line search, then all these algorithms produce an identical sequence of points on a quadratic function. Dixon (1971) gives conditions for a subset of this family to generate identical points on more general functions. It will be shown that the members of the Broyden (1967) family satisfy these conditions. As most implementations do not undertake accurate line searches, it follows that the reported differences in performance of members of this family are crucially dependent on the choice of step length.

1. Introduction

The suggestion by Davidon (1959) that the inverse Hessian matrix of a function can be constructed iteratively during the solution of an n-dimensional minimization problem provides the basis of many optimization algorithms. Since the well-known implementation by Fletcher and Powell (1963) was found to fail under certain circumstances (Bard, 1968) a considerable number of alternative updating formulae have been suggested.

Recently, it has been shown that these formulae can be combined into families, and the properties of these families have been theoretically investigated. Regrettably, most of these theoretical studies only apply to quadratic functions. Two families will be discussed in this paper: the one parameter

family introduced by Broyden (1967) and discussed in detail by Broyden (1970), Fletcher (1970), Shanno (1970) and Goldfarb (1970), and the more general family introduced by Huang (1970).

Many of the above papers include numerical data gathered by testing implementations on standard test functions. In these comparisons many different updating formulas have been used in conjunction with different strategies for determining the step length. In this paper some theoretical properties of the behaviour of these algorithms on general functions are presented. The published numerical data is then examined, with emphasis on these theoretical properties of the different strategies and supplemented by a consistent set of data.

2. Theoretical Properties

2.1. *Properties of Huang's Family*

In Huang (1970) a family of updating formulae is introduced given by

$$H_i = H_{i-1} + \rho_{i-1} \frac{\Delta x_{i-1}(C_1 \Delta x_{i-1} + C_2 H_{i-1}^T \Delta g_{i-1})^T}{(C_1 \Delta x_{i-1} + C_2 H_{i-1}^T \Delta g_{i-1})^T \Delta g_{i-1}} +$$

$$- \frac{H_{i-1} \Delta g_{i-1}(K_1 \Delta x_{i-1} + K_2 H_{i-1}^T \Delta g_{i-1})^T}{(K_1 \Delta x_{i-1} + K_2 H_{i-1}^T \Delta g_{i-1})^T \Delta g_{i-1}}, \tag{1}$$

where the parameters ρ_{i-1}, $C^{(i-1)} = C_1/C_2$, $K^{(i-1)} = K_1/K_2$ provide three degrees of freedom at each iteration.

He considered the following implementation. At a point x_i calculate f and g_i and hence the direction

$$p_i = H_i^T g_i. \tag{2}$$

Choose a step length α_i so that

$$x_{i+1} = x_i + \Delta x_i \tag{3}$$

and

$$\Delta x_i = -\alpha_i p_i, \tag{4}$$

where α_i is chosen such that

$$g_{i+1}^T p_i = 0. \tag{5}$$

The matrix H is then updated using (1) where

$$\Delta g_{i-1} = g_i - g_{i-1}, \tag{6}$$

and the process repeated until a point is found at which

$$g_i^T g_i < \epsilon_1. \tag{7}$$

Huang (1970) showed theoretically that all members of family (1) generate the same sequence of points x_i when applied to a quadratic function. Dixon (1971) extended this result by considering the behaviour on any differentiable function and showed that

$$p_i = -bu_i \tag{8}$$

where

$$u_i = \left\{\prod_{j=0}^{i-1}\left(I - \frac{\Delta x_j \Delta g_j^T}{\Delta x_j^T \Delta g_j}\right)\right\} H_0^T g_i + \\ + \sum_{j=0}^{i-2}\left\{\prod_{k=j+1}^{i-1}\left(I - \frac{\Delta x_k \Delta g_k^T}{\Delta x_k^T \Delta g_k}\right)\right\}\left(\rho_j \frac{\Delta x_j^T g_i}{\Delta x_j^T \Delta g_j}\right)\Delta x_j \tag{9}$$

and

$$b = + \frac{(\alpha_{i-1} K_1 + K_2)\Delta g_{i-1}^T H_{i-1}^T g_{i-1}}{(K_1 \Delta x_{i-1} + K_2 H_{i-1}^T \Delta g_{i-1})^T \Delta g_{i-1}}. \tag{10}$$

Suppose that at any iteration

$$\alpha_{i-1} K_1 + K_2 \neq 0 \qquad \text{and} \qquad (K_1 \Delta x_{i-1} + K_2 H_{i-1}^T \Delta g_{i-1})^T \Delta g_{i-1} \neq 0. \tag{11}$$

Let us extend the definition (5) of α_i, so that α_i is defined uniquely on non-convex functions. Then all members of the family having the same values of ρ_i generate identical sequences x_i on any general function, provided conditions (11) are not satisfied. (The proof remains valid if ρ_i varies from iteration to iteration.)

The unique value of α_i can be obtained by accepting the first local minimum in the downhill direction $\pm p_i$.

It will be noted that these necessary and sufficient conditions agree with the numerical data in Huang and Levy (1970) who tested four algorithms with $\rho_i = 1$ and three with $\rho_i = 0$ and noted that these produced two identical sequences of points.

2.2. *Properties of Broyden's Family*

In the notation given above the original Davidon-Fletcher-Powell (DFP) formula is given by

$$H_{\text{DFP}} = H + \frac{\Delta x \Delta x^T}{\Delta x^T \Delta g} - \frac{H \Delta g \Delta g^T H}{\Delta g^T H \Delta g} \tag{12}$$

which is a member of Huang's family with $\rho = 1$, $C = \infty$, $K = 0$ and was included in the Huang and Levy test series.

In Broyden (1967) the family of formulae given by

$$H_i = H_{\text{DFP}} + \beta \left(\frac{H\Delta g \, \Delta g^T H}{\Delta g^T H \Delta g} (\Delta g^T \Delta x) - H\Delta g \Delta x^T - \Delta x \Delta g^T H + \frac{\Delta x \Delta x^T}{\Delta x^T \Delta g} \Delta g^T H \Delta g \right) \tag{13}$$

is introduced. In Broyden (1970) the properties of this family are examined for a quadratic function, and for reasons of stability in the convergence of the sequence H_i to G^{-1}, the value

$$\beta = \frac{1}{\Delta x^T \Delta g} \tag{14}$$

is recommended. In this report the combination of (13) and (14) will be denoted as the Broyden-Fletcher-Shanno (BFS) formula H_{BFS}.

The same family is considered in Fletcher (1970) but in the form

$$H_i = (1 - \phi) H_{\text{DFP}} + \phi H_{\text{BFS}} \tag{15}$$

and also in Shanno (1970), where it is written as

$$H_i = H_{i-1} + t \frac{\Delta x \Delta x^T}{\Delta x^T \Delta g} + \frac{[(1-t)\Delta x - H\Delta g][(1-t)\Delta x - H\Delta g]^T}{[(1-t)\Delta x - H\Delta g]^T \Delta g}. \tag{16}$$

An important member of this family is the symmetric rank-one (R1) formula

$$H_{\text{R1}} = H + \frac{(\Delta x - H\Delta g)(\Delta x - H\Delta g)^T}{(\Delta x - H\Delta g)^T \Delta g} \tag{17}$$

which corresponds to $\rho = 1$, $C = -1$, $K = -1$ and was also included in the Huang and Levy test series. Wolfe (1967) showed that with this formula it is unnecessary to satisfy condition (5) if the sequence $\{H_i\}$ is to tend to G^{-1} in a finite number of steps on a quadratic function.

For this reason the author prefers to consider the family in the form

$$H = H_{\text{R1}} - avv^T \left(\frac{(\Delta x^T \Delta g)(\Delta g^T H \Delta g)}{(\Delta x - H\Delta g)^T \Delta g} \right) \tag{18}$$

where

$$v = \frac{\Delta x}{\Delta x^T \Delta g} - \frac{H\Delta g}{\Delta g^T H \Delta g} \tag{19}$$

leads to Broyden's family but is not the only possible choice of v. This form emphasizes the rank-two nature of the update and also demonstrates that it is the presence of the terms avv^T, introduced for stability reasons, that introduces

TABLE 1

Algorithm	Broyden		Fletcher	Shanno		Huang's Family	
	a	β	ϕ	t	ρ	C	K
RANK 1	0	$\dfrac{1}{(\Delta x - H\Delta g)^T \Delta g}$	$\dfrac{\Delta x^T \Delta g}{(\Delta x - H\Delta g)^T \Delta g}$	0	1	-1	-1
H_{DFP}	1	0	0	1	1	∞	0
H_{BFS}	$\dfrac{\Delta g^T H \Delta g}{\Delta x^T \Delta g}$	$\dfrac{1}{\Delta x^T \Delta g}$	1	∞	1	$\dfrac{-(\Delta x + H\Delta g)^T \Delta g}{\Delta x^T \Delta g}$	∞
H	a	$\dfrac{1-a}{(\Delta x - H\Delta g)^T \Delta g}$	$\dfrac{(1-a)\Delta x^T \Delta g}{(\Delta x - H\Delta g)^T \Delta g}$	$\dfrac{a(\Delta x - H\Delta g)^T \Delta g}{(a\Delta x - H\Delta g)^T \Delta g}$	1	$\dfrac{(\Delta x - aH\Delta g)^T \Delta g}{(a-1)\Delta x^T \Delta g}$	$\dfrac{-(a-i)\Delta g^T H \Delta g}{(a\Delta x - H\Delta g)^T \Delta g}$

the need for condition (5). The relationship between these expressions is given in Table 1.

Fletcher (1970) demonstrated that for stability reasons a must lie at or between the limits corresponding to the DFP and BFS algorithms, and recommended using H_{DFP} if

$$\Delta g^T H \Delta g > \Delta x^T \Delta g$$

or otherwise switching to H_{BFS}.

It follows from the table that all members of the Broyden (1967) family are members of Huang's family with $\rho_i = 1$, and hence belong to a subset. If α_i were chosen to satisfy (5) and (11) they would all generate identical sequences $\{x_i\}$. The fact that this had never been observed prior to Huang and Levy (1970) led the author to undertake an analysis of the published data on this family. Before proceeding to this analysis some comments will be made on the likely effect of common safeguards on these identical sequences.

2.3. *The Predicted Step Length*

From formula (9) it follows that members of Huang's family with identical ρ_i generate the same directions and hence when combined with conditions 5 and 12 the same sequence $\{x_i\}$. However,the initial prediction

$$p_i = -bu_i \tag{20}$$

depends on the value of b. As

$$b = + \frac{(\alpha K + 1)\, \Delta g_{i-1}^T H_{i-1}^T g_{i-1}}{(K \Delta x_{i-1} + H_{i-1}^T \Delta g_{i-1})^T \Delta g_{i-1}} \tag{21}$$

this obviously depends upon K, and hence is different for each member of Broyden's family. It will be noted that b can take positive, negative, zero and infinite values, and while b positive corresponds to a downhill direction, the others are not so convenient.

If $b < 0$ then the initial slope is uphill and the matrix must be non-positive definite. Theoretically this cannot occur with the H_{DFP} or H_{BFS} updating formula, but it is possible with H_{R1} when most implementations reject direction (19) and reset H. This policy destroys the sequence $\{x_i\}$.

If $\alpha_i = -1/K$, then $b = 0$, and the next step identically vanishes. This result has been noted before for the rank-one formula ($K = -1$).

If $\Delta x^T \Delta g = -\Delta g^T H \Delta g / K$, then division by zero would have to be prevented.

Neither of these difficulties can arise when $K = 0$ or ∞, which correspond to the DFP and BFS formulae.

Of the three commonly used members of the family only the rank-one needs

to be protected for the last two difficulties. In the Sargent and Murtagh (1970) implementation the safeguard for $\alpha = 1$ is to retain

$$H_i = H_{i-1} \tag{22}$$

and the safeguard for division by zero is to replace the formulae by

$$H_i = H_{i-1} + \frac{(\Delta x - H\Delta g)(\Delta x - H\Delta g)^T}{(\Delta x - H\Delta g)^T(\Delta x - H\Delta g)} \tag{23}$$

which is not a member of Huang's family. The use of (22) or (23) destroys the sequence $\{x_i\}$ and to obtain a consistent sequence the author found it necessary to treat b and u_i separately in (20) restricting extreme values of b and accepting uphill directions. It is not implied that this necessarily improves the efficiency.

3. Numerical Evidence

In order to confirm that the above behaviour occurred, some detailed runs were undertaken. In these runs a parabolic interpolation process was used. As it was necessary to find the minimum exactly, a highly safeguarded search was employed. The fact that simple parabolic routines could terminate at points other than the minimum has been reported in an earlier paper (Dixon, 1969). The procedure used, first evaluated the function at

$$|\alpha^{(0)}| = \min(1, s)$$

where s was a safety limit, the step being taken in the downhill direction. A parabola was then fitted to the known function value and slope at $\alpha = 0$ to give a further prediction $\alpha^{(1)}$.

If $|\alpha^{(1)}| > |\alpha^{(0)}|$ and a bracket had not been formed, then the outerpoint was extended by a constant factor 5 until a bracket occurred. If $|\alpha^{(1)}| < |\alpha^{(0)}|$ and a bracket had not been found, then $\alpha^{(0)} := \alpha^{(1)}$ and a parabola was fitted again to the function value and slope at $\alpha = 0$ until a bracket occurred. Once a bracket $L < \alpha < M$ has been found, parabolic interpolation was employed; if the prediction $\alpha^{(E)} = L + E(M - L)$ and $0{\cdot}25 < E < 0{\cdot}75$ then $\alpha^{(E)}$ was accepted, otherwise $E = 0{\cdot}25$ or $0{\cdot}75$ was preferred. If this point was an improvement then the new bracket was established and the process repeated. If the point was not an improvement then the far end point was also rejected in favour of a point midway between it and the current best point; the bracket enclosing the best of these points was then accepted. This process ensures that on a skew function one quarter of the bracket round the minimum is removed at each interpolation.

The point was accepted when either the new predicted value $|\alpha^{(E)}|$ agreed with the current value $|\alpha^{(c)}|$ within $\epsilon_2|\alpha^{(c)}|$, or the bracket itself was reduced to $\epsilon_2|\alpha^{(c)}|$ in size. Using this procedure with $\epsilon_2 = 10^{-7}$ on Powell's function gave

TABLE 2
Behaviour on Powell's function: accurate line search

Iteration number	DFP	BFS	Fletcher Switch	Rank-one	Rank-one safeguarded
1	30·8302	30·8302	30·8302	30·8302	30·8302
2	18·5408	18·5408	18·5408	18·5408	18·5408
3	10·4095	10·4095	10·4095	10·4095	10·4095
4	2·93573 E-2	2·93559 E-2	2·93573 E-2	2·93556 E-2	2·93556 E-2
5	2·31547 E-2	2·31539 E-2	2·31547 E-2	2·31537 E-2	2·31537 E-2
6	7·06969 E-3	7·06806 E-3	7·06418 E-2	7·05737 E-3	7·05737 E-3
7	4·47354 E-3	4·4725 E-3	4·47069 E-3	4·46561 E-3	4·46561 E-3
8	2·13856 E-3	2·13865 E-3	2·13918 E-3	2·14008 E-3	2·14008 E-3
9	1·99420 E-3	1·99509 E-3	1·99756 E-3	2·00075 E-3	2·00075 E-3
10	1·32662 E-3	1·33148 E-3	1·34282 E-3	1·35640 E-3	1·94696 E-3
11	1·28077 E-4	1·19872 E-4	1·19219 E-4	1·18368 E-4	6·95484 E-5
12	4·47069 E-5	4·23202 E-5	4·21829 E-5	4·19831 E-5	1·75158 E-5
13	6·15553 E-6	5·54279 E-6	5·42274 E-6	5·32000 E-6	4·83724 E-7
14	1·63994 E-6	1·48582 E-6	1·45576 E-6	1·42964 E-6	9·79914 E-8
15	2·20355 E-7	1·87257 E-7	1·79436 E-7	1·72515 E-7	1·46662 E-9
16	5·86361 E-8	5·00679 E-8	4·79858 E-8	4·61418 E-8	2·39827 E-10
17	7·15174 E-9	5·62893 E-9	5·22315 E-9	4·88093 E-9	1·84082 E-12
18	1·89154 E-9	3·94280 E-9	3·62250 E-9	3·33838 E-9	C
19	1·81931 E-9	1·36482 E-9	1·27209 E-9	1·20459 E-9	
20	C	2·65064 E-10	2·62303 E-10	2·88125 E-10	

the function values in Table 2; it will be noted that the first four columns which should theoretically be identical are remarkably consistent. The amount of computer rounding error is very small. The fifth column gives the numbers for a rank-one process in which uphill directions were rejected and steepest descent used instead; this occurred at iteration 10 and in this case lead to improved convergence. When the runs were repeated with less exact line searches, $\epsilon_2 = 10^{-4}$ for example, the similarity in the first four columns was no longer obvious. As no implementation (known to the author) attempts to find the minimum on a non-quadratic function as accurately as this, the fact that this property had not been observed before is not too surprising.

4. Alternative Strategies for Choosing the Step Length

4.1. *Acceptability Criteria*

In Section 2 implementations of the variable metric idea were considered that assumed that the step length α_i was chosen to satisfy condition (5) namely

$$g_{+1}^T p_i = 0.$$

Very few implementations have attempted to satisfy this condition in practice, and of those that have, e.g. Broyden (1970), there is little indication that the tolerances used were similar to those discussed in Section 3.

Most implementations are satisfied if an improved point has been found by an interpolation process, either cubic or parabolic. Such an implementation would, of course, find the true minimum along the line on a quadratic function. The implementations due to Fletcher and Powell (1963) and Fletcher (1966) are of this form.

More recently, following Goldstein and Price (1967), it has been suggested that a point may be accepted if it gives an acceptable reduction in function value. The acceptability conditions are often a combination of the following conditions.

I. The following conditions are often used to prevent the step size from being too large:

(1) $f(x_i) - f(x_{i+1}) > 0$. This has now been shown to be unsatisfactory, Fletcher (1970b).

(2) $f(x_i) - f(x_{i+1}) > \epsilon_3 \alpha g_i^T H_i g_i$.

(3) $f(x_i) - f(x_{i+1}) > \epsilon_3 \delta f_Q$;

where $\alpha g_i^T H_i g_i$ would be the reduction in f if the function were truly linear and δf_Q the reduction if it were truly quadratic. As these conditions must be satisfied as $\alpha \to 0$, their failure indicates that the step tried is probably too large. In the implementations leading to the results discussed in Section 5.2, condition (2) was used with $\epsilon_3 = 0{\cdot}1$.

II. If

$$(1-\epsilon_3)\,\alpha g_i^T H_i g_i < f(x_i) - f(x_{i+1}) < (1+\epsilon_3)\,\alpha g_i^T H_i g_i$$

then the move taken has been effectively linear and should be increased. This is a slight modification of a test given in Goldstein and Price (1967). The value $\epsilon_3 = 0{\cdot}1$ was used to obtain the values given in Section 5.2.

III. Positive-definite step. In most implementations positive definite matrices are required and it is usual only to accept a point that is consistent with this condition, i.e. is such that $\Delta x^T \Delta g > 0$.

Alternative versions of conditions I and II above, together with a discussion of their convergence properties, is given in Wolfe (1969).

4.2. *Search Procedure*

Implementations will be given a four-digit code. The *first* digit will be A if the line search is nominally accurate, B if a bracketed point is accepted and C if a combination of conditions I–III is used.

Three main aspects of the search procedure will be distinguished in the rest of the code: the initial step, any subsequent interpolation, and any subsequent extrapolation.

The *second* letter in the code will denote the *initial step.*

A will denote

$$\alpha^{(0)} = 1.$$

B will denote

$$\alpha^{(0)} = \min(1, s),$$

where s is a safety limit. Commonly used values of s have been

$$s = 2(f - est)/g^T Hg, \text{ if } f > est,$$

$$s = 1 \text{ otherwise,}$$

where *est* is the estimated value of the function value at the minimum; or

$$s = 2|\Delta x_{i-1}|/|Hg|,$$

using the previously successful step to limit the subsequent initial step.

C will denote a more general system where the initial value of α is determined by the previous behaviour.

The *third* letter of the code will denote the *interpolation procedure.* The main feature will be coded where alternatives are possible.

A will denote constant factor reduction.

B will denote parabolic interpolation.

C will denote cubic interpolation.

E will include any other policy.

The *fourth* and final letter of the code will denote the *extrapolation procedure*.
A will denote constant factor extrapolation.
B will denote parabolic extrapolation.
E will include any other policy.

As an example of the use of this code, the algorithm described in Fletcher and Powell (1963) is B, BCA. It accepts a bracketed point, applies a safety limit to the first step, and uses cubic interpolation and constant factor extrapolation.

4.3. *Published Implementations*

In Table 3 the codings of a number of published implementations are listed. This table is no doubt incomplete but indicates the distribution of categories that have been tried.

In preparing Table 3 it became obvious that there were notable gaps; there were no DFP or BFS results in which the acceptability criteria were applied, whilst the switching policy had only been reported in combination with these

TABLE 3
Codings of published algorithms

Implementation		Line search A	Bracket B	Conditions C
DAVIDON-FLETCHER-POWELL FORMULA				
Fletcher-Powell	63		BCA	
Fletcher	66		CCA	
Stewart[a]	67	BBB		
Fletcher-Lill[a]	68		CBB	
Pearson	69	EA[b]		
Shanno	70		ACA	
Greenstadt	70	BBA	BBA	
Broyden	70	BBB		
Bard	70	CBB		
Murtagh and Sargent	70	BCA		
Biggs	70	CCA	CCA, ACB	
Huang and Levy	70	[b]		
Fletcher	70		CCA	
This report		BBA	BBA, ACB	BBA
BROYDEN-FLETCHER-SHANNO FORMULA				
Shanno	70		ACA	
Broyden	70	BBB		
Shanno and Kettler	70		ACB	
Biggs	70		ACB, CCA	
This report		BBA	BBA, ACB	BBA

TABLE 3—*continued*

Implementation		Line search A	Bracket B	Conditions C
		RANK-ONE		
Powell	69			Special
Shanno	70		ACA	
Bard	70		CBB	CBB
Murtagh and Sargent	70	BCA		ACE, BCE
Huang and Levy	70	[b]		
This report		BBA	BBA, ACB	BBA
		SWITCHING		
Fletcher	70			ACA
This report		BBA	BBA, ACB	BBA

The first letter of the code has been transferred to the head of the column.
[a] Indicates the use of approximate derivatives.
[b] Indicates that the relevant information was not available to the author when writing this paper.

criteria. There were very few comparisons of the effect of parabolic as distinct from cubic interpolation and no direct comparison of all four updating procedures. It was therefore decided to complete the table by including results obtained on all twelve combinations using the parabolic approach and all four updating policies using the cubic interpolation bracketing policy. Before introducing a more complete comparison based on these results it seemed desirable to see how consistent the published data was. For consistency comparisons only the most commonly tested functions could be used and only three seemed to be sufficiently mutual for conclusions to be drawn. These were the Rosenbrock, Powell and Wood quartic functions.

5. Numerical Results

5.1. *Accurate Line Search Routines and Comparison of Iterations Required*

The published data giving the number of iterations required for convergence when accurate line searches are undertaken is given in Table 4.

New data derived especially for this report is also included. The iteration numbers are remarkably consistent considering the different interpretations of 'accurate line search' and the different termination criteria that are, no doubt, combined there. It is known that the number of iterations required for convergence on Powell's quartic is very sensitive to the termination criteria whilst the other functions are relatively insensitive and this no doubt explains the

higher figures required by Huang and Levy on this function. The other variations in the table are all well within the scatter found by the author during tests with different values of ϵ_2. For instance with $\epsilon_2 = 10^{-5}$, on Rosenbrock's function, the small differences in starting points on iteration 11 when using the DFP and BFS formulae were enough to produce very different unidirectional minima, separated by a region of high function values, and the subsequent convergence favoured the BFS formulae by four iterations (a very similar result to that reported by Broyden, 1970).

TABLE 4
Comparison of iteration numbers with accurate line search routines

Algorithm details				Function details		
Formula	Author		Code	Rosenbrock	Powell	Wood
DFP	Stewart	67	A, BBB[a]	24	19	
	Pearson	68	A, EA[b]	19		40
	Greenstadt	70	A, BBA	26	18	
	Broyden	70	A, BBB	23	18	
	Huang and Levy	70	A,[b]	22	32	40
	This report		A, BBA	19	20	34
BFS	Broyden	70	A, BBB	19	26	
	This report		A, BBA	19	21	40
Fletcher switch	This report		A, BBA	19	21	40
Rank-one	Huang and Levy	70	A,[b]	22	32	40
	This report		A, BBA	19	21	40

[a] Indicates the use of approximate derivatives.
[b] Indicates that the relevant information was not available to the author, when writing this paper.

In conclusion, it can be stated that, when the iteration is undertaken sufficiently accurately, then the numerical evidence confirms that the iteration pattern is independent of updating formulae in this family.

5.2. *Systematic Comparison*

To evaluate the significance of the different choices of updating formula and step-length, a series of tests was undertaken using all sixteen permutations described above on twelve test functions. These functions include some standard test functions, a series of exponential functions, and three new functions, a skew penalty function of the type generated in S.U.M.T., and two new

functions with singular matrices at the minimum, one ROS(8) having a flat minimum formed by raising both brackets in the usual Rosenbrock function to power 8 and the other RECIP having a cusp at the minimum. These twelve functions are divided into two groups for discussion purposes: Group A (nine functions) on which most methods were successful and Group B (three functions) on which the picture was more complex. The details of the functions are given below:

1. ROS(2), introduced by Rosenbrock (1960):

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Starting approximation: $x = (-1,2,1)$.

2. POW, introduced by Powell (1962):

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

Starting approximation: $x = (3,-1,0,1)$.

3. WOOD, introduced by Wood (cited in Colville, 1968):

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10{\cdot}1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19{\cdot}8(x_2 - 1)(x_4 - 1).$$

Starting approximation: $x = (-3,-1,-3,-1)$.

4. BOX(2), introduced by Box (1966):

$$f(x) = \sum_{i=1}^{10} [\exp(-x_1 z_i) - \exp(-x_2 z_i) - \exp(-z_i) + \exp(-10z_i)]^2,$$

$(z_i = i/10)$.

Starting approximation: $x = (5,0)$.

5. EXP(2), introduced by Biggs (1971):

$$f(x) = \sum_{i=1}^{10} [\exp(-x_1 z_i) - 5\exp(-x_2 z_i) - \exp(-z_i) + 5\exp(-10z_i)]^2,$$

$(z = i/10)$.

Starting approximation: $x = (1,2)$.

6. EXP(3), introduced by Biggs (1971):

$$f(x) = \sum_{i=1}^{10} [\exp(-x_1 z_i) - x_3\exp(-x_2 z_i) - \exp(-z_i) + 5\exp(-10z_i)]^2,$$

$(z_i = i/10)$.

Starting approximation: $x = (1,2,1)$.

7. EXP(4), introduced by Biggs (1971):

$$f(x) = \sum_{i=1}^{10} [x_3 \exp(-x_1 z_i) - x_4 \exp(-x_2 z_i) - \exp(-z_i) + 5\exp(-10z_i)]^2,$$

$(z_i = i/10)$.

Starting approximation: $x = (1, 2, 1, 1)$.

8. PEN:

$$f(x) = (x_1 - 5)^2 + x_2^2 + 10^{-4}/(x_2 - x_1^2), \qquad \text{if } x_2 > x_1^2.$$

Starting approximation: $x = (2, 5)$.

9. ROS(8):

$$f(x) = 100(x_2 - x_1^2)^8 + (1 - x_1)^8.$$

Starting approximation: $x = (-1{\cdot}2, 1)$.

10. EXP(5), introduced by Biggs (1971):

$$f(x) = \sum_{i=1}^{11} [x_3 \exp(-x_1 z_i) - x_4 \exp(-x_2 z_i) + 3\exp(-x_5 z_i) - \exp(-z_i) + 5\exp(-10z_i) - 3\exp(-4z_i)]^2, \quad (z_i = i/10).$$

Starting approximation: $x = (1, 1, 1, 1, 1)$.

11. WEIBULL, introduced by Shanno (1970):

$$f(x) = \sum_{i=1}^{99} \left[\exp\left(-\frac{(y_i - x_3)^{x_2}}{x_1} \right) - z_1 \right]^2,$$

where

$$y_i = 25 + [50 \log_e (1/z_i)]^{2/3}, \qquad (z_i = i/100).$$

Starting approximation: $x = (250, 0{\cdot}3, 5)$.

12. RECIP:

$$f(x) = (x_1 - 5)^2 + x_2^2 + x_3^2/(x_2 - x_1^2), \qquad \text{if } x_2 > x_1^2.$$

Starting approximation: $x = (2, 5, 1)$.

The detailed runs on the first nine easier functions are presented in Table 5, in which the upper of each pair of numbers represents the number of equivalent function evaluations required and the lower the number of iterations. The total number of equivalent function evaluations required is given in the final column.

TABLE 5
Comparison of performance on nine test functions

Function		ROS(2)	Powell	Wood	BOX(2)	EXP(2)	EXP(3)	EXP(4)	PEN	ROS(8)	Total
					ACCURATE LINE SEARCH						
DFP	A, BBA	267 19	352 20	564 34	179 11	85 6	151 11	316 19	191 12	135 8	2240
BFS		261 19	334 21	599 40	111 8	88 6	151 11	308 20	170 12	111 7	2133
Switch		260 19	329 21	619 40	159 11	86 6	147 11	288 19	184 12	111 7	2183
R1		266 19	290 18	553 39	108 8	87 6	152 11	296 19	190 13	130 8	2072
					PARABOLIC BRACKETING TECHNIQUE						
DFP	B, BBA	160 33	256 33	885 122	90 17	64 13	79 13	416 57	249 55	134 25	2333
BFS		132 29	136 19	320 49	53 11	53 11	67 11	152 22	121 26	71 12	1105
Switch		138 30	203 28	341 50	53 11	54 11	73 12	182 26	181 38	90 16	1315
R1		155 32	291 40	407 60	49 10	52 11	73 12	132 19	267 56	98 18	1524

ACCEPTABLE POINT TECHNIQUE

DFP	C, BBA	138	1035	F	361	172	155	439	208	511	3019F
		39	180		110	54	36	84	53	157	
BFS		103	134	298	62	56	78	172	112	134	1149
		31	25	56	17	16	18	33	28	42	
Switch		121	189	370	67	56	87	147	228	138	1403
		36	35	70	18	16	19	27	56	43	
R1		164	178	407	41	47	64	164	270	129	1464
		46	32	72	10	13	14	30	67	40	

CUBIC BRACKETING TECHNIQUE

DFP	B, ACB	210	280	705	F	63	128	305	267	150	2108F
		21	14	43		6	11	20	40	12	
BFS		183	270	445	114	96	128	285	309	129	1959
		21	14	36	7	8	11	19	40	9	
Switch		234	340	475	114	63	128	255	261	144	2014
		23	20	23	7	6	11	17	40	11	
R1		138	336	490	96	63	128	270	363	102	1986
		13	16	26	6	6	11	18	40	9	

TABLE 6
Comparison of two difficult test-functions

	EXP(5) function				Weibull function			
Updating	A, BBA	B, BBA	C, BBA	B, ACB	A, BBA	B, BBA	C, BBA	B, ACB
DFP	3540 204	F	837 133	F	F	F	F	F
BFS	1366 82	349 44	204 30	792 41	767 47	772 102	1673 316	F[a]
Switch	1996 118	F	682 102	828 42	F	1003 155	F	F
R1	1205 75	502 63	1058 147	882 50	1016 63	F	2882 546	F

[a] Shanno and Kettler (1970) report convergence in 444 e.f.e. for their implementation. This difference is apparently due to different safeguards being applied when the function could not be evaluated due to overflow errors.

The following conclusions can be drawn:

(1) on this set of functions, with an accurate line search, the updating formula chosen is not particularly important. The number of iterations on each function is very consistent;
(2) in a comparison based on equivalent function evaluations, the use of a cubic bracketing technique provides little improvement over an efficient function value line search routine, even on functions for which n is small. For larger functions it might be anticipated that the accurate parabolic line search would be more efficient;
(3) the use of a parabolic bracketing technique, or an acceptable point technique gives a much improved efficiency when used in conjunction with the Broyden-Fletcher-Shanno formula. Of the two strategies the parabolic bracketing technique seems slightly more efficient. This is similar to the algorithm listed in Fielding (1971), but the first bracketed point having a reduced function value is accepted.

On the three more difficult functions the behaviour is rather different. The behaviour on EXP(5) and WEIBULL is shown in Table 6.

(1) With these more difficult functions, the identity of performance with the accurate line search routines is lost and a study of the print out reveals that the DFP formula leads to a sequence $\{x_i\}$ that is effectively confined to a subspace. These results are presumably due to an accumulation of round off errors in both the line searches and updating strategies.
(2) These results again indicate that the Broyden-Fletcher-Shanno formula combined with the parabolic bracketing technique is both efficient and reliable.

When these 16 routines were run on the final function RECIP none succeeded in finding the correct answer. Many methods have been tried on this problem, Rosenbrocks (1961) technique reached 16·502 in 560 function evaluations, Hooke and Jeeves (1959) reached 16·5634 in 192 function evaluations whilst the BFS, B, BBA algorithm stopped with 16·76 after 93 function evaluations. The function is included to show that there are situations in which the matrix updating approach should and does fail.

5.3. *A Comparison with Dominant Degree Results*

In Biggs (1971) the principle of modifying the updating formulae to include information obtained about their non-quadratic nature was proposed. The algorithms that result from that proposal are part of Huang's family but have

variable ρ_i and therefore do not form part of the same subset. In his original paper results were published that used the modified updating formula and a dominant degree line search routine that, as in the cubic routine, calculates derivatives at each point; these results have been extended to include the remaining functions given above. The total equivalent to those given in Table 5 is 1138, emphasizing that his combination is a marked improvement over cubic interpolation when function and gradient calculations are always combined.

6. Conclusions

(1) The numerical evidence presented confirms the theoretical result (Dixon, 1971) that, with an accurate line search procedure for determining the step length, all Broyden's (1967) family form a subset of Huang's (1970) family that has identical behaviour on non-quadratic functions. On the two functions where this behaviour failed, due to rounding errors, the Broyden-Fletcher-Shanno algorithm was far more efficient.

(2) All the recent proposals [the Broyden-Fletcher-Shanno formula, total 1105 in Table 4; the Fletcher (1970) Switching policy, total 1403; the rank-one approach, total 1464; and the Biggs (1971) Dominant Degree method, total 1138], are confirmed to be more efficient than the original DFP implementation (total 2108F).

(3) Of these recent proposals, the combination given by Broyden and Fielding was the most efficient for this test series.

(4) The benefit of using the BFS formulae seems to be directly linked with not finding the minimum along the line. This was true for 9 of the 12 functions tested. As most of the theoretical studies of this formula have assumed that exact unidirectional minima were found, further investigation of this point seems necessary.

References

Bard, Y. (1968). On a numerical instability of Davidon-like Methods. *Math. Comput.* **22**, 665–666.

Bard, Y. (1970). Comparison of gradient methods for the solution of non-linear parameter estimation problems *SIAM Numer. Anal.* 7, 157–186.

Biggs, M. C. (1970). 'Computational Experience with Variable Metric Methods for Function Minimisation'. The Hatfield Polytechnic, Numerical Optimisation Centre, T.R.7.

Biggs, M. C. (1971). 'A New Variable Metric Technique Taking Account of Non-quadratic Behaviour of the Objective Function.' The Hatfield Polytechnic, Numerical Optimisation Centre, T.R.17. To appear in *J. Inst. Maths Applics.*

Box, M. J. (1966). A comparison of several current optimisation methods, and the use of transformations in constrained problems. *Comput. J.* **9**, 67–77.

Broyden, C. G. (1967). Quasi-Newton methods and their application to function minimisation. *Math. Comput.* **21**, 368–381.

Broyden, C. G. (1970). The convergence of a class of double-rank minimisation algorithms. *J. Inst. Maths Applics* **6**, 76–90, and 222–231.

Colville, A. R. (1968). 'A Comparative Study of Non-linear Programming Codes'. I.B.M. New York Scientific Centre, T.R. 320–2925.

Davidon, W. C. (1959). 'Variable Metric Method for Minimisation'. A.E.C. R and D Report, ANL-5990.

Dixon, L. C. W. (1969). 'A Comparison of the Relative Efficiency of Several Methods of Finding the Minimum of a Function of one Variable'. The Hatfield Polytechnic, Numerical Optimisation Centre, T.R.2.

Dixon, L. C. W. (1971). 'Variable Metric Algorithms. Necessary and Sufficient Conditions for Identical Behaviour on Non-quadratic Functions'. The Hatfield Polytechnic, Numerical Optimisation Centre, T.R.26.

Fielding, K. (1970). Algorithm 387, Function minimisation and linear search (E4) *Comm. A.C.M.* **13**, No. 8, 509–510.

Fletcher, R., and Powell, M. J. D. (1963). A rapidly convergent descent method for minimisation. *Comput. J.* **6**, 163–168.

Fletcher, R. (1966). Certification of Algorithm 251, Function Minimisation. *Comm. A.C.M.* **9**, No. 9.

Fletcher, R. (1968). A review of methods for unconstrained optimisation. *In* 'Optimisation' (R. Fletcher, ed.), pp. 1–12. Academic Press, London.

Fletcher, R. (1970a). A new approach to variable metric algorithms. *Comput. J.* **13**, 317–322.

Fletcher, R. (1970b). 'An Efficient, Globally Convergent, Algorithm for Unconstrained and Linearly Constrained Optimisation Problems'. Paper presented at the 7th International Mathematical Programming Symposium, The Hague.

Goldfarb, D. (1970). A family of variable metric methods devised by varational means. *Math. Comput.* **24**, 23–26.

Goldstein, A. A., and Price, J. F. (1967). An efficient algorithm for minimisation. *Numer. Math.* **10**, 184–189.

Greenstadt, J. (1970). Variations on variable metric methods. *Math. Comput.* **24**, 1–22.

Huang, H. Y. (1970). Unified approach to quadratically convergent algorithms for function minimisation. *J. Optim. Theory Applns* **5**, 405–423.

Huang, H. Y., and Levy, A. V. (1970). Numerical experiments on quadratically convergent algorithms for function minimisation. *J. Optim. Theory Applns* **6**, 269–282.

Murtagh, B. A., and Sargent, R. W. H. (1970). Computational experience with quadratically convergent minimisation methods. *Comput. J.* **13**, 185–194.

Pearson, J. D. (1969). Variable metric methods of minimisation. *Comput. J.* **12**, 171–178.

Powell, M. J. D. (1962). An iterative method for finding stationary values of functions of several variables. *Comput. J.* **5**, 147–151.

Powell, M. J. D. (1970). Rank one methods for unconstrained optimisation. *In*: 'Integer and Nonlinear Programming' (J. Abadie, ed.), pp. 139–156. North Holland Publishing Co., Amsterdam.

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or the least value of a function. *Comput. J.* **3**, 175–184.

Shanno, D. F. (1970). Conditioning of Quasi-Newton methods for function minimisation. *Math. Comput.* **24**, 647–657.

Shanno, D. F., and Kettler, P. C. (1970). Optimal conditioning of Quasi-Newton methods. *Math. Comput.* **24**, 657–665.

Wolfe, P. (1967). Another variable metric method (unpublished).

Wolfe, P. (1969). Convergence conditions for ascent methods. *SIAM Rev.* **11**, 226–235.

11. Some Aspects of Non-linear Least Squares Calculations

M. R. OSBORNE
Australian National University, Canberra, Australia

Summary

This paper considers a general method for minimizing a sum of squares which has the property that a linear least squares problem is solved at each stage, and which includes the methods of Gauss-Newton and Levenberg, Marquardt, and Morrison as particular special cases. First results relevant to linear least squares are summarized. Then the algorithm is defined and convergence and rate of convergence results derived. Finally, a computational scheme is suggested and test problems and numerical results given.

1. Introduction

The problem of minimizing a sum of squares arises naturally from the problem of determining parameters x_i, $i = 1, 2, \ldots, p$ in the model equation

$$y(t) = F(t, \mathbf{x}) \tag{1}$$

from observations

$$y_i = y(t_i) + \epsilon_i, \qquad (i = 1, 2, \ldots, n), \tag{2}$$

where the ϵ_i (the experimental errors) are independent, normally distributed random variables with mean zero and standard deviation σ. In the case $n > p$ the appropriate maximum likelihood analysis indicates that $\mathbf{x}$ should be estimated by minimizing $\|\mathbf{f}(\mathbf{x})\|$, where

$$f_i(\mathbf{x}) = y_i - F(t_i, \mathbf{x}) \tag{3}$$

and

$$\|\mathbf{f}\|^2 = \sum_{i=1}^{n} f_i(\mathbf{x})^2.$$

This problem will be referred to as the *model problem*, and it is stressed that we have offered a statistical justification for minimizing a sum of squares. It is by no means clear that this is the best strategy if the aim is merely to make the components of a vector valued function small. A comparative study for this purpose of algorithms which use only function and first-derivative information is given in Osborne (1971). Here it is shown that a faster rate of convergence

can frequently be obtained by using the maximum norm, and this norm also offers possibilities for economization of storage.

In this paper we treat algorithms for the model problem which have the property that a linear least squares problem is solved at each step of the iteration. This class of algorithms includes as special cases the Gauss-Newton algorithm and the method of Levenberg, Marquardt, and Morrison. We begin by summarizing results for the linear least squares problem and the related problem of calculating the generalized inverse of a rectangular matrix. Certain approximate methods are defined and algorithms for their implementation suggested. The key tool is the orthogonal reduction of a rectangular matrix to upper triangular form using elementary orthogonal (Householder) transformations given by Golub. The presentation is based on the report of Jennings and Osborne (1970), which also gives program details and numerical results. A general form for an algorithm for the non-linear problem is then presented, together with results on convergence and rate of convergence. Finally, a computational scheme for implementing the non-linear algorithm is suggested, and we give test problems and numerical results.

2. Linear Least Squares Problems

Consider the linear least squares problem

$$\mathbf{y} - A\mathbf{x} = \mathbf{r}, \tag{4}$$

where A is an $n \times p$ matrix, $n \geqslant p$. The vector of minimum norm minimizing $\|\mathbf{r}\|$ is given by

$$\mathbf{x} = A^{+}\mathbf{y} \tag{5}$$

where the $+$ indicates the generalized inverse defined by

$$A^{+} = \lim_{\nu \to 0} (A^T A + \nu^2 I)^{-1} A^T. \tag{6}$$

An alternative form for the generalized inverse can be given in terms of the singular value decomposition of A. Here we write

$$A = UDV^T \tag{7}$$

where D is diagonal and $D_i = [\lambda_i(A^T A)]^{1/2}$, $i = 1, 2, \ldots, p$; V is orthogonal and $A^T A \kappa_i(V) = \lambda_i \kappa_i(V)$, where $\kappa_i(V)$ denotes the ith column of V; U is an $n \times p$ matrix which is column wise orthogonal, i.e., $U^T U = I$, and $AA^T \kappa_i(U) = \lambda_i \kappa_i(U)$. The D_i are frequently called the singular values of A. In this notation we have

$$A^{+} = VD^{+} U^T \tag{8}$$

where

$$\begin{aligned} D_i^{+} &= 1/D_i, \qquad D_i \neq 0, \\ &= 0, \qquad D_i = 0. \end{aligned} \tag{9}$$

Note that if any D_i is small but non-zero, then A^+ will have large elements. A consequence of this is that the calculation of the generalized inverse is likely to be numerically unstable.

This potential instability makes it important to consider approximate methods which can provide control over the size of the solution to the least squares problem. For example, if we are modelling a physical problem then we would expect continuous dependence of the solution on the data and some freedom from the effects of rounding error in our calculations. In this content the *truncation* estimate of A is important. Let

$$A_\nu = UD_\nu V^T \tag{10}$$

where

$$\begin{aligned}(D_\nu)_i &= D_i, \qquad D_i \geqslant \nu,\\ &= 0 \text{ otherwise.}\end{aligned} \tag{11}$$

Let the rank of D_ν be $q \leqslant p$, then A_ν is the best approximation to A in the euclidean norm by matrices of rank q. This result is due to Eckart and Young (see Golub and Kahan, 1965).

More readily available computationally are the *damped least squares* estimates. In this case what is calculated is the solution to the linear least squares problem

$$\begin{bmatrix} A \\ B \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{r}^{(1)} \\ \mathbf{r}^{(2)} \end{bmatrix} \tag{12}$$

so that $\mathbf{x}$ is given by

$$[A^T A + B^T B]\mathbf{x} = A^T \mathbf{y}. \tag{13}$$

Two cases have been considered in the literature,

(a) $B^T B = \nu^2 I$, which corresponds to using a finite value of ν in equation (6), and

(b) $B^T B = \nu^2 I + \nu^2 (A^T A + \nu^2 I)^{-1}$.

This second form is due to Rutishauser (1969), who called it doubly relaxed least squares. Numerical experience with both cases is reported in Jennings and Osborne (1970).

Some idea of what can be gained by using these approximate methods (or perhaps smoothing methods is a better term) can be obtained by looking at explicit solutions. We have

(i) equation (6),

$$\mathbf{x} = \sum_{D_i > 0} \frac{\mathbf{y}^T \kappa_i(U)}{D_i} \kappa_i(V), \tag{14}$$

(ii) truncation,

$$\mathbf{x}_T = \sum_{D_i \geq \nu} \frac{\mathbf{y}^T \kappa_i(U)}{D_i} \kappa_i(V), \tag{15}$$

(iii) damped least squares,

(a) $B^T B = \nu^2 I$,

$$\mathbf{x}_L = \sum_{i=1}^{P} \frac{D_i}{D_i^2 + \nu^2} [\mathbf{y}^T \kappa_i(U)] \kappa_i(V), \tag{16}$$

and

(b) $B^T B = \nu^2 I + \nu^2 (A^T A + \nu^2 I)^{-1}$,

$$\mathbf{x}_R = \sum_{i=1}^{P} \frac{D_i}{D_i^2 + \nu^2 + [\nu^2/(D_i^2 + \nu^2)]} [\mathbf{y}^T \kappa_i(U)] \kappa_i(V). \tag{17}$$

We note that the coefficients in $\mathbf{x}_L$ and $\mathbf{x}_R$ are in the ratio

$$\left(1 + \frac{\nu^2}{(D_i^2 + \nu^2)^2}\right) \Big/ 1$$

so that, in particular, $\|\mathbf{x}_R\| < \|\mathbf{x}_L\|$. Also, the coefficients associated with D_i values small compared with ν are attenuated by a factor ν^2 in $\mathbf{x}_R$ as compared to $\mathbf{x}_L$. In this sense $\mathbf{x}_R$ is closer to $\mathbf{x}_T$ than is $\mathbf{x}_L$. Presumably, its best approximation property makes the truncation estimate attractive. However, its computation involves the calculation of singular values which could well be expensive in the inner loop of an iteration, and for this reason further investigation of doubly relaxed least squares would appear justified.

In computing the solution of the damped least squares problem it is now well appreciated that the formation of the normal matrix $M = A^T A + B^T B$ is a retrograde step, and that it is preferable to use a stable method for factorizing the rectangular matrix into the product of an orthogonal matrix times a rectangular matrix in upper triangular form (Golub and Wilkinson, 1966). In the case $B^T B = \nu^2 I$ it is convenient to build up the orthogonal factorization in two stages

$$\text{(a)} \begin{bmatrix} A \\ \nu I \end{bmatrix} \rightarrow \left[\begin{array}{c|c} Q_1 & \\ \hline & I \end{array}\right] \begin{bmatrix} R_1 \\ 0 \\ \nu I \end{bmatrix}, \qquad \text{(b)} \begin{bmatrix} R_1 \\ 0 \\ \nu I \end{bmatrix} \rightarrow Q_2 \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix}$$

where Q_1 is orthogonal and R_1 upper triangular, and where Q_2 is orthogonal and R upper triangular. Note that the band of zeros introduced in the first step is unaffected by the subsequent calculation if Golub's algorithm is used.

This can lead to considerable savings if n is much larger than p and if calculations for several values of ν are required (this is often the case in the iterative methods to be considered subsequently). This suggestion is due to Golub.

If we now partition Q_2 in the form (where we indicate only the components of immediate interest)

$$Q_2 = \begin{bmatrix} & & \\ & & \\ S^T & & \end{bmatrix}$$

then we see that

$$S^T R = \nu I \tag{18}$$

whence

$$R^T R + \nu^2 (R^T R)^{-1} = R^T R + S^T S. \tag{19}$$

Thus the factorization in the case $B^T B = \nu^2 I + \nu^2 (A^T A + \nu^2 I)^{-1}$ can be obtained by a third step in which we compute the orthogonal factorization of

$$\begin{bmatrix} R \\ S \end{bmatrix}.$$

3. Non-linear Problems

We give now a general form for an algorithm for the non-linear problem in which a linear least squares problem is solved at each stage. We write

$$\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i),\ A_i = \nabla \mathbf{f}(\mathbf{x}_i).$$

Algorithm

(i) Solve the linear least squares problem

$$\begin{bmatrix} \mathbf{f}_i \\ 0 \end{bmatrix} + \begin{bmatrix} A_i \\ B_i \end{bmatrix} \mathbf{h}_i = \begin{bmatrix} \mathbf{r}_i^{(1)} \\ r_i^{(2)} \end{bmatrix} = \mathbf{r}_i. \tag{20}$$

(ii) Determine a step length γ_i (for example by choosing γ_i to minimize $\|\mathbf{f}(\mathbf{x}_i + \gamma \mathbf{h}_i)\|$).

(iii) Set $\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma_i \mathbf{h}_i$, test convergence, repeat if convergence test not satisfied.

Remark

(i) The case $B_i = 0$ corresponds to the Gauss-Newton algorithm, while $B_i = \nu_i^2 I$ gives the method of Levenberg, Marquardt, and Morrison.

(ii) The truncation estimate could be used in the first step of the algorithm. In this case equation (20) becomes

$$\mathbf{f}_i + A_{\nu_i}^{(i)} \mathbf{h}_i = \mathbf{r}_i. \tag{20a}$$

(iii) The number of function evaluations required to calculate γ_i is an important factor in determining the cost of the algorithm. For this reason it is of interest to determine when $\gamma = 1$ can be used. This case is referred to as the full step method.

We now give a lemma which gives optimality conditions in a form appropriate to the above algorithm.

Lemma 1

The following conditions are equivalent:

(i) $\begin{bmatrix} \mathbf{f}_i \\ 0 \end{bmatrix} = \mathbf{r}_i$,

(ii) $\|\mathbf{f}_i\| = \|\mathbf{r}_i\|$, and

(iii) $\mathbf{x}_i$ is a stationary point of $\|\mathbf{f}(\mathbf{x})\|$.

Proof

The solution of the linear least squares problem gives

$$[A_i^T \quad B_i^T]\, \mathbf{r}_i = 0 \tag{21}$$

whence

$$\mathbf{f}_i^T \mathbf{r}_i^{(1)} = \|\mathbf{r}_i\|^2. \tag{22}$$

Thus (ii) implies (i) as a consequence of the Cauchy inequality. If (i) holds then

$$0 = [\mathbf{f}_i^T, 0] \begin{bmatrix} A_i \\ B_i \end{bmatrix} = \tfrac{1}{2} \nabla(\|\mathbf{f}\|^2) \tag{23}$$

so that (i) implies (iii). Now assuming (iii) and taking the scalar product of equation (20) with

$$[\mathbf{f}_i^T, 0]$$

gives

$$\|\mathbf{f}_i\|^2 + \tfrac{1}{2} \nabla(\|\mathbf{f}\|^2) \mathbf{h}_i = \mathbf{r}_i^{(1)T} \mathbf{f}_i = \|\mathbf{r}_i\|^2 \tag{24}$$

so that (iii) implies (ii).

Remark

(i) The corresponding result for equation (20a) is that $\|\mathbf{f}_i\| = \|\mathbf{r}_i\|$ implies $\mathbf{f}_i = \mathbf{r}_i$ and either $\|\mathbf{f}_i\| = 0$ or $\|\nabla(\|\mathbf{f}\|)\| < \nu_i$. In this case the condition for a stationary point is satisfied to within the tolerance used in defining A_{ν_i}.

(ii) As $\|\mathbf{r}_i\| \leqslant \|\mathbf{f}_i\|$, we see that $\mathbf{h}_i$ is downhill for minimizing $\|\mathbf{f}\|$ at $\mathbf{x}_i$ unless $\mathbf{x}_i$ is a stationary point of $\|\mathbf{f}\|$.

We now analyse the convergence of the algorithm, assuming that γ_i is chosen to minimize $\|\mathbf{f}(\mathbf{x}_i + \gamma\mathbf{h}_i)\|$, that the iteration is bounded (so that R is bounded), that $\mathbf{f}(\mathbf{x})$ is sufficiently smooth for us to be able to write

$$\mathbf{f}(\mathbf{x} + \gamma\mathbf{t}) = \mathbf{f}(\mathbf{x}) + \gamma\nabla\mathbf{f}\,\mathbf{t} + \gamma^2\|\mathbf{t}\|^2\,\mathbf{w}(\mathbf{x}, \gamma, \mathbf{t}) \tag{25}$$

and that $\|\mathbf{w}\| \leqslant W$ for points in R.

Lemma 2

Provided $0 \leqslant \gamma \leqslant 1$, then $\|\mathbf{f}(\mathbf{x}_i + \gamma\mathbf{h}_i)\| \leqslant Q_i(\gamma)$, where

$$Q_i(\gamma) = (1 - \gamma)\|\mathbf{f}_i\| + \gamma\|\mathbf{r}_i^{(1)}\| + \gamma^2\|\mathbf{h}_i\|^2\,W. \tag{26}$$

Proof

Substituting $\mathbf{h}_i$ for $\mathbf{t}$ in equation (25), and using equation (20), gives

$$\mathbf{f}(\mathbf{x}_i + \gamma\mathbf{h}_i) = \mathbf{f}_i + \gamma(\mathbf{r}_i^{(1)} - \mathbf{f}_i) + \gamma^2\|\mathbf{h}_i\|^2\,\mathbf{w}_i(\gamma).$$

The result now follows from the triangular inequality.

Lemma 3

Let $A_i^T A_i + B_i^T B_i = M_i$, and

$$\min_{\mathbf{t},\|\mathbf{t}\|=1} (\mathbf{t}^T M_i\,\mathbf{t}) = \delta_i^2,$$

then

$$\|\mathbf{h}_i\| \leqslant \frac{1}{\delta_i}\{\|\mathbf{f}_i\|^2 - \|\mathbf{r}_i\|^2\}^{1/2}. \tag{27}$$

Proof

Taking the scalar product of equation (20) with itself gives

$$\mathbf{h}_i^T M_i\,\mathbf{h}_i = \left\|\begin{bmatrix}\mathbf{f}_i\\ 0\end{bmatrix} - \mathbf{r}_i\right\|^2.$$

The result now follows from the definition of δ_i and equation (22).

Theorem 1

Let B_i be chosen so that the condition $\delta_i \geqslant \delta > 0$ is satisfied, then the sequence $\{\|\mathbf{f}_i\|\}$ is convergent, and the limit points of the sequence $\{\mathbf{x}_i\}$ are stationary points of $\|\mathbf{f}\|$.

Proof

We have necessarily that

$$\|\mathbf{f}_{i+1}\| \leqslant \min_{0\leqslant\gamma\leqslant 1} Q_i(\gamma). \tag{28}$$

If the minimum of the right-hand side is attained for $\gamma < 1$ then $\mathrm{d}Q_i/\mathrm{d}\gamma$ must vanish at the minimum. We have

$$\frac{dQ_i}{\mathrm{d}\gamma} = -(\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|) + 2\gamma\|\mathbf{h}_i\|^2 W$$

whence

$$\gamma = \frac{\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|}{2\|\mathbf{h}_i\|^2 W},$$

and the corresponding value of Q_i is

$$Q_i(\gamma) = \|\mathbf{f}_i\| - \frac{\frac{1}{2}(\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|)^2}{2\|\mathbf{h}_i\|^2 W} = \|\mathbf{f}_i\| - \frac{\gamma}{2}(\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|). \tag{29}$$

However, if $\gamma = 1$ gives the minimum, then

$$\frac{\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|}{2\|\mathbf{h}_i\|^2 W} \geqslant 1.$$

In this case we have

$$Q_i(1) = \|\mathbf{r}_i^{(1)}\| + \|\mathbf{h}_i\|^2 W \leqslant \|\mathbf{f}_i\| - \tfrac{1}{2}(\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|). \tag{30}$$

Let

$$\gamma_i^* = \min\left(1, \frac{\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|}{2\|\mathbf{h}_i\|^2 W}\right), \tag{31}$$

then

$$\|\mathbf{f}_{i+1}\| \leqslant \|\mathbf{f}_i\| - \frac{\gamma_i^*}{2}(\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|). \tag{32}$$

Using lemma 3 gives

$$\gamma_i^* \geqslant \min\left(1, \frac{\delta_i^2}{2W}\frac{1}{(\|\mathbf{f}_i\| + \|\mathbf{r}_i^{(1)}\|)}\right) \geqslant \min\left(1, \frac{\delta_i^2}{2W}\frac{1}{2\|\mathbf{f}_i\|}\right)$$

$$\geqslant \min\left(1, \frac{\delta_i^2}{2W}\frac{1}{2\|\mathbf{f}_0\|}\right) = \mu > 0. \tag{33}$$

Thus the sequence $\{\|\mathbf{f}_i\|\}$ is decreasing and bounded below, and therefore convergent. Further,

$$\|\mathbf{f}_i\| - \|\mathbf{r}_i\| \leqslant \|\mathbf{f}_i\| - \|\mathbf{r}^{(1)}_i\| \leqslant \frac{2}{\mu}(\|\mathbf{f}_i\| - \|\mathbf{f}_{i+1}\|)$$

so that the limit points of the sequence $\{\mathbf{x}_i\}$ are stationary points of $\|\mathbf{f}\|$ by lemma 1.

Corollary 1

If

$$\|\mathbf{f}_i\| \leqslant \frac{\delta_i^2}{4W},$$

then $\gamma^* = 1, j \geqslant i$. If, in addition, the stationary points of $\|\mathbf{f}\|$ are isolated, then the full step method $\mathbf{x}_{j+1} = \mathbf{x}_j + \mathbf{h}_j, j \geqslant i$ is convergent.

Remark

(i) The matrix B_i is available to ensure that the first condition of corollary 1 is satisfied.
(ii) A sufficient condition for an isolated minimum is

$$\|\mathbf{f}_i\| < \min_{1 \leqslant i \leqslant p} \frac{\lambda_i(A^T A)}{2W}.$$

Theorem 2

If the full step method is convergent and $\delta_i \geqslant \delta > 0$, then the rate of convergence is first order.

Proof

In this case the correction at $\mathbf{x}_{i+1}$ is given by

$$\begin{bmatrix} \mathbf{f}_{i+1} \\ 0 \end{bmatrix} + \begin{bmatrix} A_{i+1} \\ B_{i+1} \end{bmatrix} \mathbf{h}_{i+1} = \mathbf{r}_{i+1}$$

where

$$\begin{aligned}\mathbf{f}_{i+1} &= \mathbf{f}_i + A_i\,\mathbf{h}_i + \|\mathbf{h}_i\|^2\,\mathbf{w}_i(1)\\ &= \mathbf{r}_i^{(1)} + \|\mathbf{h}_i\|^2\,\mathbf{w}_i(1).\end{aligned}$$

The normal equations become

$$-M_{i+1}\,\mathbf{h}_{i+1} = [A_{i+1}^T | B_{i+1}^T]\begin{bmatrix}\mathbf{r}_i^{(1)} + \|\mathbf{h}_i\|^2\mathbf{w}_i(1)\\ 0\end{bmatrix}. \tag{34}$$

Equation (21) gives

$$A_i^T\,\mathbf{r}_i^{(1)} + B_i^T\,\mathbf{r}_i^{(2)} = A_i^T\,\mathbf{r}_i^{(1)} + B_i^T\,B_i\,\mathbf{h}_i = 0$$

so that equation (34) can be written

$$\begin{aligned}-M_{i+1}\,\mathbf{h}_{i+1} &= (A_{i+1}^T - A_i^T)\,\mathbf{r}_i^{(1)} - B_i^T\,B_i\,\mathbf{h}_i + \|\mathbf{h}_i\|^2\,A_{i+1}^T\,\mathbf{w}_i(1)\\ &= \|\mathbf{h}_i\|\left\{\frac{\mathrm{d}\overline{A}_i^T}{\mathrm{d}t_i}\,\mathbf{r}_i^{(1)} - B_i^T\,B_i\,\mathbf{t}_i\right\} + \|\mathbf{h}_i\|^2\,A_{i+1}^T\,\mathbf{w}_i(1)\end{aligned} \tag{35}$$

where $\mathbf{h}_i = \|\mathbf{h}_i\|\,\mathbf{t}_i$, $\mathrm{d}/\mathrm{d}t_i$ denotes differentiation in the direction $\mathbf{t}_i$, and the bar denotes that mean values are appropriate. Equation (35) demonstrates the first-order convergence.

Corollary 2

The Gauss-Newton method is divergent from an arbitrarily good starting point if

$$\min_{\mathbf{t}} \|M^{-1}\frac{\mathrm{d}A^T}{\mathrm{d}t}\mathbf{f}\| > 1$$

for all directions $\mathbf{t}$ through the solution (Kowalik and Osborne, 1968).

Example

Let

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix}\eta_1 - x^2\\ \eta_2 - x^3\end{bmatrix}$$

then

$$A(x) = \begin{bmatrix}-2x\\ -3x^2\end{bmatrix}.$$

Writing $\eta_1 = u^2 + \epsilon_1$, $\eta_2 = u^3 + \epsilon_2$ then $x = u$ is a solution provided

$$[2u\ 3u^2]\begin{bmatrix}\epsilon_1\\ \epsilon_2\end{bmatrix} = 0,$$

that is provided $\epsilon_1 = -(3u/2)\epsilon_2$. This solution is a minimum if $6u\epsilon_2 < 8u^2 + 18u^4$. We have $M = A^T A = 4x^2 + 9x^4$,

$$\frac{\mathrm{d}A}{\mathrm{d}x} = -\begin{bmatrix}2\\ 6x\end{bmatrix},$$

and

$$\|M^{-1}\frac{\mathrm{d}A^T}{\mathrm{d}x}\mathbf{f}\|_{x=u} = \frac{1}{4u^2 + 9u^4}|2\epsilon_1 + 6u\epsilon_2| = \frac{|3u\epsilon_2|}{4u^2 + 9u^4}.$$

This can be made arbitrarily large by an appropriate choice of ϵ_2.

Remark

It is of interest to ask if B_i can be chosen to improve the convergence rate. This would require that the term

$$\frac{\mathrm{d}A_i^T}{\mathrm{d}t_i}\mathbf{r}_i^{(1)} - B_i^T B_i \mathbf{t}_i$$

in equation (35) gets small. This requires that

$$\mathbf{t}_i^T\frac{\mathrm{d}A_i^T}{\mathrm{d}t_i}\mathbf{r}_i^{(1)} - \mathbf{t}_i^T B_i^T B_i \mathbf{t}_i$$

also gets small. As $B_i^T B_i$ is positive semidefinite, this requires that

$$\mathbf{t}_i^T\frac{\mathrm{d}A_i^T}{\mathrm{d}t_i}\mathbf{r}_i^{(1)} \geqslant 0$$

at least ultimately. In terms of the model problem we have at the minimum

$$\mathbf{t}^T\frac{\mathrm{d}A^T}{\mathrm{d}t}\mathbf{r}^{(1)} = -\sum_{i=1}^{n}\epsilon_i\sum_{p,q}\frac{\partial^2 F_i}{\partial x_p\,\partial x_q}t_p\,t_q$$

so that the expected value of

$$\mathbf{t}^T\frac{\mathrm{d}A^T}{\mathrm{d}t}\mathbf{r}^{(1)}$$

is

$$E\left\{\mathbf{t}^T \frac{\mathrm{d}A^T}{\mathrm{d}t}\mathbf{r}^{(1)}\right\} = -\sum_{i=1}^{n} E\{\epsilon_i\} \sum_{p,q} \frac{\partial^2 F_i}{\partial x_p \partial x_q} t_p t_q = 0.$$

We conclude that it is unlikely that the order of convergence can be improved by appropriate choice of B_i. However, this probable restriction is a shortcoming of our formalism. For example, an algorithm with super-linear convergence has been given by Goldstein and Price (1967).

Theorem 3

If the full step method is convergent, and if $\mathbf{f}(\mathbf{x}) = 0$ is compatible (so that there exists an $\mathbf{x}^*$ satisfying these equations) then the rate of convergence is second order provided

(i) $\min_{1 \leqslant j \leqslant p} \lambda_j(A_i^T A_i) \geqslant \delta^2 > 0$, and

(ii) $\|B_i^T B_i\| = \max_{\mathbf{t}} \mathbf{t}^T B_i^T B_i \mathbf{t} \leqslant \beta \|\mathbf{h}_i\|$.

Proof

The convergence rate equation (35) can be written

$$-M_{i+1} h_{i+1} = \|\mathbf{h}_i\| \left\{ \frac{\mathrm{d}\overline{A}_i^T}{\mathrm{d}t_i} \mathbf{r}_i^{(1)} - B_i^T B_i \mathbf{t}_i + \|\mathbf{h}_i\| \mathbf{L}_i \right\}$$

so that

$$\|\mathbf{h}_{i+1}\| \leqslant \frac{\|\mathbf{h}_i\|}{\delta^2} \left\{ \left\| \frac{\mathrm{d}\overline{A}_i^T}{\mathrm{d}t_i} \right\| \|\mathbf{r}_i^{(1)}\| + \|B_i^T B_i\| + \|\mathbf{h}_i\| \|\mathbf{L}_i\| \right\}. \tag{36}$$

Taylor expansion gives

$$\mathbf{f}_i + A_i(\mathbf{x}^* - \mathbf{x}_i) = -\|\mathbf{x}^* - \mathbf{x}_i\|^2 \mathbf{w} \tag{37}$$

and comparison of this equation with equation (20) gives

$$\|\mathbf{r}_i^{(1)}\| \leqslant \|\mathbf{r}_i\| \leqslant \{w^2 \|\mathbf{x}^* - \mathbf{x}_i\|^4 + \|B_i^T B_i\| \|\mathbf{x}^* - \mathbf{x}_i\|^2\}. \tag{38}$$

Thus, by condition (ii) of the Theorem,

$$\|\mathbf{r}_i^{(1)}\| = o(\|\mathbf{x}^* - \mathbf{x}_i\|). \tag{39}$$

Also, subtracting the first n equations of (20) from (37) gives

$$A_i(\mathbf{h}_i - (\mathbf{x}^* - \mathbf{x}_i)) = \mathbf{r}_i^{(1)} + \|\mathbf{x}^* - \mathbf{x}_i\|^2 \mathbf{w}$$

whence

$$\|\mathbf{h}_i - (\mathbf{x}^* - \mathbf{x}_i)\| \leqslant \frac{1}{\delta} \{\|\mathbf{r}_i^{(1)}\| + W\|\mathbf{x}^* - \mathbf{x}_i\|^2\} \tag{40}$$

which, by equation (39), implies that

$$\|\mathbf{x}^* - \mathbf{x}_i\| \leqslant K_i \|\mathbf{h}_i\|$$

where $K_i \to 1$ as $\|\mathbf{h}_i\| \to 0$. Thus $\|\mathbf{r}_i^{(1)}\| = o(\|\mathbf{h}_i\|)$, and the inequality (36) becomes

$$\|\mathbf{h}_{i+1}\| \leqslant \frac{\|\mathbf{h}_i\|^2}{\delta^2} \{\beta + \|\mathbf{L}_i\|\} + o(\|\mathbf{h}_i\|^2). \tag{41}$$

This inequality establishes second-order convergence.

Remark

Greenstadt (1967) notes in function minimization with positive definite Hessian H that (i) it is the eigenvectors of H corresponding to small eigenvalues that are likely to be profitable search directions (the others being likely to cause 'hemstitching'), and (ii) it is the components of the predicted search direction $\mathbf{h}$ which corresponds to these eigenvectors which are most sensitive to the effects of perturbation due to rounding error. This should be contrasted with our problem. Here

$$H_{pq} = \sum_{i=1}^{n} \left\{ \frac{\partial f_i}{\partial x_p} \frac{\partial f_i}{\partial x_q} + f_i \frac{\partial^2 f_i}{\partial x_p \partial x_q} \right\}$$

so that in calculating $\mathbf{h}$ we are ignoring the contribution to the Hessian due to the terms

$$f_i \frac{\partial^2 f_i}{\partial x_p \partial x_q}.$$

Thus we are already introducing a significant perturbation unless $\|\mathbf{f}\|$ is small. In terms of our model problem this perturbation ultimately depends on the experimental errors, but the solution to the problem must be very largely independent of them providing the modelling is correct. This observation provides the justification for trying to control and smooth the $\mathbf{h}_i$ even though the cost may be some loss of efficiency, as reflected in the rate of convergence results. However, when the system is compatible (or nearly so) an efficient rate of convergence can be attained if $\|B_i^T B_i\|$ is allowed to get small as the minimum is approached.

4. Numerical Experience

Particular importance attaches to the special case $B_i = \nu_i I$ which gives the algorithm of Levenberg, Marquardt, and Morrison. In this case the following results are readily derived (see for example Kowalik and Osborne, 1968):

(i) $\|\mathbf{h}_i(\nu)\|$ is a monotonic decreasing function of ν.
(ii) The angle between $\mathbf{h}_i(\nu)$ and $-\nabla(\|\mathbf{f}\|^2)(\mathbf{x}_i)$ decreases monotonically as ν increases.
(iii) $\mathbf{x} = \mathbf{h}_i(\nu)$ minimizes $\|A_i\mathbf{x} - \mathbf{f}_i\|^2$ subject to the constraint $\|\mathbf{x}\| = \|\mathbf{h}_i(\nu)\|$.

The first result is of particular relevance for implementing the algorithm (note the way in which it supplements corollary 1). It is not difficult to show, using equation (17), that it holds also for doubly relaxed least squares.

A use of the result (i) is illustrated in the computational scheme outlined below. In this two constants $EXP > 1$ and $DECR < 1$ are kept. If the current value of ν does not give an $\mathbf{h}_i$ such that $\|\mathbf{f}(\mathbf{x}_i + \mathbf{h}_i)\| < \|\mathbf{f}_i\|$, then the current stage is repeated with $\nu = EXP^* \nu$. However, if the first attempt at the current stage gives an $\mathbf{h}_i$ reducing the sum of squares, then ν is reduced by the factor $DECR$ before beginning the next stage. Hopefully, multiple passes through the current stage will not be too frequent, but when these do occur the amount of work resulting is significantly reduced by the device of the two stage orthogonal factorization described in Section 2.

4.1 *Computational Scheme*

We assume that $B_i = \nu_i B(\nu_i)$.

(i) Estimate ν_1.
(ii) Calculate the orthogonal factorization of A_i and set $IC = 0$.
(iii) Complete the factorization of $\begin{bmatrix} A_i \\ B_i \end{bmatrix}$.
(iv) Calculate $\mathbf{h}_i$, $\|\mathbf{f}(\mathbf{x}_i + \mathbf{h}_i)\|$, and set $IC = IC + 1$.
(v) If $\|\mathbf{f}(\mathbf{x}_i + \mathbf{h}_i)\| < \|\mathbf{f}(\mathbf{x}_i)\|$ then

begin $\mathbf{x}_i := \mathbf{x}_i + \mathbf{h}_i$;

if $IC = 1$ then $\nu_i := DECR^* \nu_i$;

$i := i + 1$; go to (vi)

end

else $\nu_i := EXP * \nu_i$;

go to (iii);

(vi) Test convergence. If no convergence, go to (ii).

A FORTRAN program implementing this scheme is given in Jennings and Osborne (1970). In the numerical experiments reported here we took

$$B_i = \nu_i I, \qquad \nu_1 = \left\{ \sum_{i,j} \frac{A_{ij}^2}{np} \right\}^{1/2}, \qquad EXP = 1{\cdot}5, \qquad DECR = 0{\cdot}5.$$

The values of the parameters EXP and $DECR$ were chosen to favour decreasing ν_i as a strategy.

Two problems are considered.

4.2 *Exponential Fitting*

Here the given data values are fitted by the model

$$F(t, \mathbf{x}) = x_1 + x_2 \exp(-x_4 t) + x_3 \exp(-x_5 t).$$

The data values are given in Table 1. They were supplied by Dr A. M. Sargeson of the Research School of Chemistry in the Australian National University. The progress of the algorithm is summarized in Table 2.

TABLE 1
Data for exponential fitting problem

i	t_i	y_i	i	t_i	y_i
1	0	0·844	18	170	0·558
2	10	0·908	19	280	0·538
3	20	0·932	20	190	0·522
4	30	0·936	21	200	0·506
5	40	0·925	22	210	0·490
6	50	0·908	23	220	0·478
7	60	0·881	24	230	0·467
8	70	0·850	25	240	0·457
9	80	0·818	26	250	0·448
10	90	0·784	27	260	0·438
11	100	0·751	28	270	0·431
12	110	0·718	29	280	0·424
13	120	0·685	30	290	0·420
14	130	0·658	31	300	0·414
15	140	0·628	32	310	0·411
16	150	0·603	33	320	0·406
17	160	0·580			

TABLE 2
Summary of numerical results in exponential fitting problem

i	ν	$\|\mathbf{f}\|^2$	x_1	x_2	x_3	x_4	x_5
0	18·586	0·879 E0	0·5	1·5	−1·0	0·01	0·02
1	18·586	0·161 E0	0·4984	1·4993	−1·0007	0·01443	0·02329
2	9·293	0·103 E0	0·4888	1·4967	−1·0034	0·01693	0·02879
3	4·646	0·523 E-1	0·4586	1·4896	−1·0106	0·01462	0·02606
4	2·323	0·941 E-2	0·4140	1·4796	−1·0209	0·01309	0·02634
5	1·116	0·771 E-3	0·3831	1·4799	−1·0211	0·01207	0·02573
6	0·5808	0·987 E-4	0·3715	1·4866	−1·0150	0·01174	0·02511
7	0·2904	0·770 E-4	0·3691	1·4893	−1·0134	0·01167	0·02489
8	0·1452	0·766 E-4	0·3690	1·4913	−1·0152	0·01167	0·02485
9	0·0726	0·754 E-4	0·3691	1·4987	−1·0227	0·01169	0·02479
10	0·0363	0·720 E-4	0·3697	1·5246	−1·0491	0·01179	0·02455
11	0·0182	0·782 E-4					
	0·0272	0·684 E-4	0·3705	1·5616	−1·0867	0·01191	0·02423
12	0·0272	0·650 E-4	0·3711	1·5914	−1·1169	0·01201	0·02401
13	0·0136	0·740 E-4					
	0·0204	0·629 E-4	0·3719	1·6322	−1·1158	0·01213	0·02371
14	0·0204	0·602 E-4	0·3723	1·6641	−1·1906	0·01222	0·02350
15	0·0102	0·679 E-4					
	0·0153	0·591 E-4	0·3729	1·7061	−1·2331	0·01234	0·02324
16	0·0153	0·573 E-4	0·3733	1·7380	−1·2652	0·01242	0·02306
17	0·0077	0·615 E-4					
	0·0115	0·567 E-4	0·3738	1·7779	−1·3056	0·01252	0·02284
18	0·0115	0·557 E-4	0·3742	1·8071	−1·3350	0·01259	0·02270
19	0·0057	0·570 E-4					
	0·0086	0·554 E-4	0·3745	1·8413	−1·3694	0·01267	0·02253
20	0·0086	0·549 E-4	0·3748	1·8648	−1·3932	0·01272	0·02242
21	0·0043	0·551 E-4					
	0·0065	0·548 E-4	0·3750	1·8898	−1·4183	0·01277	0·02231
22	0·0065	0·547 E-4	0·3751	1·9054	−1·4340	0·01281	0·02225
23	0·0032	0·547 E-4	0·3753	1·9250	−1·4538	0·01285	0·02217
24	0·0016	0·547 E-4	0·3754	1·9343	−1·4631	0·01286	0·02213
25	0·0008	0·546 E-4	0·3754	1·9358	−1·4646	0·01287	0·02212
26	0·0004	0·546 E-4	0·3754	1·9358	−1·4647	0·01287	0.02212
27	0·0002	0·546 E-4	0·3754	1·9358	−1·4647	0·01287	0·02212

4.3 *Fitting Gaussians plus an Exponential Background*

In this case the model has the form

$$F(t, \mathbf{x}) = x_1 \exp(-x_5 t) + x_2 \exp[-x_6(t - x_9)^2] + x_3 \exp[-x_7(t - x_{10})^2] + x_4 \exp[-x_8(t - x_{11})^2].$$

The data values are listed in Table 3, and the progress of the iteration is summarized in Table 4. The data was supplied by Dr W. J. Caelli of the

TABLE 3
Data for fitting gaussians plus exponential background

i	t_i	y_i	i	t_i	y_i
1	0·0	1·366	34	3·3	0·375
2	0·1	1·191	35	3·4	0·372
3	0·2	1·112	36	3·5	0·391
4	0·3	1·013	37	3·6	0·396
5	0·4	0·991	38	3·7	0·405
6	0·5	0·885	39	3·8	0·428
7	0·6	0·831	40	3·9	0·429
8	0·7	0·847	41	4·0	0·523
9	0·8	0·786	42	4·1	0·562
10	0·9	0·725	43	4·2	0·607
11	1·0	0·746	44	4·3	0·653
12	1·1	0·679	45	4·4	0·672
13	1·2	0·608	46	4·5	0·708
14	1·3	0·655	47	4·6	0·633
15	1·4	0·616	48	4·7	0·668
16	1·5	0·606	49	4·8	0·645
17	1·6	0·602	50	4·9	0·632
18	1·7	0·626	51	5·0	0·591
19	1·8	0·651	52	5·1	0·559
20	1·9	0·724	53	5·2	0·597
21	2·0	0·649	54	5·3	0·625
22	2·1	0·649	55	5·4	0·739
23	2·2	0·694	56	5·5	0·710
24	2·3	0·644	57	5·6	0·729
25	2·4	0·624	58	5·7	0·720
26	2·5	0·661	59	5·8	0·636
27	2·6	0·612	60	5·9	0·581
28	2·7	0·558	61	6·0	0·428
29	2·8	0·533	62	6·1	0·292
30	2·9	0·495	63	6·2	0·162
31	3·0	0·500	64	6·3	0·098
32	3·1	0·423	65	6·4	0·054
33	3·2	0·395			

Research School of Physical Sciences in the Australian National University.

It will be seen that the calculations are very satisfactory. The values chosen for *EXP* and *DECR* have proved successful also in other cases, and the exponential fitting problem has proved to be one in which this comparatively simple strategy for adjusting ν was least successful. It is possible that the convergence analysis might give a useful strategy here. In constructing the dominating function $Q_i(\gamma)$ all that is actually required is the quantity

$$W_i = \max_{0 \leq \gamma \leq 1} \|\mathbf{w}_i(\gamma)\|.$$

TABLE 4
Summary of numerical results for fitting gaussians plus exponential background

i	ν	$\|\mathbf{f}\|^2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
0	0·3047	2·094	1·3	0·65	0·65	0·7	0·6	3·0	5·0	7·0	2·0	4·5	5·5
1	0·3407	0·7010	1·1140	0·08846	0·5711	0·7017	0·2778	3·1329	4·6853	6·7556	2·1401	4·5281	5·5507
2	0·1704	0·2984	1·1095	0·1124	0·4842	0·6124	0·3216	3·2142	3·9956	6·2774	2·6650	4·5478	5·5731
3	0·0852	0·1794	1·1410	0·1878	0·5028	0·6181	0·3725	2·6710	2·3188	5·3592	2·4151	4·5400	5·6054
4	0·0426	1·561											
	0·0639	0·3172											
	0·0958	0·1359	1·1949	0·2877	0·5538	0·5996	0·4654	2·0006	1·3213	5·0849	2·4858	4·5573	5·6565
5	0·0958	0·0874	1·2654	0·3686	0·6115	0·5623	0·5990	1·2003	1·1355	5·0755	2·4163	4·5754	5·6829
6	0·0479	0·0512	1·3046	0·4270	0·6382	0·5906	0·7220	0·9211	1·2792	4·9222	2·4048	4·5714	5·6819
7	0·0240	0·0402	1·3098	0·4317	0·6339	0·5993	0·7229	0·9062	1·3597	4·8302	2·3996	4·5694	5·6760
8	0·0120	0·0402	1·3100	0·4314	0·6336	0·5993	0·7539	0·9057	1·3650	4·8252	2·3988	4·5689	5·6754
9	0·0060	0·0402	1·3100	0·4315	0·6336	0·5993	0·7539	0·9056	1·3650	4·8248	2·3988	4·5689	5·6754
10	0·0030	0·0402	1·3100	0·4315	0·6336	0·5993	0·7539	0·9056	1·3651	4·8248	2·3988	4·5689	5·6754
11	0·0015	0·0402	1·3100	0·4315	0·6336	0·5993	0·7539	0·9056	1·3651	4·8248	2·3988	4·5689	5·6754

Let $\mathbf{f}_{i+1}^* = \mathbf{f}(\mathbf{x}_i + \mathbf{h}_i)$. Frequently, we can expect $\|\mathbf{w}_i(1)\|$ to be a reasonable estimate of W_i so that we can estimate the value of γ for which $\mathrm{d}Q_i/\mathrm{d}\gamma = 0$ by

$$\bar{\gamma} = \frac{\|\mathbf{f}_i\| - \|\mathbf{r}_i^{(1)}\|}{2\|\mathbf{f}_{i+1}^* - \mathbf{r}_i^{(1)}\|}.$$

Presumably, we would not want to reduce ν unless $\bar{\gamma} \geqslant 1$. Also, note that a strategy of using a full step provided $\|\mathbf{f}\|$ is decreasing is probably too simple a strategy to guarantee convergence to a point where $\|\mathbf{f}\| = \|\mathbf{r}\|$ (a related problem is discussed in Fletcher, 1970). However, a failure would be indicated by the sequence

$$\left\{ \frac{\|\mathbf{f}_i\| - \|\mathbf{r}_i\|}{\|\mathbf{f}_i\| - \|\mathbf{f}_{i+1}\|} \right\}$$

being unbounded.

References

Fletcher, R. (1970). An efficient, globally convergent algorithm for unconstrained and linearly constrained optimization problems. Paper presented at the 7th Mathematical Programming Symposium, The Hague, Netherlands.

Goldstein, A. A., and Price, J. F. (1967). An efficient algorithm for minimization. *Num. Math.* **10**, 184–189.

Golub, G., and Kahan, W. (1965). Calculating the singular values and pseudo inverses of matrices. *SIAM J. Numer. Anal.* **2**, 205–224.

Golub, G., and Wilkinson, J. H. (1966). Note on the iterative refinement of least squares solutions. *Numer. Math.* **9**, 139–148.

Greenstadt, J. (1967). On the relative efficiencies of gradient methods. *Math. Comput.* **21**, 360–367.

Jennings, L. S., and Osborne, M. R. (1970). 'Applications of Orthogonal Matrix Transformations to the Solution of Systems of Linear and Nonlinear Equations'. Technical Report 37, Computer Centre, Australian National University.

Kowalik, J., and Osborne, M. R. (1968). 'Methods for Unconstrained Optimization Problems'. Elsevier, New York.

Osborne, M. R. (1971). An algorithm for discrete, nonlinear, best approximation problems. *In*: 'Proceedings of the Conference on Numerical Methods in Approximation Theory'. Oberwolfach.

Rutishauser, H. (1968). The least square problem. *Linear Algeb. Appls* **1**, 479–488.