MATHEMATICS OF COMPUTATION, VOLUME 33, NUMBER 145
JANUARY 1979, PAGES 249–264

A Bench Mark Experiment for Minimization Algorithms*

By J. N. Lyness

Abstract. In this paper we suggest a single bench mark problem family for use in evaluating unconstrained minimization algorithms or routines. In essence, this problem consists of measuring, for each algorithm, the rate at which it descends an unlimited helical valley. The periodic nature of the problem allows us to exploit affine scale invariance properties of the algorithm. As a result, the capacity of the algorithm to minimize a wide range of helical valleys of various scales may be summarized by calculating a single valued function $g_Q(X_1)$. The measurement of this function is not difficult, and the result provides information of a simple, general character for use in decisions about choice of algorithm.

1. Introduction. At the present time, considerable effort is being expended on constructing numerical software for unconstrained minimization. In order to allow an orderly development of this area, it is necessary to compare the performance of different algorithms which carry out the same task using information of a similar nature. In general, a theoretical comparison is not feasible, and one must resort to comparison by numerical experiment.

There are many difficulties which present themselves when one comes to construct such an experiment. One of the first is simply that there is such a wide choice of calculations which could be performed. For any of the large number of potentially interesting objective functions (many of which have n-dimensional versions for all values of n) one has to assign a set of initial parameters. The consequent trajectory for the same objective function is different for each different assignment of these parameters. And from each run a great deal of information may be obtained. It is only too easy to generate an enormous amount of information about the behavior of routines. The investigator then faces a daunting task in processing this in some coherent manner. Moreover, in many problems the details of the trajectory are unstable with respect to the initial parameters and parameters defining the objective function. A minor perturbation (of machine accuracy magnitude) in one such parameter may result in a completely different trajectory.

Naturally, there are other difficulties too. But it is to the alleviation of these particular ones that the present article is addressed.

Our approach is based on three underlying themes.

Received January 4, 1978.

AMS (MOS) subject classifications (1970). Primary 60K05, 90C30.

Key words and phrases. Numerical software evaluation, affine scale invariance, minimization algorithms, optimization algorithms.

*Work performed under the auspices of the U. S. Department of Energy.

First, we should glean information about one topographical feature at a time. We deal in this paper only with classes of helical valleys.

Second, we should choose quantities to measure which are not unstable with respect to minor parameter perturbation. An individual trajectory is not such a quantity. Quantities such as the percentiles y_Q of the cost distribution functions φ_δ defined in Section 2 are suitable, being mathematically defined and relatively easy to measure.

Third, we require that the numerical results, while covering a relatively wide class of problems, may be expressed or summarized in a reasonably concise form. This may be accomplished to some extent by recognizing the affine scale invariance properties of the algorithms under consideration (described briefly in Section 3) and constructing an objective function family for which these properties may be exploited to reduce the number of parameters on which φ_δ depends. The calculation in Sections 4 and 5 is devoted to this.

In Section 6, we present some numerical results for four well-known algorithms. The author feels that these, which may be displayed on a single graph, provide simple unambiguous information about the relative performance of the algorithms in a three-dimensional helical valley topography.

2. **Cost Distribution Functions.** The underlying ideas, on which the theoretical framework described in this section is based, may be applied to almost every type of minimization algorithm. The description given in this section is in the context of unconstrained minimization. This section is quite independent of the concepts of affine scale invariance of algorithms (Lyness 1979). In order to provide a reasonably concise description, we assume that the algorithm has the following features. It proceeds by making a sequence of function calls which provides information (of an identical nature) about the objective function. In order to start, it requires the assignment of elements of a parameter list, called Π in Lyness (1979). In this paper, we assume that one of these parameters is $x^{(0)}$ a starting iterate and use Π to denote the other elements of the parameter list.

The parameter list contains an element N, which assures termination in N iterations or less. Let us suppose that in a run in which $x^{(0)}$, Π and $f(x)$ are specified, the algorithm proceeds, calculating interates

$$x^{(0)}, x^{(1)}, \ldots, x^{(\tilde{N})}, \qquad \tilde{N} \leqslant N. \tag{2.1a}$$

Our key assumption is that if we make another almost identical run, specifying $x^{(0)}$, Π' and $f(x)$, where Π' differs from Π only in that N' replaces N, one obtains iterates

$$x^{(0)}, x^{(1)}, \ldots, x^{(\tilde{N}')}, \qquad \tilde{N}' \leqslant N', \tag{2.1b}$$

where an individual iterate $x^{(j)}$ if it occurs in both (2.1a) and (2.1b) is identical; moreover, when $N' = N$, $\tilde{N}' = \tilde{N}$. This states first that the trajectory is determinate, and second that altering N may extend or curtail this trajectory but not alter it in any other respect. Associated with each iterate is a function value $f(x^{(j)}) = h^{(j)}$. To simplify our analysis, let us assume that the function values at successive iterates form a monotonic decreasing sequence, that is

$$h^{(0)} > h^{(1)} > h^{(2)} > \cdots > h^{(N)}. \tag{2.2}$$

Also associated with each iterate $x^{(j)}$ is $\nu^{(j)}$ the number of function calls made by the algorithm including those required before and during the jth iteration. Naturally these form a monotonic increasing sequence, starting with $\nu^{(0)} = 1$. Thus,

$$\nu^{(0)} < \nu^{(1)} < \nu^{(2)} < \cdots < \nu^{(N)}. \tag{2.3}$$

A basic cost function $\nu(h, x^{(0)}, \Pi, f)$ is a continuous function of h which fills in or approximates in some well-defined manner the discrete valued function defined by (2.2) and (2.3). This may be defined in various ways, depending on the specific application in mind.

Definition 2.4. An *iterate based piecewise linear basic cost function* $\nu(h, x^{(0)}, \Pi, f)$ is a continuous function of h satisfying

$$\begin{aligned} \nu(h, x^{(0)}, \Pi, f) &= 0 \;\; \forall\, h > h^{(0)} = f(x^{(0)}), \\ \nu(h^{(j)}, x^{(0)}, \Pi, f) &= \nu^{(j)}, \qquad j = 0, 1, \ldots, \end{aligned} \tag{2.4}$$

$$\frac{\partial \nu}{\partial h} = \frac{\nu^{(j+1)} - \nu^{(j)}}{h^{(j+1)} - h^{(j)}} \qquad \forall\, h \in (h^{(j+1)}, h^{(j)}). \tag{2.5}$$

When $f(x)$ has a global minimum $x_{\min}$, we define

$$\nu(h, x^{(0)}, \Pi, f) = \infty \qquad \forall\, h < f(x_{\min}).$$

If, for all finite values of N, the algorithm terminates, or cycles and does not obtain a function value smaller than $h'_{\min}$, then

$$\nu(h, x^{(0)}, \Pi, f) = \infty \qquad \forall\, h < h'_{\min}.$$

We shall be particularly interested in the derivative of this function with respect to h. To this end we define

$$\dot\nu_\delta(h, x^{(0)}, \Pi, f) = (\nu(h + \delta, x^{(0)}, \Pi, f) - \nu(h - \delta, x_0, \Pi, f))/2\delta \tag{2.6}$$

and

$$\dot\nu(h, x^{(0)}, \Pi, f) = \lim_{\delta \to 0} \dot\nu_\delta(h, x^{(0)}, \Pi, f). \tag{2.7}$$

In view of (2.2) and (2.3) these take negative values (except where they may be indeterminate).

A function such as ν defined above, while it may be of interest, suffers from two principal drawbacks. First it is somewhat erratic. Second, minor perturbations in parameters such as $x^{(0)}$ or those contained in Π may lead to an entirely different trajectory, though usually one of the same general nature. For this reason instead of treating cost functions directly we treat the distributions to which they give rise.

We define a finite region R of configuration space and let $x^{(0)}$ be a variate, uniformly distributed in $\mathcal{R}$. (While we could do the same with respect to the other parameters contained in Π, this would be unnecessary in the application we shall consider.) When this is done, the values of $\dot{\nu}_\delta(h, x^{(0)}, \Pi, f)$ for fixed h, Π, f form a distribution function, defined below. In order to deal with cases where algorithm failure is possible, i.e., for some values of x_0 within $\mathcal{R}$, a premature termination occurs before minimization has occurred, we define $\mathcal{R}_h(\mathcal{R}, \Pi, f)$ as follows.

Definition 2.8.

$$\mathcal{R}_h(\mathcal{R}, \Pi, f) = \{x^{(0)} \text{ such that } x^{(0)} \in \mathcal{R} \text{ and } \nu(h, x^{(0)}, \Pi, f) < \infty\}. \tag{2.8}$$

If one is prepared to assume that no premature termination occurs, one may set

$$\mathcal{R}_h(\mathcal{R}, \Pi, f) = \mathcal{R}. \tag{2.9}$$

We now define distribution functions.

Definition 2.10.

$$\varphi_\delta(y;\, h,\, \mathcal{R}, \Pi, f) = \int_{\mathcal{R}_{h+\delta}} H(y + \dot{\nu}_\delta(h, x^{(0)}, \Pi, f))\, dx^{(0)} \bigg/ \int_{\mathcal{R}_{h+\delta}} dx^{(0)} \tag{2.10}$$

where

$$\mathcal{R}_{h+\delta} = \mathcal{R}_{h+\delta}(\mathcal{R}, \Pi, f) \tag{2.11}$$

and

$$H(t) = \begin{cases} 1, & t > 0, \\ ½, & t = 0, \\ 0, & t < 0, \end{cases} \tag{2.12}$$

is the unit step function (or Heaviside function). Like all distribution functions, $\varphi_\delta(y)$ is a monotonic nondecreasing function of y, taking values in the interval $[0, 1]$. Unless $\varphi_\delta(t)$ is discontinuous at $t = y$, the function $\varphi_\delta(y)$ is the probability that $-\dot{\nu}_\delta$ is less than y. (In general, $\varphi_\delta(y)$ is the average of the probability that $-\dot{\nu}_\delta$ is less than y and the probability that $-\dot{\nu}_\delta$ is less than or equal to y.) The value $y_{0.5}$ of y for which $\varphi_\delta(y) = ½$ is the median value of $-\dot{\nu}_\delta$.

In the following discussion we restrict our attention to values of h satisfying

$$h < \min_{x \in \mathcal{R}} f(x) \tag{2.13}$$

so that $-\dot{\nu}$ is positive. From this it follows that $\varphi_\delta(0) = 0$ and it also follows from the definition that (for zero δ), $\varphi(\infty) = 1$. However, it is the quantity $\dot{\nu}_\delta$ with finite δ which is measured experimentally and in some respects, this measurement is more useful as it allows a failure probability to be calculated. From Definition 2.8 it follows that

$$\mathcal{R} \supseteq \mathcal{R}_{h+\delta} \supseteq \mathcal{R}_{h-\delta}. \tag{2.14}$$

Thus, when $\mathcal{R}_{h+\delta} \not\equiv \mathcal{R}_{h-\delta}$ there are values of $x^{(0)} \in \mathcal{R}$ for which $\nu(h + \delta, x^{(0)}, \Pi, f)$ is finite, but $\nu(h - \delta, x^{(0)}, \Pi, f)$ is infinite. For these values of $x^{(0)}$ the algorithm terminates prematurely, returning a function value between $h - \delta$ and $h + \delta$.

Examination of definitions (2.6) and (2.10) shows that the integrand function in (2.10) (the unit step function) is zero for all finite y for such a value of $x^{(0)}$. This gives

$$\varphi_\delta(\infty) = \int_{R_{h-\delta}} dx \Big/ \int_{R_{h+\delta}} dx, \tag{2.15}$$

which may be less than 1. Thus, a knowledge of the distribution function $\varphi_\delta(y)$ provides usable information. The value of $(1 - \varphi_\delta(\infty))$ indicates a failure probability. So long as this is small, we can still use the median $y_{0.5}$ or other properties of $\varphi_\delta(y)$ for evaluation purposes. There is no need either to introduce arbitrary penalties for failure, or to unfairly ignore algorithm failure.

In the next section we shall apply the definitions presented in this section to a particular "bench mark" problem. The details of the above definitions were introduced with this particular problem in mind. However, the author hopes that the same sort of definitions will prove helpful for wider classes of problems. However, in other problems different definitions of the same general nature may be more appropriate. There is no *need* to use a measure such as $\dot{\nu}$, the number of function calls per unit drop in function value. One could use measures related to the distance from the minimum. Different types of function calls occurring in the same algorithm simply introduce different cost elements and each of these can be measured. Again, even if one decides to use a basic cost function, it need not be defined precisely as in Definition 2.4. And both this and any other definition may be modified to avoid the necessity of inequalities (2.2). However, the essence of this approach is that one is measuring quantities which are properly defined, which are insensitive to minor parameter variation and which can be measured however unexpectedly a particular algorithm behaves. The median $y_{0.5}$ of this distribution $\varphi_\delta(y)$ is such a quantity.

3. Affine Scale Invariance Properties. In a companion paper (Lyness (1979)) we discussed the concept of T-scale invariance of algorithms, where T stands for a group of affine transformations. An element $t(k, m, A, d) \in T$ is specified by assigning values to k, $m > 0$, A an $n \times n$ nonsingular matrix and d an $n \times 1$ vector. The transformation t applied to an objective function $f(x)$ gives

$$\bar{f} = tf, \tag{3.1}$$

where

$$\bar{f}(x) = k + mf(Ax + d). \tag{3.2}$$

An algorithm is T-scale invariant under the following circumstances. When applied to $f(x)$ with parameters Π the algorithm calculates iterates

$$x^{(0)}, x^{(1)}, \ldots, x^{(N)}. \tag{3.3}$$

Then, for all $t \in T$, it is possible to assign parameters $\bar{\Pi}$ (which depend on t and Π but not on f) so that when applied to $\bar{f}(x)$, the algorithm calculates iterates

$$\bar{x}^{(0)}, \bar{x}^{(1)}, \ldots, \bar{x}^{(N)}, \tag{3.4}$$

which satisfy

$$\bar{x}^{(j)} = A^{-1}(x^{(j)} - d); \quad \bar{f}(\bar{x}^{(j)}) = k + mf(x^{(j)}). \tag{3.5}$$

In Lyness (1979) we defined various groups T with respect to which an algorithm might be affinely scale invariant. In particular, we define the *full* transformation group

$$T_F^{(n)} = \{t(k, m, A, d) \; \forall \; k, m > 0, d, \text{ nonsingular } A\}. \tag{3.6}$$

We showed that many standard versions of the quasi Newton algorithm are fully scale invariant, i.e., scale invariant with respect to $T_F^{(n)}$. However, we showed that conjugate direction algorithms are not fully scale invariant. They are usually scale invariant to a group $T_G^{(n)}$ defined as

$$\begin{aligned} T_G^{(n)} = \{t(k, m, A, d) \; \forall \; k, m > 0, d \text{ and for all} \\ A \text{ satisfying } AA^T = \lambda^2 I \; (\lambda \neq 0)\}. \end{aligned} \tag{3.7}$$

When two functions $f(x)$ and $\bar{f}(x)$ are related by (3.2) and the algorithm is scale invariant, there are consequent relations between the functions ν constructed for $f(x)$ and $\bar{f}(x)$. It follows from (3.5) without difficulty that we have

THEOREM 3.8. *When the algorithm is scale invariant under $t(k, m, A, d)$*

$$\nu(h, x_0, \Pi, f) = \nu(k + mh, A^{-1}(x^{(0)} - d), \bar{\Pi}, \bar{f}). \tag{3.8}$$

This equation merely implies that if one has taken the trouble to evaluate the ν-function for a particular problem, and the algorithm is scale invariant, the result applies also in scaled form to a scaled version of the original problem.

From (3.8) it follows immediately that both $\dot{\nu}_\delta$ and $\dot{\nu}$ satisfy

$$\dot{\nu}_\delta(h, x^{(0)}, \Pi, f) = m\dot{\nu}_{m\delta}(k + mh, A^{-1}(x^{(0)} - d), \bar{\Pi}, \bar{f}). \tag{3.9}$$

To define the corresponding relationship between cost distribution functions, we define a region $\bar{\mathcal{R}}$, denoted by $t\mathcal{R}$, in terms of $\mathcal{R}$ by

$$x \in \mathcal{R} \Longleftrightarrow A^{-1}(x - d) \in \bar{\mathcal{R}} = t\mathcal{R}; \tag{3.10}$$

and we find

$$\varphi_\delta(y; h, \mathcal{R}, \Pi, f) = \varphi_{m\delta}(m^{-1}y; k + mh, \bar{\mathcal{R}}, \bar{\Pi}, \bar{f}). \tag{3.11}$$

The foregoing relations are all derived from (3.5), and require no detailed knowledge about the construction of $\bar{\Pi}$. The circumstance that (3.5) is valid presumes that $\bar{\Pi}$ can be constructed. In practice this is one of the difficult aspects of establishing scale invariance. In the rest of this paper, we shall assume that the parameter list includes $x^{(0)}$, $\Gamma^{(0)}$, $\Delta f^{(0)}$, N, where $x^{(0)}$ is the initial iterate $\Gamma^{(0)}$ an approximation (possibly unrealistic) to the inverse Hessian at $x^{(0)}$, $\Delta f^{(0)}$ an estimate (again possibly arbitrary) to the expected reduction $h^{(0)} - h^{(1)}$ during the first iteration and N a limit on the number of function calls. The associated parameter list then includes $\bar{x}^{(0)}$, $\bar{\Gamma}^{(0)}$, $\overline{\Delta f}^{(0)}$ and $\bar{N}$ related by

$$\bar{x}^{(0)} = tx^{(0)} = A^{-1}(x^{(0)} - d), \tag{3.12}$$

$$\bar{\Gamma}^{(0)} = t\Gamma^{(0)} = m^{-1}(A^{-1})^T\Gamma^{(0)}A^{-1}, \tag{3.13}$$

$$\overline{\Delta f}^{(0)} = t\Delta f^{(0)} = m\Delta f^{(0)}, \tag{3.14}$$

$$\bar{N} = N. \tag{3.15}$$

4. A Family of Helical Valley Objective Functions. In this section we shall describe a single numerical experiment of a somewhat extensive nature. The purpose of this experiment is to quantify how well or badly a single algorithm handles a particular curved valley.

We consider an objective function defined in detail by (4.2), (6.2) or (6.3) below. Each of these is of the form

$$f(x, y, z) = F_H(x, y, z; T, R, P) + Mz, \tag{4.1}$$

where T, R, P and M are positive parameters. The locus $F_H(x, y, z; T, R, P) = 0$ is a circular helix, passing through $(R, 0, 0)$, described on the cylinder $x^2 + y^2 = R^2$ having pitch (or period) P. The function $F_H(x, y, z; T, R, P)$ is positive at all points (x, y, z) not on the helix and generally increases in value, at a rate depending on T, as the shortest distance from (x, y, z) to the helix increases.

The objective function (4.1) has no global minimum, its value on the helix decreasing steadily with decreasing z. The trajectory followed by a minimization algorithm presented with such an objective function, might be expected to follow this helical path in a rough sort of way.

The reader may find it helpful to visualize the following mechanical system. A light bead is threaded onto a helical wire having vertical axis. A heavier bead is attached to the light bead by an elastic string. There is friction present. When the system is released the subsequent motion of the heavier bead resembles to some extent the "trajectory" of the iterates in the minimization problem. This analogy should not be taken too far. Energy or angular momentum conservation laws do not usually apply to minimization algorithms.

However, the mechanical system and the behavior of the algorithm do have some features in common. The detailed behavior in each case would be expected to be rather spasmodic and one would expect to be able to define for each an overall or average ultimate rate of descent.

We now specify an objective function family of type (4.1) and provide parameters to specify the input parameters required to define a cost distribution function φ_δ of Definition 2.10.

We define an objective function

$$f(x, y, z; T, R, P, M)$$

$$(4.2) \qquad = \frac{T^2}{\sqrt{x^2 + y^2}} \left[\left(x - R \cos \frac{2\pi z}{P}\right)^2 + \left(y - R \sin \frac{2\pi z}{P}\right)^2\right] + Mz$$

and refer to parameters T, R, P, M as problem family parameters. We define a region $\mathcal{R}$ by

$$(4.3) \qquad \mathcal{R}: \ \rho_1 R < \sqrt{x^2 + y^2} < \rho_2 R; \qquad z = \rho_3 P,$$

an initial approximation to the inverse Hessian by

$$(4.4) \qquad \Gamma^{(0)} = \operatorname{diag}\{\gamma_1 R/T^2, \gamma_1 R/T^2, \gamma_3 P/M\},$$

and an initial estimate of the expected improvement by

$$(4.5) \qquad \Delta f^{(0)} = \gamma_4 T^2 R.$$

We refer to $\rho_1, \rho_2, \rho_3, \gamma_1, \gamma_3$ and γ_4 as secondary parameters.

Definition 4.6. The distribution function

$$(4.6) \qquad \psi_\delta(y; \ h, T, R, P, M, \rho_1, \rho_2, \rho_3, \gamma_1, \gamma_3, \gamma_4)$$

is the function $\varphi_\delta(y; h, \mathcal{R}, \Pi, f)$ defined by 2.10 when $\mathcal{R}$, Π, and f are replaced by the parameter defined quantities in (4.2) to (4.5) above.

An assignment of $\mathcal{R}$, $\Gamma^{(0)}$ and $\Delta f^{(0)}$ of this general nature is crucial to the following analysis. However, it is not unreasonable. When $\rho_3 > 0$, the starting region is a disc symmetrically arranged above the helical valley. The approximation to the inverse hessian is diagonal, the elements being dimensionally correct, and the guess for the initial reduction is also dimensionally correct.

Our relative evaluation procedure will be based on comparing this twelve-argument function, evaluated using one algorithm, with the same function evaluated using the other algorithm. If it were expected that ψ_δ would depend significantly and independently on all these arguments, it would be hopless to seriously attempt such a comparison.

We shall be treating the situation in which $f(x, y, z)$ is positive for $f \in \mathcal{R}$ and for which h is negative. Since the secondary parameters normally affect directly only the beginning of the iteration, we can reasonably expect that ψ_δ will be to a significant extent independent of these parameters.

At this point we have merely defined a twelve-parameter cost distribution function. We now assume that the algorithm is affinely scale invariant either with respect to T_F or T_G, defined in (3.6) and (3.7) above. The rest of this section is devoted to exploiting this scale invariance in order to obtain information about ψ_δ. This information is in the form of functional equations.

LEMMA 4.7. *If the algorithm is scale invariant under* $t(0, m, I, 0)$,

$$(4.7) \quad \psi_\delta(y; \ h, T, R, P, M, \vec{\rho}, \vec{\gamma}) = \psi_{m\delta}(m^{-1}y; \ mh, \sqrt{m}T, R, P, mM, \vec{\rho}, \vec{\gamma}).$$

If the algorithm is scale invariant under $t(0, 1, \lambda I, 0)$,

$$\psi_\delta(y;\ h,\ T,\ R,\ P,\ M,\ \vec{\rho},\ \vec{\gamma}) = \psi_\delta(y;\ h,\ \sqrt{\lambda T},\ R/\lambda,\ P/\lambda,\ \lambda M,\ \vec{\rho},\ \vec{\gamma}). \tag{4.8}$$

If the algorithm is scale invariant under $t(k, 1, A, d)$ *where* k, A *and* d *are defined in terms of* l *by*

$$d = \begin{pmatrix} 0 \\ 0 \\ -l \end{pmatrix}, \quad A = \begin{pmatrix} \cos\theta, & \sin\theta, & 0 \\ -\sin\theta, & \cos\theta, & 0 \\ 0\ , & 0\ , & 1 \end{pmatrix}, \quad \theta = \frac{2\pi l}{P}, \quad k = Ml, \tag{4.9a}$$

then

$$\begin{aligned} &\psi_\delta(y;\ h,\ T,\ R,\ P,\ M,\ \rho_1,\ \rho_2,\ \rho_3,\ \vec{\gamma}) \\ &\qquad = \psi_\delta(y;\ h + Ml,\ T,\ R,\ P,\ M,\ \rho_1,\ \rho_2,\ \rho_3 + l/P,\ \vec{\gamma}). \end{aligned} \tag{4.9}$$

If the algorithm is scale invariant under $t(0, 1, A, 0)$ *where* $A = \operatorname{diag}\{1, 1, \lambda\}$, *then*

$$\psi_\delta(y; h,\ T,\ R,\ P,\ M,\ \vec{\rho},\ \vec{\gamma}) = \psi_\delta(y; h,\ T,\ R,\ P/\lambda,\ \lambda M,\ \vec{\rho},\ \vec{\gamma}). \tag{4.10}$$

These transformations have the following property in common. The function $\bar{f} = tf$ *is a member of the same problem family, having different principal parameters. In addition, the entities* $\bar{\mathcal{R}} = t\mathcal{R}$, $\bar{h}^{(0)} = th^{(0)}$ *and* $\overline{\Delta f}^{(0)} = t\Delta f^{(0)}$ *are, respectively, the same function of these different principal parameters as the unbarred entities are of the original principal parameters.*

We give a detailed proof of the second result (4.8) only. The others are proved in the same way.

When

$$t = (0,\ 1,\ \lambda I,\ 0) \tag{4.11}$$

we may use the relations of Section 3 with

$$k = 0, \quad m = 1, \quad d = 0, \quad A = \lambda I. \tag{4.11a}$$

From Definition 4.6 and Equation (3.11) we may write

$$\psi_\delta(y; h,\ T,\ R,\ P,\ M,\ \vec{\rho},\ \vec{\gamma}) = \varphi_\delta(y; h,\ \mathcal{R},\ \Pi,\ f) = \varphi_\delta(y; h,\ \bar{\mathcal{R}},\ \bar{\Pi},\ \bar{f}) \tag{4.12}$$

where f, $\mathcal{R}$, $\Gamma^{(0)}$ and $\Delta f^{(0)}$ are defined by (4.2) to (4.5) and $\bar{f} = tf$, $\bar{\mathcal{R}} = t\mathcal{R}$, $\bar{\Gamma}^{(0)} = t\Gamma^{(0)}$ and $\overline{\Delta f}^{(0)} = t\Delta f^{(0)}$ are defined in accordance with (3.2), (3.10), (3.13) and (3.14), respectively. Direct substitution in (4.2) gives

$$\begin{aligned} \bar{f}(x, y, z) &= tf = f(\lambda x,\ \lambda y,\ \lambda z,\ T,\ R,\ P,\ M) \\ &= \frac{T^2\lambda}{\sqrt{x^2 + y^2}} \left(\left(x - \frac{R}{\lambda} \cos \frac{2\pi z}{P/\lambda} \right)^2 + \left(y - \frac{R}{\lambda} \sin \frac{2\pi z}{P/\lambda} \right)^2 \right) + \lambda M z. \end{aligned} \tag{4.13}$$

This may be written in the form

$$\bar f(x, y, z) = f(x, y, z, \bar T, \bar R, \bar P, \bar M), \tag{4.14}$$

where

$$\bar T^2 = \lambda T^2, \quad \bar R = R/\lambda, \quad \bar P = P/\lambda, \quad \bar M = \lambda M. \tag{4.15}$$

In words, $\bar f$ is the same function of the barred variables (4.15) as f is of the corresponding unbarred variables. Simple manipulation using successively (3.10), (3.13) and (3.14) together with (4.15) leads to a similar statement with respect to $\bar{\mathcal{R}}$, $\bar\Gamma^{(0)}$ and $\overline{\Delta f}^{(0)}$ being valid. Specifically, we find

$$\bar{\mathcal{R}}\colon\ \rho_1\bar R < \sqrt{x^2 + y^2} < \rho_2\bar R; \quad z = \rho_3\bar P, \tag{4.16}$$

$$\bar\Gamma^{(0)} = t\Gamma^{(0)} = \lambda^{-2}\Gamma^{(0)} = \operatorname{diag}\{\gamma_1\bar R/\bar T^2, \gamma_1\bar R/\bar T^2, \gamma_3\bar P/\bar M\}, \tag{4.17}$$

$$\overline{\Delta f}^{(0)} = t\Delta f^{(0)} = \Delta f^{(0)} = \gamma_4\bar T^2\bar R. \tag{4.18}$$

We note that $\bar f$, $\bar{\mathcal{R}}$, $\bar\Gamma^{(0)}$ and $\overline{\Delta f}^{(0)}$ given by (4.14)–(4.28) are, respectively, the same functions of $\bar T, \bar R, \bar P, \bar M, \vec\rho, \vec\gamma$ as f, $\mathcal{R}$, $h^{(0)}$ and $\Delta f^{(0)}$ given by (4.2)–(4.5) are of $T, R, P, M, \vec\rho, \vec\gamma$. Consequently according to Definition 4.6

$$\varphi_\delta(y; h, \bar{\mathcal{R}}, \bar\Pi, \bar f) = \psi_\delta(y; h, \bar T, \bar R, \bar P, \bar M, \vec\rho, \vec\gamma). \tag{4.19}$$

The result (4.8) in the theorem then follows immediately from (4.12) when the barred parameters in (4.19) are replaced by their values given in (4.15).

The proof of the other three parts of Lemma 4.7 is virtually identical. In fact, the same text may be used, making the appropriate alterations in displayed equations (4.11)–(4.19).

When a function, such as ψ_δ above, of several specified independent variables satisfies a functional relationship, it is often possible to express it as a different function of fewer different independent variables. For example, in (4.10), it is clear that altering P and M in such a way that the product PM remains constant does not affect the value of ψ_δ. If we had used $X_3 = P$ and $X_4 = MP$ as independent variables instead of P and M, the purport of (4.10) is that the new function is independent of X_3 To provide a straightforward treatment of this and the other relations, we introduce new variables as follows:

Definition 4.20.

$$\theta_D(Y; H, X_1, X_2, X_3, X_4, \vec\rho, \vec\gamma) = \psi_\delta(y; h, T, R, P, M, \vec\rho, \vec\gamma), \tag{4.21}$$

where

$$\begin{gathered} D = \delta/MP; \quad Y = MPy; \quad H = h/MP, \\ X_1 = T^2R/MP; \quad X_2 = R/P; \quad X_3 = P; \quad X_4 = MP. \end{gathered} \tag{4.22}$$

Rewriting (4.7)–(4.10) in terms of the function $\bar\theta_D$ gives, respectively:

$$\theta_D(Y, H, X_1, X_2, X_3, X_4, \vec{\rho}, \vec{\gamma}) = \theta_D(Y, H, X_1, X_2, X_3, mX_4, \vec{\rho}, \vec{\gamma}), \tag{4.23}$$

$$\theta_D(Y, H, X_1, X_2, X_3, X_4, \vec{\rho}, \vec{\gamma}) = \theta_D(Y, H, X_1, X_2, X_3/\lambda, X_4, \vec{\rho}, \vec{\gamma}), \tag{4.24}$$

$$\begin{aligned} &\theta_D(Y, H, X_1, X_2, X_3, X_4, \vec{\rho}, \vec{\gamma}) \\ &\qquad = \theta_D(Y, H + l/P, X_1, X_2, X_3, \rho_1, \rho_2, \rho_3 + l/P, \vec{\gamma}), \end{aligned} \tag{4.25}$$

$$\theta_D(Y, H, X_1, X_2, X_3, X_4, \vec{\rho}, \vec{\gamma}) = \theta_D(Y, H, X_1, \lambda X_2, X_3/\lambda, X_4, \vec{\rho}, \vec{\gamma}). \tag{4.26}$$

THEOREM 4.27. *When the algorithm is fully scale invariant, the distribution function ψ_δ may be reexpressed as a function θ_D which is independent of X_2, X_3, and X_4.*

Proof. Such an algorithm has a distribution function ψ_δ which satisfies (4.23), (4.24), and (4.26) above. The first two indicate that θ_D is independent of X_4 and X_3, respectively. When θ_D is independent of X_3, the third shows that it is independent of X_2.

When the algorithm satisfies only the more limited T_G-scale invariance, θ_D is independent of X_3 and X_4 but may depend on X_2.

5. The Nature of a Limiting Cost Distribution Function. From this point on, only limited progress is possible without making a further assumption (or approximation) about the behavior of the algorithm. This is essentially that ultimately the distribution function

$$\psi_\delta(y;\, h, T, R, P, M, \vec{\rho}, \vec{\gamma})$$

settles down to either a function independent of h, or to one which depends on h in a quasi-periodic manner.

Assumption 5.1. A limiting cost distribution function $\bar{\psi}_\delta$, defined by

$$\bar{\psi}_\delta(y;\, T, R, P, M, \vec{\rho}, \vec{\gamma}) = \lim_{h_2 \to -\infty} \frac{1}{h_1 - h_2} \int_{h_2}^{h_1} \psi_\delta(y;\, h, T, R, P, M, \vec{\rho}, \vec{\gamma})\, dh$$

exists, is finite and is independent of h_1 as indicated.

It is important to appreciate the nature of this assumption. In the previous section we assumed that the algorithm has certain scale invariance properties and showed that as a consequence, the distribution function θ_D associated with this problem has a certain form. In practice one can determine *analytically* whether or not the algorithm has these properties and, if it does, the results of the previous section are rigorously established.

Assumption 5.1 is of quite a different nature. It stems from the author's belief that the algorithm, faced with this particular problem, settles down to a quasi-steady rate of minimization and defines a new function, a limiting cost distribution function which quantifies this rate. While theoretically it might be possible to analyze any particular algorithm to the extent that one could establish this, or show that it is not true, the effort involved would be out of all proportion to the utility of the result. Part of the numerical experiment to evaluate algorithms has to include an attempt to verify this assumption numerically. Such verifications show at most that for certain ranges of the incidental parameters, immediate consequences of such an assumption appear to be close approximations to reality.

In terms of the function θ_D, this assumption may be written in the form

$$\begin{aligned}(5.2)\quad &\bar{\theta}_D(Y; X_1, X_2, X_3, X_4, \vec{\rho}, \vec{\gamma})\\ &= \operatorname*{Lim}_{H_2 \to -\infty} \frac{1}{H_1 - H_2} \int_{H_2}^{H_1} \theta_D(Y; H, X_1, X_2, X_3, X_4, \vec{\rho}, \vec{\gamma})\, dH\end{aligned}$$

and $\bar{\theta}_D = \bar{\psi}_\delta$.

Simple substitution of the functional relationships (4.7)–(4.10) or (4.23)–(4.26) into (5.1) or (5.2), respectively, gives new functional relationships. These have an appearance almost identical with the previous ones, the difference being that barred functions replace unbarred functions and the second argument (corresponding to h or H) is missing. For example, corresponding to (4.9) we find

$$(5.3)\quad \bar{\psi}_\delta(y; T, R, P, M, \rho_1, \rho_2, \rho_3, \vec{\gamma}) = \bar{\psi}_\delta(y; T, R, P, M, \rho_1, \rho_2, \rho_3 + l/P, \vec{\gamma}),$$

a relation valid when the algorithm is invariant under $t(k, 1, A, d)$ defined in (4.9a). Since for all values of l, this transformation is a member of the group T_G, this establishes the comparatively minor result that $\bar{\psi}_\delta$ is independent of ρ_3 for all T_G-invariant algorithms.

In the following theorems, $\tilde{\theta}$ and g should be read as "a function of". In fact, in each case $\tilde{\theta}$ is the function obtained from θ_D by removing redundant arguments. These theorems follow directly from (4.23)–(4.26), (5.2) and (5.3).

THEOREM 5.4. *When the algorithm is T_G-scale invariant and Assumption* 5.1 *is valid,*

$$(5.4)\quad \bar{\psi}_\delta = \tilde{\theta}_{\delta/MP}(MPy, T^2R/MP, R/P, \rho_1, \rho_2, \gamma_1, \gamma_3, \gamma_4)$$

and its percentile $\bar{y}_Q$ may be expressed in the form

$$(5.5)\quad \bar{y}_Q = \frac{1}{MP} g_{Q,\delta/MP}(T^2R/MP, R/P, \rho_1, \rho_2, \gamma_1, \gamma_3, \gamma_4).$$

THEOREM 5.6. *When the algorithm is fully scale invariant and Assumption* 5.1 *is valid, then*

$$(5.6)\quad \bar{\psi}_\delta = \tilde{\theta}_{\delta/MP}(MPy, T^2R/MP, \rho_1, \rho_2, \gamma_1, \gamma_3, \gamma_4)$$

and

$$(5.7)\quad \bar{y}_Q = \frac{1}{MP} g_{Q,\delta/MP}(T^2R/MP, \rho_1, \rho_2, \gamma_1, \gamma_3, \gamma_4).$$

Relations (5.5) and (5.7) are simple consequences of the definition of a percentile.

6. Experimental Results and Conclusions. In this section we present some results obtained using implementations of four well-known algorithms. All four are fully scale invariant so, if one can rely on Assumption 5.1, the cost distribution function

ψ_0 for each is of the form described in Theorem 5.6, and the median (see (5.7)) has the form

$$\bar{y}_{0,5} = \frac{1}{MP} g_{0.5}(T^2R/MP, \rho_1, \rho_2, \gamma_1, \gamma_3, \gamma_4) \tag{6.1}$$

where we have suppressed the dependence on δ which is zero. The comparison is based on calculating by numerical means the function $g_{0.5}$ for each of the four algorithms; in general, the more economic algorithm is the one with the smaller value of $g_{0.5}$.

As it stands, $g_{0.5}$ is a function of six variables. However, two of these define a starting disc and three define starting conditions. It is heuristically plausible to believe that the function $g_{0.5}$ will be relatively insensitive to these five parameters as it describes the behavior of the algorithm long after it has started. However, so far as γ_1 and γ_3 are concerned, this argument is less plausible in algorithms having reset mechanisms (Fletcher (1972)).

The first stage in the measurement process is naturally to acquire evidence about the nature of g_Q. In a pilot project (described in some detail in Lyness and Greenwell (1977)) considerable effort was devoted to obtaining numerical evidence relating to the nature of g_Q. This evidence established a *prima facie* case for the following statements.

(1) For a wide range of values of T^2R/MP, Assumption 5.1 appears to be valid, and g_Q is of form (5.5) or (5.7).

(2) The function g_Q appears to be almost independent of parameters ρ_1, ρ_2, γ_1, γ_3 and γ_4 for a significant range of values of these parameters.

Our experiments were naturally limited. However, we did search for counterexamples to these statements and found none. All discrepancies were minor and noncoherent and could be accounted for by the crudeness of the numerical technique. In no case did any of the algorithms terminate prematurely. Apart from this (which implies $\varphi(\infty) = 1$) our experiments were too crude to obtain detailed information about the tails of the distribution. A phenomenon which we termed "rung jumping" was encountered. This is described in Lyness and Greenwell (1977).

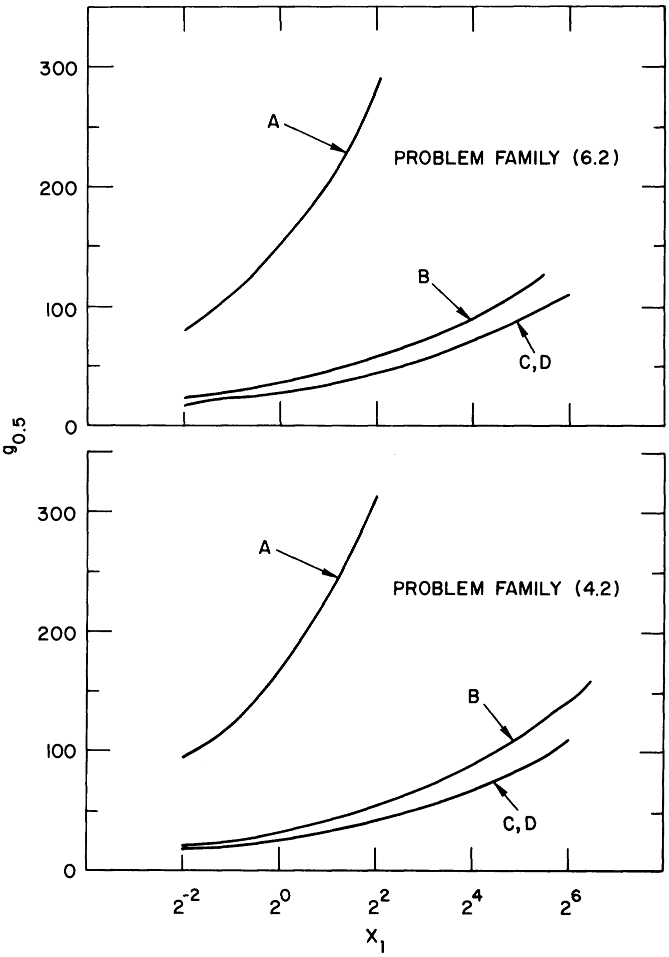
In the figures we present the function $g_{0.5}(X_1)$ as a function of X_1 for four routines and for three problem families.

The problem families are (4.2) above, and two variant families, namely

$$\begin{aligned} &f(x, y, z; T, R, P, M) \\ &\quad = T^2\left[\left(x - R\cos\frac{2\pi z}{P}\right)^2 + \left(y - R\sin\frac{2\pi z}{P}\right)^2\right] + Mz, \end{aligned} \tag{6.2}$$

$$f(x, y, z; T, R, P, M, A) = T^2(r - R)^2 + 2A\left(1 - \cos\left(\frac{2\pi z}{P} - \theta\right)\right) + Mz, \tag{6.3}$$

where $r^2 = x^2 + y^2$ and $\theta = \arctan(y/x)$.



For these latter two, certain definitions given in the text have to be modified. In place of the definition of X_1 in (4.22) one must set

(5.6′) $$X_1 = R^2 T^2/MP$$

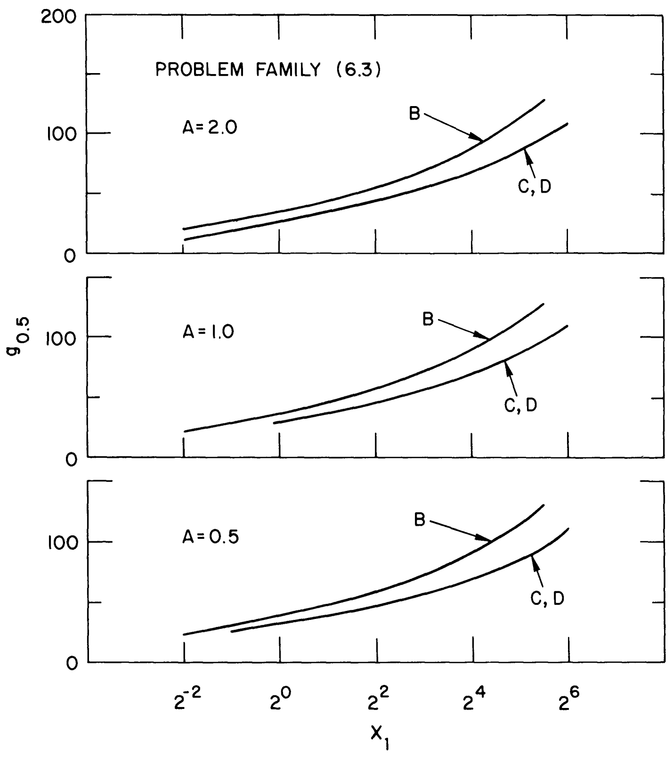
and one must set

(4.4′) $$\Gamma^{(0)} = \text{diag}\{\gamma_1/T^2, \gamma_1/T^2, \gamma_3 P/M\},$$

(4.5′) $$\Delta f^{(0)} = \gamma_4 T^2 R^2,$$

in place of Eqs. (4.4) and (4.5), respectively.

With these modifications, all the theory given in Sections 4 and 5 is valid, except that, for the third problem family, an additional problem parameter $A = X_5$ is present in many argument lists, specifically all lists containing X_1 or T. These families conform to the description given in the beginning of Section 4. They are based on an identical helix, but the objective function has a different nature away from the helix. Our reason for including these is to demonstrate at least some limited generality for conclusions which may be based on the results illustrated in one of the figures.



The four routines, whose behavior is illustrated in the figures are:

(A) An in-house implementation of the Davidon-Fletcher-Powell quasi Newton algorithm.

(B) An in-house version of DRVOCR (Davidon and Nazareth 1977) which is based on an optimally conditioned optimization algorithm without line searches (Davidon (1975)).

(C) A quasi Newton algorithm QNMDER (Gill, Murray, Picken, Graham and Wright (1975)).

(D) A quasi Newton algorithm VA13AD of the Harwell Library.

At no stage were we able to differentiate between the performance of routines (C) and (D)

Besides calculating $\bar{y}_{0.5}$ the median, we retained other numerical features of the statistical distribution functions. Among these was an average between the first and third quartiles, defined in Lyness and Greenwell (1977). In the results appearing on the figure, this average coincided with the corresponding median to within one percent.

The following comments are in order. It seems that for these helical valleys, the BFGS methods are marginally more economic than DRVOCR by factors of between 15 and 20% and that the in-house DFP implementation is less economic than any of these by a substantial margin. Moreover, the two BFGS routines gave almost identical results which coincide with results produced by an in-house implementation having a poor line search.

However, the author feels that the more significant conclusions to be drawn from this bench mark experiment are qualitative. By means of a carefully conducted experiment, measuring properly defined functionals, we are able to obtain information about algorithms of a definite and useful character. This information can be added to as and when other algorithms are forthcoming. And the information provides clear and unambiguous evidence about the relative merits of the routines when faced with a particular topography.

The author hopes that similar bench mark experiments will be carried out using other topographies, using definitions of the same character as those outlined at the end of Section 2.

Applied Mathematics Division
Argonne National Laboratory
Argonne, Illinois 60439

W. C. DAVIDON (1975), "Optimally conditioned optimization algorithms without line searches", *Math. Programming,* v. 9, pp. 1–30.

W. C. DAVIDON & L. NAZARETH (1977), DRVOCR–*A Fortran Implementation of Davidon's Optimally Conditioned Method,* ANL-AMD Technical Memorandum No. 306.

R. FLETCHER (1972), "Conjugate direction methods," *Numerical Methods for Unconstrained Optimization,* (W. Murray, Editor), Academic Press, London, pp. 73–86.

P. E. GILL, W. MURRAY, S. M. PICKEN, S. R. GRAHAM & M. H. WRIGHT (1975), *Subroutine* QNMDER, *A Quasi-Newton Algorithm to Find the Unconstrained Minimum of a Function of N Variables When First Derivatives are Available,* Technical Memorandum E4/02/0/Fortran/11/75, National Physical Laboratory, Teddington, Middlesex TW11 OLW, England.

J. N. LYNESS (1979), "The affine scale invariance of minimization algorithms," *Math. Comp.,* v. 33, pp. 265–287.

J. N. LYNESS & C. GREENWELL (1977), *A Pilot Scheme for Minimization Software Evaluation,* ANL-AMD Technical Memorandum No. 323.

M. J. D. POWELL (1975), *Some Global Convergence Properties of a Variable Metric Algorithm for Minimization Without Exact Line Searches,* Technical Memorandum C. S. S. 15, AERE Harwell.