DOCUMENTATION OF PROGRAMMES ORDERING THE SERIES AND INTERPOLATION

S RAMASESHAN
DIRECTOR

STUDY GROUP

S M SETH
P NIRUPAMA

NATIONAL INSTITUTE OF HYDROLOGY
JAL VIGYAN BHAVAN
ROORKEE - 247667 (UP)
INDIA

1983-84

CONTENTS

CONTENTS (continued)

List of Symbols

X_i	:	X array
Y_i		Y array
$C_{i,k}$		Spline constants
S_{i-1}, S_i, S_{i+1}		Three pivotal points
A_0, A_1, A_2		Coefficients
$Q(x)$		Interpolated value at desired point x
$f(x)$		Function of x
$a_0, a_1, \ldots\ldots, a_{n-1}$		n unknowns
$P_j(x)$		m^{th} degree polynomial
$P_m(x)$		The linear combination of $P_j(x)$
h		Equal interval between two pivotal points
C_s		Coefficient of skewness
K_n		Standard normal deviate
K		Frequency factor

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The documentation for ordering the series describes the comparative studies carried out using four subroutines available in the literature. The comparison is made on the basis of compilation time, run time and memory requirements of the programme. The mean and standard deviation of run times are also compared. The test data used are ten different series of 600 real numbers each generated by a random number generation subroutine. The study shows that the subroutine ORDER2 which uses the principle of 'division and comparison' takes minimum average run time. Also it requires less total time including compilation and execution times.

The input description and the subroutine listing are given in the documentation.

The documentation for interpolation describes the use of three subroutines. The methods employed in these subroutines are (1) spline fit, (2) second order parabolic, and (3) Lagrange's interpolation. No internal data storage is required for any of the three subroutines listed in the documentation. Input data requirements are specified. Example input and output of the subroutines are listed in the documentation.

ORDERING THE SERIES

A.1.0 INTRODUCTION

Arranging the series in order, plays an important role in the area of statistics. It is generally used for making a pictorial representation in the form of frequency histogram or cumulative frequency polygon of an experimental outcome. When the number of measurements X_1, X_2 ....X_n becomes large (say 25 or more) ungrouped measurements become too cumbersome to deal with. Grouping the data simplifies the matter. In such a case the least 'L' and greatest 'G' values of the measurements are found and the range (G-L) is divided into some number K of equal intervals which is known as 'class interval'. These groups are used for the construction of the frequency histogram and cumulative frequency polygon.

Arranging the series in order is also useful in hydrological studies for the purpose of flood frequency analysis, development of flow duration curve etc. In the present study the available programmes for ordering the series have been studied for identifying the most efficient one. The subroutines studied herein are meant for arranging the given set of real numbers in ascending/descending order.

A.2.0 METHOD USED FOR VARIOUS SUBROUTINES

A.2.1 ORDER1

Here the given series is being arranged in descending/ ascending order in the following manner: first of all the comparison is made between two consecutive numbers and in case the j^{th} number is less than $(j+1)^{th}$ number then $(j+1)^{th}$ number is replaced by j^{th} and the j^{th} number is replaced by $(j+1)^{th}$ number. A temporary location is used for this operation. Thus the smallest number in the given series finds its position in the last location of the given series. The rest of the numbers are subjected to the same procedure and the next smallest number occupies the last but one position in the given series. Thus the series is arranged in descending order. The subroutine which uses this procedure for arranging the series in order is given in appendix I.

A.2.2 ORDER2

Here the series is divided in two equal parts and then the first number of the first part has been compared with the first number of the second part, the second number of the first part is being compared with the second number of the second part and so on. In this process the number in the first part is replaced by the number in the second part of the series if former is lesser than the latter, otherwise not. Such 'division and comparison' process is repeated until the given series is arranged in descending order. If the series has to be arranged in ascending order, then one has to supply the argument KO other than zero. In this case the same descending order series is arranged in the reverse order. The subroutine which employes this procedure is given in appendix II.

A.2.3 ORDER3

The procedure adopted for arranging the series in descending order is similar to that of subroutine ORDER1 except that the already compared and properly placed numbers are compared repeatedly until the first number is replaced by the largest number in the given series. The subroutine based on this procedure is given in appendix III.

A.2.4 ORDER4

The first number of the given series to be arranged, say in descending order, is compared with the rest of the numbers in the same series and this number is replaced by a larger number next found in the series. The first number taken for comparison occupies the location of the larger number. A temporary location is used for this operation. Now that larger number is compared with the rest of the numbers which were not being compared previously. If any further larger number is found, then it is replaced by the newly found larger number . The same process is repeated until the first number of the series is replaced by the largest number. This procedure is repeated for the rest of the numbers also, until the descending series is formulated from the given series. The subroutine based on this procedure is given in appendix IV.

A.3.0 COMPUTER SUBROUTINES

All the four subroutines used in this study are written in FORTRAN IV language and they have been adapted from the following sources:

Sl.No.	Source of the programme	Subroutine	Reference No.
1.	Frequency and Risk Analysis in Hydrology	ORDER1	2
2.	'FORTRAN-HYDRO' a documentation of computer programmes	ORDER2	4
3.	'Computer Programming in FORTRAN IV'	ORDER3	3
4.	'HECWRC' - Programme for flood flow frequency analysis	ORDER4	1

A.4.0 INPUT SPECIFICATION, OUTPUT DESCRIPTION AND TEST DATA

There is no internal data storage requirement for any of the four subroutines for ordering the series. Data input is any array containing more than one element i.e. a series of real or/and integer numbers having at least two numbers.

A set of 600 random real numbers have been used for testing the four subroutines, and they are listed in table 1.

Table 1 - 600 Random Numbers used in this Study

TOTAL NUMBERS = 600

0.464	0.060	1.486	1.022	1.394	0.906
1.179	-1.501	-0.690	1.372	-0.482	-1.376
-1.010	-0.005	1.393	-1.787	-0.105	-1.339
1.041	0.279	-1.805	-1.186	0.658	-0.439
-1.399	0.199	0.159	2.273	0.041	-1.132
0.768	0.375	-0.513	0.292	1.026	-1.334
-0.287	0.161	-1.346	1.250	0.630	0.375
-1.420	-0.151	-0.309	0.424	0.593	0.862
0.235	-0.853	0.137	-2.526	-0.354	-0.472
-0.555	-0.513	-1.055	-0.488	0.756	0.225
1.678	-0.150	0.598	-0.899	-1.163	-0.261
-0.357	1.827	0.535	-2.056	-2.008	1.180
-1.141	0.358	-0.230	0.208	0.272	0.606
-0.307	-2.098	0.079	-1.658	-0.344	-0.521
2.990	1.278	-0.144	-0.886	0.193	-0.199
-0.537	-1.941	0.489	-0.243	0.531	-0.444
0.658	-0.885	-0.628	0.402	2.455	-0.531
-0.634	1.279	0.046	-0.525	0.007	-0.162
-1.618	0.378	-0.057	1.356	-0.918	0.012
-0.911	1.237	-1.384	-0.959	0.731	0.717
-1.633	1.114	1.151	-1.939	0.385	-1.083
-0.313	0.606	0.121	0.921	-1.473	-0.851
0.210	1.266	-0.574	-0.568	-0.254	-0.921
-1.202	-0.288	0.782	0.247	-1.711	-0.430
0.416	0.593	-1.127	-0.142	-0.023	0.777
-0.323	-0.194	0.697	3.521	0.321	0.595
0.769	-0.136	-0.345	0.761	-1.229	-0.361
1.598	-0.725	1.231	1.046	0.360	0.424
1.377	-0.873	0.542	0.882	-1.210	0.891
-0.649	-0.219	0.084	-0.747	0.790	0.145
0.034	0.234	-0.736	1.206	-0.491	-0.109
0.574	-0.509	0.394	1.810	0.060	-0.491
-1.186	-0.762	-1.541	0.993	-1.407	-0.504
-0.463	0.833	-0.068	0.543	0.926	0.571
2.945	0.881	0.971	1.033	-0.511	0.181
-0.486	-0.256	0.065	1.147	-0.199	-0.508
-0.992	0.969	0.983	-1.096	0.250	1.265
-0.927	-0.227	-0.577	-0.291	-2.828	0.247
-0.584	0.446	-2.127	-0.656	1.041	-0.899
-1.114	-0.515	-0.451	1.410	-1.045	1.378
0.499	0.665	0.754	0.298	1.456	-0.106

-1.579	0.532	-0.899	0.410	0.296	-1.558
1.375	-1.851	1.974	-0.934	0.712	0.203
-2.051	-0.736	0.856	-0.212	0.415	-0.121
-0.246	-1.630	-0.116	-1.141	-1.330	-1.396
-0.166	-0.202	0.425	0.602	0.237	1.221
-0.439	1.291	0.541	-1.661	0.665	0.340
0.008	0.110	1.297	-0.566	-1.181	-0.518
0.843	0.584	-0.431	-0.135	-0.732	1.049
2.040	0.116	-1.616	1.381	-0.394	-0.349
-0.288	0.187	0.785	0.194	-0.258	1.879
1.090	0.448	-0.457	0.960	-0.491	0.219
-0.169	1.096	1.239	-0.146	-1.693	-1.041
1.620	1.047	0.032	0.151	0.290	0.873
-0.289	1.119	-0.792	0.063	0.484	1.045
0.084	-0.086	0.427	-0.528	-1.433	2.923
-1.190	0.192	0.942	1.216	1.705	-0.145
-0.066	1.810	-0.124	0.484	1.458	0.022
-0.538	-1.094	1.298	-1.190	-0.963	1.192
0.412	0.161	-0.631	0.748	-0.218	-1.530
-1.983	0.779	0.313	0.481	-2.574	-0.292
-2.832	0.362	-1.040	0.089	0.079	-0.376
-0.902	-0.437	0.513	0.004	-1.275	-1.793
-0.986	-1.363	-0.880	-0.158	-0.831	-0.813
-1.345	0.500	-0.318	-0.438	1.045	0.733
1.164	-0.498	1.006	2.885	0.196	-1.272
1.262	-0.281	1.707	0.580	0.241	0.022
-0.853	-0.501	0.439	-1.885	-0.255	-0.423
0.857	-0.260	-2.330	0.953	-0.973	-1.691
-0.558	-0.627	-1.108	-1.726	0.524	-0.573
0.471	-0.310	0.610	-0.220	0.738	-2.013
-0.623	-0.699	0.481	-0.586	-0.579	-0.129
0.191	0.071	-3.001	0.359	-0.094	1.501
0.031	0.402	0.884	0.457	-0.798	-0.768
0.023	1.066	0.736	-0.342	-0.188	1.395
-0.957	0.525	-1.865	-0.273	-0.035	0.371
-0.702	-0.432	-0.465	0.120	-0.238	-0.869
-1.016	0.417	0.056	0.561	-2.357	1.956
-0.281	0.932	-1.029	0.479	2.709	-0.057
-0.300	-0.594	-1.047	-1.347	0.996	-1.023
0.551	0.418	0.074	0.524	0.479	0.326
1.114	1.068	0.772	0.226	-0.298	1.064
0.162	-0.129	-1.204	1.097	-0.916	1.222
-1.153	1.298	-1.329	1.284	0.619	0.699
0.101	-1.381	-0.574	0.096	1.389	1.249
0.756	-0.860	-0.778	0.037	2.619	-0.420
1.048	1.000	0.170	0.389	-0.305	-0.321
1.900	-0.778	0.617	-1.430	0.267	0.978
-1.235	-0.258	0.243	-0.292	-0.405	0.397
-0.605	1.360	0.490	-0.027	-1.482	-1.256
-1.132	-0.780	-0.859	0.447	0.269	0.097

-0.686	0.957	0.976	0.274	-0.238	0.229
0.628	-0.269	0.202	0.201	1.365	0.210
-0.236	0.706	-0.397	0.412	-0.979	0.662
-0.440	-0.287	0.220	-0.658	-1.131	-0.435
0.838	-0.039	1.572	-0.288	-1.719	-0.953
-0.048	-0.520	-1.168	0.638	0.629	-1.455
0.359	-0.287	-0.470	0.182	-0.699	-0.457
-0.815	-0.567	2.039	-0.239	-0.213	1.479
0.735	-1.382	1.596	0.037	-1.789	0.226

A.5.0 EXAMPLE CALCULATION

All the four subroutines, explained in the preceding section were run on the VAX-11/780 computer system, available at National Institute of Hydrology.

The compilation and link timings required for each of the subroutine used are given in table 2.

Table 2 - Comparison Table for Different Subroutines for Arranging a Given Real Numbered Series in Order

		Subroutine name			
		ORDER1	ORDER2	ORDER3	ORDER4
Compilation time(sec)		2.13	2.21	2.20	2.04
Linking time (sec)		2.10	1.76	1.66	1.73
Memory required(bytes)		8435	8412	8545	8425
Run time for ten different runs	I	3.84	2.43	4.59	3.17
	II	3.55	2.52	4.97	3.24
	III	3.82	2.54	4.86	3.22
	IV	3.72	2.43	5.48	2.54
	V	3.81	2.28	4.86	3.29
	VI	3.88	2.59	5.09	3.33
	VII	3.89	2.35	5.34	3.38
	VIII	3.90	2.68	5.24	3.48
	IX	3.74	2.48	5.20	3.35
	X	3.73	2.50	4.88	3.20
Mean run time (sec)		3.788	2.480	5.051	3.320

Standard deviation of run time	0.1067	0.2253	0.2684	0.1215
Total time required for one run	8.018	6.450	8.911	7.090

Since the run timings of each of the subroutine by making use of only one set of data may not be representative, it was decided to find the average run time and standard deviation of the run timings for each of the subroutine by using ten different data sets. For this purpose a random number generation programme was used to generate 6000 real numbers and these generated numbers were divided into ten equal sets of 600 numbers in length to create ten different data files.

The random number generation programme is given in appendix V.

A.6.0 APPLICATION, SAMPLE INPUT, SAMPLE OUTPUT

Arranging the series in order is applicable in flood frequency analysis where the annual peak flood series or the partial duration series is to be arranged in order.

The following sample input is used for testing the various subroutines studied in this report:

0.464, 0.060, 1.486, 1.022, 1.394, 0.906,
1.179, -1.501, -0.690, 1.372, -0.482,
-1.376, -1.010, -0.005, 1.393

The following sample output, where the series has been arranged in order has been obtained using the subroutines studied in this report:

-1.501	0.464
-1.376	0.906
-1.010	1.022
-0.690	1.179
-0.482	1.372
-0.005	1.393
0.060	1.394
	1.486

A.7.0 RECOMMENDATIONS

All the four computer subroutines which have been studied in the present study are compared on the basis of compilation time, run time and memory requirement, in order to find out the most efficient subroutine.

Table 2 shows that the subroutine ORDER2 uses minimum average run time. The standard deviation is moderate. Although it requires more CPU time for the compilation, the run time required is minimum among all the other competing subroutines. Also it requires less total time requirement (6.45 secs) including compilation and linking, when compared with other subroutines. Subroutines ORDER3 requires maximum average run time and standard deviation. Also the total time requirement (8.91 secs) for one run including the compilation and link timings, is higher than any other subroutine. Therefore, the subroutine ORDER2 may be considered as the most efficient one among the four subroutines studied for arranging the series in order.

B. INTERPOLATION

B.1.0 INTRODUCTION

Interpolation has been defined as the art of reading between the lines of a table, and in elementary mathematics the term usually denotes the process of computing intermediate values of a function from a set of given or tabular values of that function. The general problem of interpolation, however is much larger than this. Frequently we have to deal with functions whose analytical form is either totally unknowr or else is of such a nature that the function can not be easily subjected to such operations, as may be required. In either case it is desirable to replace the given function by another which can be more readily handled. This operation of replacing or representing a given function by a simple one constitutes interpolation in the broad sense of term.

The general problem of interpolation consists, then, in representing a function known or unknown in a form chosen in advance with the aid of given values which thus function takes for definite values of the independent variable.

There are two kinds of interpolation depending on the type of data provided and the kind of results wanted. In the standard type of interpolation we are given a set of data points and want a curve that passes smoothly through them.

In least square interpolation, the data generally have some uncertainty associated with them and we want to find a smooth curve that passes sufficiently near the data points. In standard interpolation, the equation of the approximating curve must have as many parameters as there are data points. In least square fitting the number of parameters typically is much smaller than the number of data points. Nearly all the standard formulae of interpolation are polynomial formulae. In case the given function is known to be periodic, however, it is better to represent it by a trigonometric series.

The purpose of the subroutines studied here is to arrive at the value of a dependent variable for a given independent variable, given the relationship between dependent and independent variable, either in tabular form or in the form of a mathematical equation.

B.2.0 METHOD USED FOR VARIOUS SUBROUTINES

B.2.1 INTER2

This subroutine does the interpolation by spline fit. Suppose a table of values : x_k, y_k, $k = 1, 2, \ldots\ldots, n$ is given. The value of y corresponding to a specific value of x is determined by interpolation as follows:

a) The points x_k and x_{k+1}, adjacent to x such that $x_k \leq x \leq x_{k+1}$ are determined.

b) The value of y at x is then determined by the formula:

$$y = C_{1,k}(x_{k+1} - x)^3 + C_{2,k}(x - x_k)^3 + C_{3,k}(x_{k+1} - x) + C_{4,k}(x - x_k)$$

where,

$C_{1,k}$, $C_{2,k}$, $C_{3,k}$, and $C_{4,k}$ are spline constants.

B.2.2 INTER3

This subroutine is based on 2nd order parabolic method interpolation for equal interval pivotal points. Here three pivotal points, S_{i-1} ; S_i ; S_{i+1} are considered. The curve which passes through these three points is given by the equation:

$$Q(x) = A_0 + A_1(x - S_i) + A_2(x - S_i)^2$$

The coefficient A_0, A_1 and A_2 are being evaluated using the known values of Q at pivotal points S_{i-1} ; S_i ; and S_{i+1}. After substituting the values of these coefficients in the above equation, we get the value of Q(x) i.e. the interpolated value at desired point x.

B.2.3 INTER4

This subroutine uses the Lagrange's interpolation. In Lagrange interpolation we pass a polynomial of lowest possible degree through the n given data points. Since n parameters are needed the degree required is n-1 so that

$$f(x) = a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots\ldots\ldots + a_1 x + a_0 \qquad \ldots(1)$$

and from the problem statement we must have

$$f(x_i) = y_i \qquad \ldots(2)$$

where,

$$i = 1, 2, \ldots\ldots\ldots, n$$

plug equation (1) into equation (2), and get

$$a_{n-1} x_i^{n-1} + a_{n-2} x_i^{n-2} + \ldots\ldots\ldots + a_1 x_i + a_0 = y_i \qquad \ldots(3)$$

where,

$$i = 1, 2, \ldots\ldots\ldots, n$$

which represents a set of n linear algebraic equations in the n unknown a_0, a_1 ...., a_{n-1} , since the x_i and y_i are known. This set can be solved by standard linear equation solvers, but this is not a good way to proceed. Due to

Lagrange the following method is used:

Let the m^{th} degree polynomial be:

$$P_j(x) = A_j(x - x_0)\ (x - x_1)\ \ldots\ldots\ (x - x_{j-1})\ (x - x_{j+1})\ \ldots\ldots\ldots\ldots\ (x - x_m) \qquad \ldots(4)$$

which is zero at all x_i except x_j and equals 1 at x_j if the constant A_j is chosen equal to

$$A_j = \frac{1}{(x_j - x_0)\ (x_j - x_1)\ \ldots\ldots (x_j - x_{j-1})\ (x_j - x_{j+1})\ \ldots\ldots\ldots\ (x_j - x_m\ )}$$

Hence, the value of A_j,

$$P_j\ (x_i) = \begin{matrix} 0 \text{ for } i \neq j \\ 1 \text{ for } i = j \end{matrix} \qquad \ldots(5)$$

The linear combination of $P_j(x)$

$$P_m(x) = f_o\ P_o\ (x) + f_1\ P_1\ (x) + \ldots\ldots\ldots + f_m\ P_m\ (x)$$

$$= \sum_{i=0}^{m} f_i\ P_i\ (x) \qquad \ldots(6)$$

is an m^{th} degree polynomial and by equation(5) has, at $x = x_i$, the value

$$P_m(x_i) = f_o\ P_o(x_i) + f_1\ P_1(x_i) + \ldots\ldots\ldots + f_i\ P_i(x_i) + \ldots\ldots\ldots\ldots + f_m\ P_m\ (x_i)$$

$$= f_o . 0 + f_1 . 0 + \ldots\ldots + f_i . 1 + \ldots\ldots + f_m . 0$$

$$= f_i \qquad (i = 0, 1, \ldots\ldots, m)$$

Hence $P_m(x)$ is the m^{th} degree polynomial passing through the m+1 unevenly spaced pivotal points x_i $(i=0,1,\ldots\ldots,m)$.

When the pivotal points are evenly spaced by h, let

$$p = \frac{x - x_0}{h} \qquad \ldots(7)$$

equation (3) and (4) reduce to:

$$P_j(x) = A_j \, h^m \, p \, (p - 1) \, (p - 2) \ldots\ldots\ldots (p-j+1)$$

$$(p - j - 1) \ldots\ldots\ldots (p - m) \qquad \ldots(8)$$

$$A_j = \frac{1}{h^m \, j(j-1)(j-2) \ldots\ldots(1)(-1)\ldots\ldots(j-m)}$$

$$= \frac{(-1)^{m-j}}{h^m \, j! \, (m-j)!}$$

and equation (6) becomes the m^{th} degree interpolation formula:

$$P_m(x) = \sum_{i=0}^{m} \frac{(-1)^{m-i}}{i!(m-i)!} \, p(p-1) \, (p-2) \ldots\ldots (p-i+1)$$

$$(p-i-1) \ldots\ldots\ldots (p-m) \; f_i \qquad \ldots(9)$$

B.3.0 COMPUTER SUBROUTINES

All the three subroutines are written in FORTRAN IV computer programming language. After going through the standard literature available in hydrological computer programming and other standard text books on computer programming the following subroutines were selected for this study:

Sl No.	Source of the subroutine	Subroutine name	Reference No.
1.	HEC-1 package US Army Corps of Engineers, Hydrologic Engineering Centre	INTER2	5
2.	'FORTRAN HYDRO' - A Computer Programmes Documentation	INTER3	4
3.	Numerical Methods in Fortran	INTER4	6

The listing of the above subroutines are given in appendix VI, VII and VIII respectively.

B.4.0 INPUT SPECIFICATION, OUTPUT DESCRIPTION

No internal data storage is required for any of the three subroutines used in the present study.

All the three subroutines which have been explained in the preceding section were run on the VAX-11/780 system, available at National Institute of Hydrology.

Input data requirement is a set of equal/unequal interval array x and the corresponding array y containing values against each number in the array x. Also a set of numbers (points) at which the interpolated values are to be evaluated.

The numbers x_1, x_2 and so on in the input series x should be equally spaced for subroutine INTER3. For subroutines INTER2 and INTER4 the array x should be in ascending order.

Subroutine INTER3 can not extrapolate beyond the supplied tabular points limits. The programme identifies the location of the value, for which interpolation/extrapolation is needed, among the given tabular points of independent variates using the expression (S(I)-x), where S(I) is the supplied independent tabular points and x is the value to be interpolated/extrapolated. When this expression is positive then only the programme switches to the expression meant for interpolation. If this expression is negative even after comparing the given x value with all the given S(I) values then the programme returns to the main having the interpolated value either zero or the previous interpolated value depending on whether the interpolation is made for the first time or not.

This point is clarified by the following example:

A table with independent variable ranging between -3.0 and +3.0, has been supplied. If we supply the value to be extrapolated, say, +4.1, the expression (S(I)-x) will always be negative and the previous stored value in the location of y will be written as the extrapolated value. If we supply -4.0 as the value for which extrapolation is needed then in the first trial itself the expression (S(I)-x) becomes positive and the programme switches to the expression for interpolation taking S(I-1) and Q(I-1) as zero. Thus the interpolated value is wrong. Therefore, the subroutine can not extrapolate both in the higher and lower ranges.

B.5.0 TEST DATA

The first set of data (table 3) consists of a set of K values corresponding to the coefficient of skewness (C_s) which lies between -3.0 and +3.0 and for the exceedance probability = 0.001. The table is meant for interpolating the value of K, the frequency factor of the Pearson or log Pearson type III distribution, corresponding to the given probability of exceedance 0.001 and for the given C_s.

The second set of data (table 4) consists of x and y values of a known function $y = -2x^2+20x-2$ for all the values of x ranging between 0 and 10 with an increment of 0.1.

The points at which the interpolation is required are obtained randomly within the range.

Table 3 - Test Data Set-I for Interpolation

TEST DATA : GIVEN TABLE FOR INTERPOLATION - COEFFICIENT OF SKEWNESS VS. FREQUFNCY FACTOR FOR RETURN PERIOD = 1000 YR.
CS LIES BETWEEN -3.0 AND +3.0 WITH INCREMENT = 0.1

S.N.	COEFFICIENT OF SKEWNESS	FREQUENCY FACTOR	S.N. S.N.	COEFFICIENT OF SKEWNESS	FREQUENCY FACTOR
1	-3.0	0.67	31	0.0	3.09
2	-2.9	0.69	32	0.1	3.23
3	-2.8	0.71	33	0.2	3.38
4	-2.7	0.74	34	0.3	3.52
5	-2.6	0.77	35	0.4	3.67
6	-2.5	0.80	36	0.5	3.81
7	-2.4	0.83	37	0.6	3.96
8	-2.3	0.87	38	0.7	4.10
9	-2.2	0.91	39	0.8	4.24
10	-2.1	0.95	40	0.9	4.39
11	-2.0	1.00	41	1.0	4.53
12	-1.9	1.05	42	1.1	4.67
13	-1.8	1.11	43	1.2	4.81
14	-1.7	1.17	44	1.3	4.96
15	-1.6	1.24	45	1.4	5.10
16	-1.5	1.31	46	1.5	5.23
17	-1.4	1.39	47	1.6	5.37
18	-1.3	1.48	48	1.7	5.51
19	-1.2	1.58	49	1.8	5.64
20	-1.1	1.68	50	1.9	5.78
21	-1.0	1.79	51	2.0	5.91
22	-0.9	1.90	52	2.1	6.04
23	-0.8	2.02	53	2.2	6.17
24	-0.7	2.14	54	2.3	6.30
25	-0.6	2.27	55	2.4	6.42
26	-0.5	2.40	56	2.5	6.55
27	-0.4	2.53	57	2.6	6.67
28	-0.3	2.67	58	2.7	6.79
29	-0.2	2.81	59	2.8	6.92
30	-0.1	2.95	60	2.9	7.03
			61	3.0	7.15

Table 4 - Test Data Set-II for Interpolation

FUNCTION IS Y=-2X*X+20X-2

X	Y	X	Y	X	Y
0.0	-2.00	0.1	-0.02	0.2	1.92
0.3	3.82	0.4	5.68	0.5	7.50
0.6	9.28	0.7	11.02	0.8	12.72
0.9	14.38	1.0	16.00	1.1	17.58
1.2	19.12	1.3	20.62	1.4	22.08
1.5	23.50	1.6	24.88	1.7	26.22
1.8	27.52	1.9	28.78	2.0	30.00
2.1	31.18	2.2	32.32	2.3	33.42
2.4	34.48	2.5	35.50	2.6	36.48
2.7	37.42	2.8	38.32	2.9	39.18
3.0	40.00	3.1	40.78	3.2	41.52
3.3	42.22	3.4	42.88	3.5	43.50
3.6	44.08	3.7	44.62	3.8	45.12
3.9	45.58	4.0	46.00	4.1	46.38
4.2	46.72	4.3	47.02	4.4	47.28
4.5	47.50	4.6	47.68	4.7	47.82
4.8	47.92	4.9	47.98	5.0	48.00
5.1	47.98	5.2	47.92	5.3	47.82
5.4	47.68	5.5	47.50	5.6	47.28
5.7	47.02	5.8	46.72	5.9	46.38
6.0	46.00	6.1	45.58	6.2	45.12
6.3	44.62	6.4	44.08	6.5	43.50
6.6	42.88	6.7	42.22	6.8	41.52
6.9	40.78	7.0	40.00	7.1	39.18
7.2	38.32	7.3	37.42	7.4	36.48
7.5	35.50	7.6	34.48	7.7	33.42
7.8	32.32	7.9	31.18	8.0	30.00
8.1	28.78	8.2	27.52	8.3	26.22
8.4	24.88	8.5	23.50	8.6	22.08
8.7	20.62	8.8	19.52	8.9	17.58
9.0	16.00	9.1	14.38	9.2	12.72
9.3	11.02	9.4	9.28	9.5	7.50
9.6	5.68	9.7	3.82	9.8	1.92
9.9	-0.02				

B.6.0 EXAMPLE CALCULATION

Several computer runs are made to test the interpolation as well as the extrapolation capacity of the subroutines. The example data used are:

i) A table containing values of coefficient of skewness ($-3.0 \leq C_s \leq +3.0$) with increment of 0.1 vs. frequency factor corresponding to 1000 years return period. They are 61 in number (table 3). For interpolation 64 random number (table 5) are being supplied as x.

ii) A known parabolic function

$$y = -2x^2+20x-2$$

Table 5 gives the true values of the valuesof y for all the values of x where $0 \leq x \leq 10$ with increment of 0.1.

Extrapolation capability of the subroutines has been tested in the following manner. A set of tabular points x and y has been supplied and interpolation is done for x points which are well within the supplied range of table, but at the tail end of the table. Now extrapolation is done for the same value of x after its curtailment from the supplied tabular points x and y. In the subroutine INTER2 extrapolation is being made for some of the x values, for which the corresponding y values have been established as interpolated values using the previous table. The extrapolated values are highly deviating (table 6) from the established interpolated values. Also it is clear that the interpolated value of a point which is very nearer to any of the extreme ends of the supplied table is deviating from the interpolated value of the same point when it is located well within the same supplied table except for the extension of the tail ends by some additional x and y points.

For Lagrange's interpolation many runs are made to test the effect of the number of x and y values supplied for the purpose of interpolation for the known parabolic function.

Table 5 - Random numbers at Which Interpolated Values are to be Computed

0.464	0.060	1.486	1.022	1.394
0.906	1.179	-1.501	-0.690	1.372
-0.492	-1.376	-1.010	-0.005	1.393
-1.787	-0.105	-1.339	1.041	0.279
-1.805	-1.186	0.658	-0.439	-1.399
0.199	0.159	2.273	0.041	-1.132
0.768	0.375	-0.513	0.292	1.026
-1.334	-0.287	0.161	-1.346	1.250
0.630	0.375	-1.420	-0.151	-0.309
0.424	0.593	0.862	0.235	-0.853
0.137	-2.526	-0.354	-0.472	-0.535
-0.313	-1.055	-0.488	0.736	0.225
1.678	-0.150	0.598	-0.899	

Table 6 - Extrapolation Test Results

SN	X	EXTRAPOLATED VALUE (Y) USING INTER2	CORRECT VALUE OF Y	DIFFFRENCE
1	-2.95	0.232	0.680	0.45
2	-2.85	0.498	0.700	0.202
3	-2.75	0.659	0.724	0.065
4	-2.65	0.746	0.755	0.009
5	-2.55	0.787	0.785	-0.002
6	2.55	6.619	6.610	-0.009
7	2.65	6.764	6.730	-0.034
8	2.75	6.919	6.855	-0.064
9	2.85	7.084	6.975	-0.109
10	2.95	7.259	7.089	-0.170
11	-3.10	0.650	not available	
12	-4.00	0.470	--do--	
13	3.10	7.283	--do--	
14	4.00	10.733	--do--	

B.7.0 APPLICATION, SAMPLE INPUT, SAMPLE OUTPUT

Interpolation is an important tool for getting accurate estimates from tabluated values, say trigonometric, exponential or logarithmic functions. Also, most of the numerical methods for finding integrals and solving differential equations are based upon interpolation formulae.

The sample input is given in table 3, which contains coefficient of skewness varying between -3.0 and +3.0 vs. the frequency factor for return period 1000 years.

Another sample input is a known function $y = -2x^2+20x-2$ for $0 \leq x \leq 10$. The random numbers at which the interpolation is required are given in table 5.

The sample output is given in table 7 and table 6. Table 7 shows the interpolated values for the known function $y = -2x^2+20x-2$ and table 6 is checking the routine INTER2 for extrapolation.

Table 7 - Interpolation Results by Supplying Random Numbers

SN	Value of X (RandomNo)	Values of Y INTER2	Values of Y INTER3	SN	Value of X (RandomNo)	Values of Y INTER2	Values of Y INTER3
1	0.464	3.759	3.758	33	0.375	3.633	3.633
2	0.060	3.174	3.173	34	-0.513	2.383	2.383
3	1.486	5.211	5.211	35	0.292	3.508	3.508
4	1.022	4.561	4.561	36	1.026	4.566	4.566
5	1.794	5.092	5.092	37	-1.334	1.448	1.449
6	0.906	4.398	4.398	38	-0.287	2.688	2.688
7	1.179	4.781	4.780	39	0.161	3.321	3.323
8	-1.301	1.309	1.309	40	-1.346	1.437	1.437
9	-0.693	2.152	2.153	41	1.250	4.884	4.886
10	1.372	5.062	5.062	42	0.630	4.002	4.002
11	-0.482	2.423	2.423	43	0.375	3.633	3.633
12	-1.376	1.410	1.411	44	-1.420	1.373	1.372
13	-1.010	1.779	1.779	45	-0.151	2.879	2.879
14	-0.005	3.083	3.083	46	-0.309	2.657	2.657
15	1.393	5.091	5.091	47	0.424	3.704	3.703
16	-1.787	1.118	1.117	48	0.593	3.950	3.950
17	-0.105	2.943	2.943	49	0.862	4.334	4.334
18	-1.339	1.443	1.444	50	0.235	3.429	3.428
19	1.041	4.587	4.587	51	-0.853	1.955	1.956
20	0.279	3.490	3.490	52	0.137	3.284	3.287
21	-1.805	1.107	1.107	53	-2.016	0.792	0.792
22	-1.186	1.594	1.593	54	-0.354	2.594	2.594
23	0.638	4.041	4.041	55	-0.472	2.436	2.435
24	-0.439	2.479	2.478	56	-0.555	2.329	2.329
25	-1.399	1.391	1.391	57	-0.513	2.383	2.383
26	0.199	3.379	3.379	58	-1.055	1.730	1.730
27	0.159	3.318	3.320	59	-0.488	2.416	2.415
28	2.273	6.263	6.266	60	0.756	4.178	4.177
29	0.041	3.147	3.146	61	0.225	3.415	3.414
30	-1.132	1.647	1.647	62	1.678	5.479	5.480
31	0.768	4.195	4.194	63	-0.150	2.880	2.880
32	-0.899	1.901	1.901	64	0.598	3.957	3.957

B.8.0 RECOMMENDATIONS

Subroutine INTER2 which involves interpolation using spline fit gives the interpolated values exactly coinciding with the true values for the known function. The performance of the subroutine is equally good for the given equal interval as well as unequal interval table. So this subroutine can be considered as 'efficient' for interpolation. As it is clear from table 3, subroutine INTER2 is inappropriate for the purpose of extrapolation. In this example table which contains the x values lying between -2.5 and +2.5, extrapolated values nearer to the extremes are having lesser error (0.2% and 0.9%). The percentage of error increases as we go further away from the given extreme ends. Subroutine INTER2 gives the correct results, hence standard error is zero in this case.

Subroutine INTER3, which uses the 2nd order polynomial interpolation method is limited to evenly spaced pivotal points only. Though it is giving correct interpolated values, it is not able to do the extrapolation.

Subroutine INTER4 which does the interpolation using the Lagrange formula is very sensitive to the number of given pivotal points (may or may not be) evenly spaced. This sensitivity is illustrated in figures 1 to 6. Figure 7 shows the overall effect of number of points on the error in interpolated values. It is clear from these illustrations that the interpolated curves tend to oscillate about the exact results. Smooth functions are treated more accurately than oscillatory

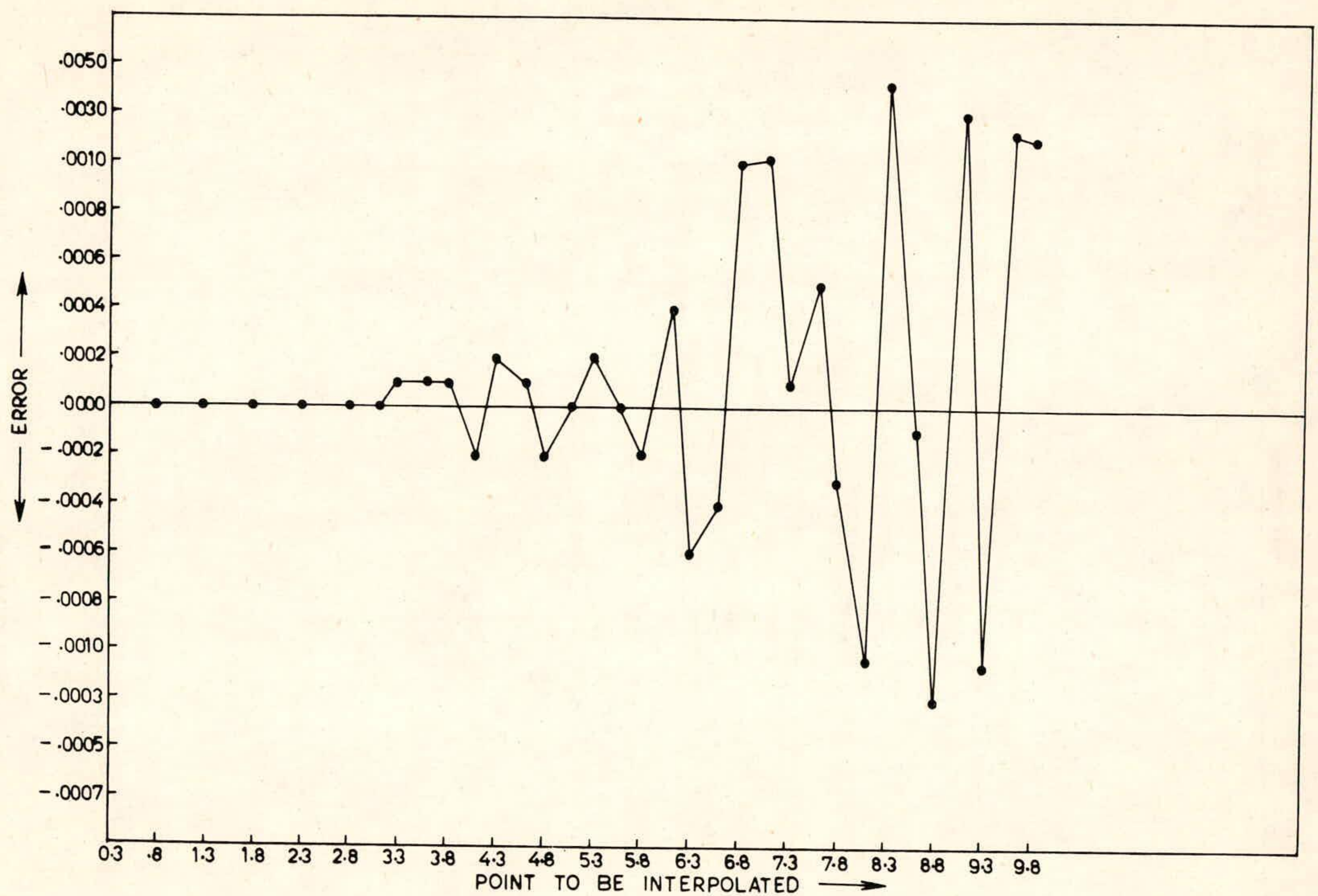

FIGURE 1 - ERROR IN LAGRANGE INTERPOLATION OF $-2x^2+20x-2$ USING 4 INITIAL POINTS(0.0,0.5,1.0,1.5)

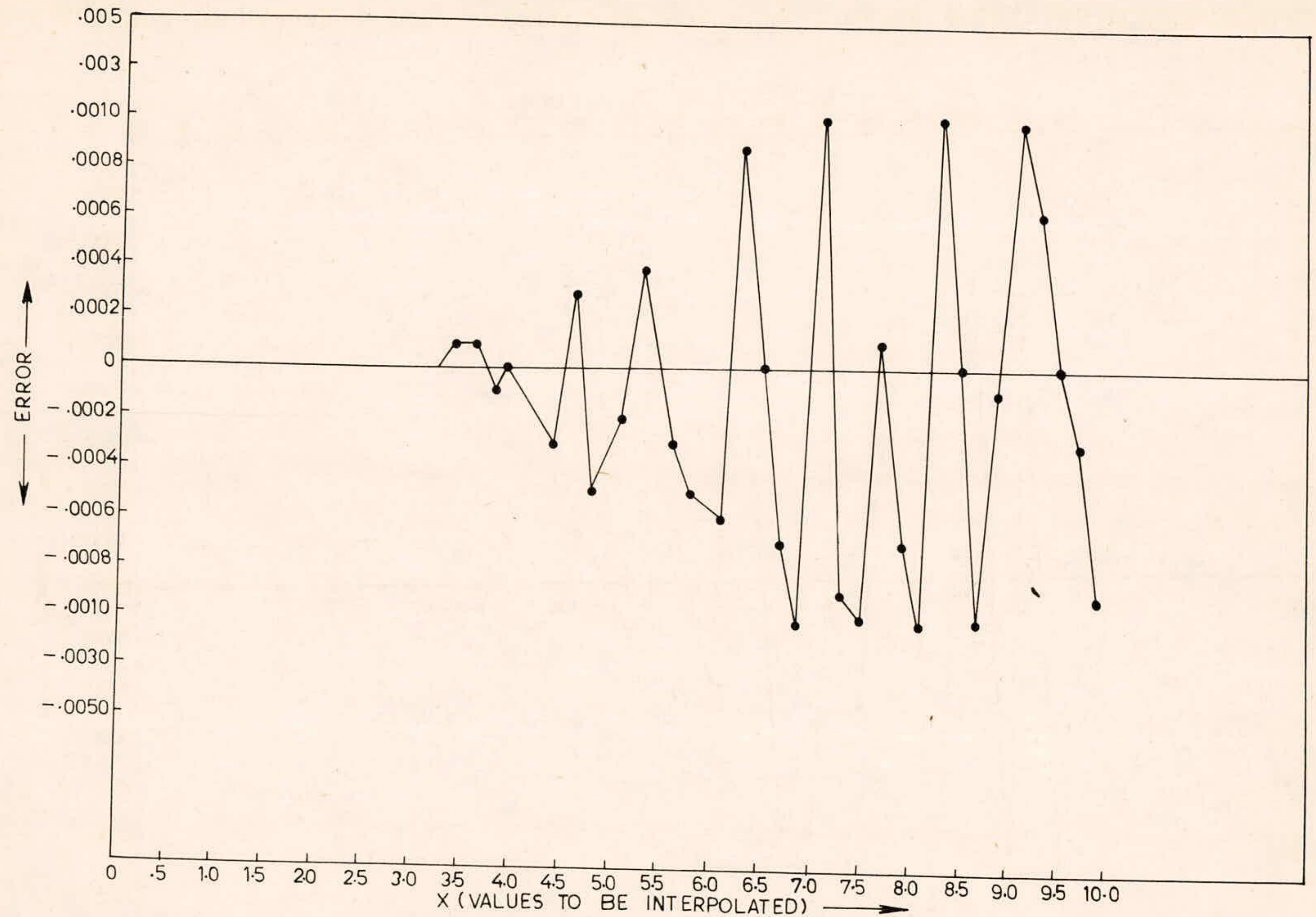

FIGURE 2 - ERROR IN LAGRANGE INTERPOLATION FOR $-2x^2+20x-2$ USING 4 POINTS (1.0, 1.5, 2.0, 2.5)

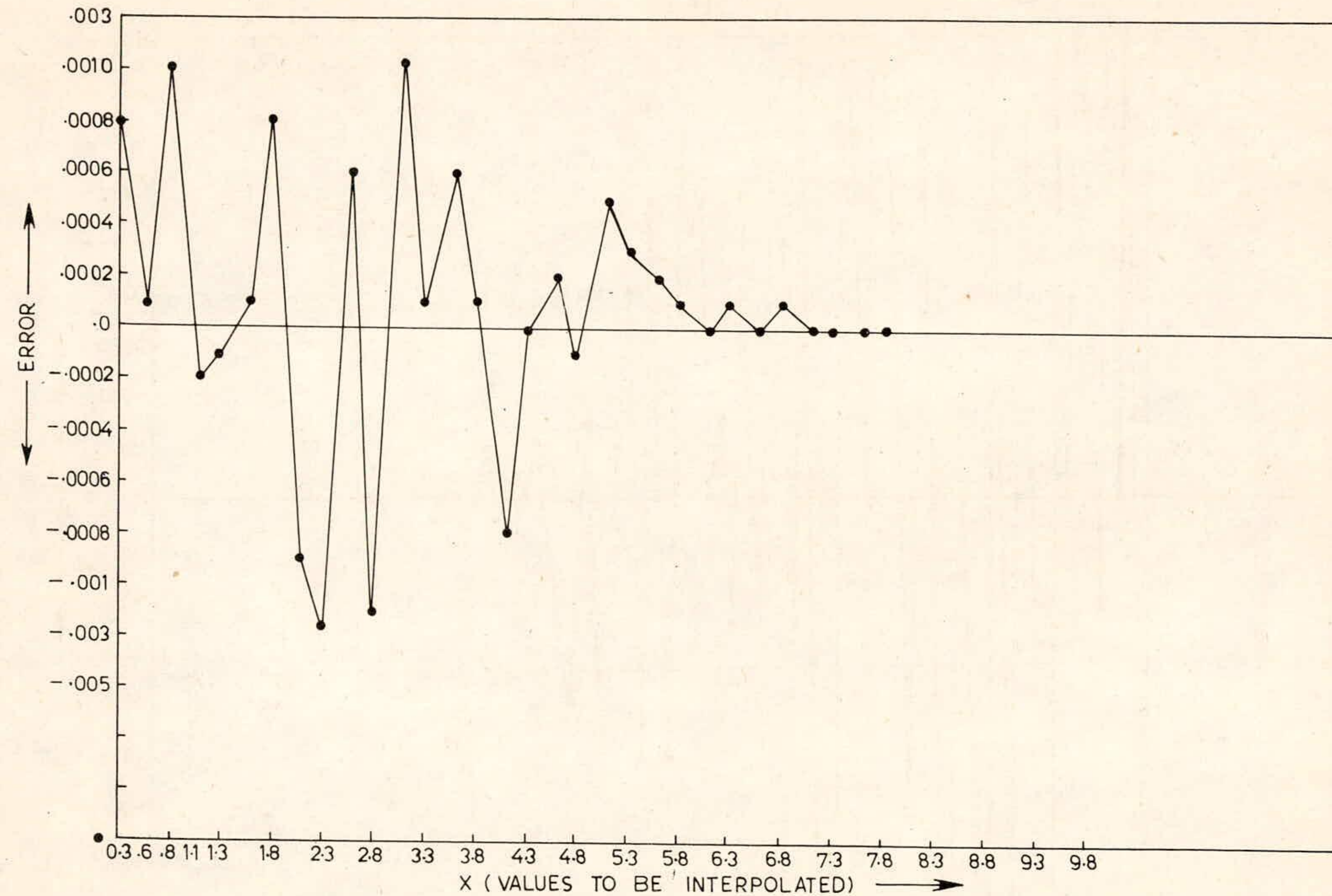

FIGURE 3 - ERROR IN LAGRANGE INTERPOLATION USING 4 END POINTS (8.5, 9.0, 9.5, 10.0)

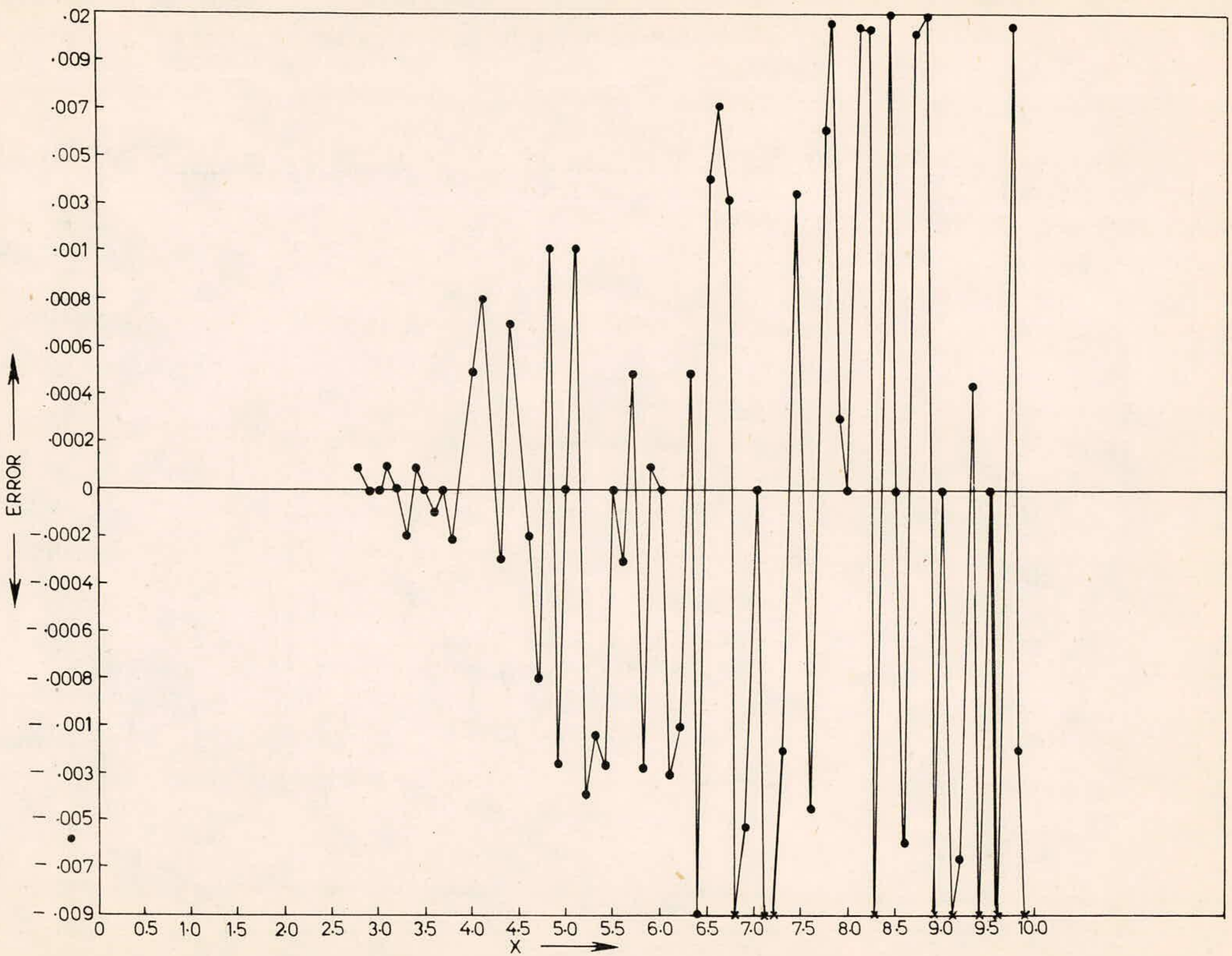

FIGURE 4 - ERROR IN LAGRANGE INTERPOLATION USING 5 INITIAL POINTS (0.0, 0.5, 1.0, 1.5, 2.0)

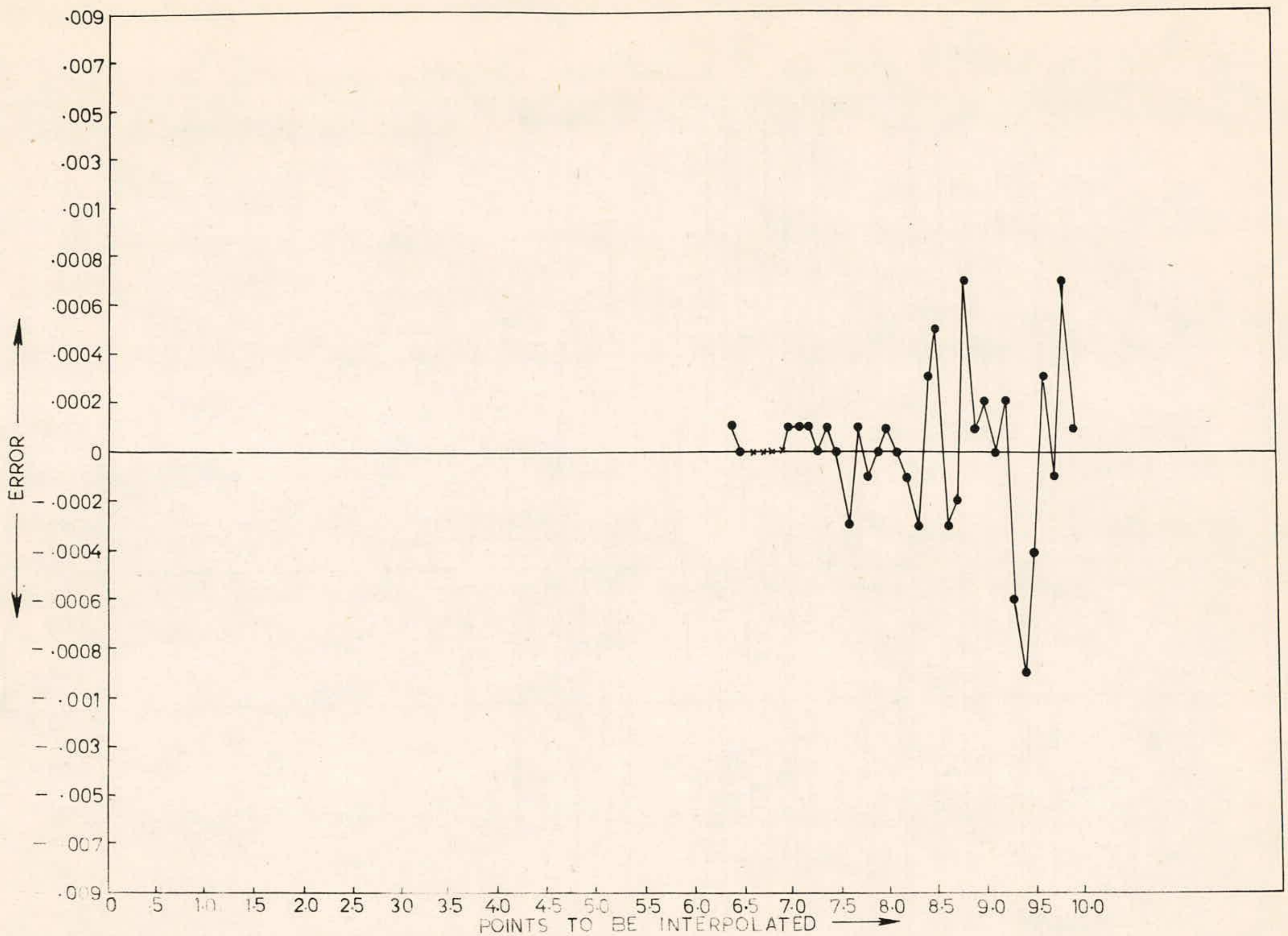

FIGURE 5 - ERROR IN LAGRANGE INTERPOLATION USING 5 POINTS (0.0, 0.5, 1.5, 3.0, 5.0)

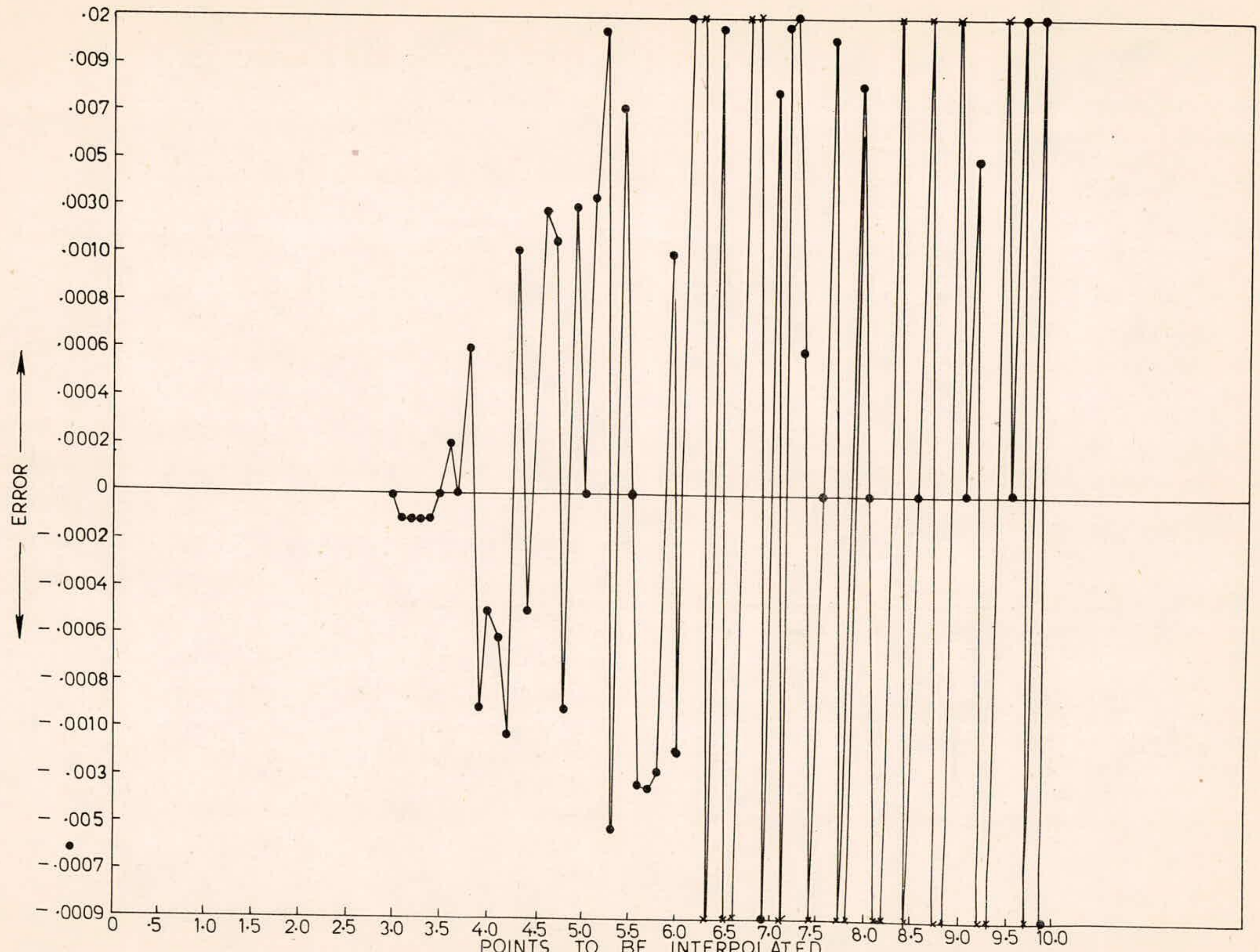

FIGURE 6 - ERROR IN LAGRANGE INTERPOLATION USING 6 INITIAL POINTS (0.0, 0.5, 1.0, 1.5, 2.0, 2.5)

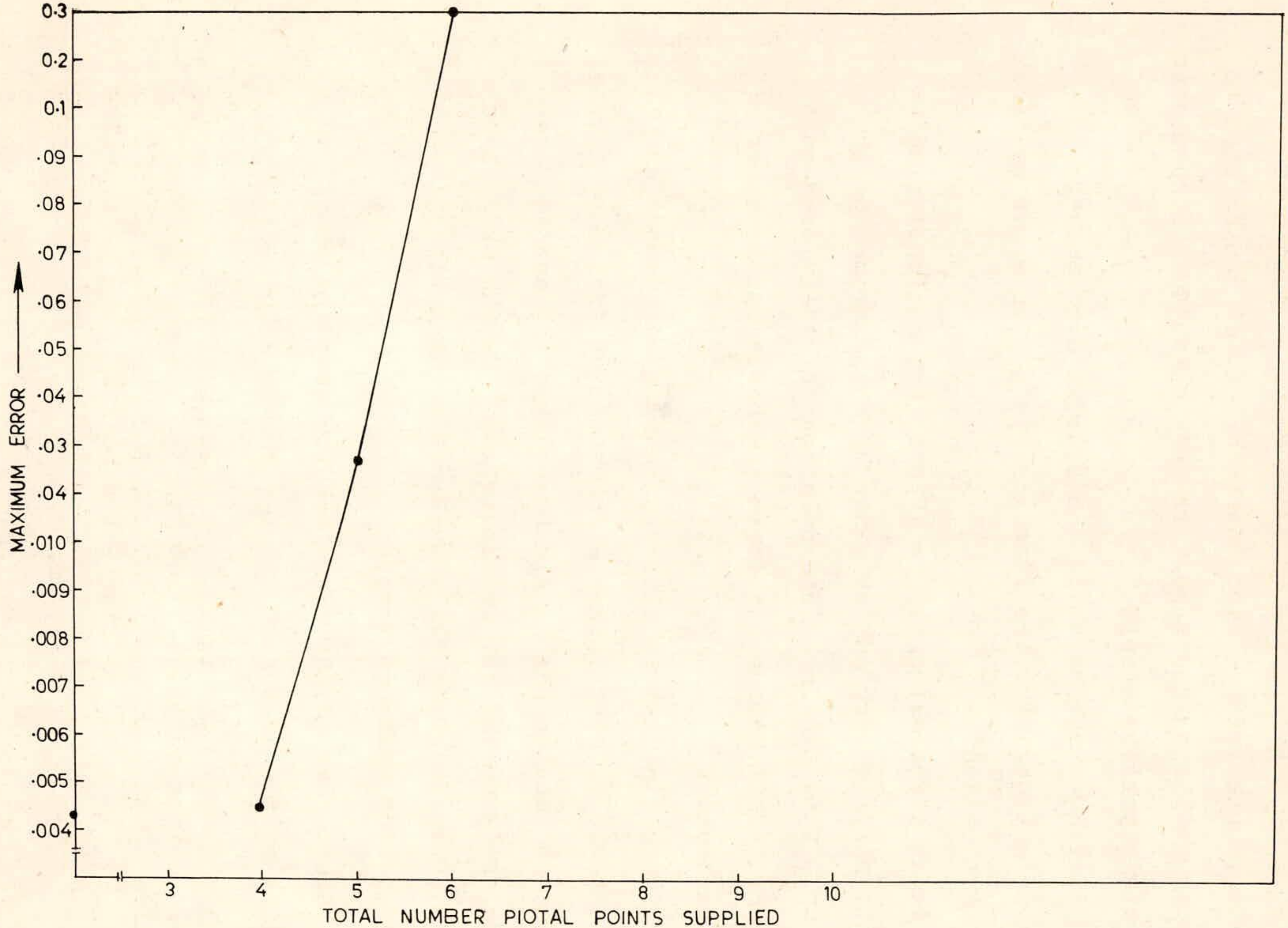

FIGURE 7 - MAXIMUM ERROR IN LAGRANGE INTERPOLATION FOR $-2x^2+20x-2$ VERSUS NUMBER OF POINTS USED

ones or ones with concentrated curvature. For these reasons, Lagrange interpolation for more than 3 or 4 points is rarely used. Higher accuracy can be obtained by having the computer used double precision. Also double precision can be used to determine whether a bad result is due to roundoff accumulation or to the algorithm itself.

Lagrange interpolation works as an extrapolator to a certain extent. For the function $y = x^3$, at x =10 we find

$f(x) = 1000$

which is an exact value ($10^3 = 1000$). But extrapolation is possible only to very nearer range of the extremes of the function. Thus Lagrange interpolation is rarely used as a method by itself because it forms the basis for a number of other methods.

Subroutine INTER3 gives standard error 0.0007243 for the table of coefficient of skewness vs frequency factor, in which 60 tabular points are being supplied and 64 random numbers are supplied for interpolation. On the other hand for the known function $-2x^2+20x-2$ this subroutine gives standard error equal to zero.

Subroutine INTER4 gives varying standard error depending on the number of tabular points supplied:

Sl.No.	Points supplied	Location	Standard error
1.	4	Initial 4 points	0.001087
2.	4	Upper middle	0.006679
3.	4	End points	0.000928
4.	5	Initial	0.00777

5.	5	Upper middle	0.00018
6.	6	Initial	0.092416
7.	6	Middle (covering the whole range)	0.000023
8.	10	All the points with equal interval	0.015157
9.	7	Covering full range	0.00

It has also been observed in Lagrange's interpolation of the function $-2x^2+20x-2$ that if the points used are covering the full range of the given function (if known), in the present example $0 \leq x \leq 10$, the interpolated values are more close to the true values, irrespective to the number of points used. Following are the set of points used. Error is found only when the interval between two points is a bit more.

Sl.No.	I Set	II Set	III Set	IV Set (The only set of points that gives the correct values)
1.	1.5	1.5	0.0	0.0
2.	4.0	4.0	0.5	1.5
3.	7.0	4.5	1.5	3.5
4.	8.5	7.0	3.0	4.0
5.	9.5	8.5	5.0	6.5

6.	10.0	9.5	6.0	8.0
7.	-	10.0	8.5	10.5

Finally subroutine INTER2 which uses the spline function can be considered to be the best one.

REFERENCES

1. Hydrologic Engineering Centre, U.S.Army Corps of Engineers (1979) ,'Generalized Computer Program on Flood Frequency Analysis'.

2. Kite, G.W. (1977),'Frequency and Risk Analysis in Hydrology', Water Resources Pdblications, Fort Collins, Colo.,p.165-166.

3. Rajaraman, V.,(1976),'Computer Programming in Fortran IV', Prentice Hall of India Private Limited, New Delhi p.102-103.

4. Ramaseshan, S., (Editor),(1978),'Intensive Course on Computer Application in Hydrology', Lecture Notes Volume III. IIT Kanpur.

5. Hydrologic Engineering Centre, U.S.Army Corps of Engineers (1981),'Generalized Computer Program HEC-1, Flood Hydrograph package'.

6. Mc Cornick John, M. and Mario G. Salvadori, (1979) 'Numerical Methods in Fortran', Prentice Hall of India, New Delhi, p.324.

APPENDIX I

SUBROUTINE ORDER1

```
C       SUBROUTINE ORDER1
C       THIS SUBROUTINE ARRANGES A GIVEN SERIES IN ORDER
C       REFFERENCE : KITE,G.W.,FREQUENCY AND RISK ANALYSIS
C                    IN HYDROLOGY , WATER RESOURCES PUBLIC-
C                    ATIONS
C       N - SAMPLE SIZE
C       X(J) - JTH NUMBER OF THE ARRAY
C       XT - TEMPORARY STORAGE VARIABLE
        SUBROUTINE ORDER1(N,X)
        DIMENSION X(1)
        K=N-1
        DO 2 L=1,K
        M=N-L
        DO 2 J=1,M
C       FOR DESCENDING ORDER
        IF (X(J)-X(J+1))1,1,2
C       FOR ASCENDING ORDER INSERT
C       IF (X(J+1)-X(J))1,1,2
1       XT=X(J)
        X(J)=X(J+1)
        X(J+1)=XT
2       CONTINUE
        RETURN
        END
```

APPENDIX II

SUBROUTINE ORDER2

```
C       SUBROUTINE ORDER2
C       THIS SUBROUTINE ARRANGES A GIVEN SERIES IN ORDER
C       REFFERENCE : RAMASESHAN,S.,(EDITOR),1972,'INTENSIVE
C                   COURSE ON COMPUTER APPLICATION IN HYDR-
C                   OLOGY LECTURE NOTES , VOL.3
C       FOR ASCENDING ORDER SUPPLY KO=1
C       FOR DESCENDING ORDER SUPPLY KO=0
C       X - X ARRAY TO BE ARRANGED IN ORDER
C       N - LENGTH OF THE ARRAY
        SUBROUTINE ORDER2(X,N,KO)
        DIMENSION X(N)
        M=N
1       M=M/2
        IF(M.EQ.0) GO TO 5
        K=N-M
        J=1
2       I=J
3       IM=I+M
        IF(X(I).GE.X(IM)) GO TO 4
        TEMP=X(I)
        X(I)=X(IM)
        X(IM)=TEMP
        I=I-M
        IF(I.GE.1) GO TO 3
4       J=J+1
        IF(J.GT.K) GO TO 1
        GO TO 2
5       IF(KO.EQ.0) GO TO 7
        M=N/2
        DO 6 I=1,M
        TEMP=X(I)
        J=N-I+1
        X(I)=X(J)
        X(J)=TEMP
6       CONTINUE
7       RETURN
        END
```

APPENDIX III

SUBROUTINE ORDER3

```
C       SUBROUTINE ORDER3
C       THIS SUBROUTINE ARRANGES A GIVEN SERIES IN ORDER
C       REFFERENCE : RAJARAMAN,V.,COMPUTER PROGRAMING IN
C                    FORTRAN VI,PRENTICE HALL OF INDIA
C                    PRIVATE LIMITED
C       FOR ASCENDING ORDER KO=0
C       FOR DESCENDING ORDER KO=1
C       X - ARRAY WHICH IS TO BE ARRANGED IN ORDER
C       N - LENGTH OF THE SERIES
C       TEMP - TEMPORARY STORAGE VARIABLE
        SUBROUTINE ORDER3 (N,X,KO)
        DIMENSION X(1000)
        LIMIT=N-1
10      LAST=1
        DO 30 I=1,LIMIT
        IF(KO.EQ.0)GO TO 2
        IF(X(I+1).GT.X(I))GO TO 20
2       IF(X(I+1).LT.X(I))GO TO 20
        GO TO 30
20      TEMP=X(I)
        X(I)=X(I+1)
        X(I+1)=TEMP
        LAST=I
30      CONTINUE
        IF(LAST.NE.1)GO TO 10
        RETURN
        END
```

APPENDIX IV

SUBROUTINE ORDER4

```
C       SUBROUTINE ORDER4
C       THIS SUBROUTINE ARRANGES A GIVEN SERIES IN ORDER
C       REFFERENCE : HYDROLOGIC ENGINEERING CENTRE,US ARMY
C                   CORPS OF ENGINEERS,GENERALIZED COMPUTER
C                   PROGRAM ON FLOOD FREQUENCY ANALYSIS,1979
C       X - GIVEN ARRAY TO BE ARRANGED IN ORDER
C       N - LENGTH OF THE SERIES
C       TEMP - TEMPORARY STORAGE VARIABLE
        SUBROUTINE ORDER4 (X,N,KO)
        DIMENSION X(1000)
        NA=N-1
        DO 20 J=1,NA
        M=J
        MA=J+1
        DO 10 L=MA,N
C       FOR ASCENDING ORDER REPLACE THE
C       FOLLOWING STATEMENT LIKE THIS --
C       IF(X(L).LT.X(M)) M=L
        IF (X(L).GT.X(M)) M=L
10      CONTINUE
        TEMP=X(J)
        X(J)=X(M)
        X(M)=TEMP
20      CONTINUE
        RETURN
        END
```

APPENDIX V

SUBROUTINE RANDOM

```
C     SUBROUTINE RANDOM
C     THIS PROGRAM IS MEANT FOR GENERATING RANDOM NUMBERS
C     S - STANDARD DEVIATION = 1.0
C     AM - MEAN = 0.0
      SUBROUTINE RANDOM(E,N)
      DIMENSION E(N)
      IX=1073741823
      DO 20 I=1,N
      CALL GAUSS (IX,1.0,0.,V)
20    E(I)=V
      RETURN
      END
      SUBROUTINE GAUSS (IX,S,AM,V)
      A=0.0
      DO 50 I=1,12
      Y=RAN(IX)
50    A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
```

APPENDIX VI

SUBROUTINE INTER2

```
C       PROGRAM FOR SPLINE INTERPOLATION
C       REFFERENCE : HEC-1 PACKAGE(SUBROUTINE SPLINE AND AKIMA)
C       NPTS-NO. OF POINTS
C
C         DIMENSIONS SET FOR DAMAGE CLACULATION IN HEC-1
C
C         ARRAY NAME   DIMENSION
C         ----------   ---------
C         EM, T        KPTS=(10*(KRTIO-1)+1)
C
C......    COMMON FOR INPUT AND OUTPUT UNITS
C
C
        DIMENSION X(100),Y(100),XINP1(100)
        DIMENSION EM(81),T(81)
        OPEN(UNIT=1,FILE='INTER.DAT',STATUS='OLD')
        OPEN(UNIT=2,FILE='INTER2.OUT',STATUS='NEW')
        DATA KPTS/81/
C
C     *** COMPUTES SPLINE COEFFICIENTS BY AKIMA METHOD -- A NEW METHOD
C     OF INTERPOLATION AND SMOOTH CURVE FITTING BASED ON LOCAL PROCEDURE
C     H. AKIMA, J.A.C.M., 17, 589-602, 1970.
C     PROGRAM BY H. KUBIK, HYDROLOGIC ENGINEERING CENTER -- AUGUST 1976
C          N   - NUMBER OF POINTS IN ARRAY.
C          X   - X ARRAY, VALUES MUST BE UNIQUE AND INCREASE.
C          Y   - Y ARRAY.
C          T   - COEFFICIENT ARRAY
C
C     * * * *  IF ONLY TWO POINTS, USE LINEAR INTERPOLATION
        READ(1,*)NPTS
        WRITE(2,5) NPTS,N1
5       FORMAT(5X,'THE NUMBER OF GIVEN TABULAR POINTS ARE=',I2)
        READ(1,*) (X(I),I=1,NPTS),(Y(I),I=1,NPTS)
        WRITE(2,6)
6       FORMAT(5X,'THE FOLLOWING ARE THE GIVEN TABULAR POINTS')
        WRITE(2,1) (X(I),Y(I),I=1,NPTS)
        READ (1,*)(XINP1(I),I=1,N1)
2       FORMAT(5X,'NUMBER OF POINTS TO BE INTERPOLATED ARE=',I2/)
        WRITE (2,2)N1
        WRITE (2,7)
7       FORMAT(5X,'THE FOLLOWING ARE THE POINTS AT WHICH INTERPO
```

```
     1LATIONS ARE REQUIRED')
      WRITE(2,1)(XINP1(I),I=1,N1)
1     FORMAT(5X,10F10.3)
      N=NPTS
      IF(N.LE.2) GO TO 555
      IF (N.GT.KPTS) GO TO 1080
      DO 1000 I=2,N
      J=I
      TMP=X(I)-X(I-1)
      IF(TMP.LE.0.) GO TO 1100
 1000 EM(I-1)=(Y(I)-Y(I-1))/TMP
      DO 1070 I=1,N
      IF(I.EQ.1) GO TO 1010
      IF(I.EQ.2) GO TO 1020
      IF(I.EQ.(N-1)) GO TO 1030
      IF(I.EQ.N) GO TO 1040
      TEMP1=ABS(EM(I+1)-EM(I))
      TEMP2=ABS(EM(I-1)-EM(I-2))
      TEMP3=EM(I-1)
      TEMP4=EM(I)
      GO TO 1050
 1010 TEMP1=ABS(EM(2)-EM(1))
      TEMP2=TEMP1
      TEMP3=2.*EM(1)-EM(2)
      TEMP4=EM(1)
      GO TO 1050
 1020 TEMP1=ABS(EM(3)-EM(2))
      TEMP2=ABS(EM(2)-EM(1))
      TEMP3=EM(1)
      TEMP4=EM(2)
      GO TO 1050
1030  TEMP1=ABS(EM(N-1)-EM(N-2))
      TEMP2=ABS(EM(N-2)-EM(N-3))
      TEMP3=EM(N-2)
      TEMP4=EM(N-1)
      GO TO 1050
1040  TEMP1=ABS(EM(N-1)-EM(N-2))
      TEMP2=TEMP1
      TEMP3=EM(N-1)
      TEMP4=2.*EM(N-1)-EM(N-2)
1050  IF(TEMP1.LE.0..AND.TEMP2.LE.0.) GO TO 1060
      T(I)=(TEMP3*TEMP1+TEMP4*TEMP2)/(TEMP1+TEMP2)
      GO TO 1070
1060  T(I)=(TEMP4+TEMP3)/2.
1070  CONTINUE
      DO 21 I=1,N1
      XINP=XINP1(I)
      CALL AKIMA(NPTS,X,Y,XINP,T)
21    CONTINUE
555   CALL AKIMA(NPTS,X,Y,XINP,T)
```

```
      STOP
1080    WRITE(2,1090) NPTS,N
 1090 FORMAT (/44H *** ERROR *** ARRAY SIZE EXCEEDED IN AKIMA.,
     1         13H  DIMENSION =,I4,12H, BUT NPTS =,I4)
      STOP 1301
 1100 WRITE (2,1110) J,X(J-1),X(J)
 1110 FORMAT (/47H *** HEC-1 ERROR 30 *** X VALUES ARE NOT UNIQUE,
     1         49H AND/OR INCREASING FOR CUBIC SPLINE INTERPOLATION/
     2         22X,3H J=,I3,5X,8H X(J-1)=,F10.2,5X,6H X(J)=,F10.2)
      STOP 1302
      END
      SUBROUTINE AKIMA (NPTS,X,Y,XINP,T)
      DIMENSION X(NPTS), Y(NPTS) ,T(81)
C
C
C     *** INTEROLATION BY AKIMA METHOD. SEE SUBROUTINE AKIMA FOR REF.
C     PROGRAM BY H. KUBIK, HYDROLOGIC ENGINEERING CENTER -- AUGUST 1976
C
      N=NPTS
      XIN=XINP
      IF(N.LT.2) GO TO 1060
 1020 IF (N.GT.2) GO TO 1030
C             USE LINEAR INTERPOLATION IF ONLY TWO POINTS
      FUNCT=Y(1)+(XIN-X(1))/(X(2)-X(1))*(Y(2)-Y(1))
       GO TO 666
 1030 DO 1040 II=2,N
      I=II-1
      IF(XIN.LT. X(II)) GO TO 1050
 1040 CONTINUE
 1050 TP1=XIN-X(I)
      TP2=Y(I+1)-Y(I)
      TP3=X(I+1)-X(I)
        FUNCT=Y(I)+T(I)*TP1+((3.*TP2)/TP3-2.*T(I)-T(I+1))/TP3*TP1**2
        1+(T(I)+T(I+1)-2.*TP2/TP3)*TP1**3/TP3**2
666     WRITE(2,50) XINP,FUNCT
50      FORMAT(/10X,'Y('F6.3,')='F10.3/)
      RETURN
 1060 WRITE (IP,1070)
 1070 FORMAT (/47H *** HEC-1 ERROR 29 *** ONLY ONE DATA POINT FOR,
     1        14H INTERPOLATION)
       RETURN
       END
```

APPENDIX VII

SUBROUTINE INTER3

```
C       SUBROUTINE INTER3
C       INTERPOLATION PROGRAM FOR EQUAL INTERVAL TABLE
C       USING 2ND ORDER POLYNOMIAL METHOD
C       REFFERENCE : FORTRAN HYDRO-A COMPUTER PROGRAM
C                    DOCUMENTATION
C       S - EQUAL INTERVAL ARRAY OF KNOWN VARIABLE
C       Q - GIVEN VALUES CORRESPONDING TO ARRAY S
C       X - POINTS AT WHICH INTERPOLATION IS TO BE DONE
C       Y - INTERPOLATED VALUES
C       N - NUMBER OF PIVOTAL POINTS IN THE GIVEN ARRAY S
        COMMON S(100),Q(100)
        DIMENSION X1(100),Y1(100)
        OPEN (UNIT=21,FILE='INTER.DAT',STATUS='OLD')
        READ (21,*)N,N1
        READ (21,*) (S(I),I=1,N),(Q(I),I=1,N)
        READ (21,*) (X1(J),J=1,N1)
        TYPE 101,N
        TYPE 1001,(S(I),Q(I),I=1,N)
101     FORMAT(5X,I2)
1001    FORMAT(10X,10F10.3)
        DO 20 J=1,N1
        X=X1(J)
        CALL INTER(S,Q,X,Y,N)
        Y1(J)=Y
20      CONTINUE
        TYPE 102,N1
        TYPE 1002,(X1(J),Y1(J),J=1,N1)
102     FORMAT(5X,I2)
1002    FORMAT(10X,10F10.3)
        STOP
        END
C       SUBROUTINE INTER
        SUBROUTINE INTER(S,Q,X,Y,N)
        DIMENSION S(100),Q(100)
        J=2
        IF(X.LE.S(2))GO TO 2
4       IF(S(J+1).LT.X) GO TO 3
        GO TO 2
3       J=J+1
        IF (J.EQ.(N-1)) GO TO 2
        GO TO 4
2       H=S(J+1)-S(J)
        PQ=(X-S(J))/H
        Y=Q(J)+(PQ/2.)*(Q(J+1)-Q(J-1))+(PQ*PQ/2.)
        1*(Q(J+1)-2.*Q(J)+Q(J-1))
        RETURN
        END
```

APPENDIX VIII

SUBROUTINE INTER4

```
C       SUBROUTINE INTER4
C       PROGRAM FOR LAGRANGIAN INTERPOLATION
C       UNEVENLY SPACED PIVOTAL POINTS
C       REFFERENCE : NUMERICAL METHODS IN FORTRAN
C                    JOHN C.Mc.CORMICK,SALVADORI
C       X - X ARRAY HAVING UNEVEN PIVOTAL POINTS
C       F - F ARRAY , VALUES CORRESPONDING TO X ARRAY
C       P - INTERPOLATED VALUES
C       XO1 - POINTS TO BE INTERPOLATED
C       N - NUMBER OF PIVOTAL POINTS IN X ARRAY
C       M - NUMBER OF POINTS IN XO ARRAY
        DIMENSION X(100),P(100),F(100),XO1(100)
        OPEN (UNIT=21,FILE='INTER.DAT',STATUS='OLD')
        READ (21,*)N,M,(X(I),I=1,N),(F(I),I=1,N)
        READ (21,*)(XO1(I),I=1,M)
        TYPE 50
        TYPE 200,(I,X(I),F(I),I=1,N)
        DO 5K=1,M
        XO=XO1(K)
        DO 10 J=1,N
        P(J)=1
        DO 10 I=1,N
        IF (I-J) 9,10,9
9       P(J)=P(J)*(XO-X(I))/(X(J)-X(I))
10      CONTINUE
        FO=0.
        DO 20 I=1,N
20      FO=FO+P(I)*F(I)
C       TYPE 50
C       TYPE 200,(I,X(I),F(I),P(I),I=1,N)
        TYPE 300,XO,FO
5       CONTINUE
200     FORMAT(I10,2F14.7)
50      FORMAT(//9X,'I',8X,'X(I)',10X,'F(I)'/)
300     FORMAT(/9X,'F(',F8.4,') = ',F12.4)
30      STOP
        END
```