TWO NEW DIRECT MINIMUM SEARCH PROCEDURES FOR FUNCTIONS OF SEVERAL VARIABLES

Bruno F. W. Witte Design Specialist
and
William R. Holst Senior Research Engineer
General Dynamics/Astronautics
San Diego 12, California

(1) LINEAR SEARCH METHOD FOR n VARIABLES

(1.1) *Summary Of The Method*

The method can somewhat vaguely be classified as a modified "steepest descent." It is, of course, an iterative procedure. Each cycle, to be iterated on, consists essentially of two parts: in Part I a "best" line is found, in Part II an attempt is made to minimize the given function along this line. Thus, each cycle resembles the corresponding cycle in the method of steepest descent. The method of steepest descent differs from our method in the manner in which in Part I the "best" line is found. Steepest descent, in fact, implies that this line (let us call it the "baseline" from now on) be defined by the starting point for the cycle and the gradient of the function at this starting point, where the starting point, in turn, is the minimum point of the preceding cycle. Well known modifications of the steepest descent are concerned with, for example, baselines restricted to a subspace normal to the preceding baseline, or with different ways of minimizing along a given baseline, or perhaps with the question as to how feasible it is to seek a minimum at all along a given baseline during each cycle before switching to the next baseline.

(1.2) *Symbol Explanations*

y — The ordinate of arbitrary points in the space Φ, i.e. that coordinate which is plotted in the same direction as is the function value.

F — The function to be minimized.

n — Number of independent variables in $F = F(x_1 \ldots x_n)$.

S — Designates the hyper-surface defined in Φ by F.

Φ — Designates the space spanned by the n independent variables $x_1 \ldots x_n$, the abscissas, and by y, the ordinate.

ϕ — The space spanned by the n independent variables $x_1 \ldots x_n$.

x_i — The abscissas, i.e. the independent variables, $i = 1 \ldots n$.

x — The point with coordinates x_i, i.e. $x = (x_1 \ldots x_n)$.

x^k — The starting point in ϕ for minimizing F along b^k in cycle C^k. Also the minimum point found by minimizing F along b^{k-1} in cycle C^{k-1}. $x_1^k \ldots x_n^k)$, $k = 0, 1, 2, \ldots$.

C^k — Designates the k-th cycle of the iterative procedure leading to the minimum of F. $k = 0, 1, 2, 3, \ldots$.

F^k — An abbreviation for $F(x^k)$.

S^k — The starting point in Φ corresponding to x^k, i.e. $S^k = (x^k, F^k)$, $k = 0, 1, 2, \ldots$.

T^k — The n-dimensional hyper-plane tangent on S in S^k.

b^k — The baseline chosen in cycle C^k.

grad — An n-vector in the space ϕ pointing in the direction of steepest acscent of any subspace of Φ or manifold of Φ with dimension 1, 2, . . . , or n. The magnitude of this vector is equal to the slope of steepest ascent. For linear subspaces this vector is independent of x. For the subspace S this vector is the familiar gradient of F.

Λ — An operator designating the intersection of the two subspaces between which it stands.

P^i — A subspace of Φ with dimension n — i defined by $P^i = T^0 \Delta T^1 \Delta \ldots \Delta T^i$.

Q^i — A straight line in Φ defined by $Q^i = T^{i-n+1} \Lambda T^{i-n+2} \Lambda \ldots \Lambda T^i$, $(i \geqq n-1)$.

p^i — The intersection $P^i \Lambda \phi$.

q^i — The point $Q^i \Lambda \phi$.

L^i — The point on Q^i with $y = 1$.

^{i} — The projection of L^i on ϕ.

E^k — The equation of a hyper-plane parallel to T^k through the origin of Φ.

λ_j — Lagrangian undetermined multipliers.

η^k —
ξ^k — $\xi^k = (\xi_1^k \ldots \xi_n^k)$ } The running coordinates of an arbitrary point on T^k.

a_i^k — Defined by $a_i^k = -\partial F/\partial x_i$, evaluated at x^k.

s — The distance from x^k to a point on b^k. s is positive in the downward direction.

G — The function F considered as a function of only s, i.e. $G(s) = F(x_1(s), \ldots, x_n(s))$.

G′ — The derivative dG/ds.

(1.3) *The Baseline Choice*

The method used to choose the next baseline can best be visualized as follows: The function $y = F(x_1 \ldots x_n)$ defines an n-dimensional hyper-surface, S, embedded in the (n + 1) — space, Φ, spanned by the n-spaced, ϕ, of the independent variables $x_1 \ldots x_n$ and the dependent variable y. Let $x^0 = (x_1^0 \ldots x_n^0)$ be the starting point of the first cycle, C^0, of our search procedure, and let $F^0 = F(x^0)$ and x^0 define the corresponding starting point $S^0 = (x^0, F^0)$ on the hyper-surface.* Let T^0 designate the hyperplane tangent in S^0 on S (whenever necessary we will assume that such tangent hyperplanes exist). The baseline, b^0, for the first cycle, C^0, is then chosen to be the line through x^0 and with direction defined by the gradient of T^0, grad T^0. Naturally, we have for this cycle grad T^0 = grad $F(x^0)$. We will now assume for the moment that we know how to proceed with part II, and that at the end of part II we obtain an approximation $x^1 = (x_1^1 \ldots x_n^1)$ for the minimum of S along b^0. x^1 is also the starting point of cycle C^1. (In general, x^i designates the starting point and s^{i+1} the minimum point of cycle C^i; $S^i = (x^i, F^i)$ is the point on the hyper-surface corresponding to x^i, and T^i is the hyperplane tangent in S^i on S; finally b is the baseline for cycle C^i). The next baseline b^1 is then chosen to be the line through S^1 and with direction defined by grad $(T^0 \Lambda T^1)$, where $T^0 \Lambda T^1$ is the intersection of T^0 and T^1, i.e. an (n—1)-dimensional linear subspace of Φ. Assuming again that we know how to find x^2, the minimum point along b^1, b^2 is found as the line through x^2 in a direction given by grad $(T^0 \Lambda T^1 \Lambda T^2)$.

In general, for $i < n-1$, b^i is determined so that it becomes the baseline through x^i with a direction given by grad $(T^0 \Lambda T^1 \Lambda \ldots \Lambda T^i)$. The argument of the gradient, $P^i = T^0 \Lambda T^1 \Lambda T^2 \Lambda \ldots \Lambda T^i$, is a linear subspace of Φ, and has dimension n—i. For $i = n-1$, P^i would then be a straight line; for $i = n$, it would be only one point, for $i > n$, P^i is an empty set; for $i \geqq n$, therefore, grad P^i is not defined. For $0 \leqq i < n-1$, grad P^i is defined, of course, as the direction in ϕ along which P^i ascends most rapidly. Since all P^i are linear, grad P^i is independent of x.

For $i \geqq n-1$, b^i is determined so that it becomes the baseline through x^i with a direction given by grad Q^i, where $Q^i = T^{i-n+1} \Lambda T^{i-n+2} \Lambda \ldots \Lambda T^i$. Thus, all Q^i are straight lines in Φ, and their gradients are parallel to their normal projections on ϕ.

*An attempt is made in the above and subsequent notation to designate by lower-case letters all points, or point sets, or subspaces, which are completely embedded in the n-space of the independent variables $x_1 \ldots x_n$, and to use capital letters if they are not completely contained in this n-space.

In general, for all i, the above defined gradients are determined as follows: Let p^i and q^i be defined by $p^i = P^i \Lambda \Phi$ (for $0 \leqq i < n-1$), $q^i = Q^i \Lambda \phi$ (for $n-1 \leqq i$). Then all q^i are single points. Next consider the point L^i ($i \geqq n-1$) with ordinate $y=1$ on one of the lines Q^i, and its projection ℓ^i on ϕ. The directed line from q^i to ℓ^i is parallel to grad Q^i, and $|\text{ grad } Q^i| = 1/|q^i - \ell^i|$. The n coordinates of $\ell^i = (\ell^i_1 \ldots \ell^i_n)$ are found by solving simultaneously the n linear equations, E^k, corresponding to the n T^k in the expression for Q^i and setting $y = 1$. Setting $y = 1$ is arbitrary, and insures only that $y \neq 0$. The n coordinates of $q^i = (q^i_1 \ldots q^i_n)$ are found by solving the same system with $y = 0$. Next consider some arbitrary point L^i ($0 \leqq i < n-1$) with ordinate $y = 1$ on one of the linear subspaces P^i of dimension $n - i$, and its projection ℓ^i on ϕ. Let q^i (for $0 \leqq i < n-1$) be the point on p^i closest to ℓ^i. The directed line from q^i to ℓ^i is then, again, parallel to grad P^i, and also $|\text{ grad } P^i| = 1/|q^i - \ell^i|$. Since L^i can be chosen arbitrarily from all the points on P^i with $y = 1$, we set $\ell^i_k = 0$ (for $k = i+2, \ldots, n$), and find the first $i+1$ coordinates of $\ell^i = (\ell^i_1 \ldots \ell^i_{i+1}, \ell^i_{i+2} \ldots \ell^i_n)$ by solving simultaneously the $i+1$ linear equations, E^k, corresponding to the $i+1$ T^k in the expression for P^i, again also setting $y = 1$. The n coordinates of $q^i = (q^i_1 \ldots q^i_n)$ are determined by requiring that the square of the distance from ℓ^i to q^i be a minimum, i.e. $d^2 = (\ell^i_1 - q^i_1)^2 + (\ell^i_2 - q^i_2)^2 + \ldots + (\ell^i_n - q^i_n)^2 = \min$. Furthermore, we note that the q^i_k must satisfy the constraint that q^i is to be on p^i, which is the same as saying that the q^i_k must also satisfy the $i + 1$ linear equations, E^k ($k = 0 \ldots i$), corresponding to the $i + 1$ T^k in the expression for P^i, but now setting $y = 0$. The Lagrangian method of undetermined multipliers leads us then to the unconstrained minimization of the function $g(q^i_1 \ldots q^i_n, \lambda_0 \ldots \lambda_i) = d^2 + \lambda_0 E^0 + \ldots + \lambda_i E^i$. Equating to zero the partial derivatives of g with respect to the n q^i_k and the $i + 1$ λ_j, we obtain again a system of linear algebraic equations, here of order $n + i + 1$. Note that this way of finding q^i applies only to the initial cycles of the search for which $i < n - 1$; hence the largest linear system to be solved here will have order $n + (n-2) + 1 = 2n-1$.

The linear equations E^k ($0 \leqq k$) correspond to our hyperplanes T^k which are tangent to the hyper-surface S at the locations x^k. The analytic expression for one of the E^k can readily be obtained from a Taylor expansion of F about x^k: $F(x) = F(x^k) + (x - x^k) \cdot \text{grad } F(x^k) +$ higher-order terms. If we replace $F(x)$ by η^k and $x = (x_1 \ldots x_n)$ by $\xi^k = (\xi^k_1 \ldots \xi^k_n)$, delete the higher-order terms, collect all constant terms, and rewrite this expansion in terms of the individual running coordinates on the tangent plane, we obtain with grad $F(x^k) = -(a^k_1 \ldots a^k_n)$: $a^k_1 \xi^k_1 + a^k_2 \xi^k_2 + \ldots + a^k_n \xi^k_n + \eta^k = b^k$. This can be simplified considerably for computational purposes by always setting $b^k = 0$. This is permissible because we are only interested in the slopes of the T^k, or in the slopes of subspaces common to several T^k. The E^k are then explicitly: $E^k = a^k_1 \xi^k_1 + \ldots + a^k_n \xi^k_n + \eta^k = 0$, and they describe hyperplanes which are parallel to the tangent planes, and contain the origin of the space Φ. The a^k_j could be evaluated directly at x^k as the partial derivatives of F with respect to x_j. Or they may be approximated by finite difference approximations to these derivatives. The computer program referred to in the abstract uses first-order difference approximations.

(1.4) *Minimizing along the Baseline*

The search for the baselines was explained in the preceding section. In this section we describe what to do with a baseline once it is selected, i.e. how to minimize the given function $F(x_1 \ldots x_n)$ along this line. Let us introduce an (s, G)-coordinate system with origin at x^k, with the positive s-axis in the downward direction of the baseline, and with the positive G-axis in the positive y-direction. The function G(s) is defined as $G(s) = F(x_1(s) \ldots x_n(s))$. We restate now our problem more precisely: given $s_0 = 0$, $G_0 = G(s_0)$, $G_0 = G'(s_0)$, and $\bar{s}_1$ (the initial step size), find a "bracketed" approximation, s_m, to a relative minimum of G(s), and verify the existence of two bracketing numbers s_a and s_b, such that $s_a < s_m < s_b$ and $G(s_a) > G(s_m) < G(s_b)$. The procedure is described below in several steps.

STEP (1)

s_1 is equated to $\bar{s}_1$, and $G = G(s_1)$ is evaluated. This defines the point $P_i = (s_1, G_i)$ for $i = 1$, and with $G_i = G(s_i)$.

Question: Is $G_1 \geqq G_0$?

If yes, a parabola is passed through P_0 with the given slope G_0, and through P_1. This is done in the following Step (2).

If no, we go directly to Step (3) to check whether P_1 is above or below the line through P_0 with slope G_0.

STEP (2)

The minimum location s_2 is found for the parabola through P_0 and P_1 with slope G_0 at P_0. s_2 will satisfy: $|s_0| < |s_2| \leqq |s_1|/2$. $G_2 = G(s_2)$ is evaluated.

Question: Is $G_2 < G_0$?

If yes, we have solved our problem, i.e. we have obtained a bracketed approximation to the minimum, with $s_a = s_0$, $s_m = s_2$, $s_b = s_1$. However, our experience showed that the expenditure of one more function evaluation often allowed a considerable improvement of the value for s_m. This is particularly desirable toward the end of our search for the minimum of $F(x_1 \ldots x_n)$. The improvement of s_m is obtained by cubic interpolation, and is described in Step (4). Before going there, however, the indices 1 and 2 are interchanged on the quantities P_1, G_1, s_1 and P_2, G_2, s_2 so as to number the points P_0, P_1, P_2 in the order of increasing $|s|$-values.

If no, we substitute P_2 for P_1, and repeat Step (2) until either $G_2 < G_0$, or $|s_2| < |\bar{s}_1|/100$. In the latter case we replace $\bar{s}_1$ by $-\bar{s}_1$, G_0 by the slope of the line (P_0, P_2), and return to Step (1). This starts a search for the minimum of G in the range of negative s-values (assuming that G_0 was inaccurate). In case our search should then again lead us back to the vicinity of the same s-value, $s = s_0$, we set $s_m = s_0$ with brackets $s_a = -|\bar{s}_1|/100$ and $s_b = |\bar{s}_1|/100$. In this case our starting point and our approximation to the minimum are one and the same point. That situation will always happen at the final minimum of $F(x_1 \ldots x_n)$, but may also happen occasionally sooner. The termination procedure, described further below, then is triggered.

STEP (3)

Question: Is P_1 above the line through P_0 with slope G_0 ?

If yes, the minimum location s_2 is found for the parabola through P_0 and P_1 with slope G_0 at P_0. It satisfies $|s_2| > |s_1|/2$. If $|s_2| < |s_1|$, the indices 1 and 2 are interchanged.

If no, the minimum location s_2 is found for the cubic with clope G_0 at P_0 and with its point of inflection at P_1. It satisfies $|s_2| > 2|s_1|$.

Question: Is $G_1 < G_2 = G(s_2)$?

If yes, we found a bracketed approximation to s_{min}, and proceed with Step (4).

If no, we re-define G_0 to be the slope of the parabola through P_0, P_1, P_2 at point P_1, then subtract 1 from all index values occurring in $P_i = (s_i, G_i)$ $(i = 0, 1, 2)$, and return to Step (3).

STEP (4)

A cubic is passed through P_0 with slope G_0, and through P_1 and P_2. $\bar{s}_m$ is the location of the minimum of this cubic. Note that the existence of brackets for $\bar{s}_m$ guarantees the existence of the minimum.

STEP (5)

We test next the *three* lowest values of G, found so far, whether they agree with each other within a given tolerance.

If yes we set s_m equal to that s_i-value for which $G(s_i)$ is smallest. ($\bar{s}_m$ usually is, but need not be, that s_i-value.)

If no, a parabola is passed through the lowest three points, and the minimum value of this parabola is compared with the two lowest available values of G. If these three values agree with each other within the given tolerance we have found s_m; if they do not agree, we evaluate G for that s-value for which the parabola had its minmium, and go to Step (6).

STEP (6)

We test the *two* lowest values of G with the minimum value of the last parabola. If these three values agree with each other within the given tolerance we have found s_m; if they do not agree, a parabola is passed through the lowest three points of G and we compare the minimum values of the last *two* parabolas with the lowest value of G.

If these three values agree with each other within the given tolerance, we have found s_m; if they do not agree, we evaluate G for that

s-value for which the last parabola had its minimum, and repeat Step (6).

(1.5) *Detection of the Minimum*

The iterated execution of the procedures for Parts I and II of each cycle may be discontinued whenever one of the following conditions is detected:

Condition A: The partial derivatives of $F = (x_1^k \ldots x_n^k)$ satisfy $|\partial F/\partial x_i^k| \leqq \epsilon$ for all $i = \ldots n$, and for a given small value of ϵ, say $\epsilon = 10^{-6}$. As explained at the end of section (1.3, these partial derivatives are evaluated to obtain the coefficients a^k in the equations E^k.

Condition B: The starting point and the approximation to the minimum point on the baseline are one and the same point. How this can happen was explained in Step (2) of section (1.4).

Condition C: The matrix of the linear equations E^k, discussed in section (1.3), is very ill-conditioned or singular. This may occasionally happen before either Condition (A) or Condition (B) is satisfied.

Condition D: The numerical values of the coordinates of the minimum location and/or the minimum of the function have stabilized to a given number of significant figures.

Whenever one, or perhaps two of the above conditions are satisfied, there is an excellent chance that the minimum has been found. One may then do one of two things: (i) terminate the execution of the program, or (ii) begin a somewhate more extensive testing procedure in the course of which the vicinity of the suspected minimum location is explored, and it is thereby made even more likely that the solution has been found. This is what was programmed at G.D./Astronautics. Preference was given to method (ii) because it also provides an answer to the question how sensitive the function is to small deviations from the minimum location. The knowledge of this sensitivity is usually desired together with the minimum location itself.

(2) CIRCULAR ARC SEARCH METHOD FOR 2 VARIABLES

(2.1) *Summary of the Method*

The iterative circular search is preceded by a starting procedure, which finds five points near the spine of the valley of the function to be minimized. Thereafter, each iteration begins with an attempt to approximate by a portion of a circle the course of the projection of the valley onto the space of the independent variables. The first circle is fit to three of the five valley points found. All succeeding circles are fit to four or more valley points. As the iteration progresses toward the minimum these circles approach the osculating circle at the minimum location. During each iteration the determination of a circular arc approximation to the spine of the valley is followed by an attempt to predict a point on this arc which is closer to the desired minimum than any other point obtained so far. Again during each iteration, the prediction of such a point is followed by a steepest descent from this point into the valley, which amounts to a minor, yet important correction. The correction is important since it insures that deviations between the courses of the circular arcs and the portions of the spine of the valley approximated by the arcs will not accumulate as the iteration progresses. The new valley point then replaces the highest of the four lowest valley points found before, and this predictor-corrector type method is repeated in the next iteration. Empirical results show that the sequence of valley points obtained converges quite rapidly to the desired minimum. In the following sections we emphasize again the geometric aspects of the more important steps summarized above, rather than to give the algebraic details. This is done since some of the details are still subject to experimentation and analysis.

(2.2) *The Starting Procedure*

We assume that a first guess is available as a starting location x^1. We find its valley point v^1. A valley point v^i will always mean the location of the minimum of the given function F along a line through x^i in the direction of $-\text{grad } F(x^i)$. We find the distance $d^1 = |v^1 - x^1|$ and a unit vector e^1 in the direction of the projection of $-\text{grad } F(v^1)$ on a line normal to $-\text{grad } F(x^1)$. We then find four more valley points v^i $(i = 2 \ldots 5)$ by taking a step $d^i = (3/4)^{i-2}d^1$ in the direction of e^{i-1} from v^{i-1} to x^i, and then by minimizing from x^i to v^i. The directional minimizations are done by using the

same method which was described in section (1.4).

(2.3) *The Approximating Circular Arcs*

The circular arcs are obtained as arcs of "best fitting" circles to the locations v^i of the lowest three, four, or five valley points. Three valley points are chosen only once, i.e. for the first circle; four are usually chosen thereafter; if it happened, however, that the last valley point found had a function value higher than the highest of the four lowest valley points previously known, then five points were chosen.

(2.4) *The Predictor-Corrector Step*

A "best fitting" two-dimensional plane E_2 is found for the valley points $V^i = (v^i, F^i)$ in space Φ, where v^i and i were explained above, and $F^i = F(v^i)$. It is helpful to visualize now a cylinder in Φ parallel to the y-axis and intersecting ϕ along the circle of the preceding section. This cylinder and the above plane E_2 intersect in an ellipse. The location u of the point on this ellipse with the smallest y-value is our predicted point. The distance of the point u from the spine of the valley decreases rapidly as the iteration progresses, and appears to become insignificantly small long before the minimum is reached. Nevertheless, it is unknown at the present time how damaging the cumulative effect woud be. We are employing, therefore, during all iterations a straight steepest descent correction which leads from u to the next valley point. The correction is done in the same way as was explained for the five starting points in (2.2), i.e. by using the method described in (1.4). The initial step taken in minimizing from the low point on the circle to the valley is set equal to one-half the length of the gradient vector at the low point divided by the coefficient of the second-degree term of the parabola approximating the shape of the function along the previous line of minimization. We expect that further studies of this particular correction will lead to considerable savings in the number of function evaluations. For example, it appears that the correction usually need be made only in a radial direction rather than the negative gradient direction, while the latter is needed only occasionally. Moreover, it is doubtful whether any such correction is needed at all in each iteration.

(2.5) *The Switch to Parabolas*

The above predictor-corrector step is taken at least four times. If thereafter the function value of the lowest valley point is within ½% of the function value of the second-lowest valley point, the method of predicting a new point u on the circle is modified somewhat. Instead of finding the above described plane E_2, we find the "best-fitting" line L to the same four points v^i defining our circular arc and in the same plane with this circular arc. The lowest three v^i are then projected on L to give the three points w^i on L. These and the F^i —values associated with the v^i define three points $W^i = (w^i, F^i)$ in the plane through L and parallel to the y-axis. The location w^m on L of the minimum of the parabola through the three points W^i is then projected back onto our circular arc to give our predicted point u. The steepest descent correction is gain applied to u, and produces the next valley point.

(2.6) *The Tail Correction*

After successive approximations to the minimum value of the function have stabilized to five figures, the search terminates with a simple tail correction, which consists in minimizing the function along the line through the two lowest valley points, and which usually adds several significant figures to the answers.

It is interesting to observe the behavior of the radii of the circular arcs after the iteration has begun to converge. As was said before in Section (2.1) the circles approach the osculating circle of the spine of the valley at the minimum location. However, once the sequence of valley points has converged to the minimum, all subsequent points fall very close to each other; they then no longer define the course of the valley nor do they define its osculating circle. Instead, the radii of the "best-fitting" circles *suddenly* shrink by several orders or magnitude to dimensions comparable to the inaccuracies in the coordinates of the minimum location. When this shrinkage occurs the search should be terminated. Usually, however, this does not happen before the search had already been terminated with the above described tail correction.

(2.7) *An Extension to n Variables: The Spherical Shells Method*

The good results obtained with the two-

dimensional Circular Arcs Method suggest its extension to n independent variables as follows:

(—1) START. A starting point x^1 is given in n space as a first approximation. Find its valley point v^1, the distance $d^1 = |v^1 - x^1|$, the n — 1 space E^1 normal to grad $F(x^1)$, and the direction e^1 in E^1 as the projection of —grad $F(v^1)$ on E^1. Find n + 3 more initial valley points v^i ($i = 2 \ldots n + 4$) by taking steps $d^i = (n + 1)/(n + 2))^{i-2}d^1$ from v^{i-1} in the direction e^{i-1} to x^i, and then by minimizing F in the direction of —grad $F(x^i)$ from x^i to v^i, for all values of $i = 2 \ldots n + 4$.

(— 2) FLATS. Fit an m flat (i.e. a linear m space) to the lowest m + 2 valley points in n space, such that its dimension m is as small as feasible. (The feasibility could be judged by observing successive standard deviations σ_m of the points from the m flats for the sequence of decreasing m — values: m = n, n — 1, . . . , 1. A sudden large increase of σ_{k-1} over σ_k would indicate that m = k is feasible while m = k—1 is not feasible.) Skip to (— 4) if m = 1.

(— 3) SPHERES. Fit the surface of an m-dimensional sphere to the projections of the lowest m + 2 valley points on the above m flat. Some shell of this sphere will approximate the course of the (hyper —) valley. Pin-point next an optimum surface point on this shell to which to go next, find its valley point, and return to (—2).

(— 4) PARABOLAS. Find the parabola through the projections of the lowest three valley points on the plane spanned by the function axis and the regression line found in (—2). Go to the minimum of this parabola, find its valley point, and return to (— 2) unless the minimum value of the function has begun to stabilize.

(— 5) TAIL. Minimize the function along a line through the lowest two valley points in n space. Find the valley point corresponding to this directional minimum. Repeat (— 5) till the function minimum has stabilized to the desired number of figures.

The above described Spherical Shells Method will be programmed at General Dynamics-Astronautics in the near future, and results will be published when available.

(3) EXAMPLES AND COMPARISONS

The following functions of x and y were minimized. ($r = (x^2 + y^2)^{1/2}$ and $\theta = \tan^{-1}(y/x)$.)

$$
\begin{aligned}
&(3.1)\ \text{ROSIE} &&= 100(y - x^2)^2 &&+ (1 - x)^2\\
&(3.2)\ \text{SHALOW} &&= (y - x^2)^2 &&+ (1 - x)^2\\
&(3.3)\ \text{STRAIT} &&= (y - x^2)^2 &&+ 100(1 - x)^2\\
&(3.4)\ \text{CUBE} &&= 100(y - x^3)^2 &&+ (1 - x)^2\\
&(3.5)\ \text{DOUBLE} &&= 100(r - 1)^2(r - 2)^2 &&+ 0.1(2 - x)^2\\
&(3.6)\ \text{HEART} &&= 1000(1 + \cos\theta - r/10)^2 + y^2(x - 10)^4
\end{aligned}
$$

All were programmed in double-precision for an I.B.M. 7094 computer. Results are listed in the following tables, and the results for ROSIE are discussed in detail and compared with other methods.

(3.1) ROSIE

Minimum (F = 0) at (1, 1), with a steep valley along $y = x^2$, and a side valley along the negative y-axis.

Table (3.1A) summarizes the end results of various investigators. Table (3.1B) summarizes the rapidity of convergence of some of these methods. Table (3.1C) shows the rate of convergence of the circular arcs method for various starting points. Table (3.1D) gives minimization details for the circular arcs method.

Table (3.1A) End results of various methods of minimizing the function ROSIE when starting from the point (x, y) = (— 1.2, 1)

k = number of valley points found.

N = number of function evaluations

Note: The value of the function given by ref. (5) in the table is not consistent with the values given for x and y. The correct value of F equals 1(— 6) for (x, y) = (1.0001, 1.0001).

Discussion: Parentheses in column N indicate that the number of function evaluations was not published in the indicated papers, but instead

was estimated by us to correspond to each individual method, assuming an average number of 8 function evaluations for minimizing the function in a given direction. The method of row 5, ref. (7), requires the programming of the analytic expression of the gradient of F; none of the other methods do. The numbers in parentheses in the (min F) column give exponents of the powers-of-ten factors. It is obvious that the method of ref. (10) is the least efficient, while the circular arcs method is the most efficient for the function ROSIE.

Table (3.1A)

k	N	min F	x	y	Method
17	146	6(— 8)	0.999957	0.999938	"Lin. Search", this paper, 1959
—	200	2(— 5)	0.995	0.991	Rosenbrock, ref. (4), 1960
33	(264)	8(— 9)	1.0001	1.0001	Powell, ref. (5), 1962
85	(765)	6(— 5)	1.000653	1.002077	Baer, ref. (10), 1962
18	(144)	1(— 8)	?	?	Flethcer, ref. (7), 1963
12	103	3(— 9)	1.000008	1.000010	"Cir. Arcs", this paper, 1964

Table (3.1B)

	References:				
	(E) Ref. (5)	(G) Ref. (7)	"Linear Search"	"Circ. Arcs"	
k					k
0	2.4+1	2.4+1	2.4+1	2.4+1	0
3	3.6 0	3.7 0	3.5 0	3.5 0	3
6	2.9 0	1.6 0	1.9 0	7.3—2	6
9	2.2 0	7.5—1	8.3—1	8.9—3	9
12	1.4 0	2.0—1	3.4—1	3.4—9	12
15	8.3—1	1.2—2	1.1—1	—	15
18	4.3—1	1 —8	6.0—8	—	18
21	1.8—1	—	—	—	21
24	5.2—2	—	—	—	24
27	4 —3	—	—	—	27
30	5 —5	—	—	—	30
33	8 —9	—	—	—	33

Table (3.1B) Rate of minimization of function ROSIE for four different methods, all starting from (— 1.2, 1). All function values are in the powers-of-ten notation, e.g. 2.4 + 1 = 24 or 3.4 — 9 = 0.0000000034.

k = number of valley points found

The table clearly shows the better rate of convergence of the circular arcs method for the function ROSIE.

Table (3.1C) Rate of minimization of function ROSIE with the circular arcs method for five different starting points (x, y). Bottom row gives the total number, N, of function evaluations. The location of the minimum is at x = 1 and y = 1.

Table (3.1C)

x	—1.200	5.621	—0.221	—2.547	—2.000	x
y	1.000	—3.635	0.639	1.489	—2.000	y
k						k
0	2.4+1	1.2+5	3.6+1	2.5+3	3.6+3	0
1	4.1 0	9.8+2	2.0 0	5.4 0	2.2+2	1
2	3.8 0	2.6+1	8.9—1	3.2 0	2.6 0	2
3	3.5 0	1.5 0	5.5—1	1.4 0	1.4 0	3
4	3.3 0	2.4+2	2.7—1	8.6—1	2.1—2	4
5	3.2 0	1.1—4	1.4—1	2.2—1	5.2—2	5
6	7.3—2	7.5—1	5.8—2	1.9—1	1.7—3	6
7	3.3—3	4.7—1	3.5—2	1.8—1	1.6—3	7
8	8.4—3	5.3+2	1.2—2	7.4—2	1.4—4	8
9	8.9—3	4.7—1	2.6—3	1.9—2	7.5—6	9
10	7.0—7	4.9—1	4.1—4	5.1—3	6.1—8	10
11	4.4—6	3.2—2	1.2—5	2.7—3	3.7—11	11
12	3.4—9	4.2—3	2.6—7	6.4—5	—	12
13	—	2.6—5	6.6—10	3.4—6	—	13
14	—	1.3—7	—	4.3—8	—	14
15	—	5.3—10	—	2.1—11	—	15
16	—	8.9—14	—	—	—	16
N	103	195	93	118	101	N

Table (3.1D)

k	N	F	t	x	y	R	grad | total correction	
0	1	24.2		—1.2000	1.0000			
1	9	4.1	1—2	—1.0298	1.0694		2—1	
2	17	3.8	1—2	—0.9464	0.9043		2—2	
3	25	3.5	1—2	—0.8800	0.7829		2—2	
4	33	3.3	1—2	—0.8266	0.6924		1—2	
5	41	3.2	1—2	—0.7852	0.6256		2—2	
6	59	7—2	1—2	1.2705	1.6151	4.1	5—0	2—0
7	69	3—3	1—2	0.9427	0.8885	1.2	3—1	8—1
8	75	8—3	4—3	1.0918	1.1923	1.3	5—2	3—1
9	82	9—3	4—3	1.0940	1.1972	1.4	3—2	3—1
10	89	7—7	3—3	1.0008	1.0017	7.1	4—4	1—1
11	95	4—6	3—6	1.0021	1.0042	5.8	6—6	3—3
12	103	3—9	7—7	1.000008	1.000010			2—3

Table (3.1D) Details of minimization progress with function ROSIE using circular arcs method and starting at (— 1.2, 1).

k = number of valley point

N = number of function evaluations

F = function value:
for k = 0 . . . 5 in decimal notation,
for k = 6 . . . 12 in powers-of-ten notation,
e.g. "7 — 2" = 0.07.

t = accuracy achieved by F(k),
= tolerance to be satisfied by F(k + 1).

The t-values are calculated from $(F(k) - F(m))/(F(m) + C)$, where $F(m)$ is the lowest available valley point prior to $F(k)$, and where C is a suitably chosen constant (here $C = 1{,}23456$) to avoid the possibility of the divisor tending to zero. These t-values are then used as a tolerance when making the next gradient correction to find $F(k + 1)$. This automatically tightens the tolerance as the minimum is approached.

x, y= coordinates of starting points and all valley points.

R= radius of circle fitted to lowest three or four valley points.

grad correction= distance from predicted point on circular arc to gradient corrected point in valley.

total correction= distance from previous lowest valley point to new valley point.

The first five valley points ($k = 1 \ldots 5$) were found by the starting procedure described in (2.2). The first circle (with radius $R = 4.1$) is a circle through points 3, 4, and 5; its predicted low point was at $x = 6.4$ and $y = 12.2$. It was corrected to give the indicated valley point ($k = 6$). The next valley point ($k = 7$) was obtained from a circle fitted to points 3, 4, 5, 6. Likewise ($k = 8$) was obtained from 4, 5, 6, 7; and ($k = 9$) from 5, 6, 7, 8. Valley point ($k = 10$) was obtained from a circle which was no longer fitted to any of the five starting points, but instead only to the much more accurate points 6, 7, 8, 9; hence, we had a sudden speed-up in the convergence from point 9 to point 10, as evidenced also by the sudden decrease in the ratio of the grad correction to the total correction. Also, the radius R began to approximate more accurately the radius of the osculating circle ($R = 5.59$) of the valley $y = x^2$ at the minimum point (1, 1). Also, the program had automatically switched at this time from the regression plane to the regression line when predicting point 10 on the circular arc (sec. 2.5). The last valley point ($k = 12$) was the result of applying the tail correction (2.6); note the sizeable increase in accuracy.

(3.2) SHALOW

Minimum ($F = 0$) at (1, 1) with valleys along $y = x^2$ and $x = 1$. SHALOW is similar to the function ROSIE, but has a shallow valley compared with the steep valley of ROSIE.

Table (3.2)

x	—2.000	1.184	0.803	0.211	0.820	x
y	—2.000	0.574	—0.251	3.505	4.690	y
k						k
0	4.5+1	7.2—1	8.4—1	1.3+1	1.6+1	0
1	3.0 0	2.1—2	3.3—1	1.1—1	9.5—1	1
2	2.2—1	3.6—6	1.1—1	2.7—1	4.0—1	2
3	5.9—2	4.3—3	4.7—2	4.6—2	1.0—1	3
4	9.0—2	1.1—3	1.3—2	5.6—2	2.1—3	4
5	1.6—2	6.1—4	3.0—3	1.6—2	3.3—2	5
6	4.9—4	7.7—6	9.9—5	2.8—4	3.4—4	6
7	4.6—4	8.1—6	1.1—4	1.9—4	4.4—4	7
8	3.3—4	5.2—8	9.9—6	1.7—4	2.4—5	8
9	4.5—6	2.7—10	6.0—8	2.5—8	5.8—5	9
10	7.1—6	4.7—13	1.1—10	1.2—7	6.7—9	10
11	5.4—7	—	2.8—16	8.4—13	1.1—10	11
N	93	92	86	94	96	N

Table (3.2) Rate of minimization of function SHALOW with the circular arcs method for five different starting points (x, y). Bottom row gives the total number N, of function evaluations.

(3.3) STRAIT

Minimum (F = 0) at (1, 1) with a steep valley along x = 1.

Table (3.3

x	2.000	2.000	2.019	1.992	1.986	x
y	—2.000	—2.322	—1.505	—3.222	5.227	y
k						k
0	1.4+2	1.4+2	1.3+2	1.5+2	9.9+1	0
1	8.4 0	1.0+1	5.8 0	1.7+1	1.7+1	1
2	3.4 0	4.6 0	1.8 0	9.0 0	1.0+1	2
3	1.3 0	2.1 0	3.9—1	5.4 0	7.3 0	3
4	3.0—1	7.3—1	1.6—3	2.9 0	4.8 0	4
5	1.7—2	1.9—1	1.6—1	1.7 0	3.5 0	5
6	5.5—8	6.2—6	9.0—11	1.6—3	3.7—2	6
7	3.4—5	4.6—3	8.9—9	1.9—1	3.8—1	7
8	9.6—9	3.6—6	—	2.2—4	1.2—2	8
9	3.9—5	1.8—8	—	1.9—5	5.1—5	9
10	0.0 0	2.7—14	—	1.6—12	2.7—5	10
11	—	6.4—15	—	0.0 0	5.7—8	11
12	—	—	—	—	2.5—11	12
13	—	—	—	—	0.0 0	13
N	78	83	58	83	95	N

Table (3.3) Rate of minimization of function STRAIT with the circular arcs method for five different starting points (x, y). Bottom row gives the total number, N, of function evaluations.

(3.4) CUBE

Minimum (F = 0) at (1, 1) with a steep valley along $y = x^3$.

Table (3.4)

x	1.200	1.391	1.243	0.284	—1.200	x
y	—2.000	—2.606	—1.974	—3.082	—1.000	y
k						k
0	1.4+3	2.8+3	1.5+3	9.6+2	5.8+1	0
1	4.6 0	1.7+2	8.2+1	2.1 0	4.1 0	1
2	4.8—1	1.8 0	1.2 0	5.4 0	3.8 0	2
3	5.2 0	2.4—3	1.2—1	1.0+1	3.6 0	3
4	1.8—1	3.5—1	7.2—1	1.7—1	3.4 0	4
5	1.3—3	4.8—3	2.3—2	3.8—1	3.3 0	5
6	1.9—2	1.3—5	4.5—2	1.5—1	1.5+2	6
7	2.4—2	1.6—5	3.7—2	6.0—2	3.4 0	7
8	6.3—4	8.5—6	2.5—2	7.4—2	6.7—4	8
9	1.2—6	2.5—9	1.7—2	1.0—1	4.9—3	9
10	6.7—7	8.8—11	2.0—3	7.9—2	1.9—3	10
11	7.5—10	—	5.5—4	1.6—2	6.2—5	11
12	—	—	5.9—5	6.4—3	2.6—6	12
13	—	—	1.8—6	2.4—3	1.9—8	13
14	—	—	3.6—8	2.5—4	4.2—11	14
15	—	—	5.6—11	2.3—5	—	15
16	—	—	—	8.5—7	—	16
17	—	—	—	7.2—9	—	17
18	—	—	—	4.7—12	—	18
N	99	98	113	135	162	N

Table (3.4) Rate of minimization of function CUBE with the circular arcs method for five different starting points (x, y). Bottom row gives the total number, N, of function evaluations.

(3.5) DOUBLE

One minimum (F = 0.099899899) at (x = 1.001005; y = 0), another minimum (F = 0) at (x = 2; y = 0); with one steep valley along r = 1,and another steep valley along r = 2; saddle points around (—2, 0) and (—1, 0).

Table (3.5)

x	—4.000	—3.709	—3.935	—5.392	—7.633	x
y	0.000	—0.957	0.042	—1.708	—3.785	y
k						k
0	3.6+3	2.7+3	3.2+3	2.9+4	2.4+5	0
1	1.6 0	1.5 0	1.6 0	1.5 0	1.4 0	1
2	7.3—1	6.5—1	7.3—1	5.8—1	1.7 0	2
3	5.1—1	1.1 0	7.3—1	7.2—1	2.5 0	3
4	1.8—1	1.5—1	1.8—1	2.2—5	1.0 0	4
5	7.5—1	4.4—1	6.2—1	7.0—2	2.4 0	5
6	2.4—1	1.6—1	2.2—1	4.4—4	1.0 0	6
7	2.6—1	1.5—1	2.3—1	5.2—4	1.1 0	7
8	1.3—1	1.2—1	1.8—1	2.9—5	1.4—1	8
9	1.1—1	1.3—1	1.1—1	1.8—6	1.1—1	9
10	1.0—1	1.0—1	1.0—1	1.5—6	1.2—8	10
11	1.0—1	1.0—1	1.0—1	1.2—6	5.7—6	11
12	1.0—1	1.0—1	1.0—1	—	2.3—6	12
13	1.0—1	1.0—1	1.0—1	—	1.2—8	13
14	1.0—1	—	1.0—1	—	—14	14
15	—	—	1.0—1	—	—	15
N	115	113	118	106	139	N

Table (3.5) Rate of minimization of function DOUBLE with the circular arcs method for five different starting points (x, y). Bottom row gives the total number, N, of function evaluations.

(3.6) HEART

Four minima (all with F = 0) at left (0, 0), at right (20, 0), at top (10, +12.72), and at bottom (10, —12.72). Three valleys: along the x-axis, along x = 10, and along the cardioid r = 10 (1 + cos θ).

Table (3.6)

x	—1.000	—9.870	—11.43	—6.255	—7.539	x
y	0.000	—2.159	—7.344	—21.13	—27.04	y
k						
0	1.0+3	7.3+5	1.1+7	3.1 +7	6.9 +7	0
1	9.5+1	8.8+2	4.1+2	5.3 +1	8.6 +2	1
2	9.9+2	5.2+2	7.6+2	1.0 +3	1.4 +3	2
3	4.8+2	3.1+2	4.2+2	1.9 +1	6.2 +1	3
4	1.6+1	1.9+2	1.2+2	3.9 +2	7.4 +2	4
5	8.3—2	1.1+2	5.8—2	2.5 +2	3.6 +0	5
6	1.8—2	1.4+1	3.4 0	3.0 +1	3.6 +0	6
7	1.4—4	2.2 0	1.8—3	4.5 +0	1.8 —1	7
8	1.3—3	2.3+1	2.2—3	2.4 —8	4.3 —6	8
9	9.1—4	8.7 0	3.7—5	1.2 —2	2.7 —3	9
10	3.1—4	6.2 0	3.5—9	3.3 —10	4.1 —3	10
11	1.7—4	2.1 0	6.7—7	6.9 —4	4.1 —4	11
12	7.7—5	5.6—3	8.2—12	5.7 —11	3.3 —5	12
13	4.8—4	8.3—6	(bottom)	(right)	6.8 —6	13
14	9.7—5	1.7—7	—	—	2.0 —6	14
15	7.3—5	1.7—7	—	—	(bottom)	15
16	7.1—5 (top)	(left)	—	—	—	16
N	149	127	119	121	150	N

Table (3.6) Rate of minimization of function HEART with the circular arcs method for five different starting points (x, y). Bottom row gives the total number, N, of function evaluations.

CONCLUSION

We feel that the foregoing results conclusively show the gain in efficiency obtainable for a direct minimum search procedure when an attempt is made to approximate by suitable curves the valleys of the function to be minimized. The circular arcs method which strives to do so can only be a beginning. Various steps and details of its procedure can and will doubtlessly be improved in the future; most interesting should be the results of its generalization to n variables. It is possible that the gain in efficiency is larger when the circular arcs method is applied to functions with steep-sided valleys than when it is applied to functions with more clearly defined minimum points and less clearly defined valleys. On the other hand, when penalty terms are added to an objective function, in order to also observe inequality constraints, functions with steep-sided valleys will automatically be generated; thus, the methed may become useful in the field of non-linear programming. The authors wish to acknowledge the constant support and encouragement they received from Dr. Wm. J. Schart, Chief of Mathematical Analysis at General Dynamics/Astronautics, and Mr. Carl E. Diesen, Manager of Scientific Programming and Analysis at General Dynamics/Astronautics. We also wish to thank Mrs. Marilee Culver for the efficient and neat typing of the manuscript.

LITERATURE REFERENCES

1. BROWN, R. R.—"A Generalized Computer Procedure for the Design of Optimum Systems." Presented at the Winter General Meeting of the American Institute of Electrical Engineers, New York, N.Y., Feb. 2-7, 1958.
2. JOHNSON, S. M.—"Best Exploration for Maximum is Fibonaccian." RAND Corporation, Santa Monica, Calif., Report No. P-856 (1959).
3. GELFAND, I. M., and TSETLIN, M. L.—"The Principle of Non-local Search in Automatic Optimization Systems." Soviet Physics—Doklady, Vol. 6, No. 3, Sept. 1961, pp. 192-194.
4. ROSENBROCK, H. H.—"An Automatic Method for Finding the Greatest or Least Value of a Function." Computer Journal, October 1960, Vol. 3, pp. 175-184.
5. POWELL, M. J. D.—"An Iterative Method for Finding Stationary Values of a Function of Several Variables." Computer Journal, July 1962, Vol. 5, No. 2, pp. 147-151.
6. FORSYTHE, G. E., and MOTZKIN, T. S.—"Acceleration of the Optimum Gradient Method, Preliminary Report" (Abstract). Bull. Amer. Math, Soc., Vol. 57, 1951, pp. 304–305.
7. FLETCHER, R., and POWELL, M. J. D.—"A Rapidly Convergent Descent Method for Minimization." Computer Journal, July 1963, Vol. 6, No. 2, pp. 163-168.
8. DAVIDSON, W. C.—"Variable Metric Method for Minimization." A. E. C. Research and Development Report, ANL-5990 (Rev.).
9. SHAH, B. V., BUEHLER, R. J., and KEMPTHORNE, O.—"The Method of Parallel Tangents (Partan) for Finding an Optimum." Office of Naval Research, Report NR-042-207 (No. 2).
10. BAER, R. M.—"Note on an Extremum Locating Algorithm." Computer Journal, Oct. 1962, Vol. 5, No. 3, p. 193.
11. KROLAK, P., and COOPER, L.—"An Extension of Fibonaccian Search to Several Variables." Communications of the A.C.M., Oct. 1963, Vol. 6, No. 10, pp. 639-641.