Mathematical Programming 14 (1978) 149–160.
North-Holland Publishing Company

MATRIX CONDITIONING AND NONLINEAR OPTIMIZATION

D.F. SHANNO

The University of Arizona, Tucson, AZ, U.S.A.

Kang-Hoh PHUA

Nanyang University, Singapore

Received 25 November 1975
Revised manuscript received 2 August 1977

In a series of recent papers, Oren, Oren and Luenberger, Oren and Spedicato, and Spedicato have developed the self-scaling variable metric algorithms. These algorithms alter Broyden's single parameter family of approximations to the inverse Hessian to a double parameter family. Conditions are given on the new parameter to minimize a bound on the condition number of the approximated inverse Hessian while insuring improved step-wise convergence.

Davidon has devised an update which also minimizes the bound on the condition number while remaining in the Broyden single parameter family.

This paper derives initial scalings for the approximate inverse Hessian which makes members of the Broyden class self-scaling. The Davidon, BFGS, and Oren–Spedicato updates are tested for computational efficiency and stability on numerous test functions, with the results indicating strong superiority computationally for the Davidon and BFGS update over the self-scaling update, except on a special class of functions, the homogeneous functions.

Key words: Unconstrained optimization, Variable metric methods, SSVM methods, Quasi-Newton methods.

1. Introduction

In a series of recent papers, Oren [9, 10], Oren and Luenberger [11], Oren and Spedicato [12], and Spedicato [15], have developed the self-scaling variable metric algorithms (SSVM's) for minimizing an unconstrained nonlinear function $f(x)$ of a vector variable x. These algorithms are modifications of the well-known variable metric (quasi-Newton) algorithms which have been the subject of much recent study (see, for example, Dennis and Moré [4]).

Briefly, these algorithms begin with an estimate x_0 to the minimizer $\hat{x}$ and an estimate H_0 to the inverse of the Hessian matrix of $f(x)$. A sequence of points x_i is then defined by

$$x_{i+1} = x_i - \alpha_i H_i g_i, \quad g_i = \nabla f(x_i), \ \alpha_i \text{ a scalar}, \tag{1}$$

and

$$H_{i+1} = [H_i - (H_i y_i y_i' H_i / y_i' H_i y_i) + \theta_i v_i v_i'] \gamma_i + (p_i p_i' / p_i' y_i), \tag{2}$$

with

$$\begin{aligned} &p_i = x_{i+1} - x_i, \qquad y_i = g_{i+1} - g_i, \\ &v_i = (y_i' H_i y_i)^{1/2} [(p_i / p_i' y_i) - (H_i y_i / y_i' H_i y_i)], \end{aligned} \tag{3}$$

and α_i, θ_i, and γ_i are appropriately chosen scalars. For the remainder of this paper, we will omit the subscript i, replace the subscript $i+1$ by the superscript *, and use the notation

$$a = y'Hy, \qquad b = p'y, \qquad c = p'H^{-1}p = \alpha^2 g'Hg. \tag{4}$$

We first note that the sequence of updates (2) reduces to the Broyden single parameter class [1], in the form introduced by Fletcher [5], by setting $\gamma_i = 1$. Oren and Luenberger motivate the introduction of the second parameter γ by considering scaling the object function by multiplication by a scalar. We note that under the scaling

$$\hat{f}(x) = cf(x), \qquad \hat{g}(x) = cg(x) \quad \text{and} \quad \hat{J}(x) = cJ(x),$$

where $J(x)$ is the Hessian matrix of f. Thus the Newton sequence for f is defined by

$$x^* = x - J^{-1}g, \tag{5}$$

and for $\hat{f}$ by

$$x^* = x - \hat{J}^{-1}\hat{g} = x - c^{-1}J^{-1}(cg) = x - J^{-1}g. \tag{5a}$$

Thus the sequences are invariant under this scaling, while the sequence defined by eqs. (1)–(3) is not, unless either γ_i is appropriately chosen or H_0 is chosen to be a suitable function of either the initial gradient or the initial inverse Hessian matrix.

Having thus motivated γ, Oren and Luenberger proceed to show that if $f(x)$ is of the form

$$f(x) = \tfrac{1}{2}(x - \hat{x})'A(x - \hat{x}) + f(\hat{x}), \tag{6}$$

proper choice of γ insures at each step a reduction in the condition number of

$$R = A^{1/2}HA^{1/2}. \tag{7}$$

Thus use this result to argue that proper choice of γ can be used to improve the step-wise rate of convergence over members of the Broyden class, and show examples where for $\gamma = 1$, $\theta = 0$ can actually increase the condition number of R^* over that of R. They further show that there exist a continuum of γ's which not only improve the rate of convergence, but make the sequence (2) self-scaling. Subsequent papers then study principally the choice of γ, and include numerical results purporting to show the superior computational efficiency of the new algorithm.

In particular, Oren and Spedicato [12] consider the problem of minimizing the condition number of the matrix $H^{-1}H^*$, and from this derive the relationship

$$\theta = (b(c - b\gamma)/\gamma(ac - b^2)). \tag{7a}$$

They then introduce Fletcher's concept of duality to derive the self-dual update

defined by (2) with

$$\theta = (1 + (ac/b^2)^{1/2})^{-1}, \quad \gamma = (c/a)^{1/2}. \tag{8}$$

Davidon [3] similarly considered minimizing the condition number of $H^{-1}H^*$ for the Broyden class, and arrived at the update defined by

$$\theta = b(c-b)/(ac-b^2) \quad \text{when} \quad b \leq 2ac/(a+c), \tag{9}$$

and

$$\theta = b/(b-a) \quad \text{when} \quad b > 2ac/(a+c), \tag{9a}$$

where (9a) is the well-known symmetric rank one update. It should be noted here that Davidon's full new algorithm also modified the vectors y and p, using projections. In this paper, we are concerned solely with the choice of update parameters, and use no projections. The computational effects of using projections have been studied by Shanno and Phua [16].

The principal drawback to the SSVM updates is that for a quadratic object function, the sequence (2) fails to converge to the inverse Hessian matrix. McCormick and Ritter [8] have shown that this convergence is desirable on a non-quadratic problem, and it is a property not to be cast aside lightly.

Section 2 deals with a choice of H_0 which leaves the SSVM sequence unchanged, but which makes the Broyden sequence self-scaling. The implications of this choice on matrix conditioning and computational stability will be discussed.

Section 3 documents a special class of functions where the SSVM algorithms are superior, under certain conditions, to the Broyden class algorithms.

Section 4 tests the scaled and unscaled BFGS and Davidon algorithms and the scale invariant SSVM defined by (2) and (8), and shows marked computational superiority of the scaled BFGS and Davidon algorithms over all problems tested except the special class discussed in Section 2, where the BFGS algorithm, defined by $\gamma = 1$ and $\theta = 1$, is the algorithm introduced by Broyden [2], Fletcher [5], Goldfarb [6], and Shanno [13].

2. Scaling the initial approximation H_0

As noted in Section 1, invariance of the sequence (1) under scaling of the object function by multiplication by a constant depends either upon appropriately choosing γ, or scaling of H_0. In this section, we consider two possible initial scalings of H_0, and show that in addition to providing the desired invariance, they also greatly enhance the numerical stability of members of the Broyden class of updates.

Spedicato [15] has considered the problem of initializing H_0, suggesting that H_0 be set to a diagonal matrix whose elements are the reciprocals of the true diagonal elements of the Hessian evaluated at the initial approximation x_0. There

is a major drawback to this, namely that it is often expensive to obtain these estimates. A second objection is that computational efficiency is not always improved.

We here consider much simpler initial scalings which require no additional information about the object function than that routinely required by variable metric algorithms. They thus overcome the major objection to Spedicato's scaling. Further, they have the property that they leave the SSVM algorithms unchanged, while making the Broyden class algorithms self-scaling. Finally, the computational results documented in Section 5 show that while they sometimes improve, sometimes hurt computational efficiency on small problems (problems with few variables), they uniformly improve performance of both the BFGS and Davidon algorithms on large problems, and the amount of improvement increases monotonically as the number of variables increases.

The salient point in both of these scalings is that $H_0 = I$ may be used initially to determine x_1, where α_0 is chosen according to some steplength or linear search criterion to assure sufficient reduction in the function f. Once x_1 has been chosen, but before H_1 is calculated, we now scale H_0 by

$$\hat{H}_0 = \alpha_0 H_0, \tag{10}$$

then compute H_1 using $\hat{H}_0$ rather than H_0. Assuming only that the criterion for choosing α_0 is sensitive to multiplication of the object function by a constant, an assumption satisfied by virtually all commonly used step determination methods, the sequence of approximations (2) thus becomes self-scaling without use of the parameter γ. In fact, the parameter γ derives from a composite scaling of the form

$$\hat{H} = \alpha H \tag{11a}$$

and

$$H^* = \hat{H} - (\hat{H}yy'\hat{H}/y'\hat{H}y) + \theta\hat{v}\hat{v}' + (pp'/p'y),$$
$$\hat{v} = y'\hat{H}y[(p/p'y) - (\hat{H}y/y'\hat{H}y)].$$

Substitution of (11a) into (11b) yields

$$H^* = [H - (Hyy'H/y'Hy) + \theta vv']\alpha + (pp'/p'y). \tag{12}$$

Thus a motivation for the SSVM updates is that if H is a good approximation to the true inverse Hessian, α will be 1. If $\alpha \neq 1$, H is first scaled by multiplication by α, then the Broyden update applied. We note that while γ is seldom chosen to be exactly equal to α, it is chosen to be α-sensitive, and increases or decreases directly as α is greater or less than 1. Then the above motivation is a valid motivation. The more sophisticated choice of γ, however, used by Oren and Spedicato has a distinct advantage over using α, as will be explained in Section 3.

The difference between the SSVM's and the scaled Broyden class updates of this paper are that the SSVM's scale H by γ at each step, while we propose

scaling only H_0. After the initial iteration, the approximate Hessian is never rescaled.

In practice, the above scaling has a major advantage beyond making the sequence (2) self-scaling with $\gamma = 1$. Spedicato [15] tested a version of the BFGS and the Oren–Spedicato Switch 2 SSVM on both the Rosenbrock and Powell functions, generalized to many dimensions. For example, the generalized Rosenbrock function is

$$f(x) = \sum_{i=1}^{1/2n} 100(X_{2i} - X_{2i-1}^2)^2 + (1 - X_{2i-1})^2.$$

He shows that theoretically, as n increases, the number of iterations and function evaluations must remain constant on such functions with perfect arithmetic. His computational results, however, show that both iterations and function evaluations increase markedly as n increases, with or without his initial scaling.

Our computer results, however, show that without the scaling (10), both the BFGS and Davidon updates greatly increase the number of iterations and function evaluations as n increases, but with the scaling (10), the theoretical results of no increased computational effort are exactly borne out. However, the Switch 2 SSVM algorithm is not improved by this scaling, nor are any of the SSVM's, and computational work does grow significantly as n grows.

To examine the reasons for this phenomenon, we first note that for the Switch 2 SSVM defined by (2) and (8), if at any step we use the composite update

$$\hat{H} = \rho H, \quad \rho \text{ an arbitrary scalar} \tag{13a}$$

and

$$H^* = [\hat{H} - (\hat{H}yy'\hat{H}/y'\hat{H}y) + \theta\hat{v}\hat{v}']\gamma + (pp'/p'y), \tag{13b}$$

substituting yields $y'\hat{H}y = \rho y'Hy$, and $p'\hat{H}^{-1}p = \rho^{-1}p'H^{-1}p$, and hence $\hat{\theta}$ and $\hat{\gamma}$ defined by (8) satisfy

$$\hat{\theta} = \{1 + [(\rho a)(\rho^{-1}c/b^2)]^{1/2}\}^{-1} = \theta \tag{14a}$$

and

$$\hat{\gamma} = (\rho^{-1}c/\rho a)^{1/2} = \rho^{-1}\gamma. \tag{14b}$$

But as $\hat{v} = \rho^{1/2}v$, substitution in (13b) yields

$$H^* = [H - (Hyy'H/y'Hy) + \theta vv']\gamma + (pp'/p'y). \tag{15}$$

Thus the Switch 2 algorithm, and indeed all of the SSVM's considered by Oren and Luenberger, are invariant under transformations of this type. Thus no computational improvement can be expected.

For the Broyden class, however, improvement on large problems is due entirely to improved numerical stability. If the initial matrix is unscaled, or scaled without knowledge of the actual magnitude of the elements of the true inverse Hessian, severe loss of accuracy due to round-off can occur. To see this,

note that for the Broyden class, the update formula

$$H^* = H - (Hyy'H/y'Hy) + \theta vv' + (pp'/p'y) \tag{16}$$

can be grouped into the two components

$$H_1 = H - (Hyy'H/y'Hy) + \theta vv' \tag{16a}$$

and

$$H_2 = (pp'/p'y). \tag{16b}$$

All terms in H_1 are equally magnified when H is multiplied by a constant, but H_2 is invariant. An arbitrary initial scaling can cause large loss of accuracy in terms emanating from H_2 if the scale of H is too large, or in terms from H_1 if the scale of H is too small. Both losses are significant, as loss of accuracy in H_2 prevents convergence to an approximate Hessian, while loss in H_1 drastically alters the search directions. As nothing can be determined about a good approximate scaling until one iteration has been completed, it then becomes important to scale at the end of the first iteration.

We have so far considered only the initial scaling $\hat{H}_0 = \alpha_0 H_0$. While this is a natural scaling, another initial scaling in also attractive for the BFGS update. Recalling the Oren–Spedicato formula (7a) for the relationship between θ and γ which will optimally condition $H^{-1}H^*$, we use the fact that for the BFGS $\theta = 1$ to derive the optimal γ as

$$\gamma = b/a, \tag{17}$$

yielding the initial scaling

$$\hat{H}_0 = (b/a)H_0. \tag{18}$$

The computational evidence of Section 5 shows both scalings are remarkably similar on small problems, but limited computational experience on a large problem where effort does increase with increase in size indicates the update (18) may be preferable.

As an example, we will examine the results of updating H_0 for the Rosenbrock function, both for scaled and unscaled H_0. The MINI02 algorithm [14] initially sets H_0 to $H_0 = 1.8219\,\mathrm{I}$. The step length found for the first search is $\alpha_0 = 0.00082253$, and the Oren–Spedicato parameter $\gamma = 0.00044281$. The matrix H_1 is

$$H_1 = \begin{bmatrix} 0.27830 & -0.65550 \\ -0.65550 & 1.5439 \end{bmatrix},$$

while

$$H_2 = \begin{bmatrix} 0.00069168 & 0.00028232 \\ 0.00028232 & 0.00011523 \end{bmatrix}.$$

Thus the unscaled BFGS update is

$$H^* = \begin{bmatrix} 0.27899 & -0.65522 \\ -0.65522 & 1.5440 \end{bmatrix},$$

while the scaled BFGS update is

$$H^* = \begin{bmatrix} 0.00081482 & -0.0000079435 \\ -0.0000079435 & 0.00079891 \end{bmatrix}.$$

The fact that α is of the order of magnitude 10^{-3} while the arbitrary H_0 is of order 1 predicts the demonstrated result that the matrix H_2 is significant in only the fourth significant figure unscaled, but is significant in every significant figure scaled.

This problem is solely a truncation, rather than a stability problem, as the example shows, and cannot be improved by using an LDL' factorization of H^{-1}. Indeed, the same problem, for $n = 2$ and 20 variables, was run using an LDL' factorization, both scaled and unscaled, with virtually identical results in each case to those reported in Section 4.

As a final note in this section, the Davidon update can also be scaled by either (10) or (18), but if (18) is chosen, care must be used. In the case that $b < 2ac/(a + c)$, the initial scaling (18) yields the BFGS update on the scaled matrix. However, if $b > 2ac/(a + c)$, the Davidon update is the symmetric rank one with a zero denominator. As there is no intuitive reason to use (18) rather than (10) with the Davidon update, this latter problem would appear to suggest (10). However, on the Mancino problem documented in Section 5, the scaling (18) performed marginally better. Thus further investigation of the use of (18) with the Davidon update on large problems may be indicated.

3. Homogeneous functions and the SSVM algorithms

The subsequent section will demonstrate that the SSVM algorithms perform poorly compared to initially scaled BFGS and Davidon algorithms. Before proceeding, however, we should note that there is a class of functions on which the Switch 2 SSVM algorithm is markedly superior to the Broyden class algorithm if exact searches are not used to determine α at each step. These are the homogeneous functions.

Jacobson and Oksman [7] define a homogeneous function as one which can be written as

$$f(x) = \tau^{-1}(x - \hat{x})g(x) + f(\hat{x}), \tag{19}$$

where $g(x) = \nabla f(x)$, $\hat{x}$ is the minimizer, and τ is the degree of homogeneity. If $J(x)$ is the Hessian of f, they then show by differentiating (19) that

$$\hat{x} = x - (\tau - 1)J^{-1}(x)g(x). \tag{20}$$

Thus the usual Newton step $J^{-1}(x)g(x)$ must be multiplied by $(\tau - 1)$ to get to the minimum. A typical homogeneous function of degree four is

$$f(x) = (x'Ax)^2. \tag{21}$$

Spedicato [15] has shown that the SSVM method is equivalent to using a formula from the Huang two parameter family in which updates satisfy

$$H^*y = \beta p. \tag{22}$$

The corresponding value of β is dependent upon the choice of γ.

Examination of (20) shows that, since $g(\hat{x}) = 0$,

$$(\tau - 1)^{-1}(x - \hat{x}) = J(x)(g(x) - g(\hat{x})), \tag{23}$$

so a similar equation, with $\beta = (\tau - 1)^{-1}$ is satisfied by homogeneous equations, not at all points, but at the minimum. Thus for proper choice of β, SSVM's should predict better step-sizes than Broyden class updates for homogeneous functions.

We have tested Oren's test problem of the form (21), and this is certainly the case for the step size algorithm we use. In this algorithm, a point is accepted if $p'y > 0$ and $f^* < f + 0.0001p'g$, where $p'g < 0$. As $f(x)$ as defined by (26) is strictly convex, $p'y > 0$ always holds. As the rate of descent is rapid, a step size too small always satisfies $f^* < f + 0.0001p'g$. Thus small steps are predicted and accepted, and convergence is very slow. The switch 2 choice of γ, however, corresponds to a good choice of β, and the SSVM automatically increases the step size, producing vastly improved computational performance. Note that if exact searches are used, as were used by Spedicato [15], this advantage largely disappears.

As Jacobson and Oksman present a method with guaranteed $n+2$ step convergence on homogeneous problems, a careful analysis of the SSVM's on this type of function must include a comparison with the Jacobson–Oksman algorithm.

4. Computational results

Five algorithms, a BFGS algorithm without initial scaling (BFGS), with the initial scaling (18), (BFGS18), Davidon unscaled (OCON), Davidon with scaling (10), (OCON10), and the Oren–Spedicato Switch 2 SSVM (OSS2) were tested using a variant of the MINI02 step length algorithm (see Shanno and Phua [14]). The algorithm was exactly as documented, except that for all algorithms except BFGS, the portion of the code scaling the search vector during the first n iterations was removed. The test functions included the extended Rosenbrock and extended Powell functions documented by Spedicato [15], defined respectively by

$$f(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \tag{24}$$

and

$$f(x) = \sum_{i=1}^{n/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4. \tag{25}$$

Both of these share the property that as increases, the number of function evaluations should not. Also included were the Wood and Weibull functions, documented in Shanno and Phua [14], Oren's power function

$$f(x) = (x'Ax)^2, \qquad A = [a_{ij}], \quad a_{ij} = i\delta_{ij}, \tag{26}$$

and the Mancino function documented by Spedicato and defined by

$$F(x) = \sum_{i=1}^{n} f_i^2, \tag{27}$$

and

$$f_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} [(x_j^2 + i/j)^{1/2}(\sin^{\alpha} \log(x_j^2 + i/j)^{1/2} + \cos^{\alpha} \log(x_j^2 + i/j)^{1/2})] + \beta n x_i + (i - n/2)^{\gamma},$$

with $\alpha = 5$, $\beta = 14$, $\gamma = 3$, $x_0 = (af_1(0), \ldots, af_n(0))$, and

$$a = \frac{-n\beta}{\beta^2 n^2 - (\alpha + 1)^2 (n-1)^2}.$$

The functions defined by (26) and (27) are the two of general size where the number of function evaluations increases with n. (26) is homogeneous, as mentioned in Section 3. The Mancino function is the sole general function tested for relatively large n that gives some indication of the relative efficiencies of the two initial scalings.

The Rosenbrock function was run for $n = 2$ and $n = 20$. In fact, runs with large values of n (up to 80) were tried, with the scaled versions remaining invariant, and all others increasing, although the increase in OSS2 was minimal above 20. We here document only $n = 2$ and $n = 20$.

The Powell function was run with $n = 4$ and $n = 36$, the power function with $n = 20$, and the Mancino function with $n = 10$, 20 and 30. The Wood and Weibull functions have $n = 4$ and $n = 3$, respectively. The results, for the initial estimate x_0 in parentheses, are summiarized in Table 1. ITER denotes iterations, IFUN function and gradient evaluations.

For all problems, the step bound was limited to $|\Delta x_i| \leq 3$, with the exception of the Weibull function, where $\Delta x_1 = 300$, $\Delta x_2 = 3$ and $\Delta x_3 = 30$. The runs were done on a DEC10 computer, using double precision, and convergence achieved when all gradient elements were less than 10^{-5}.

As Table 1 clearly shows, with the exception of the power function, the OSS2 algorithm is clearly not competitive with either OCON10 or BFGS18. Also, while there is little to choose between them, BFGS 18 outperforms OCON10 on all but the Powell function. Also, as previously noted, BEGS18 and OCON 10 dominate BFGS and OCON as problem size increases.

The comparisons between the four methods (excluding OSS2) on small problems, however, may slightly favor the unscaled OCON. Exact reasons for the erratic behavior of scaled and unscaled methods on small problems to date

Table 1

	BFGS		BFGS18		OCON		OCON10		OSS2	
	ITER	IFUN	ITER	IFUN	ITER	IFUN	ITER	IFUN	ITER	IFUN
Rosenbrock 2										
(−1.2, 1.0)	35	42	35	42	32	42	40	45	47	58
(2, −2)	16	18	38	47	15	17	38	47	37	44
(−3.635, 5.621)	46	55	50	62	55	63	61	71	70	82
(6.39, −0.221)	54	64	29	34	53	64	29	35	21	23
(1.489, −2.547)	21	24	29	35	23	26	32	40	23	28
Rosenbrock 20										
(−1.2, 1.0)	129	173	35	42	132	158	39	45	87	100
(2, −2)	34	42	38	47	29	37	38	47	65	77
(−3.635, 5.621)	155	175	50	62	155	168	59	72	160	170
(6.39, −0.221)	122	137	29	34	158	176	29	35	22	24
(1.489, −2.547)	85	100	29	35	56	67	32	40	49	57
Wood										
(−3, −1, −3, −1)	79	98	38	42	83	100	44	48	119	133
(−3, 1, −3, 1)	79	102	95	112	81	96	101	119	167	199
(−1.2, 1, −1.2, 1)	69	89	85	111	71	86	97	122	130	152
(−1.2, 1, 1.2, 1)	36	48	44	51	37	47	47	56	54	62
Weibull										
(5, 0.15, 2.5)	37	54	49	57	33	40	53	60	50	61
(250, 0.3, 5)	39	56	57	77	44	59	69	90	87	115
(100, 3, 12.5)	42	49	45	57	39	48	51	65	61	68
Power 20										
(1, ... , 1)	96	100	232	234	69	80	212	214	26	28
Powell 4										
(−3, −1, 0, 1)	37	39	59	61	32	36	41	43	41	43
Powell 36										
(−3, −1, 0, 1)	87	89	59	61	51	63	41	43	76	86
Mancino										
$n = 10$	18	28	5	8	15	29	5	8	5	8
$n = 20$	35	47	8	10	21	34	9	11	8	10
$n = 30$	52	69	8	10	23	37	13	15	10	12

elude us. The MINI02 algorithm sets $H_0 = 100 * S/G$, where $S = \|\Delta x_i\|$, the Euclidean norm of the vector of step bounds, and $G = \|g_0\|$. Thus a very small α is always chosen, and a great deal of noise is introduced. Our experience seems to indicate that superiority here is almost totally due to round-off considerations, and defies careful analysis, but further testing may uncover heretofore undiscovered patterns.

In all events, for all problems of moderate to large size, we recommend BFGS18, and let the user draw his own conclusions concerning small problems.

Acknowledgment

The authors wish to thank the anonymous referee for many very useful contributions to the final form of this paper. Seldom has it been our privilege to benefit from such careful and thoughtful refereeing.

References

[1] C.G. Broyden, "Quasi-Newton methods and their application to functional minimization", *Mathematics of Computation* 21 (1967) 368–381.

[2] C.G. Broyden, "The convergence of a class of double rank minimization algorithms. 2. The new algorithm", *Journal of the Institute of Mathematics and its Applications* 6 (1970) 222–231.

[3] W.C. Davidon, "Optimally conditioned optimization algorithms without line searches", *Mathematical Programming* 9 (1975) 1–30.

[4] J.E. Dennis and J. More, "Quasi-Newton methods, motivation and theory", Computer Science Report TR74-217, Cornell University (1974).

[5] R. Fletcher, "A new approach to variable metric algorithms", *The Computer Journal* 13 (1970) 317–322.

[6] D. Goldfarb, "A family of variable-metric method derived by variational means", *Mathematics of Computation* 24 (1970) 23–26.

[7] D.H. Jacobson and W. Oksman, "An algorithm that minimizes homogeneous functions of n variables in $N+2$ iterations and rapidly minimizes general functions", Technical Report 618, Division of Engineering and Applied Physics, Harvard University, Cambridge, MA, (1970).

[8] G.P. McCormick and K. Ritter, "Methods of conjugate directions versus quasi-Newton methods", *Mathematical Programming* 3 (1972) 101–116.

[9] S.S. Oren, "Self-scaling variable metric algorithm. Part II", *Management Science* 20 (1974) 863–874.

[10] S.S. Oren, "On the selection of parameters in self-scaling variable metric algorithms", *Mathematical Programming* 3 (1974) 351–367.

[11] S.S. Oren and D.G. Luenberger, "Self-scaling variable metric algorithm. Part I", *Management Science* 20 (1974) 845–862.

[12] S.S. Oren and E. Spedicato, "Optimal conditioning of self-scaling variable metric algorithms", *Mathematical Programming* 10 (1976) 70–90.

[13] D.F. Shanno, "Conditioning of quasi-Newton methods for function minimization", *Mathematics of Computation* 24 (1970) 647–656.

[14] D.F. Shanno and K.H. Phua, "Minimization of unconstrained multivariate functions", *TOMS* 2 (1976) 87–94.

[15] E. Spedicato, "Computational experience with quasi-Newton algorithms for minimization problems of moderately large size", Report CISE-N-175, CISE Documentation Service, Segrate (Milano) (1975).

[16] D.F. Shanno and K.H. Phua, "Numerical comparison of several variable metric algorithms", MIS Tech. Rept 21, University of Arizona (1977).