ISSN 0252-1075
Research Report No. RR-046

Contributions from

Indian Institute of Tropical Meteorology

A FORTRAN-77 ALGORITHM FOR CUBIC SPLINE INTERPOLATION FOR REGULAR AND IRREGULAR GRIDS

By

M. K. Tandon

PUNE - 411 008
INDIA

NOVEMBER 1991

A FORTRAN-77 Algorithm For Cubic Spline Interpolation For Regular and Irregular Grids

M.K.Tandon

Indian Institute of Tropical Meteorology, Pune-411008, India

Abstract

The algorithm described here gives a procedure for cubic spline interpolation technique. This algorithm was successfully used on a number of different data sets. Results of two of these cases are presented in the text. Algorithm has provision to interpolate the data on regular and irregular grids. The ANSI FORTRAN-77 code of this algorithm was written, developed and tested on ND-560/Cx supermini computer system.

Key words : Interpolation, Cubic, Spline, Knots, FORTRAN, Polynomial, Slope, Curvature, Supermini, Grid, Derivative.

Introduction

Spline is a flexible string used in drawing curves. Spline functions constitute a relatively new subject in analysis. The theory of splines and experience with their use in numerical analysis have undergone a considerable degree of development in the recent past. The natural starting point for a study of spline functions is cubic spline. Its simplicity motivates much of its applications to the problems in numerical analysis. The spline proves to be an effective tool in the process of interpolation. The spline approximation was first reported by Schoenberg (1946). This technique is one of the most widely used techniques for interpolation. The algorithm discussed here has options of giving interpolated output on both regular and irregular grids depending upon the computational requirements of the problem. The ANSI FORTRAN-77 code of this algorithm is listed in Appendix-A.

Methodology

The methodology adopted in this algorithm is similar to Carnahan and Wilkes (1973), Ahlberg, Nilson and Walsh (1967), with some minor variants. Input to the FORTRAN code is the values of the base points $x_1, x_2, \ldots, x_N$ and the data available at these base points. In literature, base points are sometimes referred to as knots. Let data be available at N knots viz. $x_1, x_2, \ldots, x_N$ be $f(x_1), f(x_2), \ldots f(x_N)$. The domain (x_1, x_N) is considered as a combination of (N-1) subdomains ,Fig.1, $(x_1, x_2), (x_2, x_3), \ldots, (x_{N-1}, x_N)$ where width of any i^{th} domain (x_i, x_{i+1}) is represented by $h_i = x_{i+1} - x_i$. The procedure involves fitting a cubic polynomial $p_{3,i}(x)$ in the domain (x_i, x_{i+1}) for $i=1,2,\ldots N-1$. According to the notation used in this algorithm, the cubic polynomial $p_{3,i}(x)$ is the one which is fitted in the interval bounded by knots x_i and x_{i+1}. With this notation we can have utmost (N-1) polynomials in the domain (x_1, x_N). Any cubic polynomial in general, can be expressed as

$$P(x) = Ax^3 + Bx^2 + Cx + D \qquad (1)$$

Eq.(1) shows that the complete evaluation of any cubic polynomial involves evaluation of four constants, as A,B,C and D in Eq.(1), which requires a set of four equations. Since in the domain (x_1, x_N), we have

(N-1) cubic polynomials, so the total number of unknowns (coefficients of N-1 cubic polynomials) become 4N-4. For any unique evaluation of N-1 cubic polynomials, we must therefore have a set of 4N-4 simultaneous equations. These equations are obtained by the following procedure.

While fitting any cubic polynomial , $P_{3,i}(x)$, in the interval (x_i, x_{i+1}) we must ensure that the data values generated by the fitted cubic polynomial , $P_{3,i}(x)$, at the knots x_i and x_{i+1} must be identical with the values $f(x_i)$ and $f(x_{i+1})$, respectively. This constraint leads to the following 2N-2 equations.

$$p_{3,i}(x_i) = f(x_i) \quad ; \; i=1,2,...N-1 \tag{2.1}$$

$$p_{3,i}(x_{i+1}) = f(x_{i+1}) \; ; \; i = 1,2,...N-1 \tag{2.2}$$

Slope S of any curve y=f(x) at the point x is defined by the following relation.

$$S = [df/dx]_x \tag{3.1}$$

Similarly, curvature of the curve y=f(x) at the point x is computed ,Edwards(1961), using the following relation.

$$CURV = [d^2f/dx^2][1+(df/dx)^2]^{-3/2} \tag{3.2}$$

While fitting cubic polynomials, $p_{3,i}(x)$, in the domain (x_i, x_{i+1}) we must also ensure that slope of the cubic polynomials $p_{3,i-1}(x)$ and $p_{3,i}(x)$ must match at the interface knot x_i. This leads to the following set of N-2 equations.

$$p'_{3,i-1}(x_i) = p'_{3,i}(x_i) \quad ; \; i=2,3,\ldots N-1 \tag{4.1}$$

Here prime denoted first order derivative. After matching the slopes of the two cubic polynomials, i.e. first derivative, at the interface base point, x_i, we find from Eqs.(3.1) and (3.2) that the matching of second derivative at the interface base point x_i will ensure matching of curvatures of two cubic polynomials at the knot x_i. This leads to the following set of N-2 equations.

$$p''_{3,i-1}(x_i) = p''_{3,i}(x_i) \; ; \; i=2,3,\ldots,N-1 \tag{4.2}$$

Here double prime denotes second order derivtive. Eqs.(4.1), (4.2) account for all the interior knots of the fundamental domain (x,x_N) i.e. excluding the bounding knots x_1 and x_N. For these two bounding knots x_1 and x_N we prescribe the condition that curvatures of the cubic polynomials $p_{3,1}(x)$ and $p_{3,N-1}(x)$ are zero at x_1 and x_N, respectively. This condition leads to the following two equations.

$$p''_{3,1}(x_1) = 0 \tag{4.3}$$

$$p''_{3,N-1}(X_N) = 0 \tag{4.4}$$

Set of Eqs.(2.1),(2.2),(4.1),(4.2),(4.3) and (4.4) give in total [(N-1) + (N-1)+ (N-2) + (N-2) + 2] i.e. 4N-4 which is the required number for uniquely solving the set of 4N-4 simultaneous equations represented by the coefficients of N-1 cubic polynomials. From Eq.(1) we find that P"(x) represented by

$$P''(x) = 6Ax + 2B \tag{5}$$

is a linear function of x. This shows that for any cubic polynomial, $p_{3,i}(x)$, $p''_{3,i}(x)$ will have a linear variation in its domain (x_i,x_{i+1}) implying that the function $p''_{3,i}(x)$ can always be obtained by linear interpolation as in the following equation.

$$p''_{3,i}(x) = p''_{3,i}(x_i) + \text{DELTA} * (x-x_i) \tag{6.1}$$

where

$$\text{DELTA} = [p''_{3,i}(x_{i+1}) - p''_{3,i}(x_i)][x_{i+1}-x_i]^{-1} \tag{6.2}$$

Here DELTA represents the rate of variation of the linear function $p''_{3,i}(x)$ in the domain (x_i,x_{i+1}). Eqs.(6.1),(6.2) on some algebraic adjustments give the following functional form of $p''_{3,i}(x)$; i=1,3, ... N-1.

$$p''_{3,i}(x) = [(x_{i+1} - x)q_i]h_i + [(x - x_i)q_{i+1}]h_i \tag{7}$$

where

$$h_i = x_{i+1} - x_i \text{ ; } i = 1,2,\ldots, N-1 \tag{8.1}$$

$$q_i = p''_{3,i}(x_i) = p''_{3,i-1}(x_i) \text{ ; } i = 2,3,\ldots N-1 \tag{8.2}$$

For evaluating the functional form of the cubic polynomial, $p_{3,i}(x)$, Eq.(7) is integrated twice in succession giving the following relation.

$$p_{3,i}(x) = - q_i(x_{i+1} - x)^2/(2h_i) + Ax + B \tag{9.1}$$

where A and B are the constants of integration and are obtained by using the relation $p_{3,i}(x_i) = f(x_i) = f_i$. The following are the expressions of these constants A and B.

$$A = (f_{i+1} - f_i)\, h_i^{-1} + h_i(q_i - q_{i+1})/6 \tag{9.2}$$

and

$$B = x_i\,[h_i\, q_{i+1}/6 - f_{i+1}/h_i]$$

$$- x_{i+1}[\,h_i q_i/6 - f_i/h_i] \tag{9.3}$$

Eqs. (9.1),(9.2),(9.3) give the complete functional form of the cubic polynomial $p_{3,i}(x)$ in the following form.

$$p_{3,i}(x) = q_i(x_{i+1}-x)^3/(6h_i) + q_{i+1}(x-x_i)^3/(6hi)$$

$$+ [f_{i+1}/h_i - h_i q_{i+1}/6](x-x_i)$$

$$+ [f_i/h_i - h_i q_i/6](x_{i+1} - x) \tag{10}$$

Eq.(10) gives the explicit functional dependence of the cubic polynomial $p_{3,i}(x)$. For any numerical evaluation using the Eq.(10), we must first evaluate the values of q_i (second derivative of $p_{3,i}$) for all $i=1,2,\ldots,N$. This is achieved by differentiating the cubic polynomials $p_{3,i}(x)$ and $p_{3,i-1}(x)$ and evaluating them at the common knot x_i. On matching $p'_{3,i}(x_i)$ and $p'_{3,i-1}(x_i)$ at the common knot x_i in accordance with Eq.(4.1) we get the following recurrence relation for q_i; $i=2,3,\ldots N-1$.

$$(h_{i-1}/h_i)\ q_{i-1} + 2(1+h_{i-1}/h_i)\ q_i + q_{i+1} =$$

$$6/h_i[(f_{i+1}-f_i)/h_i - (f_i-f_{i-1})/h_{i-1}] \qquad (11.1)$$

Prescription of zero curvature at the bounding knots of the fundamental domain (x_1,x_N) gives us the following additional relations.

$$q_1 = 0 \qquad (11.2)$$

$$q_N = 0 \qquad (11.3)$$

Eq.(11.1) represents a set of N-2 simultaneous linear equations and these equations possess a tridiagonal structure. These equations can be solved using matrix algebra but still the better approach will be the one which takes care of the tridiagonal feature of these equations. Systems of tridiagonal equations occur notably in the finite difference solution of ordinary and partial differential equations. For furthr computations, we reexpress these equations in the following form.

$$b_2q_2 + q_3 = c_2$$

$$a_3q_2 + b_3q_3 + q_4 = c_3$$

$$a_4q_3 + b_4q_4 + q_5 = c_5 \qquad (12)$$

$$\cdots$$

$$a_iq_{i-1} + b_iq_1, + q_{i+1} = c_i$$

$$\cdots$$

$$a_{N-1}q_{N-2} + b_{N-1}q_{N-1} = c_{N-1}$$

where a_i, b_i, c_i involve $h_i, h_{i-1}, f_i, f_{i+1}, f_{i-1}$.
After a simple grouping of these equations it is seen that these equations posses a recursion solution of the following type.

$$q_i = Y_i - q_{i+1}/Z_i \; ; \; i=2,3,\ldots N-2 \tag{13}$$

From the first equation of Eq.(12) we find that

$$Z_2 = b_2 \tag{14.1}$$

and

$$Y_2 = C_2/Z_2 \tag{14.2}$$

Substitution of Eq.(13) in the ith equation of Eq.(12) gives the following recurrence relations for Y_i and Z_i.

$$Y_i = b_i - a_i/Y_{i-1} \tag{14.3}$$

$$Z_i = (C_i - a_i/Z_{i-1})/Y_i \tag{14.4}$$

On using Eq.(13) in the last equation of Eq.(12) and using Eqs.(14.3), (14.4) we get the following relation.

$$q_{N-1} = Y_{N-1} \tag{14.5}$$

From Eqs.(13), (14.1),(14.2),(14.3),(14.4) and (14.5), we find that the solution to the set of N-2 simultaneous linear equations represented by Eq.(12) is the following.

$$q_N = 0.0 \tag{15.1}$$

$$q_{N-1} = Y_{N-1}$$

$$q_i = Y_i - q_{i+1}/Z_i \; ; \; i = 2,3,\ldots N-2$$

$$q_1 = 0.0$$

WHERE

$$Z_2 = b_2 \qquad (15.2)$$

$$Y_2 = C_2/Z_2$$

$$Z_i = b_i - a_i/Z_{i-1} \; ; \; i = 3,4,\ldots,N-1$$

$$Y_i = (c_i - a_i \; Y_{i-1})/Z_i \; ; \; i=3,4,\ldots,N-1$$

AND

$$a_i = h_{i-1}/h_i \qquad (15.3)$$

$$b_i = 2(1+a_i)$$

$$= 2(1+h_{i-1}/h_i)$$

$$c_i = 6 \; h_i^{-1}[(f_{i+1} - f_i)/h_i^{-1} - (f_i - f_{i-1})/h_{i-1}]$$

where set of equations in Eq.(15.3) are valid for i=2,3,...N-1. Values of q_i thus obtained are substituted in the cubic polynomial, $p_{3,i}(x)$, of Eq.(10) for interpolating the value at any user specified base point x in the domain (x_{i+1},x_i).

Results

The results of application of this algorithm to two data sets are presented here with a provision of interpolating data on both regular and irregular grids.

Data set (i) : In this case, the FORTRAN code of the algorithm ,listed in the Appendix-A, was run with the INPUT generated by the FORTRAN library function ALOG10(X) for a set of base points ,X, generated by the FORTRAN loop X(I) = FLOAT(I*I)*0.1 for I=1,2,...,10. These base points (knots) were used in generating the values of the initial data by FORTRAN loop f(X)= ALOG10(X(I)) for I=1,2,...,10. The algorithm was run to interpolate the data values on both regular and irregular grids. Table 1A shows the INPUT for this data set i.e. set of base points X and the corresponding data values, f(X), at these X.For the regular grid we prescribed XMIN=1.15 , XMAX=10.00 and DELX=1.0. Table 1B shows the values of the base points of this uniform grid with the corresponding values of the interpolated and the actual values.Results show accuracy upto 3rd decimal places. Table 1C shows the OUTPUT analogous to Table 1B but for the irregular grid. Interpolation was carried out at the additional knots X=0.12,0.48,1.08,1.92,3.0,4.32,5.88,7.68 i.e. 8 iregularly spaced base points. Information recorded in Table 1C shows the values of base points of the combined grid i.e. initial plus the irregular grid of these 8 points. Results show a reasonable accuracy in the interpolated values.

Data set (ii) : The annual global mean vertical profile of water vapour mixing ratio (gm/gm) in the troposphere is consedered fer generating INPUT for this case.This profile,Katayama(1974), is defined as under.

$$f(p) = \begin{cases} 0.51\times10^{-3}[p/400]^{3.14} & ; \; p \geq 400\text{ mb} \\ 2.5\times10^{-6}\;[p/100]^{3.83} & ; \; 400\text{ mb} > p > 100\text{ mb} \\ 2.5\times10^{-6} & ; \; p \leq 100\text{ mb} \end{cases} \qquad (16)$$

For INPUT to the algorithm for this case, we considered X (=p) from 100 to 1000 mb at an interval of 100 mb. Table 2A is analogous to Table 1A but for data set (ii). Like the earlier case the algorithm was run for interpolation on both regular and irregular grids. For the former case we prescribed XMIN=150 , XMAX=900 and DELX=150. Table 2B shows the results for this class of interpolation.In this case some base points are common in the original and the grid of interpolated base points.The interpolated values are found to agree with the actual values upto 2nd decimal place. For interpolation on the irregular grid we choose these additional irregularly spaces knots i.e. X= 175,250,400,475,590,700,850,950. Table 2C shows the results analogous to Table 1C but for data set (ii).Results show an accuracy upto 2nd or 3rd decimal places.

It is quite possible that the precision of the interpolated values generated by this algorithm may increase further if we run the FORTRAN code of this algorithm with DOUBLE PRECISION.Instead of comparing the interpolated values with the corresponding values obtained from any standard FORTRAN mathematical library like SSP,NAg,IMSl etc. we have compared them with the actual values generated by the corresponding functions i.e. ALOG10(X) and f(p) of Eq.(16).

Epilogue

The algorithm described in this text can be used in any numerical problem requiring any of the two types of interpolations. FORTRAN code of this algorithm ,listed in Appendix-A, has some validation checks. These validation checks ensure that the algorithm is not subjected to any corrupt (i) data set or (ii) interpolation requirement. Following validations checks are incorporated in the FORTRAN code.

(1) Correctness of the value of N is checked in SUBROUTINE UNIFRM and SUBROUTINE VARBLE.

(2) Correctness of the value of NM1 is checked in SUBROUTINEs ABC QQ and P3X.

(3) Correctness of the values of XMIN , XMAX and DELX are checked in SUBROUTINE UNIFRM.

(4) Correctness of the values of XW(1) and XW(LAST) are checked in SUBROUTINE VARBLE.

Execution of the FORTRAN code is aborted if any of these checks gives FALSE result.

Acknowledgements

The author thanks the Director,IITM, for providing facilities for carrying out this work and to Dr. S.K.Mishra for encouragement. Author also thanks Dr. S.Rajamani and Dr.(Mrs.) P.S.Salvekar for reviewing this work. Author thanks Mrs. S.S.Naik for typing the manuscript and the DRAWING SECTION of IITM for preparing tracing of the figure. Utilisation of IITM's ND-560/Cx supermini computer system is thankfully acknowledged.

References

Ahlberg, J.,H., Nilson,E.,N., and Walsh,J.,L.,1967 : The theory of splines and their applications, Academic Press, pp.284.

Carnahan,Brice and Wilkes,O.,James,1973 : Digital computing and numerical methods,John Willey and Sons,Inc.,pp.477.

Edwards, Joseph,1961 : An elementary treatise on the differential calculus,Macmillan & Co. Ltd.,pp.529.

Katayama,A.,1974 : A simplified scheme for computing radiative transfer in the troposphere,Tech.Rept.No.6,Deptt. Met. UCLA,pp.77.

Schoenberg,I.,J.,1946 : Contributions to the problem of approximation of equidistant data by analytic functions,Quat.Appl.Math.,4,45-99,112-141.

+++++++++++++++

APPENDIX-A

LISTING OF ALGORITHM

A FORTRAN-77 ALGORITHM FOR CUBIC SPLINE INTERPOLATION FOR REGULAR AND IR-REGULAR GRIDS

.... DESCRIPTION OF VARIABLES ....

.... SCALAR VARIABLES ....

N	:	SCALAR VARIABLE REPRESENTING NUMBER OF DATA POINTS FOR INITIAL DATA SET (>3).
NM1	:	SCALAR VARIABLE REPRESENTING NUMBER OF DATA POINTS. (NM1=N-1).
XNEW	:	SCALAR VARIABLE REPRESENTING VALUE OF THE BASE POINT (KNOT) FOR INTERPOLATING DATA.
XMIN	:	SCALAR VARIABLE REPRESENTING THE LOWER BOUND FOR REGULAR GRID INTERPOLATION.
XMAX	:	SCALAR VARIABLE REPRESENTING THE UPPER BOUND FOR REGULAR GRID INTERPOLATION.
DELX	:	SCALAR VARIABLE REPRESENTING THE INTERVAL FOR REGULAR GRID INTERPOLATION.
NR	:	SCALAR VARIABLE REPRESENTING THE NUMBER OF BASE POINTS IN THE INTERPOLATED DATA.
NW	:	SCALAR VARIABLE REPRESENTING THE NUMBER OF BASE POINTS FOR INTERPOLATION IN THE IR-REGULAR GRID.

.... ARRAY VARIABLES ....

A,B,C	:	ARRAY VARIABLES REPRESENTING COEFFICIENTS OF QM'S. (SEE TEXT).
Y,Z	:	ARRAY VARIABLES OF RECURRENCE SOLUTION (SEE TEXT).
F	:	ARRAY VARIABLE OF SIZE N REPRESENTING INITIAL DATA.
Q	:	ARRAY VARIABLE REPRESENTING SECOND DERIVATIVE OF THE CUBIC POLYNOMIAL AT N KNOTS X1,X2,....,XN.
X	:	ARRAY VARIABLE OF SIZE N REPRESENTING N KNOTS IN ASCENDING ORDER OF THE INITIAL DATA.
XW	:	ARRAY VARIABLE OF SIZE NW REPRESENTING NW KNOTS IN ASCENDING ORDER FOR INTERPOLATION IN THE IR-REGULAR GRID.
XR	:	ARRAY VARIABLE OF SIZE NR REPRESENTING NR KNOTS IN ASCENDING ORDER FOR THE INTERPOLATED DATA IN THE IR-REGULAR GRID.
RESULT	:	ARRAY VARIABLE OF SIZE N REPRESENTING INTERPOLATED DATA IN THE INTERPOLATED GRID OF SIZE NR.
H	:	ARRAY VARIABLE OF SIZE NM1 REPRESENTING DIFFERENCE BETWEEN

SUCCESSIVE KNOTS (X2-X1),(X3-X2),...

.... DESCRIPTION OF ROUTINES ....

```
ABC    : TO COMPUTE (A) THE ARRAY COEFFICIENTS AI,BI & CI
                        FOR I= 2,3,...,NM1
                    (B) THE ARRAY COEFFICIENT HI FOR I=1,2,...,NM1

YZ     : TO COMPUTE THE ARRAY COEFFICIENTS YI & ZI OF THE RECURRENCE
         SOLUTION FOR I=2,3,...,NM1.

QS     : TO COMPUTE THE ARRAY VARIABLE Q REPRESENTING THE VALUES OF
         THE SECOND DERIVATIVE OF THE CUBIC POLYNOMIAL P3(X) AT N
         BASE POINTS X1,X2,...,XN.

P3X    : TO DO THE INTERPOLATION AT THE BASE POINT XNEW.

UNIFRM : TO INTERPOLATE DATA IN THE REGULAR   GRID.

VARBLE : TO INTERPOLATE DATA IN THE IR-REGULAR GRID.
```

```
      SUBROUTINE UNIFRM(F,N,X,XMIN,XMAX,DELX,RESULT,NR)
      DIMENSION F(N),X(N),H(39),Q(40),A(2:39),B(2:39),C(2:39),
     &          Y(2:39),Z(2:39),RESULT(40)
      IF(N.LE.3) STOP ' INSUFFICIENT DATA (N < 4) ! JOB ABORTED ...'
      IF((XMIN.LT.X(1)).OR.(XMAX.GT.X(N))) STOP ' IMPROPER XMIN & XMAX '
      NM1=N-1
      CALL ABC(X,F,N,A,B,C,H,NM1)
      CALL YZ(A,B,C,NM1,Y,Z)
      CALL QS(Q,N,Y,Z,NM1)
      NR=0
      DO 1 XX=XMIN,XMAX,DELX
      NR=NR+1
    1 CALL P3X(RESULT(NR),F,X,Q,N,H,NM1,XX)
      RETURN
      END
      SUBROUTINE VARBLE(F,N,X,XW,NW,RESULT,XR,NR)
      DIMENSION F(N),X(N),H(39),Q(40),A(2:39),B(2:39),C(2:39),
     &          Y(2:39),Z(2:39),RESULT(40),XW(NW),G(40),XR(40)
      IF((XW(1).LT.X(1)).OR.(XW(NW).GT.X(N))) STOP ' IMPROPER  RANGE XW'
      NM1=N-1
      CALL ABC(X,F,N,A,B,C,H,NM1)
      CALL YZ(A,B,C,NM1,Y,Z)
      CALL QS(Q,N,Y,Z,NM1)
      DO 1 I=1,NW
    1 CALL P3X(G(I),F,X,Q,N,H,NM1,XW(I))
      NR=N+NW
      K=1
      NF=1
      NG=1
      DO 5 L=1,NR
      IF(X(NF)-XW(NG)) 2,4,3
    2 RESULT(K)=F(NF)
      XR(K)=X(NF)
      NF=NF+1
      K=K+1
      IF(NF-N)5,5,6
    3 RESULT(K)=G(NG)
```

```
      XR(K)=XW(NG)
      NG=NG+1
      K=K+1
      IF(NG-NW)5,5,7
    4 RESULT(K)=F(NF)
      XR(K)=X(NF)
      NG=NG+1
      NF=NF+1
      K=K+1
    5 CONTINUE
    6 NR=K-1
      RETURN
    7 DO 8 L=NF,N
      RESULT(K)=F(L)
      XR(K)=X(L)
      K=K+1
    8 CONTINUE
      NR=K-1
      RETURN
      END
      SUBROUTINE ABC(X,F,N,A,B,C,H,NM1)
      DIMENSION F(N),X(N),A(2:NM1),B(2:NM1),C(2:NM1),H(NM1)
      IF(NM1.NE.(N-1)) STOP ' IMPROPER VALUE OF NM1 ! JOB ABORTED ...'
      DO 1 I=1,NM1
    1 H(I)=X(I+1)-X(I)
      DO 2 I=2,NM1
      A(I)=H(I-1)/H(I)
      B(I)=2.*(1.+A(I))
    2 C(I)=6.*((F(I+1)-F(I))/H(I)+(F(I-1)-F(I))/H(I-1))/H(I)
      RETURN
      END
      SUBROUTINE YZ(A,B,C,NM1,Y,Z)
      DIMENSION A(2:NM1),B(2:NM1),C(2:NM1),Y(2:NM1),Z(2:NM1)
      Z(2)=B(2)
      Y(2)=C(2)/Z(2)
      DO 1 I=3,NM1
      Z(I)=B(I)-A(I)/Z(I-1)
    1 Y(I)=(C(I)-A(I)*Y(I-1))/Z(I)
      RETURN
      END
      SUBROUTINE QS(Q,N,Y,Z,NM1)
      DIMENSION Q(N),Y(2:NM1),Z(2:NM1)
      IF(NM1.NE.(N-1)) STOP ' IMPROPER VALUE OF NM1 ! JOB ABORTED ...'
      Q(1)=0.
      Q(N)=0.
      Q(NM1)=Y(NM1)
      DO 1 I=NM1-1,2,-1
    1 Q(I)=Y(I)-Q(I+1)/Z(I)
      RETURN
      END
      SUBROUTINE P3X(VALUE,F,X,Q,N,H,NM1,XNEW)
      LOGICAL HERE
      DIMENSION F(N),X(N),H(NM1),Q(N)
      IF(NM1.NE.(N-1)) STOP ' IMPROPER VALUE OF NM1 ! JOB ABORTED ...'
      DO 1 I=1,N
      IF(XNEW.EQ.X(I)) GO TO 4
    1 CONTINUE
      DO 2 I=1,NM1
      HERE=(XNEW.GT.X(I)).AND.(XNEW.LT.X(I+1))
      IF(HERE) GO TO 3
    2 CONTINUE
      STOP ' SOMETHING WRONG WITH THE LOGIC ! CHECK IT UP ...'
    3 A=Q(I)*(X(I+1)-XNEW)**3/(6.*H(I))
      B=Q(I+1)*(XNEW-X(I))**3/(6.*H(I))
      C=(XNEW-X(I))*(F(I+1)/H(I)-H(I)*Q(I+1)/6.)
      D=(X(I+1)-XNEW)*(F(I)/H(I)-H(I)*Q(I)/6.)
      VALUE=A+B+C+D
      RETURN
    4 VALUE=F(I)
      END
```

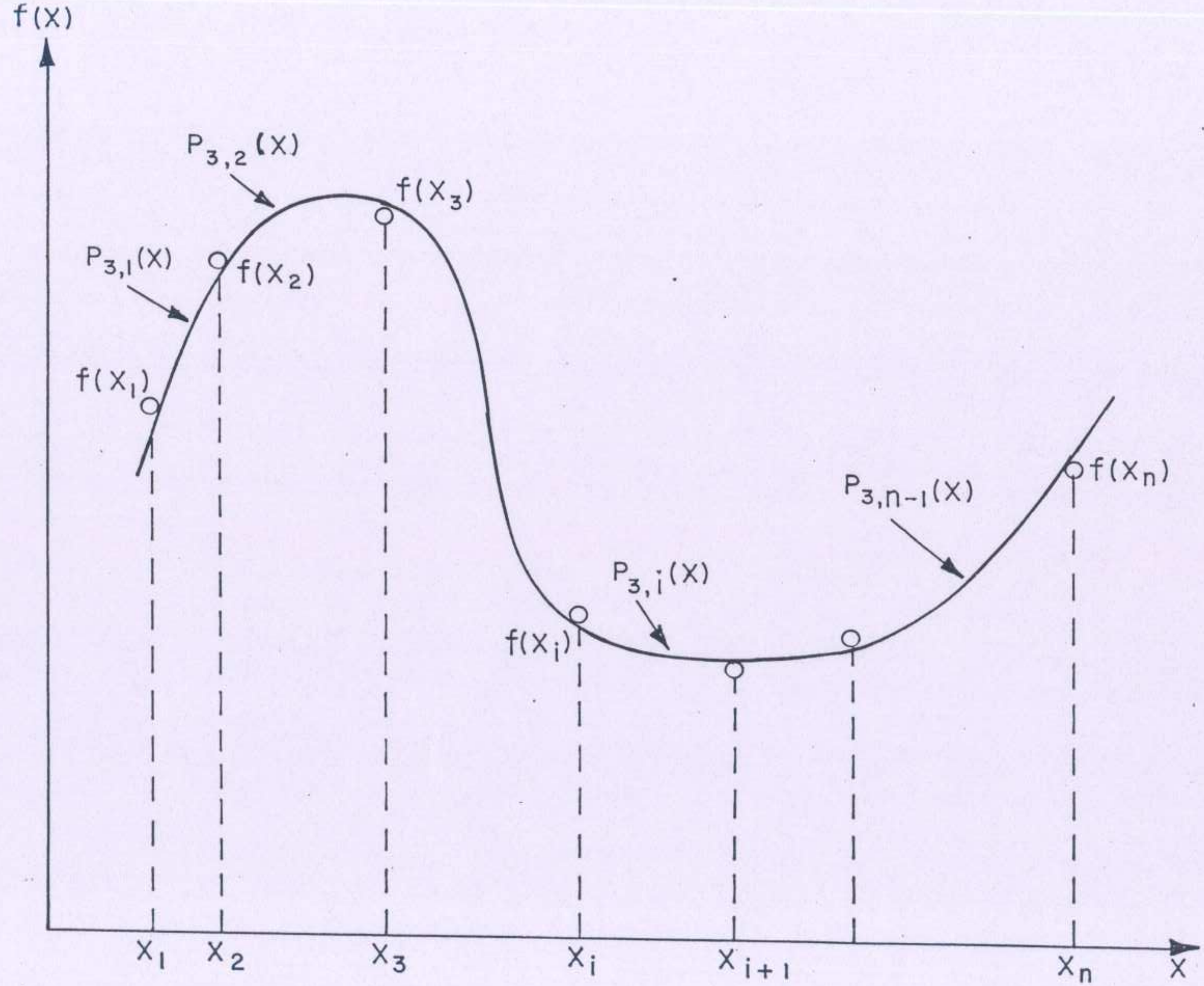

Fig.1 Cubic spline interpolation.

Table 1A : Values of the base points and the corresponding data values f(x), used as INPUT for Data set (i)

x	f(x)
0.1	- 1.00000
0.4	- 0.39794
0.9	- 0.04575
1.6	0.20412
2.5	0.39794
3.6	0.55630
4.9	0.69019
6.4	0.80618
8.1	0.90848
10.0	1.00000

Table 1B : Values of the base points x, with the corresponding values of interpolated data, f(x), and actual value, F(x), for interpolation on a regular grid with XMIN = 1.15, XMAX = 10.0 and DELX = 1.0 for Data set (i)

x	f(x)	F(x)
1.15	0.04355	0.06069
2.15	0.33788	0.33243
3.15	0.49643	0.49831
4.15	0.61891	0.61804
5.15	0.71160	0.71180
6.15	0.78877	0.78887
7.15	0.85455	0.85430
8.15	0.91112	0.91115
9.15	0.96067	0.96142

Table 1C : Values of the base points, x, with the corresponding values of interpolated data, f(x), and the actual values, F(x), for interpolation on a irregular grid for Data set (i)

x	f(x)	F(x)
0.10	- 1.00000	- 1.00000
0.12	- 0.95491	- 0.92081
0.40	- 0.39794	- 0.39794
0.48	- 0.29210	- 0.31875
0.90	- 0.04575	- 0.04575
1.08	0.01766	0.03342
1.60	0.20412	0.20412
1.92	0.29009	0.28330
2.50	0.39794	0.39794
3.00	0.47485	0.47712
3.60	0.55630	0.55630
4.32	0.63625	0.63548
4.90	0.69019	0.69019
5.88	0.76917	0.76937
6.40	0.80618	0.80618
7.68	0.88556	0.88536
8.10	0.90848	0.90848
10.00	1.00000	1.00000

Table 2A : Values of the base points x and the corresponding data values, f(x) used as INPUT for Data set (ii)

x	f(x)
100	0.25000×10^{-5}
200	0.20000×10^{-4}
300	0.67500×10^{-4}
400	0.51000×10^{-3}
500	0.99609×10^{-3}
600	0.17212×10^{-2}
700	0.27332×10^{-2}
800	0.40800×10^{-2}
900	0.58092×10^{-2}
1000	0.79687×10^{-2}

Table 2B : Same as Table 1B but for XMIN = 150, XMAX = 950, DELX = 150 for Data set (ii)

x	f(x)	F(x)
150	0.18905×10^{-4}	0.84375×10^{-5}
300	0.67500×10^{-4}	0.67500×10^{-4}
450	0.74343×10^{-3}	0.72615×10^{-3}
600	0.17212×10^{-2}	0.17212×10^{-2}
750	0.33630×10^{-2}	0.33618×10^{-2}
900	0.58092×10^{-2}	0.58092×10^{-2}

Table 2C : Same as Table 1C but for data set (ii)

x	f(x)	F(x)
100	0.25000×10^{-5}	0.25000×10^{-5}
175	0.22323×10^{-4}	0.13398×10^{-4}
200	0.20000×10^{-4}	0.20000×10^{-4}
250	0.95362×10^{-5}	0.39062×10^{-4}
300	0.67500×10^{-4}	0.67500×10^{-4}
400	0.51000×10^{-3}	0.51000×10^{-3}
475	0.86323×10^{-3}	0.85402×10^{-3}
500	0.99609×10^{-3}	0.99609×10^{-3}
590	0.16357×10^{-2}	0.16666×10^{-2}
600	0.17212×10^{-2}	0.17212×10^{-2}
700	0.27332×10^{-2}	0.27332×10^{-2}
800	0.40800×10^{-2}	0.40800×10^{-2}
850	0.48880×10^{-2}	0.48938×10^{-2}
900	0.58092×10^{-2}	0.58092×10^{-2}
950	0.68540×10^{-2}	0.68322×10^{-2}
1000	0.79687×10^{-2}	0.79687×10^{-2}