ON THE OPTIMAL SEARCH FOR THE ROOT OF A FUNCTION COMPUTED APPROXIMATELY

MATHEMATICS

1967

SovietRxiv

Abstract

Full Text

UDC 518.12

MATHEMATICS

F. L. CHERNOUSKO

ON THE OPTIMAL SEARCH FOR THE ROOT OF A FUNCTION COMPUTED APPROXIMATELY

(Presented by Academician A. A. Dorodnitsyn, 11 I 1967)

In a number of practical and computational problems it is necessary to find the root of a function which, for any values of the argument, can be computed only approximately. Since computing the function is often associated with a laborious process, it is natural to pose the problem of an optimal algorithm for finding the root. Below, for one class of functions, the formulation and solution of this problem are given. Ideas of the method of dynamic programming are used, which were applied earlier in constructing optimal search algorithms without taking into account errors of computation ([1]).

1°. Definitions and formulation of the problem. Denote by F the class of functions $f(x)$ possessing the following properties: 1) $f(x)$ is defined and continuous on the interval $L = [a, b]$ of the real axis; 2) for any x', x'' from L $(x' \neq x'')$ the inequalities hold

$$m \leq [f(x') - f(x'')]/(x' - x'') \leq M,$$

where m, M are constants $(0 < m < M < \infty)$; 3) the root of the function $f(x)$, unique by condition 2), lies in L. It follows from condition 2) that the derivative $f'(x)$ of the function $f(x)$ exists and satisfies the inequalities $m \leq f'(x) \leq M$ for almost all $x \in L$.

Suppose that in the course of searching for the root the function $f \in F$ has been computed at n points $x_1, \ldots, x_n$ from L, and that the approximate values of the function obtained are $y_1, \ldots, y_n$, while the errors of computation are known and equal to $\delta_1, \ldots, \delta_n$. Thus the function $f(x)$ is subject to the conditions

$$|f(x_i) - y_i| \leq \delta_i \qquad (\delta_i \geq 0,\ i = 1, \ldots, n). \tag{1}$$

The conditions (1) are such that there exists at least one function $f(x) \in F$ satisfying them.

The set of those values x which are roots of functions $f(x)$ from the class F satisfying the conditions (1) will be called the **interval of localization of the root under the conditions** (1) and denoted by D. It can be shown that D is a closed interval, representable in the form

$$D = [a_n, b_n] = L \cap \left\{ \bigcap_{i=1}^{n} [\xi_i^+, \xi_i^-] \right\}; \tag{2}$$

$$\xi_i^+ = \begin{cases} x_i - (y_i + \delta_i)/M & \text{if } y_i \le -\delta_i, \\ x_i - (y_i + \delta_i)/m & \text{if } y_i \ge -\delta_i, \end{cases}$$

$$\xi_i^- = \begin{cases} x_i - (y_i - \delta_i)/m & \text{if } y_i \le \delta_i, \\ x_i - (y_i - \delta_i)/M & \text{if } y_i \ge \delta_i. \end{cases} \qquad (i = 1, \dots, n)$$

The sought root of the function $f(x)$ certainly lies in the interval D and may be located at any point of this interval. Therefore the length of the interval D characterizes the accuracy of determining the root.

It is required to find an optimal algorithm for searching for the root of a function from the class F, i.e., to indicate a method for the successive choice of points $x_1, \dots, x_n$ from L, under which the smallest (in comparison with other methods) length of the interval D at the end of the computations is guaranteed. The interval L, the numbers m, M, the integer n, and the errors $\delta_i \geqslant 0$ for $i = 1, \dots, n$ are assumed given, while the numbers y_i are not known in advance and may turn out to be the "worst" from the point of view of minimizing the length of the interval D. Therefore we require that, under the optimal algorithm, the minimax be attained

$$\Delta = \min_{x_1} \max_{y_1} \min_{x_2} \max_{y_2} \cdots \min_{x_n} \max_{y_n} (b_n - a_n). \tag{3}$$

Here the minima are computed over $x_i \in L$, and the maxima over those y_i for which there exists at least one function $f(x) \in F$ satisfying conditions (1). Obviously, the problem of the optimal search method is here considered as an n-step game between the "computer," which chooses the numbers x_i, and "nature," which chooses y_i, with both sides successively informing one another of the moves made. The payoff is the length of the interval D.

2°. Optimal search algorithm. The problem of the minimax (3) is solved successively, beginning with the determination of x_n, y_n, and all steps turn out to be analogous. As a result of solving problem (3), the following n-step optimal search algorithm is obtained.

Put $a_0 = a$, $b_0 = b$, and describe the i-th step of the algorithm $(i = 1, \dots, n)$. Suppose that after $i - 1$ steps the root-localization interval $[a_{i-1}, b_{i-1}]$, corresponding to the first $i - 1$ inequalities (1), has been found. Set

$$x_i = (a_{i-1} + b_{i-1})/2 \qquad (i = 1, \dots, n) \tag{4}$$

and determine y_i, i.e., the approximate value $f(x_i)$ with known error δ_i. Then we find the root-localization interval after i steps

$$[a_i, b_i] = [a_{i-1}, b_{i-1}] \cap [\xi_i^+, \xi_i^-] \qquad (i = 1, \dots, n). \tag{5}$$

Here ξ_i^+, ξ_i^- are defined by equalities (2). After this the next step is carried out, and so on. At each step the length of the root-localization interval does not increase, and

$$b_i - a_i \leqslant (b_{i-1} - a_{i-1})\, h\left[\frac{2\delta_i}{m(b_{i-1} - a_{i-1})}\right] \qquad (i = 1, \dots, n), \tag{6}$$

where

$$h(z) = \begin{cases} (1-k)/2 + kz, & \text{for } 0 \leqslant z \leqslant 1/2, \\ z, & \text{for } 1/2 \leqslant z \leqslant 1, \\ 1, & \text{for } z \geqslant 1 \end{cases} \qquad (k = m/M). \tag{7}$$

Equality in (6) is attained for those y_i for which the minimax (3) is realized. If, in particular, the function $f(x)$ is computed exactly, then $\delta_i = 0$, and from equalities (6), (7) we obtain

$$\frac{b_i - a_i}{b_{i-1} - a_{i-1}} \leqslant \frac{1}{2}\left(1 - \frac{m}{M}\right), \qquad \frac{b_n - a_n}{b - a} \leqslant \frac{1}{2^n}\left(1 - \frac{m}{M}\right)^n. \tag{8}$$

The optimal algorithm, as follows from (4), reduces to bisecting the root-localization interval. From formulas (8) it is seen that it gives a faster reduction of the root-localization interval than the ordinary bisection method.

3°. Optimal allocation of resources in computations. The approximate values $f(x)$ for each x are determined as a result of some computational or experimental process. For many such processes the error δ in computing the func-

of $f(x)$ at one point is a known function of the resource u spent on the computation: $\delta = R(u)$. The resource may mean the labor or cost of measurements, computer time, etc. Thus, if $f(x)$ for each x is determined as the arithmetic mean of a number of independent measurement results, then $R(u) \sim u^{-1/2}$, where u is the number of measurements. If $f(x)$ for each x is computed as the result of an iterative process converging as a geometric progression with ratio $q < 1$, then $R(u) \sim q^u$, where u is the number of iterations. If computing $f(x)$ for each x requires solving on a computer a Cauchy problem for a

system of ordinary differential equations (for example, this occurs when solving boundary-value problems by selecting the missing initial conditions), then $R(u) \sim u^{-p}$. Here u is the computer time, inversely proportional to the integration step, and the number p depends on the order of the finite-difference scheme.

Let us pose the problem of an optimal search algorithm taking into account the allocation of resources. It is known that $f(x) \in F$; an interval L, numbers m, M, an integer n, a total resource $U \geqslant 0$, and a function $R(u)$, which is defined, nonnegative, and nonincreasing for all $u \geqslant 0$, are given. It is required to indicate a method of successively choosing points x_i from L and numbers $u_i \geqslant 0$, $i = 1, \ldots, n$, for which the minimax

$$\Delta' = \min_{x_1, u_1} \max_{y_1} \min_{x_2, u_2} \max_{y_2} \ldots \min_{x_n, u_n} \max_{y_n} (b_n - a_n). \tag{9}$$

is attained. Here the ranges of the variables x_i, y_i are the same as in the determination of the minimax (3), the interval $[a_n, b_n]$ is specified by equality (2), the errors are $\delta_i = R(u_i)$, and the variables u_i are subject to the constraints

$$u_i \geqslant 0, \qquad \sum_{i=1}^{n} u_i = U \quad (i = 1, \ldots, n). \tag{10}$$

For any u_i satisfying (10) and the corresponding $\delta_i = R(u_i)$, problem (9) reduces to problem (3), whose solution was given above (see (4)–(7)). We shall dwell on the problem of allocating the resources u_i.

Denote by $\Phi_j(s, v)$ the minimal value of the length of the root-localization interval $b_j - a_j$ that can be obtained under the optimal allocation of resources in a j-step search process, if s is the length of the initial interval of root localization and v is the total resource for j steps. Using relation (6), in which equality can be attained, it is not difficult to establish the relations

$$\Phi_j(s, v) = \min_{0 \leqslant u \leqslant v} \Phi_{j-1} \left\{ h \left[\frac{2R(u)}{ms} \right] s,\ v - u \right\}, \qquad \Phi_0(s, v) = s \quad (j = 1, 2, \ldots). \tag{11}$$

With the aid of the change of variables ($\alpha \geqslant 0,\ \beta \geqslant 0$ arbitrary)

$$R(u) = mr(u)/2, \qquad u = tv, \qquad s = r(v)/\sigma,$$
$$\Phi_j(\alpha, \beta) = \alpha g_j(r(\beta)/\alpha, \beta) \quad (j = 0, 1, \ldots) \tag{12}$$

relations (11) are brought to the form

$$g_j(\sigma, v) = \min_{0 \leqslant t \leqslant 1} \left\{ h\left[\sigma \frac{r(tv)}{r(v)}\right] g_{j-1}\left[\frac{\sigma r[(1-t)v]}{r(v)h[\sigma r(tv)/r(v)]},\ (1-t)v\right]\right\}, \qquad (13)$$

$$g_0(\sigma, v) = 1 \quad (j = 1, 2, ...).$$

Here g_j is the ratio of the length of the root-localization interval at the end of the j-step optimal search algorithm to the length of the initial root-localization interval. The calculation of the optimal allocation of resources reduces to computations by the recurrence formulas (13), and simultaneously with $g_j(\sigma, v)$ the value $t = T_j(\sigma, v)$ (generally speaking, nonunique) will be determined, at which the minimum in (13) is attained. The function $T_j(\sigma, v)$ is the fraction of the total resource expended at the first step of the j-step optimal algorithm.

Let the functions $g_j(\sigma, v)$, $T_j(\sigma, v)$, for $\sigma \geq 0$, $v \geq 0$, $j = 1, ..., n$, be found. The optimal search algorithm with allowance for the distribution of resources reduces to the following. Put $v_0 = U$, $a_0 = a$, $b_0 = b$, and describe the i-th step of the n-step algorithm ($i = 1, ..., n$). Suppose that after $i-1$ steps the interval of localization of the root $[a_{i-1}, b_{i-1}]$ has been found and the remaining amount of resource $v_{i-1} \geq 0$ is known. We set $s_{i-1} = b_{i-1} - a_{i-1}$, $\sigma_{i-1} = r(v_{i-1})/s_{i-1}$, $u_i = v_{i-1} T_{n-i+1}(\sigma_{i-1}, v_{i-1})$, $\delta_i = R(u_i)$, $x_i = (a_{i-1} + b_{i-1})/2$, and determine y_i, i.e., the approximate value of the function $f(x_i)$, expending on it the resource u_i. Then we find the interval $[a_i, b_i]$ by formulas (5), (2), and pass to the next step. The length of the interval of localization of the root after n steps will be bounded by the inequality

$$b_n - a_n \leq (b-a) g_n(r(U)/(b-a), U),$$

in which the equality sign can be attained.

Let us note some properties of the functions g_j, T_j. The function $g_j(\sigma, v)$ does not decrease as σ increases for fixed j, v, and does not increase as j increases for fixed σ, v. The functions T_j for $\sigma \geq 1/2$ are discontinuous and may turn out to be equal to zero. The equalities

$$g_1(\sigma, v) = h(\sigma), \qquad T_1(\sigma, v) = 1,$$

$$g_j(0, v) = \left[\frac{(1-k)}{2}\right]^j \quad (j = 1, 2, ...),$$

$$g_j(\sigma, v) = \sigma \text{ for } 1/2 \leq \sigma \leq 1, \qquad g_j(\sigma, v) = 1 \text{ for } \sigma \geq 1. \qquad (14)$$

are valid.

The functions g_j, T_j for $j \geq 2$ and $0 \leq \sigma \leq 1/2$ depend on the function $r(u)$ and on the number k entering into equality (7) for $h(z)$, and their computation by formulas (13) is a rather cumbersome problem.

For a number of important computational and experimental processes we have (A, p are constants)

$$R(u) = Au^{-p}, \qquad r(u) = (2A/m)u^{-p} \qquad (A > 0,\ p > 0). \tag{15}$$

In case (15), the functions g_j, T_j depend only on the single argument σ, which substantially simplifies the problem of tabulating them. Relations (13), with account of (15), give

$$g_j(\sigma) = \min_{0 \leq t \leq 1} \left\{ h(\sigma t^{-p})\, g_{j-1}\left[\frac{(1-t)^{-p}\sigma}{h(\sigma t^{-p})} \right] \right\}, \qquad g_0(\sigma) = 1 \quad (j = 1, 2, \ldots). \tag{16}$$

For sufficiently small σ, one can obtain an analytic solution of equations (16)

$$g_j(\sigma) = \left(\frac{1-k}{2}\right)^j + k\left(\frac{1-q^j}{1-q}\right)^{p+1}\sigma, \qquad T_j(\sigma) = \frac{q^{j-1}(1-q)}{1-q^j},$$

$$q = \left(\frac{1-k}{2}\right)^{1/(p+1)} \qquad (j = 1, 2, \ldots).$$

The functions g_j, T_j were computed on a computer by formulas (16), taking into account equalities (7), (14). For illustration we give some results for $k = 0.1$, $p = 1$

σ	0	0.1	0.2	0.3	0.4	0.5	1
$g_2(\sigma)$	0.2025	0.230	0.303	0.454	0.490	0.5	1
$T_2(\sigma)$	0.40	0.40	0.33	0.33	0	0	0

Computing Center
Academy of Sciences of the USSR

Received
9 I 1967

CITED LITERATURE

1. R. Bellman, S. Dreyfus, *Applied Problems of Dynamic Programming*, "Nauka," 1965.

Note: Figure translations are in progress. See original paper for figures.