Algorithms for piecewise straight line approximations

By G. M. Phillips*

A simple algorithm is described for obtaining approximations to a given function, by means of straight line segments, with any pre-assigned accuracy. The approximation obtained is a best one in the minimax sense. Secondly, given the number of segments k another algorithm finds the best approximation by means of k segments.

(First received November 1967)

1. Introduction

Stone (1961) gives an algorithm for finding best approximations to a function $f(x)$ on a finite interval $[a, b]$ by k straight line segments, using the least squares norm. He justifies the usefulness of his algorithm by showing how it may be applied in the solution of certain non-linear programming problems. In the examples given by Stone, $f''(x)$ is of constant sign. Nearly all functions of practical interest satisfy this condition at least piecewise. Here I describe two algorithms for approximating to such functions by straight line segments, using the minimax norm.

2. Approximations with a given maximum error

If $f''(x) > 0$ on $[\alpha, \beta]$ it is well known that the straight line, $px + q$, which minimises the maximum error $|f(x) - (px + q)|$ on $[\alpha, \beta]$ satisfies

$$f(\alpha) - (p\alpha + q) = \epsilon \qquad (1)$$

$$f(\xi) - (p\xi + q) = -\epsilon \qquad (2)$$

$$f(\beta) - (p\beta + q) = \epsilon \qquad (3)$$

$$f'(\xi) - p = 0. \qquad (4)$$

In these equations, ξ is an interior point of $[\alpha, \beta]$ and ϵ is the maximum error. Given α and β, the equations have a unique solution for p, q, ξ and ϵ. (See, for example, Davis, 1963.) However, here I shall wish to use these equations in a different way, as in the following lemma.

Lemma. Given α and ϵ, the equations (1)–(4) have at most one solution for p, q, ξ and β.

Proof. From equations (1), (2) and (4) we have

$$f'(\xi)(\xi - \alpha) + f(\alpha) - f(\xi) - 2\epsilon = 0. \qquad (5)$$

Let us write this last equation in ξ as $G(\xi) = 0$. Then we can see that

$$G'(\xi) = f''(\xi)(\xi - \alpha),$$

so that $G'(\xi) > 0$ for $\xi > \alpha$. Thus equation (5) has at most one solution $\xi > \alpha$. Since from (5) $G(\alpha) < 0$, a solution ξ of (5) will exist on $[\alpha, b]$ if and only if $G(b) \geqslant 0$. If (5) does have a solution, we may find p from (4) and q from (1). Equation (3) then provides an equation to determine β, say $H(\beta) = 0$, and

$$H'(\beta) = f'(\beta) - f'(\xi) > 0 \text{ for } \beta > \xi.$$

So there is at most one solution for β and, since $H(\xi) < 0$, a solution will exist on $[\xi, b]$ if and only if $H(b) \geqslant 0$. This concludes the proof of the lemma, which will be used repeatedly in the following algorithm.

Algorithm 1. Given any $\epsilon > 0$, we can construct k sub-intervals $[a, x_1]$, $[x_1, x_2]$, . . ., $[x_{k-1}, b]$ and straight lines $p_r x + q_r$, for $r = 1, 2, \ldots, k$, such that on each sub-interval the largest error in approximating to $f(x)$ by the associated straight line is ϵ.

With the notation of the lemma, if $G(b) \geqslant 0$ and $H(b) \geqslant 0$, then given a $(=\alpha)$ and ϵ we can find p_1, q_1, ξ_1 and x_1 $(=\beta)$. The solution of the equations $G(x) = 0$ and $H(x) = 0$ may be most conveniently found by any iterative method which 'brackets' the root at any stage: for example the bisection method or *regula falsi*.

This process may be repeated with x_1 $(=\alpha)$ and ϵ, and so on. At some stage, say with $\alpha = x_{k-1}$, we will find that either $G(b) < 0$ or $H(b) < 0$. The geometrical interpretation of this is that the kth straight line segment with maximum error ϵ overshoots the end-point b. We may choose as $p_k x + q_k$ the straight line which passes through the points $(x_{k-1}, f(x_{k-1}) - \epsilon)$ and $(b, f(b) - \epsilon)$. Thus, given any $\epsilon > 0$, the algorithm obtains a *continuous* approximation to $f(x)$ with maximum error ϵ, and the approximation is clearly effected by the smallest possible number of straight line segments.

3. Best approximations by k segments

Let us now consider the problem of finding the best minimax approximation to $f(x)$ on $[a, b]$ by means of k straight line segments. In this case, k is given; in the previous section ϵ was given. It is obvious that, in the best approximation of this type, each straight line segment must be the best minimax straight line approximation to $f(x)$ on the corresponding interval, and that

* *Department of Applied Mathematics, University of St. Andrews, St. Andrews, Fife.*

the maximum errors attained on every interval must be equal.

Algorithm 2. In Algorithm 1 it is clear that k is a non-increasing function of ϵ, say $k = F(\epsilon)$. We can find lower and upper bounds for ϵ as follows. First choose $\epsilon_0 > 0$ arbitrarily and use Algorithm 1 to calculate $k_0 = F(\epsilon_0)$. If $k_0 > k$, ϵ_0 will be a lower bound for ϵ. We may then set $\epsilon_1 = 2\epsilon_0$ and calculate $k_1 = F(\epsilon_1)$. We may repeat this calculation for k with ϵ_1 replaced by $2\epsilon_1$ until we obtain a value of $k_1 \leqslant k$. This will give an upper bound for ϵ, say ϵ_1. However, if initially we get $k_0 \leqslant k$ we may set $\epsilon_1 = \epsilon_0$ as an upper bound for ϵ and repeatedly halve ϵ_0, calculating $k_0 = F(x_0)$ each time, until we obtain a value of $k_0 > k$, showing that ϵ_0 is a lower bound for ϵ.

Having obtained lower and upper bounds for ϵ, we may refine these by repeated bisection of the interval $[\epsilon_0, \epsilon_1]$, using Algorithm 1 at each stage to calculate $F(\frac{1}{2}(\epsilon_0 + \epsilon_1))$. The process may be terminated when $\epsilon_1 - \epsilon_0$ is sufficiently small. The operation of Algorithm 1 corresponding to the final value of ϵ_0 gives the values of the sub-dividing points x_r and the straight lines $p_r x + q_r$. It may be noted that again the approximating function is continuous.

4. Numerical example

As a numerical example, consider e^x on [0, 1]. **Table 1** gives the best minimax approximation to e^x by four straight line segments obtained by using Algorithm 2 with $k = 4$. The corresponding value of ϵ is 0·006579. *A priori* bounds for ϵ may be obtained from

$$\epsilon = \left(\frac{b-a}{4k}\right)^2 |f''(\xi)|, \tag{6}$$

for some ξ in $[a, b]$. (See, for example, Phillips, 1968.) In this case we have $0{\cdot}0039 < \epsilon < 0{\cdot}0107$. An asymptotic relation connecting ϵ and k may be found in Ream (1961).

Table 1

Algorithm 2 applied to e^x on [0, 1] with $k = 4$

r	x_{r-1}	x_r	p_r	q_r
1	0·000000	0·300570	1·166545	0·993421
2	0·300570	0·561833	1·543487	0·880124
3	0·561833	0·792888	1·973057	0·638777
4	0·792888	1·000000	2·455255	0·256448

References

DAVIS, P. J. (1963). *Interpolation and Approximation*, Blaisdell. (Especially p. 151.)
PHILLIPS, G. M. (1968). Estimate of the maximum error in best polynomial approximations, *The Computer Journal*, Vol. 11, pp. 110–111.
REAM, N. (1961). Note on 'Approximation of Curves by Line Segments', *Mathematics of Computation*, Vol. 15, pp. 418–419.
STONE, H. (1961). Approximation of Curves by Line Segments, *Mathematics of Computation*, Vol. 15, pp. 40–47.

Book Review

Computers and the Human Mind, by DONALD G. FINK, 1968; 301 pages. (*Heinemann*, 30s.)

This book is concerned with cybernetics, and rather especially with the central problem of cybernetics which is that of artificial intelligence. The fact that 'computers' is a word used in the title of this book encourages in some measure the usual confusion in many people that somehow computers are very like human beings.

A sheet of newspaper is like the Magna Carta because both have writing on them and both are made of paper—we are here of course trying to avoid all the usual logical pitfalls over whether the Magna Carta is a concept or set of concepts or something written on paper. Computers can be made to behave very much like human brains if they are suitably programmed, and they can even be made 'self-programming' in some measure. Dangers arise when we point to the *differences* rather than the similarities and even here we may miss the point about the possibility of generating artificially intelligent systems more 'intelligent' than human beings.

Mr. Fink has, in his book, touched on all these problems to a greater or lesser extent and adduces the well known Turing criteria for artificial intelligence. He mentions very recent work by such people as Bert Green and A. L. Samuel on question-answering techniques and on playing draughts by computer. Indeed this is a competent, professional book and must be commended as such.

The disadvantages of the book are that it sometimes adopts a patronising tone, and sometimes it has the ring of journalese about it, as in the following extract:

> 'But brain surgeons and psychologists agree that the brain is so complex an organ, and so difficult to study both inside and outside the skull, that the present body of knowledge is a small island in a sea of ignorance.'

Mr. Fink does not always seem to believe himself in the quest of cybernetics nor does he manage to convey the full significance of the problem of artificial intelligence. A good book, which could have been better if the subject matter were better integrated.

F. H. GEORGE (Bristol)