User Manual and Listing of

Taku
A Three-Dimensional, Time-Dependent Flow Model for Glaciers

Bryan R. Pearce
Liaqat Ali Khan
Peter S. Sucsy
Vijay G. Panchang

Maine/New Hampshire Sea Grant College Program
with the University of Maine Center for Marine Studies
and Department of Civil Engineering

TR-MSG-89-2
11/89 300

Published by the Sea Grant College Program at the University of Maine under grant NA89-AA-D-SG020 from the National Sea Grant College Program, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, and with support from the University of Maine's Center for Marine Studies and Department of Civil Engineering.

User Manual and Listing of

TAKU

A Three-Dimensional, Time-Dependent Flow Model for Glaciers

by

Bryan R. Pearce
Liaqat Ali Khan
Peter S. Sucsy
Vijay G. Panchang

August 1989

Department of Civil Engineering
University of Maine
Orono, ME
04469

TABLE OF CONTENTS

I. INTRODUCTION

The TAKU is a time-dependent, numerical model for the simulation of the flow characteristics of glaciers. The model is three-dimensional in the sense that it computes the vertical variations of the horizontal velocity components. It is assumed that the glacial ice can be treated as a highly viscous Newtonian fluid. The model solves the depth integrated two-dimensional unsteady continuity equation, together with the Navier-Stoke's equations in the x- and y-directions.

The details of the model are been described in "A Numerical Flow Model of the Taku Glacier, Alaska", by Pearce et al.(1989). The equation number cited in this report refers to Chapter-2 of that report.

Figure-1 shows the grid set-up for a hypothetical glacier. Note the additional layer of cells surrounding the glacier. The length and the width of the glacier are two grids less than the grid set-up of the model. The bed and the surface elevations in any cell outside the glacier domain are set equal to zero, unless the glacier is expected to advance in the cell over that simulation period. The surface elevation in the cell in which the glacier is expected to advance in the simulations is set equal to the bed elevation. It should be noted that the bed elevation in any cell inside the glacier domain can not be zero. If actual bed elevation is zero, then a small positive/negative number should be assigned to the grid.

This manual describes the model, the EXEC file used for executing the FORTRAN program, the input/ output files, the user specified input variables, the output variables, a listing of the model and the auxiliary plotting routines.

The research was supported in part by the University of Maine Center for Marine Studies, by the Maine Sea Grant College Program (Grant No. NA86AA-D-SG047), and by the University of Maine Department of Civil Engineering.

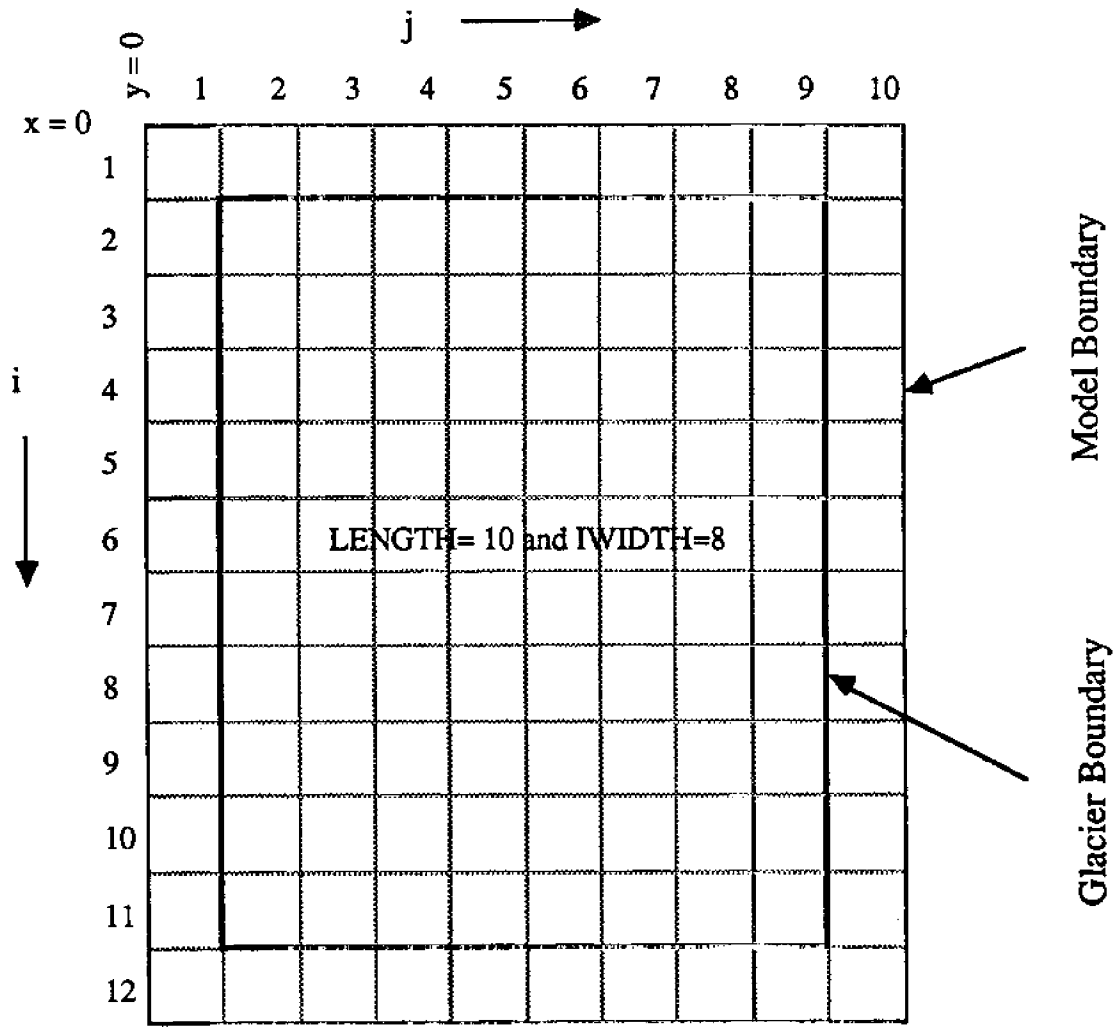

Fig. 1: Grid Setup of the Model

II. DESCRIPTION OF THE COMPUTER PROGRAM

The model consists of a main program (PROGRAM TAKU), fourteen Subroutines and a Function Subroutine. These are as follows:

Module Name	Function of the Module
PROGRAM TAKU	...This is the main program, mainly controlling the sequence of the computations by calling subroutines.
INPUT	...Subroutine for reading all user defined input data from data file [TAKU DATA]. This subroutine also performs some preliminary computations.
INITIL	...Subroutine for the initialization of variables, and reading data for "hot start", if desired.
FILTER	...This subroutine filters the bed topography, friction coefficients, and accumulation/ablation rates specified at the computational cell. This is done only once at the beginning the the computations.
ACALC	...Determines the slip angle at the bed, using the relationship given by equation(2.23c). The Newton-Rapson method is used for the solution of the transcendental equation.
GALIN1	...Evaluates the integrals S1(J), S2(I,J), and S3(J), defined by equations(2.29a), (2.29b) and (2.29c). These integrals are evaluated analytically.
GALIN2	...Evaluates the integrals S4(I,J), S5(I,J), and S6(I,J), defined by equations(2.29d), (2.29e) and (2.29f). These integrals are evaluated by Gaussian Quadrature.
GALIN3	...Evaluates the integrals S8(I,J), S9(I,J), and S10(I,J), defined by equations(2.29h), (2.29i) and (2.29j), by Gaussian Quadrature.
XMOM	...Evaluates the coefficients α_x, β_x and γ_x in

	equation(2.35a) and solves the transformed x-momentum equation by an explicit finite difference scheme.
YMOM	...Evaluates the coefficients α_y, β_y and γ_y in equation(2.35b) and solves the transformed y-momentum equation by an explicit finite difference scheme.
CONTIN	...This subroutine solved the continuity equation [equation 3.36] explicitly.
MOVEIN	...Applies the moving boundary condition at the terminus [equations (2.10), (2.11) and (2.12)].
CLIMATE	...Determines the altitudinal shift of the equilibrium line, and the distribution of ablation rates below the equilibrium line.
VELOB	...Computes the bed velocity based on equation(2.22).
VELOC	...Computes the vertical distribution of velocity at selected, equally spaced, intervals. This routine also determines the mean velocity of flow.
OUTPUT	...This subroutine is used for printing/storing output of the model. This subroutine is also called by the main program for storing output for "hot start".

Subroutines INITIL and OUTPUT have ENTRY Points, defined as INIT and OUPT respectively, and are used for reading/ storing data for "hot start".

III. EXEC, INPUT/ OUTPUT FILES

The listing of the EXEC file used for running the model is as follows:

```
GLOBAL TXTLIB VSF2FORT CMSLIB
FI 1 DISK TAKU  DATA  D (LRECL 80 BLOCK 80 RECFM F DSORG DA
FI 2 DISK HOTST DATA  D (RECFM VS LRECL 32000 BLOCK 32004
FI 3 TERMINAL
FI 11 DISK TEMPA DATA D (LRECL 132 BLOCK 132 RECFM F DSORG DA
FI 12 DISK TEMPB DATA D (LRECL 132 BLOCK 132 RECFM F DSORG DA
FI 13 DISK TEMPC DATA D (LRECL 132 BLOCK 132 RECFM F DSORG DA
FI 14 DISK TEMPD DATA D (LRECL  80 BLOCK  80 RECFM F DSORG DA
FI 15 DISK TEMPE DATA D (LRECL  80 BLOCK  80 RECFM F DSORG DA
LOAD TAKU  (START
```

The TAKU DATA is the only user defined input file. The HOTST DATA is generated by the model at time specified by the user [details discussed later] and is used as input file if "hot start" of the model is desired.

The output files stores the following information:

TEMPA.DATA	..Temporal variation of surface elevations at selected locations.
TEMPB.DATA	..Temporal variation of surface velocities at selected locations
TEMPC.DATA	..Vertical variation of velocity at selected location
TEMPD.DATA	..Spatial distributions of the following parameters: i. Change of surface elevation over the glacier ii. Surface, bed velocities and the ratio of bed to surface velocity. iii. Bed shear stress.
TEMPE.DATA	...Temporal variations of volume of the glacier, accumulation and ablation areas.

The temporal variations of surface elevations and surface velocities are stored at ten year intervals, while the output for vertical velocity profiles, change in surface elevation, spatial distributions of surface and bed velocities, and their ratio, and bed shear stress are stored every hundred years.

IV. USER DEFINED INPUT VARIABLES

A. Parameter Statements

The variables in the Parameter Statements are:

1. JP	Number of cosine terms to be used in constructing the vertical variation of velocity [equation 2.22].
2. KP	Number of levels [σ= 0 to 2 , equation 2.14] at which vertical velocity is to be computed.
3. LENGTH	Number of grid elements in the x-direction.
4. IWIDTH	Number of grid elements in the y-direction.

B. Namelist

The model input is defined in five namelists. These are as follows:

1. LOGCL

HOTST= .T., if "hot start" is desired [input file HOTST DATA must exists],
.F., if otherwise.

FTTOM=.T., if input data for surface and bed elevations are in FPS units,
.F., if these data are in SI units.

HVISC =.T., if horizontal viscosity is to be included in the computations,
.F., if otherwise.

VFRICT=.T., if the friction coefficient is assumed function of bed elevation ,
=.F., if the friction coefficient to be used at each cell is as specified in the input file [TAKU DATA].

MOVE =.T., if moving boundary condition is to be used at the terminus of the glacier,
=.F., if no-flow boundary condition is to be used at the terminus.

CLIMET=.T., if global warming of the atmosphere is to be included in the computations,
=.F., if otherwise.

PRT =.T., if print of the input data is desired on the screen,
=.F., if otherwise.

If CLIMET is set to .T., then HOTST should be set to .F., as the present version of the model does not store the data necessary for "hot start" with climatic warming.

2. SYSTEM

H(I,J)	= bed topography of the glacier (ft or m).
ETA(I,J)	= initial surface topography of the glacier (ft or m).
SOS(I,J)	= source (accumulation) and sink (ablation) (m/year).
FMAX	= maximum value of the friction coefficient (m/day).
FMIN	= minimum value of the friction coefficient (m/day).
FRICT(I,J)	= friction coefficient (m/day).
NV	= vertical viscosity of ice (m^2/s).
NH	= horizontal viscosity of ice (m^2/s).
RHO	= density of ice (km/m^3).

The actual values of the friction coefficients, used in the model, are FMAX ($x10^6$), FMIN ($x10^6$), and FRICT ($x10^6$). For numerical stability of the model, NH ≤ NV.

3. CNTRL

START	= computational step when the model starts printing/ storing the output.
PHOTST	= a "dummy " variable.
POUTPT	= a "dummy" variable.
NHOTST	= time step when "hot start" data is to be stored.

4. MODPAR

DELT	= time step in days.
DELX	=distance step in the x-direction(km).
DELY	=distance step in the y-direction (km).
NFIN	=time step over which the simulation is to be performed.
UPAI	=interval for updating the integrals.

5. METEO

HEQL	=initial altitude of the equilibrium line(m).
DELC	=change in accumulation rate (kg/m^2).
GRADC	=altitudinal gradient of accumulation rate (kg/m^2-m).
GRADT	=altitudinal gradient of temperature (°C/m).
ALFA1	=bulk heat transfer coefficient ($J\ m^2\ d^{-1}$ °C).
ALFA2	=radiative coefficient($J\ m^2\ d^{-1}$ °C).
TDEL	=rate of increase in temperature (°C every 10 years).
SMIN	=ablation rate at and below the mean sea level(m/year).

V. MODEL OUTPUT VARIABLES

The subroutine OUTPUT is exclusively used for printing and storing the model output. The variables stored in the output files are as follows:

NETA(I,J)	=elevation of the glacier surface (m).
VV(K,I,J)	=resultant velocity [=$\sqrt{(u^2 + v^2)}$], K= KP is the surface velocity, K= 1 is the bed velocity, while velocity at intermediate levels can be obtained by setting K to any value between 1 and KP (m/day).
SIGMA	=variable defined by equation(2.14).
HT(I,J)	=change in surface elevation over the simulation period(m).
HTMAX	=maximum value of HT(m).
HTMIN	=minimum value of HT(m).
TW(I,J)	=bed shear stress (kN/m^2).
TWMAX	=maximum value of TW (kN/m^2).
TWMIN	=minimum value of TW (kN/m^2).
VSMAX	=maximum value of surface velocity (m/day).
VSMIN	=minimum value of surface velocity (m/day).
VBMAX	=maximum value of bed velocity (m/day).
VBMIN	=minimum value of bed velocity (m/day).
VR(I,J)	=ratio of bed velocity to surface velocity.
VRMAX	=maximum value of VR.
VRMIN	=minimum value of VR.
VOL	=volume of the glacier (km^3).
ACC	=accumulation area (km^2).
ABL	=ablation area (km^2).
SPO	=total accumulation ($x10^6$ m^2/year).
SNE	=total ablation ($x10^6$ m^2/year).
HEQL	=elevation of the equilibrium line (m).

VI. AUXILIARY PLOTTING ROUTINES

Beside CRICKET GRAPH, for making x-y plots, three additional plotting routines are used. These are:

A. VELOCITY FIELD

The velocity field in the glacier is plotted by VECTOR FORTRAN. The EXEC file used for obtaining the plot on the computer screen is as follows:

```
/* HELLO */
'GLOBAL TXTLIB VSF2FORT CMSLIB TEK1073 CALCOMP'
'FI 1 DISK VECTOR DATA D (LRECL 80 BLOCK 80 RECFM F DSORG DA'
'FI 2 DISK HOTST DATA  D (RECFM VS LRECL 32000 BLOCK 32004'
'LOAD VECTOR (START'
```

The user defined variables, in the input VECTOR DATA file, are as follows:

GRID	=.T., if grid setup is to be plotted,
	=.F., if otherwise.
PLTSIZ	= plot size in inches.
DELX	=grid size in the x-direction in km.
DELY	=grid size in the y-direction in km.
IWMAX	=maximum value of grid number in the y-direction for the plot.
LMAX	=maximum value of grid number in the x-direction for the plot.
IWMIN	=minimum value of grid number in the y-direction for the plot.
LMIN	=minimum value of grid number in the x-direction for the plot.
TIME	=the time corresponding to the plotting data, in year.
SLEVEL	=any value between 0 and 2,
	=2.0, for surface velocity plot,
	=0.0, for bed velocity plot.

Other information necessary for the plot is obtained from the HOTST DATA generated by the TAKU.

To use this routine it is necessary to shear ACADEMIC 491 to access CALCOMP plotting routines. Note that by erasing "TEK1073", the plot can be send to CALCOMP plotter.

A listing this routine is provided in the appendix.

B. 3-D TOPOGRAPHY

Three-dimensional plots of the bed and the surface topography of the glacier can be obtained by using PROC G3D of SAS GRAPH. The following routine, called PLT3D SAS, will plot the topography on the computer screen.

```
GOPTIONS DEVICE=TEK4105 TRANTAB=GTABCMS
GPROTOCOL=GSAS7171 NOTEXT82;
CMS FI PLT  DISK PLT DATA D1 (RECFM F LRECL 80 BLOCK 80;
DATA;
INFILE   PLT;
INPUT X Y Z @@;
PROC G3D;
PLOT Y*X=Z/ ROTATE= 30, 45, 60;
TITLE 'SURFACE TOPOGRAPHY OF THE TAKU GLACIER';
```

The input data file is PLT DATA. The variables X and Y are the distance in the x and y-directions respectively, while Z is either surface or bed topography. The values of these variables should be stored consecutively in the PLT DATA file. Any data (elevation) outside the model domain should be set to any arbitrary value, preferably to zero. The routine will successively plot the topography at three different angles (30°, 45° and 60°).

Note that by replacing "TEK4105" by "CAL5500" the plots can be send to CALCOMP plotter. It is necessary to have 2 MB of working memory and access SAS disk.

C. CONTOURS

The contour plots of the bed, surface topography or the mass balance of the glacier can be obtained by using PROC GCONTOUR of SAS GRAPH. The following routine, called PLTC SAS, will plot the contours on the computer screen.

```
GOPTIONS DEVICE=TEK4105 TRANTAB=GTABCMS
GPROTOCOL=GSAS7171 NOTEXT82;
CMS FI PLT   DISK PLT   DATA D1 (RECFM F LRECL 80 BLOCK 80;
DATA;
INFILE   PLT;
INPUT Y X Z @@;
PROC GCONTOUR;
PLOT Y*X=Z/LEVELS= 200,400,600,800,1000,1200,1400,1600,1800;
TITLE 'SURFACE TOPOGRAPHY OF THE TAKU GLACIER';
```

The input data file is PLT DATA. The variables X and Y are the distance in the x and y-directions respectively, while Z is the elevation of surface/ bed or mass balance. The values of these variables should be stored consecutively in the PLT DATA file. Any data (elevation) outside the model domain should be set arbitrarily to very high or low value (outside the range of the input data). The contour levels to be plotted by the routine are 200, 400, 600, 800, 1000, 1200, 1400, 1600 and 1800.

Note that by replacing "TEK4105" by "CAL5500" the plots can be send to CALCOMP plotter. It is necessary to have 2 MB of working memory and access SAS disk.

REFERENCE

Pearce, B.R., et al.,1989, A Numerical Flow Model of the Taku Glacier, Alaska, Tech. Rept., Department of Civil Engineering, University of Maine, Orono, Maine, U.S.A.

APPENDIX-A: LISTING OF TAKU FORTRAN

```
      PROGRAM TAKU                                                      TAK00010
C*****************************************************************    TAK00020
C*                                                               *    TAK00030
C*     TAKU: A THREE-DIMENSIONAL, TIME-DEPENDENT FLOW MODEL      *    TAK00040
C*                OF THE TAKU GLACIER, ALASKA                    *    TAK00050
C*                                              APRIL 1989.      *    TAK00060
C*                                                               *    TAK00070
C*****************************************************************    TAK00080
C*                                                               *    TAK00090
C*   DEVELOPED BY:    LIAQAT ALI KHAN                            *    TAK00100
C*                    BRYAN R. PEARCE                            *    TAK00110
C*                                        DATE:   JULY, 1989.    *    TAK00120
C*    NOTES:                                                     *    TAK00130
C*      1. THE DESCRIPTION OF THE MODEL CAN BE FOUND IN          *    TAK00140
C*         PEARCE ET AL.(1989).                                  *    TAK00150
C*      2. THE LIST OF INPUT AND OUTPUT VARIABLES ARE LISTED IN  *    TAK00160
C*         THE 'USERS MANUAL' BY PEARCE ET AL.(1989)             *    TAK00170
C*      3. THIS MODEL IS BASED ON THE HYDRODYNAMIC MODEL:'TIDE',  *    TAK00180
C*         SUCSY AND PEARCE (1986)                               *    TAK00190
C*                                                               *    TAK00200
C*****************************************************************    TAK00210
C*                                                                    TAK00220
C*                                                                    TAK00230
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                       TAK00240
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,         TAK00250
     &           IWPLS=IWIDTH+2)                                       TAK00260
C                                                                      TAK00270
      REAL   NV,NH,NC,ND,NETA                                          TAK00280
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                 TAK00290
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE,CLIMET               TAK00300
C                                                                      TAK00310
      DIMENSION HI(LPLS,IWPLS),SSS(LPLS,IWPLS)                         TAK00320
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO          TAK00330
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,           TAK00340
     &              ISTRT,NFIN,NHOTST                                  TAK00350
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE,CLIMET               TAK00360
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),            TAK00370
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)           TAK00380
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)   TAK00390
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)              TAK00400
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),      TAK00410
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),      TAK00420
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),   TAK00430
     &              INTG7(JP,JP,LPLS,IWPLS)                            TAK00440
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                       TAK00450
      COMMON/ BLK10/ QIF(LPLS,IWPLS),QIB(LPLS,IWPLS),                  TAK00460
     &               QJF(LPLS,IWPLS),QJB(LPLS,IWPLS)                   TAK00470
      COMMON/ BLK11/ HEQL,DELC,GRADC,GRADT,ALFA1,ALFA2,TDEL,SMIN       TAK00480
C                                                                      TAK00490
C                                                                      TAK00500
C                                                                      TAK00510
        PI=3.14159                                                     TAK00520
        CNT=1.0                                                        TAK00530
        TIMEST=0.0                                                     TAK00540
```

```
*                                                                       TAK00550
*****.........READ INPUT DATA                                          TAK00560
*                                                                       TAK00570
         CALL INPUT                                                     TAK00580
*                                                                       TAK00590
      PHOTST=PHOTST/DELT                                                TAK00600
      POUTPT=POUTPT/DELT                                                TAK00610
*                                                                       TAK00620
*****......... INITIALIZES AND SETS COUNTERS                           TAK00630
*                                                                       TAK00640
         CALL INITIL(HI,SSS)                                            TAK00650
*                                                                       TAK00660
*****.........DETERMINE GRIDS INSIDE MODEL DOMAIN                      TAK00670
*                                                                       TAK00680
         IF(MOVE) THEN                                                  TAK00690
            DO 101 I=1,LPLS                                             TAK00700
               DO 101 J=1,IWPLS                                         TAK00710
                  IF(H(I,J).EQ.0.0) THEN                                TAK00720
                     IGRID(I,J)=1                                       TAK00730
                  ELSE IF( ABS(ETA(I,J)-H(I,J)).LE.1.0) THEN            TAK00740
                     IGRID(I,J)=2                                       TAK00750
                  ELSE                                                  TAK00760
                     IGRID(I,J)=0                                       TAK00770
                  ENDIF                                                 TAK00780
101         CONTINUE                                                    TAK00790
         ELSE                                                           TAK00800
            DO 102 I=1,LPLS                                             TAK00810
               DO 102 J=1,IWPLS                                         TAK00820
                  IF( ABS(ETA(I,J)-H(I,J)).LE.1.0) THEN                 TAK00830
                     IGRID(I,J)=1                                       TAK00840
                  ELSE                                                  TAK00850
                     IGRID(I,J)=0                                       TAK00860
                  ENDIF                                                 TAK00870
102         CONTINUE                                                    TAK00880
          ENDIF                                                         TAK00890
*                                                                       TAK00900
*****.........FILTER INPUT DATA IF NECESSARY                           TAK00910
*                                                                       TAK00920
          IF(.NOT.HOTST)THEN                                            TAK00930
              CALL FILTER(H,IGRID)                                      TAK00940
              CALL FILTER(ETA,IGRID)                                    TAK00950
              CALL FILTER(SOS,IGRID)                                    TAK00960
              CALL FILTER(FRICT,IGRID)                                  TAK00970
          ENDIF                                                         TAK00980
*                                                                       TAK00990
*                                                                       TAK01000
*****.........READ INITIAL VALUES FOR HOT START                        TAK01010
*                                                                       TAK01020
       IF(HOTST) THEN                                                   TAK01030
          CALL INIT(HI,SSS)                                             TAK01040
        ENDIF                                                           TAK01050
*                                                                       TAK01060
*****.........CALCULATION OF CONSTANT FACTORS ***                       TAK01070
*                                                                       TAK01080
       DTDX=DELT/DELX                                                   TAK01090
       DTDY=DELT/DELY                                                   TAK01100
       DXDY=DELX*DELY                                                   TAK01110
       NFINAL=FLOAT(NFIN)*DELT                                          TAK01120
*                                                                       TAK01130
       WRITE(3,1001)                                                    TAK01140
 1001  FORMAT(' ',T10,'BEGIN EXECUTION.....')                           TAK01150
```

```
*                                                                       TAK01160
*****.........SOLVE FOR THE AI'S ***                                    TAK01170
*                                                                       TAK01180
      CALL ACALC                                                        TAK01190
*                                                                       TAK01200
*****.........STORE FREQUENTLY USED NUMBERS                             TAK01210
*                                                                       TAK01220
      DO 103 K=1,IWPLS                                                  TAK01230
          DO 103 L=1,LPLS                                               TAK01240
             IF(IGRID(L,K).NE.0) THEN                                   TAK01250
                DO 104 J=1,JP                                           TAK01260
                  IF(A(J,L,K).LT.0.00001) THEN                          TAK01270
                      CONTI(J,L,K)=1.0                                  TAK01280
                  ELSE                                                  TAK01290
                     CONTI(J,L,K)= SIN(A(J,L,K))/A(J,L,K)               TAK01300
                  ENDIF                                                 TAK01310
 104             CONTINUE                                               TAK01320
             ELSE                                                       TAK01330
                DO 105 J=1,JP                                           TAK01340
                   CONTI(J,L,K)=0.0                                     TAK01350
 105            CONTINUE                                               TAK01360
             ENDIF                                                      TAK01370
 103  CONTINUE                                                          TAK01380
*                                                                       TAK01390
*****.........ALL INTEGRALS BASED ON AI'S                               TAK01400
*                                                                       TAK01410
      CALL GALIN1                                                       TAK01420
      IF(HVISC) THEN                                                    TAK01430
        CALL GALIN2                                                     TAK01440
        CALL GALIN3                                                     TAK01450
      ENDIF                                                             TAK01460
*                                                                       TAK01470
       UPAI=UPAI/(DELT)                                                 TAK01480
       IF(HOTST) THEN                                                   TAK01490
           ISTRT=ISTRT+1                                                TAK01500
       ELSE                                                             TAK01510
           ISTRT=1                                                      TAK01520
       ENDIF                                                            TAK01530
*                                                                       TAK01540
*                                                                       TAK01550
*****.................THE TIME LOOP.....................  *****        TAK01560
*                                                                       TAK01570
*                                                                       TAK01580
      WRITE(3,1002)                                                     TAK01590
 1002 FORMAT(' ',T10,'BEGIN TIME LOOP.........')                        TAK01600
*                                                                       TAK01610
      DO 900 J=ISTRT,NFIN                                               TAK01620
         TIME=DELT*CNT                                                  TAK01630
         ISCRN=MOD(J,1  )                                               TAK01640
         IF(ISCRN.EQ.0) WRITE(3,1003) J                                 TAK01650
 1003     FORMAT(' ',T5,'J=',2X,I15)                                    TAK01660
*                                                                       TAK01670
*                                                                       TAK01680
*****.........UPDATE AI'S AT SELECTED TIME INTERVAL                     TAK01690
*                                                                       TAK01700
            CALL ACALC                                                  TAK01710
            DO 201 K=1,IWPLS                                            TAK01720
               DO 201 L=1,LPLS                                          TAK01730
                  IF(IGRID(L,K).NE.0) GOTO 201                          TAK01740
                  DO 202 I=1,JP                                         TAK01750
                     IF(A(I,L,K).LT.0.00001) THEN                       TAK01760
```

```
                        CONTI(I,L,K)=1.0                                    TAK01770
                      ELSE                                                  TAK01780
                        CONTI(I,L,K)= SIN(A(I,L,K))/A(I,L,K)                TAK01790
                      ENDIF                                                 TAK01800
 202               CONTINUE                                                 TAK01810
 201         CONTINUE                                                       TAK01820
*                                                                           TAK01830
*****.........UPDATE THE INTEGRALS                                         TAK01840
*                                                                           TAK01850
            CALL GALIN1                                                     TAK01860
            IF(HVISC) THEN                                                  TAK01870
              CALL GALIN2                                                   TAK01880
              CALL GALIN3                                                   TAK01890
            ENDIF                                                           TAK01900
*                                                                           TAK01910
*****.........SOLVE MOMENTUM EQUATIONS                                     TAK01920
*                                                                           TAK01930
          CALL XMOM                                                         TAK01940
          CALL YMOM                                                         TAK01950
*                                                                           TAK01960
*****.........SMOOTH C & D AND SAVE C & D                                  TAK01970
*                                                                           TAK01980
          DO 204 K=1,JP                                                     TAK01990
            DO 205 N=1,IWIDTH                                               TAK02000
              DO 205 I=1,LENP1                                              TAK02010
                IF(IGRID(I+1,N+1).EQ.0.AND.IGRID(I,N+1).EQ.0) THEN          TAK02020
                  CX=(NC(K,I+1,N)+3.0*NC(K,I,N)+NC(K,I-1,N))/5.0            TAK02030
                ELSE IF(IGRID(I+1,N+1).EQ.0) THEN                           TAK02040
                  CX=(NC(K,I+1,N)+3.0*NC(K,I,N))/4.0                        TAK02050
                ELSE IF(IGRID(I-1,N).EQ.0) THEN                             TAK02060
                  CX=(3.0*NC(K,I,N)+NC(K,I-1,N))/4.0                        TAK02070
                ELSE                                                        TAK02080
                  CX=NC(K,I,N)                                              TAK02090
                ENDIF                                                       TAK02100
                IG1=IGRID(I,N+2)                                            TAK02110
                IG2=IGRID(I+1,N+2)                                          TAK02120
                IG3=IGRID(I,N)                                              TAK02130
                IG4=IGRID(I+1,N)                                            TAK02140
                IF(IG1.EQ.0.AND.IG2.EQ.0.AND.                               TAK02150
     &             IG3.EQ.0.AND.IG4.EQ.0) THEN                              TAK02160
                  CY=(NC(K,I,N+1)+3.0*NC(K,I,N)+NC(K,I,N-1))/5.0            TAK02170
                ELSE IF(IG1.EQ.0.AND.IG2.EQ.0) THEN                         TAK02180
                  CY=(NC(K,I,N+1)+3.0*NC(K,I,N))/4.0                        TAK02190
                ELSE IF(IG3.EQ.0.AND.IG4.EQ.0) THEN                         TAK02200
                  CY=(3.0*NC(K,I,N)+NC(K,I,N-1))/4.0                        TAK02210
                ELSE                                                        TAK02220
                  CY=NC(K,I,N)                                              TAK02230
                ENDIF                                                       TAK02240
                C(K,I,N)=0.5*(CX+CY)                                        TAK02250
 205        CONTINUE                                                        TAK02260
*                                                                           TAK02270
            DO 206 N=1,IWIDP1                                               TAK02280
              DO 206 I=1,LENGTH                                             TAK02290
                IF(IGRID(I+1,N+1).EQ.0.AND.IGRID(I+1,N).EQ.0) THEN          TAK02300
                  DY=(ND(K,I,N+1)+3.0*ND(K,I,N)+ND(K,I,N-1))/5.0            TAK02310
                ELSE IF(IGRID(I+1,N+1).EQ.0) THEN                           TAK02320
                  DY=(ND(K,I,N+1)+3.0*ND(K,I,N))/4.0                        TAK02330
                ELSE IF(IGRID(I+1,N).EQ.0) THEN                             TAK02340
                  DY=(3.0*ND(K,I,N)+ND(K,I-1,N))/4.0                        TAK02350
                ELSE                                                        TAK02360
                  DY=ND(K,I,N)                                              TAK02370
```

```
                ENDIF                                                    TAK02380
                IG1=IGRID(I,N)                                           TAK02390
                IG2=IGRID(I,N+1)                                         TAK02400
                IG3=IGRID(I+2,N)                                         TAK02410
                IG4=IGRID(I+2,N+1)                                       TAK02420
                IF(IG1.EQ.0.AND.IG2.EQ.0.AND.                            TAK02430
     &                 IG3.EQ.0.AND.IG4.EQ.0) THEN                      TAK02440
                   DX=(ND(K,I+1,N)+3.0*ND(K,I,N)+ND(K,I-1,N))/5.0        TAK02450
                ELSE IF(IG1.EQ.0.AND.IG2.EQ.0) THEN                      TAK02460
                   DX=(ND(K,I-1,N)+3.0*ND(K,I,N))/4.0                    TAK02470
                ELSE IF(IG3.EQ.0.AND.IG4.EQ.0) THEN                      TAK02480
                   DX=(3.0*ND(K,I,N)+ND(K,I+1,N))/4.0                    TAK02490
                ELSE                                                     TAK02500
                   DX=ND(K,I,N)                                          TAK02510
                ENDIF                                                    TAK02520
                D(K,I,N)=0.5*(DX+DY)                                     TAK02530
 206         CONTINUE                                                    TAK02540
 204      CONTINUE                                                       TAK02550
*                                                                        TAK02560
*****.......CHANGE ABLATION RATE FOR CLIMATIC WARMING                    TAK02570
*                                                                        TAK02580
         IF(CLIMET) THEN                                                 TAK02590
            ICALL=MOD(J,60)                                              TAK02600
            IF(ICALL.EQ.0) THEN                                          TAK02610
               CALL CLIMATE(SSS)                                         TAK02620
            ENDIF                                                        TAK02630
         ENDIF                                                           TAK02640
*                                                                        TAK02650
*****.........SOLVE CONTINUITY EQUATION                                  TAK02660
*                                                                        TAK02670
          CALL CONTIN(DTDX,DTDY,SSS)                                     TAK02680
*                                                                        TAK02690
*****.........SAVE ETA VALUES                                            TAK02700
*                                                                        TAK02710
          DO 208 I=1,LPLS                                                TAK02720
             DO 208 N=1,IWPLS                                            TAK02730
               ETA(I,N)=NETA(I,N)                                        TAK02740
 208      CONTINUE                                                       TAK02750
*                                                                        TAK02760
*****.........CHECK TO RETREAT IF ICE IS TOO THIN                        TAK02770
*                                                                        TAK02780
          DO 209 M=1,LPLS                                                TAK02790
             DO 209 N=1,IWPLS                                            TAK02800
                IF(IGRID(M,N).EQ.0) THEN                                 TAK02810
                  EDIFF=NETA(M,N)-H(M,N)                                 TAK02820
                  IF( ABS(EDIFF).LE. 5.0) THEN                           TAK02830
                    IF(I.LE.40) THEN                                     TAK02840
                    WRITE(3,*) '** SURFACE FALLS BELOW BED LEVEL**',J    TAK02850
                    WRITE(3,*) 'M,N,H,E:',M,N,H(M,N),NETA(M,N)           TAK02860
                     ENDIF                                               TAK02870
                     IGRID(M,N)=2                                        TAK02880
                     ETA(M,N)=H(M,N)                                     TAK02890
                     NETA(M,N)=H(M,N)                                    TAK02900
                  ENDIF                                                  TAK02910
                ENDIF                                                    TAK02920
*                                                                        TAK02930
                IF(IGRID(M,N).EQ.2) THEN                                 TAK02940
                   IF(SSS(M,N).GT.0.0) THEN                              TAK02950
                     ETA(M,N)=ETA(M,N)+SSS(M,N)                          TAK02960
                     NETA(M,N)=NETA(M,N)+SSS(M,N)                        TAK02970
                     EDIFF=NETA(M,N)-H(M,N)                              TAK02980
```

```
                    IF(EDIFF.GE. 5.5) THEN                              TAK02990
                       IGRID(M,N)=0                                     TAK03000
                    ENDIF                                               TAK03010
                  ENDIF                                                 TAK03020
                ENDIF                                                   TAK03030
 209          CONTINUE                                                  TAK03040
*                                                                       TAK03050
*****.........ADVANCE OF RETREAT OF GLACIER                             TAK03060
*                                                                       TAK03070
          IF(MOVE) THEN                                                 TAK03080
             CALL MOVEIN(J,SSS,DXDY)                                    TAK03090
          ENDIF                                                         TAK03100
*                                                                       TAK03110
*****.........WRITE OUTPUT AT SELECTED TIME INTERVAL                    TAK03120
*                                                                       TAK03130
          CNT=CNT+1.0                                                   TAK03140
          IF(TIME.LT.START) GOTO 900                                    TAK03150
          ICALL=MOD(J,120)                                              TAK03160
          IF(ICALL.EQ.0) THEN                                           TAK03170
            CALL OUTPUT(J,HI,SSS,HEQL)                                  TAK03180
          ENDIF                                                         TAK03190
*                                                                       TAK03200
*****......WRITE OUTPUT FOR HOT START                                   TAK03210
*                                                                       TAK03220
          IF(J.EQ.NHOTST.OR.J.EQ.NFIN) THEN                             TAK03230
              CALL OUPT(J,HI,SSS)                                       TAK03240
          ENDIF                                                         TAK03250
 900      CONTINUE                                                      TAK03260
       WRITE(3,1005)                                                    TAK03270
 1005  FORMAT(' ',T10,'END TIME LOOP.........')                         TAK03280
*                                                                       TAK03290
       STOP                                                             TAK03300
       END                                                              TAK03310
```

```
      SUBROUTINE INPUT                                                  TAK03320
*****************************************************************     TAK03330
*                                                               *     TAK03340
*     READS INPUT DATA AND MAKES NECESSARY CONVERSIONS          *     TAK03350
*                                                               *     TAK03360
*****************************************************************     TAK03370
*                                                                     TAK03380
*                                                                     TAK03390
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK03400
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK03410
     &           IWPLS=IWIDTH+2)                                      TAK03420
*                                                                     TAK03430
      CHARACTER*1 T/Z05/                                              TAK03440
      REAL   NV,NH,NC,ND,NETA                                         TAK03450
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                TAK03460
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE,CLIMET,PRT          TAK03470
*                                                                     TAK03480
      DIMENSION HH(LPLS,IWPLS)                                        TAK03490
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK03500
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,          TAK03510
     &              ISTRT,NFIN,NHOTST                                 TAK03520
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE,CLIMET              TAK03530
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK03540
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)             TAK03550
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK03560
      COMMON/ BLK11/ HEQL,DELC,GRADC,GRADT,ALFA1,ALFA2,TDEL,SMIN      TAK03570
*                                                                     TAK03580
      DATA FACT/3.17097E-8/                                           TAK03590
*                                                                     TAK03600
      NAMELIST /LOGCL/HOTST,FTTOM,HVISC,VFRICT,MOVE,CLIMET,PRT        TAK03610
      NAMELIST /SYSTEM/H,ETA,SOS,FMAX,FMIN,FRICT,NV,NH,RHO            TAK03620
      NAMELIST /CNTRL/START,PHOTST,POUTPT,NHOTST                      TAK03630
      NAMELIST /MODPAR/DELT,DELX,DELY,NFIN,UPAI                       TAK03640
      NAMELIST /METEO/ HEQL,DELC,GRADC,GRADT,ALFA1,ALFA2,TDEL,SMIN    TAK03650
*                                                                     TAK03660
*                                                                     TAK03670
*****.....READ AND ECHO INPUT DATA                                    TAK03680
*                                                                     TAK03690
        WRITE(3,2001)                                                 TAK03700
 2001   FORMAT(' ',T10,'READING INITIAL INPUT DATA.....')             TAK03710
*                                                                     TAK03720
      READ(1,LOGCL)                                                   TAK03730
      IF(PRT) THEN                                                    TAK03740
          WRITE(3,LOGCL)                                              TAK03750
      ENDIF                                                           TAK03760
*                                                                     TAK03770
      READ(1,SYSTEM)                                                  TAK03780
      IF(PRT) THEN                                                    TAK03790
          WRITE(3,SYSTEM)                                             TAK03800
       ENDIF                                                          TAK03810
*                                                                     TAK03820
      READ(1,CNTRL)                                                   TAK03830
      IF(PRT) THEN                                                    TAK03840
          WRITE(3,CNTRL)                                              TAK03850
      ENDIF                                                           TAK03860
*                                                                     TAK03870
      READ(1,MODPAR)                                                  TAK03880
      IF(PRT) THEN                                                    TAK03890
          WRITE(3,MODPAR)                                             TAK03900
```

```
      ENDIF                                                              TAK03910
*                                                                        TAK03920
      IF(CLIMET) THEN                                                    TAK03930
         READ(1,METEO)                                                   TAK03940
      ENDIF                                                              TAK03950
      IF(PRT) THEN                                                       TAK03960
          WRITE(3,METEO)                                                 TAK03970
      ENDIF                                                              TAK03980
*                                                                        TAK03990
      DELT=86400.0*DELT                                                  TAK04000
      START=86400.0*START                                                TAK04010
      DELX=1000.0*DELX                                                   TAK04020
      DELY=1000.0*DELY                                                   TAK04030
*                                                                        TAK04040
*                                                                        TAK04050
*****.....CONVERT ELEVATIONS TO METER FROM FEET                          TAK04060
*                                                                        TAK04070
      IF(FTTOM) THEN                                                     TAK04080
         DO 102 I=1,LPLS                                                 TAK04090
            DO 102 J=1,IWPLS                                             TAK04100
              H(I,J)=0.3048*H(I,J)                                       TAK04110
              ETA(I,J)=0.3048*ETA(I,J)                                   TAK04120
102      CONTINUE                                                        TAK04130
      ENDIF                                                              TAK04140
*                                                                        TAK04150
***.....CONVERT PPT. FROM MM/YR TO M/S                                   TAK04160
*                                                                        TAK04170
         SMIN=FACT*SMIN*DELT                                             TAK04180
         DO 103 I=1,LPLS                                                 TAK04190
            DO 103 J=1,IWPLS                                             TAK04200
               SOS(I,J)=FACT*SOS(I,J)*DELT                               TAK04210
103      CONTINUE                                                        TAK04220
*                                                                        TAK04230
*                                                                        TAK04240
        IF(VFRICT) THEN                                                  TAK04250
           HMAX=-9999.0                                                  TAK04260
           HMIN=+9999.0                                                  TAK04270
           DO 104 I=1,LPLS                                               TAK04280
              DO 104 J=1,IWPLS                                           TAK04290
                 HMAX=AMAX1(HMAX,H(I,J))                                 TAK04300
                 HMIN=AMIN1(HMIN,H(I,J))                                 TAK04310
104        CONTINUE                                                      TAK04320
           HDIF=HMAX-HMIN                                                TAK04330
           HDIF=1.0/HDIF                                                 TAK04340
           FDIF=FMAX-FMIN                                                TAK04350
           DO 105 J=1,IWPLS                                              TAK04360
             DO 105 I=1,LPLS                                             TAK04370
                FRICT(I,J)= FMIN+(H(I,J)-HMIN)*FDIF*HDIF                 TAK04380
105        CONTINUE                                                      TAK04440
         ENDIF                                                           TAK04450
         DO 106 I=1,LPLS                                                 TAK04460
            DO 106 J=1,IWPLS                                             TAK04470
               FRICT(I,J)=FRICT(I,J)*1.0E06                              TAK04480
106      CONTINUE                                                        TAK04490
*                                                                        TAK04500
      RETURN                                                             TAK04510
      END                                                                TAK04520
```

```
      SUBROUTINE INITIL(HI,SSS)                                          TAK04530
*********************************************************************    TAK04540
*                                                                   *    TAK04550
*     INITIALIZES AND SETS COUNTERS FOR MODEL RUN                   *    TAK04560
*                                                                   *    TAK04570
*********************************************************************    TAK04580
*                                                                        TAK04590
*                                                                        TAK04600
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                         TAK04610
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,           TAK04620
     &           IWPLS=IWIDTH+2)                                         TAK04630
*                                                                       TAK04640
      REAL    NV,NH,NC,ND,NETA                                           TAK04650
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                   TAK04660
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                        TAK04670
*                                                                       TAK04680
      DIMENSION HI(LPLS,IWPLS),SSS(LPLS,IWPLS)                           TAK04690
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,             TAK04700
     &              ISTRT,NFIN,NHOTST                                    TAK04710
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                        TAK04720
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),              TAK04730
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)             TAK04740
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS),    TAK04750
     &              H(LPLS,IWPLS)                                        TAK04760
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)                TAK04770
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),        TAK04780
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),        TAK04790
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),     TAK04800
     &              INTG7(JP,JP,LPLS,IWPLS)                              TAK04810
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                         TAK04820
      COMMON/ BLK9/ VX(KP,LPLS,IWPLS),VY(KP,LPLS,IWPLS),                 TAK04830
     &              VV(KP,LPLS,IWPLS),VM(LPLS,IWPLS),DELSIG              TAK04840
      COMMON/ BLK10/ QIF(LPLS,IWPLS),QIB(LPLS,IWPLS),                    TAK04850
     &              QJF(LPLS,IWPLS),QJB(LPLS,IWPLS)                      TAK04860
*                                                                       TAK04870
*                                                                       TAK04880
*                                                                       TAK04890
      PI=3.1415927                                                       TAK04900
*                                                                       TAK04910
*                                                                       TAK04920
*****.....INITIALIZE C'S AND D'S                                        TAK04930
*                                                                       TAK04940
      DO 102 J=1,JP                                                      TAK04950
           DO 103 K=1,LENP1                                              TAK04960
                DO 103 L=1,IWIDTH                                        TAK04970
                     C(J,K,L)=0.0                                        TAK04980
                     NC(J,K,L)=0.0                                       TAK04990
103        CONTINUE                                                      TAK05000
           DO 104 K=1,LENGTH                                             TAK05010
                DO 104 L=1,IWIDP1                                        TAK05020
                     D(J,K,L)=0.0                                        TAK05030
                     ND(J,K,L)=0.0                                       TAK05040
104        CONTINUE                                                      TAK05050
102   CONTINUE                                                           TAK05060
*                                                                       TAK05070
*.............. INITIALIZATION OF AI'S                                  TAK05080
*                                                                       TAK05090
      DO 106 K=1,LPLS                                                    TAK05100
           DO 106 L=1,IWPLS                                              TAK05110
```

```
                DO 107 J=1,JP                                          TAK05120
                     A(J,K,L)=FLOAT(J-1)*PI+0.5*PI                     TAK05130
107             CONTINUE                                               TAK05140
106       CONTINUE                                                     TAK05150
*                                                                      TAK05160
*****.........INITIALIZE NETA                                          TAK05170
*                                                                      TAK05180
          DO 108 I=1,LPLS                                              TAK05190
               DO 108 J=1,IWPLS                                        TAK05200
                    NETA(I,J)=ETA(I,J)                                 TAK05210
108       CONTINUE                                                     TAK05220
*                                                                      TAK05230
****.............INITIALIZE INTEGRALS                                 TAK05240
*                                                                      TAK05250
          DO 109 I=1,LPLS                                              TAK05260
               DO 109 J=1,IWPLS                                        TAK05270
                    DO 109 K=1,JP                                      TAK05280
                         INTG1(K,I,J)=0.0                              TAK05290
                         INTG3(K,I,J)=0.0                              TAK05300
                         DO 109 L=1,JP                                 TAK05310
                              INTG2(L,J,I,J)=0.0                       TAK05320
                              INTG4(L,J,I,J)=0.0                       TAK05330
                              INTG5(L,J,I,J)=0.0                       TAK05340
                              INTG6(L,J,I,J)=0.0                       TAK05350
                              INTG7(L,J,I,J)=0.0                       TAK05360
 109       CONTINUE                                                    TAK05370
*                                                                      TAK05380
*****...........INTITALIZE INFLOW/OUTFLOW VALUES                       TAK05390
*                                                                      TAK05400
          DO 110 I=1,LPLS                                              TAK05410
               DO 110 J=1,IWPLS                                        TAK05420
                    QIF(I,J)=0.0                                       TAK05430
                    QIB(I,J)=0.0                                       TAK05440
                    QJF(I,J)=0.0                                       TAK05450
                    QJB(I,J)=0.0                                       TAK05460
110       CONTINUE                                                     TAK05470
*                                                                      TAK05480
          DO 111 I=1,LPLS                                              TAK05490
               DO 111 J=1,IWPLS                                        TAK05500
                     HI(I,J)=ETA(I,J)                                  TAK05510
                     SSS(I,J)=SOS(I,J)                                 TAK05520
111       CONTINUE                                                     TAK05530
*                                                                      TAK05540
*****.........INITIALIZE VELOCITY                                      TAK05550
*                                                                      TAK05560
          DELSIG=2.0/(KP-1)                                            TAK05570
          DO 112 I=1,LPLS                                              TAK05580
             DO 112 J=1,IWPLS                                          TAK05590
                 VM(I,J)=0.0                                           TAK05600
                 DO 112 K=1,KP                                         TAK05610
                    VX(K,I,J)=0.0                                      TAK05620
                    VY(K,I,J)=0.0                                      TAK05630
                    VV(K,I,J)=0.0                                      TAK05640
 112      CONTINUE                                                     TAK05650
*                                                                      TAK05660
*****.......SET LOCATION & WEIGHTS OF GAUSSIAN QUADRATURE              TAK05670
*                                                                      TAK05680
          XI(1)=-0.8611363116                                          TAK05690
          XI(2)=-0.3399810436                                          TAK05700
          XI(3)=-XI(2)                                                 TAK05710
          XI(4)=-XI(1)                                                 TAK05720
```

```
*                                                                       TAK05730
          W(1)=0.3478548451                                             TAK05740
          W(2)=0.6521451549                                             TAK05750
          W(3)=W(2)                                                     TAK05760
          W(4)=W(1)                                                     TAK05770
*                                                                       TAK05780
       RETURN                                                           TAK05790
*                                                                       TAK05800
*****.....READ INITIAL CONDITION FOR HOT START                          TAK05810
*                                                                       TAK05820
       ENTRY INIT(HI,SSS)                                               TAK05830
*                                                                       TAK05840
         WRITE(3,2001)                                                  TAK05850
 2001    FORMAT(' ',T10,'READING HOT START DATA.......')                TAK05860
          READ(2) ISTRT                                                 TAK05870
          READ(2) IGRID                                                 TAK05880
          READ(2) A                                                     TAK05890
          READ(2) ETA                                                   TAK05900
          READ(2) NETA                                                  TAK05910
          READ(2) C                                                     TAK05920
          READ(2) NC                                                    TAK05930
          READ(2) D                                                     TAK05940
          READ(2) ND                                                    TAK05950
          READ(2) SSS                                                   TAK05960
          READ(2) FRICT                                                 TAK05970
          READ(2) QIF                                                   TAK05980
          READ(2) QIB                                                   TAK05990
          READ(2) QJF                                                   TAK06000
          READ(2) QJB                                                   TAK06010
*                                                                       TAK06020
       RETURN                                                           TAK06030
       END                                                              TAK06040
```

```
      SUBROUTINE FILTER(XP,IG)                                          TAK06050
*****************************************************************     TAK06060
*                                                               *     TAK06070
*            FILTERS INPUT DATA IF NECESSARY                    *     TAK06080
*                                                               *     TAK06090
*****************************************************************     TAK06100
*                                                                     TAK06110
*                                                                     TAK06120
      PARAMETER (LENGTH=54,IWIDTH=33)                                 TAK06130
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK06140
     &           IWPLS=IWIDTH+2)                                      TAK06150
      DIMENSION XP(LPLS,IWPLS),IG(LPLS,IWPLS),YP(LPLS,IWPLS)          TAK06160
*                                                                     TAK06170
*                                                                     TAK06180
       DO 101 J=1,IWPLS                                               TAK06190
          DO 101 I=1,LPLS                                             TAK06200
             YP(I,J)=XP(I,J)                                          TAK06210
101    CONTINUE                                                       TAK06220
       DO 102 I=1,LPLS                                                TAK06230
          DO 102 J=1,IWPLS                                            TAK06240
             IF(IG(I,J).EQ.0) THEN                                    TAK06250
               IF(IG(I-1,J).EQ.0.AND.IG(I+1,J).EQ.0)THEN              TAK06260
                  FX=0.25*(YP(I-1,J)+2.0*YP(I,J)+YP(I+1,J))           TAK06270
               ELSE IF(IG(I-1,J).EQ.0) THEN                           TAK06280
                  FX=0.33*(YP(I-1,J)+2.0*YP(I,J))                     TAK06290
               ELSE IF(IG(I+1,J).EQ.0) THEN                           TAK06300
                  FX=0.33*(2.0*YP(I,J)+YP(I+1,J))                     TAK06310
               ELSE                                                   TAK06320
                  FX=YP(I,J)                                          TAK06330
               ENDIF                                                  TAK06340
*                                                                     TAK06350
               IF(IG(I,J-1).EQ.0.AND.IG(I,J+1).EQ.0)THEN              TAK06360
                  FY=0.25*(YP(I,J-1)+2.0*YP(I,J)+YP(I,J+1))           TAK06370
               ELSE IF(IG(I,J-1).EQ.0) THEN                           TAK06380
                  FY=0.33*(YP(I,J-1)+2.0*YP(I,J))                     TAK06390
               ELSE IF(IG(I,J+1).EQ.0) THEN                           TAK06400
                  FY=0.33*(2.0*YP(I,J)+YP(I,J+1))                     TAK06410
               ELSE                                                   TAK06420
                  FY=YP(I,J)                                          TAK06430
               ENDIF                                                  TAK06440
               XP(I,J)=0.5*(FX+FY)                                    TAK06450
             ENDIF                                                    TAK06460
102       CONTINUE                                                    TAK06470
          RETURN                                                      TAK06480
          END                                                         TAK06490
```

```
      SUBROUTINE ACALC                                                  TAK06500
*****************************************************************     TAK06510
*                                                               *     TAK06520
*          CALCULATES VALUES OF A(I)'S                          *     TAK06530
*                                                               *     TAK06540
*****************************************************************     TAK06550
*                                                                     TAK06560
*                                                                     TAK06570
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK06580
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK06590
     &           IWPLS=IWIDTH+2)                                      TAK06600
*                                                                     TAK06610
      REAL    NV,NH,NC,ND,NETA                                        TAK06620
      LOGICAL FLAG                                                    TAK06630
*                                                                     TAK06640
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK06650
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK06660
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)             TAK06670
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK06680
*                                                                     TAK06690
      DATA   PI/    3.1515927/                                        TAK06700
*                                                                     TAK06710
*                                                                     TAK06720
*                                                                     TAK06730
        DO 101 M=1,LPLS                                               TAK06740
           DO 101 N=1,IWPLS                                           TAK06750
              IF(IGRID(M,N).NE.0) GOTO 101                            TAK06760
                 IF(FRICT(M,N).EQ.0.0) THEN                           TAK06770
                    GOTO 501                                          TAK06780
                 ELSE                                                 TAK06790
                    IF(FRICT(M,N).EQ.-999.0) THEN                     TAK06800
                     GOTO 502                                         TAK06810
                 ENDIF                                                TAK06820
              ENDIF                                                   TAK06830
              CB=FRICT(M,N)                                           TAK06840
              DP= NETA(M,N)-H(M,N)                                    TAK06850
              DPCB=DP*CB                                              TAK06860
              C=DPCB/NV                                               TAK06870
              IF(C.LT.0.0001) GOTO 501                                TAK06880
              IF(C.GT.250.0) GOTO 502                                 TAK06890
*                                                                     TAK06900
*****.....NEWTON-RAPSON ITERATION                                     TAK06910
*                                                                     TAK06920
              DO 102 J=1,JP                                           TAK06930
                 SOL=(((FLOAT(J)-1.0)*PI))+PI/8.0                     TAK06940
                 FLAG=.TRUE.                                          TAK06950
 503             AO=C/SOL                                             TAK06960
                 A(J,M,N)=(ATAN(AO)+(FLOAT(J-1))*PI)                  TAK06970
                 DIFF= ABS(A(J,M,N)-SOL)                              TAK06980
                 IF(DIFF.LT.0.01) FLAG=.FALSE.                        TAK06990
                 SOL=A(J,M,N)                                         TAK07000
                 IF(FLAG) GOTO 503                                    TAK07010
 102          CONTINUE                                                TAK07020
              GOTO 101                                                TAK07030
 501          DO 104 J=1,JP                                           TAK07040
                 A(J,M,N)=FLOAT(J-1)*PI                               TAK07050
 104          CONTINUE                                                TAK07060
              GOTO 101                                                TAK07070
```

```
 502              DO 105 J=1,JP                                          TAK07080
                      A(J,M,N)=FLOAT(J-1)*PI+PI*0.5                      TAK07090
 105              CONTINUE                                               TAK07100
 101         CONTINUE                                                    TAK07110
*                                                                        TAK07120
       RETURN                                                           TAK07130
       END                                                              TAK07140
```

```
      SUBROUTINE GALIN1                                                  TAK07150
*****************************************************************      TAK07160
*                                                               *      TAK07170
*          EVALUATES 1, 2 AND 3 INTEGRALS                       *      TAK07180
*                                                               *      TAK07190
*****************************************************************      TAK07200
*                                                                      TAK07210
*                                                                      TAK07220
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                       TAK07230
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,         TAK07240
     &           IWPLS=IWIDTH+2)                                       TAK07250
*                                                                      TAK07260
      REAL   NV,NH,NC,ND,NETA                                          TAK07270
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                 TAK07280
*                                                                      TAK07290
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)              TAK07300
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),      TAK07310
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),      TAK07320
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),   TAK07330
     &              INTG7(JP,JP,LPLS,IWPLS)                            TAK07340
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                       TAK07350
*                                                                      TAK07360
*                                                                      TAK07370
*                                                                      TAK07380
*****.......FIRST INTEGRAL                                            TAK07390
*                                                                      TAK07400
      DO 101 I=1,LPLS                                                  TAK07410
         DO 101 J=1,IWPLS                                              TAK07420
            IF(IGRID(I,J).NE.0) GOTO 101                               TAK07430
            DO 102 K=1,JP                                              TAK07440
               IF(A(K,I,J).EQ.0.) THEN                                 TAK07450
                 INTG1(K,I,J)=2.0                                      TAK07460
               ELSE                                                    TAK07470
                 INTG1(K,I,J)=1.0+ SIN(2.0*A(K,I,J))/(2.0*A(K,I,J))    TAK07480
               ENDIF                                                   TAK07490
 102        CONTINUE                                                   TAK07500
 101  CONTINUE                                                         TAK07510
*                                                                      TAK07520
*****.........SECOND INTEGRAL                                         TAK07530
*                                                                      TAK07540
      DO 104 M=1,LPLS                                                  TAK07550
           DO 104 N=1,IWPLS                                            TAK07560
                IF(IGRID(M,N).NE.0) GOTO 104                           TAK07570
                DO 105 I=1,JP                                          TAK07580
                     IF(A(I,M,N).EQ.0.) THEN                           TAK07590
                         DO 106 J=1,JP                                 TAK07600
                              INTG2(I,J,M,N)=0.0                       TAK07610
 106                     CONTINUE                                      TAK07620
                         GOTO 502                                      TAK07630
                         ENDIF                                         TAK07640
                         DO 105 J=1,JP                                 TAK07650
                              F1=0.0                                   TAK07660
                              SI=0.5*A(I,M,N)                          TAK07670
                              SJ=0.5*A(J,M,N)                          TAK07680
                              DO 107 L=1,4                             TAK07690
                                   X=(1.0+XI(L))                       TAK07700
                                   T1=SI*(2.0-X)                       TAK07710
                                   T2=SJ*(2.0-X)                       TAK07720
                                   XNUB=X* SIN(T1)* COS(T2)*SI         TAK07730
                                   F1=F1+XNUB*W(L)                     TAK07740
 107                          CONTINUE                                 TAK07750
```

```
                              INTG2(I,J,M,N)=F1                                  TAK07760
 105                        CONTINUE                                             TAK07770
 502                  CONTINUE                                                   TAK07780
 104           CONTINUE                                                         TAK07790
*                                                                               TAK07800
*****.........THIRD INTEGRAL                                                    TAK07810
*                                                                               TAK07820
        DO 109 I=1,LPLS                                                         TAK07830
             DO 109 J=1,IWPLS                                                   TAK07840
                  IF(IGRID(I,J).NE.0) GOTO 109                                  TAK07850
                  DO 110 K=1,JP                                                 TAK07860
                       IF(A(K,I,J).EQ.0.) THEN                                  TAK07870
                          INTG3(K,I,J)=2.0                                      TAK07880
                       ELSE                                                     TAK07890
                          INTG3(K,I,J)=2.0* SIN(A(K,I,J))/(A(K,I,J))            TAK07900
                       ENDIF                                                    TAK07910
 110                 CONTINUE                                                   TAK07920
 109    CONTINUE                                                                TAK07930
*                                                                               TAK07940
       RETURN                                                                   TAK07950
       END                                                                      TAK07960
```

```
      SUBROUTINE GALIN2                                                 TAK07970
*****************************************************************     TAK07980
*                                                               *     TAK07990
*           EVALUATES INTEGRALS S4, S5, AND S6                  *     TAK08000
*                                                               *     TAK08010
*****************************************************************     TAK08020
*                                                                     TAK08030
*                                                                     TAK08040
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK08050
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK08060
     &           IWPLS=IWIDTH+2)                                      TAK08070
*                                                                     TAK08080
      REAL   NV,NH,NC,ND,NETA                                         TAK08090
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                TAK08100
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK08110
*                                                                     TAK08120
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK08130
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,          TAK08140
     &              ISTRT,NFIN,NHOTST                                 TAK08150
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK08160
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),           TAK08170
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)          TAK08180
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK08190
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)             TAK08200
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),     TAK08210
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),     TAK08220
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),  TAK08230
     &              INTG7(JP,JP,LPLS,IWPLS)                           TAK08240
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK08250
*                                                                     TAK08260
*                                                                     TAK08270
*                                                                     TAK08280
      DO 101 M=2,LPLS-1                                               TAK08290
         DO 101 N=2,IWPLS-1                                           TAK08300
*                                                                     TAK08310
          IF(IGRID(M,N).NE.0) GOTO 101                                TAK08320
*                                                                     TAK08330
*****.......COMPUTE DIFFERENTIALS IN X-DIRECTION                     TAK08340
*                                                                     TAK08350
          IF(IGRID(M+1,N).EQ.0.AND.IGRID(M-1,N).EQ.0) THEN            TAK08360
             DHDX=(H(M+1,N)-H(M-1,N))/(2.0*DELX)                      TAK08370
             DEDX=(NETA(M+1,N)-NETA(M-1,N))/(2.0*DELX)                TAK08380
             DDFX=NETA(M+1,N)-H(M+1,N)                                TAK08390
             DDMX=NETA(M,N)-H(M,N)                                    TAK08400
             DDBX=NETA(M-1,N)-H(M-1,N)                                TAK08410
             DDX =(DDFX+5.0*DDMX+DDBX)/7.0                            TAK08420
          ELSE IF(IGRID(M+1,N).EQ.0) THEN                             TAK08430
             DHDX=(H(M+1,N)-H(M,N))/ DELX                             TAK08440
             DEDX=(NETA(M+1,N)-NETA(M,N))/ DELX                       TAK08450
             DDFX=NETA(M+1,N)-H(M+1,N)                                TAK08460
             DDMX=NETA(M,N)-H(M,N)                                    TAK08470
             DDX =(DDFX+5.0*DDMX)/6.0                                 TAK08480
           ELSE IF(IGRID(M-1,N).EQ.0) THEN                            TAK08490
             DHDX=(H(M,N)-H(M-1,N))/ DELX                             TAK08500
             DEDX=(NETA(M,N)-NETA(M-1,N))/ DELX                       TAK08510
             DDMX=NETA(M,N)-H(M,N)                                    TAK08520
             DDBX=NETA(M-1,N)-H(M-1,N)                                TAK08530
             DDX =(5.0*DDMX+DDBX)/6.0                                 TAK08540
           ELSE                                                       TAK08550
             DHDX=0.0                                                 TAK08560
             DEDX=0.0                                                 TAK08570
```

```
            DDX =NETA(M,N)-H(M,N)                                          TAK08580
         ENDIF                                                             TAK08590
*                                                                          TAK08600
*****.....COMPUTE DIFFERENTIALS IN Y-DIRECTION                            TAK08610
*                                                                          TAK08620
         IF(IGRID(M,N+1).EQ.0.AND.IGRID(M,N-1).EQ.0) THEN                  TAK08630
            DHDY=(H(M,N+1)-H(M,N-1))/(2.0*DELY)                            TAK08640
            DEDY=(NETA(M,N+1)-NETA(M,N-1))/(2.0*DELY)                      TAK08650
            DDFY=NETA(M,N+1)-H(M,N+1)                                      TAK08660
            DDMY=NETA(M,N)-H(M,N)                                          TAK08670
            DDBY=NETA(M,N-1)-H(M,N-1)                                      TAK08680
            DDY =(DDFY+5.0*DDMY+DDBY)/7.0                                  TAK08690
         ELSE IF(IGRID(M,N+1).EQ.0) THEN                                   TAK08700
            DHDY=(H(M,N+1)-H(M,N))/ DELY                                   TAK08710
            DEDY=(NETA(M,N+1)-NETA(M,N))/ DELY                             TAK08720
            DDFY=NETA(M,N+1)-H(M,N+1)                                      TAK08730
            DDMY=NETA(M,N)-H(M,N)                                          TAK08740
            DDY =(DDFY+5.0*DDMY)/6.0                                       TAK08750
         ELSE IF(IGRID(M,N-1).EQ.0) THEN                                   TAK08760
            DHDY=(H(M,N)-H(M,N-1))/ DELY                                   TAK08770
            DEDY=(NETA(M,N)-NETA(M,N-1))/ DELY                             TAK08780
            DDMY=NETA(M,N)-H(M,N)                                          TAK08790
            DDBY=NETA(M,N-1)-H(M,N-1)                                      TAK08800
            DDY =(5.0*DDMY+DDBY)/6.0                                       TAK08810
         ELSE                                                              TAK08820
            DHDY=0.0                                                       TAK08830
            DEDY=0.0                                                       TAK08840
            DDY =NETA(M,N)-H(M,N)                                          TAK08850
         ENDIF                                                             TAK08860
         DD=(DDX+DDY)/2.0                                                  TAK08870
*                                                                          TAK08880
******.....EVALUATE THE INTEGRALS                                         TAK08890
*                                                                          TAK08900
         DO 102 I=1,JP                                                     TAK08910
            DO 105 J=1,JP                                                  TAK08920
               F1=0.0                                                      TAK08930
               F2=0.0                                                      TAK08940
               F3=0.0                                                      TAK08950
               SI=0.5*A(I,M,N)                                             TAK08960
               SJ=0.5*A(J,M,N)                                             TAK08970
               DO 104 L=1,4                                                TAK08980
                  X=(1.0+XI(L))                                            TAK08990
                  T1=SI*(2.0-X)                                            TAK09000
                  T2=SJ*(2.0-X)                                            TAK09010
                  XNUB=(SI)* SIN(T1)* COS(T1)* COS(T2)                     TAK09020
                  X1=((2.0-X)*DHDX-X*DEDX)/DD                              TAK09030
                  Y1=((2.0-X)*DHDY-X*DEDY)/DD                              TAK09040
                  F1=F1+(( COS(T1)* COS(T2))**2)*W(L)                      TAK09050
                  F2=F2+X1*XNUB*W(L)                                       TAK09060
                  F3=F3+Y1*XNUB*W(L)                                       TAK09070
104            CONTINUE                                                    TAK09080
               INTG4(I,J,M,N)=F1                                           TAK09090
               INTG5(I,J,M,N)=F2                                           TAK09100
               INTG6(I,J,M,N)=F3                                           TAK09110
105         CONTINUE                                                       TAK09120
102         CONTINUE                                                       TAK09130
101   CONTINUE                                                             TAK09140
      RETURN                                                               TAK09150
      END                                                                  TAK09160
```

```
      SUBROUTINE GALIN3                                                 TAK09170
*****************************************************************     TAK09180
*                                                               *     TAK09190
*           EVALUATES SUM OF INTEGRALS 7 TO 10                  *     TAK09200
*                                                               *     TAK09210
*****************************************************************     TAK09220
*                                                                     TAK09230
*                                                                     TAK09240
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK09250
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK09260
     &           IWPLS=IWIDTH+2)                                      TAK09270
*                                                                     TAK09280
      REAL   NV,NH,NC,ND,NETA                                         TAK09290
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                TAK09300
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK09310
*                                                                     TAK09320
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK09330
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,          TAK09340
     &              ISTRT,NFIN,NHOTST                                 TAK09350
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK09360
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK09370
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)             TAK09380
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),     TAK09390
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),     TAK09400
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),  TAK09410
     &              INTG7(JP,JP,LPLS,IWPLS)                           TAK09420
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK09430
*                                                                     TAK09440
*                                                                     TAK09450
*                                                                     TAK09460
        DO 101 M=2,LPLS-1                                             TAK09470
          DO 101 N=2,IWPLS-1                                          TAK09480
*                                                                     TAK09490
          IF(IGRID(M,N).NE.0) GOTO 101                                TAK09500
*                                                                     TAK09510
*****.......COMPUTE DIFFERENTIALS IN X-DIRECTION                     TAK09520
*                                                                     TAK09530
          IF(IGRID(M+1,N).EQ.0.AND.IGRID(M-1,N).EQ.0) THEN            TAK09540
            DHDX=(H(M+1,N)-H(M-1,N))/(2.0*DELX)                       TAK09550
            DEDX=(NETA(M+1,N)-NETA(M-1,N))/(2.0*DELX)                 TAK09560
            D2HDX=((H(M+1,N)-H(M,N))+(H(M-1,N)-H(M,N)))/(DELX**2)     TAK09570
            D2EDX=((NETA(M+1,N)-NETA(M,N))+(NETA(M-1,N)-NETA(M,N)))   TAK09580
     &                                                 /(DELX**2)     TAK09590
            DDFX=NETA(M+1,N)-H(M+1,N)                                 TAK09600
            DDMX=NETA(M,N)-H(M,N)                                     TAK09610
            DDBX=NETA(M-1,N)-H(M-1,N)                                 TAK09620
            DDX =(DDFX+5.0*DDMX+DDBX)/7.0                             TAK09630
          ELSE IF(IGRID(M+1,N).EQ.0) THEN                             TAK09640
            DHDX=(H(M+1,N)-H(M,N))/ DELX                              TAK09650
            DEDX=(NETA(M+1,N)-NETA(M,N))/ DELX                        TAK09660
            D2HDX=(H(M+1,N)-H(M,N))/(DELX**2)                         TAK09670
            D2EDX=(NETA(M+1,N)-NETA(M,N))/(DELX**2)                   TAK09680
            DDFX=NETA(M+1,N)-H(M+1,N)                                 TAK09690
            DDMX=NETA(M,N)-H(M,N)                                     TAK09700
            DDX =(DDFX+5.0*DDMX)/6.0                                  TAK09710
          ELSE IF(IGRID(M-1,N).EQ.0) THEN                             TAK09720
```

```
                DHDX=(H(M,N)-H(M-1,N))/ DELX                                    TAK09730
                DEDX=(NETA(M,N)-NETA(M-1,N))/ DELX                              TAK09740
                D2HDX=(H(M-1,N)-H(M,N))/(DELX**2)                               TAK09750
                D2EDX=(NETA(M-1,N)-NETA(M,N))/(DELX**2)                         TAK09760
                DDMX=NETA(M,N)-H(M,N)                                           TAK09770
                DDBX=NETA(M-1,N)-H(M-1,N)                                       TAK09780
                DDX =(5.0*DDMX+DDBX)/6.0                                        TAK09790
             ELSE                                                               TAK09800
                DHDX=0.0                                                        TAK09810
                DEDX=0.0                                                        TAK09820
                D2HDX=0.0                                                       TAK09830
                D2EDX=0.0                                                       TAK09840
                DDX =NETA(M,N)-H(M,N)                                           TAK09850
             ENDIF                                                              TAK09860
*                                                                               TAK09870
*****.........COMPUTE DIFFERENTIALS IN Y-DIRECTION                              TAK09880
*                                                                               TAK09890
           IF(IGRID(M,N+1).EQ.0.AND.IGRID(M,N-1).EQ.0) THEN                     TAK09900
              DHDY=(H(M,N+1)-H(M,N-1))/(2.0*DELY)                               TAK09910
              DEDY=(NETA(M,N+1)-NETA(M,N-1))/(2.0*DELY)                         TAK09920
              D2HDY=((H(M,N+1)-H(M,N))+(H(M,N-1)-H(M,N)))/(DELY**2)             TAK09930
              D2EDY=((NETA(M,N+1)-NETA(M,N))+(NETA(M,N-1)-NETA(M,N)))           TAK09940
     &                                                          /(DELY**2)      TAK09950
              DDFY=NETA(M,N+1)-H(M,N+1)                                         TAK09960
              DDMY=NETA(M,N)-H(M,N)                                             TAK09970
              DDBY=NETA(M,N-1)-H(M,N-1)                                         TAK09980
              DDY =(DDFY+5.0*DDMY+DDBY)/7.0                                     TAK09990
           ELSE IF(IGRID(M,N+1).EQ.0) THEN                                      TAK10000
              DHDY=(H(M,N+1)-H(M,N))/ DELY                                      TAK10010
              DEDY=(NETA(M,N+1)-NETA(M,N))/ DELY                                TAK10020
              D2HDY=(H(M,N+1)-H(M,N))/(DELY**2)                                 TAK10030
              D2EDY=(NETA(M,N+1)-NETA(M,N))/(DELY**2)                           TAK10040
              DDFY=NETA(M,N+1)-H(M,N+1)                                         TAK10050
              DDMY=NETA(M,N)-H(M,N)                                             TAK10060
              DDY =(DDFY+5.0*DDMY)/6.0                                          TAK10070
           ELSE IF(IGRID(M,N-1).EQ.0) THEN                                      TAK10080
              DHDY=(H(M,N)-H(M,N-1))/ DELY                                      TAK10090
              DEDY=(NETA(M,N)-NETA(M,N-1))/ DELY                                TAK10100
              D2HDY=(H(M,N-1)-H(M,N))/(DELY**2)                                 TAK10110
              D2EDY=(NETA(M,N-1)-NETA(M,N))/(DELY**2)                           TAK10120
              DDMY=NETA(M,N)-H(M,N)                                             TAK10130
              DDBY=NETA(M,N-1)-H(M,N-1)                                         TAK10140
              DDY =(5.0*DDMY+DDBY)/6.0                                          TAK10150
           ELSE                                                                 TAK10160
              DHDY=0.0                                                          TAK10170
              DEDY=0.0                                                          TAK10180
              D2HDY=0.0                                                         TAK10190
              D2EDY=0.0                                                         TAK10200
              DDY =NETA(M,N)-H(M,N)                                             TAK10210
           ENDIF                                                                TAK10220
           DD=(DDX+DDY)/2.0                                                     TAK10230
*                                                                               TAK10240
******......EVALUATE INTEGRALS AND SUM                                          TAK10250
*                                                                               TAK10260
         DO 102 I=1,JP                                                          TAK10270
            DO 102 J=1,JP                                                       TAK10280
               F1=0.0                                                           TAK10290
               F2=0.0                                                           TAK10300
               F3=0.0                                                           TAK10310
               F4=0.0                                                           TAK10320
               SI=0.5*A(I,M,N)                                                  TAK10330
```

```
                SJ=0.5*A(J,M,N)                                                  TAK10340
                DO 103 L=1,4                                                     TAK10350
                   X=(1.0+XI(L))                                                 TAK10360
                   T1=SI*(2.0-X)                                                 TAK10370
                   T2=SJ*(2.0-X)                                                 TAK10380
                   GAMX=((2.0-X)*DHDX-X*DEDX)/DD                                 TAK10390
                   GAMY=((2.0-X)*DHDY-X*DEDY)/DD                                 TAK10400
                   GX=((2.0-X)*D2HDX-X*D2EDX-GAMX*(DHDX+DEDX+1.0))/DD            TAK10410
                   GY=((2.0-X)*D2HDY-X*D2EDY-GAMY*(DHDY+DEDY+1.0))/DD            TAK10420
                   F1=F1+GX*A(I,M,N)*0.5* SIN(T1)* COS(T2)*W(L)                  TAK10430
                   F2=F2+(GAMX**2)*(-(A(I,M,N)**2)*0.25)*                        TAK10440
     &                 COS(T1)* COS(T2)*W(L)                                     TAK10450
                   F3=F3+GY*A(I,M,N)*0.5* SIN(T1)* COS(T2)*W(L)                  TAK10460
                   F4=F4+(GAMY**2)*(-(A(I,M,N)**2)*0.25)*                        TAK10470
     &                 COS(T1)* COS(T2)*W(L)                                     TAK10480
 103            CONTINUE                                                         TAK10490
                IF(A(I,M,N).EQ.0.) F1=0.0                                        TAK10500
                IF(A(I,M,N).EQ.0.) F3=0.0                                        TAK10510
                INTG7(I,J,M,N)=F1+F2+F3+F4                                       TAK10520
 102         CONTINUE                                                            TAK10530
*                                                                                TAK10540
 101      CONTINUE                                                               TAK10550
          RETURN                                                                 TAK10560
          END                                                                    TAK10570
```

```
      SUBROUTINE XMOM
****************************************************************
*                                                              *
*        SOLVES THE X-MOMENTUM EQUATION                        *
*                                                              *
****************************************************************
*
*
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,
     &           IWPLS=IWIDTH+2)
*
      REAL    NV,NH,NC,ND,NETA
      REAL    INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7
      LOGICAL        FTTOM,HOTST,HVISC,VFRICT,MOVE
*
      DIMENSION  AA(JP),AI1(JP),AI2R(JP),AI2L(JP),AI3(JP),AI5R(JP),
     &           AI5L(JP),AI6L(JP),AI6R(JP),AI7R(JP),AI7L(JP)
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,
     &              ISTRT,NFIN,NHOTST
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),
     &              INTG7(JP,JP,LPLS,IWPLS)
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)
*
      DATA   G/   9.81  /
*
*
*
       DO 101 N=1,IWIDTH
         NX=N+1
         DO 101 I=1,LENP1
           IX=I+1
           IF(IGRID(I,NX).EQ.1) GOTO 101
           IF(IGRID(IX,NX).EQ.1) GOTO 101
           IF(IGRID(I,NX).EQ.2.AND.IGRID(IX,NX).EQ.2) GOTO 101
*
*****.......TERMS INVOLVING VERTICAL VISCOSITY
*
           IF(IGRID(I,NX).EQ.0.AND.IGRID(IX,NX).EQ.0) THEN
             AVGNX=0.5*(NETA(I,NX)+NETA(IX,NX))
             AVGEX=0.5*(ETA(I,NX)+ETA(IX,NX))
             AVGHX=0.5*(H(I,NX)+H(IX,NX))
             ETAX =(NETA(IX,NX)-NETA(I,NX))/DELX
           ELSE IF(IGRID(I,NX).EQ.0) THEN
             AVGNX=  NETA(I,NX)
             AVGEX=  ETA(I,NX)
             AVGHX=  H(I,NX)
             ETAX =(NETA(I,NX)-NETA(I-1,NX))/DELX
           ELSE
             AVGNX= NETA(IX,NX)
```

```
            AVGEX= ETA(IX,NX)                                          TAK11160
            AVGHX= H(IX,NX)                                            TAK11170
            ETAX =(NETA(IX+1,NX)-NETA(IX,NX))/DELX                     TAK11180
          ENDIF                                                        TAK11190
          AVGD =AVGNX-AVGHX                                            TAK11200
          ETAT =(AVGNX-AVGEX)/DELT                                     TAK11210
*                                                                      TAK11220
*****.....TERMS INVOLVING HORIZONTAL VISCOSITY                        TAK11230
*                                                                      TAK11240
        IF(HVISC) THEN                                                 TAK11250
          IF(IGRID(I,NX).EQ.0.AND.IGRID(IX,NX).EQ.0) THEN              TAK11260
            DELEX=(NETA(IX,NX)-NETA(I,NX))/DELX                        TAK11270
            DELHX=(H(IX,NX)-H(I,NX))/DELX                              TAK11280
          ELSE IF(IGRID(I,NX).EQ.0) THEN                               TAK11290
            DELEX=(NETA(I,NX)-NETA(I-1,NX))/DELX                       TAK11300
            DELHX=(H(I,NX)-H(I-1,NX))/DELX                             TAK11310
          ELSE                                                         TAK11320
            DELEX=(NETA(IX+1,NX)-NETA(IX,NX))/DELX                     TAK11330
            DELHX=(H(IX+1,NX)-H(IX,NX))/DELX                           TAK11340
          ENDIF                                                        TAK11350
          DELDX=DELEX-DELHX                                            TAK11360
*                                                                      TAK11370
            IF(IGRID(I,N).EQ.0.AND.IGRID(I,NX+1).EQ.0) THEN            TAK11380
              DELE1=(NETA(I,NX+1)-NETA(I,N))/(2.0*DELY)                TAK11390
              DELH1=(H(I,NX+1)-H(I,N))/(2.0*DELY)                      TAK11400
            ELSE IF(IGRID(I,NX+1).EQ.0) THEN                           TAK11410
              DELE1=(NETA(I,NX+1)-NETA(I,NX))/DELY                     TAK11420
              DELH1=(H(I,NX+1)-H(I,NX))/DELY                           TAK11430
            ELSE IF(IGRID(I,NX).EQ.0) THEN                             TAK11440
              DELE1=(NETA(I,NX)-NETA(I,N))/DELY                        TAK11450
              DELH1=(H(I,NX)-H(I,N))/DELY                              TAK11460
            ELSE                                                       TAK11470
              DELE1=0.0                                                TAK11480
              DELH1=0.0                                                TAK11490
            ENDIF                                                      TAK11500
*                                                                      TAK11510
            IF(IGRID(IX,NX+1).EQ.0.AND.IGRID(IX,N).EQ.0) THEN          TAK11520
              DELE2=(NETA(IX,NX+1)-NETA(IX,N))/(2.0*DELY)              TAK11530
              DELH2=(H(IX,NX+1)-H(IX,N))/(2.0*DELY)                    TAK11540
            ELSE IF(IGRID(IX,NX+1).EQ.0) THEN                          TAK11550
              DELE2=(NETA(IX,NX+1)-NETA(IX,NX))/DELY                   TAK11560
              DELH2=(H(IX,NX+1)-H(IX,NX))/DELY                         TAK11570
            ELSE IF(IGRID(IX,N).EQ.0) THEN                             TAK11580
              DELE2=(NETA(IX,NX)-NETA(IX,N))/DELY                      TAK11590
              DELH2=(H(IX,NX)-H(IX,N))/DELY                            TAK11600
            ELSE                                                       TAK11610
              DELE2=0.0                                                TAK11620
              DELH2=0.0                                                TAK11630
            ENDIF                                                      TAK11640
            DELEY=0.5*(DELE1+DELE2)                                    TAK11650
            DELHY=0.5*(DELH1+DELH2)                                    TAK11660
            DELDY=DELEY-DELHY                                          TAK11670
*                                                                      TAK11680
            IG1=IGRID(I,NX+1)                                          TAK11690
            IG2=IGRID(IX,NX+1)                                         TAK11700
            IG3=IGRID(I,N)                                             TAK11710
            IG4=IGRID(IX,N)                                            TAK11720
        ENDIF                                                          TAK11730
*                                                                      TAK11740
*****.....TERMS INVOLVING VERTICAL VISCOSITY                          TAK11750
*                                                                      TAK11760
```

```
      DO 102 K=1,JP                                                    TAK11770
        AA(K)=0.0                                                      TAK11780
        AI1(K)=0.0                                                     TAK11790
        AI3(K)=0.0                                                     TAK11800
        IF(IGRID(I,NX).EQ.0.AND.IGRID(IX,NX).EQ.0) THEN                TAK11810
           AA(K)=0.5*(A(K,I,NX)+A(K,IX,NX))                            TAK11820
           AI1(K)=0.5*(INTG1(K,I,NX)+INTG1(K,IX,NX))                   TAK11830
           AI3(K)=0.5*(INTG3(K,I,NX)+INTG3(K,IX,NX))                   TAK11840
        ELSE IF(IGRID(I,NX).EQ.0) THEN                                 TAK11850
           AA(K)= A(K,I,NX)                                            TAK11860
           AI1(K)= INTG1(K,I,NX)                                       TAK11870
           AI3(K)= INTG3(K,I,NX)                                       TAK11880
        ELSE                                                           TAK11890
           AA(K)= A(K,IX,NX)                                           TAK11900
           AI1(K)= INTG1(K,IX,NX)                                      TAK11910
           AI3(K)= INTG3(K,IX,NX)                                      TAK11920
        ENDIF                                                          TAK11930
*                                                                      TAK11940
        AI2L(K)=0.0                                                    TAK11950
        AI2R(K)=0.0                                                    TAK11960
        DO 103 L=1,JP                                                  TAK11970
           IF(IGRID(I,NX).EQ.0.AND.IGRID(IX,NX).EQ.0) THEN             TAK11980
             AI2=0.5*(INTG2(L,K,I,NX)+INTG2(L,K,IX,NX))                TAK11990
           ELSE IF(IGRID(I,NX).EQ.0) THEN                              TAK12000
             AI2= INTG2(L,K,I,NX)                                      TAK12010
           ELSE                                                        TAK12020
             AI2= INTG2(L,K,IX,NX)                                     TAK12030
           ENDIF                                                       TAK12040
           IF(K.NE.L) THEN                                             TAK12050
              AI2R(K)=AI2R(K)+C(L,I,N)*AI2                             TAK12060
           ELSE                                                        TAK12070
              AI2L(K)= AI2                                             TAK12080
           ENDIF                                                       TAK12090
 103    CONTINUE                                                       TAK12100
*                                                                      TAK12110
        C11= AI1(K)/DELT                                               TAK12120
        C12=-ETAT*AI2L(K)/AVGD                                         TAK12130
        C13= AA(K)**2*NV*AI1(K)/AVGD**2                                TAK12140
*                                                                      TAK12150
        C21=-AI1(K)*C(K,I,N)/DELT                                      TAK12160
        C22=-ETAT*AI2R(K)/AVGD                                         TAK12170
        C23= G*ETAX*AI3(K)                                             TAK12180
*                                                                      TAK12190
*****.....TREMS INVLOVING HORIZONTAL VISCOSITY                         TAK12200
*                                                                      TAK12210
        IF(HVISC) THEN                                                 TAK12220
*                                                                      TAK12230
          AI5L(K)=0.0                                                  TAK12240
          AI5R(K)=0.0                                                  TAK12250
          AI6L(K)=0.0                                                  TAK12260
          AI6R(K)=0.0                                                  TAK12270
          AI7L(K)=0.0                                                  TAK12280
          AI7R(K)=0.0                                                  TAK12290
          DO 105 L=1,JP                                                TAK12300
            IF(IGRID(IX,NX).EQ.0.AND.IGRID(I,NX).EQ.0)THEN             TAK12310
               AI5=0.5*(INTG5(L,K,I,NX)+INTG5(L,K,IX,NX))              TAK12320
               AI6=0.5*(INTG6(L,K,I,NX)+INTG6(L,K,IX,NX))              TAK12330
               AI7=0.5*(INTG7(L,K,I,NX)+INTG7(L,K,IX,NX))              TAK12340
            ELSE IF(IGRID(IX,NX).EQ.0) THEN                            TAK12350
               AI5= INTG5(L,K,IX,NX)                                   TAK12360
               AI6= INTG6(L,K,IX,NX)                                   TAK12370
```

```
                AI7= INTG7(L,K,IX,NX)                                   TAK12380
              ELSE                                                      TAK12390
                AI5= INTG5(L,K,I,NX)                                    TAK12400
                AI6= INTG6(L,K,I,NX)                                    TAK12410
                AI7= INTG7(L,K,I,NX)                                    TAK12420
              ENDIF                                                     TAK12430
              IF(K.NE.L) THEN                                           TAK12440
                 AI5R(K)=AI5R(K)+C(L,I,N)*AI5                           TAK12450
                 AI6R(K)=AI6R(K)+C(L,I,N)*AI6                           TAK12460
                 AI7R(K)=AI7R(K)+C(L,I,N)*AI7                           TAK12470
               ELSE                                                     TAK12480
                 AI5L(K)=AI5                                            TAK12490
                 AI6L(K)=AI6                                            TAK12500
                 AI7L(K)=AI7                                            TAK12510
              ENDIF                                                     TAK12520
 105       CONTINUE                                                     TAK12530
*                                                                       TAK12540
          IF(IGRID(I,NX).EQ.0.AND.IGRID(IX,NX).EQ.0) THEN               TAK12550
             CDX2=(C(K,I+1,N)-2.0*C(K,I,N)+C(K,I-1,N))/DELX**2          TAK12560
          ELSE IF(IGRID(I,NX).EQ.0) THEN                                TAK12570
             CDX2=(C(K,I-1,N)-C(K,I,N))/DELX**2                         TAK12580
          ELSE                                                          TAK12590
             CDX2=(C(K,I+1,N)-C(K,I,N))/DELX**2                         TAK12600
          ENDIF                                                         TAK12610
*                                                                       TAK12620
          IF(IG1.EQ.0.AND.IG2.EQ.0.AND.                                 TAK12630
     &          IG3.EQ.0.AND.IG4.EQ.0) THEN                             TAK12640
             CDY2=(C(K,I,N+1)-2.0*C(K,I,N)+C(K,I,N-1))/DELY**2          TAK12650
          ELSE IF(IG1.EQ.0.AND.IG2.EQ.0) THEN                           TAK12660
             CDY2=(C(K,I,N+1)-C(K,I,N))/DELY**2                         TAK12670
          ELSE IF(IG3.EQ.0.AND.IG4.EQ.0) THEN                           TAK12680
             CDY2=(C(K,I,N-1)-C(K,I,N))/DELY**2                         TAK12690
          ELSE                                                          TAK12700
             CDY2=0.0                                                   TAK12710
          ENDIF                                                         TAK12720
*                                                                       TAK12730
          C14=-NH*AI7L(K)                                               TAK12740
          C15= NH*DELDX*AI5L(K)/AVGD                                    TAK12750
          C16= NH*DELDY*AI6L(K)/AVGD                                    TAK12760
*                                                                       TAK12770
          C24=-NH*AI7R(K)                                               TAK12780
          C25= NH*DELDX*AI5R(K)/AVGD                                    TAK12790
          C26= NH*DELDY*AI6R(K)/AVGD                                    TAK12800
          C27=-NH*(CDX2+CDY2)*AI1(K)                                    TAK12810
*                                                                       TAK12820
          C28=0.0                                                       TAK12830
          DO 112 L=1,JP                                                 TAK12840
             CDX1=(C(L,I+1,N)-C(L,I-1,N))/(2.0*DELX)                    TAK12850
             IF(IG1.EQ.0.AND.IG2.EQ.0.AND.                              TAK12860
     &             IG3.EQ.0.AND.IG4.EQ.0) THEN                          TAK12870
               CDY1=(C(L,I,N+1)-C(L,I,N-1))/(2.0*DELY)                  TAK12880
             ELSE IF(IG1.EQ.0.AND.IG2.EQ.0) THEN                        TAK12890
               CDY1=(C(L,I,N+1)-C(L,I,N))/DELY                          TAK12900
             ELSE IF(IG3.EQ.0.AND.IG4.EQ.0) THEN                        TAK12910
               CDY1=(C(L,I,N)-C(L,I,N-1))/DELY                          TAK12920
             ENDIF                                                      TAK12930
             AI5=0.5*(INTG5(L,K,I,NX)+INTG5(L,K,IX,NX))                 TAK12940
             IF(IGRID(I,NX).EQ.0.AND.IGRID(IX,NX).EQ.0) THEN            TAK12950
               AI6=0.5*(INTG5(L,K,I,NX)+INTG5(L,K,IX,NX))               TAK12960
             ELSE IF(IGRID(I,NX).EQ.0) THEN                             TAK12970
               AI5= INTG5(L,K,I,NX)                                     TAK12980
```

```
            ELSE                                                        TAK12990
                AI5= INTG5(L,K,I,NX)                                    TAK13000
            ENDIF                                                       TAK13010
            C28A=(NH*CDX1)*AI5                                          TAK13020
            C28B=(NH*CDY1)*AI6                                          TAK13030
            C28=C28-2.0*(C28A+C28B)                                     TAK13040
112       CONTINUE                                                      TAK13050
         ELSE                                                           TAK13060
            C14=0.0                                                     TAK13070
            C15=0.0                                                     TAK13080
            C16=0.0                                                     TAK13090
            C24=0.0                                                     TAK13100
            C25=0.0                                                     TAK13110
            C26=0.0                                                     TAK13120
            C27=0.0                                                     TAK13130
            C28=0.0                                                     TAK13140
          ENDIF                                                         TAK13150
*                                                                       TAK13160
            CC1=C11+C12+C13+C14+C15+C16                                 TAK13170
            CC2=C21+C22+C23+C24+C25+C26+C27+C28                         TAK13180
            IF( ABS(CC2).LT.1.0E-25) CC2=0.0                            TAK13190
            NC(K,I,N)=-CC2/CC1                                          TAK13200
102         CONTINUE                                                    TAK13210
101    CONTINUE                                                         TAK13220
       RETURN                                                           TAK13230
       END                                                              TAK13240
```

```
      SUBROUTINE YMOM                                                   TAK13250
*****************************************************************     TAK13260
*                                                               *     TAK13270
*       SOLVES THE Y-MOMENTUM EQUATION                          *     TAK13280
*                                                               *     TAK13290
*****************************************************************     TAK13300
*                                                                     TAK13310
*                                                                     TAK13320
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK13330
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK13340
     &           IWPLS=IWIDTH+2)                                      TAK13350
      REAL   NV,NH,NC,ND,NETA                                         TAK13360
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                TAK13370
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK13380
*                                                                     TAK13390
      DIMENSION  AA(JP),AI1(JP),AI2R(JP),AI2L(JP),AI3(JP),AI5R(JP),   TAK13400
     &           AI5L(JP),AI6L(JP),AI6R(JP),AI7R(JP),AI7L(JP)         TAK13410
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK13420
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,          TAK13430
     &              ISTRT,NFIN,NHOTST                                 TAK13440
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK13450
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),           TAK13460
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)          TAK13470
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK13480
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)             TAK13490
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),     TAK13500
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),     TAK13510
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),  TAK13520
     &              INTG7(JP,JP,LPLS,IWPLS)                           TAK13530
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK13540
      DATA   G/   9.81/                                               TAK13550
*                                                                     TAK13560
*                                                                     TAK13570
*                                                                     TAK13580
       DO 101 N=1,IWIDP1                                              TAK13590
         NY=N+1                                                       TAK13600
         DO 101 I=1,LENGTH                                            TAK13610
           IY=I+1                                                     TAK13620
*                                                                     TAK13630
           IF(IGRID(IY,N).EQ.1) GOTO 101                              TAK13640
           IF(IGRID(IY,NY).EQ.1) GOTO 101                             TAK13650
           IF(IGRID(IY,N).EQ.2.AND.IGRID(IY,NY).EQ.2) GOTO 101        TAK13660
*                                                                     TAK13670
*****.....TERMS INVOLVING VERTICAL VISCOSITY                          TAK13680
*                                                                     TAK13690
           IF(IGRID(IY,N).EQ.0.AND.IGRID(IY,NY).EQ.0) THEN            TAK13700
              AVGNY=0.5*(NETA(IY,NY)+NETA(IY,N))                     TAK13710
              AVGEY=0.5*(ETA(IY,NY)+ETA(IY,N))                       TAK13720
              AVGHY=0.5*(H(IY,NY)+H(IY,N))                           TAK13730
              ETAY =(NETA(IY,NY)-NETA(IY,N))/DELY                    TAK13740
           ELSE IF(IGRID(IY,N).EQ.0) THEN                            TAK13750
              AVGNY= NETA(IY,N)                                      TAK13760
              AVGEY= ETA(IY,N)                                       TAK13770
              AVGHY= H(IY,N)                                         TAK13780
              ETAY =(NETA(IY,N)-NETA(IY,N-1))/DELY                   TAK13790
           ELSE                                                      TAK13800
              AVGNY= NETA(IY,NY)                                     TAK13810
              AVGEY= ETA(IY,NY)                                      TAK13820
```

```
                AVGHY= H(IY,NY)                                          TAK13830
                ETAY =(NETA(IY,NY+1)-NETA(IY,NY))/DELY                   TAK13840
              ENDIF                                                      TAK13850
              AVGD =AVGNY-AVGHY                                          TAK13860
              ETAT =(AVGNY-AVGEY)/DELT                                   TAK13870
*                                                                        TAK13880
*****.......TERMS INVLOVING HORIZONTAL VISCOSITY                        TAK13890
*                                                                        TAK13900
            IF(HVISC) THEN                                               TAK13910
              IF(IGRID(I,NY).EQ.0.AND.IGRID(IY+1,NY).EQ.0) THEN          TAK13920
                  DELE1=(NETA(IY+1,NY)-NETA(I,NY))/(2.0*DELX)            TAK13930
                  DELH1=(H(IY+1,NY)-H(I,NY))/(2.0*DELX)                  TAK13940
              ELSE IF(IGRID(I,NY).EQ.0) THEN                             TAK13950
                  DELE1=(NETA(IY,NY)-NETA(I,NY))/DELX                    TAK13960
                  DELH1=(H(IY,NY)-H(I,NY))/DELX                          TAK13970
              ELSE IF(IGRID(IY+1,NY).EQ.0) THEN                          TAK13980
                  DELE1=(NETA(IY+1,NY)-NETA(IY,NY))/DELX                 TAK13990
                  DELH1=(H(IY+1,NY)-H(IY,NY))/DELX                       TAK14000
              ELSE                                                       TAK14010
                  DELE1=0.0                                              TAK14020
                  DELH1=0.0                                              TAK14030
              ENDIF                                                      TAK14040
*                                                                        TAK14050
              IF(IGRID(I,N).EQ.0.AND.IGRID(IY+1,N).EQ.0) THEN            TAK14060
                DELE2=(NETA(IY+1,N)-NETA(I,N))/(2.0*DELX)                TAK14070
                DELH2=(H(IY+1,N)-H(I,N))/(2.0*DELX)                      TAK14080
              ELSE IF(IGRID(I,N).EQ.0) THEN                              TAK14090
                DELE2=(NETA(IY,N)-NETA(I,N))/DELX                        TAK14100
                DELH2=(H(IY,N)-H(I,N))/DELX                              TAK14110
              ELSE IF(IGRID(IY+1,N).EQ.0) THEN                           TAK14120
                DELE2=(NETA(IY+1,N)-NETA(IY,N))/DELX                     TAK14130
                DELH2=(H(IY+1,N)-H(IY,N))/DELX                           TAK14140
              ELSE                                                       TAK14150
                DELE2=0.0                                                TAK14160
                DELH2=0.0                                                TAK14170
              ENDIF                                                      TAK14180
              DELEX=0.5*(DELE1+DELE2)                                    TAK14190
              DELHX=0.5*(DELH1+DELH2)                                    TAK14200
              DELDX=DELEX-DELHX                                          TAK14210
*                                                                        TAK14220
              IF(IGRID(IY,N).EQ.0.AND.IGRID(IY,NY).EQ.0) THEN            TAK14230
                DELEY=(NETA(IY,NY)-NETA(IY,N))/DELY                      TAK14240
                DELHY=(H(IY,NY)-H(IY,N))/DELY                            TAK14250
              ELSE IF(IGRID(IY,N).EQ.0) THEN                             TAK14260
                DELEY=(NETA(IY,N)-NETA(IY,N-1))/DELY                     TAK14270
                DELHY=(H(IY,N)-H(IY,N-1))/DELY                           TAK14280
              ELSE                                                       TAK14290
                DELEY=(NETA(IY,NY+1)-NETA(IY,NY))/DELY                   TAK14300
                DELHY=(H(IY,NY+1)-H(IY,NY))/DELY                         TAK14310
              ENDIF                                                      TAK14320
              DELDY=DELEY-DELHY                                          TAK14330
*                                                                        TAK14340
             IG1=IGRID(I,N)                                              TAK14350
             IG2=IGRID(I,NY)                                             TAK14360
             IG3=IGRID(IY+1,N)                                           TAK14370
             IG4=IGRID(IY+1,NY)                                          TAK14380
*                                                                        TAK14390
           ENDIF                                                         TAK14400
*                                                                        TAK14410
*****......TERMS INVOLVING VERTICAL VISCOSITY                           TAK14420
*                                                                        TAK14430
```

```
            DO 102 K=1,JP                                                    TAK14440
              AA(K)=0.0                                                      TAK14450
              AI1(K)=0.0                                                     TAK14460
              AI3(K)=0.0                                                     TAK14470
              IF(IGRID(IY,N).EQ.0.AND.IGRID(IY,NY).EQ.0) THEN                TAK14480
                AA(K)=0.5*(A(K,IY,NY)+A(K,IY,N))                             TAK14490
                AI1(K)=0.5*(INTG1(K,IY,NY)+INTG1(K,IY,N))                    TAK14500
                AI3(K)=0.5*(INTG3(K,IY,NY)+INTG3(K,IY,N))                    TAK14510
              ELSE IF(IGRID(IY,N).EQ.0) THEN                                 TAK14520
                AA(K)= A(K,IY,N)                                             TAK14530
                AI1(K)= INTG1(K,IY,N)                                        TAK14540
                AI3(K)= INTG3(K,IY,N)                                        TAK14550
              ELSE                                                           TAK14560
                AA(K)= A(K,IY,NY)                                            TAK14570
                AI1(K)= INTG1(K,IY,NY)                                       TAK14580
                AI3(K)= INTG3(K,IY,NY)                                       TAK14590
              ENDIF                                                          TAK14600
*                                                                            TAK14610
              AI2L(K)=0.0                                                    TAK14620
              AI2R(K)=0.0                                                    TAK14630
              DO 103 L=1,JP                                                  TAK14640
                IF(IGRID(IY,N).EQ.0.AND.IGRID(IY,NY).EQ.0) THEN              TAK14650
                   AI2=0.5*(INTG2(L,K,IY,NY)+INTG2(L,K,IY,N))                TAK14660
                ELSE IF(IGRID(IY,N).EQ.0) THEN                               TAK14670
                   AI2= INTG2(L,K,IY,N)                                      TAK14680
                ELSE                                                         TAK14690
                   AI2= INTG2(L,K,IY,NY)                                     TAK14700
                ENDIF                                                        TAK14710
                IF(K.NE.L) THEN                                              TAK14720
                   AI2R(K)=AI2R(K)+D(L,I,N)*AI2                              TAK14730
                ELSE                                                         TAK14740
                   AI2L(K)= AI2                                              TAK14750
                ENDIF                                                        TAK14760
 103        CONTINUE                                                         TAK14770
*                                                                            TAK14780
            D11= AI1(K)/DELT                                                 TAK14790
            D12=-ETAT*AI2L(K)/AVGD                                           TAK14800
            D13= AA(K)**2*NV*AI1(K)/AVGD**2                                  TAK14810
*                                                                            TAK14820
            D21=-AI1(K)*D(K,I,N)/DELT                                        TAK14830
            D22=-ETAT*AI2R(K)/AVGD                                           TAK14840
            D23= G*ETAY*AI3(K)                                               TAK14850
*                                                                            TAK14860
*****.....TREMS INVLOVING HORIZONTAL VISCOSITY                               TAK14870
*                                                                            TAK14880
        IF(HVISC) THEN                                                       TAK14890
*                                                                            TAK14900
              AI5L(K)=0.0                                                    TAK14910
              AI5R(K)=0.0                                                    TAK14920
              AI6L(K)=0.0                                                    TAK14930
              AI6R(K)=0.0                                                    TAK14940
              AI7L(K)=0.0                                                    TAK14950
              AI7R(K)=0.0                                                    TAK14960
              DO 105 L=1,JP                                                  TAK14970
                IF(IGRID(IY,N).EQ.0.AND.IGRID(IY,NY).EQ.0) THEN              TAK14980
                  AI5=0.5*(INTG5(L,K,IY,NY)+INTG5(L,K,IY,N))                 TAK14990
                  AI6=0.5*(INTG6(L,K,IY,NY)+INTG6(L,K,IY,N))                 TAK15000
                  AI7=0.5*(INTG7(L,K,IY,NY)+INTG7(L,K,IY,N))                 TAK15010
                ELSE IF(IGRID(IY,N).EQ.0.) THEN                              TAK15020
                  AI5= INTG5(L,K,IY,N)                                       TAK15030
                  AI6= INTG6(L,K,IY,N)                                       TAK15040
```

```
                AI7= INTG7(L,K,IY,N)                                    TAK15050
              ELSE                                                      TAK15060
                AI5= INTG5(L,K,IY,NY)                                   TAK15070
                AI6= INTG6(L,K,IY,NY)                                   TAK15080
                AI7= INTG7(L,K,IY,NY)                                   TAK15090
              ENDIF                                                     TAK15100
              IF(K.NE.L) THEN                                           TAK15110
                 AI5R(K)=AI5R(K)+D(L,I,N)*AI5                           TAK15120
                 AI6R(K)=AI6R(K)+D(L,I,N)*AI6                           TAK15130
                 AI7R(K)=AI7R(K)+D(L,I,N)*AI7                           TAK15140
              ELSE                                                      TAK15150
                 AI5L(K)=AI5                                            TAK15160
                 AI6L(K)=AI6                                            TAK15170
                 AI7L(K)=AI7                                            TAK15180
              ENDIF                                                     TAK15190
  105       CONTINUE                                                    TAK15200
*                                                                       TAK15210
            IF(IG1.EQ.0.AND.IG2.EQ.0.AND.IG3.EQ.0.AND.                  TAK15220
     &                                        IG4.EQ.0) THEN            TAK15230
              DDX2=(D(K,I+1,N)-2.0*D(K,I,N)+D(K,I-1,N))/DELX**2         TAK15240
            ELSE IF(IG1.EQ.0.AND.IG2.EQ.0) THEN                         TAK15250
              DDX2=(-D(K,I,N)+D(K,I-1,N))/DELX**2                       TAK15260
            ELSE IF(IG3.EQ.0.AND.IG4.EQ.0) THEN                         TAK15270
              DDX2=(D(K,I+1,N)-D(K,I,N))/DELX**2                        TAK15280
            ELSE                                                        TAK15290
              DDX2=0.0                                                  TAK15300
            ENDIF                                                       TAK15310
*                                                                       TAK15320
            IF(IGRID(IY,NY).EQ.0.AND.IGRID(IY,N).EQ.0) THEN             TAK15330
              DDY2=(D(K,I,N+1)-2.0*D(K,I,N)+D(K,I,N-1))/DELY**2         TAK15340
            ELSE IF(IGRID(IY,NY).EQ.0) THEN                             TAK15350
              DDY2=(D(K,I,N+1)-D(K,I,N))/DELY**2                        TAK15360
            ELSE                                                        TAK15370
              DDY2=(-D(K,I,N-1)+D(K,I,N))/DELY**2                       TAK15380
            ENDIF                                                       TAK15390
*                                                                       TAK15400
            D14=-NH*AI7L(K)                                             TAK15410
            D15= NH*DELDX*AI5L(K)/AVGD                                  TAK15420
            D16= NH*DELDY*AI6L(K)/AVGD                                  TAK15430
*                                                                       TAK15440
            D24=-NH*AI7R(K)                                             TAK15450
            D25= NH*DELDX*AI5R(K)/AVGD                                  TAK15460
            D26= NH*DELDY*AI6R(K)/AVGD                                  TAK15470
            D27=-NH*(DDX2+DDY2)*AI1(K)                                  TAK15480
*                                                                       TAK15490
            D28=0.0                                                     TAK15500
            DO 112 L=1,JP                                               TAK15510
              IF(IG1.EQ.0.AND.IG2.EQ.0.AND.                             TAK15520
     &             IG3.EQ.0.AND.IG4.EQ.0) THEN                          TAK15530
                DDX1=(D(L,I+1,N)-D(L,I-1,N))/(2.0*DELX)                 TAK15540
              ELSE IF(IG1.EQ.0.AND.IG2.EQ.0) THEN                       TAK15550
                DDX1=(D(L,I,N)-D(L,I-1,N))/DELX                         TAK15560
              ELSE IF(IG3.EQ.0.AND.IG4.EQ.0) THEN                       TAK15570
                DDX1=(D(L,I+1,N)-D(L,I,N))/DELX                         TAK15580
              ENDIF                                                     TAK15590
              DDY1=(D(L,I,N+1)-D(L,I,N-1))/(2.0*DELY)                   TAK15600
              IF(IGRID(IY,NY).EQ.0.AND.IGRID(IY,N).EQ.0) THEN           TAK15610
                AI5=0.5*(INTG5(L,K,IY,NY)+INTG5(L,K,IY,N))              TAK15620
              ELSE IF(IGRID(IY,NY).EQ.0) THEN                           TAK15630
                AI5= INTG5(L,K,IY,NY)                                   TAK15640
              ELSE                                                      TAK15650
```

```
                AI5= INTG5(L,K,IY,N)                                    TAK15660
              ENDIF                                                     TAK15670
              AI6=0.5*(INTG6(L,K,IY,NY)+INTG6(L,K,IY,N))                TAK15680
              D28A=(NH*DDX1)*AI5                                        TAK15690
              D28B=(NH*DDY1)*AI6                                        TAK15700
              D28=D28-2.0*NH*(DDX1*AI5+DDY1*AI6)                        TAK15710
112         CONTINUE                                                    TAK15720
          ELSE                                                          TAK15730
            D14=0.0                                                     TAK15740
            D15=0.0                                                     TAK15750
            D16=0.0                                                     TAK15760
            D24=0.0                                                     TAK15770
            D25=0.0                                                     TAK15780
            D26=0.0                                                     TAK15790
            D27=0.0                                                     TAK15800
            D28=0.0                                                     TAK15810
          ENDIF                                                         TAK15820
*                                                                       TAK15830
            DD1=D11+D12+D13+D14+D15+D16                                 TAK15840
            DD2=D21+D22+D23+D24+D25+D26+D27+D28                         TAK15850
            IF( ABS(DD2).LT.1.0E-25) DD2=0.0                            TAK15860
            ND(K,I,N)=-DD2/DD1                                          TAK15870
102       CONTINUE                                                      TAK15880
101     CONTINUE                                                        TAK15890
        RETURN                                                          TAK15900
        END                                                             TAK15910
```

```
      SUBROUTINE CONTIN(DTDX,DTDY,SSS)                                  TAK15920
*****************************************************************     TAK15930
*                                                               *     TAK15940
*       SOLVES THE CONTINUITY EQUATION                          *     TAK15950
*                                                               *     TAK15960
*****************************************************************     TAK15970
*                                                                     TAK15980
*                                                                     TAK15990
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK16000
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK16010
     &           IWPLS=IWIDTH+2)                                      TAK16020
      REAL   NV,NH,NC,ND,NETA                                         TAK16030
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                TAK16040
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK16050
*                                                                     TAK16060
      DIMENSION SSS(LPLS,IWPLS)                                       TAK16070
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK16080
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,          TAK16090
     &              ISTRT,NFIN,NHOTST                                 TAK16100
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                     TAK16110
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),           TAK16120
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)          TAK16130
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK16140
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)             TAK16150
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),     TAK16160
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),     TAK16170
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),  TAK16180
     &              INTG7(JP,JP,LPLS,IWPLS)                           TAK16190
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK16200
      DATA   ITERS    /1  /                                           TAK16210
*                                                                     TAK16220
*                                                                     TAK16230
*                                                                     TAK16240
      DO 101 ITER=1,ITERS                                             TAK16250
           DO 102 N=2,IWPLS-1                                         TAK16260
                DO 102 I=2,LPLS-1                                     TAK16270
                     IF(IGRID(I,N).EQ.0) THEN                         TAK16280
                         Z1=0.0                                       TAK16290
                         Z2=0.0                                       TAK16300
                         Z3=0.0                                       TAK16310
                         Z4=0.0                                       TAK16320
                         SS=SSS(I,N)                                  TAK16330
*                                                                     TAK16340
*****.....COMPUTATIONS IN X-DIRECTION                                 TAK16350
*                                                                     TAK16360
                         DO 103 K=1,JP                                TAK16370
                             IF(IGRID(I+1,N).EQ.0) THEN               TAK16380
                                SQ1=(CONTI(K,I,N)+CONTI(K,I+1,N))*0.5 TAK16390
                             ELSE IF(IGRID(I+1,N).EQ.2) THEN          TAK16400
                                SQ1=CONTI(K,I,N)                      TAK16410
                             ELSE                                     TAK16420
                                SQ1=0.                                TAK16430
                             ENDIF                                    TAK16440
*                                                                     TAK16450
                             IF(IGRID(I-1,N).EQ.0) THEN               TAK16460
                                 SQ2=(CONTI(K,I,N)+CONTI(K,I-1,N))*0.5 TAK16470
                             ELSE IF(IGRID(I-1,N).EQ.2) THEN          TAK16480
                                 SQ2=CONTI(K,I,N)                     TAK16490
                             ELSE                                     TAK16500
```

```
                                SQ2=0.                                    TAK16510
                              ENDIF                                       TAK16520
*                                                                         TAK16530
*****.....COMPUTATIONS IN Y-DIRECTION                                     TAK16540
*                                                                         TAK16550
                               IF(IGRID(I,N+1).EQ.0) THEN                 TAK16560
                                   SQ3=(CONTI(K,I,N)+CONTI(K,I,N+1))*0.5  TAK16570
                               ELSE IF(IGRID(I,N+1).EQ.2) THEN            TAK16580
                                   SQ3=CONTI(K,I,N)                       TAK16590
                               ELSE                                       TAK16600
                                   SQ3=0.                                 TAK16610
                               ENDIF                                      TAK16620
*                                                                         TAK16630
                               IF(IGRID(I,N-1).EQ.0) THEN                 TAK16640
                                   SQ4=(CONTI(K,I,N)+CONTI(K,I,N-1))*0.5  TAK16650
                               ELSE IF(IGRID(I,N-1).EQ.2) THEN            TAK16660
                                   SQ4=CONTI(K,I,N)                       TAK16670
                               ELSE                                       TAK16680
                                   SQ4=0.                                 TAK16690
                              ENDIF                                       TAK16700
*                                                                         TAK16710
                              Z1=Z1+C(K,I,N-1)*SQ1                        TAK16720
                              Z2=Z2+C(K,I-1,N-1)*SQ2                      TAK16730
                              Z3=Z3+D(K,I-1,N)*SQ3                        TAK16740
                              Z4=Z4+D(K,I-1,N-1)*SQ4                      TAK16750
*                                                                         TAK16760
 103                    CONTINUE                                          TAK16770
*                                                                         TAK16780
                       IF(IGRID(I+1,N).EQ.0) THEN                         TAK16790
                            DA=0.5*(NETA(I,N)+ETA(I,N))                   TAK16800
                            DB=0.5*(NETA(I+1,N)+ETA(I+1,N))               TAK16810
                            D1=0.5*(DA-H(I,N)+DB-H(I+1,N))                TAK16820
                       ELSE                                               TAK16830
                            DA=0.5*(NETA(I,N)+ETA(I,N))                   TAK16840
                            D1= DA-H(I,N)                                 TAK16850
                       ENDIF                                              TAK16860
*                                                                         TAK16870
                       IF(IGRID(I-1,N).EQ.0) THEN                         TAK16880
                            DA=0.5*(NETA(I,N)+ETA(I,N))                   TAK16890
                            DB=0.5*(NETA(I-1,N)+ETA(I-1,N))               TAK16900
                            D2=0.5*(DA-H(I,N)+DB-H(I-1,N))                TAK16910
                       ELSE                                               TAK16920
                            DA=0.5*(NETA(I,N)+ETA(I,N))                   TAK16930
                            D2=DA-H(I,N)                                  TAK16940
                       ENDIF                                              TAK16950
*                                                                         TAK16960
                       IF(IGRID(I,N+1).EQ.0) THEN                         TAK16970
                           DA=0.5*(NETA(I,N)+ETA(I,N))                    TAK16980
                           DB=0.5*(NETA(I,N+1)+ETA(I,N+1))                TAK16990
                           D3=0.5*(DA-H(I,N)+DB-H(I,N+1))                 TAK17000
                       ELSE                                               TAK17010
                           DA=0.5*(NETA(I,N)+ETA(I,N))                    TAK17020
                           D3=DA-H(I,N)                                   TAK17030
                       ENDIF                                              TAK17040
*                                                                         TAK17050
                       IF(IGRID(I,N-1).EQ.0) THEN                         TAK17060
                            DA=0.5*(NETA(I,N)+ETA(I,N))                   TAK17070
                            DB=0.5*(NETA(I,N-1)+ETA(I,N-1))               TAK17080
                            D4=0.5*(DA-H(I,N)+DB-H(I,N-1))                TAK17090
                       ELSE                                               TAK17100
                            DA=0.5*(NETA(I,N)+ETA(I,N))                   TAK17110
```

```
                        D4=DA-H(I,N)                                    TAK17120
                      ENDIF                                             TAK17130
*                                                                       TAK17140
                    Z1=Z1*D1                                            TAK17150
                    Z2=Z2*D2                                            TAK17160
                    Z3=Z3*D3                                            TAK17170
                    Z4=Z4*D4                                            TAK17180
                    NETA(I,N)=ETA(I,N)-DTDX*(Z1-Z2)-DTDY*(Z3-Z4)+SS     TAK17190
                  ENDIF                                                 TAK17200
  102           CONTINUE                                                TAK17210
  101       CONTINUE                                                    TAK17220
            RETURN                                                      TAK17230
            END                                                         TAK17240
```

```
      SUBROUTINE MOVEIN(JJ,SSS,DXDY)                                    TAK17250
*****************************************************************     TAK17260
*                                                               *     TAK17270
*     SUBROUTINE DETERMINES THE ADVANCE/RETREAT OF GLACIER      *     TAK17280
*                                                               *     TAK17290
*****************************************************************     TAK17300
*                                                                       TAK17310
*                                                                       TAK17320
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                        TAK17330
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,          TAK17340
     &           IWPLS=IWIDTH+2)                                        TAK17350
      REAL   NV,NH,NC,ND,NETA                                           TAK17360
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                  TAK17370
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                       TAK17380
*                                                                       TAK17390
      DIMENSION SSS(LPLS,IWPLS)                                         TAK17400
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO           TAK17410
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,            TAK17420
     &              ISTRT,NFIN,NHOTST                                   TAK17430
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                       TAK17440
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),             TAK17450
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)            TAK17460
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)    TAK17470
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)               TAK17480
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),       TAK17490
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),       TAK17500
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),    TAK17510
     &              INTG7(JP,JP,LPLS,IWPLS)                             TAK17520
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                        TAK17530
      COMMON/ BLK10/ QIF(LPLS,IWPLS),QIB(LPLS,IWPLS),                   TAK17540
     &              QJF(LPLS,IWPLS),QJB(LPLS,IWPLS)                     TAK17550
*                                                                       TAK17560
*                                                                       TAK17570
*                                                                       TAK17580
*****.....ADVANCE OR RETREAT IN X-DIRECTION                            TAK17590
*                                                                       TAK17600
*                                                                       TAK17610
      DO 101 J=2,IWPLS-1                                                TAK17620
         DO 101 I=2,LPLS-1                                              TAK17630
*                                                                       TAK17640
            IF(IGRID(I,J).EQ.2) THEN                                    TAK17650
*                                                                       TAK17660
               IF(IGRID(I+1,J).EQ.0) THEN                               TAK17670
*                                                                       TAK17680
                  UU=0.0                                                TAK17690
                  DO 102 K=1,JP                                         TAK17700
                     UU=UU+CONTI(K,I+1,J)*NC(K,I,J-1)                   TAK17710
102               CONTINUE                                              TAK17720
                  DEP=NETA(I+1,J)-H(I+1,J)                              TAK17730
                  FLOW= UU*DEP*DELY*DELT                                TAK17740
                  QIF(I,J)=QIF(I,J)+FLOW                                TAK17750
                  VOL=0.5*DEP*DELX*DELY                                 TAK17760
                  IF(UU.LE.0.0) THEN                                    TAK17770
                     SSS(I+1,J)=SOS(I+1,J)+ABS(FLOW)/DXDY               TAK17780
                  ELSE                                                  TAK17790
                     SSS(I+1,J)=SOS(I+1,J)-ABS(FLOW)/DXDY               TAK17800
                  ENDIF                                                 TAK17810
                  IF( ABS(QIF(I,J)).GT.VOL) THEN                        TAK17820
                     SSS(I+1,J)=SOS(I+1,J)                              TAK17830
                     IF(QIF(I,J).GT.0.0) THEN                           TAK17840
                        IGRID(I+1,J)=2                                  TAK17850
```

```
                ETA(I+1,J)=H(I+1,J)                                    TAK17860
                NETA(I+1,J)=H(I+1,J)                                   TAK17870
                DO 103 K=1,JP                                          TAK17880
                     A(K,I+1,J)=0.0                                    TAK17890
                     CONTI(K,I+1,J)=0.0                                TAK17900
                     INTG1(K,I+1,J)=0.0                                TAK17910
                     INTG3(K,I+1,J)=0.0                                TAK17920
                     DO 103 L=1,JP                                     TAK17930
                          INTG2(L,K,I+1,J)=0.0                         TAK17940
                          INTG4(L,K,I+1,J)=0.0                         TAK17950
                          INTG5(L,K,I+1,J)=0.0                         TAK17960
                          INTG6(L,K,I+1,J)=0.0                         TAK17970
                         INTG7(L,K,I+1,J)=0.0                          TAK17980
103                  CONTINUE                                          TAK17990
                   ELSE                                                TAK18000
                     IGRID(I,J)=0                                      TAK18010
                     ETA(I,J)=H(I,J)+10.00                             TAK18020
                     NETA(I,J)=H(I,J)+10.0                             TAK18030
                     DO 104 K=1,JP                                     TAK18040
                         A(K,I,J)=A(K,I+1,J)                           TAK18050
                         CONTI(K,I,J)=CONTI(K,I+1,J)                   TAK18060
                         INTG1(K,I,J)=INTG1(K,I+1,J)                   TAK18070
                         INTG3(K,I,J)=INTG3(K,I+1,J)                   TAK18080
                         DO 104 L=1,JP                                 TAK18090
                              INTG2(L,K,I,J)=INTG2(L,K,I+1,J)          TAK18100
                              INTG4(L,K,I,J)=INTG4(L,K,I+1,J)          TAK18110
                              INTG5(L,K,I,J)=INTG5(L,K,I+1,J)          TAK18120
                              INTG6(L,K,I,J)=INTG6(L,K,I+1,J)          TAK18130
                              INTG7(L,K,I,J)=INTG7(L,K,I+1,J)          TAK18140
104                    CONTINUE                                        TAK18150
                   ENDIF                                               TAK18160
                   QIF(I,J)=0.0                                        TAK18170
                ENDIF                                                  TAK18180
*                                                                      TAK18190
                ELSE IF(IGRID(I-1,J).EQ.0) THEN                        TAK18200
*                                                                      TAK18210
                   UU=0.0                                              TAK18220
                   DO 105 K=1,JP                                       TAK18230
                        UU=UU+CONTI(K,I-1,J)*NC(K,I-1,J-1)             TAK18240
105                CONTINUE                                            TAK18250
                   DEP=NETA(I-1,J)-H(I-1,J)                            TAK18260
                   FLOW = UU*DEP*DELY*DELT                             TAK18270
                   QIB(I,J)=QIB(I,J)+FLOW                              TAK18280
                   VOL=0.5*DEP*DELX*DELY                               TAK18290
                   IF(UU.GE.0.0) THEN                                  TAK18300
                      SSS(I-1,J)=SOS(I-1,J)+ABS(FLOW)/DXDY             TAK18310
                   ELSE                                                TAK18320
                      SSS(I-1,J)=SOS(I-1,J)-ABS(FLOW)/DXDY             TAK18330
                   ENDIF                                               TAK18340
                   IF( ABS(QIB(I,J)).GT.VOL) THEN                      TAK18350
                       SSS(I-1,J)=SOS(I-1,J)                           TAK18360
                       IF(QIB(I,J).GT.0.0) THEN                        TAK18370
                          IGRID(I,J)=0                                 TAK18380
                          ETA(I,J)=H(I,J)+10.0                         TAK18390
                          NETA(I,J)=H(I,J)+10.0                        TAK18400
                          DO 106 K=1,JP                                TAK18410
                               A(K,I,J)=A(K,I-1,J)                     TAK18420
                               CONTI(K,I,J)=CONTI(K,I-1,J)             TAK18430
                               INTG1(K,I,J)=INTG1(K,I-1,J)             TAK18440
                               INTG3(K,I,J)=INTG3(K,I-1,J)             TAK18450
                               DO 106 L=1,JP                           TAK18460
```

```
                                    INTG2(L,K,I,J)=INTG2(L,K,I-1,J)   TAK18470
                                    INTG4(L,K,I,J)=INTG4(L,K,I-1,J)   TAK18480
                                    INTG5(L,K,I,J)=INTG5(L,K,I-1,J)   TAK18490
                                    INTG6(L,K,I,J)=INTG6(L,K,I-1,J)   TAK18500
                                    INTG7(L,K,I,J)=INTG7(L,K,I-1,J)   TAK18510
 106                             CONTINUE                             TAK18520
                               ELSE                                   TAK18530
                                  IGRID(I-1,J)=2                      TAK18540
                                  ETA(I-1,J)=H(I-1,J)                 TAK18550
                                  NETA(I-1,J)=H(I-1,J)                TAK18560
                                  DO 107 K=1,JP                       TAK18570
                                       A(K,I-1,J)=0.0                 TAK18580
                                       CONTI(K,I-1,J)=0.0             TAK18590
                                       INTG1(K,I-1,J)=0.0             TAK18600
                                       INTG3(K,I-1,J)=0.0             TAK18610
                                       DO 107 L=1,JP                  TAK18620
                                            INTG2(L,K,I-1,J)=0.0      TAK18630
                                            INTG4(L,K,I-1,J)=0.0      TAK18640
                                            INTG5(L,K,I-1,J)=0.0      TAK18650
                                            INTG6(L,K,I-1,J)=0.0      TAK18660
                                            INTG7(L,K,I-1,J)=0.0      TAK18670
 107                                  CONTINUE                        TAK18680
                                   ENDIF                              TAK18690
                                   QIB(I,J)=0.0                       TAK18700
                               ENDIF                                  TAK18710
*                                                                     TAK18720
                    ENDIF                                             TAK18730
*                                                                     TAK18740
                ENDIF                                                 TAK18750
*                                                                     TAK18760
 101     CONTINUE                                                     TAK18770
*                                                                     TAK18780
*                                                                     TAK18790
*****.......ADVANCE IN Y-DIRECTION                                    TAK18800
*                                                                     TAK18810
*                                                                     TAK18820
         DO 201 J=2,IWPLS-1                                           TAK18830
             DO 201 I=2,LPLS-1                                        TAK18840
*                                                                     TAK18850
                  IF(IGRID(I,J).EQ.2) THEN                            TAK18860
*                                                                     TAK18870
                     IF(IGRID(I,J+1).EQ.0) THEN                       TAK18880
*                                                                     TAK18890
                        VV=0.0                                        TAK18900
                        DO 202 K=1,JP                                 TAK18910
                             VV=VV+CONTI(K,I,J+1)*ND(K,I-1,J)         TAK18920
 202                    CONTINUE                                      TAK18930
                        DEP=NETA(I,J+1)-H(I,J+1)                      TAK18940
                        FLOW = VV*DEP*DELX*DELT                       TAK18950
                        QJF(I,J)=QJF(I,J)+FLOW                        TAK18960
                        VOL=0.5*DEP*DELX*DELY                         TAK18970
                        IF(VV.LE.0.0) THEN                            TAK18980
                           SSS(I,J+1)=SOS(I,J+1)+ABS(FLOW)/DXDY       TAK18990
                        ELSE                                          TAK19000
                           SSS(I,J+1)=SOS(I,J+1)+ABS(FLOW)/DXDY       TAK19010
                        ENDIF                                         TAK19020
                        IF( ABS(QJF(I,J)).GT.VOL) THEN                TAK19030
                           SSS(I,J+1)=SOS(I,J+1)                      TAK19040
                           IF(QJF(I,J).GT.0.0) THEN                   TAK19050
                              IGRID(I,J+1)=2                          TAK19060
                              ETA(I,J+1)=H(I,J+1)                     TAK19070
```

```
                        NETA(I,J+1)=H(I,J+1)                                TAK19080
                        DO 203 K=1,JP                                       TAK19090
                             A(K,I,J+1)=0.0                                 TAK19100
                             CONTI(K,I,J+1)=0.0                             TAK19110
                             INTG1(K,I,J+1)=0.0                             TAK19120
                             INTG3(K,I,J+1)=0.0                             TAK19130
                             DO 203 L=1,JP                                  TAK19140
                                  INTG2(L,K,I,J+1)=0.0                      TAK19150
                                  INTG4(L,K,I,J+1)=0.0                      TAK19160
                                  INTG5(L,K,I,J+1)=0.0                      TAK19170
                                  INTG6(L,K,I,J+1)=0.0                      TAK19180
                                  INTG7(L,K,I,J+1)=0.0                      TAK19190
203                     CONTINUE                                            TAK19200
                      ELSE                                                  TAK19210
                        IGRID(I,J)=0                                        TAK19220
                        ETA(I,J)=H(I,J)+10.0                                TAK19230
                        NETA(I,J)=H(I,J)+10.0                               TAK19240
                        DO 204 K=1,JP                                       TAK19250
                             A(K,I,J)=A(K,I,J+1)                            TAK19260
                             CONTI(K,I,J)=CONTI(K,I,J+1)                    TAK19270
                             INTG1(K,I,J)=INTG1(K,I,J+1)                    TAK19280
                             INTG3(K,I,J)=INTG3(K,I,J+1)                    TAK19290
                             DO 204 L=1,JP                                  TAK19300
                                INTG2(L,K,I,J)=INTG2(L,K,I,J+1)             TAK19310
                                INTG4(L,K,I,J)=INTG4(L,K,I,J+1)             TAK19320
                                INTG5(L,K,I,J)=INTG5(L,K,I,J+1)             TAK19330
                                INTG6(L,K,I,J)=INTG6(L,K,I,J+1)             TAK19340
                                INTG7(L,K,I,J)=INTG7(L,K,I,J+1)             TAK19350
204                     CONTINUE                                            TAK19360
                      ENDIF                                                 TAK19370
                      QIF(I,J)=0.0                                          TAK19380
                    ENDIF                                                   TAK19390
*                                                                           TAK19400
                ELSE IF(IGRID(I,J-1).EQ.0) THEN                             TAK19410
*                                                                           TAK19420
                  VV=0.0                                                    TAK19430
                  DO 205 K=1,JP                                             TAK19440
                       VV=VV+CONTI(K,I,J-1)*ND(K,I-1,J-1)                   TAK19450
205               CONTINUE                                                  TAK19460
                  DEP=NETA(I,J-1)-H(I,J-1)                                  TAK19470
                  FLOW = VV*DEP*DELX*DELT                                   TAK19480
                  QJB(I,J)=QJB(I,J)+FLOW                                    TAK19490
                  VOL=0.5*DEP*DELX*DELY                                     TAK19500
                  IF(VV.GE.0.0) THEN                                        TAK19510
                    SSS(I,J-1)=SOS(I,J-1)+ABS(FLOW)/DXDY                    TAK19520
                  ELSE                                                      TAK19530
                    SSS(I,J-1)=SOS(I,J-1)-ABS(FLOW)/DXDY                    TAK19540
                  ENDIF                                                     TAK19550
                  IF( ABS(QJB(I,J)).GT.VOL) THEN                            TAK19560
                    SSS(I,J-1)=SOS(I,J-1)                                   TAK19570
                    IF(QJB(I,J).GT.0.0) THEN                                TAK19580
                      IGRID(I,J)=0                                          TAK19590
                      ETA(I,J)=H(I,J)+10.0                                  TAK19600
                      NETA(I,J)=H(I,J)+10.0                                 TAK19610
                      DO 206 K=1,JP                                         TAK19620
                           A(K,I,J)=A(K,I,J-1)                              TAK19630
                           CONTI(K,I,J)=CONTI(K,I,J-1)                      TAK19640
                           INTG1(K,I,J)=INTG1(K,I,J-1)                      TAK19650
                           INTG3(K,I,J)=INTG3(K,I,J-1)                      TAK19660
                           DO 206 L=1,JP                                    TAK19670
                               INTG2(L,K,I,J)=INTG2(L,K,I,J-1)              TAK19680
```

```
                                INTG4(L,K,I,J)=INTG4(L,K,I,J-1)          TAK19690
                                INTG5(L,K,I,J)=INTG5(L,K,I,J-1)          TAK19700
                                INTG6(L,K,I,J)=INTG6(L,K,I,J-1)          TAK19710
                                INTG7(L,K,I,J)=INTG7(L,K,I,J-1)          TAK19720
206                         CONTINUE                                     TAK19730
                          ELSE                                           TAK19740
                            IGRID(I,J-1)=2                               TAK19750
                            ETA(I,J-1)=H(I,J-1)                          TAK19760
                            NETA(I,J-1)=H(I,J-1)                         TAK19770
                            DO 207 K=1,JP                                TAK19780
                                 A(K,I,J-1)=0.0                          TAK19790
                                 CONTI(K,I,J-1)=0.0                      TAK19800
                                 INTG1(K,I,J-1)=0.0                      TAK19810
                                 INTG3(K,I,J-1)=0.0                      TAK19820
                                 DO 207 L=1,JP                           TAK19830
                                      INTG2(L,K,I,J-1)=0.0               TAK19840
                                      INTG4(L,K,I,J-1)=0.0               TAK19850
                                      INTG5(L,K,I,J-1)=0.0               TAK19860
                                      INTG6(L,K,I,J-1)=0.0               TAK19870
                                      INTG7(L,K,I,J-1)=0.0               TAK19880
207                         CONTINUE                                     TAK19890
                          ENDIF                                          TAK19900
                          QIF(I,J)=0.0                                   TAK19910
                        ENDIF                                            TAK19920
*                                                                        TAK19930
                    ENDIF                                                TAK19940
*                                                                        TAK19950
                  ENDIF                                                  TAK19960
*                                                                        TAK19970
 201    CONTINUE                                                         TAK19980
*                                                                        TAK19990
*                                                                        TAK20000
        RETURN                                                           TAK20010
        END                                                              TAK20020
```

```
      SUBROUTINE CLIMATE(SSS)                                          TAK20030
*********************************************************************  TAK20040
*                                                                   *  TAK20050
*     COMPUTES THE CHANGE IN ABLATION RATE BELOW THE                *  TAK20060
*                    EQUILIBRIUM LINE                               *  TAK20070
*                                                                   *  TAK20080
*********************************************************************  TAK20090
*                                                                      TAK20100
*                                                                      TAK20110
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                       TAK20120
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,         TAK20130
     &           IWPLS=IWIDTH+2)                                       TAK20140
      REAL   NV,NH,NC,ND,NETA,LHEAT                                    TAK20150
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                 TAK20160
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                      TAK20170
*                                                                      TAK20180
      DIMENSION SSS(LPLS,IWPLS)                                        TAK20190
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO          TAK20200
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),            TAK20210
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)           TAK20220
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)   TAK20230
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                       TAK20240
      COMMON/ BLK11/ HEQL,DELC,GRADC,GRADT,ALFA1,ALFA2,TDEL,SMIN       TAK20250
*                                                                      TAK20260
      DATA LHEAT/3.35E05/                                              TAK20270
*                                                                      TAK20280
*                                                                      TAK20290
*****......DETERMINE CHANGE IN THE ELEVATON OF EL                      TAK20300
*                                                                      TAK20310
          Z=HEQL                                                       TAK20320
          HEQL1=HEQL                                                   TAK20330
          DAY=193.6 -2.646E-01*Z +4.351E-04*Z**2 -4.472E-07*Z**3       TAK20340
          DAY=DAY+2.142E-10*Z**4 -3.946E-14*Z**5                       TAK20350
          COEF=DAY*(ALFA1+ALFA2)/LHEAT                                 TAK20360
          SUMT=-DELC+COEF*TDEL                                         TAK20370
          SUMB=GRADC-COEF*GRADT                                        TAK20380
          HDEL=SUMT/SUMB                                               TAK20390
          HEQL=HEQL+HDEL                                               TAK20400
*                                                                      TAK20410
*****.......DETERMINE ABLATION BELOW EQL.                              TAK20420
*                                                                      TAK20430
      DO 101 J=1,IWIDTH                                                TAK20440
         DO 101 I=1,LPLS                                               TAK20450
            SP=SSS(I,J)                                                TAK20460
            IF(NETA(I,J).LE.HEQL.AND.NETA(I,J).GE.HEQL1) THEN          TAK20470
               DIF=HEQL-NETA(I,J)                                      TAK20480
               SSS(I,J)=DIF*SMIN/HEQL                                  TAK20490
               IF(SP.LT.SSS(I,J)) THEN                                 TAK20500
                  SSS(I,J)=SP                                          TAK20510
               ENDIF                                                   TAK20520
            ENDIF                                                      TAK20530
            IF(NETA(I,J).LE.HEQL1.AND.NETA(I,J).GT.0.0) THEN           TAK20540
               DIF1=(HEQL1-NETA(I,J))/HEQL1                            TAK20550
               DIF2=(HEQL-NETA(I,J))/HEQL                              TAK20560
               SSS(I,J)=SSS(I,J)+(DIF2-DIF1)*SMIN                      TAK20570
               IF(SP.LT.SSS(I,J)) THEN                                 TAK20580
```

```
                    SSS(I,J)=SP                                    TAK20590
                  ENDIF                                            TAK20600
                ENDIF                                              TAK20610
 101    CONTINUE                                                   TAK20620
*                                                                   TAK20630
       RETURN                                                       TAK20640
       END                                                          TAK20650
```

```
      FUNCTION VELOB(M,N)                                                TAK20660
*****************************************************************      TAK20670
*                                                               *      TAK20680
*   CALCULATES BOTTOM VELOCITY FOR QUADRATRIC FRICTION          *      TAK20690
*                                                               *      TAK20700
*****************************************************************      TAK20710
*                                                                      TAK20720
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                       TAK20730
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,         TAK20740
     &           IWPLS=IWIDTH+2)                                       TAK20750
      REAL    NC,ND                                                    TAK20760
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),            TAK20770
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)           TAK20780
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)              TAK20790
*                                                                      TAK20800
*                                                                      TAK20810
      MM=M                                                             TAK20820
      NN=N                                                             TAK20830
      IF(M.EQ.1) MM=M+1                                                TAK20840
      IF(M.EQ.LPLS) MM=M-1                                             TAK20850
      IF(N.EQ.1) NN=N+1                                                TAK20860
      IF(N.EQ.IWPLS) NN=N-1                                            TAK20870
*                                                                      TAK20880
      SIGMA=0.                                                         TAK20890
      U=0.                                                             TAK20900
      V=0.                                                             TAK20910
      DO 101 I=1,JP                                                    TAK20920
         C1=(C(I,MM,NN-1)+C(I,MM-1,NN-1))*0.5                          TAK20930
         D1=(D(I,MM-1,NN)+D(I,MM-1,NN-1))*0.5                          TAK20940
         TRIG= COS(A(I,MM,NN)*(2.0-SIGMA)*0.5)                         TAK20950
         U=U+C1*TRIG                                                   TAK20960
         V=V+D1*TRIG                                                   TAK20970
 101  CONTINUE                                                         TAK20980
      VELOB= SQRT(U*U+V*V)                                             TAK20990
*                                                                      TAK21000
      RETURN                                                           TAK21010
      END                                                              TAK21020
```

```
      SUBROUTINE VELOC                                                  TAK21030
*****************************************************************     TAK21040
*                                                               *     TAK21050
*             COMPUTES VERTICAL VELOCITY PROFILE                *     TAK21060
*                                                               *     TAK21070
*****************************************************************     TAK21080
*                                                                     TAK21090
*                                                                     TAK21100
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                      TAK21110
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,        TAK21120
     &           IWPLS=IWIDTH+2)                                      TAK21130
*                                                                     TAK21140
      REAL    NC,ND,NETA                                              TAK21150
*                                                                     TAK21160
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO         TAK21170
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),           TAK21180
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)          TAK21190
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)  TAK21200
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),AR1(JP),AR2(JP),AA(JP),          TAK21210
     &              AI1(JP),AI3(JP),CONTI(JP,LPLS,IWPLS)              TAK21220
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                      TAK21230
      COMMON/ BLK9/ VX(KP,LPLS,IWPLS),VY(KP,LPLS,IWPLS),              TAK21240
     &              VV(KP,LPLS,IWPLS),VM(LPLS,IWPLS),DELSIG           TAK21250
      DATA FACT     /86400.0  /                                       TAK21260
*                                                                     TAK21270
*                                                                     TAK21280
       DO 100 I=1,LPLS                                                TAK21290
         DO 100 J=1,IWPLS                                             TAK21300
            DO 100 K=1,JP                                             TAK21310
              VX(K,I,J)=0.0                                           TAK21320
              VY(K,I,J)=0.0                                           TAK21330
              VV(K,I,J)=0.0                                           TAK21340
              VM(  I,J)=0.0                                           TAK21350
 100    CONTINUE                                                       TAK21360
                                                                      TAK21370
        DO 101 I=2,LPLS-1                                             TAK21380
           DO 101 J=2,IWPLS-1                                         TAK21390
             IF(IGRID(I,J).NE.0) GOTO 101                             TAK21400
*                                                                     TAK21410
*****........COMPUTE X-VELOCITY COMPONENT                            TAK21420
*                                                                     TAK21430
             SIGMA=0.0                                                TAK21440
             DO 102 L=1,KP                                            TAK21450
               VSUM=0.0                                               TAK21460
               DO 103 K=1,JP                                          TAK21470
                 AVGC=0.5*(NC(K,I-1,J-1)+NC(K,I,J-1))                 TAK21480
                 TRIG=0.5*A(K,I,J)*(2.0-SIGMA)                        TAK21490
                 TRIG= COS(TRIG)                                      TAK21500
                 VSUM=VSUM+AVGC*TRIG                                  TAK21510
 103             CONTINUE                                             TAK21520
               VX(L,I,J)=FACT*VSUM                                    TAK21530
               SIGMA=SIGMA+DELSIG                                     TAK21540
 102         CONTINUE                                                 TAK21550
*                                                                     TAK21560
*****..... COMPUTE Y-VELOCITY COMPONENT                              TAK21570
*                                                                     TAK21580
             SIGMA=0.0                                                TAK21590
             DO 104 L=1,KP                                            TAK21600
               VSUM=0.0                                               TAK21610
               DO 105 K=1,JP                                          TAK21620
                  AVGD=0.5*(ND(K,I-1,J-1)+ND(K,I-1,J))                TAK21630
```

```
                  TRIG=0.5*A(K,I,J)*(2.0-SIGMA)                          TAK21640
                  TRIG= COS(TRIG)                                        TAK21650
                  VSUM=VSUM+AVGD*TRIG                                    TAK21660
 105            CONTINUE                                                TAK21670
                VY(L,I,J)=FACT*VSUM                                     TAK21680
                SIGMA=SIGMA+DELSIG                                      TAK21690
 104          CONTINUE                                                  TAK21700
*                                                                       TAK21710
*****........COMPUTE RESULTANT VELOCITY                                 TAK21720
*                                                                       TAK21730
              DO 106 L=1,KP                                             TAK21740
                VV(L,I,J)=0.0                                           TAK21750
                IF( ABS(VX(L,I,J)).LE.1.0E-25) THEN                     TAK21760
                  VX(L,I,J)=0.0                                         TAK21770
                ENDIF                                                   TAK21780
                IF( ABS(VY(L,I,J)).LE.1.0E-25) THEN                     TAK21790
                  VY(L,I,J)=0.0                                         TAK21800
                ENDIF                                                   TAK21810
                VV(L,I,J)= SQRT(VX(L,I,J)**2+VY(L,I,J)**2)              TAK21820
 106          CONTINUE                                                  TAK21830
*                                                                       TAK21840
*****........COMPUTE MEAN VELOCITY                                      TAK21850
*                                                                       TAK21860
              VSUM=0.0                                                  TAK21870
              DO 107 K=1,JP                                             TAK21880
                AVGC=0.5*(NC(K,I-1,J-1)+NC(K,I,J-1))                    TAK21890
                IF(A(K,I,J).NE.0.0) THEN                                TAK21900
                  VSUM=VSUM+AVGC*CONTI(K,I,J)                           TAK21910
                ELSE                                                    TAK21920
                  VSUM=VSUM+AVGC                                        TAK21930
                ENDIF                                                   TAK21940
 107          CONTINUE                                                  TAK21950
              VMX=VSUM                                                  TAK21960
*                                                                       TAK21970
              VSUM=0.0                                                  TAK21980
              DO 108 K=1,JP                                             TAK21990
                AVGD=0.5*(ND(K,I-1,J-1)+NC(K,I,J+1))                    TAK22000
                IF(A(K,I,J).NE.0.0) THEN                                TAK22010
                  VSUM=VSUM+AVGD*CONTI(K,I,J)                           TAK22020
                ELSE                                                    TAK22030
                  VSUM=VSUM+AVGD                                        TAK22040
                ENDIF                                                   TAK22050
 108          CONTINUE                                                  TAK22060
              VMY=VSUM                                                  TAK22070
*                                                                       TAK22080
              IF( ABS(VMX).LE.1.0E-25) THEN                             TAK22090
                 VMX=0.0                                                TAK22100
              ENDIF                                                     TAK22110
              IF( ABS(VMY).LE.1.0E-25) THEN                             TAK22120
                VMY=0.0                                                 TAK22130
              ENDIF                                                     TAK22140
              VM(I,J)=FACT* SQRT(VMX*VMX+VMY*VMY)                       TAK22150
*                                                                       TAK22160
 101      CONTINUE                                                      TAK22170
*                                                                       TAK22180
        RETURN                                                          TAK22190
        END                                                             TAK22200
```

```
      SUBROUTINE OUTPUT(JJJ,HI,SSS,HEQL)                                TAK22210
*********************************************************************  TAK22220
*                                                                   *  TAK22230
*              PRINT AND STORE COMPUTED OUTPUTS                     *  TAK22240
*                                                                   *  TAK22250
*********************************************************************  TAK22260
*                                                                      TAK22270
*                                                                      TAK22280
      PARAMETER (JP=1,KP=21,LENGTH=54,IWIDTH=33)                       TAK22290
      PARAMETER (LENP1=LENGTH+1,IWIDP1=IWIDTH+1,LPLS=LENGTH+2,         TAK22300
     &           IWPLS=IWIDTH+2)                                       TAK22310
*                                                                      TAK22320
      CHARACTER*1 T  /Z05/                                             TAK22330
      REAL   NV,NH,NC,ND,NETA                                          TAK22340
      REAL   INTG1,INTG2,INTG3,INTG4,INTG5,INTG6,INTG7                 TAK22350
      LOGICAL       FTTOM,HOTST,HVISC,VFRICT,MOVE                      TAK22360
*                                                                      TAK22370
      DIMENSION HI(LPLS,IWPLS),SSS(LPLS,IWPLS)                         TAK22380
      DIMENSION HT(LPLS,IWPLS),VR(LPLS,IWPLS),TW(LPLS,IWPLS)           TAK22390
      COMMON/ BLK1/ H(LPLS,IWPLS),FRICT(LPLS,IWPLS),NV,NH,RHO          TAK22400
      COMMON/ BLK2/ DELT,DELX,DELY,UPAI,START,PHOTST,POUTPT,           TAK22410
     &              ISTRT,NFIN,NHOTST                                  TAK22420
      COMMON/ BLK3/ FTTOM,HOTST,HVISC,VFRICT,MOVE                      TAK22430
      COMMON/ BLK4/ C(JP,LENP1,IWIDTH),NC(JP,LENP1,IWIDTH),            TAK22440
     &              D(JP,LENGTH,IWIDP1),ND(JP,LENGTH,IWIDP1)           TAK22450
      COMMON/ BLK5/ ETA(LPLS,IWPLS),NETA(LPLS,IWPLS),SOS(LPLS,IWPLS)   TAK22460
      COMMON/ BLK6/ A(JP,LPLS,IWPLS),CONTI(JP,LPLS,IWPLS)              TAK22470
      COMMON/ BLK7/ INTG1(JP,LPLS,IWPLS),INTG2(JP,JP,LPLS,IWPLS),      TAK22480
     &              INTG3(JP,LPLS,IWPLS),INTG4(JP,JP,LPLS,IWPLS),      TAK22490
     &              INTG5(JP,JP,LPLS,IWPLS),INTG6(JP,JP,LPLS,IWPLS),   TAK22500
     &              INTG7(JP,JP,LPLS,IWPLS)                            TAK22510
      COMMON/ BLK8/ IGRID(LPLS,IWPLS),XI(4),W(4)                       TAK22520
      COMMON/ BLK9/ VX(KP,LPLS,IWPLS),VY(KP,LPLS,IWPLS),               TAK22530
     &              VV(KP,LPLS,IWPLS),VM(LPLS,IWPLS),DELSIG            TAK22540
      COMMON/ BLK10/ QIF(LPLS,IWPLS),QIB(LPLS,IWPLS),                  TAK22550
     &              QJF(LPLS,IWPLS),QJB(LPLS,IWPLS)                    TAK22560
*                                                                      TAK22570
      DATA ISKIP        / 240 /                                        TAK22580
      DATA ROW          / 900.0/                                       TAK22590
      DATA FACT         / 3.17097E-8/                                  TAK22600
*                                                                      TAK22610
*                                                                      TAK22620
*****.......PRINT SURFACE ELEVATION ON THE SCREEN                      TAK22630
*                                                                      TAK22640
      WRITE( 3,1001)       NETA(9,23),  NETA(23,12),  NETA(21,20),     TAK22650
     &       NETA(28,20),  NETA(34,20),  NETA(30,26),  NETA(37,26),    TAK22660
     &       NETA(45,29)                                               TAK22670
1001  FORMAT(1X,   8(   F9.2))                                         TAK22680
*                                                                      TAK22690
*                                                                      TAK22700
*****.........COMPUTE VELOCITY                                        TAK22710
*                                                                      TAK22720
      CALL VELOC                                                       TAK22730
*                                                                      TAK22740
*****.........PRINT SURFACE ELEVATION AT SELECTED LOCATIONS           TAK22750
*                                                                      TAK22760
      WRITE(11,1002) JJJ,T,NETA(9,23),T,NETA(23,12),T,NETA(21,20),     TAK22770
     &    T,NETA(28,20),T,NETA(34,20),T,NETA(30,26),T,NETA(37,26),     TAK22780
     &    T,NETA(45,29)                                                TAK22790
1002  FORMAT(1X,I5,8(A1,F9.2))                                         TAK22800
*                                                                      TAK22810
```

```
*                                                                       TAK22820
*****.........PRINT SURFACE VELOCITY AT SELECTED LOCATIONS              TAK22830
*                                                                       TAK22840
         WRITE(12,1003) JJJ,T,VV(KP,9,23),T,VV(KP,23,12),T,            TAK22850
     &                  VV(KP,21,20),T,VV(KP,28,20),T,VV(KP,34,20),T,  TAK22860
     &                  VV(KP,30,26),T,VV(KP,37,26),T,VV(KP,45,29)     TAK22870
1003     FORMAT(1X,I5,8(A1,F9.5))                                       TAK22880
*                                                                       TAK22890
*                                                                       TAK22900
*****......COMPUTE AND PRINT DIFFERENT PARAMETERS OF THE GLACIER        TAK22910
*                                                                       TAK22920
         VOL=0.0                                                        TAK22930
         ACC=0.0                                                        TAK22940
         ABL=0.0                                                        TAK22950
         SPO=0.0                                                        TAK22960
         SNE=0.0                                                        TAK22970
         DO 101 I=1,LPLS                                                TAK22980
           DO 101 J=1,IWPLS                                             TAK22990
            IF(IGRID(I,J).EQ.0) THEN                                    TAK23000
              DIF=NETA(I,J)-H(I,J)                                      TAK23010
              VOL=VOL+DIF                                               TAK23020
              IF(SSS(I,J).GT.0.0) THEN                                  TAK23030
                ACC=ACC+1.0                                             TAK23040
                SPO=SPO+SSS(I,J)                                        TAK23050
              ELSE                                                      TAK23060
                ABL=ABL+1.0                                             TAK23070
                SNE=SNE+SSS(I,J)                                        TAK23080
              ENDIF                                                     TAK23090
            ENDIF                                                       TAK23100
101      CONTINUE                                                       TAK23110
         VOL=0.001*VOL                                                  TAK23120
         SPO=SPO/(FACT*DELT)                                            TAK23130
         SNE=SNE/(FACT*DELT)                                            TAK23140
         WRITE(15,1004) VOL,T,ACC,T,ABL,T,SPO,T,SNE,T,HEQL              TAK23150
 1004    FORMAT(1X,6(F10.2,A1))                                         TAK23160
*                                                                       TAK23170
*                                                                       TAK23180
        KK=JJJ/ISKIP                                                    TAK23190
        IF((KK*ISKIP-JJJ).EQ.0) THEN                                    TAK23200
*                                                                       TAK23210
*****......VERTICAL VELOCITY AT SELECTED LOCATIONS                      TAK23220
*                                                                       TAK23230
        SIGMA=0.0                                                       TAK23240
        DO 102 K=1,KP                                                   TAK23250
           WRITE(13,1005) SIGMA,T,VV(K,9,23),T,VV(K,23,12),T,           TAK23260
     &                    VV(K,21,20),T,VV(K,28,20),T,VV(K,34,20),T,    TAK23270
     &                    VV(K,30,26),T,VV(K,37,26),T,VV(K,45,29)       TAK23280
 1005      FORMAT(1X,9(F8.5,A1))                                        TAK23290
           SIGMA=SIGMA+DELSIG                                           TAK23300
 102    CONTINUE                                                        TAK23310
*                                                                       TAK23320
*                                                                       TAK23330
*****.....DETERMINE AND PRINT FLOW PARAMETERS OF THE GLACIER            TAK23340
*                                                                       TAK23350
        HTMAX=-99999.0                                                  TAK23360
        HTMIN=+99999.0                                                  TAK23370
        TWMAX=-99999.0                                                  TAK23380
        TWMIN=+99999.0                                                  TAK23390
        VRMAX=-99999.0                                                  TAK23400
        VRMIN=+99999.0                                                  TAK23410
        VSMAX=-99999.0                                                  TAK23420
```

```
      VSMIN=+99999.0                                                    TAK23430
      VBMAX=-99999.0                                                    TAK23440
      VBMIN=+999.0                                                      TAK23450
      DO 103 J=1,IWPLS                                                  TAK23460
         DO 103 I=1,LPLS                                                TAK23470
            IF(IGRID(I,J).EQ.0) THEN                                    TAK23480
              HT(I,J)=NETA(I,J)-HI(I,J)                                 TAK23490
              VR(I,J)=VV(1,I,J)/VV(KP,I,J)                              TAK23500
              TW(I,J)=FRICT(I,J)*ROW*VV(1,I,J)                          TAK23510
              TW(I,J)=1.1574E-8*TW(I,J)                                 TAK23520
              HTMAX=AMAX1(HTMAX,HT(I,J))                                TAK23530
              HTMIN=AMIN1(HTMIN,HT(I,J))                                TAK23540
              TWMAX=AMAX1(TWMAX,TW(I,J))                                TAK23550
              TWMIN=AMIN1(TWMIN,TW(I,J))                                TAK23560
              VRMAX=AMAX1(VRMAX,VR(I,J))                                TAK23570
              VRMIN=AMIN1(VRMIN,VR(I,J))                                TAK23580
              VSMAX=AMAX1(VSMAX,VV(KP,I,J))                             TAK23590
              VSMIN=AMIN1(VSMIN,VV(KP,I,J))                             TAK23600
              VBMAX=AMAX1(VBMAX,VV(1,I,J))                              TAK23610
              VBMIN=AMIN1(VBMIN,VV(1,I,J))                              TAK23620
             ENDIF                                                      TAK23630
 103   CONTINUE                                                         TAK23640
*                                                                       TAK23650
      WRITE(14,*) 'HTMAX & HTMIN:  ',HTMAX,HTMIN                        TAK23660
      DO 104 J=1,IWPLS                                                  TAK23670
          WRITE(14,1006) (HT(I,J),I=1,LPLS)                             TAK23680
 1006     FORMAT((9F7.2))                                               TAK23690
 104   CONTINUE                                                         TAK23700
      WRITE(14,*) 'TWMAX & TWMIN:  ',TWMAX,TWMIN                        TAK23710
      DO 105 J=1,IWPLS                                                  TAK23720
         WRITE(14,1006) (TW(I,J),I=1,LPLS)                              TAK23730
 105   CONTINUE                                                         TAK23740
      WRITE(14,*) 'VRMAX & VRMIN:  ',VRMAX,VRMIN                        TAK23750
      DO 106 J=1,IWPLS                                                  TAK23760
        WRITE(14,1006) (VR(I,J),I=1,LPLS)                               TAK23770
 106   CONTINUE                                                         TAK23780
      WRITE(14,*) 'VSMAX & VSMIN:  ',VSMAX,VSMIN                        TAK23790
      DO 107 J=1,IWPLS                                                  TAK23800
        WRITE(14,1006) (VV(KP,I,J),I=1,LPLS)                            TAK23810
 107   CONTINUE                                                         TAK23820
      WRITE(14,*) 'VBMAX & VBMIN:  ',VBMAX,VBMIN                        TAK23830
      DO 108 J=1,IWPLS                                                  TAK23840
        WRITE(14,1006) (VV(1,I,J),I=1,LPLS)                             TAK23850
 108   CONTINUE                                                         TAK23860
*                                                                       TAK23870
      REWIND(14)                                                        TAK23880
      ENDIF                                                             TAK23890
*                                                                       TAK23900
      RETURN                                                            TAK23910
*                                                                       TAK23920
*****.....WRITE OUTPUT FOR HOT START                                    TAK23930
*                                                                       TAK23940
      ENTRY OUPT(JJJ,HI,SSS)                                            TAK23950
*                                                                       TAK23960
       WRITE(3,2001) JJJ                                                TAK23970
 2001  FORMAT(' ',T10,'WRITING HOT START DATA....',I5 )                 TAK23980
        REWIND(2)                                                       TAK23990
        WRITE(2) NHOTST                                                 TAK24000
        WRITE(2) IGRID                                                  TAK24010
        WRITE(2) A                                                      TAK24020
        WRITE(2) ETA                                                    TAK24030
```

```
      WRITE(2) NETA                                                    TAK24040
      WRITE(2) C                                                       TAK24050
      WRITE(2) NC                                                      TAK24060
      WRITE(2) D                                                       TAK24070
      WRITE(2) ND                                                      TAK24080
      WRITE(2) SSS                                                     TAK24090
      WRITE(2) FRICT                                                   TAK24100
      WRITE(2) QIF                                                     TAK24110
      WRITE(2) QIB                                                     TAK24120
      WRITE(2) QJF                                                     TAK24130
      WRITE(2) QJB                                                     TAK24140
*                                                                      TAK24150
      RETURN                                                           TAK24160
      END                                                              TAK24170
```

APPENDIX-B: INPUT DATA FOR TAKU FORTRAN

```
&LOGCL      HOTST =.F.,
            FTTOM =.F.,
            HVISC =.F.,
            VFRICT=.T.,
            MOVE  =.T.,
            CLIMET=.T.,
            PRT   =.F.,                                                    &END
&SYSTEM H=
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,  1270.0,  1300.0,  1360.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,  1260.0,  1260.0,  1300.0,  1310.0,  1335.0,
1360.0,  1360.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,  1320.0,  1260.0,  1230.0,  1250.0,  1290.0,  1310.0,
1330.0,  1360.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,  1330.0,  1260.0,  1260.0,  1270.0,  1290.0,  1290.0,  1300.0,
1350.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,  1330.0,  1255.0,  1255.0,  1260.0,  1280.0,  1430.0,  1400.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,  1350.0,  1280.0,  1230.0,  1250.0,  1260.0,  1280.0,  1290.0,  1300.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
```

```
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1330.0, 1250.0, 1220.0, 1230.0, 1260.0, 1270.0, 1290.0, 1300.0,
   0.0, 1300.0, 1350.0, 1380.0, 1400.0, 1450.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1400.0, 1240.0, 1215.0, 1220.0, 1260.0, 1270.0, 1280.0,
1290.0, 1290.0, 1300.0, 1320.0, 1330.0, 1350.0, 1360.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1300.0, 1225.0, 1220.0, 1210.0, 1230.0, 1260.0, 1270.0, 1280.0,
1290.0, 1290.0, 1300.0, 1310.0, 1320.0, 1340.0, 1350.0, 1350.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1300.0, 1220.0, 1190.0, 1190.0, 1190.0, 1220.0, 1240.0, 1260.0,
1280.0, 1290.0, 1310.0, 1320.0, 1340.0, 1340.0, 1350.0, 1350.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1300.0, 1180.0, 1170.0, 1160.0, 1170.0, 1210.0,    0.0, 1330.0,
1300.0, 1300.0, 1320.0, 1340.0, 1400.0, 1440.0, 1550.0, 1350.0, 1350.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1300.0, 1220.0, 1230.0, 1210.0,
1190.0, 1180.0, 1160.0, 1130.0, 1120.0, 1120.0, 1130.0,    0.0, 1400.0,
1370.0, 1380.0,    0.0,    0.0,    0.0,    0.0, 1400.0, 1400.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1300.0, 1220.0, 1230.0, 1200.0,
1170.0, 1130.0, 1110.0, 1090.0, 1070.0, 1050.0, 1090.0, 1250.0, 1280.0,
1320.0, 1320.0, 1320.0, 1300.0, 1290.0, 1280.0, 1280.0,    0.0, 1240.0,
1280.0, 1230.0, 1220.0, 1220.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1140.0,
1100.0, 1060.0, 1020.0,  970.0,  940.0,  930.0,  940.0,  960.0,  990.0,
1010.0, 1020.0, 1040.0, 1040.0, 1050.0, 1070.0, 1090.0,    0.0, 1120.0,
1090.0, 1070.0, 1030.0,  980.0,  990.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1220.0, 1180.0, 1130.0,
1090.0, 1060.0, 1030.0,  980.0,  950.0,  910.0,  890.0,  890.0,  900.0,
 910.0,  920.0,  940.0,  980.0, 1060.0, 1140.0,    0.0, 1080.0, 1010.0,
1050.0,  970.0,  960.0,  970.0,  990.0,    0.0,    0.0,    0.0,    0.0,
```

```
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1230.0, 1250.0, 1250.0,
   0.0, 1240.0, 1065.0,  980.0,  940.0,  900.0,  870.0,  860.0,  870.0,
 900.0,  920.0, 1000.0,    0.0, 1110.0, 1110.0,    0.0, 1000.0,  940.0,
 910.0,  900.0,  950.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1380.0, 1380.0,    0.0,
   0.0, 1040.0, 1020.0,  980.0,  930.0,  900.0,  860.0,  850.0,  850.0,
 860.0,  860.0,  890.0,  920.0,    0.0, 1000.0,  900.0,  870.0,  880.0,
 900.0,  970.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0, 1660.0, 1650.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1700.0, 1600.0, 1540.0, 1500.0, 1460.0, 1390.0, 1400.0, 1400.0,
1170.0, 1140.0, 1060.0, 1010.0,  970.0,  950.0,  830.0,  800.0,  760.0,
 740.0,  730.0,  770.0,    0.0,  930.0,  960.0,  950.0,  950.0, 1050.0,
1080.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0, 1610.0, 1600.0, 1550.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1690.0, 1560.0, 1680.0,    0.0, 1470.0, 1360.0, 1270.0, 1220.0,
1110.0, 1080.0, 1050.0, 1030.0, 1080.0,    0.0,  960.0,  890.0,  770.0,
 740.0,  710.0,  700.0,  780.0,  800.0,  850.0,  910.0,  970.0,  990.0,
1020.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0, 1760.0, 1620.0, 1610.0, 1700.0, 1720.0,    0.0,    0.0,
1550.0, 1530.0, 1500.0, 1470.0, 1450.0, 1400.0, 1360.0, 1210.0, 1150.0,
1110.0, 1080.0, 1140.0,    0.0,    0.0, 1040.0, 1000.0, 1060.0,  900.0,
 820.0,  730.0,  700.0,  700.0,  720.0,  740.0,  850.0,  890.0,  900.0,
 900.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0, 1690.0, 1600.0, 1660.0, 1640.0, 1580.0, 1560.0,
1540.0, 1520.0, 1500.0, 1420.0, 1340.0, 1310.0, 1330.0, 1190.0, 1150.0,
1110.0, 1100.0, 1200.0,    0.0,    0.0, 1020.0, 1040.0,    0.0,  880.0,
 830.0,  800.0,  700.0,  650.0,  690.0,  730.0,  860.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0, 1590.0, 1550.0, 1560.0, 1540.0, 1530.0, 1520.0,
1550.0, 1500.0, 1450.0, 1420.0, 1340.0, 1310.0, 1330.0, 1200.0, 1210.0,
1270.0, 1260.0, 1270.0, 1300.0, 1220.0, 1100.0, 1060.0,    0.0,  850.0,
   0.0,  750.0,  700.0,  600.0,  550.0,  530.0,  560.0,  540.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0, 1530.0, 1530.0, 1520.0, 1590.0, 1660.0,
1670.0, 1600.0, 1630.0, 1680.0, 1520.0, 1390.0, 1360.0,    0.0,    0.0,
   0.0, 1350.0, 1420.0, 1420.0, 1380.0, 1275.0, 1340.0,    0.0,    0.0,
   0.0,    0.0,  700.0,  600.0,  530.0,  520.0,  500.0,  480.0,  470.0,
 450.0,  500.0,  435.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1600.0, 1600.0, 1650.0, 1770.0,
1830.0, 1610.0, 1600.0, 1630.0, 1650.0,    0.0, 1390.0, 1370.0, 1230.0,
   0.0, 1220.0, 1250.0, 1260.0, 1280.0, 1280.0, 1120.0, 1100.0, 1070.0,
```

```
 750.0,  660.0,  570.0,  530.0,  520.0,  500.0,  480.0,  470.0,  450.0,
 410.0,  370.0,  340.0,  300.0,  295.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1690.0, 1620.0, 1600.0,    0.0,    0.0, 1320.0, 1180.0,
1210.0, 1140.0, 1120.0, 1090.0, 1010.0, 1020.0, 1030.0, 1060.0,  975.0,
 760.0,  640.0,  580.0,  540.0,  530.0,  510.0,  490.0,  480.0,  450.0,
 410.0,  360.0,  340.0,  290.0,  290.0,  280.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1430.0, 1350.0, 1370.0,    0.0, 1230.0, 1180.0, 1150.0,
1100.0, 1100.0, 1070.0, 1050.0, 1010.0,  980.0,  970.0, 1000.0, 1000.0,
 760.0,  640.0,  590.0,  560.0,  550.0,  530.0,  510.0,  500.0,  470.0,
 420.0,  370.0,  360.0,  310.0,  290.0,  250.0,  210.0,  170.0,   30.0,
 -20.0,  -20.0,  -20.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1320.0, 1300.0, 1260.0, 1220.0, 1190.0, 1170.0, 1130.0,
1220.0, 1080.0, 1050.0, 1000.0,  990.0,  970.0,  920.0,  870.0,  800.0,
 760.0,  740.0,  730.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,  290.0,  280.0,  230.0,  160.0,   90.0,    0.5,
 -20.0,  -20.0,  -20.0,  -30.0,  -20.0,  -27.0,  -34.0,  -41.0,  -27.0,
 -13.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1320.0, 1290.0, 1270.0, 1230.0, 1200.0, 1180.0, 1200.0,
   0.0, 1210.0, 1050.0, 1080.0, 1030.0, 1000.0,  930.0,  880.0,  810.0,
 760.0,  780.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,  250.0,  220.0,  170.0,  120.0,   90.0,   70.0,
   0.5,  -20.0,  -20.0,  -20.0,  -20.0,  -30.0,  -37.0,  -44.0,  -51.0,
 -36.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0, 1450.0, 1300.0, 1220.0, 1240.0, 1390.0,
1370.0, 1230.0, 1170.0, 1150.0,    0.0, 1100.0,  995.0,  970.0,  850.0,
 770.0,  790.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,  200.0,  160.0,  120.0,   80.0,   90.0,
  80.0,   50.0,  -20.0,  -20.0,  -20.0,  -20.0,  -10.0,   -1.0,   10.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1390.0, 1300.0, 1300.0, 1340.0,
1340.0, 1300.0, 1200.0,    0.0,    0.0, 1070.0,  980.0,  980.0,  860.0,
 790.0,  830.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,  -30.0,  -20.0,  -20.0,  -27.0,  -34.0,  -41.0,  -48.0,
 -55.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1340.0, 1230.0, 1280.0, 1200.0, 1030.0, 1020.0,  900.0,  890.0,
 820.0,  850.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,  -30.0,  -20.0,  -27.0,  -34.0,  -41.0,  -53.0,  -55.0,
 -62.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1200.0, 1200.0,  990.0,  990.0,    0.0,    0.0,  900.0,  830.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,   30.0,  -20.0,  -30.0,  -20.0,   10.0,   20.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
```

```
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,  1070.0,   990.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,   -10.0,   -20.0,   -30.0,   -20.0,   -10.0,    10.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
 ETA=
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,  1530.0,  1550.0,  1600.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,  1500.0,  1500.0,  1530.0,  1555.0,  1570.0,
1600.0,  1610.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,  1530.0,  1500.0,  1480.0,  1500.0,  1530.0,  1560.0,
1590.0,  1600.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,  1540.0,  1480.0,  1480.0,  1490.0,  1500.0,  1520.0,  1540.0,
1560.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,  1530.0,  1470.0,  1470.0,  1480.0,  1530.0,  1600.0,  1570.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,  1520.0,  1440.0,  1450.0,  1460.0,  1480.0,  1500.0,  1500.0,  1530.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
   0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
```

```
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1500.0, 1430.0, 1440.0, 1450.0, 1470.0, 1470.0, 1500.0, 1520.0,
   0.0, 1510.0, 1520.0, 1540.0, 1560.0, 1580.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1550.0, 1450.0, 1420.0, 1430.0, 1450.0, 1460.0, 1490.0,
1500.0, 1510.0, 1520.0, 1540.0, 1560.0, 1580.0, 1600.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1470.0, 1410.0, 1370.0, 1370.0, 1380.0, 1410.0, 1460.0, 1490.0,
1500.0, 1515.0, 1530.0, 1550.0, 1560.0, 1560.0, 1580.0, 1600.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1440.0, 1400.0, 1360.0, 1350.0, 1360.0, 1390.0, 1410.0, 1450.0,
1475.0, 1490.0, 1520.0, 1540.0, 1550.0, 1560.0, 1570.0, 1590.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1440.0, 1380.0, 1330.0, 1320.0, 1340.0, 1370.0,    0.0, 1520.0,
1480.0, 1500.0, 1530.0, 1550.0, 1560.0, 1580.0, 1650.0, 1590.0, 1600.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1410.0, 1390.0, 1360.0, 1350.0,
1350.0, 1360.0, 1350.0, 1310.0, 1280.0, 1280.0, 1300.0,    0.0, 1510.0,
1480.0, 1520.0,    0.0,    0.0,    0.0,    0.0, 1530.0, 1550.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1410.0, 1390.0, 1360.0, 1340.0,
1320.0, 1300.0, 1280.0, 1260.0, 1220.0, 1230.0, 1280.0, 1400.0, 1440.0,
1450.0, 1450.0, 1450.0, 1430.0, 1430.0, 1420.0, 1420.0,    0.0, 1380.0,
1410.0, 1390.0, 1380.0, 1380.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1290.0,
1270.0, 1250.0, 1230.0, 1200.0, 1170.0, 1170.0, 1170.0, 1170.0, 1190.0,
1190.0, 1190.0, 1200.0, 1200.0, 1220.0, 1240.0, 1260.0,    0.0, 1290.0,
1240.0, 1220.0, 1200.0, 1180.0, 1190.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1380.0, 1320.0, 1280.0,
1280.0, 1270.0, 1270.0, 1220.0, 1200.0, 1170.0, 1130.0, 1140.0, 1140.0,
```

```
1140.0, 1150.0, 1170.0, 1190.0, 1240.0, 1300.0,    0.0, 1250.0, 1200.0,
1240.0, 1180.0, 1170.0, 1180.0, 1190.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1380.0, 1350.0, 1350.0,
   0.0, 1350.0, 1275.0, 1220.0, 1180.0, 1150.0, 1120.0, 1120.0, 1120.0,
1140.0, 1170.0, 1200.0,    0.0, 1280.0, 1280.0,    0.0, 1170.0, 1140.0,
1120.0, 1120.0, 1180.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0, 1550.0, 1550.0,    0.0,
   0.0, 1280.0, 1270.0, 1230.0, 1200.0, 1150.0, 1120.0, 1100.0, 1100.0,
1100.0, 1080.0, 1110.0, 1130.0,    0.0, 1160.0, 1110.0, 1090.0, 1100.0,
1130.0, 1200.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0, 1850.0, 1840.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1850.0, 1825.0, 1750.0, 1700.0, 1650.0, 1560.0, 1530.0, 1530.0,
1400.0, 1360.0, 1280.0, 1240.0, 1200.0, 1180.0, 1220.0, 1090.0, 1070.0,
1060.0, 1050.0, 1060.0,    0.0, 1150.0, 1060.0, 1080.0, 1090.0, 1130.0,
1150.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0, 1830.0, 1810.0, 1780.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1840.0, 1790.0, 1840.0,    0.0, 1660.0, 1560.0, 1480.0, 1420.0,
1350.0, 1330.0, 1310.0, 1270.0, 1270.0,    0.0, 1160.0, 1130.0, 1070.0,
1060.0, 1050.0, 1040.0, 1040.0, 1030.0, 1050.0, 1090.0, 1120.0, 1130.0,
1140.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0, 1900.0, 1830.0, 1820.0, 1870.0, 1860.0,    0.0,    0.0,
1780.0, 1760.0, 1700.0, 1680.0, 1650.0, 1600.0, 1530.0, 1400.0, 1380.0,
1340.0, 1330.0, 1330.0,    0.0,    0.0, 1220.0, 1190.0, 1160.0, 1100.0,
1070.0, 1050.0, 1020.0, 1000.0, 1000.0, 1000.0, 1040.0, 1060.0, 1070.0,
1080.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0, 1870.0, 1810.0, 1850.0, 1840.0, 1800.0, 1780.0,
1760.0, 1720.0, 1680.0, 1630.0, 1570.0, 1540.0, 1490.0, 1400.0, 1380.0,
1350.0, 1340.0, 1390.0,    0.0,    0.0, 1220.0, 1220.0,    0.0, 1090.0,
1050.0, 1040.0, 1020.0,  980.0,  980.0,  980.0, 1000.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0, 1830.0, 1790.0, 1790.0, 1780.0, 1770.0, 1760.0,
1740.0, 1720.0, 1670.0, 1630.0, 1570.0, 1540.0, 1490.0, 1420.0, 1410.0,
1440.0, 1420.0, 1420.0, 1435.0, 1440.0, 1300.0, 1280.0,    0.0, 1040.0,
   0.0, 1000.0,  980.0,  960.0,  930.0,  920.0,  890.0,  860.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0, 1790.0, 1780.0, 1780.0, 1820.0, 1860.0,
1880.0, 1810.0, 1850.0, 1870.0, 1725.0, 1550.0, 1480.0,    0.0,    0.0,
   0.0, 1480.0, 1460.0, 1475.0, 1480.0, 1425.0, 1460.0,    0.0,    0.0,
   0.0,    0.0,  960.0,  940.0,  930.0,  900.0,  880.0,  860.0,  840.0,
 800.0,  770.0,  745.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1800.0, 1800.0, 1830.0, 1900.0,
```

```
1950.0, 1840.0, 1830.0, 1850.0, 1870.0,    0.0, 1540.0, 1530.0, 1460.0,
   0.0, 1450.0, 1450.0, 1450.0, 1460.0, 1460.0, 1350.0, 1340.0, 1240.0,
1000.0,  970.0,  950.0,  940.0,  920.0,  900.0,  880.0,  860.0,  820.0,
 780.0,  740.0,  720.0,  700.0,  680.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1900.0, 1860.0, 1890.0,    0.0,    0.0, 1540.0, 1420.0,
1420.0, 1380.0, 1350.0, 1300.0, 1230.0, 1230.0, 1240.0, 1270.0, 1165.0,
1010.0,  970.0,  950.0,  930.0,  900.0,  880.0,  860.0,  850.0,  820.0,
 780.0,  740.0,  720.0,  700.0,  660.0,  650.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1620.0, 1560.0, 1560.0,    0.0, 1450.0, 1410.0, 1370.0,
1350.0, 1340.0, 1290.0, 1250.0, 1220.0, 1200.0, 1180.0, 1170.0, 1150.0,
1150.0, 1000.0,  960.0,  950.0,  920.0,  900.0,  880.0,  870.0,  840.0,
 800.0,  740.0,  700.0,  660.0,  620.0,  575.0,  535.0,  480.0,  420.0,
 380.0,  330.0,  -20.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1540.0, 1510.0, 1480.0, 1450.0, 1430.0, 1400.0, 1360.0,
1340.0, 1330.0, 1280.0, 1230.0, 1210.0, 1190.0, 1150.0, 1120.0, 1090.0,
1060.0, 1020.0, 1000.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,  610.0,  600.0,  550.0,  490.0,  440.0,  410.0,
 380.0,  350.0,  -20.0,  -30.0,  -20.0,  -27.0,  -34.0,  -41.0,  -27.0,
 -13.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0, 1530.0, 1490.0, 1480.0, 1450.0, 1430.0, 1410.0, 1410.0,
   0.0, 1370.0, 1280.0, 1240.0, 1220.0, 1220.0, 1160.0, 1120.0, 1090.0,
1060.0, 1030.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,  580.0,  550.0,  500.0,  460.0,  440.0,  420.0,
 370.0,  330.0,  -20.0,  -20.0,  -20.0,  -30.0,  -37.0,  -44.0,  -51.0,
 -36.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0, 1530.0, 1480.0, 1450.0, 1440.0, 1530.0,
1530.0, 1410.0, 1360.0, 1350.0,    0.0, 1270.0, 1185.0, 1130.0, 1100.0,
1070.0, 1040.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,  530.0,  480.0,  450.0,  430.0,  420.0,
 380.0,  275.0,  -20.0,  -20.0,  -20.0,  -20.0,  -10.0,   -1.0,   10.0,
   0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0, 1530.0, 1480.0, 1480.0, 1510.0,
1510.0, 1500.0, 1380.0,    0.0,    0.0, 1220.0, 1170.0, 1170.0, 1120.0,
1100.0, 1100.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,  -30.0,  -20.0,  -20.0,  -27.0,  -34.0,  -41.0,  -48.0,
 -55.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1520.0, 1430.0, 1370.0, 1370.0, 1220.0, 1200.0, 1165.0, 1140.0,
1110.0, 1120.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,  -30.0,  -20.0,  -27.0,  -34.0,  -41.0,  -53.0,  -55.0,
 -62.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0, 1400.0, 1390.0, 1180.0, 1180.0,    0.0,    0.0, 1150.0, 1120.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
   0.0,    0.0,   30.0,  -20.0,  -30.0,  -20.0,   10.0,   20.0,    0.0,
```

```
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0, 1270.0, 1190.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,  -10.0,  -20.0,  -30.0,  -20.0,  -10.0,   10.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
SOS=
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    2.2,    2.3,    2.3,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    2.2,    2.2,    2.2,    2.2,    2.2,
  2.3,    2.3,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    2.2,    2.1,    2.1,    2.2,    2.2,    2.2,
  2.3,    2.3,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    2.3,    2.2,    2.1,    2.1,    2.1,    2.2,    2.2,
  2.2,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    2.3,    2.1,    2.1,    2.1,    2.2,    2.3,    2.2,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    2.2,    2.1,    2.0,    2.0,    2.0,    2.1,    2.1,    2.1,
```

```
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  2.2,  2.1,  2.0,  2.0,  2.1,  2.1,  2.1,  2.1,
0.0,  2.2,  2.2,  2.3,  2.3,  2.3,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  2.2,  2.0,  2.0,  2.0,  2.1,  2.1,  2.1,
2.2,  2.2,  2.2,  2.2,  2.3,  2.3,  2.3,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  2.1,  2.0,  2.0,  2.0,  2.0,  2.1,  2.1,  2.2,
2.2,  2.2,  2.2,  2.2,  2.3,  2.3,  2.3,  2.3,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  2.2,  2.1,  2.0,  2.0,  2.0,  2.0,  2.0,  2.2,
2.2,  2.2,  2.2,  2.2,  2.2,  2.2,  2.2,  2.3,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  2.1,  1.9,  1.9,  1.9,  1.9,  2.0,  0.0,  2.1,
2.1,  2.1,  2.2,  2.2,  2.3,  2.4,  2.6,  2.3,  2.3,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  2.1,  2.1,  2.0,  2.0,
2.0,  2.0,  2.0,  1.9,  1.9,  1.9,  2.0,  0.0,  2.2,
2.2,  2.4,  0.0,  0.0,  0.0,  0.0,  2.4,  2.4,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  2.1,  2.1,  2.0,  2.0,
2.0,  2.0,  1.9,  1.8,  1.8,  1.8,  1.9,  1.9,  2.0,
2.1,  2.1,  2.2,  2.2,  2.2,  2.5,  2.5,  0.0,  2.0,
2.0,  2.0,  2.0,  2.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  1.8,
1.8,  1.8,  1.8,  1.7,  1.6,  1.6,  1.6,  1.6,  1.7,
1.7,  1.7,  1.8,  1.8,  1.9,  1.9,  1.9,  0.0,  1.8,
1.7,  1.7,  1.7,  1.7,  1.7,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
0.0,  0.0,
0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
```

```
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    1.8,    1.8,    1.8,
1.7,    1.7,    1.6,    1.6,    1.6,    1.6,    1.6,    1.6,    1.6,
1.6,    1.7,    1.7,    1.8,    1.8,    1.8,    0.0,    1.9,    1.9,
1.8,    1.7,    1.7,    1.7,    1.7,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    2.0,    1.8,    1.8,
0.0,    1.8,    1.8,    1.7,    1.6,    1.6,    1.5,    1.5,    1.5,
1.7,    1.7,    1.9,    0.0,    2.0,    2.0,    0.0,    1.7,    1.7,
1.7,    1.7,    1.7,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    2.0,    1.9,    0.0,
0.0,    1.8,    1.8,    1.7,    1.7,    1.6,    1.6,    1.6,    1.6,
1.6,    1.6,    1.6,    1.7,    0.0,    1.7,    1.6,    1.6,    1.6,
1.7,    1.8,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    2.2,    2.2,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    2.5,    2.4,    2.3,    2.2,    2.2,    2.2,    2.2,    2.2,
2.1,    2.0,    1.9,    1.8,    1.8,    1.7,    1.6,    1.6,    1.4,
1.2,    1.1,    1.2,    0.0,    1.4,    1.2,    1.3,    1.4,    1.4,
1.4,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    2.2,    2.2,    2.2,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    2.5,    2.4,    2.4,    0.0,    2.2,    2.2,    2.0,    1.9,
1.9,    1.9,    1.9,    1.8,    1.8,    0.0,    1.5,    1.4,    1.3,
1.2,    1.0,    0.8,    0.8,    0.9,    1.0,    1.2,    1.3,    1.4,
1.4,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    2.3,    2.2,    2.2,    2.4,    2.4,    0.0,    0.0,
2.3,    2.3,    2.2,    2.2,    2.1,    2.1,    2.0,    1.9,    1.9,
1.8,    1.8,    1.8,    0.0,    0.0,    1.7,    1.7,    1.2,    1.4,
1.2,    1.0,    0.8,    0.7,    0.7,    0.6,    0.9,    1.2,    1.3,
1.3,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    2.2,    2.2,    2.4,    2.4,    2.4,    2.3,
2.3,    2.3,    2.1,    2.0,    2.0,    2.0,    1.9,    1.9,    1.9,
1.8,    1.8,    1.8,    0.0,    0.0,    1.7,    1.7,    0.0,    1.0,
1.0,    1.0,    0.8,    0.5,    0.4,    0.4,    0.6,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    2.0,    2.0,    2.3,    2.3,    2.3,    2.3,
2.4,    2.3,    2.2,    2.0,    2.0,    2.0,    1.9,    1.9,    1.9,
1.9,    1.9,    2.0,    2.0,    1.9,    1.8,    1.8,    0.0,    0.9,
0.0,    0.6,    0.4,    0.2,    0.0,   -0.2,   -0.8,   -1.2,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    2.0,    2.0,    2.1,    2.4,    2.4,
2.4,    2.3,    2.3,    2.4,    2.4,    2.2,    2.1,    0.0,    0.0,
0.0,    2.0,    2.0,    2.0,    1.9,    1.9,    1.9,    0.0,    0.0,
0.0,    0.0,    0.7,    0.2,    0.0,   -0.4,   -0.6,   -0.8,   -1.3,
-2.0,   -2.4,   -2.4,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
```

```
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    2.0,    2.0,    2.0,    2.2,
  2.4,    2.3,    2.3,    2.3,    2.3,    0.0,    2.0,    2.0,    1.9,
  0.0,    1.9,    1.7,    1.7,    1.7,    1.7,    1.4,    1.2,    1.0,
  0.5,    0.3,    0.2,    0.0,   -0.4,   -0.7,   -0.9,   -1.4,   -2.1,
 -2.6,   -3.1,   -3.5,   -4.0,   -4.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    2.4,    2.3,    2.3,    0.0,    0.0,    2.0,    1.9,
  1.8,    1.6,    1.5,    1.4,    1.0,    1.0,    1.0,    0.9,    0.9,
  0.5,    0.3,    0.2,    0.0,   -0.7,   -0.7,   -0.9,   -1.3,   -2.0,
 -2.5,   -3.2,   -3.6,   -4.0,   -4.8,   -4.8,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    2.0,    2.0,    1.9,    0.0,    1.7,    1.6,    1.5,
  1.4,    1.4,    1.1,    1.0,    0.8,    0.8,    0.7,    0.7,    0.5,
  0.4,    0.3,    0.2,    0.0,   -0.5,   -0.6,   -0.8,   -1.5,   -1.8,
 -2.0,   -3.4,   -3.8,   -4.2,   -4.8,   -5.0,   -6.0,   -7.0,   -7.5,
 -7.9,   -8.4,   -8.8,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    2.0,    2.0,    1.9,    1.8,    1.7,    1.6,    1.5,
  1.4,    1.4,    1.1,    1.0,    1.0,    0.9,    0.8,    0.6,    0.5,
  0.4,    0.3,    0.2,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,   -4.5,   -4.5,   -5.5,   -5.9,   -6.2,   -6.7,
 -7.2,   -8.0,   -8.7,  -10.0,  -12.0,  -14.0,  -16.0,  -18.0,  -20.0,
-20.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    2.0,    2.0,    2.0,    1.8,    1.7,    1.6,    1.5,
  0.0,    1.4,    1.1,    1.0,    1.0,    0.9,    0.8,    0.6,    0.5,
  0.5,    0.3,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.^,    0.0,   -4.7,   -5.0,   -5.4,   -5.8,   -6.2,   -6.7,
 -7.0,   -7.8,   -8.5,   -9.3,  -10.0,  -12.0,  -14.0,  -16.0,  -18.0,
-20.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    1.9,    1.8,    1.7,    1.6,    1.7,
  1.7,    1.5,    1.5,    1.5,    0.0,    1.4,    1.3,    0.9,    0.8,
  0.5,    0.3,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,   -5.2,   -5.4,   -5.9,   -6.3,   -7.0,
 -7.5,   -8.0,   -8.6,   -9.3,  -10.0,  -13.0,  -15.0,  -17.0,  -19.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    1.9,    1.8,    1.8,    1.7,
  1.7,    1.7,    1.6,    0.0,    0.0,    1.4,    1.0,    1.0,    0.7,
  0.4,    0.4,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,   -4.0,   -4.0,
 -4.0,    0.0,   -8.4,   -8.7,  -11.0,  -14.0,  -16.0,  -18.0,  -20.0,
-20.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    1.6,    1.4,    1.3,    1.3,    0.9,    0.8,    0.8,    0.7,
  0.5,    0.5,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,   -5.0,
 -5.0,    0.0,  -16.0,  -17.0,  -18.0,  -19.0,  -20.0,  -20.0,  -20.0,
-20.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    1.8,    1.7,    1.1,    1.1,    0.0,    0.0,    0.7,    0.6,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
```

```
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,   -17.0,   -18.0,   -19.0,   -20.0,   -20.0,   -20.0,     0.0,
 0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     1.5,     1.4,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,   -20.0,   -20.0,   -20.0,   -20.0,   -20.0,   -20.0,     0.0,
 0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,
FMAX=100.0,
FMIN=20.0,
FRICT =
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,    75.0,    75.0,    75.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,      ,       0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,      ,       0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,    75.0,    75.0,    75.0,    75.0,    75.0,
75.0,    75.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,    75.0,    75.0,    75.0,    75.0,    75.0,    75.0,
75.0,    75.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,    75.0,    75.0,    75.0,    75.0,    75.0,    75.0,    75.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
 0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
```

```
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   0.0,    75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   0.0,    0.0,    0.0,    0.0,    75.0,   75.0,   0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,            0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   0.0,    75.0,
75.0,   75.0,   75.0,   75.0,   0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   0.0,    75.0,
```

```
75.0,   75.0,   75.0,   75.0,   75.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,   75.0,   75.0,   75.0,
 0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,    0.0,   75.0,   75.0,    0.0,   75.0,   75.0,
75.0,   75.0,   75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,   75.0,   75.0,    0.0,
 0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,    0.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,   75.0,   75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,   75.0,   75.0,   75.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,   75.0,   75.0,   75.0,    0.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,    0.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,    0.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,    0.0,    0.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,    0.0,    0.0,   75.0,   75.0,    0.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,
 0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
 0.0,    0.0,
 0.0,    0.0,    0.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,
75.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,    0.0,
```

```
 0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,   0.0,   0.0,
 0.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,   0.0,  75.0,  75.0,  75.0,
 0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,  75.0,  75.0,  75.0,   0.0,   0.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,  75.0,  75.0,  75.0,   0.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
 0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,   0.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,  75.0,  75.0,
75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,   0.0,
 0.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,  75.0,   0.0,   0.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,  75.0,
75.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,   0.0,

 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,  75.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
 0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0,  75.0,
75.0,   0.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,  75.0,
75.0,   0.0,
```

```
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,   75.0,   75.0,   75.0,   75.0,    0.0,    0.0,   75.0,   75.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,   75.0,   75.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,   75.0,   75.0,   75.0,   75.0,   75.0,   75.0,    0.0,
  0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
  0.0,    0.0,
 NV    =
   14000000000.000,
 NH    =
   1200000000000.0,
 RHO   =
 1000.,                                                          &END
&CNTRL      START = 000.0,
            PHOTST= 200.,
            POUTPT=10.,
            NHOTST=1500,                                        &END
&MODPAR     DELT  =30.0
            DELX  =1.0,
            DELY  =1.0,
            NFIN  =2400,
            UPAI  =100.0,                                       &END
&METEO      HEQL  =925.0,
            DELC  =0.0,
            GRADC =1.84,
            GRADT =-0.0055,
            ALFA1 =1820000,
            ALFA2 =362000,
            TDEL  =0.125,
            SMIN  =-15.0,                                       &END
```

APPENDIX-C: LISTING OF VECTOR FORTRAN

```
      PROGRAM VECTOR                                                    VEC00010
*************************************************************         VEC00020
*                                                           *         VEC00030
*     PROGRAM FOR PLOTTING VELOCITY FILED IN A GLACIER      *         VEC00040
*                                                           *         VEC00050
*************************************************************         VEC00060
*                                                                       VEC00070
      PARAMETER (JP=1,LENGTH=54,IWIDTH=33,LENP1=LENGTH+1,               VEC00080
     &           IWIDP1=IWIDTH+1,LPLS=LENGTH+2,IWPLS=IWIDTH+2)          VEC00090
*                                                                       VEC00100
      LOGICAL GRID                                                      VEC00110
*                                                                       VEC00120
      COMMON/BLK1/ ETA(LPLS,IWPLS),IGRID(LPLS,IWPLS)                    VEC00130
      COMMON/BLK2/ A(JP,LPLS,IWPLS),C(JP,LENP1,IWIDTH),                 VEC00140
     &             D(JP,LENGTH,IWIDP1)                                  VEC00150
      COMMON/BLK3/ DELX,DELY,TIME,XMAX,YMAX,IWMAX,LMAX,IWMIN,           VEC00160
     &             LMIN,SLEVEL                                          VEC00170
*                                                                       VEC00180
      NAMELIST /MODPAR/ GRID,PLTSIZ,DELX,DELY,IWMAX,LMAX,IWMIN,LMIN,    VEC00190
     &                  SLEVEL,TIME                                     VEC00200
*                                                                       VEC00210
*                                                                       VEC00220
       READ(1,MODPAR)                                                   VEC00230
*      WRITE(*,MODPAR)                                                  VEC00240
       READ(2) ISTRT                                                    VEC00250
       READ(2) IGRID                                                    VEC00260
       READ(2) A                                                        VEC00270
       READ(2) ETA                                                      VEC00280
       READ(2) ETA                                                      VEC00290
       READ(2) C                                                        VEC00300
       READ(2) C                                                        VEC00310
       READ(2) D                                                        VEC00320
       READ(2) D                                                        VEC00330
*                                                                       VEC00340
       DO 101 I=1,LPLS                                                  VEC00350
          DO 101 J=1,IWPLS                                              VEC00360
             IF(IGRID(I,J).EQ.2) THEN                                   VEC00370
                IGRID(I,J)=1                                            VEC00380
             ENDIF                                                      VEC00390
101    CONTINUE                                                         VEC00400
*                                                                       VEC00410
*                                                                       VEC00420
      XMAX=FLOAT(LMAX)*DELX                                             VEC00430
      YMAX=FLOAT(IWMAX)*DELY                                            VEC00440
      CALL DEVICE('CAL5500')                                            VEC00450
      CALL HEADER('LIAQAT    ',10)                                      VEC00460
      FACT=PLTSIZ/XMAX                                                  VEC00470
      CALL PLOTS(0,0,14)                                                VEC00480
      CALL FACTOR(FACT)                                                 VEC00490
      CALL PLOT(0.0,0.0,-3)                                             VEC00500
*                                                                       VEC00510
       IF(GRID) THEN                                                    VEC00520
          CALL MESH                                                     VEC00530
       ELSE                                                             VEC00540
          CALL BORDER                                                   VEC00550
```

```
            CALL VELO
         ENDIF
*
999      CONTINUE

*
         CALL PLOT(0.0,0.0,999)
*
       STOP
       END


       SUBROUTINE MESH
***************************************************************
*                                                             *
*     GENERATES GRID FOR THE VELOCITY VECTO[illegible]TS                *
*                                                             *
***************************************************************
*
       PARAMETER (JP=1,LENGTH=54,IWIDTH=33,LENP1=LENGTH+1,
     &           IWIDP1=IWIDTH+1,LPLS=LENGTH+2,IWPLS=IWIDTH+2)
*
*
       COMMON/BLK1/ ETA(LPLS,IWPLS),IGRID(LPLS,IWPLS)
       COMMON/BLK3/ DELX,DELY,TIME,XMAX,YMAX,IWMAX,LMAX,IWMIN,
     &             LMIN,SLEVEL
*
*
         SLEN=DELX
         ANGLE=0.0
         SN=4.0
         DO 10 J=1,IWMAX-1
            JJ=J-1
            NN=J+1
            YP=FLOAT(JJ)*SLEN
            DO 10 I=1,LENGTH
               MM=I+1
               IF(IGRID(MM,NN).EQ.1) GO TO 10
               II=I-1
               XP=FLOAT(II)*SLEN
               CALL POLY(XP,YP,SLEN,SN,ANGLE)
 10      CONTINUE
*
         RETURN
         END


         SUBROUTINE BORDER
****************************************************************
*                                                              *
*      PLOTS THE BOUNDARY OF THE MODEL DOMAIN                  *
*                                                              *
****************************************************************
*
       PARAMETER (JP=1,  [illegible]GTH=54,IWIDTH=33,LENP1=LENGTH+1,
     &           IWIDP1=IWIDTH+1,LPLS=LENGTH+2,IWPLS=IWIDTH+2)
*
       COMMON/BLK1/ ETA(LPLS,IWPLS),IGRID(LPLS,IWPLS)
       COMMON/BLK3/ DELX,DELY,TIME,XMAX,YMAX,IWMAX,LMAX,IW[illegible],
```

```
     &              LMIN,SLEVEL                                          VEC01110
*                                                                        VEC01120
*                                                                        VEC01130
          XM=FLOAT(LMAX-LMIN+1)*DELX                                     VEC01140
          YM=FLOAT(IWMAX-IWMIN+1)*DELY                                   VEC01150
*                                                                        VEC01160
           ID=0                                                          VEC01170
           DO 30 J=IWMIN,IWMAX-1                                         VEC01180
              ID=ID+1                                                    VEC01190
              JJ=J+1                                                     VEC01200
              IC=0                                                       VEC01210
              DO 30 I=LMIN,LMAX-1                                        VEC01220
                 IC=IC+1                                                 VEC01230
                 II=I+1                                                  VEC01240
                 XPAGE=FLOAT(IC)*DELX                                    VEC01250
                 YPAGE=FLOAT(ID)*DELY                                    VEC01260
                 IF(IGRID(II,JJ).NE.IGRID(I,JJ)) THEN                    VEC01270
                    CALL PLOT(XPAGE,YPAGE,3)                             VEC01280
                    YP=DELY+YPAGE                                        VEC01290
                    CALL PLOT(XPAGE,YP,2)                                VEC01300
                 ENDIF                                                   VEC01310
*                                                                        VEC01320
                 IF(IGRID(II,JJ).NE.IGRID(II,J)) THEN                    VEC01330
                    CALL PLOT(XPAGE,YPAGE,3)                             VEC01340
                    XP=DELX+XPAGE                                        VEC01350
                    CALL PLOT(XP,YPAGE,2)                                VEC01360
                 ENDIF                                                   VEC01370
30         CONTINUE                                                      VEC01380
*                                                                        VEC01390
       RETURN                                                            VEC01400
       END                                                               VEC01410
```

```
       SUBROUTINE VELO                                                  VEC01420
*****************************************************************      VEC01430
*                                                               *        VEC01440
*       COMPUTES AND PLOTS VELOCITY VECTOR                      *        VEC01450
*                                                               *        VEC01460
*****************************************************************      VEC01470
*                                                                        VEC01480
       PARAMETER (JP=1,LENGTH=54,IWIDTH=33,LENP1=LENGTH+1,              VEC01490
     &            IWIDP1=IWIDTH+1,LPLS=LENGTH+2,IWPLS=IWIDTH+2)        VEC01500
*                                                                        VEC01510
       DIMENSION XMAG(LENGTH,IWIDTH),DIR(LENGTH,IWIDTH)                 VEC01520
       COMMON/BLK1/ ETA(LPLS,IWPLS),IGRID(LPLS,IWPLS)                   VEC01530
       COMMON/BLK2/ A(JP,LPLS,IWPLS),C(JP,LENP1,IWIDTH),                VEC01540
     &              D(JP,LENGTH,IWIDP1)                                 VEC01550
       COMMON/BLK3/ DELX,DELY,TIME,XMAX,YMAX,IWMAX,LMAX,IWMIN,          VEC01560
     &              LMIN,SLEVEL                                         VEC01570
*                                                                        VEC01580
         SIGMA=SLEVEL                                                   VEC01590
         XL=FLOAT(LMAX-LMIN+1)*DELX                                     VEC01600
         YL=FLOAT(IWMAX-IWMIN+1)*DELY                                   VEC01610
*                                                                        VEC01620
         DO 100 I=2,LPLS-1                                              VEC01630
            DO 100 J=2,IWPLS-1                                          VEC01640
               IF(IGRID(I,J).EQ.1) GO TO 100                            VEC01650
               II=I-1                                                   VEC01660
```

```
                JJ=J-1                                                    VEC01670
                U1=0.0                                                    VEC01680
                U2=0.0                                                    VEC01690
                V1=0.0                                                    VEC01700
                V2=0.0                                                    VEC01710
                DO 300 K=1,JP                                             VEC01720
                   IF(SLEVEL.EQ.-1) THEN                                  VEC01730
                     IF(A(K,I,J).LT.0.00001) THEN                         VEC01740
                        TRIG=1.0                                          VEC01750
                     ELSE                                                 VEC01760
                        TRIG=(SIN(A(K,I,J)))/(A(K,I,J))                   VEC01770
                     ENDIF                                                VEC01780
                   ELSE                                                   VEC01790
                      TRIG=COS(A(K,I,J)*(2.0-SIGMA)*0.5)                  VEC01800
                   ENDIF                                                  VEC01810
                   U1=U1+C(K,II,JJ)*TRIG                                  VEC01820
                   U2=U2+C(K,I,JJ)*TRIG                                   VEC01830
                   V1=V1+D(K,II,JJ)*TRIG                                  VEC01840
                   V2=V2+D(K,II,J)*TRIG                                   VEC01850
300             CONTINUE                                                  VEC01860
                U=86400*(U1+U2)*0.5                                       VEC01870
                V=86400*(V1+V2)*0.5                                       VEC01880
                XMAG(II,JJ)=(U**2+V**2)**0.5                              VEC01890
                IF(U.EQ.0.0.AND.V.EQ.0.0) THEN                            VEC01900
                   DIR(II,JJ)=0.0                                         VEC01910
                  GO TO 100                                               VEC01920
                ENDIF                                                     VEC01930
                DIR(II,JJ)=ATAN2(V,U)                                     VEC01940
100          CONTINUE                                                     VEC01950
*                                                                         VEC01960
             VMAX=0.0                                                     VEC01970
             DO 20 I=1,LENGTH                                             VEC01980
                DO 20 J=1,IWIDTH                                          VEC01990
                  IF(XMAG(I,J).GT.VMAX) VMAX=XMAG(I,J)                    VEC02000
20           CONTINUE                                                     VEC02010
             VMAX=1.0                                                     VEC02020
             FUDGE=DELX/VMAX                                              VEC02030
             DO 600 I=1,LENGTH                                            VEC02040
             DO 600 J=1,IWIDTH                                            VEC02050
                XMAG(I,J)=XMAG(I,J)*FUDGE                                 VEC02060
600          CONTINUE                                                     VEC02070
*                                                                         VEC02080
             ID=0                                                         VEC02090
             DO 10 J=IWMIN  ,IWMAX                                        VEC02100
                ID=ID+1                                                   VEC02110
                JJ=J-1                                                    VEC02120
                YPAGE=(FLOAT(ID-1)*DELY+DELY*0.5)                         VEC02130
                IC=0                                                      VEC02140
                DO 10 I=LMIN  ,LMAX                                       VEC02150
                   IC=IC+1                                                VEC02160
                   II=I-1                                                 VEC02170
                   XPAGE=(FLOAT(IC-1)*DELX+DELX*0.5)                      VEC02180
                   IF(IGRID(I,J).EQ.1) GO TO 10                           VEC02190
                   AHLEN=XMAG(II,JJ)                                      VEC02200
                   XTIP=AHLEN*COS(DIR(II,JJ))+XPAGE                       VEC02210
                   YTIP=AHLEN*SIN(DIR(II,JJ))+YPAGE                       VEC02220
                   AHLEN=AHLEN*0.2                                        VEC02230
                   CALL AROHD(XPAGE,YPAGE,XTIP,YTIP,AHLEN,0.0,11)         VEC02240
10           CONTINUE                                                     VEC02250
*                                                                         VEC02260
             CALL PLOT(0.0    ,3)                                         VEC02270
```

```
      XPAGE=7.0*DELX                                                    VEC02280
      YPAGE=5.0*DELY                                                    VEC02290
      YM= 0.10*YL                                                       VEC02300
      HEI=0.6*DELX                                                      VEC02310
      CALL SYMBOL(XPAGE,YPAGE,HEI,'TIME(YEARS): ',90.0,13)              VEC02320
      YPAGE=YPAGE+7.5*DELY                                              VEC02330
      CALL NUMBER(XPAGE,YPAGE,HEI,TIME,90.0,0)                          VEC02340
      XP=8.0*DELX                                                       VEC02350
      YP=5.0*DELY                                                       VEC02360
      CALL SYMBOL(XP,YP,HEI,'SCALE(M/DAY):',90.0,13)                    VEC02370
      OMAG=0.01                                                         VEC02380
      YP=13.0*DELY                                                      VEC02390
      XTIP=8.0*DELX                                                     VEC02400
      YTIP=YP+VMAX*FUDGE                                                VEC02410
      AHLEN=VMAX*0.15                                                   VEC02420
      CALL AROHD(XP,YP,XTIP,YTIP,AHLEN,0.0,11)                          VEC02430
      XP= 9.0*DELX                                                      VEC02440
      YP= 12.5*DELY                                                     VEC02450
      CALL NUMBER(XP,YP,HEI,VMAX,90.0,1)                                VEC02460
      XM=XL-10.0*DELX                                                   VEC02470
      YM=4.0*DELY                                                       VEC02480
      CALL SYMBOL(XM,YM,HEI,'VELOCITY FIELD OF THE TAKU GLACIER',       VEC02490
     &            90.0,34)                                              VEC02500
      XM=XM+2.0*DELX                                                    VEC02510
      YM=YM+5.0*DELY                                                    VEC02520
      IF(SLEVEL.EQ.2.0) THEN                                            VEC02530
           CALL SYMBOL(XM,YM,HEI,'   S                    ',           VEC02540
     &            90.0,24)                                              VEC02550
      ELSE IF(SLEVEL.EQ.0.0) THEN                                       VEC02560
           CALL SYMBOL(XM,YM,HEI,'   B                    ',           VEC02570
     &            90.0,24)                                              VEC02580
      ENDIF                                                             VEC02590
*                                                                       VEC02600
      RETURN                                                            VEC02610
      END                                                               VEC02620
```

APPENDIX-D: INPUT DATA FILE OF VECTOR FORTRAN

```
&MODPAR     GRID  =.F.,
            PLTSIZ= 8.0,
            DELX  =1.0,
            DELY  =1.0,
            IWMAX =35,
            LMAX  =56,
            IWMIN =1,
            LMIN  =1,
            TIME  =200.0,
            SLEVEL=2.0,         &END
```