Correspondence

Convex differentiable curves

Sir,

G. M. Phillips (1968) describes two algorithms for the piecewise minimax approximation of convex differentiable curves, by straight lines. We (1968) have taken a somewhat more general approach, in that we do not assume differentiability or convexity at the start.

Let us specify an approximation problem P by the function $F = \{\langle x,y \rangle\}$ to be approximated as well as the value ϵ of the upper bound of the approximation error, and let n be the minimum number of straight line segments required to solve the problem. Then, given two problems P_1 and P_2, we shall say that P_1 is at least as simple as P_2 if $n_1 \leqslant n_2$. If P_1 and P_2 are such that the set inclusion $F_1 \subset F_2$ holds and if $\epsilon_1 = \epsilon_2$, then clearly P_1 is at least as simple as P_2. For all we have to do is to solve P_2 and to restrict the solution to F_1.

The piecewise linear approximation of a function F, with a given upper bound (ϵ) of the approximation error proceeds as follows.

Let a and b be defined as

$$a = \underset{\langle x, y \rangle \epsilon F}{Inf} \; x \qquad b = \underset{\langle x, y \rangle \epsilon F}{Sup} \; x$$

We determine the first straight line segment l_1 so as to maximise the range covered by it, i.e. we determine the coefficients p_1 and q_1 of its equation

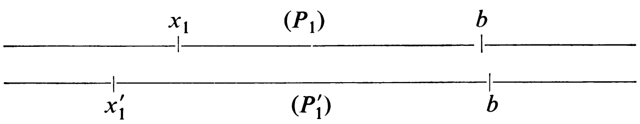
$$y^* = p_1 x + q_1$$

and the abscissa x_1 of its endpoint so that

(1) for all $\langle x, y \rangle \epsilon F$ such that $a \leqslant x < x_1$
we have $y - \epsilon \leqslant y^* \leqslant y + \epsilon$

(2) x_1 is as large as possible.

To show that l_1 is optimal, let l_1' be a straight line segment with parameters p_1', q_1' and x_1', satisfying the corresponding condition (1) but with $x_1' < x_1$. Consider the two approximation problems P_1 and P_1' on the remaining part of F, respectively between x_1 and b and between x_1' and b. Then obviously, P_1 is at least as simple as P_1' because $F_1 \subset F_1'$ and $\epsilon_1 = \epsilon_1' = \epsilon$.



Starting out from x_1 we proceed similarly with P_1, computing p_2, q_2 and x_2, etc. . . . Optimality of the algorithm follows by induction. Here 'optimality' means that for fixed ϵ the integer n cannot be decreased, but this does not exclude the possibility of decreasing ϵ without increasing n. It should also be pointed out that the approximating function is in general discontinuous at the abscissae $x_1, x_2, \ldots, x_{n-1}$.

When F is convex and differentiable, the construction of the segments l_i becomes particularly simple and the discontinuities disappear.

Define F^+ and F^- as

$$F^+ = \{\langle x, y + \epsilon \rangle\} \qquad F^- = \{\langle x, y - \epsilon \rangle\}.$$

Consider the strip between F^+ and F^- and assume that F is such that this strip lies on the hollow side of F^-. Then the straight line segments l_i and their intercepts with F^-, the points A_i, can inductively be determined as follows:

A_0 has abscissa a
l_i is the tangent to F^+ drawn from A_{i-1}.

The approximating function is therefore a convex polygonal line with vertices A_i on F^-. This construction is still applicable when F is convex, continuous and piecewise differentiable, provided we generalise slightly the notion of *tangent*.

Yours faithfully,

D. Bruyendonck
D. Hirschberg

IBM Belgium
Brussels

3 October 1968

References

Fossoul, E., Godesar, R., Haubert, P., Hirschberg, D., and Morlet, E. (1968). Application of Dynamic Programming to long-range planning of nuclear power. *European meeting of TIMS* (*The Institute of Management Science*). Amsterdam.

Phillips, G. M. (1968). Algorithms for piecewise straight line approximations, *The Computer Journal*, Vol. 11, No. 2, pp. 211–212.

Sir,

We write in reply to Professor Barron's letter (this *Journal* Vol. 12, p. 105) commenting on our use of the word 'processor' in our article on MLS (this *Journal* Vol. 11, p. 256).

In our opinion, the term 'processor' may be applied equally to software as well as to hardware. We cite, for example, the fact that the concept of software processes is central to the philosophy of the Multics system (Saltzer, 1966).

To the user of MLS, each phase he enters processes some text. Intuitively, therefore, we are led to use the term 'processor' for the program used in such a phase. Although this may be a systems program, i.e. a program provided by the management, it may also be a private preprocessor belonging to the user. Furthermore, it is worth noting that not all systems programs are processors in the MLS sense.

Thus while agreeing with Professor Barron that the word 'processor' has an established use to describe an item of hardware, we feel that this is by no means its only use in the computing world and to restrict the word to this meaning is an extravagant waste of a useful piece of terminology. It is up to the user of any technical term to define carefully, as we did, precisely what he means by it. To rely on 'established connotations' in a fast moving technology is indeed to tread on thin ice.

Yours faithfully,

J. Larmouth
C. Whitby-Strevens

University Mathematical Laboratory
Cambridge

19 February 1969

Reference

Saltzer, J. H. (1966). Traffic control in a Multiplexed Computer System—MAC–TR–30, Massachusetts Institute of Technology.