1969

**Smoothing and Estimation Derivatives of Equispaced Data**

J. J. Casaletto

John R. Rice  
*Purdue University, jrr@cs.purdue.edu*

Report Number:  
69-035
SMOOTHING AND ESTIMATING DERIVATIVES OF
EQUISPACED DATA

by
J. J. Casaletto and J. R. Rice
May, 1969
CSD TR 35
ABSTRACT

This paper discusses various aspects of the smoothing and estimation of derivatives of equispaced data. The background for least squares polynomial smoothing is summarized. The various alternatives for programs in a computing center's library are discussed and a particular alternative is selected as most suitable. An algorithm named SMOOTH is given (in Fortran) which implements this alternative. SMOOTH estimates the smoothed value of the data or its first or second derivative based on specified polynomial degree and number of points to enter the smoothing. The paper concludes with a discussion of methods suitable to compute large arrays of smoothing weights. There are 3 appendices which contain, respectively, explicit formulas associated with Gram polynomials, explicit formulas for the smoothing weights and tables of initial segments of arrays for computing large tables of smoothing weights.
SMOOTHING AND ESTIMATING DERIVATIVES OF EQUISPACED DATA

1. MATHEMATICAL BACKGROUND

An algorithm for estimating the values and derivatives of equispaced, tabulated functions is presented. Let \{\( (x_j, y_j) \) \} \( j = -n, -n+1, \ldots, n \) be the data with \( h = x_{j+1} - x_j \). The algorithm is based on least squares polynomial approximation and computes smoothing weight coefficients to apply to the data.

We consider the GAWN (or Tchebycheff) polynomials \( \{P_{m,n}(x)\} \) of degree \( m \). These polynomials are orthogonal

\[
\sum_{j=-n}^{n} P_{m,n}(j) P_{k,n}(j) = 0 \quad m \neq k
\]

and are given explicitly by

\[
P_{m,n}(j) = \sum_{k=0}^{m} \frac{(-1)^{k+m(m+k)!}(n+j)!(2n-k)!}{(k!)^2(m-k)!(n+j-k)!(2n)!}
\]

Let \( Q_{m,n}(j) \) be the best least squares approximation of \( y_j \) by a polynomial of degree \( m \) on the \( 2n+1 \) points \( i = 0, \pm 1, \ldots, \pm n \). We have

\[
Q_{m,n}(j) = \sum_{k=0}^{m} a_k P_{k,n}(j)
\]

where \( a_k = \left[ \sum_{j=-n}^{n} y_j P_{k,n}(j) \right] / S_{k,n} \) and \( S_{k,n} = \sum_{j=-n}^{n} [P_{k,n}(j)]^2 \).

Therefore, we have the smoothed estimate of the data

\[
y_j \sim Q_{m,n}(j) = \sum_{k=0}^{m} \left[ \frac{\sum_{j=-n}^{n} y_j P_{k,n}(j)}{S_{k,n}} \right] P_{k,n}(j)
\]
and, in particular, we have the smoothed value for $y_o$, given by

$$y_o \sim Q_{m,n}(0) = \sum_{k=0}^{m} \frac{\sum_{i=n}^{n} y_i P_{k,n}(i)}{S_{k,n}} P_{k,n}(0)$$

$$= \sum_{i=-n}^{n} \left[ \sum_{k=0}^{m} \frac{P_{k,n}(i) P_{k,n}(0)}{S_{k,n}} \right] y_i .$$

The smoothing weights are denoted by

$$A_{n,i}^m = \sum_{k=0}^{m} \frac{P_{k,n}(i) P_{k,n}(0)}{S_{k,n}} .$$

Smoothed estimates for the first and second derivatives are found by differentiating $Q_{m,n}(i)$ and evaluating it for $j = 0$. Therefore,

$$\frac{dy_o}{dx} = y'_o \sim \frac{1}{h} Q'_{m,n}(0) = \frac{1}{h} \sum_{i=-n}^{n} \sum_{k=0}^{m} \frac{P_{k,n}(i) P'_{k,n}(0)}{S_{k,n}} y_i$$

$$\frac{d^2y_o}{dx^2} = y''_o \sim \frac{1}{h^2} Q''_{m,n}(0) = \frac{1}{h^2} \sum_{i=-n}^{n} \sum_{k=0}^{m} \frac{P_{k,n}(i) P''_{k,n}(0)}{S_{k,n}} y_i$$

The weights here are denoted, respectively, by $b_{n,i}^m$ and $c_{n,i}^m$. See Appendix II for $A_{n,i}^m$, $b_{n,i}^m$, and $c_{n,i}^m$ for $m = 1, 3, 5$.

2. ALTERNATIVES FOR APPLICATIONS AND COMPUTING WEIGHTS

A computing center's library should contain a routine which will, when given the arrays $X$ and $Y$, compute the smoothed values for $y_o$, $y'_o$, and $y''_o$. The major consideration for such a routine is to obtain the weights $A_{n,j}^m$, $b_{n,j}^m$, and $c_{n,j}^m$.
One possible method is to calculate, for any \( m \) and \( n \), \( P_{n,n}(j) \) for any \( j \) and to evaluate its derivatives at zero from explicit formulae. This method has the advantages that any requested weights can be calculated and the routine is relatively short. However, the computation time would be rather large.

At the other end of the spectrum, one can limit \( m,n \) and the maximum order of the derivative; and calculate and store the necessary weights on some auxiliary storage device. The advantage here is, of course, speed. However, it is disadvantageous to restrict \( m,n \) and the order of the derivative, although these might not be, in practice, great restrictions. The major disadvantage is the number of constants which must be stored. If the maximum \( m \) is 5, maximum \( n \) is 50 (i.e. 101 points), and only \( y_0, y'_0 \) and \( y''_0 \) are allowed, then, taking into account the facts that \( a_{2k+1}^{n,j} = a_{2k}^{n,j}, b_{2k+2}^{n,j} = b_{2k+1}^{n,j}, c_{2k+2}^{n,j} = c_{2k+1}^{n,j} \) for \( k = 0, 1, \ldots \), and the symmetry and anti-symmetry of these constants about \( j = 0 \), it requires some 10,579 constants. This can be reduced further by observing that \( a_{1}^{n,j} \) is independent of \( j \) and that \( b_{1}^{n,0} = b_{3}^{n,0} = b_{5}^{n,0} = 0 \). This gives 9,150 constants and involves some additional logic. This is a large block of storage for such a routine, but not overwhelmingly large if one wants the highest possible speed.

These methods represent two extremes. The first has a large capability and small storage requirements, but is slow. The second has a limited, though practical, capability and large storage requirements, but is fast.

We present an algorithm which makes a compromise on the time-storage-capability relationship; that is, using some explicit formulae, setting a practical restriction on \( m \) and computing smoothed values only for \( y_0, y'_0 \) and \( y''_0 \).
Note in Appendix II, that the numerators of the weights are polynomials in \( n \) and \( j \), while the denominators are polynomials in \( n \); therefore, the denominators are calculated separately. The numerators are denoted by \( W_{n,j} \). The procedure is to generate integers \( W_{n,j} \), form the sum \( \sum W_{n,j} y_j \), and divide by the integral denominator and the appropriate power of \( h \). Note, further, that \( n \) is fixed for each entry into the routine and thus the \( W_{n,j} \) are polynomials in \( j, j = 0, \pm 1, \pm 2, \ldots, n \).

This suggests the method of differences to evaluate \( W_{n,j} \). This method is more efficient for evaluating polynomials at a large number of equispaced points; whereas nested evaluation is more efficient for a smaller number of evaluations.

To make this quantitative, let \( P_n(j) \) be a polynomial of degree \( n \), to be evaluated at \( k \) equispaced points. Let the unit of work be an addition, and assume one multiplication is equivalent to \( \mu \) additions (\( \mu \) is thus a machine characteristic). Then nested evaluation requires \( nk(n+1) \) additions. Evaluation by differences requires \( (n+1) \) starting values, the construction of a difference table, and final evaluations. If the \( (n+1) \) starting values are obtained from nested evaluation, then for \( k=n+1 \), \( n(n+1)(\mu+1/2)+nk \) additions are necessary. The difference table method is more efficient when \( (k-n-1) \) is greater than \( (n+1)/2\mu \).

The difference table method can be improved by representing \( P_n(j) \) in a point-value form; i.e. by the vector \( (P_n(0), P_n(1), \ldots, P_n(n)) \). Using this approach, \( nk-n(n+1)/2 \) additions are necessary. Hence, this method is always more efficient than the nested form when the values of \( P_n(j) \) are required, at least, at each \( j = 0, 1, \ldots, n \).
One may use another method of representing the polynomial $P_n(j)$, the "forward difference diagonal" form with step $h$; i.e. by the vector

$$(\Delta_h^n P_n(0), \Delta_h^{n-1} P_n(0), \ldots, \Delta_h P_n(0), P_n(0)),$$

where $\Delta_h^i P_n(x)$ is the i-th forward difference of $P_n(x)$ with step-size $h$.

This representation eliminates the construction of the difference table, $P_n(0)$ is given and successive values of $P_n(j)$ are obtained by a "rippled" (using intermediate sums as summands) addition process.

Similarly, one may use the backward difference diagonal representation with step $h$; i.e.

$$(\nu_h^n P_n(0), \nu_h^{n-1} P_n(0), \ldots, \nu_h P_n(0), P_n(0))$$

where $\nu_h^i P_n(x)$ is the i-th backward difference of $P_n(x)$ with step-size $h$.

With this form each successive $P_n(j)$ is obtained with $n$ additions. In the calculation of the next value, the diagonal is updated so that it is the backward difference diagonal for the next $x$. Thus, the vector defining the polynomial is always changing except for the first element which, of course, is the constant difference. The advantages of this form are storage, only a one-dimensional array of $n+1$ elements is needed, and ease of coding.

Consider the example $P_4(x) = x^4 + 2x^3 + 5x^2 + 6x + 1$, the backward difference diagonal at 0 with step 1 is $B = (24, -24, 12, 2, 1)$. Now $P_4(0) = B(5) = 1$, then the simple string (a simple DO loop in Fortran),

$$B(1)+B(2) \rightarrow B(2)+B(3) \rightarrow B(3)+B(4) \rightarrow B(4)+B(5) \rightarrow B(5),$$

updates the diagonal and the vector $B$ is now the backward difference diagonal of $P_4(x)$ at 1. $B$ is now $(24, 0, 12, 14, 15)$, thus $P_4(1) = B(5) = 15$. 
3. REMARKS ON THE ALGORITHM "SMOOTH"

The input parameter list, Y, X, INIT, NDERV, NDEG, NPTS, LENGTH, for the function subprogram SMOOTH is described in the initial comment cards. We use the usual convention that the zero-th derivative is the function value. The limits and checks on the arguments are also described in the comment cards.

Now, if NDERV = 0 and NDEG = 0 or 1, then \( A_{n,j}^0 = A_{n,j}^1 = 1/(2n+1) \); hence, \( W_{n,j} = 1 \). If NDERV = 1 and NDEG = 1 or 2, then \( A_{n,j}^1 = B_{n,j}^2 = 3j/(n(n+1)(2n+1)) \); hence \( W_{n,j} = 3j \). And if NDERV = 1 with NDEG = 0, or, if NDERV = 2 with NDEG = 0 or 1, then SMOOTH is set to zero.

In the remaining cases, the \( W_{n,j} \) are polynomials in \( n \), and the coefficients are polynomials in \( j \) (we call these the \( n \)-polys of \( j \)). Because of symmetry the \( W_{n,j} \) are only generated for \( j = 0, 1, \ldots, n \) and the appropriate sign attached for negative \( j \). The parameters NDERV and NDEG point to an implicit triangular array of weights, whose columns are indexed by \( n \) (corresponding to NPTS) and whose rows are indexed by \( j \) (corresponding to the (INIT+\( j \))-th ordinate). Therefore, NPTS specifies the particular column of this array; i.e. \( W_{n,j}^{NPTS} \) for \( j = 0, 1, \ldots, NPTS \). Smooth places no restriction on NPTS.

Let \( m \) be the greatest integer in \( (k+1)/2 \) where \( k \) is the degree of \( W_{n,j} \) as a polynomial in \( j \). The procedure in SMOOTH is to evaluate \( W_{n,j} \) for \( j = 0, 1, \ldots, n \), storing them as \( B(k-m+j+1) = W_{n,j} \), then reflecting the appropriate values into \( B(1), \ldots, B(k-m) \). Thus the B-vector is now a point-value form of \( W_{n,j} \). After using these \( m \) weights, the B-vector is manipulated so as to become the backward difference diagonal at \( j = m \) with step 1. Now SMOOTH continually updates the B-vector,
using the \( \frac{W_{n,j}}{n,j} = 8(k+1) \) in the sum \( \sum_j \frac{W_{n,j}}{n,j}(Y(\text{INIT}+j)+\text{SIG}!\cdot Y(\text{INIT}-j)) \),

where \( \text{SIG} = \pm 1 \) as appropriate.

In order to evaluate the initial \( \frac{W_{n,j}}{n,j} \), we use nested evaluation. The \( n \)-polys are evaluated separately and combined with the powers of \( j \). Note that all the \( n \)-polys are also polynomials in \( n(n+1) \).

We note that for large \( n \), the \( \frac{W_{n,j}}{n,j} \) and the denominators are very large integers. Thus the use of integer arithmetic is limited by machine word length. \text{SMOOTH}, as presented, uses floating point, single precision arithmetic. If exact values for these coefficients are desired, we can scale down the \( \frac{W_{n,j}}{n,j} \)'s and their denominators by canceling a common factor, and/or using double precision arithmetic. The following common factors of the numerators and denominators exist:

\[
\begin{align*}
A^3_{n,j} & : 3 & A^5_{n,j} & : 2^2 \cdot 3 \cdot 5 = 60 \\
B^3_{n,j} & : 2 \cdot 3 \cdot 5 = 30 & B^5_{n,j} & : 2^4 \cdot 3^3 \cdot 5 \cdot 7 = 15120 \\
C^3_{n,j} & : 2 \cdot 3 \cdot 5 = 30 & C^5_{n,j} & : 2^2 \cdot 3^3 \cdot 5 \cdot 7 = 3780
\end{align*}
\]

4. THE COMPUTATION OF TABLES OF SMOOTHING WEIGHTS

\text{SMOOTH} is designed to calculate and use one specific set of weights. To obtain a routine to produce tables of these weights the starting values, \( \frac{W_{n,0}}{n,0}, \frac{W_{n,1}}{n,1}, \ldots, \frac{W_{n,m}}{n,m} \) (\( m \) is the greatest integer in \( (k+1)/2 \) where \( k \) is the degree of \( \frac{W_{n,j}}{n,j} \) as a polynomial in \( j \)), for each column are most efficiently generated by difference methods. For each pair \((\text{NDEV}, \text{NDEG})\), \( \text{NDEG}=3,5 \), one would use the point-value forms \( (\frac{W_{n,j}}{n,j}, \frac{W_{n+1,j}}{n+1,j}, \ldots, \frac{W_{m,j}}{m,j}) \) for \( j = 0, 1, \ldots, m \), where:

\[
(1) \quad N = \frac{\text{NDEG}+1}{2}, \quad \text{a minimum NPTS for NDEG},
\]
(2) \( \delta_i = L^i \) where \( L \) is the degree of \( W_{n,j} \) as a polynomial in \( n \), and

(3) \( \delta_n \) as above

and the point-value form for the denominator, in lieu of using any explicit formulae. This gives the initial segments of the first \((L+1)\) columns. Each column can be completed as in SMOOTH. To obtain the initial segments for additional columns, the \( n+1 \) vectors, above, should be manipulated so that they are backward difference diagonals at \( n=M \), and then updated for each column.

Two other methods for table generation are obtained by replacing the point-value forms, for the \( W_{n,j} \) noted above, by either the forward or the backward difference diagonals at \( j=0 \). The use of these diagonals facilitates program coding. Appendix III contains all the point-value, forward difference diagonal, and backward difference diagonal vectors for the cases allowed in SMOOTH. Note that the common factors have been canceled.

Finally, we note that SMOOTH may be used to compute tables of smoothing weights with its range of allowable arguments. One inserts write statements at the appropriate points (indicated by comment cards) and runs SMOOTH through the range of desired values of NDERV, NDEG and NPTS. This approach is much less efficient in computation time (and restricts the range of NDERV and NDEG), but it requires a trivial modification of SMOOTH.
FUNCTION SMOOTH(Y,X,INIT,NDERV,NUSEG,NPTS,LENGTH)

C SMOOTH OPERATES ON AN EQUISPACED DATA FUNCTION (X,Y) TO PRODUCE THE
C SMOOTHED VALUE OF THE FUNCTION, OR ITS 1ST OR 2ND DERIVATIVES, AT A
C SPECIFIED POINT.  THE USER MUST SPECIFY THE DEGREE AND THE NUMBER OF
C POINTS, AS DESCRIBED IN THE PARAMETER DESCRIPTION BELOW, TO BE USED
C IN SMOOTHING.  SMOOTHING IS DONE BY APPLYING WEIGHTS TO THE NEIGHBORING
C ORIGINATES.  THE WEIGHTS ARE DETERMINED BY THE LEAST SQUARES
C APPROXIMATION USING GRAM-POLYNOMIALS.

C PARAMETERS
C Y *** THE ARRAY OF ORIGINATES.
C X *** THE ARRAY OF NUSEG(SAD).  USED ONLY TO COMPUTE THE STEP-SIZE H.
C INIT *** THE SMOOTHED VALUE IS DESIRED AT X(INIT).
C NDERV *** THE ORDER OF THE DERIVATIVE WITH THE USUAL CONVENTION
C CONCERNING 0.  THE LIMITS AND NDERV= 0, 1, 2,
C NUSEG *** THE DEGREE OF THE APPROXIMATING POLYNOMIAL.
C THE LIMITS ARE 0 TO 9
C NPTS *** THE NUMBER OF POINTS TO BE USED IS 2* NPTS+1.
C I.E., NPTS POINTS ON BOTH SIDES OF X(INIT)
C LENGTH *** THE NUMBER OF ELEMENTS IN THE ARRAY X.
C USED ONLY TO CHECK IF REQUEST IS LEGITIMATE.
C LENGTHW ELIMINATE THIS CHECK.

DIMENSION X(1),Y(1),I(6),NUSEG(5)
REAL N2

C STEP-SIZE CALCULATION
N=1
IF(LENGT.LE.0) H=X(1)
IF(LENST.EQ.2) LENGTH=INIT+NPTS
LENGTH=1.0*LENGTH
IF(NL=LT.1) H=X(INIT+1)-X(INIT)

C PARAMETER CHECKS
DO 1 1=1,5
1 MESSAGE(1)=
IF(NDERV.LE.0.OR.NDERV.GT.2) MESSAGE(1)=1
IF(NDURV.GT.0.OR.NDURV.GT.3) MESSAGE(2)=1
IF(NDURV.GT.NPTS+NPTS) MESSAGE(3)=1
IF(INIT.LE.0) MESSAGE(4)=1
IF(INIT.LE.NPTS+INIT+NPTS+LENGTH) MESSAGE(5)=1
IF(N1SUG(1)+N1SUG(2)+N1SUG(3)+N1SUG(4)+N1SUG(5)+N1SUG(2)) GO TO 6
WRITE(6,*)

2 FORMAT(1/46A PARAMETER ERROR CALL TO SMOOTHING ROUTINE )
IF(N1SUG(1)+N1SUG(2)+N1SUG(3)+N1SUG(4)+N1SUG(5)+N1SUG(2)) WRITE(6,*)

3 FORMAT(1/46A ROUTINE NOT EQUIPPED FOR DERIVATIVE OF ORDER(2)
IF(N1SUG(2)+N1SUG(3)+N1SUG(4)+N1SUG(5)+N1SUG(2)) WRITE(6,*)

4 FORMAT(1/46A ROUTINE NOT EQUIPPED FOR SMOOTHING BY DEGREE(121
IF(N1SUG(2)+N1SUG(3)+N1SUG(4)+N1SUG(5)+N1SUG(2)) WRITE(6,*)

5 FORMAT(1/46A NO UNIQUE SOLUTION FOR DEGREE(1213 BASED ON 2(13,10H51)+1 POINTS)
IF(MCSHDL(4).LE.1) UNITE(1,6)
6 FORMAT(I4) DERIVATIVE REQUESTED AT NEGATIVELY INDEXED POINT 
IF(MCSHDL(5).LE.1) UNITE(16+71) LENGTH
7 FORMAT(13M ARRAY SIZE OF I3,J3,JN INDICATES NOT ENOUGH POINTS TO ACC 
710=QUOTE REQUEST/100TH NDTH THAT PARAMETER 6 INDICATES THAT TRICE TH 
72AT NUMBER PLUS ONE POINTS ARE TO BE USED IN THE APPROXIMATION )
RETURN

SMOOTH=0
IF(NDEG-.NDEV.LT.0) RETURN
L=1
KEY=2
SIGA=1
N2=NPTS-1*NPTS
N3=NPTS*NPTS+1
T2=TONPT+16.
IF(NDEV-1) 9,15,13

C UNDERIVED SMOOTHING

9 IF(NDEV=1.1) GO TO 11
C DEGREE 0 OR 1
SMOOTH=Y(INIT)
DO 10 K=2,NPTS
10 SMOOTH=SMOOTH+Y(INIT+K)+Y(INIT-K)
SMOOTH=SMOOTH/TON
RETURN
11 DEVA=.1%12)@STU
IF(NDEG=1) GO TO 12
C DEGREE 2 OR 3
D(2)=3,.4/2-.3,
D(3)=6.(2-19,
GO TO 22
C DEGREE 4 OR 5
D(2)=2*12.92-2.92,92+15.92
B(4)=.1(3)-1.5U+23.22,
B(5)=.1(3)-6.20+21.62,
DEVA=4*ST2*DEVA
GO TO 22

C 2ND DERIVATIVE SMOOTHING

12 DEVA=1%1%12)@STU
IF(NDEG=1) GO TO 14
C DEGREE 2 OR 3
D(2)=3.92,N2
D(3)=3.92+9.92
GO TO 22
C DEGREE 4 OR 5
14 DEVA=2*ST2(2.42-2.+DENO.
D(3)=11.105U*92+3072,1+92-31903)*NC

RETURN
C 1ST DERIVATIVE SMOOTHING

15 DENOM=N2*TAN^H
   SIGN=-1
   IF(INCLV=LT.2) GO TO 17
C DEGREE 1 OR 2
   DO 16 K=1,NPTG
      16 SMOOTH=SGRTN*FLOAT(K)*(Y(INIT+K)-Y(INIT-K))
         SMOOTH=3.*SMOTH/DENOM
         RETURN

17 DENOM=DE,0.*((T+2*1)*N2-23.8)
   IF(NDEO<(T+4)) GO TO 16
C DEGREE 3 OR 4
   B(2)=U
      B(3)=(75.*N2-10U).N2+60.
   B(4)=(15U.*N2-Y40.)*N2+330.
   GO TO 21
C DEGREE 5
   19 DENOM=4.*T2*(N2-6.)#DNL0.;1
   B(3)=U
C A1(1),A2(1), AND A6(6) ARE USED HERD AS TEMP STORAGE
   a(1)=14.25N2-8J,N2+90.
   a(2)=15.*N2-50J*N2+12.
   z=-a(1)*.4200.*N2-140.;
   d(1)=(1.45(100.*N2-300.)*N2+10U)+1190(2)-R(2)
   B(6)=-77J.*N2+1155.
   DO 19 J=.6,6
C H AS STEP-SIZE ALREADY INCLUDED IN DENOM. USED HERE AS TEMP STORAGE.
   H=J-3
   19 B(J)=(1-9T3.*N2+9J(6)+Z(2)+Z(4)*P+11)*)t
C SET U-VECTOR TO P-V FORM AND GENERATE PARTIAL SUM
   L=2
   KEY=3
   21 L=L+1
   42 U(KEY+1)=SIGND(U(KEY+1))
      IF( KEY.LT.3) U(1)=SIGND(U(3))
      SMOOTH=S(KEY)*Y(INIT)
   33 DO 24 K=.L,L
C TO OBTAIN THE FIRST PART OF STEP AS HAY. PRINT U(NEWK) HERE.
   24 SMOOTH=SMOOTH+U(KEY+1)*(Y(INIT+K)-SIGNY(INIT-K))
      IF(L1.EQ.NPTG) GO TO 26
C TRANSFORM U-VECTOR INTO BACKWARD DIFFERENTIAL DIAGONAL
   /N=KEY+L
   M=1
DO 25 J=1,K
25 B(J)=U(J+1)-U(J)

C GENERATE AND SPILL NPT HEIGHTS INTO SUM
   N1=N+1
DO 27 K=1,NPTS
   DO 26 J=1,M
26 B(J)=B(J)+B(J-1)
C TO OBTAIN THE REST OF THE B(I,J) ARRAY, PRINT B(I,J) HERE.
27 SMOOTH=SMOOTH+B(I,J)*Y(INIT+K)+SIG.*Y(INIT-K)
C EXECUTES STEP-SIZE ADJUSTMENT
26 SMOOTH=SMOOTH/DEGEN
RETURN
END
APPENDIX I: Gram Polynomials, Derivatives and Special Values

A. The Gram Polynomials

\[ P_{0,n}(j) = 1 \]
\[ P_{1,n}(j) = \frac{1}{n} \]
\[ P_{2,n}(j) = \frac{3j^2 - n(n+1)}{n(2n-1)} \]
\[ P_{3,n}(j) = \frac{5j^3 - (3n^2 + 3n - 1)j}{n(n-1)(2n-1)} \]
\[ P_{4,n}(j) = \frac{35j^4 - 5(6n^2 + 6n - 5)j^2 + 3n(n^2 - 1)(n+2)}{2n(n-1)(2n-1)(2n-3)} \]
\[ P_{5,n}(j) = \frac{63j^5 - 35(2n^2 + 2n - 3)j^3 + (15n^4 + 30n^3 - 35n^2 - 50n + 12)j}{2n(n-1)(n-2)(2n-1)(2n-3)} \]

B. The Derived Polynomials

\[ P'_0,n(j) = 0 \]
\[ P'_1,n(j) = \frac{1}{n} \]
\[ P'_2,n(j) = \frac{6j}{n(2n-1)} \]
\[ P'_3,n(j) = \frac{15j^2 - (3n^2 + 3n - 1)}{n(2n-1)(n-1)} \]
\[ P'_4,n(j) = \frac{10j(14j^2 - (6n^2 + 6n - 5))}{2n(n-1)(2n-1)(2n-3)} \]
\[ P'_5,n(j) = \frac{315j^4 - 105(2n^2 + 2n - 3)j^2 + (15n^4 + 30n^3 - 35n^2 - 50n + 12)}{2n(n-1)(n-2)(2n-1)(2n-3)} \]

C. The Second Derivatives

\[ P''_0,n(j) = 0 \]
\[ P''_1,n(j) = 0 \]
\[ P''_2,n(j) = \frac{6}{n(2n-1)} \]
\[ P''_3,n(j) = \frac{30j}{n(n-1)(2n-1)} \]
\[ P''_4,n(j) = \frac{420j^2 - 10(6n^2 + 6n - 5)}{2n(n-1)(2n-1)(2n-3)} \]
\[ P''_5,n(j) = \frac{1260j^3 - 210(2n^2 + 2n - 3)j}{2n(n-1)(n-2)(2n-1)(2n-3)} \]
D. SPECIAL VALUES

\( p_{k,n}(0) = 0, \) for \( k \) odd

\( p_{0,n}(0) = 1 \)

\( p_{2,n}(0) = \frac{n+1}{2n-1} \)

\( p_{4,n}(0) = \frac{3(n+1)(n+2)}{2(2n-1)(2n-3)} \)

\( p'_{k,n}(0) = 0, \) for \( k \) even

\( p'_{1,n}(0) = \frac{1}{n} \)

\( p'_{3,n}(0) = \frac{5n^2+3n-1}{n(n-1)(2n-1)} \)

\( p'_{5,n}(0) = \frac{15n^4+30n^3-50n^2+12}{2n(n-1)(n-2)(2n-1)(2n-3)} \)

\( p''_{k,n}(0) = 0, \) for \( k=0 \) and \( k \) odd

\( p''_{2,n}(0) = \frac{6}{n(2n-1)} \)

\( p''_{4,n}(0) = \frac{5(6n^2+6n-5)}{n(n-1)(2n-1)(2n-3)} \)

E. THE SUM OF THE SQUARES

\( s_{0,n} = 2n+1 \)

\( s_{1,n} = \frac{(n+1)(2n+1)}{3n} \)

\( s_{2,n} = \frac{(n+1)(2n+1)(2n+3)}{5n(2n-1)} \)

\( s_{3,n} = \frac{(n+1)(n+2)(2n+1)(2n+3)}{7n(n-1)(2n-1)} \)

\( s_{4,n} = \frac{(n+1)(n+2)(2n+1)(2n+3)(2n+5)}{9n(n-1)(2n-1)(2n-3)} \)

\( s_{5,n} = \frac{(n+1)(n+2)(n+3)(2n+1)(2n+3)(2n+5)}{11n(n-1)(n-2)(2n-1)(2n-3)} \)
APPENDIX II: FORMULAE FOR THE WEIGHTS

NOTATION:

\[ n_{h,j}^m = \sum_{k=0}^{m} \frac{p_{k,n}(j)P_{k,n}(0)}{S_{k,n}} \] denotes the \( j \)-th weight for \( m \)-th degree smoothing based on \( 2n+1 \) points.

\[ b_{h,j}^m = \sum_{k=0}^{m} \frac{p_{k,n}(j)P'_{k,n}(0)}{S_{k,n}} \] denotes the \( j \)-th weight for the 1st derivative by \( m \)-th degree smoothing based on \( 2n+1 \) points.

\[ c_{h,j}^m = \sum_{k=0}^{m} \frac{p_{k,n}(j)P''_{k,n}(0)}{S_{k,n}} \] denotes the \( j \)-th weight for the 2nd derivative by \( m \)-th degree smoothing based on \( 2n+1 \) points.

\[ a_{n,j}^1 = \frac{1}{2n+1} \]

\[ a_{n,j}^3 = \frac{3[(3n^2+3n-1)-5j^2]}{(2n-1)(2n+1)(2n+3)} \]

\[ a_{n,j}^5 = \frac{15[(63j^4-35(2n^2+2n-3)j^2 + (15n^2+30n^3-35n^2-50n+12)]}{4(2n-3)(2n-1)(2n+1)(2n+3)(2n+5)} \]

\[ b_{n,j}^1 = \frac{3j}{n(n+1)(2n+1)} \]

\[ b_{n,j}^3 = \frac{25[(3n^4+6n^3-3n^2-3n+1)]j^3 - 35[3n^2+3n-1]}{n(n+1)(n+2)(2n-1)(2n+1)(2n+3)} \]

\[ b_{n,j}^5 = \frac{(693j^5-35j^3-35j^2-50n+12)]j^5 - 35[4(3n^2+3n-1)(2n^2+11n+15)(2n^2-7n+6)}{4n(n-2)(n-1)(n+1)(n+2)(n+3)(2n-3)(2n-1)(2n+1)(2n+3)(2n+5)} \]
\[ c_{n,j}^1 = 0 \]

\[ c_{n,j}^3 = \frac{30(3^j - n(n+1))}{n(n+1)(2n-1)(2n+1)(2n+3)} \]

\[ c_{n,j}^5 = \frac{-15 \left[ 105(6n^2 + 6n - 5) \right]^4 - 3(196n^4 + 392n^3 - 196n^2 - 392n + 245)j^2 + (70n^6 + 210n^5 - 35n^4}{2n(n-1)(n+1)(2n-3)(2n-1)(2n+1)(2n+3)(2n+5)} \]
NESTED FORM FOR NUMERATORS

\[ a_{n,j}^1 = 1 \]
\[ a_{n,j}^3 = (9 \cdot n + 9) \cdot n - 3 - 15 \cdot j \cdot j \]
\[ a_{n,j}^5 = [945 \cdot j - ((1050n + 1050)n - 1575)] \cdot j \cdot j + (((15n + 35)n - 50)n + 12 \]

\[ b_{n,j}^1 = 3 \cdot j \]
\[ b_{n,j}^3 = [((-105n - 105)n + 35)] \cdot j + (((75n + 150)n - 75)n + 25) \cdot j \]
\[ b_{n,j}^5 = [(((1035n + 20790)n - 24255)n - 34650)n + 8316] \cdot j \cdot j + ((((-13230n - 39690)n + 33075)n + 132300)n - 37485)n - 110250)n + 26460] \cdot j \cdot j \]
\[ + (((((3675n + 14700)n - 7350)n - 7350)n + 111720)n + 26460)n - 44100)n + 10584] \cdot j \]

\[ c_{n,j}^1 = 0 \]
\[ c_{n,j}^3 = 90 \cdot j - (30n + 30)n \]
\[ c_{n,j}^5 = [((-9450n - 9450)n + 7875)] \cdot j \cdot j + (((8820n + 17640)n - 8820)n - 17640)n + 11025] \cdot j \cdot j \]
\[ + (((((-1050n + 525)n + 3150)n - 525)n + 3150)n] \]
SPECIAL FORMS FOR NUMERATORS, \( w_{n,0}, \ldots, w_{n,m} \)

LET \( \delta = (n+1)n \)

\[ a_i^1_{n,j} : 1 \]
\[ a_i^3_{n,0} : 96 - 3 \]
\[ a_i^3_{n,1} : A_i^3 \text{ (i.e. numerator of } a_i^3_{n,0}) - 15 \]
\[ a_i^5_{n,0} : (2256 - 750)\delta + 180 \]
\[ a_i^5_{n,1} : A_i^5 \text{ - } 10506 + 2520 \]
\[ a_i^5_{n,2} : A_i^5 \text{ - } 42006 + 21420 \]
\[ b_i^1_{n,j} : 3j \]
\[ b_i^3_{n,0} : 0 \]
\[ b_i^3_{n,1} : (756 - 180)\delta + 60 \]
\[ b_i^3_{n,2} : (1506 - 990)\delta + 330 \]
\[ b_i^5_{n,0} : 0 \]

SET: \( Z_1 = (\delta - 39)\delta + 90 \) \( Z_2 = (156 - 50)\delta + 12 \)
\[ :i = -7706 + 1155 \] \( Z = -Z_1(\cdot 206 - 140) \) \( Z_1 = Z_1((3006 - 300)\delta + 100) + 11Z_2Z_2 \)

THEN FOR \( j = 1, 2, 3 \)

\[ b_i^5_{n,j} : ((693jj + \delta)Z_2 + 2)jj + Z_1j \]
\[ c_{n,j}^1: 0 \]

\[ c_{n,0}^3: -306 \]

\[ c_{n,1}^3: c_{n,0}^3 + 90 \]

\[ c_{n,0}^5: (-10506 + 3675) + 3150 \]

\[ c_{n,1}^5: c_{n,0}^5 + (88206 - 27000) + 18900 \]

\[ c_{n,2}^5: c_{n,0}^5 + (352806 - 221760) + 170100 \]
APPENDIX III. Initial Segments of Special Arrays for Computing Weights by Differences

Values are given in the following tables so that complete tables of weights may be made by differences without any direct evaluation. Also given are the factors which will appear in both numerator and denominator of the formulae. These factors have been cancelled in these tables.

<table>
<thead>
<tr>
<th>TABLE I: 3RD DEGREE SMOOTHING</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda^n )</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

\( S_{3,n} = \text{denom} \)

| factors 3 \( \lambda_n^3 \) |
| even in \( j \), degree 2 |

<table>
<thead>
<tr>
<th>TABLE II: 5TH DEGREE SMOOTHING</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda^n )</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

\( S_{5,n} = \text{denom} \)

| factors \( 2^2 \cdot 3 \cdot 5 \) \( \lambda_n^5 \) |
| even in \( j \), degree 4 |
### TABLE III: 1ST DERIVATIVE BY 3RD DEGREE SMOOTHING

<table>
<thead>
<tr>
<th>j</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>56</td>
<td>290</td>
<td>882</td>
<td>2072</td>
<td>4160</td>
<td>8316</td>
<td>36036</td>
<td>120120</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>335</td>
<td>1351</td>
<td>3521</td>
<td>7445</td>
<td>334152</td>
<td>813560</td>
<td>1790712</td>
</tr>
</tbody>
</table>

\[ S_{3,n} = \text{denom} \times \text{factors} \times 2 \times 3 \times 5 \]
odd in \( j \), degree 3

### TABLE IV: 1ST DERIVATIVE BY 5TH DEGREE SMOOTHING

<table>
<thead>
<tr>
<th>j</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1485</td>
<td>20153</td>
<td>129528</td>
<td>564840</td>
<td>1926375</td>
<td>5531295</td>
<td>13972728</td>
<td>31954728</td>
<td>67481505</td>
<td>67481505</td>
<td>67481505</td>
<td>67481505</td>
</tr>
<tr>
<td>2</td>
<td>-297</td>
<td>15883</td>
<td>158508</td>
<td>823338</td>
<td>3079725</td>
<td>9351573</td>
<td>24509058</td>
<td>57548308</td>
<td>123735843</td>
<td>123735843</td>
<td>123735843</td>
<td>123735843</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>-9667</td>
<td>52458</td>
<td>603198</td>
<td>2930411</td>
<td>10157763</td>
<td>28816158</td>
<td>71227688</td>
<td>158937093</td>
<td>158937093</td>
<td>158937093</td>
<td>158937093</td>
</tr>
</tbody>
</table>

\[ S_{5,n} = \text{denom} \times \text{factors} \times 2^4 \times 3^5 \times 5 \times 7 \]
odd in \( j \), degree 5
### TABLE V: 2ND DERIVATIVE BY 3RD DEGREE SMOOTHING

<table>
<thead>
<tr>
<th>j</th>
<th>n</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-6</td>
<td>-12</td>
<td>-20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-3</td>
<td>-9</td>
<td>-17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ S_{3,n} = \text{denom} \begin{array}{cccccccc} 21 & 126 & 462 & 1,287 & 3,003 & 6,188 \end{array} \]

Factors 2·3·5

Even in j, degree 2

### TABLE VI: 2ND DERIVATIVE BY 5TH DEGREE SMOOTHING

<table>
<thead>
<tr>
<th>j</th>
<th>n</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-350</td>
<td>-1850</td>
<td>-6650</td>
<td>-18900</td>
<td>-45780</td>
<td>-98700</td>
<td>-194700</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-95</td>
<td>-1055</td>
<td>-4760</td>
<td>-15080</td>
<td>-38859</td>
<td>-87115</td>
<td>-176440</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>335</td>
<td>755</td>
<td>35</td>
<td>-4855</td>
<td>-19751</td>
<td>-54495</td>
<td>-124335</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ S_{5,n} = \text{denom} \begin{array}{ccccccccc} 660 & 8,580 & 60,060 & 291,720 & 1,108,536 & 3,527,160 & 9,806,280 & 24,515,700 \end{array} \]

Factors 2·3·5·7

Even in j, degree 4
Tables of Forward Differences Diagonals with Respect to n.

NOTE: These numbers are used right to left in the program description.

### Table 1a: 3rd Degree Smoothing

<table>
<thead>
<tr>
<th>j</th>
<th>AT n=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17 18 6</td>
</tr>
<tr>
<td>1</td>
<td>12 18 6</td>
</tr>
</tbody>
</table>

### Table 2a: 5th Degree Smoothing

<table>
<thead>
<tr>
<th>j</th>
<th>AT n=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>593 860 890 450 90</td>
</tr>
<tr>
<td>1</td>
<td>225 720 855 450 90</td>
</tr>
<tr>
<td>2</td>
<td>-90 300 750 450 90</td>
</tr>
</tbody>
</table>

### Table 3a: 1st Derivative by 3rd Degree Smoothing

<table>
<thead>
<tr>
<th>j</th>
<th>AT n=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>56 234 358 240 60</td>
</tr>
<tr>
<td>2</td>
<td>-7 342 674 480 120</td>
</tr>
</tbody>
</table>

### Table 4a: 1st Derivative by 5th Degree Smoothing

<table>
<thead>
<tr>
<th>j</th>
<th>AT n=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>1485 18668 90707 235230 365056 351820 207270 60600 9800</td>
</tr>
<tr>
<td>2</td>
<td>-297 16180 126445 395760 673592 680960 410760 137200 19600</td>
</tr>
<tr>
<td>3</td>
<td>33 -9700 71825 416790 871068 964740 606690 205800 24900</td>
</tr>
</tbody>
</table>

### Table 5a: 2nd Derivative by 3rd Degree Smoothing

<table>
<thead>
<tr>
<th>j</th>
<th>AT n=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-6 -6 -2</td>
</tr>
<tr>
<td>1</td>
<td>-3 -6 -2</td>
</tr>
</tbody>
</table>

### Table 6a: 2nd Derivative by 5th Degree Smoothing

<table>
<thead>
<tr>
<th>j</th>
<th>AT n=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-350 -1500 -3300 -4150 -3030 -1200 -200</td>
</tr>
<tr>
<td>1</td>
<td>-95 -960 -2745 -3870 -2974 -1200 -200</td>
</tr>
<tr>
<td>2</td>
<td>335 420 -1140 -3030 -2806 -1200 -200</td>
</tr>
</tbody>
</table>
Tables of Backward Difference Diagonals with Respect to \( n \).

NOTE: These numbers are used right to left in the program description.

<table>
<thead>
<tr>
<th>( j )</th>
<th>( AT \ n = 2 )</th>
<th>[ \text{Table 1b: 3rd Degree Smoothing} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17 12 6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12 12 6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( j )</th>
<th>( AT \ n = 3 )</th>
<th>[ \text{Table 2b: 5th Degree Smoothing} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>393 330 260 180 90</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>225 225 225 180 90</td>
<td>[ \text{Table 3b: 1st Derivative by 3rd Degree Smoothing} ]</td>
</tr>
<tr>
<td>2</td>
<td>-90 -90 120 180 90</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( j )</th>
<th>( AT \ n = 2 )</th>
<th>[ \text{Table 4b: 1st Derivative by 5th Degree Smoothing} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>56 56 58 60 60</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-7 28 74 120 120</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( j )</th>
<th>( AT \ n = 3 )</th>
<th>[ \text{Table 5b: 2nd Derivative by 3rd Degree Smoothing} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1485 1485 1485 1476 1470 1470 0 9800</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-297 -297 -339 -456 -648 -840 -840 0 19600</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>33 -1353 -3061 -5274 -7992 -10710 -10710 0 29400</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( j )</th>
<th>( AT \ n = 2 )</th>
<th>[ \text{Table 6b: 2nd Derivative by 5th Degree Smoothing} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-6 -4 -2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( j )</th>
<th>( AT \ n = 3 )</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-350 -320 -290 -260 -230 -200 -200</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-95 -111 -127 -148 -174 -200 -200</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>335 336 302 188 -6 -200 -200</td>
<td></td>
</tr>
</tbody>
</table>
Complete Initial Difference Tables For The Weights, Displaying The Point-Value, Forward Diagonal, and Backward Diagonal Forms At Minimum n.

<table>
<thead>
<tr>
<th>$A^3_{n,j}$</th>
<th>$j = 0$</th>
<th>0</th>
<th>12</th>
<th>18</th>
<th>24</th>
<th>point-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>backward</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>forward</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>j = 1</td>
<td>12</td>
<td>30</td>
<td>54</td>
<td>12</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A^5_{n,j}$</th>
<th>$j = 0$</th>
<th>393</th>
<th>1253</th>
<th>3003</th>
<th>6093</th>
<th>11063</th>
<th>350</th>
<th>860</th>
<th>1750</th>
<th>3020</th>
<th>4970</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>260</td>
<td>890</td>
<td>1540</td>
<td>1880</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>450</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>j = 1</td>
<td>225</td>
<td>945</td>
<td>2520</td>
<td>5400</td>
<td>10125</td>
<td>225</td>
<td>720</td>
<td>1575</td>
<td>2880</td>
<td>4725</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>855</td>
<td>1305</td>
<td>1845</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>450</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>j = 2</td>
<td>-90</td>
<td>210</td>
<td>1260</td>
<td>3510</td>
<td>7500</td>
<td>-90</td>
<td>300</td>
<td>1050</td>
<td>2250</td>
<td>3990</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>750</td>
<td>1200</td>
<td>1740</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>450</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$B^3_{n,j}$</th>
<th>$j = 1$</th>
<th>56</th>
<th>290</th>
<th>882</th>
<th>2072</th>
<th>4160</th>
<th>56</th>
<th>234</th>
<th>592</th>
<th>1190</th>
<th>2088</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>58</td>
<td>358</td>
<td>598</td>
<td>898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>240</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>j = 2</td>
<td>-7</td>
<td>335</td>
<td>1351</td>
<td>3521</td>
<td>7445</td>
<td>28</td>
<td>342</td>
<td>1016</td>
<td>2170</td>
<td>3924</td>
</tr>
<tr>
<td></td>
<td>74</td>
<td>674</td>
<td>1154</td>
<td>1754</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>480</td>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\begin{tabular}{c|c|c|c|c|c|c|c|c}
\hline
\( j \) & \( n \) & 1485 & 1486 & 1487 & 1470 & 1470 & 1470 & 9800 \\
\hline
1 & 1 & 129528 & 564840 & 1926375 & 5531295 & 13972728 & 31954728 & 67481505 \\
& & 18668 & 109375 & 435312 & 1361535 & 3604920 & 8441435 & 17982000 & 3552677 \\
& & 90707 & 325937 & 926223 & 2243385 & 4836513 & 9540567 & 17544777 \\
& & 235230 & 600285 & 1517162 & 2593128 & 4704054 & 8004210 \\
& & 365056 & 716876 & 1275966 & 2110926 & 3300156 \\
& & 351820 & 559090 & 834960 & 1189230 \\
& & 207270 & 275870 & 354270 & 68600 & 78400 \\
& & 0 \\
\hline
2 & 0 & 15883 & 158508 & 823338 & 3079725 & 9351573 & 24509058 & 57508308 & 123735843 \\
& & 16180 & 142625 & 664830 & 2256387 & 6271848 & 15157485 & 32999250 & 66227535 \\
& & 126445 & 522205 & 1591557 & 4015461 & 8885637 & 17841765 & 35228285 \\
& & 395760 & 1069952 & 2425904 & 4870176 & 8956128 & 15386520 \\
& & 673592 & 1354552 & 2446272 & 4085952 & 6430392 \\
& & 680960 & 1091720 & 1639680 & 2344440 \\
& & 410760 & 547960 & 704760 & 137200 & 156800 \\
& & 0 \\
\hline
3 & 19600 & & & & & & & \\
\hline
\end{tabular}
| $C^3_{n,j}$ | $j=0$ | -6 | -12 | -20 |
|           |      | -4 | -6  | -8  |
|           |      | -2 | -2  |     |
|           | $j=1$ | -3 | -9  | -17 |
|           |      | -4 | -6  | -8  |
|           |      | -2 | -2  |     |

| $C^5_{n,j}$ | $j=0$ | 350 | -1850 | -6650 | -18900 | -45780 | -98700 | -194700 |
|            |      | -320 | -1500 | -4800 | -12250 | -26880 | -52920 | -96000  |
|            |      | -290 | -3300 | -7450 | -14630 | -26040 | -43080 |         |
|            |      | -260 | -4150 | -7180 | -11410 | -17040 |         |         |
|            |      | -230 | -3030 | -4230 | -5630  |         |         |         |
|            |      | -200 |       | -1200 | -1400  |         |         |         |
|            | $j=1$ | -95  | -1055 | -4760 | -15080 | -38859 | -87115 | -176440 |
|            |      | -111 | -960  | -3705 | -10320 | -23779 | -48256 | -89325  |
|            |      | -127 | -2745 | -6615 | -13450 | -24477 | -41069 |         |
|            |      | -142 | -3870 | -6844 | -11018 | -16592 |         |         |
|            |      | -174 | -2974 | -4174 | -5574  |         |         |         |
|            |      | -200 |       | -1200 | -1400  |         |         |         |
|            |      | -200 |       |        |         |         |         |         |
|            | $j=2$ | 335  | 755   | 35    | -4855  | -19751 | -54495 | -124335 |
|            |      | 336  | 420   | -720  | -4890  | -14896 | -34744 | -69840  |
|            |      | 302  | -1140 | -4170 | -10006 | -19848 | -35096 |         |
|            |      | 188  | -3030 | -5836 | -9842  | -15248 |         |         |
|            |      | -6   | -2806 | -4006 | -5406  |         |         |         |
|            |      | -200 |       | -1200 | -1400  |         |         |         |
|            |      | -200 |       |        |         |         |         |         |