An algorithm for the interpolation of functions using quintic splines

E. H. Mund, P. Hallet and J. P. Hennart (*)

ABSTRACT

A method is described for the interpolation of \( N \) arbitrarily given data points using fifth degree polynomial spline functions. The interpolating spline is built from a set of basis functions belonging to the fifth degree smooth Hermite space. The resulting algebraic system is symmetric and block-tridiagonal. Its solution is calculated using a direct inversion method, namely a block-gaussian elimination without pivoting. Various boundary conditions are provided for independently at each end point. The stability of the algorithm is examined and some examples are given of experimental convergence rates for the interpolation of elementary analytical functions. A listing is given of the two FORTRAN subroutines INSPL5 and SPLIN5 which form the algorithm.

1. INTRODUCTION

This paper is concerned with the interpolation of numerical data points using polynomial spline functions. Various algorithms have already been published on this subject which are characterized either by their choice of a set of basis functions [1, 2], or by the numerical method used to solve the resulting system of algebraic equations [3, 4]. Frequent use is made for instance, of the so-called B-splines [5] and iterative methods appear in SPATH’s work to invert a block-tridiagonal matrix.

In this work, a set of piecewise Hermite polynomials of fifth degree is used in which the unknown parameters may be determined by specifying continuity conditions at each node. A system of algebraic equations arises which is always symmetric and block-tridiagonal. The same boundary conditions as in SPATH’s work [4] are provided, but independently at each boundary. After balancing the system of equations [6] a direct method adapted from Cholesky’s inversion technique [7] is used, allowing an evaluation of the interpolating spline function at any point of interest.

2. GENERAL FORMALISM

Following GREVILLE [5], a quintic spline function (QSF) \( S(x) \) with \( N \) knots \( (x_1 < \ldots < x_N) \) may be defined as a function with the following properties:

a/ in each subinterval \( I_i = [x_i, x_{i+1}] \),

\[ i = 1, \ldots, N-1, \quad S(x) \text{ is given by some polynomial of degree five or less.} \]

b/ \( S(x) \in C^4[x_1, x_N] \)

For given \( x_i \), the QSF-S form a \( (N+4) \)-dimensional vector space. An interpolating QSF (i.e. with given \( S(x_i) \)), may thus be made unique when four well-chosen boundary conditions are added. In this algorithm the following conditions are provided:

1. Type I : \( S^I \) and \( S^II \) are given
2. Type II : \( S^III = S^IV = 0 \)
3. Type III : \( S^II \) given \( S^IV = 0 \)

If boundary conditions of the same type are used at both extremities, one gets the same set of QSF-S as considered by SPATH in his algorithm [3]. But here, the choice is made independently at each end.
point. By the way, condition of type I at each boundary would give KERSHAW's D2 QSF [8], whereas condition of type II at $x_1$ and $x_N$ corresponds to the standard natural QSF [5].

Consider a subinterval $I_i$ with $i = 1, \ldots, N - 1$. Let $Q_i(x)$ be the unique Hermite-type interpolation polynomial whose value, first and second derivatives at each end-point of $I_i$, are respectively $y_i(j), y_i(j+1), j = 0, 1, 2$.

$$Q_i(x) = \sum_{j=0}^{2} h_{i}^{j} y_i(j) s_{j}(1 - \xi)$$

with $h_{i} = x_{i+1} - x_{i}$

$$\xi = (x - x_{i}) / h_{i}$$

and $S_{0}(\xi) = (1 - \xi)^3 (1 + 3 \xi + 6 \xi^2)$

$$S_{1}(\xi) = \xi (1 - \xi)^3 (1 + 3 \xi)$$

$S_{2}(\xi) = \xi^2 (1 - \xi)^3 / 2$

Now consider the set of polynomials $Q_i(x)$, $i = 1, \ldots, N - 1$ with given $y_i$ and yet unknown $y'_i$ and $y''_i$. They obviously form a piecewise polynomial function $Q(x)$ of class $C^2[x_1, x_N]$. $Q$ will moreover be a QSF if both third and fourth derivatives are continuous across each inner knot. These $2(N-2)$ conditions, along with four extra boundary conditions chosen among types I, II or III given above, define uniquely the unknowns $y'_i$ and $y''_i$ as the solutions of an algebraic system of linear equations.

In the general case, the system may be written

$$M Z = k$$

with

$$M = \begin{bmatrix}
A_1 & -B_1 & 0 \\
-B_1^T & A_2 & -B_2 \\
0 & -B_{N-2}^T & A_{N-1} & -B_{N-1} \\
-B_{N-1}^T & 0 & A_N
\end{bmatrix}$$

$$Z = (Z_1, Z_2, \ldots, Z_N)^T$$

and

$$k = (k_1, k_2, \ldots, k_N)^T$$

In these notations, $A_i, B_i$ represent the following $2 \times 2$ matrices:

$$A_i = \begin{bmatrix}
64 (g_i^3 - g_i^3) & 12 \alpha (g_i^3 - g_i^3) \\
12 \alpha (g_i^3 - g_i^3) & 3 \alpha^2 (g_i^3 - g_i^3)
\end{bmatrix}$$

$$B_i = \begin{bmatrix}
-56 g_i^3 & -8 \alpha g_i^2 g_i + 1 \\
8 \alpha g_i^3 & \alpha^2 g_i^2 g_i + 1
\end{bmatrix}$$

with $g_i = \frac{1}{h_{i}}$

where $Z_i$ and $k_i$ are 2-vectors given by

$$Z_i = (-y'_i, y''_i / \alpha g_i)^T$$

$$k_i = (-120 \frac{4}{g_i} (y_i + 1 - y_i) - 120 \frac{4}{g_i} (y_i - y_i-1),$$

$$-20 \alpha g_i^3 (y_i + 1 - y_i) - 20 \alpha g_i^3 (y_i - y_i-1))^T$$

First of all, let us point out that by performing a slight modification of the original unknowns, all the submatrices $A_i, B_i$ may be rendered homogeneous in the power of the partition size. Moreover a parameter $\alpha$ has been introduced and chosen in such a way as to minimize $\|A_i^{-1}\| (\|B_i^{-1}\| + \|B_i\|)$ where $\|\cdot\|$ is any matrix norm, e.g. the spectral norm. According to VARAH [6], this should improve the numerical stability of the algorithm chosen, namely the following extension of Cholesky's method [7]:

1. Forward elimination:

$$H_m = (A_m - B_{m-1} H_{m-1})^{-1} B_m$$

$$P_m = (A_m - B_{m-1} H_{m-1})^{-1} (k_m + B_{m-1} P_{m-1})$$

$$H_1 = A_1^{-1} B_1$$

$$P_1 = A_1^{-1} k_1$$

2. Backward substitution:

$$Z_N = P_N$$

$$Z_m = P_m + H_m Z_{m+1} \quad m = N-1, N-2, \ldots, 1$$

In the particular case of an homogeneous partition, the best $\alpha$ turned out to be $(64/3)^{1/2}$ : this value
was introduced into the algorithm for the general case.
If a boundary condition of type I is chosen at \( x_1 \) and (or) \( x_N \), the corresponding “unknowns” are automatically skipped and a system of reduced order is solved along similar lines. For a boundary condition of type III, for instance at \( x_1 \), \( A_1, B_1 \) and \( k_1 \), instead of being of the general form explicited here-above, become:

\[
A_1 = \begin{bmatrix}
64 g_1^3 & -12 \alpha g_1^3 \\
0 & 3 \alpha^2 g_1^3 \\
\end{bmatrix}
\]

\[
B_1 = \begin{bmatrix}
-56 g_1^3 & -8 \alpha g_1 g_2 \\
0 & 0 \\
\end{bmatrix}
\]

\[
k_1 = (-120 g_1^2 y_2 - y_1^1, 3 \alpha g_1^2 y_1^2)
\]

Similar expressions would be obtained at \( x_N \).

At this point, it is worthwhile to mention that in case of natural spline boundary conditions, the same general set of algebraic equations (2) would have been obtained using the extremal properties of the spline functions [9]. Indeed, it is easy to show that by introduction of \( Q_i(x) \) \( i = 1, \ldots, N \) into the quadratic functional

\[
I = \int_{x_1}^{x_N} (\sigma(x))^2 dx \quad \sigma(x) \in H^3[x_1, x_N]
\]

and minimization of \( I \), with respect to the unknown first and second derivatives in \( Q_i(x) \), one obtains exactly the same symmetric, blocktridiagonal linear system. \( H^3[x_1, x_N] \) is the set of functions which, on \([x_1, x_N]\) have an absolutely continuous second derivative and whose third derivative is square integrable.

Finally, having solved the linear system, values of the spline function and its successive derivatives may be determined with the aid of expression (1).

3. DESCRIPTION OF THE ROUTINES

The package is made of two FORTRAN subroutines: INSPL5 and SPLIN5. The subroutine INSPL5 computes the elements of the matrix \( M \) and the vector \( k \) and solves the system of equations (2). The evaluation of the elements and the forward elimination are performed simultaneously to reduce core storage and improve coding efficiency.

The user provides:

- \( N \): Number of data points
- \( X(N) \): vector giving their abscissa:
  \( X(I) = x_i \quad i = 1, \ldots, N \)
- \( \text{IND1, INDN} \): boundary condition switches at \( x_1 \) and \( x_N \) respectively. For both switches, one has the correspondence:
  - Type I : +1
  - Type II : 0
  - Type III : -1
- \( H \): auxiliary vector with at least \( 6N - 3 \) components
- \( \text{CF}(3, N) \): coefficients matrix. The first row must contain the given ordinates:
  \( \text{CF}(1, I) = y_i \quad i = 1, \ldots, N \)

The subroutine returns the values of the first two derivatives of the interpolating spline into the other rows of \( \text{CF} \):

\[
\text{CF}(2, I) = y_i' \quad i = 1, \ldots, N \\
\text{CF}(3, I) = y_i'' \quad i = 1, \ldots, N
\]

When boundary conditions of type I or III are used, at one or both end-points, the corresponding values of the derivatives must, of course, be put into \( \text{CF}(2, 1), \text{CF}(3, 1), \text{CF}(2, N) \) or \( \text{CF}(3, N) \).

The subroutine SPLIN5 provides values of the interpolating spline and of its derivatives at any point. The user provides:

- \( XX \): abscissa where values of the function and its derivatives are required
- \( \text{CF}(3, N) \): coefficients of the matrix described above
- \( N \): number of points in the partition

The subroutine returns:

- \( F(3) \): vector with three components which contains the required values of the function and its derivatives.

4. NUMERICAL TESTS AND RESULTS

The algorithm has been applied to various elementary analytical functions (e.g. \( e^x \), sin \( x \), \( 1/(1+x^2) \) ...) on the interval \([0, 0.98]\) with \( 2^n (n = 1, \ldots, 6) \) subintervals. Analytical simulations of sharp, multilevel resonance cross sections such as those encountered in nuclear physics, were also considered in these tests. Three values of \( \delta \), the ratio of largest to smallest subinterval in the partition of \([0,0.98]\), were chosen in order to include some cases with strongly non-homogeneous partition sizes, namely: \( \delta = 1, 19 \) and \( 199 \). An extensive study of convergence orders was performed.
made for both natural and D12 spline functions. As predicted by theory [10], the absolute error for the ith derivative (i = 0, 1, 2) decreased as the (3-i)th power of h (largest subinterval width) for natural splines, and as its (6-i)th power for D12 splines. This was fairly well corroborated by our numerical tests, a sample of which has been displayed in Table 1.

A double precision version of our algorithm was run in the same conditions as hereabove, but with natural splines only. The single precision version was then applied, with the same knots to the resulting spline, in order to isolate the "finite accuracy errors" due to machine limitations. Indeed, in this case, no interpolation error appears since both the interpolating and the interpolated functions ought to be equal. For δ = 1, the interpolation accuracy, as measured by the maximum value of the absolute error, was the machine accuracy (i.e. 14 digits on a CDC 6500 computer). This error grew with increasing values of δ, but in a systematic way.

ACKNOWLEDGEMENTS

The authors would like to thank Professor P. J. LAURENT (Université Scientifique et Médicale de Grenoble) for many fruitful discussions about this work.

REFERENCES


### TABLE I. Experimental Convergence Orders of natural and D12 Quintic Functions for some elementary Analytical Functions

<table>
<thead>
<tr>
<th>Nat. S</th>
<th>( y(x) = \exp(x) )</th>
<th>( y'(x) )</th>
<th>( y''(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta )</td>
<td>( |\varepsilon|_{L^\infty} )</td>
<td>( \alpha )</td>
<td>( |\varepsilon|_{L^\infty} )</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1.34 (-2)</td>
<td>1.50 (-1)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.06 (-3)</td>
<td>2.69 (-2)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1.31 (-4)</td>
<td>6.70 (-3)</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>1.67 (-5)</td>
<td>1.71 (-3)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>2.11 (-6)</td>
<td>4.32 (-4)</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>2.66 (-7)</td>
<td>1.08 (-4)</td>
</tr>
<tr>
<td>199</td>
<td>3</td>
<td>3.55 (-2)</td>
<td>2.67 (-1)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.26 (-3)</td>
<td>5.49 (-2)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4.21 (-4)</td>
<td>1.43 (-2)</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>5.47 (-5)</td>
<td>3.68 (-3)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>6.98 (-6)</td>
<td>9.36 (-4)</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>8.80 (-7)</td>
<td>2.36 (-4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D12 S</th>
<th>( y(x) = 1/(1 + x^2) )</th>
<th>( y'(x) )</th>
<th>( y''(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta )</td>
<td>( |\varepsilon|_{L^\infty} )</td>
<td>( \alpha )</td>
<td>( |\varepsilon|_{L^\infty} )</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7.16 (-5)</td>
<td>5.73 (-4)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2.10 (-5)</td>
<td>2.63 (-4)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1.48 (-7)</td>
<td>4.31 (-6)</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>3.16 (-9)</td>
<td>1.48 (-7)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>5.31 (-11)</td>
<td>5.02 (-9)</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>8.46 (-13)</td>
<td>1.60 (-10)</td>
</tr>
<tr>
<td>199</td>
<td>3</td>
<td>4.18 (-3)</td>
<td>1.51 (-2)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.99 (-5)</td>
<td>3.06 (-4)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>2.68 (-6)</td>
<td>3.70 (-5)</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>6.87 (-8)</td>
<td>1.84 (-6)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>1.66 (-9)</td>
<td>8.65 (-8)</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>3.27 (-11)</td>
<td>3.37 (-9)</td>
</tr>
</tbody>
</table>

\(N\) : number of points in the partition of \([0,0.98]\)  
\(\delta\) : \(\max h_i / \min h_i\) ; \(i = 1, \ldots, N - 1\)  
\(\|\varepsilon\|_{L^\infty}\) : discrete \(L^\infty\) norm of the interpolation error, evaluated with \(10^3\) points covering \([0,0.98]\)  
\(\alpha^{(i)}\) : \(\log(\|\varepsilon\|_{L^\infty}^{(i)}/\|\varepsilon\|_{L^\infty}^{(i-1)}) / \log(h^{(i)}/h^{(i-1)})\) at any level \(i\) with \(2^i + 1\) interpolation points, \(i = 2, \ldots\)

SUBROUTINE INSPLS(N,X,IND1,INDN,CF,H)

CALCULATION OF THE PARAMETERS OF AN INTERPOLATING
QUINTIC SPLINE

INPUT
N NUMBER OF INTERPOLATION POINTS
X(N) VECTOR OF ABSCISSAE
CF(1,N) ORDINATES
IND1,INDN BOUNDARY CONDITION SWITCHES AT X1 AND XN
FOR BOTH SWITCHES ONE HAS THE CORRESPONDENCE
= 1 TYPE 1
= 0 TYPE 2 (CFR. TEXT FOR MORE DETAILS
=-1 TYPE 3

OUTPUT
CF(2,N) FIRST DERIVATIVES
CF(3,N) SECOND DERIVATIVES
H(6*N-3) AUXILIARY VECTOR

DIMENSION X(1),CF(3,11),H(1)

AS=64./3.
A=SQR(T(AS)
P1=0.0
P2=0.0
P3=0.0
Q1=0.0
Q2=0.0
SF1=0.0
SF2=0.0
IM=0

FORWARD ELIMINATION PROCEDURE FOR CHOLESKYS ALGORITHM
APPLIED TO BLOCK-TRIDIAGONAL SYSTEMS OF ALGEBRAIC
EQUATIONS.

I=1
GO TO 15
5  DI1=DS1
   DI2=DS3
   DI3=DS2
   DI4=DS4
10  SF1=FP4
    SF2=FP3*Q1*A
15  DP1=64.0*P3
    DP2=12.0*P2*Q1*A
    DP3= DP2
    DP4= 3.0*P1*Q2*A
    IF (I.LT.N) GO TO 30

BOUNDARY CONDITION AT XN

IF (INDN) 20,50,25
20  DP2=0.0
    DP4=1.0
    DI2=0.0

Journal of Computational and Applied Mathematics, volume I, no 4, 1975. 284
DI4=0.0
SF2=CF(3,N)*(X(N)-X(N-1))/A
GO TO 50
25  DP1=1.0
DP2=0.0
DP3=0.0
DP4=1.0
DI1=0.0
DI2=0.0
DI3=0.0
DI4=0.0
SF1=-CF(2,N)
SF2=CF(3,N)*(X(N)-X(N-1))/A
GO TO 50
C
30  P1=1.0/(X(I+1)-X(I))
P2=P1*P1
P3=P1*P2
Q1=P1
Q2=P2
IF (I.GE.(N-1)) GO TO 35
Q1=1.0/(X(I+2)-X(I+1))
Q2=Q1*Q1
35  FP =CF(1,I)-CF(1,I+1)
FP3=20.0*P3*FP
FP4=6.0*P1*FP3
DP1=DP1+64.0*P3
DP2=DP2-12.0*P3*A
DP3=DP3-3.0*P3*A
DP4=DP4+3.0*P3*A
DS1=-56.0*P3
DS2=-8.0*P3*A
DS3=-8.0*P2*Q1*A
DS4=-8.0*P2*Q1*A
SF1=SF1+FP4
SF2=SF2+FP3*P1*A
IF (I.GT.1) GO TO 50
C
DI1=DS1
DI2=DS3
DI3=DS2
DI4=DS4
C
C BOUNDARY CONDITION AT X1
C
50  IF (IND1) 40,55,45
40  DP2=0.0
DP4=1.0
DS2=0.0
DS4=0.0
SF2=CF(3,1)*(X(2)-X(1))/A
GO TO 55
45  DP1=1.0
DP2=0.0
DP3=0.0
DP4=1.0
DS1=0.0
C
DS2 = 0.0
DS3 = 0.0
DS4 = 0.0
SFI = -CF(2,1)
SF2 = CF(3,1)*(X(2)-X(1))/A
GO TO 55
C
50 DP1 = DP1 - DI1 * H(IM-3) - DI3 * H(IM-2)
DP2 = DP2 - DI2 * H(IM-3) - DI4 * H(IM-2)
DP3 = DP3 - DI3 * H(IM-1) - DI3 * H(IM-1)
DP4 = DP4 - DI2 * H(IM-1) - DI4 * H(IM-1)
DET = DP1 * DP4 - DP2 * DP3
IF (I.EQ.1) GO TO 60
SFI = SFI + DI1 * H(IM-5) + DI3 * H(IM-4)
SF2 = SF2 + DI2 * H(IM-5) + DI4 * H(IM-4)
C
60 H(IM+1) = (+DP4 * SFI - DP3 * SF2) / DET
H(IM+2) = (-DP2 * SFI + DP1 * SF2) / DET
IF (I.GE.N) GO TO 65
H(IM+3) = (+DP4 * DS1 - DP3 * DS2) / DET
H(IM+4) = (-DP2 * DS1 + DP1 * DS2) / DET
H(IM+5) = (+DP4 * DS3 - DP3 * DS4) / DET
H(IM+6) = (-DP2 * DS3 + DP1 * DS4) / DET
IM = IM + 6
I = I + 1
IF (I = 2) 5,10,5
C
C BACKWARD SUBSTITUTION AND SOLUTION OF THE ALGEBRAIC SYSTEM
C
65 CF(2,N) = -H(IM+1)
DPD1 = H(IM+2) * A
CF(3,N) = DPD1 / (X(N) - X(N-1))
IM = IM - 6
C
DO 70 I = 2, N
K = N + 1 - I
CF(2, K) = -H(IM+1) * H(IM+3) * CF(2, K+1) - H(IM+5) * DPD1 / A
DPD2 = H(IM+2) * A - H(IM+4) * CF(2, K+1) * A + H(IM+6) * DPD1
CF(3, K) = DPD2 / (X(K+1) - X(K))
DPD1 = DPD2
IM = IM - 6
70 CONTINUE
C
RETURN
END

SUBROUTINE SPLINS (N,X,CF,XX,F)

CALCULATION OF THE VALUES OF AN INTERPOLATING QUINTIC SPLINE AND OF ITS FIRST AND SECOND DERIVATIVES AT ANY POINT XX

INPUT
N  NUMBER OF INTERPOLATION POINTS
X  (N)  ABSCISSAE
CF(3,N)  ORDINATES AND FIRST TWO DERIVATIVES
XX  ABSCISSA WHERE VALUES OF THE FUNCTION AND ITS DERIVATIVES ARE COMPUTED

OUTPUT
F(1)  VALUE OF THE INTERPOLATING FUNCTION
F(2)  VALUE OF ITS FIRST DERIVATIVE
F(3)  VALUE OF ITS SECOND DERIVATIVE

DIMENSION F(3),X(1),CF(3,1)

I=1
XN=XX-X(1)
IF (XN) 5>20,10
5 I=2
GO TO 25
10 DO 15 I=2,N
  XN=XX-X(I)
  IF (XN) 25>20,15
15 CONTINUE
I=N
GO TO 25

20 F(1)=CF(1,I)
  F(2)=CF(2,I)
  F(3)=CF(3,I)
  GO TO 30

25 CC=CF(1,I-1)-CF(1,I)
  H=X(I)-X(I-1)
  XN=XX-X(I-1)
  XR1=XN/H
  XR2=XR1*XR1
  XR3=XR1*XR2
  Y=CF(1,I-1)+CC*XR3*(-10.*XR1*(+15.- 6.*XR1))
  W=CF(2,I-1)*XR1
  1  +CF(2,I-1)*XR3*(- 6.+XR1*(+ 8.- 3.*XR1))
  2  -CF(2,I-1)*XR3*(+ 4.+XR1*(- 7.+ 3.*XR1))
  U=CF(3,I-1)*XR2
  1  +CF(3,I-1)*XR3*(- 3.+XR1*(+ 3.- XR1))
  2  +CF(3,I-1)*XR3*(+ 1.+XR1*(- 2.- XR1))
  F(1)=Y+H*(W+H*U/2.)
  Y= 30.*CC*XR2*(- 1.+XR1*(+ 2.- XR1))
  W=CF(2,I-1)
  1  +CF(2,I-1)*XR2*(-18.+XR1*(+32.-15.*XR1))
  2  -CF(2,I-1)*XR2*(+12.+XR1*(-28.+15.*XR1))
  U=CF(3,I-1)*XR1*2.*
  1  +CF(3,I-1)*XR2*(- 9.+XR1*(+12.- 5.*XR1))

2  \*CF(3,1)*XR2*(- 3.*XR1*)(- 8.* 5.*XR1))
F(2)=Y/H+XRI*U/2.
XN=10.*XR1
Y= XN*XR1*(- 1.*XR1*)(- 3.*XR1))
W = CF(2*I-1)*(-6.*XR1*(+16.*XN))
1  =CF(2*I-1)*(+4.*XR1*(-14.*XN))
U=CF(3,I-1)
1  \*CF(3*I-1)*(-9.*XR1*(+18.*XN))*XR1
2  \*CF(3*I-1)*(+3.*XR1*(-12.*XN))*XR1
F(3)=(Y/H+XR1*W)*6./H+U
3) RETURN
C
END