FLEXWAL: A COMPUTER PROGRAM FOR PREDICTING THE WALL MODIFICATIONS FOR TWO-DIMENSIONAL, SOLID, ADAPTIVE-WALL WIND TUNNELS

Joel L. Everhart

November 1983

FOR REFERENCE

NOT TO BE TAKEN FROM THIS ZOOM

Library Copy

Review for general release November 20, 1985

NASA
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665
A program called FLEXWAL for calculating wall modifications for solid, adaptive-wall wind tunnels is presented. The method used is the iterative technique of NASA TP-2031 and is applicable to subsonic and transonic test conditions. The program usage, program listing, and a sample case are given.
SUMMARY

A program called FLEXWAL is presented. The program was written to predict the wall modifications necessary to remove the upper and lower wall interference effects in solid, flexible, adaptive-wall wind tunnels. The procedure used is the iterative method of NASA TP-2081 and is valid for both subsonic and transonic test conditions. Instructions on program usage, functional statements of program routines, a program listing, and a sample case are given.
INTRODUCTION

The computer program FLEXWAL described in this report is used to calculate the required upper and lower boundary modifications necessary for removing wind-tunnel wall interference effects in solid flexible-wall wind tunnels. The method used is the iterative technique of reference 1 and requires experimentally obtained information as input. The program is completely self-contained and is written in a modular form so that changes to the program for specific applications or wind-tunnel facilities can be made with ease.

Both analytical and experimental applications of the program at subsonic and transonic speeds have been previously presented along with the convergence properties of the method (See ref. 1.). This report gives a general description of the method, instructions for using the program, sample input, functional statements of routines used, and a listing of the computer code.
GENERAL DESCRIPTION

When testing in a wind tunnel, the flow field about the model is constrained to something different than that which would be present in free air. The principal difference is referred to as the wall interference and is generally taken as a perturbation about the free-air case. Classical methods for determining the wind-tunnel wall interference make mathematical approximations of the perturbation from which correction formulae are derived. The present method is that of reference 1 and uses the Cauchy integral formula to analytically extend the real flow field in the wind tunnel to infinity by solving for an imaginary flow exterior to the wind tunnel. These two flows are then iteratively coupled at the boundary with continuity being established when the measured data match the results of calculated data at the wall. If there is a mismatch then the method predicts a correction to the wind-tunnel wall, new data are measured, and the continuity is again checked.

Functional statements of the routines used in the FLEXWAL program are given in Appendix A and a listing of the program is contained in Appendix B. The program is completely self-contained and modular in construction. All input data are read into subroutine INPT and an echo listing of the input is given in the output listing. Subroutine SCALE uses the Goethert scaling laws to remove the effects of compressibility at the test Mach number prior to application of the correction technique and then, after completion of the correction calculations, SCALE is called again so as to include the compressibility at the desired Mach number MSET. MSET
has been included for consistency during the iterative process because often the desired Mach number is slightly different than that recorded during the test due to experimental error. Subroutine SMOOTH is a simple least squares smoothing subroutine which may or may not be needed depending on the quality of the experimentally obtained data. For each set of five adjacent pressures, a least squares parabola is determined and the center pressure is calculated from the least squares solution. When all pressures have been smoothed, a single pass has been completed and the process will be repeated as specified by the IS parameter in the input. The subroutine WALLDTA determines values of the perturbation velocities and sets up an integration contour as specified in reference 1. A spline through the wall jack coordinates allows the wall position to be determined at intermediate stations and by taking the derivative of the spline curve, the wall slopes (i.e. vertical velocities) may also be obtained along the walls. In the present form of the computer code, the vertical velocity components on the upstream and downstream legs of the contour have been set equal to the average of those on the upper and lower walls at the same station. This can easily be changed if the appropriate data are measured or better approximated.

The actual application of the method is in subroutine VJACK. It is here that the contour integration is performed for the velocities at the wall jack stations. It is important to note that
the most upstream and downstream wall jacks do not actually have to lie on the data contour. The program will determine whether the velocity is being calculated at a field point or a point on the contour and will adjust the solution accordingly. The difference is a factor of 2 which can be seen by referring to reference 1.

Subroutine DELYJ calculates the predicted wall change by integrating slopes determined from the new jack velocities along each wall. Convergence of the method is declared when these changes become sufficiently small or when some other criteria such as negligible variations in the airfoil force coefficients is met.

Subroutine NEWYJ prints the results of the program. All other subroutines act in support of the previously noted routines.

PROGRAM USAGE

The program is written in the FORTRAN IV computer language and requires 42000 words of execution field on a CDC CYBER 173 computer. Typical program run times are about 2 seconds. Appendix C gives a sample case. The output listing gives an echo listing of the input data required to generate the results. A user's guide containing the input format, a glossary of all terms used in the input and output, and program restrictions are presented in the computer program and are found at the top of the program listing in Appendix B.
REFERENCES

APPENDIX A

SUMMARY OF PROGRAM ROUTINES AND THEIR FUNCTIONS
<table>
<thead>
<tr>
<th>ROUTINE NAME</th>
<th>EXTERNAL REFERENCES</th>
<th>FUNCTIONAL STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLEXWAL</td>
<td>AVGDY, DELYJ, INITIAL, INPT, NEWYJ, RMACHNO, SCALE, SMOOTH, UINFIN, VJACK, WALLDTA</td>
<td>This is the main or executive routine of a FORTRAN IV computer program for calculating the jack settings of the upper and lower walls of a two-dimensional solid flexible wall wind tunnel.</td>
</tr>
<tr>
<td>CURVD</td>
<td>INTRVL, SNHCHS</td>
<td>This function subprogram differentiates a curve at a given point using a spline under tension. The subroutine CURVI should be called earlier to determine certain necessary parameters.</td>
</tr>
<tr>
<td>CURVI</td>
<td>INTRVL, SNHCHS</td>
<td>This function subprogram integrates a curve specified by a spline under tension between two given limits. The subroutine HMCURVI should be called earlier to determine certain necessary parameters.</td>
</tr>
<tr>
<td>CURV2</td>
<td>INTRVL, SNHCHS</td>
<td>This function subprogram interpolates a curve at a given point using a spline under tension. The subroutine HMCURVI should be called earlier to determine certain necessary parameters.</td>
</tr>
<tr>
<td>IBDRY</td>
<td></td>
<td>This function subprogram determines whether a point is on the data surface or a field point.</td>
</tr>
<tr>
<td>INTRVL</td>
<td></td>
<td>This function subprogram determines the index of the interval (determined by a given increasing sequence) in which a given value lies.</td>
</tr>
<tr>
<td>ISTRIP</td>
<td></td>
<td>This function subprogram determines whether the strip being considered is a normal or singular strip.</td>
</tr>
<tr>
<td>RMACHNO</td>
<td></td>
<td>This function subprogram calculates the test Mach numbers.</td>
</tr>
<tr>
<td>RMS</td>
<td></td>
<td>This subroutine function determines the RMS value of array A.</td>
</tr>
<tr>
<td>SWITCH</td>
<td></td>
<td>This subroutine reorders the indices of array x from high to low.</td>
</tr>
<tr>
<td>ROUTINE NAME</td>
<td>EXTERNAL REFERENCES</td>
<td>FUNCTIONAL STATEMENT</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>UCAL</td>
<td></td>
<td>This function subprogram calculates the U component of velocity.</td>
</tr>
<tr>
<td>UINFIN</td>
<td></td>
<td>This function subprogram calculates the free-stream velocity.</td>
</tr>
<tr>
<td>AVGDYJ</td>
<td></td>
<td>This subroutine averages the top and bottom wall values of DYJ.</td>
</tr>
<tr>
<td>DELYJ</td>
<td>CURVI HMCURV1 RMS</td>
<td>This subroutine integrates the delta theta values to obtain new wall location predictions.</td>
</tr>
<tr>
<td>HMCURV1</td>
<td></td>
<td>This subroutine fits a cubic or tension spline through a set of Y vs. X values.</td>
</tr>
<tr>
<td>INITIAL</td>
<td></td>
<td>This subroutine presets certain arrays to zero.</td>
</tr>
<tr>
<td>INPT</td>
<td></td>
<td>This subroutine handles the input of the raw data as taken from the wind-tunnel test.</td>
</tr>
<tr>
<td>LSQSMTH</td>
<td></td>
<td>This subroutine provides a least squares smoothing for either a straight line or a parabolic fit of the input data.</td>
</tr>
<tr>
<td>NEWYJ</td>
<td>RMS</td>
<td>This subroutine writes the predicted results onto TAPE 6.</td>
</tr>
<tr>
<td>SCALE</td>
<td></td>
<td>This subroutine scales the input using the Goethert scaling rules.</td>
</tr>
<tr>
<td>SING</td>
<td></td>
<td>This subroutine evaluates the velocity component along the singular strip.</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>LSQSMTH</td>
<td>This subroutine controls the smoothing of the wind-tunnel data.</td>
</tr>
<tr>
<td>SNHCSH</td>
<td></td>
<td>This subroutine returns approximations to: [ \text{SINH}(X) = \text{SINH}(X) - X ] [ \text{COSH}(X) = \text{COSH}(X) - 1 ] and [ \text{COSHMM}(X) = \text{COSH}(X) - 1 - X^2/2 ] with relative error less than $3.24E-14$.</td>
</tr>
<tr>
<td>STRIP</td>
<td></td>
<td>This subroutine calculates the jack velocity component resulting from some strip excluding that containing the singularity.</td>
</tr>
<tr>
<td>ROUTINE NAME</td>
<td>EXTERNAL REFERENCES</td>
<td>FUNCTIONAL STATEMENT</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>VJACK</td>
<td>IBDRY, ISTRIP, STRIP</td>
<td>This subroutine integrates the velocity integral to obtain the jack velocities.</td>
</tr>
<tr>
<td>WALLDTA</td>
<td>CURVD, CURV2, HMCURV1, SWITCH, UCAL</td>
<td>This subroutine takes input data and calculates velocities and wall locations then fills appropriate array locations.</td>
</tr>
</tbody>
</table>
APPENDIX B

PROGRAM LISTING
FLEXWAL is a FORTRAN IV computer program for calculating the jack settings of the upper and lower walls of a two-dimensional solid flexible-wall wind tunnel as predicted by the method of NASA TP-2081. The program was written by Joel L. Everhart of NASA Langley Research Center.

Required input is read into the program from tapes and output is returned on tape6.

**INPUT:**

<table>
<thead>
<tr>
<th>READ ORDER</th>
<th>DESCRIPTION</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MFAD(I), I=1:8</td>
<td>HA10</td>
</tr>
<tr>
<td>2</td>
<td>TEST<em>RUN</em>POINT</td>
<td>315</td>
</tr>
<tr>
<td>3</td>
<td>RN, ALPHA, TEMP, PT, PINF, MSET</td>
<td>6F10.0</td>
</tr>
<tr>
<td>4</td>
<td>NJT, NJB, NTW, NDS, NWS, NUS, NRLAX,</td>
<td>13IS</td>
</tr>
<tr>
<td>5</td>
<td>IS(I), I=1:4, IAVG, IPRT</td>
<td>4F15</td>
</tr>
<tr>
<td>6</td>
<td>IOPT(I), I=1:4</td>
<td>4F10.0</td>
</tr>
<tr>
<td>7</td>
<td>EC(I), I=1:4</td>
<td>4F10.0</td>
</tr>
<tr>
<td>8</td>
<td>(XJ(I)+YJ(I), I=1:NJ)</td>
<td>2F10.0</td>
</tr>
<tr>
<td>9</td>
<td>(XD(I)+YD(I), I=1:ND)</td>
<td>3F10.0</td>
</tr>
</tbody>
</table>

**DEFINITION OF TERMS:**

- **ALPHA** - Model angle of attack
- **DUJ** - Predicted change in axial jack perturbation velocity
- **DVJ** - Predicted change in vertical jack perturbation velocity
- **DYJ** - Predicted change in wall position
- **EC** - End conditions on the spline through the jack locations. The spline is used to determine wall position between jacks and wall slopes.
- **HEAD** - Descriptor card of up to 80 characters.
- **IAVG** - Wall jack averaging parameter.
- At a given XJ station the upper and lower YJ position are
AVERAGED GIVING SYMMETRICAL RESULTS
IAVG=0, AVERAGING ON
IAVG=1, AVERAGING OFF

IOPT - SPLINE FND CONDITION OPTION PARAMETER
IOPT(1), UPSTREAM, UPPER WALL
IOPT(2), DOWNSTREAM, UPPER WALL
IOPT(3), UPSTREAM, LOWER WALL
IOPT(4), DOWNSTREAM, LOWER WALL
IOPT(1)=0, EC UNKNOWN
IOPT(2)=1, EC=D(Y)/DY
IOPT(3)=2, EC=D2(Y)/D(Y)

IPRT - PRINT CONTROL PARAMETER.
IPRT=0, MAXIMUM OUTPUT
IPRT=1, NORMAL OUTPUT

IS - LEAST SQUARES SMOOTHING PARAMETER FOR INPUT DATA ON EACH WALL OF CONTOUR. IS(I) IS THE NUMBER OF SMOOTHING PASSS MADE ON EACH WALL STARTING CLOCKWISE ON UPPER WALL.
IS(I)=0 IMPLIES NO SMOOTHING

MSET - DESIRED MACH NUMBER TO WHICH RESULTS ARE SCAL.

ND - TOTAL NUMBER OF PRESSURE MEASUREMENTS (CALCULATED INTERNALLY TO PROGRAM)

NRW - NUMBER OF PRESSURE MEASUREMENTS ON BOTTOM WALL (LOWER LEG OF CONTOUR)

NDS - NUMBER OF PRESSURE MEASUREMENTS ON DOWNSTREAM LEG OF CONTOUR

NTW - NUMBER OF PRESSURE MEASUREMENTS ON TOP WALL (UPPER LEG OF CONTOUR)

NUS - NUMBER OF PRESSURE MEASUREMENTS ON UPSTREAM LEG OF CONTOUR

NJ - TOTAL NUMBER OF JACKS (CALCULATED INTERNALLY TO PROGRAM)

NJR - NUMBER OF JACKS ON BOTTOM WALL

NJT - NUMBER OF JACKS ON TOP WALL

NPLAX - NUMBER OF WALL RELAXATION FACTORS

PINF - FREESTREAM STATIC PRESSURE, PSI OR PSF

POINT - DATA IDENTIFIER

PT - FREESTREAM TOTAL PRESSURE, PSI OR PSF

PLAX - VALUE OF WALL CHANGE RELAXATION FACTORS FOR FINAL RESULTS

RMS - ROOT-MEAN-SQUARE ERROR

RN - TEST REYNOLDS NUMBER

RUN - DATA IDENTIFIER

TEMP - FREESTREAM TOTAL TEMPERATURE, DEG. RANKINE

TEST - DATA IDENTIFIER

THETA - FLOW ANGLE AT WALL

THETAN - PREDICTED FLOW ANGLE AT WALL

UD - ON INPUT, MEASURED WALL PRESSURE COEFFICIENTS. ON OUTPUT, CALCULATED AXIAL VELOCITY.

UJ - AXIAL VELOCITY AT JACK

UJN - PREDICTED AXIAL JACK VELOCITY

VJ - VERTICAL VELOCITY AT JACK

VJN - PREDICTED VERTICAL VELOCITY AT JACK

XD - AXIAL STATION OF WALL ORIFICES

XJ - AXIAL STATION OF WALL JACKS

YD - NOMINAL VERTICAL LOCATION OF UPPER AND LOWER WALL ORIFICES.
ACTUAL VERTICAL LOCATION OF UPSTREAM AND DOWNSTREAM ORIFICES.
ACTUAL UPPER AND LOWER WALL ORDNATES ARE CALCULATED FROM A
SPLINE THROUGH THE JACK ORDINATES.

YJ  - VERTICAL LOCATION OF WALL JACKS.
YJN  - PREDICTED VERTICAL LOCATION OF WALL JACKS.

RESTRICTIONS:

1. WALL DATA ARE INPUT CLOCKWISE STARTING AT THE MOST UPSTREAM UPPER
   WALL ORIFICE.
2. THE JACK ORDINATES ARE INPUT FROM UPSTREAM TO DOWNSTREAM UPPER WALL
   AND FROM UPSTREAM TO DOWNSTREAM LOWER WALL.
3. USE CONSISTENT PRESSURE UNITS.
4. SMOOTHING OF UPSTREAM AND DOWNSTREAM DATA IS NOT INCLUDED.
   HOWEVER, NOTF IS MADE IN THE LISTING OF SUBROUTINE SMOOTH WHERE
   IT SHOULD BE INCLUDED IF DESIRED.
5. THE FLOW ANGLES (IE. VD/UD) ON THE UPSTREAM AND DOWNSTREAM
   PORTIONS OF THE DATA CONTOURS ARE PRESENTLY SET EQUAL TO THE
   AVERAGE OF THE SLOPES OF THE UPPER AND LOWER WALLS AT THAT POSITION.

PROGRAM FLEXWAL (INPUT,OUTPUT,TAPE5,TAPE6)
REAL M,MSET
INTEGER TEST,RUN,POINT
COMMON/JACK/NJ,NJT,MJXJ(100),YJ(100),UJ(100),VJ(100),
   SJN(100),VJN(100),YJN(100),DYJ(100)
COMMON/PARAM/HEAD(8),TEST,RUN,POINT,M,MRN,ALPHA,TEMP,PLAX(10),
   SNR,LP,PT,PINF,UIINF,IAVG,IS(4),IOPT(4),EC(4),GAMMA,MSET
COMMON/CONT/ND,NTW,NDS,NW,NUS,XD(100),YD(100),UD(100),VD(100)
DIMENSION WK1(1000),WK2(1000)
GAMMA=1.4
GASP=171.0
NWK1=1000
NWK2=100
CALL INITIAL(NWK2,WK2)
CALL INPT(IPRT)
M=MACHNO(GAMMA,PT,PINF)
UIINF=UIINF(GAMMA,GASP,TEMP,M)
WRITE(6,96)M,UIINF
IF(IPRT,FO,0)WRITE(6,99)
CALL SCALE(RJXJ,XJ,YJ,YJ,UJ,VJ,M,SET,O,I,IPPT)
IF(IPPT.EQ.O)WRITE(6,98)
CALL SCALE(NDXD,YD,WK2,KD,YD,M,SET,O,IPPT)
CALL SMOOTH(500,WK1(1),WK1(501),IS,IPRT)
CALL WAILDA(WK1,WK1,IPRT)
CALL VJACK(IPRT)
CALL DEFYJ(NWK2,WK2,NWK1,WK2,UPINF)
IF(IPRT.EQ.O)WRITE(6,97)
CALL SCALE(RJXJ,YJ,YD,VJN,M,SET,1,IPRT)
IF(IPRT.EQ.O)CALL AVGDYJ(RJXJ,YD)
DO 25 IN=1,NPLAX
CALL NEWYJ(IN,NWK2,WK2)
25 CONTINU
STOP
96 FORMAT(18M,INF=G12.6, UINF=G12.6//)
97 FORMAT(18M,/* CALCULATED JACK DATA*/)
98 FORMAT(18M,/* INPUT WALL DATA*/)
99 FORMAT(18M,/* INPUT JACK DATA*/)
END
C
THIS FUNCTION CALCULATES THE TEST MACH NUMBER

FUNCTION PMACHNO(GAMMA,PT,INF)
REAL M2
M2=(PT/INF)**((GAMMA-1.0)/GAMMA)
M2=(PT-1.0)*2.0/(GAMMA-1.0)
M2=MIN(M2,0.0)
PMACHNO=SQRT(M2)
RETURN
END
C
THIS FUNCTION CALCULATES THE FREESTREAM VELOCITY

FUNCTION UINFIN(GAMMA,GASR,TEMP,M)
REAL M
TINF=TEMP/(1.0+(GAMMA-1.0)*M*M/2.0)
A2=GAMMA*GASR*TINF
ASSORT(A2)
UINFNAAM
RETURN
END
C
THIS SUBROUTINE PRESETS CERTAIN ARRAYS TO ZERO

SUBROUTINE INITIAL(NWK2,WK2)
THIS SUBROUTINE HANDLES THE INPUT OF THE RAW DATA AS TAKEN FROM THE WIND TUNNEL TEST.

SUBROUTINE INPT(IPRT)
REAL M,MSET
INTEGER TEST,RUN,POINT
COMMON/JACK,NJ,NJT,NJR,XJ(100),YJ(100),UJ(100),VJ(100),
SUJ(100),VJN(100),YJN(100),DYJ(100)
COMMON/CONTOUR/ND,NTW,NDS,NBW,NUS,XD(100),YD(100),UD(100),VD(100)
DIMENSION WK2(NWK2)
DO 10 I=1,100
XJ(I)=0.0
YJ(I)=0.0
UJN(I)=0.0
VJN(I)=0.0
DYJ(I)=0.0
XD(I)=0.0
YD(I)=0.0
UD(I)=0.0
VD(I)=0.0
WK2(I)=0.0
10 CONTINUE
RETURN
END
ECHO CHECK OF INPUT DATA

WRITE (6, 296) (HEAD(I), I=1,8)
WRITE (6, 297) TEST, PUN, POINT
WRITE (6, 298) M, ALPHA, TEMP, PT, PINF, MSET
WRITE (6, 297) NJT, NUR, NTW, NDS, NBW, NUS, NRLAX, (IS(I), I=1,4), IAVG, IPRINT
WRITE (6, 297) (ILOPT(I), I=1,4)
WRITE (6, 298) (EC(I), I=1,4)
WRITE (6, 298) (RLAX(I), I=1, NRLAX)
WRITE (6, 298) (XJ(I), YJ(I), I=1, NJ)
WRITE (6, 287) (XD(I), YD(I), UD(I), I=1, ND)
RETURN

DIMENSION IS(4), WK1(NWK), WK2(NWK), DUM(10)
N1 = IS(1)
N2 = IS(2)
N3 = IS(3)
N4 = IS(4)

COMMON CONTOUR/NW, NTW, NDS, NRW, NUS, XD(100), YD(100), UD(100), VU(100)

THIS SUBROUTINE CONTROLS SMOOTHING OF THE WIND TUNNEL DATA

SUBROUTINE SMOOTH(NWK, WK1, WK2, IS, IPRINT)

CALL LSOSMTH(5, WK1(1), WK2(1), 5, DUM, 1, WK1(1), WK5, 0)
WK5 = WK5
CALL LSOSMTH(5, WK1(1), WK2(1), 5, DUM, 1, WK1(2), WK5, 0)
WK5 = WK5
NS = NTW - 2
DO 10 J = 3, NS
   10 CONTINUE

287 FORMAT (3X, 3F10.5)
288 FORMAT (3X, 2F10.5)
290 FORMAT (8A10)
291 FORMAT (16I5)
292 FORMAT (8F10.0)
294 FORMAT (2F10.0)
295 FORMAT (1H1, /* * INPUT DATA LISTING */
296 FORMAT (3X, 8A10)
297 FORMAT (3X, 16 I5)
298 FORMAT (3X, 16 I5)
299 FORMAT (3F10.0)
END
CALL LSOSMTH(5, WK1(IJ), WK2(IJ), 5, DUM, 1, WK1(J), WKS)  
WK2(J) = WKS

10 CONTINUE
IJ = NTW + 4  
NS = NS + 1  
CALL LSOSMTH(5, WK1(IJ), WK2(IJ), 5, DUM, 1, WK1(NS), WKS)  
WK2(NS) = WKS  
NS = NS + 1  
CALL LSOSMTH(5, WK1(IJ), WK2(IJ), 5, DUM, 1, WK1(NS), WKS)  
WK2(NS) = WKS

15 CONTINUE
IF(IPRT.EQ.1) GO TO 26
DO 20 I = NTW  
WRITE(6, 100) XD(I), UD(I), WK2(I)  
XD(I) = WK(I)
UD(I) = WK2(I)

20 CONTINUE
C***** SMOOTH DOWNSTREAM DATA
C***** SMOOTH BOTTOM WALL DATA
26 IF(N2.EQ.0) GO TO 51

51 IF(N3.EQ.0) GO TO 76
DO 55 I = 1, NEW  
IJ = NTW + NS + NEW + 1 - I  
WK1(I) = XD(I)  
WK2(I) = UD(I)

55 CONTINUE
DO 55 I = 1, NEW  
CALL LSOSMTH(5, WK1(I), WK2(I), 5, DUM, 1, WK1(I), WKS)  
WK2(I) = WKS  
CALL LSOSMTH(5, WK1(I), WK2(I), 5, DUM, 1, WK1(2), WKS)  
WK2(2) = WKS  
NS = NEW + 2  
DO 50 J = 3, NS  
IJ = J - 2  
CALL LSOSMTH(5, WK1(IJ), WK2(IJ), 5, DUM, 1, WK1(J), WKS)  
WK2(J) = WKS

50 CONTINUE
IJ = NBW + 4  
NS = NS + 1  
CALL LSOSMTH(5, WK1(IJ), WK2(IJ), 5, DUM, 1, WK1(NS), WKS)  
WK2(NS) = WKS  
NS = NS + 1  
CALL LSOSMTH(5, WK1(IJ), WK2(IJ), 5, DUM, 1, WK1(NS), WKS)  
WK2(NS) = WKS

65 CONTINUE
IF(IPRT.EQ.1) GO TO 76
DO 70 I = 1, NBW  
II = NTW + NS + NBW + 1 - I  
WRITE(6, 100) XD(II), UD(II), WK2(II)  

70 CONTINUE
XD(I)=WK1(I)
UD(I)=WK2(I)
70 CONTINUE
C***** SMOOTH UPSTREAM DATA
76 IF(N4.EQ.0) GO TO 99
99 RETURN
100 FORMAT(* XD=**F10.5*, UD=**F10.5*, UD(SMOOTHED)=**F10.5*)
END

C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C C

SUBROUTINE LSQSMT(M,X,Y,NC,MS,XS,YS,IC)
C
C*****************************************************************************
C
C LSQSMT IS A LEAST SQUARES SMOOTHING ROUTINE FOR EITHER A STRAIGHT LINE OR A PARABOLIC FIT OF THE INPUT DATA
C
C M--------NUMBER OF DATA POINTS USED IN THE SMOOTHING
C X,Y------1-D ARRAYS OF COORDINATES OF DATA TO BE SMOOTHED
C NC-------NUMBER OF LEAST SQUARES COEFFICIENTS TO BE COMPUTED PLUS 2
C NC=4----STRAIGHT LINE FIT
C NC=5----PARABOLIC FIT
C C-------1-D ARRAY OF LENGTH M+2 FOR COEFFICIENTS
C MS-------NUMBER OF POINTS TO BE INTERPOLATED AFTER FITTING DATA
C XS-------1-D ARRAY OF LENGTH MS CONTAINING LOCATION OF POINT TO BE SMOOTHED
C YS-------1-D ARRAY OF LENGTH MS CONTAINING SMOOTHED ORDINATE OF XS
C IC-------IC=0----COMPUTE COEFFICIENTS
C IC=1----COEFFICIENTS PREVIOUSLY COMPUTED AND SUPPLIED
C
C*****************************************************************************
C
DIMENSION X(M),Y(M),C(NC),XS(MS),YS(MS)
N=NC-2
IF(M.LT.N) GO TO 50
IF(IC.EQ.1) GO TO 21
SY=SX=SXY=SX2=SX2Y=SX3=SX4=0.0
DO 3 I=1,M
SX=SX+X(I)
SY=SY+Y(I)
3 CONTINUE
C(N+1)=SX/M $ C(N+2)=SY/M
DO 5 I=1,M
X(I)=X(I)-C(N+1)
Y(I)=Y(I)-C(N+2)
5 CONTINUE
DO 10 I=1,M
SXY=SXY*X(I)*Y(I)
SX2=SX2+X(I)*X(I)
IF(N.EQ.2) GO TO 10
SX2Y = SX2Y + X(I) * X(I) * Y(I)
SX3 = SX3 + X(I) * X(I) * Y(I)
SX4 = SX4 + Y(I) * X(I) * X(I) * X(I)

10 CONTINUE
IF (N.EQ.2) GO TO 5
C(3) = (SX2Y * SX2 - SXY * SX3) / (SX2 * SX4 - SX2 * SX2 * SX2)
C(2) = (SYX - C(3) * SX3) / SX2
C(1) = C(3) * SX2 / N
GO TO 1H

15 C(?) = SXY / SX2
C(1) = 0.0

18 DO 20 I = 1, N
X(I) = X(I) + C(N+1)
Y(I) = Y(I) + C(N+2)
20 CONTINUE

20 CONTINUE
IF (M.EQ.0) GO TO 35
21 IF (M.EQ.0) GO TO 35
22 DO 30 I = 1, N
XS(I) = XS(I) - C(N) - I
30 CONTINUE

30 CONTINUE
IF (M.EQ.0) GO TO 35
Y(I) = Y(I) + C(N+1)
35 CONTINUE

35 RETURN
50 PRINT 51, M, N
51 FORMAT (*MULT*N IN LQS*MTH. M=*I3, N=*I3)
STOP
END

THIS SUBROUTINE TAKES INPUT DATA AND CALCULATES VELOCITIES AND
WALL LOCATIONS THEN FILLS APPROPRIATE ARRAY LOCATIONS.

SUBROUTINE WALDTA(NYPP, YPP, IPRT)
REAL M, SET
INTEGER TEST, RUN, POINT
COMMON/JACK/NJ, NJT, NJB, XJ(100), YJ(100), UJ(100), VJ(100),
SJN(100), VJN(100), YJN(100), DJJ(100)
COMMON/PARAM/HEAD(8), TEST, RUN, POINT, M, RN, ALPHA, TEMP, KLAX(10),
XLAX(PT), PTNF, UINF, IAVG, IS(4), IOPT(4), EC(4), GAMMA, MSET
COMMON/CONST/NX, NTS, NNS, NNS, X0(100), Y0(100), UD(100), VD(100)
DTMFNSTON, YP, NYPP, JOPT(2)
CALL HUNPVL(XJ(1),YJ(1),YPP,NJT,0,OPT(1),EC(1),EC(2),IER,1)

DO 305 T=1,NJT
   VD(I)=CHPVM(XD(I),NJT,XJ(I),YJ(I),YPP,0,0)
   VO(I)=CHPVM(XD(I),NJT,XJ(I),YJ(I),YPP,0,0)
305 CONTINUE

DO 306 T=1,NJT
   VJ(I)=CHPVM(XJ(I),NJT,XJ(I),YJ(I),YPP,0,0)
306 CONTINUE

CALL HUNPVL(XJ(NJT+1),YJ(NJT+1),YPP,NJT,0,OPT(4),EC(3),EC(4),IER2)

DO 307 T=1,NJT
   [I]=NT+BS4*1
   VD(I)=CHPVM(XD(I),NJT,XJ(NJT+1),YJ(NJT+1),YPP,0,0)
   VO(I)=CHPVM(XD(I),NJT,XJ(NJT+1),YJ(NJT+1),YPP,0,0)
307 CONTINUE

DO 308 T=1,NJT
   [I]=NJT+1
   VJ(I)=CHPVM(XJ(I),NJT,XJ(NJT+1),YJ(NJT+1),YPP,0,0)
308 CONTINUE

VDAVG=(VD(NJT)+VD(NJT+MOD+1))/2.0

DO 315 T=1,NJT
   [I]=NT+1
   VD(I)=VDAVG
315 CONTINUE

VDAVG=(VD(NJT)+MOD+HE+VD(1))/2.0

DO 320 T=1,NJT
   [I]=NT+MOD+HE+1
   VD(I)=VDAVG
320 CONTINUE

CALCULATION OF VO VELOCITIES

VO(1)=CALU('UP',VU,VO(1),VD(1),CAL.DA)

VO(I)=VO(I)+VD(I)

CONTINUE
CALCULATION OF UJ AND VJ FROM UD DATA

JOPT(1)=1
JOPT(2)=1
EC1=0.0
ECN=0.0
ST=0.0
CALL HMCURV1(XD(1),UD(1),YPP,NTW,ST,JOPT,EC1,ECN,FPP)
DO 340 I=1,NJT
UJ(I)=CURV2(XJ(I),NTW,XD(1),UD(1),YPP,ST)
VJ(I)=VJ(I)*UJ(I)
340 CONTINUE
IP=NTW+NDS+1
CALL SWITCH(NRW,XD(IP))
CALL SWITCH(NBW,UD(IP))
CALL SWITCH(NBW,VD(IP))
JOPT(1)=1
JOPT(2)=1
EC1=0.0
ECN=0.0
CALL HMCURV1(XD(IP),UD(IP),YPP,NRW,ST,JOPT,EC1,ECN,FPP)
DO 350 I=1,NJR
UJ(NJT+I)=CURV2(XJ(NJT+I),NRW,XD(IP),UD(IP),YPP,ST)
VJ(NJT+I)=VJ(NJT+I)*UJ(NJT+I)
350 CONTINUE
CALL SWITCH(NRW,XD(IP))
CALL SWITCH(NRW,UD(IP))
CALL SWITCH(NRW,VD(IP))

REMOVE FREESTREAM COMPONENT FROM UD AND UJ TO OBTAIN THE PERTURBATION QUANTITIES

DO 355 I=1,NJ
UJ(I)=UJ(I)-UINF
355 CONTINUE
DO 360 I=1,ND
UD(I)=UD(I)-UINF
360 CONTINUE

CLOSE DATA CONTOUR

ND=ND+1
XD(ND)=XD(1)
YD(ND)=YD(1)
UD(ND)=UD(1)
VD(ND)=VD(1)

LOOK AT RESULTS OF THIS SUBROUTINE
419 IF(IP3X.EQ.1)GO TO 390
420 WRITE(6,398)UINF
421 DO 365 I=1,NJ
422 WRITE(6,399)I,XJ(I),YJ(I),UJ(I),VJ(I)
365 CONTINUE
423 WRITE(6,397)
424 DO 370 I=1,NB
425 WRITE(6,399)I,XD(I),YD(I),UD(I),VD(I)
370 CONTINUE
426 RETURN
427 FORMAT(H10/ WALL DATA (SCALED) */5X*10
428 *(RX*X0)*RXY*9*RY*J(6)*RX*VD*Y/)
429 FORMAT(H10/ *INF=*G12.6/*7X*JACK DATA (SCALED) */5X*10
430 *(RX*XJ0*RXY*J(6)*RY*J(6)*RX*VJ*Y/)
431 FORMAT(3X13,4(5Y15.9))
END

C

FUNCTION UCAL(M*INF,CP,THFTA,GAMMA)
REAL M
T2=1.0+THETA*THETA
IF(M.LE.0.0)GO TO 10
P=(GAMMA-1.0)/GAMMA
A1=1.0-(1.0+GAMMA*GAMMA*CP/2.0)*EP
A2=1.0+2.0*EP/(GAMMA-1.0)*EP
UCAL=INF*SGRT(A2/T2)
RETURN
10 A2=1.0-CP
IF(A2.LT.0.0)A2=0.0
UCAL=INF*SGRT(A2/T2)
RETURN
END

C

SUBROUTINE S1WITCH(N,X)
DIMENSION X(N)
DO 10 I=1,N
J=N+1-I
IF(J.LE.1)GO TO 11
S=X(J)
X(J)=X(I)
X(I)=S
10 CONTINUE
11 RETURN
END
C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C THIS SUBROUTINE SCALES THE INPUT USING THE GORETHERT SCALING RULES

SUBROUTINE SCALE(N,X,Y,U,V,M,IS,IT,IPRT)
REAL M,SET
DIMENSION X(N),Y(N),U(N),V(N),DY(N)
BETA2=1.0*M*
BETA=SPRT(PETA2)
RETAMS=SPRT(1.0-M*SET*SET)

C*****IS=0---COMPRESSIBLE TO INCOMPRESSIBLE
C*****IS=1---INCOMPRESSIBLE TO COMPRESSIBLE

IF(IS.EQ.0.AND.IPRT.EQ.0)WRITE(6,25)
IF(IS.EQ.1.AND.IPRT.EQ.0)WRITE(6,26)
IF(IT.EQ.0.AND.IPRT.EQ.0)WRITE(6,27)
IF(IT.EQ.2.AND.IPRT.EQ.0)WRITE(6,28)
IF(IT.EQ.3.AND.IPRT.EQ.0)WRITE(6,29)
IF(IS.EQ.0)GO TO 5
BETA=1.0/BETA
BETA2=1.0/BETA2
5 DO 10 I=1,N
X(I)=X(I)
Y(I)=PETA*Y(I)
IF(IS.EQ.1)DY(I)=RETAMS*DY(I)
U(I)=BETA2*U(I)
V(I)=BETA*V(I)
IF(IT.EQ.1.AND.IPRT.EQ.0)WRITE(6,27)I,X(I),Y(I)
IF(IT.EQ.2.AND.IPRT.EQ.0)WRITE(6,28)I,X(I),Y(I),U(I)
IF(IT.EQ.3.AND.IPRT.EQ.0)WRITE(6,29)I,X(I),Y(I),U(I),V(I)
10 CONTINUE
RETURN

25 FORMAT(/# COMPRESSIBLE TO INCOMPRESSIBLE SCALING)
26 FORMAT(/# INCOMPRESSIBLE TO COMPRESSIBLE SCALING)
27 FORMAT(2X,I3,S(2X,4F12.6))
28 FORMAT(/4X,I2*X,J2*X*Y,J2*X*Y)
29 FORMAT(/4X,I2*X*YD*12*X*YD*12*X*CP*)
30 FORMAT(/4X*I2*X*YJ*12*X*YJ*12*X*UJ*12*X*VJ*)
END

C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C THIS SUBROUTINE INTEGRATES THE VELOCITY INTEGRAL TO OBTAIN THE JACOBIAN VELOCITY

SUBROUTINE VJACK(IOPT)
DIMENSION IOPT(4)
COMMON/JACK, NJ, JT, NJH, XJ(100), YJ(100), HJ(100), VJ(100),
S(JJN(100), VJL(100), YJN(100), VDJ(100),
COMMON/CONTAIN/NJ, NTW, NDS, NJH, XN(100), YN(100), VN(100),
PI=ATAP(0.0, 1.0)
TVJ=0
NPI=1
DO 40 J=1, NJ
IF(J.EQ.1) WRITE(6,94)
TVJ(J)=0.0
UJN(J)=0.0
DO 30 I=1, NPI
ISTP=ISTRIJ(J, I)
IF(ISTR.I.EQ.0, 1) CALL STRIP(J, I, I, I, USTRIP, VSTRIP)
IF(ISTR.I.EQ.2) CALL SING(J, I, USTRIP, VSTRIP)
VJN(J)=VJN(J)+VSTRIP
UJN(J)=UJN(J)-USTRIP
IF(J.EQ.1) WRITE(6,99) J, I, XJ(J), YJ(J), XN(I), YN(I)
49 VJN(I+1), YN(I+1), VSTRIP, VJN(J), ISTR, USTRIP, UJN(J)
30 CONTINUE
PIDIV=PI
IF(JPDY(J).EQ.2) PIDIV=2.0*PI
VJN(J)=VJN(J)/PI
UJN(J)=UJN(J)/PI
40 CONTINUE
IF(IOCT(1).EQ.1) GO TO 42
VJN(1)=VJ(1)
UJN(1)=UJ(1)
42 IF(IOCT(2).EQ.1) GO TO 44
VJN(NJ)=VJ(NJ)
UJN(NJ)=UJ(NJ)
44 IF(IOCT(3).EQ.1) GO TO 46
VJN(NJT)=VJ(NJT)
UJN(NJT)=UJ(NJT)
46 IF(IOCT(4).EQ.1) GO TO 48
VJN(NJT+1)=VJ(NJT+1)
UJN(NJT+1)=UJ(NJT+1)
48 WRITE(6, 97)
DO 50 I=1, NJ
DOVJ=VJN(I)-VJ(I)
DOUJ=UJN(I)-UJ(I)
WRITE(6, 98) I, XJ(I), YJ(I), UJ(I), UJN(I), VJ(I), VJN(I), DOVJ, DOUJ
50 CONTINUE
RETURN
96 FORMAT(1MW)
97 FORMAT(1MW/9) NEW JACK PERTURBATION VELOCITY CALCULATIONS:
98 (SCALEF)/9/ 10
99 FORMAT(13L7(2XFR.4), 2(2XFR.4), 4(2XFR.4))

END

C

THIS SUBROUTINE CALCULATES THE JACK VELOCITY COMPONENT RESULTING FROM
SOME STRIP EXCLUDING THAT CONTAINING THE SINGULARITY

SUBROUTINE STRIP(I,K,L,USTRIP,VSTRIP)
COMMON/JACK/NJ,NT,NU,XJ(100),YJ(100),UJ(100),VJ(100),
NUJN(100),VJN(100),YJN(100),UDP+YJ(100)
COMMON/CONTOUR/MD,NTW,ATX,RJX,XD(100),YD(100),UD(100),VD(100)
XL=XD(I)-XD(K) & YL=YD(I)-YD(K) & SL=SQR(XL**2+YL**2+YL)
RX=XJ(I)-XD(K) & RY=YJ(I)-YD(K)
UDIF=UD(I)-UD(K) & VDIF=VD(I)-VD(K)
SL2=SL*SL
C1=XL*/UDIF/SL2 & C2=XL*VDIF/SL2
C3=YL*/UDIF/SL2 & C4=YL*VDIF/SL2
D1=RX*UDIF+RY*VDIF & D2=-2*(R*X+Y*Y)/SL
F1=XL/SL & E2=YL/SL
AD=A1-A4 & RD=R1-R4 & CD=C1-C4
AP=A3+A2 & RP=R3+R2 & CP=C3+C2
Q=4.*Q1-2*Q2
IF(0)630:610:605
605 QRT=SQR(T(0))
T1=2.*ATAN(SL*QRT/(2.*CON1+SL*CON2))/QRT
AP6=1.*Q2+SL*/SL2)/Q1
IF(ARG.LF.0.0)60 TO 650
T2=1.*ALOG(AP6)-1.*Q2*T1
T3=SL-02*2-T1
GO TO 620
610 T1=2.*SL/(02/(02/2.0+SL))
ARG=2.*02/02+SL)/O2
IF(ARG.LF.0.0)60 TO 650
T2=1.*ALOG(ARG)-1.*Q2*T1
T3=SL-02*2-02/2.0*02/2.0+T1
620 VSTRIP=(F1*(AP+T1+RP*T2+CP*T3)+E2*(AP*T1+DP*T2+CP*T3))
USTRIP=(F1*(AP+T1+RP*T2+CP*T3)-F2*(AD+T1+DP*T2+CP*T3))
RETURN;
630 PRINT 640,0
640 FORMAT('/**ERROR*-THE VALUE OF Q IS NEGATIVE, Q=-F15.R**/)
STOP
650 PRINT 651,I,K,L,ARG
651 FORMAT('/** I=I2, J=I2, J+4=I2, J+1=I2, ARG=A,F25.15,**/)
STOP
END
THIS SUBROUTINE EVALUATES THE VELOCITY COMPONENT ALONG THE SINGULAR STRIP.

SUBROUTINE SING(I,J,USING,VSING)
COMMON/JACK/NJ,NJT,NJB,XJ(100),YJ(100),UJ(100),VJ(100),
$UJN(100),VJN(100),DYJ(100)
COMMON/CONTOUR/ND,NTW,NDS,NBW,NUS,XD(100),YD(100),UD(100),VD(100)
PC=SQRT((XJ(I)-XD(J))*2+(YJ(J)-YD(J))*2)
XJX=XJ(I)-XD(J)
YJY=YJ(I)*YD(J)-YD(J)/(XD(J)+1)-XD(J))
PC=SQRT(YJY*YJY+XJX*YJX)
SL=SQRT((XD(J)+1)-XD(J))*2+(YD(J)+1)-YD(J))*2)
ARG=SL/PC-1.0
IF(ARG.LE.0.)GO TO 750
T1=UD(J)+1-UD(J)
T2=UD(J)+T1*PC/SL
T3=ALOG(ARG)
T4=VD(J)+1-VD(J)
T5=VD(J)+T4*PC/SL
VSING=T1*T2*T3
USING=T4*T5*T3
RETURN
750 PRINT 751,I,J,ARG
751 FORMAT(* I=*J=* ARG=* ARG=* F25.15,/* ERROR---
$ARGUMENT OF LOG IN SING LE 0*)
STOP
END

THIS FUNCTION DETERMINES WHETHER THE STRIP BEING CONSIDERED IS A NORMAL OR SINGULAR STRIP.

FUNCTION ISTRIP(IJ,ID)
COMMON/JACK/NJ,NJT,NJB,XJ(100),YJ(100),UJ(100),VJ(100),
$UJN(100),VJN(100),DYJ(100)
COMMON/CONTOUR/ND,NTW,NDS,NBW,NUS,XD(100),YD(100),UD(100),VD(100)
C****ISTRIP=1---NONSINGULAR STRIP
C****ISTRIP=2---SINGULAR STRIP
C
ISTRIP=1
SJ=XJ(IJ)
TJ=YJ(IJ)
S1=XD(ID)
T1=VD(ID)
S2=XD(ID+1)
T2=VD(ID+1)
DIF=S?-S1
PRODUCT = T1 * T2 * T3
IF (DIF) IO, 30, 0
  IF ((SJ > T2 AND SJ < T1) AND (PRODUCT > 0)) ISTRIP = 2
  RETURN
20 IF ((SJ > T1 AND SJ < T2) AND (PRODUCT > 0)) ISTRIP = 2
30 CONTINUE
RETURN
END

THIS FUNCTION DETERMINES WHETHER A POINT IS A POINT ON THE DATA SURFACE OR A FIELD POINT
FUNCTION IBDRY(IJ)
COMMON/JACK/NJ, NJT, NJB, XJ(100), YJ(100), UJ(100), VJ(100),
%UJN(100), VJN(100), YJN(100), DYZ(I)
COMMON/CONTOUR/ND, NTW, NDS, NNB, NUS, XD(100), YD(100), UD(100), VD(100)

C *** IBDRY = 1 --- (XJ, YJ) ON DATA SURFACE
C *** IBDRY = 2 --- (XJ, YJ) IS FIELD POINT

IBDRY = 1
SJ = X(IJ)
IF((SJ < XD(1)) OR (SJ > XD(NTW))) IBDRY = 2
RETURN
END

THIS SUBROUTINE INTEGRATES THE DELTA THETA VALUES TO OBTAIN NEW WALL LOCATION PREDICTIONS
SUBROUTINE DELYJ(NTHETA, THETA, NYPP, YPP, UINF)
DIMENSION THETA(NTHETA), YPP(NYPP), JOPT(2)
COMMON/JACK/NJ, NJT, NJB, XJ(100), YJ(100), UJ(100), VJ(100),
%UJN(100), VJN(100), YJN(100), DYZ(I)
JOPT(1) = 1
JOPT(2) = 1
EC1 = 0.0
ECHN = 0.0
DO 10 I = 1, NJ
UJN(I) = UJN(I) + UINF
UJ(I) = UJ(I) + UINF
THETA(I) = VJN(I) / UJN(I)
YJN(I) = VJ(I) / UJ(I)
10 CONTINUE
CALL HMCURV1(XJ, THETA, YPP, NJT, 0.0, JOPT, EC1, ECHN, IERR)
DO 15 T = 1, NJT
DO J=1,NJT

15 CONTINUE
CALL HMCHURVI(XJ,YJ,YPP,N,JTP,EC1,ECN,IERR)
DO 20 I=1,NJT
DO J=1,NJT

20 CONTINUE

DO 25 I=1,NJT

25 CONTINUE
CALL HMCHURVI(XJ,YJ,YPP,N,JTP,EC1,ECN,IERR)
DO 30 I=1,NJT

30 CONTINUE
WRITE(6,98)
DO 40 I=1,NJT
WRITE(6,99)I,J,XJ(I),YJ(I),UJ(I),VJ(I),THETA(I)
40 CONTINUE

RMSDYT=RMS(JT,NJT,DUJ)
RMSDYR=RMS(JT,NJT,DUJ(NJT+1))
WRITE(6,97)RMSDYT,RMSDYR
RETURN
97 FORMAT(3E16.6)
98 FORMAT(11/15$2X$15E16.6) /* PREDICTED WALL CORRECTION (SCALED)*//
99 FORMAT(13+2(2X,F4.4)2(2X,F7.2)5(2X,F10.4))
END

THIS FUNCTION STATEMENTS DETERMINES THE RMS VALUE OF ARRAY A

FUNCTION RMS(N,A)
DIMENSION A(N)
RMS=0.0
DO 10 I=1,N
RMS=RMS+A(I)*A(I)
10 CONTINUE
RMS=SQRT(RMS/N)
RETURN
END

SUBROUTINE HMCHURVI(X,Y,YPP,N,JTP,EC1,ECN,IERR)

THIS SUBROUTINE FITS A CURVE OR TENSION SPINE TO THROUGH A SET
OF Y VS X VALUES.

AUTHOR -- HARRY MORGAN NASA/LANGLEY NAAH/STAN
(MODIFIED BY JOEL EVERHART NASA/LANGLEY)

************************************************************************

DESCRIPTION OF INPUT AND OUTPUT FOR SUBROUTINE SPLINE

PARAMETER DESCRIPTION

Y - INPUT ARRAY OF INDEPENDENT VARIABLE.
YPP - INPUT ARRAY OF DEPENDENT VARIABLE.
N - NUMBER OF INPUT VALUES OF X AND Y.
ST - SPLINE TENSION (ST=0.0 SAME AS CUBIC SPLINE).
OPT - END CONDITION OPTION ARRAY OF LENGTH 2. OPT(1) CORRESPONDS TO CONDITION AT X(1), OPT(2) CORRESPONDS TO CONDITION AT X(N).
1 = FC1 OR FCN UNKNOWN
2 = FC1=YP(1) OR FCN=YP(N)
3 = FC1=YPP(1) OR FCN=YPP(N)
FCN - END CONDITION AT END OF Y VS X CURVE.
IFRR - ERROR CODE
0 - NO ERROR
1 - N IS LESS THAN 3
2 - X IS NOT MONOTONICALLY INCREASING

RESTRICTIONS
X MUST BE MONOTONICALLY INCREASING
N MUST BE GREATER THAN OR EQUAL TO 3.
X AND Y MUST BE DIMENSIONED BY AT LEAST N IN CALLING PROGRAM.
ST SHOULD BE NO GREATER THAN 50.

************************************************************************

DIMENSION X(N), Y(N), YPP(1), OPT(2)

INTEGER OPT

SINH(X)=0.5*(EXP(X)-EXP(-X))
COSH(X)=0.5*(EXP(X)+EXP(-X))

CHECK ON MONOTONICALLY INCREASING X

IFPR=0
IF (N,1,3) GO TO 32
DO 1 I=2,N
IF (X(I).LE.X(I-1)) GO TO 33
1 CONTINUE

C C C
C C C
C C C
COMP TENSION PARAMETER T
C C C
T=ST*FLOAT(N-1)/(X(N)-X(1))
C C C
C C C
C C C
C C C
C C C
COMP NEEDED PROGRAM INTEGER VALUES
C C C
M=N-1
N1=N+1
N2=N1+1
N3=N2+1
N4=N3+1
N5=N4+1
N6=N5+1
N7=N6+1
N8=N7+1
N9=N8+1

C C C
C C C
C C C
LOAD MATRIX ROWS 2 THRU N-1
C C C
DO 2 I=1,N
H2=X(I+1)-X(I)
IF (ST*LE.0.0) YPP(I)=H2/H1.
IF (ST*GT.0.0) YPP(I)=(1./H2-T/SINH(T*H2))/(T*T)
K=2*(I-1)+I
YPP(K)=YPP(I)
IF (I.EQ.1) GO TO 2
K=K+N
IF (ST*LE.0.0) YPP(K)=(H2+H1)/3.
IF (ST*GT.0.0) YPP(K)=(T*COSH(T*H1)/SINH(T*H1)-1./H1+T*COSH(T*H2)/S1
*SINH(T*H2)-1./H2)/(T*T)
K=K+N
YPP(K)=(Y(I+1)-Y(I))/H2-(Y(I)-Y(I-1))/H1
2 H1=H2

C C C
C C C
LOAD MATRIX ROWS 1 AND N (END CONDITIONS)
C C C
YPP(N3)=YPP(N4-1)=1.0
YPP(N2)=YPP(N)=0.0
C C C
ROW 1
C C C
GO TO (3,4,6)OPT(1)
3 A1=Y(2)-Y(3)
A2=X(3)-X(1)
A3=X(1)-X(2)
YPP(N4)=2.*((Y(1)*A1+Y(2)*A2+Y(3)*A3)/(A1*X(1)**2+A2*X(2)**2+A3*X(3)**2)
GO TO 7
4 H1=X(2)-X(1)
   YPP(N4)=EC1-(Y(2)-Y(1))/H1
   IF (ST.GT.0.0) GO TO 5
   YPP(N3)=-H1/3.
   YPP(N2)=-H1/6.
   GO TO 7
5 YPP(N3)=(1./H1*T)-COSH(T*H1)/SINH(T*H1))/T
   YPP(N2)=(1./SINH(T*H1)-1./H1*T))/T
   GO TO 7
6 YPP(N4)=FC1

ROW N
7 GO TO (8,9,11)OPT(2)
8 A1=X(N-1)-X(N)
   A2=X(N)-X(N-2)
   A3=X(N-2)-X(N-1)
   YPP(N5-1)=2.*Y(N-2)*A1+Y(N-1)*A2+Y(N)*A3)/(A1*X(N-2)**2+A2*X(N-1)**2)
   **2+A3*X(N)**2)
   GO TO 12
9 H1=X(N)-X(M)
   YPP(N5-1)=ECN-(Y(N)-Y(M))/H1
   IF (ST.GT.0.0) GO TO 10
   YPP(M)=H1/6.
   YPP(N4-1)=H1/3.
   GO TO 12
10 YPP(M)=(1./H1*T)-1./SINH(T*H1))/T
   YPP(N4-1)=(COSH(H1*T)/SINH(T*H1)-1./H1*T))/T
   GO TO 12
11 YPP(N5-1)=ECN

SOLVE TRIDIAGONAL MATRIX
12 YPP(N7)=YPP(N3)
   YPP(N8)=YPP(N4)/YPP(N3)
   YPP(N9)=YPP(N2)/YPP(N3)
   DO 13 I=2,N
      I1=I-1
      I2=I-2
      YPP(N7+I1)=YPP(N3+I1)-YPP(I1)*YPP(N9+I2)
      IF (I.EQ.N) GO TO 13
      YPP(N9+I1)=YPP(N2+I1)/YPP(N7+I1)
   DO 13 I=1,N
      I1=N6-I-I
      I2=N-I
      I3=T2-1
14 YPP(II)=YPP(N8+I3)-YPP(N9+I3)*YPP(I1+1)
   K=0
   DO 15 I=1,N
   I1=I-1
   YPP(N6+K)=YPP(N5+I1)
15 K=K+1
   YPP(M)=YPP(N3-2)
C
C REARRANGE SOLUTION IN YPP ARRAY
C
   DO 16 I=1,N
   I1=N6+I-1
16 YPP(I)=YPP(I1)
C
   RETURN
32 IERF=1
   RETURN
33 IERF=2
   WRITE(6,35) (I*X(I)*Y(I),I=1,N)
   RETURN
35 FORMAT(1H1,5X,35) (I*X(I)*Y(I),I=1,N)
   $X'lgX,1_Y
   \end
C
FUNCTION CURV2 (T,N,X,Y,YP,SIGMA)
C
INTEGER N
REAL T,X(N),Y(N),YP(N),SIGMA
C
FROM THE SPLINE UNDER TENSION PACKAGE
CODED BY A. K. CLINE AND R. J. RENKA
DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TEXAS AT AUSTIN
C
THIS FUNCTION INTEPOLATES A CURVE AT A GIVEN POINT
USING A SPLINE UNDER TENSION. THE SUBROUTINE CURV1 SHOULD
BE CALLED EARLIER TO DETERMINE CERTAIN NECESSARY
PARAMETERS.
C
ON INPUT--
C
T CONTAINS A REAL VALUE TO BE MAPPED ONTO THE INTERPOLATING CURVE.
C
N CONTAINS THE NUMBER OF POINTS WHICH WERE SPECIFIED TO
DETERMINE THE CURVE.
C
X AND Y ARE ARRAYS CONTAINING THE ABSCISSAE AND
ORDINATES, RESPECTIVELY, OF THE SPECIFIED POINTS.
YP IS AN ARRAY OF SECOND DERIVATIVE VALUES OF THE CURVE AT THE NODES.

AND

SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).


ON OUTPUT--

CURV2 CONTAINS THE INTERPOLATED VALUE.

NONE OF THE INPUT PARAMETERS ARE ALTERED.

THIS FUNCTION REFERENCES PACKAGE MODULES INTRVL AND SNHCSH.

Determine interval

IM1 = INTRVL(T,X,N)
I = IM1+1

Denormalize tension factor

SIGMAP = ABS(SIGMA)*FLOAT(N-I)/(X(N)-X(I))

Set up and perform interpolation

DEL1 = T-X(IM1)
DEL2 = X(I)-T
DELS = X(I)-X(IM1)
SUM = (Y(I)*DEL1+Y(IM1)*DEL2)/DELS
IF (SIGMAP .NE. 0.) GO TO 1
CURV2 = SUM-DEL1*DEL2*(YP(I)*((DEL1+DELS)*YP(IM1)+
* (DELS*DEL2)))/(6.*DELS)
RETURN
1 DELP1 = SIGMAP*(DEL1+DELS)/2.
DELP2 = SIGMAP*(DEL2+DELS)/2.
CALL SNHCSH (SINHM1,DUMMY,SIGMAP*DEL1,-1)
CALL SNHCSH (SINHM2,DUMMY,SIGMAP*DEL2,-1)
CALL SNHCSH (SINHMS,DUMMY,SIGMAP*DELS,-1)
CALL SNHCSH (SINHPI,DUMMY,SIGMAP*DEL1/2.,-1)
CALL SNHCSH (SINHP2,DUMMY,SIGMAP*DEL2/2.,-1)
CALL SNHCSH (DUMMY,COSHPI,DELP1,1)
CALL SNHCSH (DUMMY, COSHP2, DELP2, 1)
CURV2 = SUM*(YP(I)*((SINHM1*DELP2-DELI*(2.*(COSHPI+1.))*(SINHP2+SIGMAP*COSHP1*DELP2))*YP(I)*((SINHM2*DELP2*(COSHP2+1.))*SINHP1+SIGMAP)*COSH2*(COSHP2*DELI)))/(SIGMAP*SIGMAP*DELS*(SINHMS*SIGMAP*DELS))
RETURN
END

FUNCTION CURVD (T, N, X, Y, YP, SIGMA)

INTEGER N
REAL T(N), X(N), Y(N), YP(N), SIGMA

FROM THE SPLINE UNDER TENSION PACKAGE
CODED BY A. K. CLINE AND R. J. RENKA
DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TEXAS AT AUSTIN

THIS FUNCTION DIFFERENTIATES A CURVE AT A GIVEN POINT
USING A SPLINE UNDER TENSION. THE SUBROUTINE CURV1 SHOULD
BE CALLED EARLIER TO DETERMINE CERTAIN NECESSARY
PARAMETERS.

ON INPUT--

T CONTAINS A REAL VALUE AT WHICH THE DERIVATIVE IS TO BE
DETERMINED.

N CONTAINS THE NUMBER OF POINTS WHICH WERE SPECIFIED TO
DETERMINE THE CURVE.

X AND Y ARE ARRAYS CONTAINING THE ABSCISSAE AND
ORDINATES, RESPECTIVELY, OF THE SPECIFIED POINTS.

YP IS AN ARRAY OF SECOND DERIVATIVE VALUES OF THE CURVE
AT THE NODES.

AND

SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).

THE PARAMETERS N, X, Y, YP, AND SIGMA SHOULD BE INPUT
UNALTERED FROM THE OUTPUT OF CURV1.

ON OUTPUT--

CURVD CONTAINS THE DERIVATIVE VALUE.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS FUNCTION REFERENCES PACKAGE MODULES INTRVL AND
C SNHCSH.
C
C DETERMINE INTERVAL
C
   IM1 = INTRVL(T*X,N)
   I = IM1+1
C
C DENORMALIZE TENSION FACTOR
C
   SIGMAP = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
C
C SET UP AND PERFORM DIFFERENTIATION
C
   DEL1 = T-X(IM1)
   DEL2 = X(I)-T
   DELS = X(I)-X(IM1)
   SUM = (Y(I)-Y(IM1))/DELS
   IF (SIGMAP .NE. 0.) GO TO 1
   CURVD = SUM*(YP(I)*(2.*DELI*DELI-DEL2*(DELI+DELS))-
           * YP(IM1)*(2.*DEL2*DEL2-DEL1*(DEL2+DELS)))+
           * /(6.*DELS)
   RETURN
1   CALL SNHCSH (DUMMY,COSH1,_SIGMAP*DELI,1)
   CALL SNHCSH (DUMMY,COSH2,_SIGMAP*DEL2,1)
   CALL SNHCSH (SINHMS,DUMMY,_SIGMAP*DELS,-1)
   CURVD = SUM*(YP(I)*(DELS*SIGMAP*COSH1-SINHMS)-
           * YP(IM1)*(DELS*SIGMAP*COSH2-SINHMS))/
           * (SIGMAP*SIGMAP*DELS*(SINHMS*SIGMAP*DELS))
   RETURN
END
C
FUNCTION: CURVI (XL,XU,N,X,Y,YP,SIGMA)
C
INTEGER N
REAL XL,XU,X(N),Y(N),YP(N),SIGMA
C
FROM THE SPLINE UNDER TENSION PACKAGE
CODED BY A. K. CLINE AND R. J. RENKA
DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TEXAS AT AUSTIN

C THIS FUNCTION INTEGRATES A CURVE SPECIFIED BY A SPLINE
UNDER TENSION BETWEEN TWO GIVEN LIMITS, THE SUBROUTINE CURVI SHOULD BE CALLED EARLIER TO DETERMINE NECESSARY PARAMETERS.

ON INPUT--

XL AND XU CONTAIN THE UPPER AND LOWER LIMITS OF INTEGRATION, RESPECTIVELY. (SL NEED NOT BE LESS THAN OR EQUAL TO XU, CURVI(XL,XU...) *EQ* -CURVI(XU,XL...).

N CONTAINS THE NUMBER OF POINTS WHICH WERE SPECIFIED TO DETERMINE THE CURVE.

X AND Y ARE ARRAYS CONTAINING THE ABSCISSAE AND ORDINATES, RESPECTIVELY, OF THE SPECIFIED POINTS.

YP IS AN ARRAY FROM SUBROUTINE CURVI CONTAINING AT THE NODES.

AND

SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).

THE PARAMETERS N, X, Y, YP, AND SIGMA SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF CURVI.

ON OUTPUT--

CURVI CONTAINS THE INTEGRAL VALUE.

NONE OF THE INPUT PARAMETERS ARE ALTERED.

THIS FUNCTION REFERENCES PACKAGE MODULES INTRVL AND SNHCSS.

STATEMENT FUNCTION FOR COEFFICIENT ASSOCIATED WITH DERIVATIVE TERMS

\[
\text{TERM} (\text{CMM1}, \text{CMM2}, T) = \frac{(\text{CMM1} - \text{CMM2} - \text{SIGMAP} \cdot T \cdot SS)}{\text{SIGMAP} \cdot \text{SIGMAP} \cdot (SS \cdot \text{SIGMAP} \cdot \text{DELS})}
\]

DEnormalize TENSION FACTOR

\[
\text{SIGMAP} = \text{ABS} (\text{SIGMA}) \cdot \text{FLOAT} (N-1)/ (X(N)-X(1))
\]

DETERMINE ACTUAL UPPER AND LOWER BOUNDS

\[
\text{XXL} = XL
\]

\[
\text{XXII} = XU
\]
SSIGN = 1.
IF (XL .LT. XU) GO TO 1
XXL = XU
XXU = XL
SSIGN = -1.
IF (XL .GT. XU) GO TO 1
RETURN
ZERO IF XL .EQ. XU

SEAPCH FOR PROPER INTERVALS

1 ILM1 = INTRVL (XXL,X,X)
   IL = ILM1+1
   IU = IUM1+1
   IF (IL .EQ. IU) GO TO 8

INTEGRATE FROM XXL TO X(IL)

SUM = 0.
IF (XXL .EQ. X(IL)) GO TO 3
DEL1 = XXL-X(ILM1)
DEL2 = X(IL)-XXL
DELS = X(IL)-X(ILM1)
T1 = (DEL1+DELS)*DEL2/(2.*DELS)
T2 = DEL2*DEL2/(2.*DELS)
SUM = T1*Y(IL)+T2*Y(ILM1)
IF (SIGMA .EQ. 0.) GO TO 2
CALL SNHCSH (DUMMY,C1*SIGMAP*DEL1*2)
CALL SNHCSH (DUMMY,C2*SIGMAP*DEL2*2)
CALL SNHCSH (SS*CS*SIGMAP*DELS*3)
SUM = SUM+TERM(CS,C1,T1)*YP(IL)
   *TERM(C2,0.,T2)*YP(ILM1)
GO TO 3
2 SUM = T1*T1*DELS*YP(IL)/6.
   +T2*(DEL1*(DEL2+DELS)+DELS*DELS)*YP(ILM1)/12.

INTEGRATE OVER INTERIOR INTERVALS

3 IF (IU-IL .EQ. 1) GO TO 6
   ILPI = IL+1
   DO 5 I = ILPI,IUM1
      DELS = X(I)-X(I-1)
      SUM = SUM+(Y(I)+Y(I-1))*DELS/2.
      IF (SIGMA .EQ. 0.) GO TO 4
      CALL SNHCSH (SS*CS*SIGMAP*DELS*3)
SUM = SUM + (YP(I)+YP(I-1))*(CS*SS*SIGMAP*DELS*DELS/2) *
      *(SIGMAP*SIGMAP*SIGMAP*(SS*SIGMAP*DELS))
GO TO 5
4  SUM = SUM - (YP(I)+YP(I-1))*DELS*DELS*DELS/24.
5  CONTINUE

C INTEGRATE FROM X(IU-1) TO XXU

6 IF (XXU.EQ. X(IUM1)) GO TO 10
  DEL1 = XXU - X(IUM1)
  DEL2 = X(IU) - XXU
  DELS = X(IU) - X(IUM1)
  TI = DEL1*DELI/2*DELS
  T2 = DEL2*DELI/2*DELS
  SUM = SUM + Y(IU)*T1*Y(IUM1)
  IF (SIGMA.EQ. 0.) GO TO 7
  CALL SNHCISH (DUMMY*CU1.SIGMAP*DELU1*2)
  CALL SNHCISH (DUMMY*CU2.SIGMAP*DELU2*2)
  CALL SNHCISH (DUMMY*CL1.SIGMAP*DELL1*2)
  CALL SNHCISH (DUMMY*CL2.SIGMAP*DELL2*2)
  CALL SNHCISH (SS*DUUMY.SIGMAP*DELS*1)
  SUM = SUM + T1*YP(IU)*T2*YP(IUM1)
    + TERM(CU1,CL1,T1)*YP(IU)
    + TERM(CL2,CL2,T2)*YP(IUM1)
GO TO 10
7 SUM = SUM - YP(IU)*DP(DEL2*(DELS+DELU1)*DELS*DELS)*DEL1*DELS*DELS/12.
     - T2*YP(IUM1)/12.
GO TO 10

C INTEGRATE FROM XXL TO XXU

8 DELU1 = XXU - X(IUM1)
  DELU2 = X(IU) - XXU
  DELL1 = XXL - X(IUM1)
  DELL2 = X(IU) - XXL
  DELS = X(IU) - X(IUM1)
  DEL1 = XXU - XXL
  TI = DELU1*DELL1*DELI/2*DELS
  T2 = DELU2*DELL2*DELI/2*DELS
  SUM = T1*YP(IU)*T2*YP(IUM1)
  IF (SIGMA.EQ. 0.) GO TO 9
  CALL SNHCISH (DUMMY*CU1.SIGMAP*DELU1*2)
  CALL SNHCISH (DUMMY*CU2.SIGMAP*DELU2*2)
  CALL SNHCISH (DUMMY*CL1.SIGMAP*DELL1*2)
  CALL SNHCISH (DUMMY*CL2.SIGMAP*DELL2*2)
  CALL SNHCISH (SS*DUUMY.SIGMAP*DELS*1)
  SUM = SUM + T1*YP(IU)*T2*YP(IUM1)
    + TERM(CU1,CL1,T1)*YP(IU)
    + TERM(CL2,CL2,T2)*YP(IUM1)
GO TO 10
9 SUM = SUM - T1*YP(IU)*DELU2*(DELS+DELU1)*DELL2*(DELS+DELL1)*YP(IU)/12.
     - T2*YP(IUM1)/12.
     - T2*(DELL1*DELS+DELL2)*DELU1*(DELS+DELU2)*YP(IUM1)/12.
C CORRECT SIGN AND RETURN
C
10 CURVI = SIGN*SUM
RETURN
END
C
FUNCTION INTRVL (T,X,N)
C
INTEGER N
REAL T,X(N)
C
FROM THE SPLINE UNDER TENSION PACKAGE
CODED BY A. K. CLINE AND R. J. RENKA
DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TEXAS AT AUSTIN
C
THIS FUNCTION DETERMINES THE INDEX OF THE INTERVAL
(DETERMINED BY A GIVEN INCREASING SEQUENCE) IN WHICH
A GIVEN VALUE LIES.
C
ON INPUT--

T IS THE GIVEN VALUE.

X IS A VECTOR OF STRICTLY INCREASING VALUES.

AND

N IS THE LENGTH OF X (N .GE. 2).

ON OUTPUT--

INTRVL RETURNS AN INTEGER I SUCH THAT

I = 1 IF T .LE. X(2)
I = N-1 IF X(N-1) .LE. T
OTHERWISE X(I) .LE. T .LE. X(I+1)

NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
-----------------------------------------------------------------
C
TT = T
IF (TT .LE. X(2)) GO TO 4
IF (TT .GE. X(N-1)) GO TO 5
IL = 2
TH = N-1
LINEAR INTERPOLATION

\begin{align*}
1 \quad & I = IL + IFIX(\text{FLOAT}(IH-IL) \ast (TT-X(IL))/(X(IH)-X(IL))) \\
& \text{IF (TT < X(I)) GO TO 2} \\
& \text{IF (TT \leq X(I+1)) GO TO 3}
\end{align*}

C TOO HIGH

\begin{align*}
& IL = I+1 \\
& \text{GO TO 1}
\end{align*}

C TOO LOW

\begin{align*}
2 \quad & IH = I \\
& \text{GO TO 1} \\
3 \quad & \text{INTRVL} = 1 \\
& \text{RETURN}
\end{align*}

C LEFT END

\begin{align*}
4 \quad & \text{INTRVL} = 1 \\
& \text{RETURN}
\end{align*}

C RIGHT END

\begin{align*}
5 \quad & \text{INTRVL} = N-1 \\
& \text{RETURN}
\end{align*}

SURROUTINE SNHCSH (SINHM, COSHM, X, ISW)

INTEGER ISW
REAL SINHM, COSHM, X

FROM THE SPLINE UNDER TENSION PACKAGE
CODED BY A. K. CLINE AND R. J. RENKA
DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TEXAS AT AUSTIN

THIS SUBROUTINE RETURNS APPROXIMATIONS TO
\begin{align*}
\text{SINHM}(X) &= \text{SIN}(X)-X \\
\text{COSHM}(X) &= \text{COSH}(X)-1 \\
\text{COSHMM}(X) &= \text{COSH}(X)-1-x^2/2 \\
\end{align*}

WITH RELATIVE ERROR LESS THAN 3.42E-14

ON INPUT--
X CONTAINS THE VALUE OF THE INDEPENDENT VARIABLE.

ISW INDICATES THE FUNCTION DESIRED
-1 IF ONLY SINHM IS DESIRED,
0 IF BOTH SINHM AND COSH ARE DESIRED,
1 IF ONLY COSH IS DESIRED,
2 IF ONLY COSHMM IS DESIRED,
3 IF BOTH SINHM AND COSHMM ARE DESIRED.

ON OUTPUT--

SINHM CONTAINS THE VALUE OF SINHM(X) IF ISW .LE. 0 OR ISW .EQ. 3 (SINHM IS UNALTERED IF ISW .EQ. 1 OR ISW .EQ. 2).

COSH CONTAINS THE VALUE OF COSHMM(X) IF ISW .EQ. 0 OR ISW .EQ. 1 AND CONTAINS THE VALUE OF COSHMM(X) IF ISW .GE. 2 (COSHMM IS UNALTERED IF ISW .EQ. -1).

AND
X AND ISW ARE UNALTERED.

-----------------------------------

DATA SP4/4.50217693381333E-08/,
* SP3/8.95278544216390E-06/,
* SP2/8.72048976791502E-04/,
* SP1/4.36314556981690E-02/,
* SQ1/-6.36654430175110E-03/,
DATA CP4/1.78419567490190E-07/,
* CP3/2.87277229799044E-05/,
* CP2/2.15151199203E-03/,
* CP1/7.581822756256E-02/,
* CP0/-7.5151505679867E-03/,
DATA ZP3/-5.92927116264720E-07/,
* ZP2/1.77943488030894E-04/,
* ZP1/1.69800461894792E-02/,
* ZP0/-1.33412535492375E-09/,
* ZP3/1.80858944138663E-07/,
* ZP2/1.27814964403863E-04/,
* ZP1/-1.63532871439181E-02/,

X = X
AX = ARS(XX)
XS = XX*XX.
IF ((AX .GE. 2.70) .OR. (AX .GE. 1.15 .AND. ISW .NE. 2)) EXPX = EXP(AX)

C
C SINHM APPROXIMATION
THE SUBROUTINE AVERAGES THE TOP AND BOTTOM WALL VALUES OF \( \text{DYJ} \)

```fortran
SUBROUTINE AVGDYJ(NJ, XJ, DYJ)
  DIMENSION XJ(NJ), DYJ(NJ)
  N = NJ/2
  IF((2*N).NE.NJ) GO TO 20
  DO 10 I = 1, N
    SIGN = 1.0
    IF(DYJ(I).LT.0.0) SIGN = -1.0
    DYJ(I) = (ABS(DYJ(I)) + ABS(DYJ(I+N)))/2.0
    DYJ(I+1) = DYJ(I) * SIGN
    DYJ(I+N) = -DYJ(I)
  CONTINUE
  WRITE(6,99) I, XJ(I), DYJ(I), XJ(I+N), DYJ(I+N)
  CONTINUE
  RETURN
END
```

43
SUBROUTINE NEWYJ(IRLAX, NRDY, RDY)
 REAL *MSET
 INTEGER TEST, RUN, POINT
 COMMON /JACK/NJ, NJT, NJR, XJ(100), YJ(100), UJ(100), VJ(100),
 $UJN(100), VJN(100), YJN(100), DYJ(100)
 COMMON/PARAM/HEAD(R), TEST, RUN, POINT, M, RN, ALPHA, TEMP, PLAX(10),
 $NLAX, PT, PINF, UINF, AVG, IS(4), OPT(4), EC(4), GAMMA, MSET
 DIMENSION RDY(NRDY)
 DO 10 I=1, NJ
 RDY(I) = PLAX(IRLAX) * DYJ(I)
 YJN(I) = YJ(I) * RDY(I)
 10 CONTINUE
 RMSDYTC = RMS(NJT, RDY(1))
 RMSDYBC = RMS(NJT, RDY(NJT+1))
 WRITE(6, 983) PLAX(IRLAX)
 WRITE(6, 984) (HEAD(I) + I = 1, 8)
 WRITE(6, 985) TEST, RUN, POINT
 WRITE(6, 986) MSET, RN, ALPHA, TEMP, PT, PINF, UINF
 WRITE(6, 987)
 C WRITE(7, 991) TEST, RUN, POINT, PLAX(IRLAX), MSET
 DO 915 I = 1, NJ
 WRITE(6, 988) I, XJ(I), YJ(I), YJN(I), RDY(I)
 915 CONTINUE
 WRITE(6, 989) RMSDYTC, RMSDYBC
 RETURN
 983 FORMAT(11/"** UNSCALED WALL CORRECTIONS FOR A RELAXATION**
 ** FACTOR=*/F10.5)
 984 FORMAT(//3X8A10)
 985 FORMAT(3X, "USING RESULTS FROM TEST", I4, " RUN", I4, " POINT", I4)
 986 FORMAT(//3X, "TUNNEL PARAMETERS ARE", 10X, M, F8.4, 4X, RN, G9.4,
 $4X, ALPHA, F8.3, 3X, TEMP (DEG R) =", F8.3, 10X, PT, G9.4, 3X,
 $PINF, G9.4, 1X, UINF =", F11.5)
 987 FORMAT(//4X, I6, XJ, 11X, YJ, 11X, YJN, 11X, DYJ)
 988 FORMAT(2X13.4, 2X612.6)
 989 FORMAT(33XRMS(DYJ(TOP)) =, G12.6/
 33XRMS(DYJ(ROT)) =, G12.6)
990 FORMAT(4F10.5)
991 FORMAT(* TEST=*I3* RUN=*I3* POINT=*I3* RLAX=*F6.4* MSET=*F10.5)
END
APPENDIX C

SAMPLE CASE
### INPUT DATA LISTING

#### 6x19 INCH TRANSONIC TUNNEL FLEXIBLE WALL TEST

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Velocity (ft/s)</th>
<th>Acceleration (ft/s²)</th>
<th>Distance (ft)</th>
<th>Wall Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0728</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>12</td>
<td>19</td>
<td>1</td>
<td>19</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0.2500</td>
<td>1.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>-29.0000</td>
<td>9.500000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-29.0000</td>
</tr>
<tr>
<td>-24.0000</td>
<td>9.49765</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-24.0000</td>
</tr>
<tr>
<td>-20.0000</td>
<td>9.49739</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-20.0000</td>
</tr>
<tr>
<td>-16.0000</td>
<td>9.50130</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-16.0000</td>
</tr>
<tr>
<td>-12.0000</td>
<td>9.50918</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-12.0000</td>
</tr>
<tr>
<td>-8.0000</td>
<td>9.51911</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-8.0000</td>
</tr>
<tr>
<td>-4.0000</td>
<td>9.55012</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-4.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>9.57337</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.0000</td>
</tr>
<tr>
<td>4.0000</td>
<td>9.54136</td>
<td>0.000000</td>
<td>0.000000</td>
<td>4.0000</td>
</tr>
<tr>
<td>8.0000</td>
<td>9.50557</td>
<td>0.000000</td>
<td>0.000000</td>
<td>8.0000</td>
</tr>
<tr>
<td>12.0000</td>
<td>9.49375</td>
<td>0.000000</td>
<td>0.000000</td>
<td>12.0000</td>
</tr>
<tr>
<td>16.0000</td>
<td>9.49179</td>
<td>0.000000</td>
<td>0.000000</td>
<td>16.0000</td>
</tr>
<tr>
<td>-29.0000</td>
<td>-9.50000</td>
<td>0.000000</td>
<td>-9.50000</td>
<td>-29.0000</td>
</tr>
<tr>
<td>-24.0000</td>
<td>-9.49765</td>
<td>0.000000</td>
<td>-9.49765</td>
<td>-24.0000</td>
</tr>
<tr>
<td>-20.0000</td>
<td>-9.49739</td>
<td>0.000000</td>
<td>-9.49739</td>
<td>-20.0000</td>
</tr>
<tr>
<td>-16.0000</td>
<td>-9.50130</td>
<td>0.000000</td>
<td>-9.50130</td>
<td>-16.0000</td>
</tr>
<tr>
<td>-12.0000</td>
<td>-9.50918</td>
<td>0.000000</td>
<td>-9.50918</td>
<td>-12.0000</td>
</tr>
<tr>
<td>-8.0000</td>
<td>-9.51911</td>
<td>0.000000</td>
<td>-9.51911</td>
<td>-8.0000</td>
</tr>
<tr>
<td>-4.0000</td>
<td>-9.55012</td>
<td>0.000000</td>
<td>-9.55012</td>
<td>-4.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-9.57337</td>
<td>0.000000</td>
<td>-9.57337</td>
<td>0.0000</td>
</tr>
<tr>
<td>4.0000</td>
<td>-9.54136</td>
<td>0.000000</td>
<td>-9.54136</td>
<td>4.0000</td>
</tr>
<tr>
<td>8.0000</td>
<td>-9.50557</td>
<td>0.000000</td>
<td>-9.50557</td>
<td>8.0000</td>
</tr>
<tr>
<td>12.0000</td>
<td>-9.49375</td>
<td>0.000000</td>
<td>-9.49375</td>
<td>12.0000</td>
</tr>
<tr>
<td>16.0000</td>
<td>-9.49179</td>
<td>0.000000</td>
<td>-9.49179</td>
<td>16.0000</td>
</tr>
<tr>
<td>-26.0000</td>
<td>9.50000</td>
<td>-0.0900</td>
<td>9.50000</td>
<td>-26.0000</td>
</tr>
<tr>
<td>-18.0000</td>
<td>9.50000</td>
<td>-0.0340</td>
<td>9.50000</td>
<td>-18.0000</td>
</tr>
<tr>
<td>-14.0000</td>
<td>9.50000</td>
<td>-0.0300</td>
<td>9.50000</td>
<td>-14.0000</td>
</tr>
<tr>
<td>-10.0000</td>
<td>9.50000</td>
<td>-0.0830</td>
<td>9.50000</td>
<td>-10.0000</td>
</tr>
<tr>
<td>-7.0000</td>
<td>9.50000</td>
<td>-0.0440</td>
<td>9.50000</td>
<td>-7.0000</td>
</tr>
<tr>
<td>-6.0000</td>
<td>9.50000</td>
<td>0.0110</td>
<td>9.50000</td>
<td>-6.0000</td>
</tr>
<tr>
<td>-5.0000</td>
<td>9.50000</td>
<td>0.0020</td>
<td>9.50000</td>
<td>-5.0000</td>
</tr>
<tr>
<td>-3.0000</td>
<td>9.50000</td>
<td>-0.3590</td>
<td>9.50000</td>
<td>-3.0000</td>
</tr>
<tr>
<td>-2.0000</td>
<td>9.50000</td>
<td>-0.6440</td>
<td>9.50000</td>
<td>-2.0000</td>
</tr>
<tr>
<td>-1.0000</td>
<td>9.50000</td>
<td>-0.9820</td>
<td>9.50000</td>
<td>-1.0000</td>
</tr>
<tr>
<td>1.0000</td>
<td>9.50000</td>
<td>-0.6770</td>
<td>9.50000</td>
<td>1.0000</td>
</tr>
<tr>
<td>2.0000</td>
<td>9.50000</td>
<td>-0.0520</td>
<td>9.50000</td>
<td>2.0000</td>
</tr>
<tr>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>3.0000</td>
<td>9.5000</td>
<td>-.02040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0000</td>
<td>9.5000</td>
<td>.00300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.0000</td>
<td>9.5000</td>
<td>.00480</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.0000</td>
<td>9.5000</td>
<td>.00780</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0000</td>
<td>9.5000</td>
<td>.00900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.0000</td>
<td>9.5000</td>
<td>.00610</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.0000</td>
<td>9.5000</td>
<td>.00470</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.0000</td>
<td>0.0000</td>
<td>.00470</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.0000</td>
<td>9.5000</td>
<td>.00470</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.0000</td>
<td>9.5000</td>
<td>.00610</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0000</td>
<td>9.5000</td>
<td>.00900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.0000</td>
<td>9.5000</td>
<td>.00780</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.0000</td>
<td>9.5000</td>
<td>.00480</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0000</td>
<td>9.5000</td>
<td>.00300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0000</td>
<td>9.5000</td>
<td>.02040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0000</td>
<td>9.5000</td>
<td>.05520</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0000</td>
<td>9.5000</td>
<td>.08770</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.0000</td>
<td>9.5000</td>
<td>.09820</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2.0000</td>
<td>9.5000</td>
<td>.06440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3.0000</td>
<td>9.5000</td>
<td>.03590</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-5.0000</td>
<td>9.5000</td>
<td>.00020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-6.0000</td>
<td>9.5000</td>
<td>.00110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-7.0000</td>
<td>9.5000</td>
<td>.00440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-10.0000</td>
<td>9.5000</td>
<td>.00830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-14.0000</td>
<td>9.5000</td>
<td>.00300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-18.0000</td>
<td>9.5000</td>
<td>.00340</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-26.0000</td>
<td>9.5000</td>
<td>.00900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-26.0000</td>
<td>0.0000</td>
<td>.00900</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UINF  = 805.542
<table>
<thead>
<tr>
<th>I</th>
<th>XJ</th>
<th>YJ</th>
<th>UJ</th>
<th>UJN</th>
<th>VJ</th>
<th>VJN</th>
<th>DVJ</th>
<th>DUJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-29.0000</td>
<td>6.0506</td>
<td>2.03</td>
<td>2.03</td>
<td>-2.802E-14</td>
<td>-2.802E-14</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>-24.0000</td>
<td>6.0491</td>
<td>1.18</td>
<td>-1.99</td>
<td>-2.428</td>
<td>-9.313E-01</td>
<td>0.1497</td>
<td>-3.171</td>
</tr>
<tr>
<td>3</td>
<td>-20.0000</td>
<td>6.0489</td>
<td>.74</td>
<td>-2.15</td>
<td>-1.943</td>
<td>7.926E-01</td>
<td>-1.151</td>
<td>-2.893</td>
</tr>
<tr>
<td>4</td>
<td>-16.0000</td>
<td>6.0514</td>
<td>.41</td>
<td>-2.21</td>
<td>-8.707</td>
<td>7.597</td>
<td>-1.111</td>
<td>-2.620</td>
</tr>
<tr>
<td>5</td>
<td>-12.0000</td>
<td>6.0564</td>
<td>.90</td>
<td>-1.96</td>
<td>8.623</td>
<td>1.722</td>
<td>0.8594</td>
<td>2.856</td>
</tr>
<tr>
<td>6</td>
<td>-8.0000</td>
<td>6.0628</td>
<td>1.31</td>
<td>-2.62</td>
<td>2.542</td>
<td>2.012</td>
<td>-5.299</td>
<td>-3.925</td>
</tr>
<tr>
<td>7</td>
<td>-4.0000</td>
<td>6.0825</td>
<td>2.17</td>
<td>1.04</td>
<td>4.754</td>
<td>8.084</td>
<td>3.329</td>
<td>-1.132</td>
</tr>
<tr>
<td>8</td>
<td>0.0000</td>
<td>6.0973</td>
<td>17.21</td>
<td>5.97</td>
<td>-6.386</td>
<td>-1.734</td>
<td>-1.095</td>
<td>-11.24</td>
</tr>
<tr>
<td>9</td>
<td>4.0000</td>
<td>6.0769</td>
<td>.02</td>
<td>.49</td>
<td>-5.611</td>
<td>-7.729</td>
<td>-2.118</td>
<td>.4667</td>
</tr>
<tr>
<td>10</td>
<td>8.0000</td>
<td>6.0541</td>
<td>1.03</td>
<td>-3.12</td>
<td>-3.022</td>
<td>-3.818</td>
<td>-7.957</td>
<td>-4.150</td>
</tr>
<tr>
<td>11</td>
<td>12.0000</td>
<td>6.0466</td>
<td>-.83</td>
<td>-2.57</td>
<td>-6.345</td>
<td>-2.634</td>
<td>-2.000</td>
<td>-1.734</td>
</tr>
<tr>
<td>12</td>
<td>16.0000</td>
<td>6.0454</td>
<td>-.98</td>
<td>-1.93</td>
<td>-2.564</td>
<td>-1.455</td>
<td>-1.711</td>
<td>-9.525</td>
</tr>
<tr>
<td>13</td>
<td>-29.0000</td>
<td>6.0506</td>
<td>2.03</td>
<td>2.03</td>
<td>-2.802E-14</td>
<td>-2.802E-14</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>14</td>
<td>-24.0000</td>
<td>6.0491</td>
<td>1.18</td>
<td>-1.99</td>
<td>-2.428</td>
<td>-9.313E-01</td>
<td>0.1497</td>
<td>-3.171</td>
</tr>
<tr>
<td>15</td>
<td>-20.0000</td>
<td>6.0489</td>
<td>.74</td>
<td>-2.15</td>
<td>-1.943</td>
<td>7.926E-01</td>
<td>-1.151</td>
<td>-2.893</td>
</tr>
<tr>
<td>16</td>
<td>-16.0000</td>
<td>6.0514</td>
<td>.41</td>
<td>-2.21</td>
<td>-8.707</td>
<td>7.597</td>
<td>-1.111</td>
<td>-2.620</td>
</tr>
<tr>
<td>17</td>
<td>-12.0000</td>
<td>6.0564</td>
<td>.90</td>
<td>-1.96</td>
<td>8.623</td>
<td>1.722</td>
<td>0.8594</td>
<td>2.856</td>
</tr>
<tr>
<td>18</td>
<td>-8.0000</td>
<td>6.0628</td>
<td>1.31</td>
<td>-2.62</td>
<td>2.542</td>
<td>2.012</td>
<td>-5.299</td>
<td>-3.925</td>
</tr>
<tr>
<td>19</td>
<td>-4.0000</td>
<td>6.0825</td>
<td>2.17</td>
<td>1.04</td>
<td>4.754</td>
<td>8.084</td>
<td>3.329</td>
<td>-1.132</td>
</tr>
<tr>
<td>20</td>
<td>0.0000</td>
<td>6.0973</td>
<td>17.21</td>
<td>5.97</td>
<td>-6.386</td>
<td>-1.734</td>
<td>-1.095</td>
<td>-11.24</td>
</tr>
<tr>
<td>21</td>
<td>4.0000</td>
<td>6.0769</td>
<td>.02</td>
<td>.49</td>
<td>-5.611</td>
<td>-7.729</td>
<td>-2.118</td>
<td>.4667</td>
</tr>
<tr>
<td>22</td>
<td>8.0000</td>
<td>6.0541</td>
<td>1.03</td>
<td>-3.12</td>
<td>-3.022</td>
<td>3.818</td>
<td>.7957</td>
<td>-4.150</td>
</tr>
<tr>
<td>23</td>
<td>12.0000</td>
<td>6.0466</td>
<td>-.83</td>
<td>-2.57</td>
<td>-6.345</td>
<td>2.634</td>
<td>2.000</td>
<td>-1.734</td>
</tr>
<tr>
<td>24</td>
<td>16.0000</td>
<td>6.0454</td>
<td>-.98</td>
<td>-1.93</td>
<td>-2.564</td>
<td>1.455</td>
<td>1.711</td>
<td>-9.525</td>
</tr>
<tr>
<td>I</td>
<td>XJ</td>
<td>YJ</td>
<td>UJ</td>
<td>UJN</td>
<td>VJ</td>
<td>VJN</td>
<td>THETA</td>
<td>THETAN</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>-29.0000</td>
<td>6.0506</td>
<td>807.57</td>
<td>807.57</td>
<td>-2802E-14</td>
<td>-2802E-14</td>
<td>-3469E-17</td>
<td>-3469E-17</td>
</tr>
<tr>
<td>2</td>
<td>-24.0000</td>
<td>6.0491</td>
<td>806.72</td>
<td>803.55</td>
<td>-2428</td>
<td>-9313E-01</td>
<td>-3010E-03</td>
<td>-1159E-03</td>
</tr>
<tr>
<td>3</td>
<td>-20.0000</td>
<td>6.0489</td>
<td>806.28</td>
<td>803.39</td>
<td>1.943</td>
<td>7926E-01</td>
<td>2410E-03</td>
<td>9865E-04</td>
</tr>
<tr>
<td>4</td>
<td>-16.0000</td>
<td>6.0514</td>
<td>805.96</td>
<td>803.34</td>
<td>8707</td>
<td>7597</td>
<td>1080E-02</td>
<td>9457E-03</td>
</tr>
<tr>
<td>5</td>
<td>-12.0000</td>
<td>6.0564</td>
<td>806.44</td>
<td>803.58</td>
<td>8623</td>
<td>1.722</td>
<td>1069E-02</td>
<td>2142E-02</td>
</tr>
<tr>
<td>6</td>
<td>-8.0000</td>
<td>6.0628</td>
<td>806.85</td>
<td>802.93</td>
<td>2.342</td>
<td>2.012</td>
<td>3150E-02</td>
<td>2350E-02</td>
</tr>
<tr>
<td>7</td>
<td>-4.0000</td>
<td>6.0825</td>
<td>807.72</td>
<td>806.58</td>
<td>4.754</td>
<td>8.084</td>
<td>5886E-02</td>
<td>1002E-01</td>
</tr>
<tr>
<td>8</td>
<td>0.0000</td>
<td>6.0973</td>
<td>822.75</td>
<td>811.51</td>
<td>6.386</td>
<td>1.734</td>
<td>7762E-03</td>
<td>2136E-02</td>
</tr>
<tr>
<td>9</td>
<td>4.0000</td>
<td>6.0769</td>
<td>805.57</td>
<td>806.03</td>
<td>-5.611</td>
<td>-7.729</td>
<td>-6966E-02</td>
<td>9589E-02</td>
</tr>
<tr>
<td>10</td>
<td>8.0000</td>
<td>6.0541</td>
<td>806.57</td>
<td>802.42</td>
<td>-3.022</td>
<td>-3.818</td>
<td>-3747E-02</td>
<td>4758E-02</td>
</tr>
<tr>
<td>11</td>
<td>12.0000</td>
<td>6.0466</td>
<td>804.71</td>
<td>802.98</td>
<td>-6.345</td>
<td>2.634</td>
<td>7885E-03</td>
<td>3280E-02</td>
</tr>
<tr>
<td>12</td>
<td>16.0000</td>
<td>6.0454</td>
<td>804.57</td>
<td>803.61</td>
<td>2.564</td>
<td>1.455</td>
<td>3186E-03</td>
<td>1811E-02</td>
</tr>
<tr>
<td>13</td>
<td>20.0000</td>
<td>6.0506</td>
<td>807.57</td>
<td>807.57</td>
<td>-2802E-14</td>
<td>-2802E-14</td>
<td>-3469E-17</td>
<td>-3469E-17</td>
</tr>
<tr>
<td>14</td>
<td>24.0000</td>
<td>6.0491</td>
<td>806.72</td>
<td>803.55</td>
<td>-2428</td>
<td>-9313E-01</td>
<td>-3010E-03</td>
<td>-1159E-03</td>
</tr>
<tr>
<td>15</td>
<td>28.0000</td>
<td>6.0489</td>
<td>806.28</td>
<td>803.39</td>
<td>-1.943</td>
<td>-7926E-01</td>
<td>-2410E-03</td>
<td>-9865E-04</td>
</tr>
<tr>
<td>16</td>
<td>32.0000</td>
<td>6.0514</td>
<td>805.96</td>
<td>803.34</td>
<td>-8707</td>
<td>7597</td>
<td>1080E-02</td>
<td>9457E-03</td>
</tr>
<tr>
<td>17</td>
<td>36.0000</td>
<td>6.0564</td>
<td>806.44</td>
<td>803.58</td>
<td>-8623</td>
<td>1.722</td>
<td>1069E-02</td>
<td>2142E-02</td>
</tr>
<tr>
<td>18</td>
<td>40.0000</td>
<td>6.0628</td>
<td>806.85</td>
<td>802.93</td>
<td>-4.342</td>
<td>-2.012</td>
<td>3150E-02</td>
<td>2350E-02</td>
</tr>
<tr>
<td>19</td>
<td>44.0000</td>
<td>6.0825</td>
<td>807.72</td>
<td>806.58</td>
<td>4.754</td>
<td>8.084</td>
<td>5886E-02</td>
<td>1002E-01</td>
</tr>
<tr>
<td>20</td>
<td>48.0000</td>
<td>6.0973</td>
<td>822.75</td>
<td>811.51</td>
<td>6.386</td>
<td>1.734</td>
<td>7762E-03</td>
<td>2136E-02</td>
</tr>
<tr>
<td>21</td>
<td>52.0000</td>
<td>6.0769</td>
<td>805.57</td>
<td>806.03</td>
<td>5.611</td>
<td>7.729</td>
<td>6966E-02</td>
<td>9589E-02</td>
</tr>
<tr>
<td>22</td>
<td>56.0000</td>
<td>6.0541</td>
<td>806.57</td>
<td>802.42</td>
<td>3.022</td>
<td>3.818</td>
<td>3747E-02</td>
<td>4758E-02</td>
</tr>
<tr>
<td>23</td>
<td>60.0000</td>
<td>6.0466</td>
<td>804.71</td>
<td>802.98</td>
<td>6.345</td>
<td>2.634</td>
<td>7885E-03</td>
<td>3280E-02</td>
</tr>
<tr>
<td>24</td>
<td>64.0000</td>
<td>6.0454</td>
<td>804.57</td>
<td>803.61</td>
<td>-2.564</td>
<td>1.455</td>
<td>3186E-03</td>
<td>1811E-02</td>
</tr>
</tbody>
</table>

RMS(DYJ(TOP)) = 802531E-02
RMS(DYJ(HOT)) = 802531E-02
UNSCALED WALL CORRECTIONS FOR A RELAXATION FACTOR = 0.25000

6X19 INCH TRANSONIC TUNNEL FLEXIBLE WALL TEST
USING RESULTS FROM TEST 39 RUN 31 POINT 35

TUNNEL PARAMETERS ARE

\( M = 0.7670 \)
\( F = 3.073 \)
\( \text{ALPHA} = 0.000 \)
\( \text{TEMP(DEG R)} = 508.472 \)
\( \text{P(TOP)} = 20.62 \)
\( \text{P(INF)} = 13.91 \)
\( \text{U(INF)} = 805.54225 \)

<table>
<thead>
<tr>
<th>I</th>
<th>X</th>
<th>Y</th>
<th>YJ</th>
<th>YJN</th>
<th>D</th>
<th>DJ</th>
<th>DJN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-29,000</td>
<td>9.50000</td>
<td>9.50000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>-24,000</td>
<td>9.49765</td>
<td>9.49794</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>-20,000</td>
<td>9.49739</td>
<td>9.49777</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>-16,000</td>
<td>9.50130</td>
<td>9.50121</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>-12,000</td>
<td>9.50918</td>
<td>9.51018</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>-8,0000</td>
<td>9.51911</td>
<td>9.51993</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>-4,0000</td>
<td>9.55012</td>
<td>9.55397</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>0.0000</td>
<td>9.57337</td>
<td>9.56013</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>4.0000</td>
<td>9.54136</td>
<td>9.54408</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>8.0000</td>
<td>9.50557</td>
<td>9.50567</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>11</td>
<td>12.000</td>
<td>9.49375</td>
<td>9.49126</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>16.000</td>
<td>9.49179</td>
<td>9.49535</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>13</td>
<td>-29,000</td>
<td>-9.50000</td>
<td>-9.50000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td>-24,000</td>
<td>-9.49765</td>
<td>-9.49794</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td>-20,000</td>
<td>-9.49739</td>
<td>-9.49777</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>16</td>
<td>-16,000</td>
<td>-9.50130</td>
<td>-9.50121</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>17</td>
<td>-12,000</td>
<td>-9.50918</td>
<td>-9.51018</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>18</td>
<td>-8,0000</td>
<td>-9.51911</td>
<td>-9.51993</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>19</td>
<td>-4,0000</td>
<td>-9.55012</td>
<td>-9.55397</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20</td>
<td>0.0000</td>
<td>-9.57337</td>
<td>-9.58013</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>21</td>
<td>4.0000</td>
<td>-9.54136</td>
<td>-9.54408</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>22</td>
<td>8.0000</td>
<td>-9.50557</td>
<td>-9.50567</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>23</td>
<td>12.000</td>
<td>-9.49375</td>
<td>-9.49126</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>24</td>
<td>16.000</td>
<td>-9.49179</td>
<td>-9.48535</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

RMS(DYJ(TOP)) = 0.312684E-02
RMS(DYJ(BOT)) = 0.312684E-02
**UNSCALED WALL CORRECTIONS FOR A RELAXATION FACTOR = 1.00000**

**6X19 INCH TRANSONIC TUNNEL FLEXIBLE WALL TEST**

**USING RESULTS FROM TEST 39 RUN 31 POINT 35**

**TUNNEL PARAMETERS ARE**

- $M = 0.7670$
- $R_n = 3.073$
- $\alpha = 0.000$
- $T_{\text{temp}}(\text{deg} R) = 508.472$
- $P_T = 20.62$
- $P_{\infty} = 13.91$
- $V_{\infty} = 805.5425$

<table>
<thead>
<tr>
<th>I</th>
<th>XJ</th>
<th>YJ</th>
<th>YJN</th>
<th>D0J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-29.0000</td>
<td>9.50000</td>
<td>9.50000</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>-24.0000</td>
<td>9.49765</td>
<td>9.49879</td>
<td>1.14142E-02</td>
</tr>
<tr>
<td>3</td>
<td>-20.0000</td>
<td>9.49739</td>
<td>9.49890</td>
<td>1.50996E-02</td>
</tr>
<tr>
<td>4</td>
<td>-16.0000</td>
<td>9.50130</td>
<td>9.50093</td>
<td>-3.71557E-03</td>
</tr>
<tr>
<td>5</td>
<td>-12.0000</td>
<td>9.50918</td>
<td>9.51318</td>
<td>3.39653E-02</td>
</tr>
<tr>
<td>6</td>
<td>-8.0000</td>
<td>9.51911</td>
<td>9.52238</td>
<td>-3.26924E-02</td>
</tr>
<tr>
<td>7</td>
<td>-4.0000</td>
<td>9.55012</td>
<td>9.56552</td>
<td>1.53961E-01</td>
</tr>
<tr>
<td>8</td>
<td>0.</td>
<td>9.57337</td>
<td>9.60039</td>
<td>2.70234E-01</td>
</tr>
<tr>
<td>9</td>
<td>4.0000</td>
<td>9.54136</td>
<td>9.55224</td>
<td>-1.08799E-01</td>
</tr>
<tr>
<td>10</td>
<td>8.0000</td>
<td>9.50557</td>
<td>9.50598</td>
<td>4.07313E-03</td>
</tr>
<tr>
<td>12</td>
<td>16.0000</td>
<td>9.49179</td>
<td>9.46603</td>
<td>-2.57616E-01</td>
</tr>
<tr>
<td>13</td>
<td>-29.0000</td>
<td>-9.50000</td>
<td>-9.50000</td>
<td>0.</td>
</tr>
<tr>
<td>14</td>
<td>-24.0000</td>
<td>-9.49765</td>
<td>-9.49879</td>
<td>-1.14142E-02</td>
</tr>
<tr>
<td>15</td>
<td>-20.0000</td>
<td>-9.49739</td>
<td>-9.49890</td>
<td>-1.50996E-02</td>
</tr>
<tr>
<td>16</td>
<td>-16.0000</td>
<td>-9.50130</td>
<td>-9.50093</td>
<td>-3.71557E-03</td>
</tr>
<tr>
<td>17</td>
<td>-12.0000</td>
<td>-9.50918</td>
<td>-9.51318</td>
<td>-3.39653E-02</td>
</tr>
<tr>
<td>18</td>
<td>-8.0000</td>
<td>-9.51911</td>
<td>-9.52238</td>
<td>-3.26924E-02</td>
</tr>
<tr>
<td>19</td>
<td>-4.0000</td>
<td>-9.55012</td>
<td>-9.56552</td>
<td>-1.53961E-01</td>
</tr>
<tr>
<td>20</td>
<td>0.</td>
<td>-9.57337</td>
<td>-9.60039</td>
<td>-2.70234E-01</td>
</tr>
<tr>
<td>21</td>
<td>4.0000</td>
<td>-9.54136</td>
<td>-9.55224</td>
<td>-1.08799E-01</td>
</tr>
<tr>
<td>22</td>
<td>8.0000</td>
<td>-9.50557</td>
<td>-9.50598</td>
<td>-4.07313E-03</td>
</tr>
</tbody>
</table>

$\text{RMS(D0J(TOP))} = 1.25074E-01$

$\text{RMS(D0J(ROT))} = 1.25074E-01$
### FLEXWAL: A Computer Program for Predicting the Wall Modifications for Two-Dimensional, Solid, Adaptive-Wall Wind Tunnels

#### Abstract

A program called FLEXWAL for calculating wall modifications for solid, adaptive-wall wind tunnels is presented. The method used is the iterative technique of NASA TP-2081 and is applicable to subsonic and transonic test conditions. The program usage, program listing, and a sample case are given.

#### Key Words
- Wind Tunnels
- Wall Interference
- Adaptive Wind-Tunnel Walls
- Transonic Flow
- Cauchy Integral Formula