

MEMORANDUM
RM-2978-PR
FEBRUARY 1962

ON THE APPROXIMATION
OF CURVES BY LINE SEGMENTS USING
DYNAMIC PROGRAMMING - II

Richard Bellman and Bella Kotkin

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The **RAND** *Corporation*
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-2978-PR
FEBRUARY 1962

**ON THE APPROXIMATION
OF CURVES BY LINE SEGMENTS USING
DYNAMIC PROGRAMMING-II**

Richard Bellman and Bella Kotkin

This research is sponsored by the United States Air Force under Project RAND—Contract No. AF 49(638)-700—monitored by the Directorate of Development Planning, Deputy Chief of Staff, Research and Technology, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force. Permission to quote from or reproduce portions of this Memorandum must be obtained from The RAND Corporation.

PREFACE

The theory of dynamic programming, which was developed to handle various problems of optimization, is applied in this Project RAND memorandum to a frequently occurring problem of mathematical representation. The present approach suggests further experiments whereby the method of dynamic programming can be applied to multidimensional problems (such as trajectory problems) that cannot be treated easily within the limitation of the present-day computers.

SUMMARY

In this paper we apply the technique of dynamic programming to approximate a given continuous function $g(x)$ by a finite number of line segments over the interval $[a,b]$. The problem as defined by Stone is to determine the constants $a_k, b_k, k = 0, \dots, N-1$ and the points of division u_1, \dots, u_{N-1} in the interval $[a,b]$ that minimize the function

$$J = \sum_{k=0}^{N-1} \int_{u_k}^{u_{k+1}} (g(x) - a_k - b_k x)^2 dx.$$

We calculate the results for $g = e^{-x}$ by means of a FORTRAN program for the IBM-7090. Moreover, we point out an analytic treatment of the functions $g(x) = x^2$ and $g(x) = e^{-cx}$ that is easily derived by utilizing the functional equation technique of dynamic programming.

CONTENTS

PREFACE	iii
SUMMARY	v
Section	
1. INTRODUCTION.	1
2. FUNCTIONAL EQUATION APPROACH.	2
3. NUMERICAL PROCEDURE	3
4. THE CASE $g(x) = x^2$	5
5. THE CASE $g(x) = e^{-cx}$	7
REFERENCES.	9

ON THE APPROXIMATION OF CURVES BY LINE SEGMENTS
USING DYNAMIC PROGRAMMING—II

1. INTRODUCTION

In a recent paper, Stone [1] considered the problem of approximating a given function $g(x)$, continuous over $[a,b]$, by a finite set of line segments

$$(1.1) \quad h(x) = a_k + b_k x, \quad u_k \leq x \leq u_{k+1}, \quad k = 0, 1, \dots, N-1,$$

where $u_0 = a$, $u_N = b$. The measure of accuracy of approximation is taken to be

$$(1.2) \quad J = \sum_{k=0}^{N-1} \int_{u_k}^{u_{k+1}} (g(x) - h(x))^2 dx.$$

If the points u_1, \dots, u_{N-1} are fixed in advance, the determination of the parameters a_k and b_k , $k = 0, 1, \dots, N-1$, is a simple matter. If we introduce an adaptive feature by allowing the u_k to depend upon the function $g(x)$ we intend to approximate, then the determination of the values of the a_k , b_k and u_k which minimize J becomes very much more involved.

In his paper, Stone points out that the case $g(x) = x^2$ can be resolved analytically, and presents a numerical algorithm and results for the case $g(x) = e^{-cx}$. His methods are based upon the conventional techniques of calculus. In a short note [2], we pointed out that the problem could be treated very simply and directly by dynamic

programming techniques [3], and for more general measures of deviation such as

$$(1.3) \quad J_M = \sum_{k=0}^{N-1} \max_{u_k \leq x \leq u_{k+1}} |g(x) - h(x)|.$$

In this paper we wish to present the results of some calculations for the case $g(x) = e^{-cx}$ and to indicate how the dynamic programming approach yields specific analytic results for the functions x^2 and e^{-cx} .

2. FUNCTIONAL EQUATION APPROACH

If we regard the minimum value of J as a function of the endpoint b and the number of intermediate points u_1, u_2, \dots, u_{N-1} , we may write

$$(2.1) \quad f_{N-1}(b) = \min J,$$

for $N = 2, 3, \dots$

If we introduce the function

$$(2.2) \quad h(c, d) = \min_{\alpha, \beta} \int_c^d (g(x) - \alpha - \beta x)^2 dx,$$

$a \leq c \leq d \leq b$, an application of the principle of optimality [3] yields the functional equation

$$(2.3) \quad f_N(b) = \min_{a \leq u_N \leq b} [h(u_N, b) + f_{N-1}(u_N)], \quad N \geq 2,$$

$$f_1(b) = \min_{a \leq u_1 \leq b} [h(u_1, b) + h(a, u_1)].$$

The problem of minimizing J has thus been reduced to that of determining the sequence of functions $\{f_N(b)\}$, a relatively simple task.

3. NUMERICAL PROCEDURE

The values of α and β which minimize in (2.2) are easily seen to be

$$(3.1) \quad \alpha = \frac{\begin{vmatrix} A & E \\ B & F \end{vmatrix}}{\begin{vmatrix} A & B \\ B & C \end{vmatrix}}, \quad \beta = \frac{\begin{vmatrix} E & B \\ F & C \end{vmatrix}}{\begin{vmatrix} A & B \\ B & C \end{vmatrix}},$$

where

$$(3.2) \quad A = \int_c^d x^2 dx, \quad B = \int_c^d x dx, \quad C = \int_c^d dx,$$

$$D = \int_c^d g(x)^2 dx, \quad E = \int_c^d xg(x) dx, \quad F = \int_c^d g(x) dx.$$

For these minimizing values of α and β we have

$$(3.3) \quad h(c,d) = \frac{\begin{vmatrix} D & E & F \\ E & A & B \\ F & B & C \end{vmatrix}}{\begin{vmatrix} A & B \\ B & C \end{vmatrix}}.$$

The presence of these determinants introduces some numerical difficulties which can be resolved in most cases by the use of double precision.

The actual computational procedure is as follows:

Stage 1. Insert a grid of equally spaced points in the interval a to b . For a fixed x selected successively along the grid, $a < x \leq b$, find the value of u_1 along the grid from a to x yielding the minimum value $f_1(x)$. The computation of $h(a,x)$ and $h(u_1,x)$ is described in Sec. 2. Store $u_1(x)$ and $f_1(x)$ for all values of x .

Stage 2. For a fixed x selected successively along the grid, find the value of u_2 along the grid from a to x furnishing the minimum $f_2(x)$. Again, $h(u_2,x)$ is computed as above, and $f_1(u_2)$ is obtained from the values stored in stage 1. Store $u_2(x)$, $f_2(x)$ for all x in the storage allotted to $u_1(x)$, $f_1(x)$.

It is clear that stage i for $i > 2$ follows the format of stage 2, and uses the stored values of $f_{i-1}(u_{i-1})$.

A FORTRAN program for the IBM-7090 was devised to test this method for the function $g(x) = e^{-x}$, $0 \leq x \leq 3$ and the values obtained checked with those given by Stone in [1]. As mentioned before, this test case required double precision arithmetic, which meant a significant increase in computing time. For a grid size of .01 involving 301 points along the grid, the time required using double precision is approximately 10 minutes per stage. This is to be compared with a time of about one minute per stage if double precision is not used. Unfortunately, the accuracy is then too poor.

At the termination of each iterative stage the values $u_i(b)$, $f_i(b)$ are put out in some form so that by cross-referencing the printed tables we obtain the solution. The following schematic table indicates the method for obtaining u_5, u_4, u_3, u_2, u_1 for a five-stage process yielding six straight line segments.

Table

b	u_1	f_1	u_2	f_2	u_3	f_3	u_4	f_4	u_5	f_5
a										
u_2	u_1	f_1								
u_3			u_2	f_2						
u_4					u_3	f_3				
⋮										
u_5							u_4	f_4		
⋮										
b									u_5	f_5

4. THE CASE $g(x) = x^2$

Let us now briefly note that the functional equation approach yields the explicit analytic solution quite readily for the case $g(x) = x^2$. We begin with the observation that

$$(4.1) \quad h(a,b) = \min_{\alpha, \beta} \int_a^b (x^2 - \beta x - \alpha)^2 dx$$

$$= h(0, b - a),$$

a result which is apparent upon a change of variable $x = y + a$, and then that

$$(4.2) \quad h(0,b) = b^5 h(0,1),$$

as a change of variable $x =$ by shows. Thus,

$$(4.3) \quad h(a,b) = (b - a)^5 h(0,1).$$

It now follows in precisely the same fashion that

$$(4.4) \quad f_N(b) \equiv f_N(a,b) = (b - a)^5 f_N(0,1).$$

Furthermore, the principle of optimality yields the relation

$$(4.5) \quad f_{2N}(b) \equiv f_{2N}(a,b) = \min_{a \leq c \leq b} [f_N(a,c) + f_N(c,b)]$$

$$= \min_{a \leq c \leq b} [(c - a)^5 + (b - c)^5] f_N(0,1).$$

Hence

$$(4.6) \quad f_{2N}(0,b) = \frac{b^5}{2^4} f_N(0,1),$$

whence, inductively,

$$(4.7) \quad f_{2^n}(0,b) = \frac{b^5}{2^{4n}} f_1(0,1).$$

We see then that for $N = 2^n$, $f_N(0,b) = b^5 f_1(0,1)/N^2$, a relatively poor degree of approximation. For other values of N we can also use the foregoing to find the explicit form of f_N and the location of the minimizing values of the u_i .

5. THE CASE $g(x) = e^{-cx}$

The same type of reasoning as above shows that

$$(5.1) \quad h(a,b) = \min_{\alpha, \beta} \int_a^b (e^{-cx} - \alpha - \beta x)^2 dx$$

$$= e^{-ac} h(0, b-a),$$

and similarly that

$$(5.2) \quad f_N(a,b) = e^{-ac} f_N(0, b-a).$$

Hence,

$$(5.3) \quad f_{2N}(0,b) = \min_u [f_N(0,u) + f_N(u,b)]$$

$$= \min_u [f_N(0,u) + e^{-cu} f_N(0, b-u)].$$

Generally,

$$(5.4) \quad f_{N+M}(0,b) = \min_u [f_N(0,u) + e^{-cu} f_M(0, b-u)].$$

Consequently, if we were interested only in obtaining the approximations to e^{-cx} , we could obtain the numerical results mentioned above by a change of variable from the results for e^{-x} .

REFERENCES

1. Stone, H., "Approximations of Curves by Line Segments," Mathematics of Computation, Vol. 15, No. 73, 1961, pp. 40-47.
2. Bellman, R., "On the Approximation of Curves by Line Segments Using Dynamic Programming," Comm. Assoc. Computing Machinery, Vol. 4, 1961, p. 284.
3. Bellman, R., and S. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.