SIAM J. NUMER. ANAL.
Vol. 7, No. 1, March 1970

COMPARISON OF GRADIENT METHODS FOR THE SOLUTION OF NONLINEAR PARAMETER ESTIMATION PROBLEMS*

YONATHAN BARD†

Abstract. The performance of several of the best known gradient methods is compared in the solution of some least squares, maximum likelihood, and Bayesian estimation problems. Modifications of the Gauss method (including Marquardt's) performed best, followed by variable metric rank one and Davidon–Fletcher–Powell methods, in that order. There appears to be no need to locate the optimum precisely in the one-dimensional searches. The matrix inversion method used with the Gauss algorithm must guarantee a positive definite inverse.

1. Introduction. The main purpose of this paper is to compare the performance of several gradient methods for solving parameter estimation problems of the least squares, maximum likelihood, and Bayesian types. All these problems require the maximization of objective functions possessing a certain form, described in § 2.

The methods tested are labeled "gradient methods" because all require at each iteration the evaluation of the gradient of the objective function. They were based on well established procedures, but in most cases significant changes were introduced to overcome some deficiencies. In all these methods, search directions of the form Rg are used, where R is a matrix and g is the gradient of an objective function. Most convergence difficulties are associated with indefiniteness or near-singularity of the matrices R that are prescribed by the various methods. We have attempted to overcome these difficulties by using the following devices:

(i) to replace negative or very small eigenvalues of R by reasonable positive ones (methods of Greenstadt [18] and Fariss and Law [11];
(ii) to increase all eigenvalues by a fixed positive amount (method of Marquardt [22]);
(iii) to choose the direction $-Rg$ if Rg fails.

A fourth device, due to Jennrich and Sampson [20], is based on the idea of treating each iteration of the nonlinear problem as a stepwise linear regression. It appeared too late for inclusion in the present survey.[1]

Next to the choice of R, the most vexing problem facing the implementation of maximization methods is the choice of step length in the direction Rg. Here we tested the following options:

(i) Accept the first step length to produce an increase in function value.
(ii) Use one quadratic approximation to predict a better step length than obtained in (i).
(iii) Continue searching along the chosen direction until the function starts decreasing again.
(iv) Locate the maximum of the function along the search direction to various degrees of precision.

* Received by the editors August 21, 1968, and in revised form April 2, 1969.

† International Business Machines Corporation, New York Scientific Center, New York, New York 10021. Now at IBM Cambridge Scientific Center, Cambridge, Massachusetts 02139.

[1] The two-exponentials problem used by Jennrich and Sampson was run with the Gc, M, and DFP3 methods of this paper. All converged in 6–7 iterations. Jennrich and Sampson report between 10 and 30 iterations for their method.

A total of thirteen choices of R and step length combinations was included in the tests. To have tested all possible combinations would have required a prohibitive effort.

2. Characterization of estimation problems. Consider a physical situation in which a set of dependent variables y is supposed to be related to independent variables x and unknown parameters θ by means of the equations

$$y = f(x, \theta). \tag{1}$$

A series of n experiments is performed, in each of which the value of y is measured for a given value of x. Let a subscript μ denote the value of a variable at the μth experiment. For each experiment we have a vector of residuals e_μ:

$$e_\mu(\theta) = y_\mu - f(x_\mu, \theta). \tag{2}$$

We define the moment matrix of the residuals as

$$M(\theta) = \sum_{\mu=1}^{n} e_\mu(\theta) e_\mu^T(\theta). \tag{3}$$

We estimate the value of θ by maximizing some objective function $\Phi(\theta)$. In the class of estimation problems that we shall consider, the objective function takes the form

$$\Phi(\theta) = \Psi(M(\theta)) + \Pi(\theta), \tag{4}$$

where Ψ and Π are functions whose form varies from one case to another.

The following are commonly employed estimation procedures which fall within our definition:

(a) Least squares. Here

$$\begin{aligned} \Pi &= 0, \\ \Psi &= -\mathrm{Tr}\,(M(\theta)). \end{aligned} \tag{5}$$

(b) Least squares, weighted by variables. Here

$$\begin{aligned} \Pi &= 0, \\ \Psi &= -\mathrm{Tr}\,(WM(\theta)), \end{aligned} \tag{6}$$

where W is a given matrix

(c) Maximum likelihood, with the x_μ known precisely and the y_μ measurements subject to errors. The latter are assumed to be normally distributed with zero mean and known covariance matrix V. Errors in different experiments are uncorrelated:

$$\begin{aligned} \Pi &= 0, \\ \Psi &= -\frac{1}{2}\mathrm{Tr}\,(V^{-1}M(\theta)). \end{aligned} \tag{7}$$

(d) As in (c), but with $V = \tau Q$, where Q is a known matrix and τ is an unknown scalar:

$$\Pi = 0, \qquad \Psi = -\frac{nm}{2} \log \operatorname{Tr}(Q^{-1}M(\theta)), \tag{8}$$

where m is the dimension of y.

(e) As in (c), but with V entirely unknown:

$$\Pi = 0, \qquad \Psi = -\frac{n}{2} \log \det M(\theta). \tag{9}$$

Equation (9) is derived in Eisenpress and Greenstadt [10]. Equation (8) can be derived analogously. If V is known to be diagonal, then off-diagonal elements in M should be replaced by zeros in (9).

(f) Bayesian estimation, by maximizing the logarithm of the posterior distribution:

$$\Pi = \log p_0(\theta), \qquad \Psi = \log L(\theta), \tag{10}$$

where $p_0(\theta)$ is the prior density function of θ, and $L(\theta)$ is the likelihood function. Under the proper assumptions, $\log L(\theta)$ coincides with Ψ as given in (7), (8) or (9).

In all cases, we seek to find the value θ^* of θ at which $\Phi(\theta)$ attains a maximum. Inequality constraints may be imposed, e.g., we may require that θ^* satisfy

$$h(\theta^*) \leqq 0. \tag{11}$$

These reflect physical conditions which the values of θ must satisfy. For instance, certain parameters may be required to be positive.

3. Gradient methods of solution. The maximizing value of θ is to be found by means of an iterative procedure. At the start of the ith iteration, we possess a current value θ_i, and we seek a new value θ_{i+1} using the formula

$$\theta_{i+1} = \theta_i + \rho_i R_i g_i, \tag{12}$$

where g_i is the gradient vector of $\Phi(\theta)$ at $\theta = \theta_i$, ρ_i is a scalar, and R_i is a matrix. Various gradient methods differ from each other in the choice of ρ_i and R_i. The following, however, are commonly considered desirable conditions on ρ_i and R_i.

CONDITION 1. R_i should be positive definite. This guarantees that if $g_i \neq 0$, then for a sufficiently small positive ρ_i we have

$$\Phi(\theta_{i+1}) > \Phi(\theta_i). \tag{13}$$

CONDITION 2. R_i should be some approximation to $-H_i^{-1}$, where H_i is the Hessian matrix of Φ at $\theta = \theta_i$. When $R_i = -H_i^{-1}$ we have the Newton–Raphson method whose efficiency within its region of convergence is well known (see Crockett and Chernoff [7], Greenstadt [18]).

CONDITION 3. ρ_i should be chosen so that (13) holds. This guarantees that θ_{i+1} is at least in some sense closer to the maximum than was θ_i. We call a value of ρ_i satisfying (13) *acceptable.*

We now describe in detail the various methods that were used in the present study.

4. Choice of R_i.

4.1. The Gauss method (G), generalized and modified. This method, originated by Gauss [15], has been extensively applied to least squares problems; see, e.g., Booth and Peterson [3], Hartley [19]. We extend it here to apply to our more general class of problems. We start by differentiating $\Psi(M(\theta))$. We have:

$$g_\alpha \equiv \frac{\partial \Psi}{\partial \theta_\alpha} = \sum_{j,k} \frac{\partial \Psi}{\partial M_{jk}} \frac{\partial M_{jk}}{\partial \theta_\alpha}. \tag{14}$$

But

$$\frac{\partial M_{jk}}{\partial \theta_\alpha} = -\sum_{\mu=1}^{n} \left(e_{\mu j} \frac{\partial f_{\mu k}}{\partial \theta_\alpha} + e_{\mu k} \frac{\partial f_{\mu j}}{\partial \theta_\alpha} \right), \tag{15}$$

where we have written $f_{\mu k}$ for $f_k(x_\mu, \theta)$. By further differentiating (14) we find:

$$H_{\alpha\beta} \equiv \frac{\partial^2 \Psi}{\partial \theta_\alpha \partial \theta_\beta} = \sum_{j,k} \frac{\partial \Psi}{\partial M_{jk}} \frac{\partial^2 M_{jk}}{\partial \theta_\alpha \partial \theta_\beta} + \sum_{j,k,l,m} \frac{\partial^2 \Psi}{\partial M_{jk} \partial M_{lm}} \frac{\partial M_{jk}}{\partial \theta_\alpha} \frac{\partial M_{lm}}{\partial \theta_\beta} \tag{16}$$

and

$$\frac{\partial^2 M_{jk}}{\partial \theta_\alpha \partial \theta_\beta} = \sum_{\mu=1}^{n} \left(\frac{\partial f_{\mu j}}{\partial \theta_\alpha} \frac{\partial f_{\mu k}}{\partial \theta_\beta} + \frac{\partial f_{\mu k}}{\partial \theta_\alpha} \frac{\partial f_{\mu j}}{\partial \theta_\beta} - e_{\mu j} \frac{\partial^2 f_{\mu k}}{\partial \theta_\alpha \partial \theta_\beta} - e_{\mu k} \frac{\partial f_{\mu j}}{\partial \theta_\alpha \partial \theta_\beta} \right), \tag{17}$$

so that due to the symmetry of M,

$$\begin{aligned} H_{\alpha\beta} = {} & 2 \sum_{\mu=1}^{n} \sum_{j,k} \frac{\partial f_{\mu j}}{\partial \theta_\alpha} \frac{\partial \Psi}{\partial M_{jk}} \frac{\partial f_{\mu k}}{\partial \theta_\beta} \\ & -2 \sum_{\mu=1}^{n} \sum_{j,k} e_{\mu j} \frac{\partial \Psi}{M_{jk}} \frac{\partial^2 f_{\mu k}}{\partial \theta_\alpha \partial \theta_\beta} \\ & +4 \sum_{\mu,\eta=1}^{n} \sum_{j,k,l,m} e_{\mu j} \frac{\partial f_{\mu k}}{\partial \theta_\alpha} \frac{\partial^2 \Psi}{\partial M_{jk} \partial M_{lm}} \frac{\partial f_{\eta m}}{\partial \theta_\beta} e_{\eta l}. \end{aligned} \tag{18}$$

In this expression, all but the first term contain some residuals $e_{\mu j}$. It is the essence of the Gauss method that all these terms are neglected, in the hope that these residuals will turn out to be small anyway. We accept, then, in place of H the approximation

$$N_{\alpha\beta} \equiv 2 \sum_{\mu=1}^{n} \sum_{j,k} \frac{\partial f_{\mu j}}{\partial \theta_\alpha} \frac{\partial \Psi}{\partial M_{jk}} \frac{\partial f_{\mu k}}{\partial \theta_\beta} \tag{19}$$

or, in matrix form,

$$N = 2 \sum_{\mu=1}^{n} \frac{\partial f_\mu^T}{\partial \theta} \frac{\partial \Psi}{\partial M} \frac{\partial f_\mu}{\partial \theta}. \tag{20}$$

Finally, in analogy to the Newton–Raphson method, we choose

$$R_i = -\left[N + \frac{\partial^2 \Pi}{\partial\theta\partial\theta}\right]^{-1}_{\theta=\theta_i}. \tag{21}$$

The function Π is usually sufficiently simple so that the evaluation of its Hessian $\partial^2\Pi/\partial\theta\partial\theta$ presents no difficulties. The virtue of the Gauss method lies in the fact that it generates an approximation to H without the need to compute second derivatives of either Ψ or f.

Incidentally, (14) combined with (15) appears in matrix form as:

$$g = -2\sum_{\mu=1}^{n} \frac{\partial f_\mu^T}{\partial\theta}\frac{\partial\Psi}{\partial M}e_\mu. \tag{22}$$

In Table 1 we give formulas for $\partial\psi/\partial M$ in the cases described by (5)–(9).

TABLE 1
Objective functions and their derivatives

Equation	Ψ	$\frac{\partial\Psi}{\partial M}$
5	$-\mathrm{Tr}(M)$	$-I$
6	$-\mathrm{Tr}(WM)$	$-W$
7	$-\frac{1}{2}\mathrm{Tr}(V^{-1}M)$	$-\frac{1}{2}V^{-1}$
8	$-\frac{nm}{2}\log \mathrm{Tr}(Q^{-1}M)$	$-\frac{nm}{2\,\mathrm{Tr}(Q^{-1}M)}Q^{-1}$
9	$-\frac{n}{2}\log\det M$	$-\frac{n}{2}M^{-1}$

It will be seen that in all the cases considered, $\partial\Psi/\partial M$ is negative definite (or at least semidefinite). It follows from (20) that N is also negative definite, and R_i is positive definite provided $\Pi = 0$ or its Hessian is negative definite, as it usually will be (e.g., if $p_0(\theta)$ is normal). Hence, at least in principle R_i is positive definite. Rounding errors in the matrix inversion called for by (21) may, however, give rise to an indefinite R_i in ill-conditioned cases. To overcome this problem, we use the following method based on a suggestion by Greenstadt [18] and by Fariss and Law [11].

Let

$$A_i = -\left[N + \frac{\partial^2\Pi}{\partial\theta\partial\theta}\right]_{\theta=\theta_i}.$$

We form the scaled version of A_i:

$$B_i \equiv C_i^{-1}A_iC_i^{-1}, \tag{23}$$

where C_i is a diagonal matrix with $C_{i,jj} = |A_{i,jj}|^{1/2}$. Clearly, B_i has all ones on the main diagonal. Let the eigenvalues and vectors of B_i be denoted by λ_j, v_j, $j = 1, 2, \cdots, l$, respectively. Then

$$B_i^{-1} = \sum_{j=1}^{l} \lambda_j^{-1} v_j v_j^T \tag{24}$$

and

$$A_i^{-1} = C_i^{-1} B_i^{-1} C_i^{-1}. \tag{25}$$

Instead of B_i, we use, however, a slightly modified matrix

$$B_i^* = \sum_{j=1}^{l} \tau_j v_j v_j^T, \tag{26}$$

where $\tau_j = |\lambda_j|^{-1}$ if $|\lambda_j| > 10^{-6}$ and $\tau_j = 1$ otherwise. We now take

$$R_i = C_i^{-1} B_i^* C_i^{-1}, \tag{27}$$

which is guaranteed positive semidefinite. Clearly, if all λ_j are positive (as they should be), then $R_i = A_i^{-1}$. The scaling implied by (23), while not strictly essential, has been found to improve significantly the accuracy of the calculations and thereby frequently reduce the number of required iterations.

4.2. The Marquardt method (M). The Marquardt method [22] is related to those of Levenberg [21] and Goldfeld, Quandt and Trotter [17]. It may be viewed as an alternative way of guaranteeing the positive definiteness of R_i. It consists of replacing (27) with

$$R_i = C_i^{-1}(B_i + \beta_i I)^{-1} C_i^{-1} = (A_i + \beta_i C_i^2)^{-1}, \tag{28}$$

where β_i is a positive constant. Since the eigenvalues of $B_i + \beta_i I$ are $\lambda_j + \beta_i$, $j = 1, 2, \cdots, l$, the choice $\beta_i > -\min \lambda_j$ guarantees the positive definiteness of R_i.

Note that C_i^2 is simply the absolute value of the diagonal part of A_i. This scaling is recommended in Marquardt's paper [22].

In Marquardt's method, $\rho_i = 1$ and the step length is determined by the choice of β_i. A sufficiently large value of β_i guarantees that (13) holds provided $g_i \neq 0$.

There are two ways of computing R_i for each value of β_i: (a) invert $A_i + \beta_i C_i^2$; (b) compute the eigenvalues and vectors of B_i. Then $(B_i + \beta_i I)^{-1}$ can be formed in analogy to (24), with $\lambda_j + \beta_i$ replacing λ_j, and $(A_i + \beta_i C_i^2)^{-1}$ is formed in analogy to (25). Method (b) requires only one eigenvalue analysis per iteration, no matter how many values of β have to be tried. Also, method (b) enables one to find the minimum β_i to ensure positive definiteness in cases where A_i is not positive definite. Method (b) was used in this study.

4.3. Variable metric methods. Let $\sigma_i \equiv \theta_{i+1} - \theta_i$ and $\eta_i \equiv g_{i+1} - g_i$. Suppose we have a matrix A_i and we wish to modify it by adding a matrix B_i so that $A_{i+1} \equiv A_i + B_i$ is an approximation to H_i^{-1}. Now we have approximately from

Taylor's series

$$\eta_i = H_i \sigma_i, \tag{29}$$

so that

$$\sigma_i = H_i^{-i} \eta_i. \tag{30}$$

We require that (30) be satisfied with A_{i+1} replacing H_i^{-1}. This leads to

$$B_i \eta_i = \sigma_i - A_i \eta_i. \tag{31}$$

Various variable metric methods differ in the manner in which B_i is chosen so as to satisfy (31).

1. *Rank one correction (ROC).* If we choose B_i to be a symmetric matrix of rank one, the only possible solution turns out to be

$$B_i = \frac{1}{p_i^T \eta_i} p_i p_i^T, \tag{32}$$

where

$$p_i \equiv \sigma_i - A_i \eta_i. \tag{33}$$

Broyden [5] has proved the following theorem.

THEOREM 1. *Let $\Phi(\theta)$ be quadratic with constant nonsingular Hessian H. Let $\theta_1, \theta_2, \cdots, \theta_{l+1}$ be a sequence of points such that $\sigma_1, \sigma_2, \cdots, \sigma_l$ are linearly independent directions: Let A_1 be an arbitrary symmetric matrix. Suppose $p_i^T \eta_i \neq 0$ for $i = 1, 2, \cdots, l$. Then $A_{l+1} = H^{-1}$. (Here l is the dimension of θ.)*

Note that neither step length nor direction is prescribed. Similar results were obtained by Davidon [9] and by Fiacco and McCormick [12].

If Φ is not quadratic, we expect the A_i to become approximations to H_i^{-1} as i increases. We would, therefore, like to use $R_i = -A_i$. This choice of R_i is not guaranteed positive definite. We have overcome this problem in two alternative ways:

(i) Determine the eigenvalues of $-A_i$ (after scaling), and replace the negative one by their absolute values. This is analogous to the Greenstadt procedure adopted for the Gauss method.

(ii) The directional derivative of Φ in the direction $R_i g_i$ is given by $g_i^T R_i g_i$. Since we want this derivative to be positive, we choose $R_i = A_i$ if $g_i^T A_i g_i > 0$, and $R_i = -A_i$ otherwise. However, when the quantity

$$\left| \frac{g_i^T R_i g_i}{[(g_i^T g_i)(g_i^T R_i^T R_i g_i)]^{1/2}} \right|$$

(this is the cosine of the angle between g_i and σ_i) is less than 10^{-4}, we revert to method (i), where no eigenvalue is allowed to be less than 10^{-4}.

2. *Davidon–Fletcher–Powell method (DFP).* In the first variable metric method originated by Davidon [8] and modified by Fletcher and Powell [14], the matrix B_i was chosen to be of rank two. Specifically,

$$B_i = \frac{1}{\sigma_i^T \eta_i} \sigma_i \sigma_i^T - \frac{1}{\eta_i^T A_i \eta_i} A_i \eta_i \eta_i^T A_i. \tag{34}$$

Fletcher and Powell prove that if we choose $R_i = -A_i$, and $\rho_i = \mu_i$ where μ_i is the value of ρ for which $\Phi(\theta_i + \rho R_i g_i)$ attains a maximum, then the following theorem holds.

THEOREM 2. *Let A_1 be an arbitrary symmetric negative definite matrix. Then all subsequent matrices A_i are negative definite. Suppose Φ is quadratic with constant negative definite Hessian H, and suppose the directions $\sigma_1, \sigma_2, \cdots, \sigma_l$ turn out linearly independent. Then $A_{l+1} = H^{-1}$.*

Note that both step length and direction are prescribed.

In view of Theorem 2, we may use $R_i = -A_i$ and be assured that R_i is positive definite. In practice, however, Bard [2] has pointed out that A_i may become nearly singular due to poor scaling of Φ. Therefore, when the cosine of the angle between $A_i g_i$ and g_i turns out to be less than 10^{-5}, we reinitialize A_i to a diagonal matrix whose (jj) element is minus the absolute value of the ratio of the jth element of σ_i to the jth element of g_i.

Both rank one and rank two methods require an initial matrix A_i. We choose the diagonal matrix whose (jj) element is as above, with $\frac{1}{2}\theta_1$ replacing σ_i and g_1 replacing g_i.

3. *Inverse rank one correction method (IROC).* If we regard A_{i+1} as an approximation to H_i rather than to H_i^{-1}, we substitute $A_i + B_i$ for H_i in (29) and are led to

$$B_i\sigma_i = \eta_i - A_i\sigma_i, \tag{35}$$

which has the rank one solution

$$B_i = \frac{1}{q_i^T\sigma_i} q_i q_i^T, \tag{36}$$

with

$$q_i \equiv \eta_i - A_i\sigma_i. \tag{37}$$

Under conditions similar to those of Broyden's theorem, the matrices A_i converge to H. In fact, if $A_i = 0$ and H is of rank $k \leqq l$, then $A_{k+1} = H$ provided the vectors η_i, $i = 1, 2, \cdots, k$, are linearly independent.

Since A_i is supposedly an approximation to H, we wish to use $R_i = -A_i^{-1}$. This choice does not guarantee positive definiteness, so we use Greenstadt's trick (described above for the Gauss method) of replacing negative eigenvalues of the scaled $-A_i^{-1}$ by their absolute values in forming R_i. Eigenvalues smaller in absolute values than 10^{-4} are replaced by 10^{-4}.

The initial value A_1 is chosen as the inverse of that chosen for A_1 in the other variable metric methods. It is easy to show that under this condition the A_i matrices in the inverse and straight rank one methods remain inverses of each other provided the same sequence of points θ_i is chosen. This, in fact, occurs only if all A_i turn out to be negative definite.

5. Choice of ρ_i. The inequality (13) represented an inviolable condition for the choice of ρ_i. This does not determine ρ_i uniquely, and we have a great deal of leeway in deciding whether merely to look for an acceptable value of ρ_i, or whether to search for the best value of ρ_i.

We introduce the following definitions:

(38) $F_i(\rho) \equiv \Phi(\theta_i + \rho R_i g_i)$.

$\mu_i \equiv$ a positive value of ρ at which $F_i(\rho)$ attains a local maximum.

$\rho_{\max} \equiv$ upper bound on ρ_i. If there are inequality constraints, $\rho_{\max}$ is determined as the minimum positive ρ such that $\theta_i + \rho R_i g_i$ is on a constraint. Otherwise, $\rho_{\max}$ is an arbitrary large number.

If $\rho^{(j)}$ is a possible value for ρ_i, then we designate

(39) $$F^{(j)} \equiv F_i(\rho^{(j)}).$$

Note that at the start of the ith iteration we possess the value of $F_i(0) = \Phi(\theta_i)$ and

(40) $$F_i'(0) \equiv \left.\frac{dF_i}{d\rho}\right]_{\rho=0} = g_i^T R_i g_i.$$

We also assume that a value $\rho^{(0)}$ is given at the start of the ith iteration. The determination of $\rho^{(0)}$ is described under (h) below. We now detail the various algorithms that were used to determine ρ_i.

(a) *First acceptable value.*

Step 1. Compute $F^{(0)}$. If $F^{(0)} > F_i(0)$, accept $\rho_i = \rho^{(0)}$. Otherwise continue.

Step 2. Determine the second degree polynomial in ρ which agrees with $F_i(\rho)$ at $\rho = 0$ and $\rho = \rho^{(0)}$, and whose slope at $\rho = 0$ agrees with $F_i'(0)$. Let $\rho^{(1)}$ be the point at which this polynomial is stationary. Define

(41) $$\rho^{(2)} \equiv \max [0.25\rho^{(0)}, \min (0.75\rho^{(0)}, \rho^{(1)})].$$

We refer to each execution of this step as an *interpolation.*

Step 3. Replace $\rho^{(0)}$ with $\rho^{(2)}$ and return to Step 1.

(b) *Additional try, Version* I. The algorithm proceeds as in (a), except that if $F^{(0)} > F_i(0)$ on the first attempt, we proceed as follows.

Step 4. Determine $\rho^{(1)}$ as in Step 2 above.

Step 5. Define

(42) $$\rho^{(3)} \equiv \min (\rho^{(1)}, 0.75\rho_{\max}).$$

Step 6. If $|\rho^{(3)} - \rho^{(0)}| \leqq 0.1\rho^{(0)}$ or $\rho^{(1)} \leqq 0.25\rho^{(0)}$, accept $\rho_i = \rho^{(0)}$. Otherwise continue.

Step 7. Compute $F^{(3)}$. Take $\rho_i = \rho^{(3)}$ or $\rho_i = \rho^{(0)}$ depending on whether $F^{(3)}$ or $F^{(0)}$ is the larger.

(c) *Additional try, Version* II. Proceed as in (b) above, with the following steps modified.

Step 5. Define

(43) $$\rho^{(3)} \equiv \begin{cases} \min [\rho^{(1)}, \rho^{(0)} + 0.5(\rho_{\max} - \rho^{(0)})] & \text{if } \rho^{(1)} > 0, \\ \min [2\rho^{(0)}, \rho^{(0)} + 0.5(\rho_{\max} - \rho^{(0)})] & \text{if } \rho^{(1)} \leqq 0. \end{cases}$$

Step 6. If $|\rho^{(3)} - \rho^{(0)}| \leqq 0.1\rho^{(0)}$, accept $\rho_i = \rho^{(0)}$. Otherwise, proceed to Step 7.

(d) *Bracket maximum, Version* I. Proceed as in (a) above, except that if $F^{(0)} > F_i(0)$ on the first attempt, we proceed as follows.

Step 4. Define

$$\rho^{(1)} \equiv 2\rho^{(0)}. \tag{44}$$

Step 5. If $\rho^{(1)} > 0.75\rho_{\max}$, accept $\rho_i = \rho^{(0)}$. Otherwise continue.

Step 6. Compute $F^{(1)}$. If $F^{(1)} \leqq F^{(0)}$, accept $\rho_i = \rho^{(0)}$. Otherwise continue.

Step 7. Replace $\rho^{(0)}$ with $\rho^{(1)}$ and return to Step 4.

(e) *Bracket maximum, Version* II. Proceed as in (a) above, except that if $F^{(0)} > F_i(0)$ on the first attempt, we proceed as follows.

Step 4. Define

$$\rho^{(1)} \equiv \min\,[2\rho^{(0)}, \rho^{(0)} + 0.5(\rho_{\max} - \rho^{(0)})]. \tag{45}$$

Step 5. Compute $F^{(1)}$. If $F^{(1)} \leqq F^{(0)}$, accept $\rho_i = \rho^{(0)}$. Otherwise continue.

Step 6. Replace $\rho^{(0)}$ with $\rho^{(1)}$ and return to Step 4.

(f) *Locate maximum with tolerance* ε.

Step 1. Let $\rho^{(0)} = \min\,[2|F_i(0)/F_i'(0)|, 0.5\rho_{\max}]$. Compute $F^{(0)}$. Set $\lambda_0 = 0$; $\lambda_1 = \rho^{(0)}$; $f_0 = F_i(0)$; $f_1 = F^{(0)}$.

Step 2. Compute $\rho^{(1)}$ as in Step 2 under (a) above.

Step 3. If $\rho^{(1)} > 0$, proceed to Step 8. Otherwise continue.

Step 4. Set $\lambda_2 = \min\,[2\lambda_1, \lambda_1 + 0.999(\rho_{\max} - \lambda_1)]$.

Step 5. Compute $f_2 = F_i(\lambda_2)$.

Step 6. If $f_2 \leqq f_1$, proceed to Step 17. Otherwise continue.

Step 7. Let $\lambda_0 = \lambda_1$; $f_0 = f_1$; $\lambda_1 = \lambda_2$; $f_1 = f_2$, and return to Step 4.

Step 8. Proceed to Step 10, 4 or 9 if $\rho^{(1)} - \lambda_1$ is negative, zero or positive, respectively.

Step 9. Set $\lambda_2 = \min\,(\rho^{(1)}, 0.999\lambda_{\max})$ and proceed to Step 5.

Step 10. Set $\rho^{(2)} = \max\,[0.25\rho^{(0)}, \min\,(0.75\rho^{(0)}, \rho^{(1)})]$.

Step 11. Compute $F^{(2)}$.

Step 12. If $F^{(2)} > f_0$, proceed to Step 14. Otherwise continue

Step 13. Set $\lambda_1 = \rho^{(2)}$, $f_1 = F^{(2)}$, $\rho^{(0)} = \rho^{(2)}$. Return to Step 2.

Step 14. If $F^{(2)} \geqq f_1$, proceed to Step 16. Otherwise continue.

Step 15. Set $\lambda_0 = \rho^{(2)}$, $f_0 = F^{(2)}$, and return to Step 4.

Step 16. Set $\lambda_2 = \lambda_1$, $f_2 = f_1$, $\lambda_1 = \rho^{(2)}$, $f_1 = F^{(2)}$.

Step 17. At this point we have three points $0 \leqq \lambda_0 < \lambda_1 < \lambda_2$ with the corresponding function values $f_0 \leqq f_1$, $f_1 \geqq f_2$, so that the maximum μ_i is bracketed by $\lambda_0 < \mu_i < \lambda_2$. From now on, we aim to narrow the bracketing interval.

Set switch $k = 1$. Compute $\rho^{(3)}$, the point where a parabola matching $F_i(\rho)$ at our three points attains its maximum.

Step 18. If $|\rho^{(3)} - \lambda_1| \leqq \varepsilon\lambda_1$, accept $\rho_i = \lambda_1$. Otherwise continue.

Step 19. Set $\rho^{(4)} = \max\,\{\lambda_0 + 0.1(\lambda_2 - \lambda_0), \min\,[\lambda_0 + 0.9(\lambda_2 - \lambda_0), \rho^{(3)}]\}$.

Step 20. Compute $F^{(4)}$.

Step 21. If $\rho^{(4)} = \lambda_1$, accept $\rho_i = \lambda_1$. If $\rho^{(4)} > \lambda_1$, proceed to Step 29. Otherwise continue.

Step 22. If $F^{(4)} > f_1$, proceed to Step 27. Otherwise continue.

Step 23. Set $\lambda_0 = \rho^{(4)}$, $f_0 = F^{(4)}$.

Step 24. If $k = 2$, return to Step 17. Otherwise continue.
Step 25. Set $\rho^{(4)} = 0.5(\lambda_1 + \lambda_2)$.
Step 26. Set $k = 2$, and return to Step 20.
Step 27. Set $\lambda_2 = \lambda_1, f_2 = f_1$.
Step 28. Set $\lambda_1 = \rho^{(4)}, f_1 = F^{(4)}$. Return to Step 17.
Step 29. If $F^{(4)} \leqq f_1$, proceed to Step 31. Otherwise continue.
Step 30. Set $\lambda_0 = \lambda_1, f_0 = f_1$, and proceed to Step 28.
Step 31. Set $\lambda_2 = \rho^{(4)}, f_2 = F^{(4)}$.
Step 32. If $k = 2$, return to Step 17. Otherwise continue.
Step 33. Set $\rho^{(4)} = 0.5(\lambda_0 + \lambda_1)$, and proceed to Step 26.

Note. While the basic idea is to use a quadratic interpolation scheme, Steps 19, 25, and 33 protect against grossly inefficient behavior in cases where the quadratic fit is particularly poor (e.g., when penalty functions are used).

(g) *β-Adjustment in Marquardt's method.* Here $\rho_i = 1$, but β_i is adjusted as follows.

Step 1. When $i = 1$, start with $\beta = 0.01$.
Step 2. Compute $\theta^{(1)} = \theta_i + (A_i + \beta C_i^2)^{-1} g_i$ (see (28)). Compute $F^{(1)} = \Phi(\theta^{(1)})$.
Step 3. If $F^{(1)} > \Phi(\theta_i)$, accept $\theta_{i+1} = \theta^{(1)}$, and replace β with $\max(0.1\beta, 10^{-7})$. Otherwise continue.
Step 4. If the computed direction forms an angle greater than 45° with the gradient, replace β with 10β and return to Step 2. Otherwise, interpolate as under (a) above.

(h) *$\rho^{(0)}$-Adjustment for methods* (a)–(e).

Step 1. When $i = 1$, set $j = 0$.
Step 2. Divide j by 2 and retain the integral part.
Step 3. Each time an interpolation is performed, increase j by 1.
Step 4. In methods (a), (c) and (e) set, for each iteration,

$$\rho^{(0)} = 2^{-j} \min(1, 0.5\rho_{\max}). \tag{46}$$

In methods (b), (d) set, for each iteration,

$$\rho^{(0)} = \min(0.75\rho_{\max}, 2^{-j}). \tag{47}$$

Note. No significance should be attached to the difference between methods (b) and (c), or between (d) and (e). These arose by chance in the course of the development of the programs.

6. Implementation details.

6.1. Inequality constraints. These were treated by means of Carroll's method [6] of penalty functions. The only constraints used were upper bounds η_j and lower bounds α_j on the parameters $\theta_j, j = (1, 2, \cdots, l)$. The penalty function

$$P(\theta) \equiv \sum_{j=1}^{l} \gamma_j \left(\frac{1}{\alpha_j - \theta_j} + \frac{1}{\theta_j - \eta_j} \right) \tag{48}$$

was added to the objective function $\Phi(\theta)$ to form a new objective function

$$Q(\theta) \equiv \Phi(\theta) + P(\theta). \tag{49}$$

The γ_j are positive constants. The initial guess θ_1 must be in the interior of the feasible region. All the algorithms we have described guarantee that all subsequent θ_i are feasible.

The calculation proceeds in two phases:

(a) find the feasible point $\tilde{\theta}$ which maximizes $Q(\theta)$;

(b) using $\tilde{\theta}$ as the initial guess, find the feasible point θ^* which maximizes $\Phi(\theta)$.

Since in all cases tested the solution was an interior point, phase (b) usually required only one or two iterations.

6.2. Termination criterion. A criterion of the type suggested by Marquardt [22] was used to terminate the calculations: If each component of the correction vector $|\rho_i R_i g_i|$ was less than 10^{-4} times the corresponding component of $|\theta_i| + 0.001$, then θ_i was accepted as the solution to the problem. When constraints were present, the same criterion was used for terminating each phase.

6.3. Derivatives. All required derivatives were computed when possible from their analytic formulas. In the case of models given in the form of differential equations requiring numerical integration, the derivatives were obtained by simultaneous integration of the sensitivity equations (see e.g., Tomovic [23]). That is, if y is given by

$$\begin{aligned} \frac{dy}{dt} &= f(y, x, t, \theta), \\ y_{t=0} &= q(x, \theta), \end{aligned} \tag{50}$$

then, differentiation with respect to θ yields

$$\begin{aligned} \frac{d}{dt}\frac{\partial y}{\partial \theta} &= \frac{\partial f}{\partial y}\frac{\partial y}{\partial \theta} + \frac{\partial f}{\partial \theta}, \\ \left.\frac{\partial y}{\partial \theta}\right]_{t=0} &= \frac{\partial q}{\partial \theta}, \end{aligned} \tag{51}$$

which yields a set of linear ordinary differential equations in the required derivatives $\partial y/\partial\theta$. All numerical integrations were performed using a variable-step, third order predictor-corrector method described by Bard [1].

7. The algorithm tested. Table 2 lists the algorithms actually tested.

8. The test problems.

8.1. Introduction. There are five test problems. Of these, one was done in six different versions, for a total of ten cases. The dimensions and nature of the objective function in each problem are summarized in Table 3. The precise formulations are given in subsequent sections. The problems are arranged in order of increasing number of unknown parameters.

The problem descriptions are followed by tables setting forth the data used, as well as the initial guess θ_1 and bounds for the parameters. Also included are the best estimates of the parameters, e.g., those that have given the largest known value of the objective function.

TABLE 2
Definition of algorithms used

Designation	Description	R_l selection described in §4	ρ_l selection described in §5	Major operations required per iteration
Ga	Modified Gauss, accept first value	4.1	5(a), (h)	EVV
Gc	Modified Gauss, try additional value	4.1	5(c), (h)	EVV
Ge	Modified Gauss, bracket maximum	4.1	5(e), (h)	EVV
M	Marquardt	4.2	5(g)	EVV (or SEQ for each evaluation)
ROC1	Rank one correction, absolute eigenvalues, try additional value	4.3.1(i)	5(b), (h)	EVV
ROC2	Rank one correction, reverse direction if necessary, try additional value	4.3.1(ii)	5(b), (h)	None
DFP1	Davidon–Fletcher–Powell, $\varepsilon = 10^{-1}$	4.3.2	5(f)	None
DFP2	Davidon–Fletcher–Powell, $\varepsilon = 10^{-2}$	4.3.2	5(f)	None
DFP3	Davidon–Fletcher–Powell, $\varepsilon = 3 \times 10^{-3}$	4.3.2	5(f)	None
DFP4	Davidon–Fletcher–Powell, $\varepsilon = 10^{-3}$	4.3.2	5(f)	None
DFP5	Davidon–Fletcher–Powell, $\varepsilon = 10^{-4}$	4.3.2	5(f)	None
IROCb	Inverse rank one correction, try additional value	4.3.3	5(b), (h)	EVV
IROCd	Inverse rank one correction, bracket maximum	4.3.3	5(d), (h)	EVV

EVV = Eigenvalues and vectors of a real symmetric matrix.
SEQ = Solution of simultaneous linear equations.

TABLE 3
Problem characterization

Problem no.	l (No. of parameters)	m (No. of equations)	n (No. of experiments)	$\Psi(\theta)$ (Equation number)	$\Pi(\theta)$	Upper and lower bounds
1	3	1	15	(5)	0	yes
2	5	3	20	(9)	0	no
3d1	6	5	8	(7) (V diagonal)	0	yes
3d2	6	5	8	(8) (Q diagonal)	0	yes
3d3	6	5	8	(9) (off diagonal elements of M set to zero)	0	yes
3f1	6	5	8	(7)	0	yes
3f2	6	5	8	(8)	0	yes
3f3	6	5	8	(9)	0	yes
4	8	1	30	(7)	(10) Normal prior	yes
5	10	1	30	(7)	(10) Normal prior	yes

The problems and data were artificially constructed for the specific purpose of demonstrating various types of parameter estimation problems. Their use in this survey was an afterthought. Thus, while not being "real world" problems, they do escape the stigma of having been devised only to test algorithms.

8.2. Problem definitions.

Problem 1. A simple nonlinear least squares problem, with all parameters required to be positive:

$$y = \theta_1 + \frac{x_1}{\theta_2 x_2 + \theta_3 x_3}. \tag{52}$$

The data appear in Tables 4 and 5.

Problem 2. A linear three equation maximum likelihood problem, with unknown covariance matrix:

$$\begin{aligned} y_1 &= \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3, \\ y_2 &= \theta_1 x_2 + \theta_2 x_3 + \theta_3 x_1, \\ y_3 &= \theta_1 x_3 + \theta_2 x_2 + \theta_3 x_1. \end{aligned} \tag{53}$$

TABLE 4
Data for Problem 1

μ	y	x_1	x_2	x_3
1	0.14	1	15	1
2	0.18	2	14	2
3	0.22	3	13	3
4	0.25	4	12	4
5	0.29	5	11	5
6	0.32	6	10	6
7	0.35	7	9	7
8	0.39	8	8	8
9	0.37	9	7	7
10	0.58	10	6	6
11	0.73	11	5	5
12	0.96	12	4	4
13	1.34	13	3	3
14	2.10	14	2	2
15	4.39	15	1	1

TABLE 5
Parameter information for Problem 1

j	Initial guess $\theta_{1,j}$	Best estimate θ_j^*	Lower bound α_j	Upper bound η_j	Penalty coefficient γ_j
1	1	0.08241040	0	100	0.0001001
2	1	1.133033	0	100	0.0001001
3	1	2.343697	0	100	0.0001001

TABLE 6
Data for Problem 2

μ	y_1	y_2	y_3	x_1	x_2	x_3
1	4.36	5.21	5.35	0.871	0.643	0.550
2	4.99	3.30	3.10	0.228	0.669	0.854
3	1.67	θ_4^*	2.75	0.528	0.229	0.170
4	2.17	1.48	1.49	0.110	0.354	0.337
5	2.98	4.69	4.23	0.911	0.056	0.493
6	4.46	3.87	3.15	0.476	0.154	0.918
7	1.79	3.18	3.57	0.655	0.421	0.077
8	1.71	3.13	3.07	0.649	0.140	0.199
9	3.07	5.01	4.58	0.995	0.045	θ_5^*
10	0.94	0.93	0.74	0.130	0.016	0.195
11	4.97	5.37	5.35	0.823	0.690	0.690
12	4.32	4.85	5.46	0.768	0.992	0.389
13	2.17	1.78	2.43	0.203	0.740	0.120
14	2.22	2.18	2.44	0.302	0.519	0.221
15	2.88	4.90	5.11	0.991	0.450	0.249
16	2.29	1.94	1.46	0.224	0.030	0.502
17	3.76	3.39	2.71	0.428	0.127	0.772
18	1.99	2.93	3.31	0.552	0.494	0.110
19	4.95	4.08	4.19	0.461	0.824	0.714
20	2.96	4.26	4.48	0.799	0.494	0.295

* Missing values.

TABLE 7
Parameter information for Problem 2

j	Initial guess θ_{1j}	Best estimate θ_j^*
1	0	0.9925145
2	0	2.005293
3	0	3.999732
4	0	2.680371
5	0	0.4977683

The data appear in Tables 6 and 7. Two data items, missing from Table 6, are treated as additional unknown parameters θ_4 and θ_5.

Problem 3. A fifth order (see Note 2 below) nonlinear dynamic system with given initial conditions. All five state variables are observed at a discrete set of points in time. All parameters are required to be positive. Six different versions of the objective function are used:

3d1 = known diagonal covariance matrix,
3d2 = diagonal covariance matrix known except for a multiplicative factor,
3d3 = unknown diagonal covariance matrix;
3f1, 3f2, 3f3: like 3d1, 3d2, 3d3 respectively, but with nondiagonal covariance matrix.

Note 1. Problems 3d1, 3d2, 3f1 and 3f2 may be regarded as weighted least squares problems. The objective functions in 3d2 and 3f2 are essentially logarithms of those in 3d1 and 3f1, so that the answers should be identical. Problems 3d3 and 3f3 may be regarded as weighted least squares problems with unknown weights.

Note 2. Although nominally fifth order, in fact two of the differential equations are linear combinations of the other three. The system represents four simultaneous chemical reactions, two of them reversible, involving five species.

The vector of dependent variables y is the solution of the following set of differential equations:

$$\begin{aligned}
\frac{dy_1}{dt} &= -\theta_1 y_1 y_2 + \theta_2 y_3, \\
\frac{dy_2}{dt} &= -\theta_1 y_1 y_2 + \theta_2 y_3 - \theta_4 y_2 y_3 + \theta_5 y_5 - \theta_6 y_2 y_4, \\
\frac{dy_3}{dt} &= \theta_1 y_1 y_2 - \theta_2 y_3 - \theta_3 y_3 - \theta_4 y_2 y_3, \\
\frac{dy_4}{dt} &= \theta_3 y_3 + \theta_5 y_5 - \theta_6 y_2 y_4, \\
\frac{dy_5}{dt} &= \theta_4 y_2 y_3 - \theta_5 y_5 + \theta_6 y_2 y_4.
\end{aligned} \tag{54}$$

The data appear in Tables 8–13.

TABLE 8
Data for Problems 3d1, 3d2 *and* 3d3

μ	t	y_1	y_2	$10y_3$	$10^3 y_4$	$10^3 y_5$
	0 (initial condition)	1.000000	1.000000	0	0	0
1	12.5	0.945757	0.961201	0.494861	0.154976	0.111485
2	25.0	0.926486	0.928762	0.690492	0.314501	0.236263
3	37.5	0.917668	0.915966	0.751806	0.709300	0.311747
4	50.0	0.928987	0.917542	0.771559	1.19224	0.333096
5	62.5	0.927782	0.920075	0.780903	1.68815	0.340324
6	75.0	0.925304	0.912330	0.790539	2.19539	0.356787
7	87.5	0.925083	0.917684	0.783933	2.74211	0.358283
8	100.0	0.917277	0.907529	0.779259	3.20025	0.361969

TABLE 9
Diagonal covariance matrix for Problem 3d1

V_{11}	V_{22}	V_{33}	V_{44}	V_{55}
25×10^{-6}	25×10^{-6}	25×10^{-8}	25×10^{-10}	25×10^{-12}

TABLE 10
Diagonal Q matrix for Problem 3d2

Q_{11}	Q_{22}	Q_{33}	Q_{44}	Q_{55}
100	100	1	0.01	0.0001

TABLE 11
Data for Problems 3f1, 3f2 *and* 3f3

μ	t	y_1	y_2	$10y_3$	10^3y_4	10^3y_5
	0 (initial condition)	1.000000	1.000000	0	0	0
1	12.5	0.951991	0.959213	0.493001	0.169742	0.107904
2	25.0	0.925986	0.933995	0.686153	0.331303	0.236097
3	37.5	0.914657	0.925282	0.747767	0.748909	0.307413
4	50.0	0.926361	0.912188	0.776737	1.15390	0.337807
5	62.5	0.927092	0.912617	0.786070	1.64478	0.346591
6	75.0	0.921343	0.909841	0.793456	2.14506	0.361028
7	87.5	0.924055	0.911927	0.788383	2.69397	0.360954
8	100.0	0.912404	0.912947	0.780271	3.21187	0.359856

TABLE 12
Matrix used as V for Problem 3f1 *and Q for Problem* 3f2

100	−50	5	−0.5	0.05
−50	100	−5	0.5	−0.05
5	−5	1	−0.05	0.005
−0.5	0.5	−0.05	0.01	−0.0005
0.05	−0.05	0.005	−0.0005	0.0001

Problem 4. A second order (see Note 3 below) nonlinear dynamic system. The observed variable is a linear combination (with approximately known coefficients) of the state variables. Observations are available from three runs, each with different initial conditions. Some of the initial conditions and coefficients are treated as unknown parameters with suitable prior distributions. All parameters are required to be positive.

Note 3. The two differential equations are linearly dependent. The system represents a temperature-dependent reversible chemical reaction with known equilibrium constant.

The model equation is:

$$y = 1 + \theta_7 s_1 + \theta_8 s_2, \tag{55}$$

TABLE 13
Parameter information for Problem 3

j	Initial guess θ_{1j}	Lower bound α_j	Upper bound η_j	Penalty coefficient γ_j	Best estimate θ_j^* for Problem:					
					3d1	3d2	3d3	3f1	3f2	3f3
1	0.01	0	1	0.11-5	0.6358106-2	0.6358247-2	0.6345738-2	0.6335020-2	0.6335091-2	0.6591551-2
2	0.01	0	1	0.11-5	0.6774396-1	0.6774551-1	0.6760359-1	0.6742591-1	0.6742680-1	0.7121992-1
3	0.001	0	0.1	0.2-6	0.5916273-4	0.5914274-4	0.5865966-4	0.5504131-4	0.5529208-4	0.4544294-4
4	0.001	0	0.1	0.2-6	0.4943798-3	0.4944010-3	0.4949819-3	0.4980820-3	0.4978131-3	0.5181578-3
5	0.02	0	2	0.21-5	0.1018756	0.1018828	0.1020560	0.1027771	0.1027024	0.1050255
6	0.001	0	0.1	0.2-6	0.4202537-3	0.4210526-3	0.4330268-3	0.4266389-3	0.4229979-3	0.3874024-5

Note. The notation a-b means $a \times 10^{-b}$.

where s_1 and s_2 are solutions of the differential equations

$$\frac{ds_1}{dt} = -\theta_1[s_1 - s_2^2 \exp(-1000/T)] \exp(-\theta_2/T),$$
$$\frac{ds_2}{dt} = -2\frac{ds_1}{dt}. \tag{56}$$

The parameters θ_3, θ_4, θ_5 and θ_6 represent unknown initial conditions as shown in Table 14.

TABLE 14
Run specifications for Problem 4

Run no.	Initial conditions		T
	s_1	s_2	
1	θ_3*	θ_4*	200
2	θ_5*	1.0	400
3	1.0	θ_6*	600

* Inexactly known values.

The log prior distribution of θ has the form

$$\Pi(\theta) = -200[(\theta_3 - 1)^2 + (\theta_4 - 1)^2 + (\theta_5 - 1)^2 + (\theta_6 - 1)^2 + (\theta_7 - 1)^2 + (\theta_8 - 2)^2], \tag{57}$$

that is, θ_3 to θ_8 are normally and independently distributed with standard deviation 0.05 around their means (1, 1, 1, 1, 1, 2).

The data appear in Tables 14–16.

TABLE 15
Data for Problem 4

μ	Run no.	t	y	μ	Run no.	t	y
1	1	0	4.001	16	2	10	4.768
2		10	4.026	17		12	4.879
3		20	4.051	18		14	4.980
4		30	4.075	19		16	5.070
5		40	4.099	20		18	5.149
6		50	4.122	21	3	0	4.000
7		60	4.145	22		0.5	4.143
8		70	4.169	23		1	4.268
9		80	4.194	24		1.5	4.381
10		90	4.216	25		2	4.481
11	2	0	4.000	26		2.5	4.570
12		2	4.186	27		3	4.648
13		4	4.354	28		3.5	4.717
14		6	4.505	29		4	4.777
15		8	4.644	30		4.5	4.830

TABLE 16
Parameter information for Problem 4

j	Initial guess $\theta_{1,j}$	Best estimate θ_j	Lower bound α_j	Upper bound η_j	Penalty coefficient γ_j
1	1.6	1.477448	0	160	0.0001601
2	1400	1493.016	0	140000	0.14
3	1	0.9829972	0	100	0.0001001
4	1	1.007708	0	100	0.0001001
5	1	0.9973492	0	100	0.0001001
6	1	0.9987541	0	100	0.0001001
7	1	1.001341	0	100	0.0001001
8	2	2.001509	0	200	0.0002001

Problem 5. Similar to Problem 4, but with two additional unknown parameters. These appear in a denominator of the reaction rate formula, and represent the effect of heterogeneous catalysis.

The model equation is

$$y \equiv 1 + \theta_9 s_1 + \theta_{10} s_2, \tag{58}$$

where s_1 and s_2 are solutions of the differential equations

$$\begin{aligned} \frac{ds_1}{dt} &= \frac{\theta_1[s_1 - s_2^2 \exp(-1000/T)] \exp(-\theta_2/T)}{[1 + \theta_3 s_1 \exp(-\theta_4/T)]^2}, \\ \frac{ds_2}{dt} &= -2\frac{ds_1}{dt}. \end{aligned} \tag{59}$$

The log prior distribution of θ has the form

$$\begin{aligned} \Pi(\theta) = -200[&(\theta_5 - 1)^2 + (\theta_6 - 1)^2 + (\theta_7 - 1)^2 + (\theta_8 - 1)^2 \\ &+ (\theta_9 - 1)^2 + (\theta_{10} - 2)^2]. \end{aligned} \tag{60}$$

The data appear in Tables 17–19.

TABLE 17
Run specifications for Problem 5

Run no.	Initial conditions		T
	s_1	s_2	
1	θ_5*	θ_6*	200
2	θ_7*	1.0	400
3	1.0	θ_8*	600

* Inexactly known values.

TABLE 18
Data for Problem 5

μ	Run no.	t	y	μ	Run no.	t	y
1	1	0	3.988	16	2	10	2.754
2		10	4.073	17		12	2.896
3		20	4.153	18		14	3.034
4		30	4.231	19		16	3.166
5		40	4.309	20		18	3.278
6		50	4.376	21	3	0	3.012
7		60	4.457	22		0.5	2.956
8		70	4.522	23		1	2.926
9		80	4.615	24		1.5	2.877
10		90	4.667	25		2	2.853
11	2	0	1.997	26		2.5	2.823
12		2	2.149	27		3	2.800
13		4	2.320	28		3.5	2.776
14		6	2.465	29		4	2.767
15		8	2.611	30		4.5	2.760

TABLE 19
Parameter information for Problem 5

j	Initial guess θ_{1j}	Best estimate θ_j^*	Lower bound α_j	Upper bound η_j	Penalty coefficients γ_j
1	2	1.392680	0	200	0.0002001
2	500	1140.023	0	50000	0.0500001
3	0.5	1.820621	0	50	0.0000501
4	50	366.5554	0	5000	0.0050001
5	1	1.00600	0	100	0.0001001
6	1	0.9988526	0	100	0.0001001
7	1	0.9868292	0	100	0.0001001
8	1	1.018978	0	100	0.0001001
9	1	1.010861	0	100	0.0001001
10	2	1.975410	0	200	0.0002001

9. Results. Each one of the ten problems defined in § 8 was run with each one of the thirteen algorithms listed in Table 2. All programs were written in FORTRAN IV, and the calculations were performed in single precision floating point arithmetic on an IBM System/360 Model 50. The following results were obtained for each one of the 130 problem-algorithm combinations:

θ^*: the best estimate of θ;

Φ^*: $= \Phi(\theta^*)$, the final value of the objective function;

n_d: the number of times the objective function and its derivatives were calculated (= number of iterations);

n_f: the number of times the objective function alone was calculated;

n_e: $= (l + 1)n_d + n_f$, the number of equivalent function evaluations (each function and each derivative is counted separately) (l is the number of parameters).

It should be noted that the numbers n_f and n_d reflect a certain redundancy in the calculation: the value θ_{i+1} for which Φ and its gradient are calculated at the start of interation $i + 1$ was one of the trial values for which Φ had already been calculated in the course of the ith iteration. This duplication is unavoidable when the derivatives are obtained from the sensitivity equations in Problems 3–5. Had it been avoided in the other Problems, n_e would have been smaller by $n_d - 1$.

The results are given in Tables 20–29, except for the values of θ^*. The values θ^* corresponding to the highest Φ^* (underlined in the tables) have already been given in the tables of § 8.

For each problem, the algorithms are divided into three classes:

Class I. The results are practically indistinguishable from the best estimates obtained.

Class II. The results are somewhat inferior to the best, but still acceptable.

Class III. The results are unacceptable; the algorithm did not converge.

No rigid criterion for the classification were established, but in no case was there any serious question as to the class to which a given result belonged.

Within classes, the results were listed in order of increasing value of n_e. Ties were broken in favor of the algorithm with the higher Φ^*. For each problem the algorithms appear listed in what we consider the ranking of their performances, from best to worst.

An exception to the ranking rules was made for Problem 3f3. Here we are trying to minimize the value of a nearly singular determinant—a very difficult task. Almost every algorithm ended up with a different result. It was felt that in this problem the only significant test of an algorithm's performance was the value of Φ^* attained, and the rankings were assigned accordingly. The classification criteria were applied rather loosely in this case.

TABLE 20
Results for Problem 1

Algorithm	Class	$-100\Phi^*$	n_d	n_f	n_e	Rank
Gc	I	0.8214876	7	8	36	1
Ga		0.8214865	8	7	39	2
M		0.8214861	9	7	43	3
Ge		0.8214857	12	31	79	4
ROC 2		0.8214876	15	20	80	5
ROC 1		0.8214857	16	20	84	6
IROC b		0.8214880	16	20	84	7
IROC d		0.8214917	16	29	93	8
DFP 4		0.8214902	21	144	228	9
DFP 5		0.8214902	21	148	232	10
DFP 1		0.8214895	43	201	373	11
DFP 2		0.8214876	38	253	405	12
DFP 3		0.8214861	45	311	491	13

TABLE 21
Results for Problem 2

Algorithm	Class	Φ^*	n_g	n_f	n_e	Rank
M	I	185.9896	6	5	41	1
Ga		185.9891	6	5	41	2
Gc		185.9891	6	9	45	3
Ge		185.9891	6	10	46	4
ROC 1		185.9888	22	43	175	5
DFP 1		185.9893	22	116	248	6
IROC b		185.9876	32	65	257	7
DFP 3		185.9888	19	160	274	8
DFP 4		185.9888	20	180	300	9
IROC d		185.9897	36	95	311	10
DFP 2		185.9898	24	185	329	11
ROC 2		185.9844	43	79	337	12
DFP 5		185.9893	22	279	411	13

TABLE 22
Results for Problem 3d1

Algorithm	Class	$-\Phi^*$	n_g	n_f	n_e	Rank
Gc	I	21.37955	9	11	74	1
Ga		21.37953	10	10	80	2
Ge		21.37939	10	31	101	3
M		21.37944	14	16	114	4
ROC 1	II*	21.82013	40	70	350	5
ROC 2		21.82480	58	102	508	6
IROC d		21.77768	57	149	548	7
DFP 1		21.85144	74	266	784	8
DFP 2		21.83237	91	436	1073	9
DFP 3	III	180.8334	30	182	392	10
DFP 4		179.8253	31	227	444	11
DFP 5		90.50511	55	374	759	12
IROC b	†	26.20027	341	963	3009	13

* Algorithms in this class produced erroneous values of θ_6, even though Φ^* is close to optimal.
† Run was discontinued even though termination criterion was not met.

TABLE 23
Results for Problem 3d2

Algorithm	Class	Φ*	n_d	n_f	n_e	Rank
Gc	I	228.9243	9	14	77	1
M		228.9245	12	12	96	2
Ge		228.9241	9	33	96	3
Ga		228.9245	12	13	97	4
IROC d		228.9243	33	61	292	5
ROC 2		228.9243	35	58	303	6
ROC 1		228.9243	36	64	316	7
IROC b		228.9238	45	80	395	8
DFP 3		228.9243	30	223	433	9
DFP 2	II*	228.4888	26	159	341	10
DFP 4		228.4922	31	237	454	11
DFP 5		228.5654	29	284	487	12
DFP 1		228.6682	66	268	730	13

* See note under Table 22.

TABLE 24
Results for Problem 3d3

Algorithm	Class	Φ*	n_d	n_f	n_e	Rank
Gc	I	281.3856	10	26	96	1
Ge		281.2442	12	36	120	2
Ga		281.2435	17	17	136	3
M		281.2435	19	18	151	4
ROC 1		281.2440	41	78	365	5
DFP 1	II*	280.7757	20	83	223	6
ROC 2		280.7354	30	49	259	7
DFP 4		280.7587	20	151	291	8
DFP 2		280.7747	24	140	308	9
DFP 3		280.7752	23	166	327	10
DFP 5		280.7330	22	176	330	11
IROC b		280.7447	40	65	345	12
IROC d		280.7925	43	81	382	13

* See note under Table 22.

TABLE 25
Results for Problem 3f1

Algorithm	Class	$-10^5\Phi^*$	n_g	n_f	n_e	Rank
Ga	I	0.5370535	13	13	104	1
Ge		0.5370480	13	22	113	2
Gc		0.5370543	15	14	119	3
M		0.5370553	15	18	123	4
ROC 2	II*	0.5454540	35	66	311	5
IROC b		0.5451777	39	74	347	6
ROC 1		0.5458247	41	76	363	7
IROC d		0.5589989	45	117	432	8
DFP 3	III	0.8239127	37	254	513	9
DFP 4		0.8207666	34	279	517	10
DFP 2		0.6681885	46	237	559	11
DFP 5		0.8895666	41	311	598	12
DFP 1		0.7418950	92	316	960	13

* See note under Table 22.

TABLE 26
Results for Problem 3f2

Algorithm	Class	Φ^*	n_g	n_f	n_e	Rank
Gc	I	228.8286	9	14	77	1
Ga		228.8286	10	11	81	2
M		228.8288	11	13	90	3
Ge		228.8293	10	40	110	4
ROC 2		228.8288	31	53	270	5
IROC b		228.8286	35	62	307	6
ROC 1		228.8288	38	60	326	7
IROC d		228.8288	35	82	327	8
DFP 5		228.8291	36	302	554	9
DFP 4		228.8286	41	324	611	10
DFP 3		228.8293	44	341	649	11
DFP 1	II*	228.3323	28	109	305	12
DFP 2		228.7036	33	193	424	13

* See note under Table 22.

TABLE 27
Results for Problem 3f3. *Ranking by* Φ^* *alone*

Algorithm	Class	Φ^*	n_d	n_f	n_e	Rank
IROC b	I	338.3492	46	89	411	1
Gc		327.9905	42	83	377	2
DFP 3		325.7381	46	432	774	3
M		325.7183	108	116	872	4
DFP 5		325.1265	34	363	611	5
Ge		322.6341	29	108	311	6
Ga		321.5184	97	103	782	7
ROC 1		318.3492	49	93	436	8
IROC d	II	304.5279	56	105	497	9
DFP 2		304.3690	37	263	522	10
DFP 4	III	287.9332	22	238	392	11
DFP 1		267.7842	64	262	710	12
ROC 2		262.7298	41	66	353	13

TABLE 28
Results for Problem 4

Algorithm	Class	Φ^*	n_d	n_f	n_e	Rank
Ga	I	174.1898	6	4	58	1
M		174.2128	6	7	61	2
Ge		174.1898	6	8	62	3
Gc		174.2076	6	12	66	4
DFP 1		174.2084	28	137	389	5
IROC d		174.2063	34	84	390	6
IROC b		174.0730	41	85	454	7
DFP 2		174.2011	35	251	566	8
DFP 3		174.1979	34	283	589	10
DFP 4		"	"	"	"	10
DFP 5		"	"	"	"	10
ROC 1		174.1429	63	127	694	12
ROC 2	II	171.8576	37	82	415	13

TABLE 29
Results for Problem 5

Algorithm	Class	Φ^*	n_t	n_f	n_r	Rank
Ge	I	107.2412	22	56	298	1
Ga		107.2417	27	37	334	2
Gc		107.2413	32	57	409	3
DFP 5		107.2344	36	323	719	4
M		107.2418	68	100	848	5
DFP 2	II	107.0571	37	241	648	6
IROC d		107.0664	55	130	735	7
DFP 4		106.9853	46	370	876	8
DFP 3		107.0436	44	418	902	9
IROC b		107.0960	113	221	1464	10
ROC 1	III	104.1322	44	84	568	11
DFP 1		104.1608	37	173	580	12
ROC 2		106.7214	86	169	1115	13

10. Discussion and conclusions. The results of § 9 are summarized in Table 30. The ranks of each algorithm in all problems are added together and the algorithms are ranked according to increasing totals. Also tabulated are the number of times that each algorithm has failed to solve a problem (occurrences in Class III), or has given an acceptable, though inferior solution (Class II). These last two numbers give a measure of the algorithm's unreliability.

The algorithms fall naturally into three groups:

A. The modified Gauss and Marquardt methods where, *in parameter estimation problems of the form specified*, an approximation of the Hessian can be computed at each point from values of first derivatives.

B. The variable metric rank one correction methods ROC and IROC. Here, in essence, estimates of the Hessian are built up from finite differences. These methods do not require maximization along a line at each iteration. They apply to all unconstrained optimization problems with smooth objective functions.

C. The various versions of the Davidon–Fletcher–Powell method. They resemble the methods of Group B, except that each iteration requires finding the maximum along a line.

It is evident from Tables 20–30 that the performance of Group A far surpassed that of Groups B and C. In fact, in eight out of ten problems the four Group A methods occupied the first four ranks. Group B slightly out-performed Group C. It is worth noting that the fastest algorithms are also the most reliable ones.

The differences in performance between algorithms within each group are small. One cannot conclude on the basis of these results that modified Gauss is really better than Marquardt, or that ROC 1 is really better than IROC d. It appears that a much more extensive and varied set of test problems is needed to establish firm within-group rankings. It does appear, however, that algorithms

TABLE 30
Summary of algorithm rankings

Group	Overall ranking	Algorithm	Data from Tables 20–29		
			Sum of ranks	Appearance in Class II*	Appearance in Class III†
A	1	Gc	20	0	0
	2	Ga	26	0	0
	3	Ge	32	0	0
	4	M	32‡	0	0
B	5	ROC 1	73	2	1
	6	IROC b	77	3	1
	7	IROC d	81	3	0
	8	ROC 2	85	4	2
C	9	DFP 3	92	2	2
	10	DFP 4	97	3	3
	11	DFP 5	98	2	2
	12	DFP 1	98§	3	4
	13	DFP 2	99	5	2

* Gave acceptable, though inferior solutions.
† Gave unacceptable solutions.
‡ Tie broken in favor of algorithm producing larger values of Φ^* in majority of cases.
§ Tie broken in favor of more reliable algorithm (fewer occurrences in Classes II and III).

which expend a moderate effort on step length determination (Gc, ROC 1, IROC b, DFP 3) are better than algorithms expending no effort at all (Ga, DFP 1) or those expending a lot of effort (Ge, IROC d, DFP 5). This confirms similar observations made by Flanagan, Vitale and Mendelsohn [13].

The superiority of Group B over group C is attributed entirely to the less exacting step length requirements of the former. In determining the best line search strategy one must realize that increasing the accuracy of the search generally reduces the total number of iterations, at the expense of an increased number of function evaluations. Judging the performance of a method depends, therefore, on the relative weight attached to an iteration, which includes gradient evaluations and matrix algebra, as compared to a function evaluation. We have adopted here the equivalent function-evaluations criterion used by Box [4] in his survey of optimization methods, in which each component of the gradient counts as a function evaluation. We feel that this criterion is quite suitable to parameter estimation problems, where the time spent on matrix algebra is negligible compared to that of evaluating the function and its gradient. Nevertheless, in problems with simple model equations and few observations, the DFP and ROC 2 methods may be relatively fast, being the only ones that require no matrix inversion or eigenvalue analysis. Unfortunately, these are precisely the methods which scored the lowest in reliability.

The Group B algorithms offer another advantage over DFP, apart from their better showing in the ratings. Since the approximate Hessian in ROC and IROC

is not restricted to be negative definite, there is a chance that it may contain useful information even in regions where the true Hessian is not definite. For instance, at the DFP solution one cannot appeal to the definiteness of the approximate Hessian for the verification of a maximum or minimum. When penalty functions are used, the DFP method often forces one to move close to the constraints by requiring that one go to the maximum along the search direction. This is not so in ROC or IROC.

In view of the fact that the solutions to all our test problems turned out to be interior maxima, one may ask what purpose did the inequality constraints serve. It turns out that at least in some problems (particularly Problem 3) their presence was essential. Without the lower bounds, some of the parameters would turn negative, and the calculations would bog down in physically infeasible regions of parameter space, where the differential equations could not even be integrated without great numerical difficulties. Yet when the bounds were imposed, convergence to the interior maximum was obtained.

The present survey was limited in its extent, and its conclusions cannot be regarded as final. Many important questions are left entirely unanswered, e.g., the following:

(a) What happens in much larger problems, say $l = 20$ or more? Since each iteration counts as $l + 1$ function evaluations, there may be an advantage to increasing the accuracy of step length determination in order to reduce the number of iterations.

(b) How do the various algorithms perform with finite difference approximations replacing analytic derivatives?

(c) How do direct search methods compare to gradient methods using finite differences?

(d) Do other methods for handling inequality constraints perform better in our type of problem? For instance, each gradient method can be combined with the gradient projection technique (this was done for DFP by Goldfarb and Lapidus [16] and incompletely for the Gauss method by Jennrich and Sampson [20]). We have evidence that this accelerates convergence slightly in some of the test problems.

(e) How do the methods perform in problems where the proposed model does not fit the data well? The Newton-like behavior of the Gauss method depends on the final residuals being small, but this is not so for the variable metric methods.

(f) How do the convergence regions (i.e., the range of initial guesses from which convergence is obtained) of the various algorithms compare?

We hope to provide some of the answers in future work.

Acknowledgment. The author wishes to thank Dr. John Greenstadt of IBM for many valuable suggestions.

REFERENCES

[1] Y. BARD, *Nonlinear estimation and programming*, Computer Program 360D-13.6.003, IBM, Hawthorne, New York, 1967.

[2] ———, *On a numerical instability of Davidon-like methods*, Math. Comp., 22 (1968), pp. 665–666.

[3] G. W. BOOTH AND T. I. PETERSON, *Nonlinear estimation*, IBM Share Program No. 687 WL NLI, New York, 1958.

[4] M. J. BOX, *A comparison of several current optimization methods, and the use of transformations in constrained optimization*, Comput. J., 9 (1966), pp. 67–77.

[5] C. G. BROYDEN, *Quasi-Newton methods and their application to function minimization*, Math. Comp., 21 (1967), pp. 368–381.

[6] C. W. CARROLL, *The created response surface technique for optimizing nonlinear, restrained systems*, Operations Res., 9 (1961), pp. 169–184.

[7] J. B. CROCKETT AND H. CHERNOFF, *Gradient methods of maximization*, Pacific J. Math., 5 (1955), pp. 33–50.

[8] W. C. DAVIDON, *Variable metric method for minimization*, A.E.C. Research and Development Rep. ANL-5990, Argonne National Laboratories, Argonne, Illinois, 1959.

[9] ———, *Variance algorithm for minimization*, Comput. J., 10 (1968), pp. 406–410.

[10] H. EISENPRESS AND J. GREENSTADT, *The estimation of nonlinear econometric systems*, Econometrica, 34 (1966), pp. 851–861.

[11] R. H. FARISS AND V. J. LAW, *Practical Tactics for Overcoming Difficulties in Non-Linear Regression and Equation Solving*, A.I.Ch.E. Meeting, Houston, Texas, 1967.

[12] A. V. FIACCO AND G. P. MCCORMICK, *Non-Linear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, 1968.

[13] P. D. FLANAGAN, P. A. VITALE AND J. MENDELSOHN, *A numerical investigation of several one-dimensional search procedures in nonlinear regression problems*, Technometrics, 11 (1969), pp. 265–284.

[14] R. FLETCHER AND M. J. D. POWELL, *A rapidly convergent descent method for minimization*, Comput. J., 6 (1963), pp. 163–168.

[15] K. F. GAUSS, *Theoria Motus Corporum Coelistiam*, Werke, vol. 7, 1809, pp. 240–254.

[16] D. GOLDFARB AND L. LAPIDUS, *A conjugate gradient method for nonlinear programming problems with linear constraints*, Indust. Engrg. Chem. Fundamentals, 7 (1968), pp. 142–151.

[17] S. M. GOLDFELD, R. E. QUANDT AND H. F. TROTTER, *Maximization by quadratic hill-climbing*, Econometrica, 34 (1966), pp. 541–551.

[18] J. GREENSTADT, *On the relative efficiencies of gradient methods*, Math. Comp., 21 (1967), pp. 360–367.

[19] H. O. HARTLEY, *The modified Gauss–Newton method for the fitting of nonlinear regression functions by least squares*, Technometrics, 3 (1961), pp. 269–280.

[20] R. I. JENNRICH AND P. F. SAMPSON, *Application of stepwise regression to non-linear estimation*, Ibid., 10 (1968), pp. 63–72.

[21] K. LEVENBERG, *A method for the solution of certain nonlinear problems in least squares*, Quart. Appl. Math., 2 (1944), pp. 164–168.

[22] D. W. MARQUARDT, *An algorithm for least squares estimation of nonlinear parameters*, this Journal, 11 (1963), pp. 431–441.

[23] R. TOMOVIC, *Sensitivity Analysis of Dynamic Systems*, McGraw-Hill, New York, 1963.