- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

372

NO

# National Aeronautics and Space Administration

HOUSTON, TEXAS

RECEIVED
AUG 1970
NASA STI FACILITY
INPUT BRANCH

FACILITY FORM 602

N70-35593 (ACCESSION NUMBER)

9 (PAGES)

TMX-65000 (NASA CR OR TMX OR AD NUMBER)

(THRU)

1 (CODE)

19 (CATEGORY)

Manned Spacecraft Center

NASA

MSC-ED-R-67-58
(Revised Jan. 1968)

MSC INTERNAL TECHNICAL NOTE

# AN ALGORITHM FOR INTERPOLATION AND NUMERICAL DIFFERENTIATION

By

John L. Engvall

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

September 13, 1967

70100

MSC INTERNAL NOTE NO. ED-R-67-58

(Revised Jan. 1968)

# AN ALGORITHM FOR INTERPOLATION AND NUMERICAL DIFFERENTIATION

Prepared by: John L. Engvall

Approved by: Henry P. Decell, Jr.
Chief, Theory and Analysis Office

Approved by: Eugene H. Brock
Chief, Computation and Analysis Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
Houston, Texas
September 13, 1967

# CONTENTS

## INTRODUCTION

The purpose of this paper is to present an easily automated technique for computing a continuous function to approximate a discrete set of data. An algorithm is outlined describing a weighted least squares solution involving the approximation for the dependent variable and the desired derivative of the approximating function. This paper will be concerned indirectly with numerical differentiation. The algorithm can be applied directly but requires a few more computations than standard least squares.

## NUMERICAL APPROXIMATIONS

Discrete data sets are often replaced by functional approximations for the purpose of interpolation, integration, or differentiation. None of these processes are precise. For example, given any finite set of ordered pairs of real numbers $(x_i, y_i)$ , $i = 1,N$ , and any finite interval $[a,b]$ , there is an infinite collection of continuous, differentiable, integrable functions $F$ defined on $[a,b]$ such that, for every function $f$ in $F$ , $f(x_i) = y_i$ . Moreover, for every $x_o$ not in the original data set $x_i$ , there is an infinite collection of functions in $F$ all having distinct values at $x_o$, distinct derivatives at $x_o$, and distinct integrals over $[a,b]$. This implies that any numerical algorithm for interpolation, differentiation, or integration of finite data sets yields an answer that is at best a "correct answer" among an infinite number of "correct answers". Because of this fact, most texts describe algorithms that require some additional properties of the data or present algorithms which appeal to the intuition of the user. This paper utilizes the latter alternative.

## A MINIMIZATION TECHNIQUE

In all that follows it will be assumed that a fixed set of data $(x_i, y_i)$ for $i = 1, N$ has been defined. For a fixed set of functions $f_1, f_2, \cdots f_k$ and the function $f = \sum_{i=1}^{k} a_i f_i$ the least squares norm of $f - y$ will be defined by

$$||f - y|| = \sqrt{\sum_{i=1}^{N} (f(x_i) - y_i)^2} \qquad (1)$$

and the derivative of $f$ with respect to $x$ will be denoted by $f'$. Least squares approximations of this type often exhibit "undesirable behavior" between the input data points.

Lagrange and Hermite interpolating polynomials provide approximations exact at the input points, but they do not provide any smoothing of the data. The following algorithm is similar to the Hermite interpolation and the least squares smoothing process.

Given two nonnegative weight functions $w_1(x)$ and $w_2(x)$, minimize

$$||w_1(f - y) + w_2(f' - \hat{y})|| \tag{2}$$

where $\hat{y}_i$ is the desired derivative of $f$ at $x_i$. A function $f$ minimizing (2) thus combines the requirements that $f$ approximates $y$ and the slope of the curve $f(x)$ approximates the desired slope at each of the initial datum points.

Rewriting (2) in the form

$$||w_1 f + w_2 f' - (w_1 y + w_2 \hat{y})|| \tag{3}$$

and considering first the restriction of $f$ to a polynomial of degree $k-1$, we see that the minimization problem can easily be solved as follows:
Define an $N \times k$ matrix $B$ where the last $k-1$ columns are

defined by

$$b_{ij} = w_1(x_i)\, x_i^{j-1} + (j-1)\, w_2(x_i)\, x_i^{j-2} \qquad (4)$$

and the first column is defined

$$b_{i1} = w_1(x_i) \qquad (5)$$

Let $z$ be a column vector such that

$$z_i = w_1(x_i)\, y_i + w_2(x_i)\, \hat{y}_i \qquad (6)$$

If $B^T B$ is nonsingular let

$$a = (B^T B)^{-1} B^T z \qquad (7)$$

and if $B^T B$ is singular let

$$a = B^+ z \qquad (8)$$

where $B^+$ is the generalized inverse of $B$ . Then a function $f$ minimizing (3) is given by

$$f(x) = \sum_{i=1}^{k} a_i\, x^{i-1} \quad .$$

The only restriction needed for the class of functions to insure a simple solution is that for each function $f_i$, $f_i = f'_j$ for some $j$. For example if $f$ is restricted to

$$f(x) = a_1 + a_2 x + a_3 x^2 + a_4 \sin(2x) + a_5 \cos(2x)$$

then the first three columns of $B$ are defined as in (4) and (5), and the other columns are defined by

$$b_{i4} = w_1(x_i) \sin(2x_i) + 2w_2(x_i) \cos(2x_i)$$

$$b_{i5} = w_1(x_i) \cos(2x_i) - 2w_2(x_i) \sin(2x_i)$$

The vector $z$ remains the same and the solution is obtained in exactly the same manner.

## CONCLUSION

This algorithm has been programed and used for several applications. It should be pointed out that orthogonal polynomials can be used to obtain the solution to Equation (7) when a polynomial approximation is computed for $w_1 = w_2 = 1$. Also, notice that when $w_1$ (or $w_2$) is constant valued only one value should be reserved in computer memory. Another option that can be implemented is to use the average value of the slopes of the left and right straight line segments between adjacent points for the value of $\hat{y}_i$ . In this case, core storage for the $\hat{y}$ array need not be reserved.