GEOLOGICAL SURVEY OF CANADA

OPEN FILE 2212

COMPLETE RAY TRACING FORTRAN77 ROUTINES

Luděk Klimeš

1990

Energy, Mines and Resources Canada
Énergie, Mines et Ressources Canada

Canada

ABSTRACT

A self-consistent system of FORTRAN77 routines for the complete ray tracing in 3-D inhomogeneous structures with curved interfaces is presented. The complete ray tracing is understood to mean the standard ray tracing supplemented by the dynamic ray tracing and the evaluation of the reduced vectorial complex-valued amplitudes.

The model is composed of "complex" geological blocks separated by interfaces consisting of several smooth surfaces. The distribution of material parameters (P and S wave velocities, density, P and S wave attenuation factors) within each complex block must be smooth. The model may be expressed in Cartesian or curvilinear coordinates.

The description of the subroutines, input and output data may be found within the corresponding FORTRAN77 source code files.

RESUME

Nous présentons un système homogène de programmes écrits en FORTRAN77 pour le traçage complet des rayons dans des structures tridimensionnelles hétérogènes possédant des interfaces curvilinéaires. Par traçage complet des rayons, nous entendons le traçage général des rayons complété par le traçage dynamique des rayons et l'évaluation des amplitudes vectorielles complexes réduites.

Le modèle est formé de blocs géologiques "complexes" séparés par des interfaces formées de plusieurs surfaces lissées. La distribution des différents paramètres (vitesses des ondes P et S, densité, facteurs d'atténuation des ondes P et S) à l'intérieur de chaque bloc doit être uniforme. Le modèle peut être exprimé en coordonnées cartésiennes ou curvilinéaires.

La description des sous-programmes, des entrées et sorties de données se trouve dans les filières FORTRAN77 correspondantes.

CONTENTS

```
C COMPLETE RAY TRACING  (FEBRUARY, 1990).
```

```
C GUIDE TO COMPLETE RAY TRACING
C
C AUTHORS:
C     VLASTISLAV CERVENY
C             MFF UK
C             KE KARLOVU 3
C             121 16  PRAHA 2,  CZECHOSLOVAKIA
C     LUDEK KLIMES
C             UGG CSAV, ODD. 204
C             PUSKINOVO NAM. 9
C             160 00  PRAHA 6,  CZECHOSLOVAKIA
C     IVAN PSENCIK
C             GFU CSAV
C             BOCNI II, 1401
C             141 31  PRAHA 4,  CZECHOSLOVAKIA
C
C REFERENCE:
C     CERVENY V., KLIMES L. AND PSENCIK I.: COMPLETE SEISMIC-RAY TRACING
C             IN THREE-DIMENSIONAL STRUCTURES.  IN: SEISMOLOGICAL
C             ALGORITHMS, ED. DOORNBOS D.J., ACADEMIC PRESS, NEW YORK
C             1988.
C     THE ABOVE PAPER CONTAINS DETAILED DESCRIPTION OF THE ALGORITHM OF
C     COMPLETE RAY TRACING AND IS FREQUENTLY REFERRED THROUGHOUT THE
C     CODE OF COMPLETE RAY TRACING.  THE REFERENCES LIKE (C.R.T.5.4) OR
C     (5.4) ARE RELATED TO THE PAPER.  THE DETAILED DESCRIPTION OF THE
C     INPUT DATA, SUBROUTINE AND FUNCTION PARAMETERS, AND OTHER
C     SPECIFICATIONS NOT CONTAINED IN THE PAPER, IS INCLUDED IN THE
C     INDIVIDUAL PROGRAM PACKAGES.
C
C THE CODE OF THE COMPLETE RAY TRACING CONSISTS OF THE FOLLOWING
C SUBROUTINE PACKAGES:
C     CRT...  MAIN PROGRAM.  IT READS THE INPUT DATA AND THEN CONTROLS
C             THE COMPLETE RAY TRACING OF THE SPECIFIED ELEMENTARY
C             WAVES.
C     METRIC..SUBROUTINE EVALUATING THE METRIC TENSOR AND CHRISTOFFEL
C             SYMBOLS AT A GIVEN POINT (SEE C.R.T.2), AND THE RELATED
C             SUBROUTINES AND EXTERNAL FUNCTIONS.
C     MODEL...SUBROUTINES READING INPUT DATA TO SPECIFY THE SEISMIC
C             MODEL FOR COMPLETE RAY TRACING (SEE C.R.T.3.2.1), AND
C             AUXILIARY SUBROUTINES RELATED TO THE MODEL (C.R.T.3.3).
C     SRFC... SUBROUTINES FOR SPECIFICATION AND INTERPOLATION, IN
C             RECTANGULAR GRIDS, OF THE FUNCTIONS DESCRIBING SMOOTH
C             SURFACES IN THE MODEL (SEE C.R.T.3.2.2).
C     PARM... SUBROUTINES FOR SPECIFICATION AND INTERPOLATION, IN
C             RECTANGULAR GRIDS, OF THE MATERIAL PARAMETERS WITHIN
C             INDIVIDUAL COMPLEX BLOCKS IN THE MODEL (SEE C.R.T.3.2.3).
C     VAL...  SUBROUTINES CONTROLING FUNCTION SPECIFICATION AND
C             INTERPOLATION.  DESIGNED TO PERFORM THE INTERPOLATION
C             OF A SET OF FUNCTIONS IN A RECTANGULAR GRID, EMPLOYING
C             SPLINES UNDER TENSION.  THESE SUBROUTINES ARE REFERENCED
C             BY THE SUBROUTINES OF PACKAGES 'SRFC', 'PARM' AND 'INIT'.
C     FITCRT..SOME ROUTINES TAKEN FROM CLINE'S SOFTWARE PACKAGE
C             'FITPACK', CALLED BY THE SUBROUTINES OF THE PACKAGE 'VAL'
C             TO PERFORM THE SPLINE-UNDER-TENSION INTERPOLATION.
C     CODE... SUBROUTINE PACKAGE DEVOTED TO THE CODES FOR ELEMENTARY
C             WAVES.  CONTAINS THE GENERAL DESCRIPTION OF THE CODES FOR
C             ELEMENTARY WAVES AND SUBROUTINES DESIGNED TO READ THE
C             INPUT DATA FOR THE CODES OF ELEMENTARY WAVES AND TO
C             TRANSFORM THE USED NUMERICAL CODE OF THE ELEMENTARY WAVE
C             UNDER CONSIDERATION INTO INSTRUCTIONS SPECIFYING THE
```

```
C                  BEHAVIOUR OF THE WAVE AT THE INITIAL POINTS OF RAYS AND AT
C                  ALL POINTS OF INCIDENCE OF THE RAYS AT INTERFACES (SEE
C                  C.R.T.4).
C         RAY...   CONTAINS THE SUBROUTINE DESIGNED TO READ THE INPUT DATA
C                  FOR COMPLETE RAY TRACING AND TO STORE THEM IN THE MEMORY
C                  (SEE C.R.T.5.6), THE LIST OF THE QUANTITIES COMPUTED ALONG
C                  RAYS (SEE C.R.T.5.2), AND THE SUBROUTINE DESIGNED TO
C                  CONTINUE THE COMPLETE RAY TRACING OF A RAY FROM THE GIVEN
C                  POINT (SEE C.R.T.5.7).
C         RAYCB...SUBROUTINES FOR COMPLETE RAY TRACING WITHIN ONE COMPLEX
C                  BLOCK (SEE C.R.T.5.8).
C         HPCG...  SUBROUTINE OF THE IBM SCIENTIFIC SUBROUTINE PACKAGE
C                  SOLVING A SYSTEM OF GENERAL FIRST ORDER ORDINARY
C                  DIFFERENTIAL EQUATIONS WITH GIVEN INITIAL VALUES.  IT IS
C                  CALLED BY THE SUBROUTINE RAYCB OF THE PACKAGE 'RAYCB'.
C         TRANS...SUBROUTINE TRANSFORMING THE COMPUTED QUANTITIES ACROSS A
C                  CURVED INTERFACE (SEE C.R.T.5.9), AND SUBROUTINE REPLACING
C                  THE AMPLITUDES BY ONES INVOLVING THE APPROPRIATE
C                  CONVERSION COEFFICIENTS (SEE C.R.T.5.5.4).
C         COEF...  SUBROUTINES COMPUTING THE REFLECTION/TRANSMISSION
C                  COEFFICIENTS (SEE C.R.T.5.9.7).
C         INIT...  SUBROUTINES DESIGNED TO READ THE INPUT DATA FOR THE
C                  INITIAL SURFACE, AND TO DEFINE THE INITIAL VALUES OF THE
C                  QUANTITIES FOR COMPLETE TRACING OF A RAY WITH GIVEN
C                  TAKE-OFF PARAMETERS (SEE C.R.T.6).
C         RPAR...  SUBROUTINES CONTROLING THE TAKE-OFF PARAMETERS OF THE
C                  RAYS.  A USER MAY INTRODUCE HIS OWN PROCEDURE OF SELECTION
C                  OF THE TAKE-OFF PARAMETERS AND HIS OWN TWO-POINT RAY
C                  TRACING ALGORITHM BY MEANS OF A MODIFICATION OF THIS
C                  SUBROUTINE PACKAGE.
C         WRIT...  SUBROUTINES CREATING THE OUTPUT OF COMPLETE RAY TRACING
C                  (SEE C.R.T.5.5).
C         SCRONUL... DUMMY VERSION OF THE SUBROUTINE PACKAGE 'SCRO'
C                  CONTAINING SCREEN OUTPUT SUBROUTINES CALLED BY THE
C                  SUBROUTINES OF THE PACKAGE 'WRIT'.
C THE ABOVE PACKAGES FORM A CONSISTENT BASIC VERSION OF THE COMPLETE RAY
C TRACING PROGRAM.  NOTE THAT NOT ALL OF THE SUBROUTINES OF THE SOFTWARE
C PACKAGE 'FITCRT' ARE USED IN THE BASIC VERSION.  THE FOLLOWING
C SUBROUTINE PACKAGES MAY BE USED TO MODIFY THE BASIC VERSION OF THE
C COMPLETE RAY TRACING PROGRAM:
C         SCROPC..VERSION OF THE SUBROUTINE PACKAGE 'SCRO' FOR THE
C                  IBM-COMPATIBLE PERSONAL COMPUTERS.  IT SHOULD ALSO WORK ON
C                  THE VAX COMPUTERS.  THE SCREEN OUTPUT CONSISTS OF A SIMPLE
C                  GRAPHIC OUTPUT (SEE THE SUBROUTINE PACKAGES 'PLOT' BELOW)
C                  AND OF WRITING THE BRIEF MESSAGES ON THE STATUS OF THE
C                  CURRENTLY COMPUTED RAY TO THE CONSOLE.  THIS OUTPUT
C                  CONTROLS THE CONSOLE BY MEANS OF THE ANSI ESCAPE SEQUENCES
C                  SUPPORTED BY MS-DOS ANSI.SYS DRIVER ON IBM-COMPATIBLE
C                  PERSONAL COMPUTERS.  THIS CONSOLE OUTPUT IS JUST AN
C                  EXAMPLE, THE SUBROUTINES SCRO1, SCRO2, SCRO3, SCRO4 AND
C                  SCRO5 HAVE TO BE MODIFIED BY A USER FOR THE PARTICULAR
C                  COMPUTER SYSTEM.  IF NO SCREEN OUTPUT IS REQUIRED, THE
C                  EXECUTABLE STATEMENTS OF THE SUBROUTINES SCRO1, SCRO2,
C                  SCRO3, SCRO4 AND SCRO5 MAY SIMPLY BE DELETED (SEE
C                  'SCRONUL' ABOVE).
C         PLOTNUL... DUMMY VERSION OF THE SUBROUTINE PACKAGE 'PLOT'
C                  CONTAINING THE PLOT SUBROUTINES CALLED BY THE SUBROUTINES
C                  OF THE PACKAGE 'SCROPC', SEE ALSO THE SECTION GRAPHICS OF
C                  THIS GUIDE.
C         PLOTVDI... VERSION OF THE SUBROUTINE PACKAGE 'PLOT' CONTAINING THE
```

```
C              PLOT INTERFACE SUBROUTINES BETWEEN THE SUBROUTINE PACKAGE
C              'SCROPC' AND THE GSS-VDI GRAPHIC FUNCTIONS.
C      PLOTH...VERSION OF THE SUBROUTINE PACKAGE 'PLOT' CONTAINING THE
C              PLOT INTERFACE SUBROUTINES BETWEEN THE SUBROUTINE PACKAGE
C              'SCROPC' AND THE HALO GRAPHIC FUNCTIONS.
C
C RELEASED VERSIONS:
C      1988, JUNE:  PRELIMINARY VERSION, CONTAINS ONLY SUBROUTINES FOR
C              MODEL SPECIFICATION, NO COMPLETE RAY TRACING SUBROUTINES.
C      1990, JANUARY:  FIRST CONSISTENT VERSION OF THE RAY TRACING
C              PROGRAM.  THE ROUTINES DETERMINING THE TAKE-OFF PARAMETERS
C              OF RAYS ARE AWFULLY PRIMITIVE.  NOT WELL-DEBUGGED.
C
C OUTPUT STATEMENTS:
C      ERROR MESSAGES:
C              THE ERROR MESSAGES ARE MOST FREQUENTLY WRITTEN BY
C              STATEMENTS LIKE
C                STOP 'ERROR MESSAGE',
C              SOMETIMES LIKE
C                WRITE(*,'(''  ERROR MESSAGE'')').
C              THEY ARE SPREAD OVER ALL SUPROUTINE PACKAGES.  THEY ARE
C              LISTED IN THE DESCRIPTIONS OF INDIVIDUAL SUBROUTINES.  THE
C              FIRST ONE OR TWO DIGITS OF THE ERROR CODE ROUGHLY INDICATE
C              THE SUBROUTINE PACKAGE:
C                1**... 'CRT',
C                2**... 'METRIC',
C                3**... 'MODEL', 'SRFC', 'PARM', 'VAL',
C                31*... 'MODEL',
C                32*... 'PARM',
C                35*... 'VAL',
C                4**... 'CODE',
C                55*... 'WRIT',
C                56*... 'RAY',
C                57*... 'RAY',
C                58*... 'RAYCB',
C                59*... 'TRANS', 'COEF',
C                6**... 'INIT',
C                7**... 'RPAR'.
C      OTHER OUTPUTS:
C              ARE CONCENTRATED IN THE SUBROUTINE PACKAGE 'WRIT'.
C      SCREEN OUTPUT:
C              IS CONCENTRATED IN THE SUBROUTINE PACKAGE 'SCRO', IN THE
C              SUBROUTINES SCRO1, SCRO2, SCRO3, SCRO4 AND SCRO5.
C
C DATA SETS:
C      THE COMPLETE RAY TRACING PROGRAM READS OR MAY WRITE THE FOLLOWING
C      DATA SETS:
C (1) CRT...   MAIN DATA SET WITH THE NAMES OF THE FILES CONTAINING THE
C              DATA SETS (2) TO (8).  IT IS READ IN BY THE MAIN PROGRAM
C              CRT AND IS DESCRIBED IN THE PACKAGE 'CRT'.
C (2) MODEL...INPUT DATA FOR THE MODEL.
C              THE DATA ARE READ BY THE SUBROUTINE MODEL1 AND ARE
C              DESCRIBED IN THE SUBROUTINE PACKAGE 'MODEL'.
C (3) DCRT...  INPUT DATA FOR THE COMPLETE RAY TRACING.
C              THE DATA ARE READ BY THE SUBROUTINE RAY1 AND ARE
C              DESCRIBED IN THE SUBROUTINE PACKAGE 'RAY'.
C (4) INIT...  INPUT DATA SPECIFYING THE INITIAL CONDITIONS FOR RAYS.
C              THE DATA ARE READ BY THE SUBROUTINE INIT1 AND ARE
C              DESCRIBED IN THE SUBROUTINE PACKAGE 'INIT'.
C (5) CODE...  INPUT DATA SPECIFYING THE CODES OF ELEMENTARY WAVES.
```

```
C               THE DATA ARE READ BY THE SUBROUTINE CODE1 AND ARE
C               DESCRIBED IN THE SUBROUTINE PACKAGE 'CODE'.
C (6) RPAR...   INPUT DATA SPECIFYING THE TAKE-OFF PARAMETERS OF THE
C               COMPUTED RAYS.  THE DATA ARE READ BY THE SUBROUTINE
C               RPAR1 AND ARE DESCRIBED IN THE SUBROUTINE PACKAGE 'RPAR'.
C (7) WRIT...   NAMES OF THE FILES CONTAINING THE DATA SETS (9) TO (11).
C               THIS DATA SET IS READ BY THE SUBROUTINE WRIT1 AND IS
C               DESCRIBED IN THE SUBROUTINE PACKAGE 'WRIT'.
C (8) LOG...    LOG OUTPUT.
C               THE DATA ARE WRITTEN BY THE SUBROUTINES WRIT1, WRIT2,
C               WRIT4 AND WRIT5 AND ARE DESCRIBED IN THE SUBROUTINE
C               PACKAGE 'WRIT'.
C (9) (WRIT31(I),I=1,KWAVE)... OUTPUT DATA SETS WITH THE COMPUTED
C               QUANTITIES ALONG THE RAYS (SEE C.R.T.5.5.3).  KWAVE IS THE
C               NUMBER OF THE COMPUTED ELEMENTARY WAVES, OR KWAVE=1 IF THE
C               QUANTITIES CORRESPONDING TO ALL WAVES ARE STORED INTO A
C               SINGLE FILE.  THE DATA ARE WRITTEN ESPECIALLY BY THE
C               SUBROUTINE WRIT31 AND ARE DESCRIBED IN THE SUBROUTINE
C               PACKAGE 'WRIT'.  THE THE DATA SETS MAY ALSO BE, IN PART,
C               WRITTEN BY THE SUBROUTINES WRIT1, WRIT2, WRIT4 OR WRIT5.
C               UNFORMATTED.
C (10) (WRIT32(I),I=1,NSTOR)... OUTPUT DATA SETS WITH THE COMPUTED
C               QUANTITIES AT THE SPECIFIED SURFACES (SEE C.R.T.5.5.2).
C               NSTOR IS THE NUMBER OF SURFACES AT WHICH THE QUANTITIES
C               ARE TO BE STORED.  THE DATA ARE WRITTEN ESPECIALLY BY THE
C               SUBROUTINE WRIT32 AND ARE DESCRIBED IN THE SUBROUTINE
C               PACKAGE 'WRIT'.  THE THE DATA SETS MAY ALSO BE, IN PART,
C               WRITTEN BY THE SUBROUTINES WRIT1, WRIT2, WRIT4 OR WRIT5.
C               UNFORMATTED.
C (11) ((WRIT33(I,J),I=1,NELEM(J)),J=1,NWAVE)... OUTPUT DATA SETS WITH
C               THE COMPUTED QUANTITIES AT THE ENDPOINTS OF THE ELEMENTS
C               OF RAYS OF INDIVIDUAL ELEMENTARY WAVES (SEE C.R.T.5.5.3).
C               NWAVE IS THE NUMBER OF THE COMPUTED ELEMENTARY WAVES.
C               NELEM(J) IS THE NUMBER OF ELEMENTS OF RAYS OF J-TH
C               ELEMENTARY WAVE WHICH DO NOT COINCIDE WITH THE ELEMENTARY
C               WAVES WHICH HAVE BEEN COMPUTED.  THE DATA ARE WRITTEN
C               ESPECIALLY BY THE SUBROUTINE WRIT33 AND ARE DESCRIBED IN
C               THE SUBROUTINE PACKAGE 'WRIT'.  THE DATA SETS MAY ALSO BE,
C               IN PART, WRITTEN BY THE SUBROUTINES WRIT1, WRIT2, WRIT4 OR
C               WRIT5.  THESE FILES ARE ARE AUXILIARY DISK STORAGE
C               LOCATIONS TO THE PROGRAM CRT AND ARE NOT INTENDED TO BE
C               THE OUTPUT OF THE COMPLETE RAY TRACING USED BY THE USER
C               APPLICATION PROGRAMS.  UNFORMATTED.
C (12) FILE WITH THE QUANTITIES AT THE INITIAL POINTS OF RAYS (SEE
C               C.R.T.6.1) CORRESPONDING TO EACH OF THE FILES (9), (10)
C               AND (11).  UNFORMATTED.
C     SEVERAL DATA SETS MAY BE READ FROM OR WRITTEN TO A SINGLE FILE.
C     THE DATA SETS ARE READ IN THE FOLLOWING ORDER:
C               SUCCESIVELY READING DATA (1), (2), (3), (4) AND THE
C               INTRODUCTION PART OF (7).
C               WRITING THE INTRODUCTION PARTS OF (8) AND, PERHAPS, OF (9)
C               AND (10).
C               FOR EACH ELEMENTARY WAVE IWAVE: SUCCESIVELY READING THE
C               PARTS OF DATA (5), (6), (7) CORRESPONDING TO THIS
C               ELEMENTARY WAVE, THEN SIMULTANEOUSLY WRITING THE PARTS OF
C               (8), (9), (10), (11) CORRESPONDING TO THIS ELEMENTARY
C               WAVE.
C               WRITING THE END PARTS OF (8) AND, PERHAPS, OF (9) AND
C               (10).
C
```

```
C REASONS OF THE TERMINATION OF THE COMPUTATION OF A RAY
C     ARE SPECIFIED BY THE OUTPUT PARAMETER IEND OF THE SUBROUTINES
C     INIT2 AND RAY2 AND ARE LISTED IN THE SUBROUTINE PACKAGES 'INIT'
C     AND 'RAY'.
C
C LIST OF THE COMPUTED QUANTITIES IS GIVEN IN THE PACKAGE 'RAY'.
C
C PROGRAMMING LANGUAGE AND CODE LAYOUT:
C     PROGRAMMING LANGUAGE:
C             FORTRAN77-FULL LANGUAGE AS SPECIFIED IN THE ANSI STANDARD:
C             X3.9-1978: AMERICAN NATIONAL STANDARD PROGRAMMING LANGUAGE
C             FORTRAN.  AMERICAN NATIONAL STANDARDS INSTITUTE, NEW YORK
C             1978.
C     CHARACTER SET:
C             CHARACTER SET OF FORTRAN77 WITH CAPITAL LETTERS (49
C             CHARACTERS INCLUDING THE SPACE):
C             ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 =+-*/(),.$':
C     LINE LENGTH:
C             72 CHARACTERS
C     INDENTATION:
C             DOCUMENTATION AND EXTENSIVE COMMENTS AT THE BEGINNINGS OF
C             SUBROUTINES:
C               COLUMNS 3, 7, 15, 17, 19, ETC. WITH STEP 2.
C             FORTRAN STATEMENTS AND COMMENTS WITHIN STATEMENTS:
C               COLUMNS 7, 9, 11, ETC. WITH STEP 2.
C             INSIDES OF DO LOOPS AND IF STATEMENTS ARE INDENTED.
C             INDENTATION OF THE CONTINUATION LINES IS HAPHAZARD IN THIS
C               VERSION.
C     EXAMPLE FORMS OF THE FORTRAN STATEMENTS AND OF THE SPACING:
C             A=B+5.*(E+F)
C               NO SPACES IN ASSIGNMENT STATEMENTS AND EXPRESSIONS,
C               EXCEPT WHEN MATCHING SIMILAR EXPRESSIONS IN SUCCEEDING
C               LINES.
C             GO TO 12
C             GO TO (10,20,30)K
C             IF(I.EQ.J) GO TO 30
C             IF(A.GT.B) THEN
C             ELSE IF(C.GT.D) THEN
C             END IF
C             DO 20 I=1,15
C             BLOCK DATA
C             BACKSPACE
C             CALL VEL(N,COOR,F)
C             SUBROUTINE VEL(N,COOR,F)
C             PARAMETER (PI=3.14159)
C             COMMON/ABC/IR,FF,HH
C             COMMON IR,FF,HH
C             DATA R/4.,2.,1./,K/0/
C             READ(LUN,*) L,M,(BER(I),I=LS,MS),ALFA,BETA
C               BUT:
C               READ (LUN,*) L,M,(BER(I),I=LS,MS),ALFA,BETA
C               WRITE(LUN,*) L,M,(BER(I),I=LS,MS),ALFA,BETA
C     MISCELANEOUS:
C             ALL SENTENCES IN DOCUMENTATION SHOULD BE ENDED BY A DOT.
C             INDIVIDUAL SECTIONS OF THE CODE MAY BE SEPARATED BY BLANK
C               LINES WITH C IN THE FIRST COLUMN.
C             ONLY CONTINUE AND FORMAT STATEMENTS SHOULD BE LABELED.
C             SPECIFIC NAMES OF THE INTRINSIC FUNCTIONS ARE PREFERRED TO
C               GENERIC ONES.
C
```

```
C GRAPHICS:
C      ALL GRAPHIC OUTPUTS ARE CONCENTRATED IN THE SUBROUTINE PACKAGE
C      'SCROPC',  IN SUBROUTINES SCRO1, SCRO2, SCRO3, SCRO4 AND SCRO5.
C      CALCOMP SUBROUTINES
C              PLOTS(0,0,0)
C              PLOT(XPAGE,YPAGE,IPEN)
C              NEWPEN(INP)
C      ARE CALLED.  THE SUBROUTINE SYMBOL CANNOT BE USED BECAUSE ITS
C      CALCOMP SPECIFICATION IS NOT CONFORMAL WITH THE FORTRAN77
C      STANDARD.  THE SUBROUTINE NUMBER IS NOT EMPLOYED.  IF THE
C      SUBROUTINES PLOTS, PLOT AND NEWPEN ARE NOT PRESENT IN THE
C      COMPUTER, THEY MUST BE SUBSTITUTED BY USER-DEFINED OR VOID
C      SUBROUTINES.  THE SPECIFICATION OF THEIR PARAMETERS CAN BE FOUND
C      IN THE SUBROUTINE PACKAGE 'SCROPC'.
C
C ACKNOWLEDGEMENTS:
C      THE DEVELOPEMENT OF THE COMPLETE RAY TRACING ALGORITHM HAS BEEN
C      PARTIALLY SUPPORTED BY:
C      FACULTY OF MATHEMATICS AND PHYSICS, CHARLES UNIVERSITY, PRAGUE.
C      INSTITUTE OF GEOLOGY AND GEOTECHNICS, CZECHOSLOVAK ACADEMY OF
C              SCIENCES, PRAGUE.
C      ENERGY, MINES AND RESOURCES CANADA, OTTAWA, ONTARIO.
C      GEOPHYSICAL INSTITUTE, CZECHOSLOVAK ACADEMY OF SCIENCES, PRAGUE.
C
C DATE: 1990, JANUARY 24
C
C=======================================================================
C
```

```
C PACKAGE 'CRT' FOR COMPLETE RAY TRACING
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF:
C      CRT...  MAIN PROGRAM CONTROLING THE COMPLETE RAY TRACING.
C      UNIT... SUBROUTINE DESIGNED TO CONTROL CONNECTING AND
C              DISCONNECTING THE DATA FILES TO LOGICAL UNITS, AND TO
C              DETERMINE THE LOGICAL UNITS FROM WHICH THE GIVEN DATA SETS
C              ARE TO BE READ.
C
C=======================================================================
C
C MAIN PROGRAM CRT FOR COMPLETE RAY TRACING
C
C THIS PROGRAM READS THE INPUT DATA AND THEN CONTROLS THE COMPLETE RAY
C TRACING OF THE SPECIFIED ELEMENTARY WAVES.
C
C INPUT DATA -  FILE(1)...MAIN DATA SET CRT (NO NAME):
C     THE NAME OF THE MAIN INPUT DATA FILE IS GIVEN BY THE FIRST
C     EXECUTIVE STATEMENT OF THE MAIN PROGRAM.  IT IS BLANK IN THE
C     ORIGINAL VERSION.  THIS MAIN DATA FILE CONTAINS THE NAMES OF OTHER
C     INPUT FILES AND THE NAME OF THE OUTPUT LOG FILE.  IT MAY OR MAY
C     NOT CONTAIN OTHER INPUT DATA.  THE NAMES OF THE OUTPUT FILES WITH
C     THE COMPUTED QUANTITIES (C.R.T.5.5) ARE SPECIFIED IN THE
C     SUBROUTINE PACKAGE 'WRIT'.
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  ALL
C     INPUT VARIABLES ARE OF THE TYPE CHARACTER.  ONLY THE FIRST 80
C     CHARACTERS OF THE STRINGS ARE SIGNIFICANT.
C (1) TEXT
C     THE STRING DESCRIBING THE INPUT DATA.
C (2) FILE(2)...DATA SET MODEL
C     THE STRING CONTAINING THE NAME OF THE FILE WITH THE INPUT DATA FOR
C     THE MODEL.  THE DATA WILL BE READ IN BY THE SUBROUTINE MODEL1.
C (3) FILE(3)...DATA SET DCRT
C     THE STRING CONTAINING THE NAME OF THE FILE WITH THE INPUT DATA FOR
C     THE COMPLETE RAY TRACING.  THE DATA WILL BE READ IN BY THE
C     SUBROUTINE RAY1.
C (4) FILE(4)...DATA SET INIT
C     THE STRING CONTAINING THE NAME OF THE FILE WITH THE INPUT DATA
C     SPECIFYING THE INITIAL CONDITIONS FOR RAYS.  THE DATA WILL BE READ
C     IN BY THE SUBROUTINE INIT1.
C (5) FILE(5)...DATA SET CODE
C     THE STRING CONTAINING THE NAME OF THE FILE WITH THE CODES OF
C     ELEMENTARY WAVES.  THE DATA WILL BE READ IN BY THE SUBROUTINE
C     CODE1.
C (6) FILE(5)...DATA SET RPAR
C     THE STRING CONTAINING THE NAME OF THE FILE WITH THE DATA
C     SPECIFYING THE TAKE-OFF PARAMETERS OF THE REQUIRED RAYS.  THE DATA
C     WILL BE READ IN BY THE SUBROUTINE RPAR1.
C (7) FILE(6)...DATA SET WRIT
C     THE STRING CONTAINING THE NAME OF THE FILE SPECIFYING THE NAMES OF
C     THE OUTPUT FILES WITH THE COMPUTED QUANTITIES.  THESE DATA WILL BE
C     READ BY THE SUBROUTINE WRIT1.
C (8) FILE(6)...DATA SET LOG
C     THE STRING CONTAINING THE NAME OF THE OUTPUT LOG FILE.  THE DATA
C     WILL BE WRITTEN BY THE SUBROUTINES WRIT1, WRIT2, WRIT4 AND WRIT5.
C THE FILENAMES FILE(1) TO FILE(8) NEED NOT BE MUTUALLY DIFFERENT,
```

```
C SEVERAL DATA SETS MAY BE READ FROM (OR WRITTEN TO) THE SAME DATA FILE.
C EACH DATA FILE IS CLOSED WHEN READ OVER, AND ITS LOGICAL UNIT NUMBER
C MAY BE CONNECTED TO ANOTHER FILE TO BE OPENED.  WHEN MORE THAN ONE
C ELEMENTARY WAVE IS COMPUTED, IT IS NOT RECOMMENDED TO WRITE THE LOG
C OUTPUT DATA SET TO THE FILE CONTAINING THE CODE, RPAR OR WRIT DATA
C SET.
C
C SUBROUTINES REFERENCED:
      EXTERNAL MODEL1,CODE1,RAY1,RAY2,INIT1,INIT2,RPAR1,RPAR2,RPAR4
      EXTERNAL WRIT1,WRIT2,WRIT4,WRIT5,UNIT
C     MODEL1..SUBROUTINE PACKAGE 'MODEL'.
C     CODE1...SUBROUTINE PACKAGE 'CODE'.
C     RAY1, RAY2... SUBROUTINE PACKAGE 'RAY'.
C     INIT1, INIT2... SUBROUTINE PACKAGE 'INIT'.
C     RPAR1, RPAR2, RPAR4... SUBROUTINE PACKAGE 'RPAR'.
C     WRIT1, WRIT2, WRIT4, WRIT5... SUBROUTINE PACKAGE 'WRIT'.
C     UNIT... THIS PACKAGE.
C NOTE THAT THE ABOVE SUBROUTINES REFERENCE MANY OTHER EXTERNAL
C PROCEDURES FROM VARIOUS SUBROUTINE PACKAGES.  THESE INDIRECTLY
C REFERENCED PROCEDURES ARE NOT NAMED HERE, BUT ARE LISTED IN THE
C PARTICULAR SUBROUTINE PACKAGES.
C
C ERRORS:
C     102...  LESS THAN 4 LOGICAL UNITS:
C             FOUR LOGICAL UNITS MUST BE AVAILABLE TO READ THE INPUT
C             DATA AND TO WRITE THE OUTPUT LOG FILE.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     STORAGE LOCATIONS:
C
C     AUXILIARY STORAGE LOCATIONS:
      CHARACTER*80 TEXT
      INTEGER I
C
C     TEXTC...THE NAME OF THE DATA. STRING OF 80 CHARACTERS.
C     I...    AUXILIARY LOOP VARIABLE
C
C     QUANTITIES DESCRIBING ELEMENTARY WAVES AND THEIR RAYS:
      REAL PAR1,PAR2,YL(6),Y(35),YY(5)
      INTEGER IWAVE,IWAVE0,IKODE,IRAY,IY(12),IEND
C
C     IWAVE...INDEX OF THE COMPUTED ELEMENTARY WAVE.
C     IWAVE0..INDEX OF THE ALREADY COMPUTED ELEMENTARY WAVE HAVING THE
C             MOST NUMEROUS COMMON ELEMENTS WITH THE CURRENT ELEMENTARY
C             WAVE.
C     IKODE...THE LENGTH OF THE COMMON PART OF THE CODES OF THE IWAVE-TH
C             AND IWAVE0-TH ELEMENTARY WAVES.
C     IRAY... INDEX OF THE COMPUTED RAY.
C     PAR1,PAR2... RAY TAKE-OFF PARAMETERS.
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT A POINT OF A RAY (SEE
C             C.R.T.5.5.4).  THE QUANTITIES ARE LISTED IN THE SUBROUTINE
C             PACKAGE 'RAY'.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG A RAY
C             (SEE C.R.T.5.2.1).  THE QUANTITIES ARE LISTED IN THE
C             SUBROUTINE PACKAGE 'RAY'.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
```

```
C             A RAY (SEE C.R.T.5.2.2).  THE QUANTITIES ARE LISTED IN THE
C             SUBROUTINE PACKAGE 'RAY'.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG A RAY (SEE C.R.T.5.2.2).  THE QUANTITIES ARE LISTED
C             IN THE SUBROUTINE PACKAGE 'RAY'.
C     IEND... REASON OF THE TERMINATION OF THE COMPUTATION OF A RAY (SEE
C             C.R.T.5.4).
C
C     QUANTITIES DESCRIBING DATA FILES AND LOGICAL UNITS:
      INTEGER IFILE,NFILE,IU,NU,LUCODE,LURPAR,LUWRIT
      PARAMETER (NFILE=8)
      CHARACTER*80 FILE(NFILE)
      PARAMETER (NU=4)
      INTEGER LU(NU),KU(NU)
      DATA LU/5,4,3,2/
C
C     IFILE, NFILE, FILE, IU, NU, LU, KU... FOR THE DESCRIPTION OF THESE
C             QUANTITIES SEE THE SUBROUTINE UNIT BELOW.
C     LUCODE..THE LOGICAL UNIT CONNECTED TO THE FILE WITH THE CODE DATA.
C     LURPAR..THE LOGICAL UNIT CONNECTED TO THE FILE WITH THE RPAR DATA.
C     LUWRIT..THE LOGICAL UNIT CONNECTED TO THE FILE WITH THE WRIT DATA.
C
C.......................................................................
C
C     OPENING DATA FILES AND READING THE INPUT DATA:
C
C     THE NAME OF THE MAIN INPUT FILE.  THIS FILE CONTAINS THE NAMES OF
C     OTHER INPUT FILES
      FILE(1)=' '
      IF(NU.LT.4) THEN
        STOP 'ERROR 102 IN CRT(MAIN): LESS THAN 4 LOGICAL UNITS'
      END IF
C
C     (1) MAIN DATA FILE - CONTAINS THE NAMES OF OTHER INPUT FILES
      CALL UNIT(1,NFILE,FILE,IU,NU,LU,KU)
      READ(LU(IU),*) TEXT
      DO 10 I=2,NFILE
        READ(LU(IU),*) FILE(I)
   10 CONTINUE
C
C     (2) DATA FOR MODEL
      CALL UNIT(2,NFILE,FILE,IU,NU,LU,KU)
      CALL MODEL1(LU(IU))
C
C     (3) DATA FOR COMPLETE RAY TRACING
      CALL UNIT(3,NFILE,FILE,IU,NU,LU,KU)
      CALL RAY1(LU(IU))
C
C     (4) DATA FOR INITIAL POINTS OF RAYS
      CALL UNIT(4,NFILE,FILE,IU,NU,LU,KU)
      CALL INIT1(LU(IU))
C
C     (5) FILE CONTAINING THE CODES OF ELEMENTARY WAVES
      CALL UNIT(5,NFILE,FILE,IU,NU,LU,KU)
C     THE DATA FILE FOR THE SUBROUTINE CODE1 REMAINS OPEN
      LUCODE=LU(IU)
      IU=0
C
C     (6) FILE CONTROLING THE TAKE-OFF PARAMETERS OF RAYS
      CALL UNIT(6,NFILE,FILE,IU,NU,LU,KU)
```

```
C     THE DATA FILE FOR THE SUBROUTINE RPAR1 REMAINS OPEN
      LURPAR=LU(IU)
      IU=0
C
C     (7) FILE SPECIFYING THE NAMES OF FILES WITH THE COMPUTED
C     QUANTITIES
      CALL UNIT(7,NFILE,FILE,IU,NU,LU,KU)
C     THE DATA FILE FOR THE SUBROUTINE WRIT1 REMAINS OPEN
      LUWRIT=LU(IU)
      IU=0
C
C     (8) THE OUTPUT LOG FILE
      CALL UNIT(8,NFILE,FILE,IU,NU,LU,KU)
C     THE OUTPUT FILE FOR THE SUBROUTINES WRIT1, WRIT2, WRIT4 AND WRIT5
C     REMAINS OPEN
C
C.....................................................................
C
C     COMPLETE RAY TRACING:
C
      CALL WRIT1(LUWRIT,LU(IU),0,0,0)
      IWAVE=0
      IRAY=0
C
C     LOOP OVER ELEMENTARY WAVES
   30 CONTINUE
C       COMPUTATION OF A SINGLE ELEMENTARY WAVE:
C       READING THE INPUT DATA FOR THE ELEMENTARY WAVE
        CALL CODE1(LUCODE,IWAVE,IWAVE0,IKODE)
        IF(IWAVE.EQ.0) THEN
C         ALL REQUIRED ELEMENTARY WAVES ARE COMPUTED
          GO TO 90
        END IF
        CALL RPAR1(LURPAR,IWAVE)
        CALL WRIT1(LUWRIT,LU(IU),IWAVE,IWAVE0,IKODE)
C       LOOP OVER RAYS
   40   CONTINUE
C         COMPLETE TRACING OF A SINGLE RAY:
C         DETERMINATION OF THE TAKE-OFF PARAMETERS
          CALL RPAR2(IRAY,PAR1,PAR2)
          IF(IRAY.EQ.0) THEN
C           ALL REQUIRED RAYS OF THE ELEMENTARY WAVE ARE COMPUTED
            GO TO 80
          END IF
C         INITIAL CONDITIONS FOR THE RAY
          CALL INIT2(PAR1,PAR2,YL,Y,YY,IY,IEND,IWAVE0,IKODE)
          CALL WRIT2(LU(IU),IRAY)
          IF(IEND.EQ.0) THEN
C           COMPUTATION OF THE RAY
            CALL RAY2(YL,Y,YY,IY,IEND)
          END IF
C         THE RAY IS COMPUTED
          CALL RPAR4(IRAY,PAR1,PAR2,YL,Y,YY,IY,IEND)
          CALL WRIT4(LU(IU),IRAY,YL,Y,YY,IY,IEND)
        GO TO 40
   80   CONTINUE
C       THE ELEMENTARY WAVE IS COMPUTED
        CALL WRIT5(LU(IU),IWAVE)
      GO TO 30
   90 CONTINUE
```

```
C
C     END OF COMPUTATION
      CALL WRIT5(LU(IU),0)
      STOP
      END
C
C=======================================================================
C
      SUBROUTINE UNIT(IFILE,NFILE,FILE,IU,NU,LU,KU)
      INTEGER IFILE,NFILE,IU,NU,LU(NU),KU(NU)
      CHARACTER*(*) FILE(NFILE)
C
C SUBROUTINE UNIT IS DESIGNED TO CONTROL CONNECTING AND DISCONNECTING
C THE DATA FILES TO LOGICAL UNITS, AND TO DETERMINE THE LOGICAL UNITS
C FROM WHICH THE GIVEN DATA SETS ARE TO BE READ.
C
C INPUT:
C     IFILE...INDEX OF THE DATA SET TO BE READ IN.  THE DATA SETS ARE
C             INDEXED BY INTEGERS FROM 1 TO NFILE.
C     NFILE...THE TOTAL NUMBER OF DATA SETS.
C     FILE... CHARACTER ARRAY CONTAINING THE NAMES OF THE FILES
C             CONTAINING INDIVIDUAL DATA SETS.  DIFFERENT DATA SETS MAY
C             BE STORED IN THE SAME FILE.  IF IFILE=1, ONLY FILE(1) HAS
C             TO BE DEFINED.
C     IU...   INDEX OF THE LOGICAL UNIT CONNECTED TO THE FILE CONTAINING
C             THE LAST READ DATA SET (I.E. THE LAST DATA SET WAS READ
C             FROM THE LOGICAL UNIT LU(IU) CONNECTED TO THE FILE
C             FILE(KU(IU)).  ZERO IF NO DATA ARE READ IN, OR IF THERE IS
C             NO DATA FILE TO CLOSE.  NEED NOT BE DEFINED IF IFILE=1.
C     NU...   THE MAXIMUM NUMBER OF AVAILABLE LOGICAL UNITS.
C     LU...   ARRAY CONTAINING REFERENCE NUMBERS OF LOGICAL UNITS.
C     KU...   AUXILIARY ARRAY OF THE DIMENSION AT LEAST NU.
C             ITS ELEMENTS KU(1) TO KU(NU) MUST NOT BE MODIFIED BETWEEN
C             TWO INVOCATIONS OF THIS SUBROUTINE.  ITS VALUES NEED NOT
C             BE DEFINED IF IFILE=1.
C             KU(I)...ZERO IF THE LOGICAL UNIT LU(I) IS CLOSED,
C               OTHERWISE THE SEQUENTIAL NUMBER OF THE NEXT DATA SET TO
C               BE READ FROM THIS UNIT.
C
C OUTPUT:
C     IU...   INDEX OF THE LOGICAL UNIT CONNECTED TO FILE WITH THE DATA
C             SET TO BE READ IN (I.E. THE NEXT DATA SET WILL BE READ
C             FROM THE LOGICAL UNIT LU(IU) CONNECTED TO THE FILE
C             FILE(IFILE)).  ZERO IF NO LOGICAL UNIT IS AVAILABLE.
C     KU...   UPDATED INPUT VALUES.
C
C ERRORS:
C     101...  OPEN FILE ERROR:
C             ERROR ENCOUNTERED DURING EXECUTION OF THE OPEN STATEMENT.
C
C DATE: 1990, FEBRUARY 6
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER IERR,JU,I
C
      IF(IFILE.EQ.1) THEN
C       NO LOGICAL UNITS ARE CONNECTED WHEN OPENING THE FIRST DATA FILE
```

```
          DO 10 JU=1,NU
            KU(JU)=0
   10     CONTINUE
        ELSE
C         UPDATING INDICES OF NEXT DATA SETS TO BE READ FROM OPEN FILES:
          DO 13 JU=1,NU
            IF(0.LT.KU(JU).AND.KU(JU).LT.IFILE) THEN
C             THE DATA SET FROM THE FILE CONNECTED TO THE LOGICAL UNIT
C             LU(JU) HAS BEEN READ.  DETERMINING THE NEXT DATA SET
C             CONTAINED IN THE FILE:
              DO 11 I=IFILE,NFILE
                IF(FILE(KU(JU)).EQ.FILE(I)) THEN
C                 THE I-TH DATA SET WILL ALSO BE READ FROM THE LAST FILE
C                 CONNECTED TO THE LOGICAL UNIT LU(JU)
                  KU(JU)=I
                  GO TO 12
                END IF
   11         CONTINUE
C             NO OTHER DATA SET WILL BE READ FROM THE LAST FILE.  THE FILE
C             MAY BE CLOSED AND THE LOGICAL UNIT LU(IU) DICONNECTED
   12         CONTINUE
            END IF
   13     CONTINUE
C         CLOSING THE DATA FILE:
          IF(IU.GT.0) THEN
C           THERE IS A FILE SUBMITTED TO BE CLOSED
            IF(0.LT.KU(IU).AND.KU(IU).LT.IFILE) THEN
C             NO OTHER DATA SET WILL BE READ FROM THIS FILE.  THE FILE
C             MAY BE CLOSED AND THE LOGICAL UNIT LU(IU) DICONNECTED
              CLOSE(LU(IU))
              KU(IU)=0
            END IF
          END IF
        END IF
C
C       OPENING NEW DATA FILE:
        IF(IFILE.GT.0) THEN
          DO 21 JU=1,NU
            IF(KU(JU).EQ.IFILE) THEN
C             THE DATA FILE IS ALREADY OPEN
              IU=JU
              RETURN
            END IF
   21     CONTINUE
C         THE DATA FILE HAS TO BE OPENED
          DO 22 JU=1,NU
            IF(KU(JU).EQ.0) THEN
C             THE LOGICAL UNIT LU(JU) MAY BE CONNECTED TO THE DATA FILE
              IU=JU
              KU(IU)=IFILE
              IF(FILE(IFILE).EQ.' ') THEN
                OPEN(LU(IU),FILE=' ',STATUS='OLD',IOSTAT=IERR,ERR=90)
              ELSE
                OPEN(LU(IU),FILE=FILE(IFILE),IOSTAT=IERR,ERR=90)
              END IF
              RETURN
            END IF
   22     CONTINUE
C         NO LOGICAL UNIT AVAILABLE
          IU=0
```

```
      END IF
      RETURN
C
   90 CONTINUE
      WRITE(*,'('' STATUS'',I5,'': '',A)') IERR,FILE(IFILE)
      STOP 'ERROR 101 IN UNIT: OPEN FILE ERROR'
      END
C
C======================================================================
C
```

```
C SUBROUTINE PACKAGE 'METRIC' TO DEFINE THE COORDINATE SYSTEM
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C      METRB...BLOCK DATA SUBROUTINE DEFINING THE COMMON BLOCK /METRC/ TO
C              STORE THE DATA DESCRIBING THE COORDINATE SYSTEM.
C      METR1...SUBROUTINE DESIGNED TO STORE THE DATA INTO THE COMMON
C              BLOCK /METRC/.
C      KOOR... INTEGER FUNCTION RETURNING THE TYPE OF THE COORDINATE
C              SYSTEM.
C      METRIC..SUBROUTINE DESIGNED TO EVALUATE THE METRIC TENSOR AND
C              CHRISTOFFEL SYMBOLS AT A GIVEN POINT.
C
C STORAGE IN THE MEMORY:
C      THE DATA DESCRIBING THE COORDINATE SYSTEM ARE STORED IN THE COMMON
C      BLOCK /METRC/ DEFINED IN THE FOLLOWING SUBROUTINE:
C      -----------------------------------------------------------------
       BLOCK DATA METRB
       INTEGER KOORS
       COMMON/METRC/KOORS
       SAVE /METRC/
       END
C      -----------------------------------------------------------------
C      KOORS...SPECIFIES THE TYPE OF THE RIGHT-HANDED COORDINATE SYSTEM:
C              KOORS.LE.0: CARTESIAN COORDINATES.
C              KOORS.EQ.1: POLAR SPHERICAL COORDINATES (X1,X2,X3)=
C                (COLATITUDE,LONGITUDE,RADIUS).
C              KOORS.GE.2: GEOGRAPHIC SPHERICAL COORDINATES (X1,X2,X3)=
C                (LONGITUDE,LATITUDE,RADIUS).
C
C DATE: 1990, FEBRUARY 6
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
       SUBROUTINE METR1(KOOR)
       INTEGER KOOR
C
C SUBROUTINE METR1 IS DESIGNED TO STORE THE DATA SPECIFYING THE
C COORDINATE SYSTEM INTO THE COMMON BLOCK /METRC/.
C
C INPUT:
C      KOOR... SPECIFIES THE TYPE OF THE RIGHT-HANDED COORDINATE SYSTEM.
C THE INPUT PARAMETER IS NOT ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK /METRC/:
       INTEGER KOORS
       COMMON/METRC/KOORS
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 18
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
```

```
C
C      NO AUXILIARY STORAGE LOCATIONS.
C
       KOORS=KOOR
       RETURN
       END
C
C=======================================================================
C
       INTEGER FUNCTION KOOR()
C
C INTEGER FUNCTION KOOR IS DESIGNED TO RETURN THE TYPE OF THE COORDINATE
C SYSTEM.
C
C NO INPUT.
C
C OUTPUT:
C      KOOR... SPECIFIES THE TYPE OF THE RIGHT-HANDED COORDINATE SYSTEM:
C              KOOR.LE.0: CARTESIAN COORDINATES.
C              KOOR.EQ.1: POLAR SPHERICAL COORDINATES (X1,X2,X3)=
C                (COLATITUDE,LONGITUDE,RADIUS).
C              KOOR.GE.2: GEOGRAPHIC SPHERICAL COORDINATES (X1,X2,X3)=
C                (LONGITUDE,LATITUDE,RADIUS).
C
C COMMON BLOCK /METRC/:
       INTEGER KOORS
       COMMON/METRC/KOORS
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 18
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C      NO AUXILIARY STORAGE LOCATIONS.
C
       KOOR=KOORS
       RETURN
       END
C
C=======================================================================
C
       SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)
       REAL COOR(3),GSQRD,G(12),GAMMA(18)
C
C THIS SUBROUTINE EVALUATES THE METRIC TENSOR AND CHRISTOFFEL
C SYMBOLS AT A GIVEN POINT.
C
C INPUT:
C      COOR... ARRAY CONTAINING COORDINATES X1, X2, X3 OF THE GIVEN POINT
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C      GSQRD...SQUARE ROOT OF THE DETERMINANT OF THE COVARIANT METRIC
C              TENSOR.
C      G...    ARRAY CONTAINING COVARIANT COMPONENTS G11, G12, G22, G13,
C              G23, G33, AND CONTRAVARIANT COMPONENTS G11, G12, G22, G13,
C              G23, G33 OF THE METRIC TENSOR AT THE GIVEN POINT.
```

```
C     GAMMA...ARRAY CONTAINING CHRISTOFFEL SYMBOLS GAMMA111, GAMMA121,
C             GAMMA221, GAMMA131, GAMMA231, GAMMA331, GAMMA112,
C             GAMMA122, GAMMA222, GAMMA132, GAMMA232, GAMMA332,
C             GAMMA113, GAMMA123, GAMMA223, GAMMA133, GAMMA233,
C             GAMMA333, WHERE THE FIRST TWO INDICES ARE SUBSCRIPTS AND
C             THE THIRD INDEX IS A SUPERSCRIPT.
C
C COMMON BLOCK /METRC/:
      INTEGER KOORS
      COMMON/METRC/KOORS
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 18
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      REAL SMALL,C,S,R
      PARAMETER (SMALL=1.E-12)
C
C     SMALL...THE LOWER LIMIT FOR THE DISTANCE FROM THE SINGULARITIES OF
C             THE COORDINATE SYSTEM MEASURED IN THE COORDINATE UNITS.
C     C,S,R...AUXILIARY STORAGE LOCATIONS.
C
C.......................................................................
C
      DO 1 I=1,12
        G(I)=0.
    1 CONTINUE
      DO 2 I=1,18
        GAMMA(I)=0.
    2 CONTINUE
C
      IF(KOORS.LE.0) THEN
        GSQRD=1.
        G(1) =1.
        G(3) =1.
        G(6) =1.
        G(7) =1.
        G(9) =1.
        G(12)=1.
      ELSE IF(KOORS.EQ.1) THEN
        C=COS(COOR(1))
        S=SIN(COOR(1))
        R=COOR(3)
        IF(S.EQ.0.) S=SMALL
        IF(R.EQ.0.) R=SMALL
        GSQRD=R*R*ABS(S)
        G(1) =R*R
        G(3) =(R*S)**2
        G(6) =1.
        G(7) =1./G(1)
        G(9) =1./G(3)
        G(12)=1.
        GAMMA(3) =-S*C
        GAMMA(4) =1./R
        GAMMA(8) =C/S
        GAMMA(11)=1./R
```

```
          GAMMA(13)=-R
          GAMMA(15)=-R*S*S
        ELSE
          C=COS(COOR(2))
          S=SIN(COOR(2))
          R=COOR(3)
          IF(C.EQ.0.) C=SMALL
          IF(R.EQ.0.) R=SMALL
          GSQRD=R*R*ABS(C)
          G(1)=(R*C)**2
          G(3)=R*R
          G(6)=1.
          G(7)=1./G(1)
          G(9)=1./G(3)
          G(12)=1.
          GAMMA(3)=-S/C
          GAMMA(4)=1./R
          GAMMA(8)=S*C
          GAMMA(11)=1./R
          GAMMA(13)=-R*C*C
          GAMMA(15)=-R
        END IF
        RETURN
        END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'MODEL' TO SPECIFY THE SEISMIC MODEL FOR
C COMPLETE RAY TRACING.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING EXTERNAL PROCEDURES:
C      MODELB..BLOCK DATA SUBROUTINE DEFINING COMMON BLOCKS MODELT AND
C              MODELC TO STORE THE DATA DESCRIBING THE MODEL.
C      MODEL1..SUBROUTINE DESIGNED TO READ THE INPUT DATA FOR THE
C              DESCRIPTION OF THE MODEL AND TO STORE THEM IN THE MEMORY.
C              SUBROUTINE MODEL1 READS THE INPUT DATA (1)-(8) LISTED
C              BELOW ITSELF, THEN CALLS SUBROUTINE SRFC1 (INCLUDED IN THE
C              SUBROUTINE PACKAGE 'SRFC') TO READ THE INPUT DATA (9) FOR
C              SMOOTH SURFACES, AND FINALY CALLS SUBROUTINE PARM1
C              (INCLUDED IN THE SUBROUTINE PACKAGE 'PARM') TO READ THE
C              INPUT DATA (10) FOR THE PARAMETERS OF THE MEDIUM.
C      NSRFC...INTEGER FUNCTION RETURNING THE NUMBER OF SURFACES COVERING
C              STRUCTURAL INTERFACES.
C      BLOCK...SUBROUTINE DESIGNED TO DETERMINE THE MUTUAL POSITION OF A
C              POINT AND A SIMPLE AND A COMPLEX BLOCK.
C      ISIDE...AUXILIARY FUNCTION TO SUBROUTINE BLOCK.
C      INTERF..AUXILIARY SUBROUTINE TO SUBROUTINE BLOCK.
C      VELOC...SUBROUTINE TRANSFORMING THE VALUES OF A MEDIUM PARAMETERS
C              INTO VELOCITIES AND LOSS FACTORS.
C
C INPUT DATA FOR THE SPECIFICATION OF THE MODEL:
C      THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C      THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C      THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C      THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C      I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C      INTEGER.  OTHERWISE (EXCEPT TEXTM), THE INPUT PARAMETER IS OF THE
C      TYPE REAL.
C (1)  TEXTM
C      THE STRING CONTAINING THE NAME OF THE MODEL.  ONLY THE FIRST 80
C      CHARACTERS OF THE STRING ARE SIGNIFICANT.
C (2)  KOOR,NEXPV,NEXPQ
C      KOOR... SPECIFIES THE TYPE OF THE RIGHT-HANDED COORDINATE SYSTEM
C              (SEE THE SUBROUTINE PACKAGE 'METRIC'):
C              KOORS.LE.0: CARTESIAN COORDINATES.
C              KOORS.EQ.1: POLAR SPHERICAL COORDINATES (X1,X2,X3)=
C                (COLATITUDE,LONGITUDE,RADIUS).
C              KOORS.GE.2: GEOGRAPHIC SPHERICAL COORDINATES (X1,X2,X3)=
C                (LONGITUDE,LATITUDE,RADIUS).
C      NEXPV,NEXPQ... SPECIFY EXPONENTS OF THE POWER OF VELOCITIES
C              (NEXPV) AND LOSS FACTORS (NEXPQ) IN INPUT DATA.  FOR
C              EXAMPLE, UNIT VALUES OF NEXPV AND NEXPQ INDICATE THAT THE
C              PARAMETERS OF THE MEDIUM ARE VELOCITIES AND LOSS FACTORS,
C              INDICES EQUAL -1 INDICATE RECIPROCAL VALUES OF THESE
C              QUANTITIES, I.E.  SLOWNESSES AND QUALITY FACTORS.
C (3)  X1MIN,X1MAX,X2MIN,X2MAX,X3MIN,X3MAX
C      BOUNDARIES OF THE MODEL.
C (4)  NSRFC
C      NUMBER OF SMOOTH SURFACES IN THE MODEL.  THE SURFACES ARE INDEXED
C      SEQUENTIALLY IN ANY ORDER BY POSITIVE INTEGERS ISRFC FROM 1 TO
C      NSRFC.
C (5)  NSB
C      NUMBER OF MATERIAL SIMPLE BLOCKS IN THE MODEL.  THE BLOCKS ARE
C      INDEXED SEQUENTIALLY IN ANY ORDER BY POSITIVE INTEGERS ISB FROM 1
C      TO NSB.  FREE-SPACE BLOCKS ARE NOT INDEXED.
```

```
C (6) NSB INPUT OPERATIONS (READ STATEMENTS):
C     FOR EACH SIMPLE BLOCK WITH INDEX ISB, THE INDICES OF THE SURFACES
C     FORMING THE SET F(+) AND THE INDICES OF THE SURFACES FORMING THE
C     SET F(-).  THE INDICES OF SURFACES FROM F(+) MUST BE POSITIVE, THE
C     INDICES OF SURFACES FROM F(-) MUST BE INDICATED BY NEGATIVE SIGNS.
C     THE INDICES MAY BE SPECIFIED IN AN ARBITRARY ORDER AND MUST BE
C     TERMINATED BY A SLASH.
C (7) NCB
C     NUMBER OF MATERIAL COMPLEX BLOCKS IN THE MODEL.  THE BLOCKS ARE
C     INDEXED SEQUENTIALLY BY POSITIVE INTEGERS ICB FROM 1 TO NCB.  THE
C     FREE-SPACE BLOCKS ARE NOT INDEXED.
C (8) NCB INPUT OPERATIONS (READ STATEMENTS):
C     FOR EACH COMPLEX BLOCK, THE INDICES OF SIMPLE BLOCKS FORMING THE
C     COMPLEX BLOCK.  THE INDICES MAY BE SPECIFIED IN AN ARBITRARY ORDER
C     AND MUST BE TERMINATED BY A SLASH.
C (9) THE DATA SPECIFYING NSRFC FUNCTIONS F(X1,X2,X3) DESCRIBING THE
C     SMOOTH SURFACES IN THE MODEL. THE DATA ARE READ BY SUBROUTINE
C     SRFC1.  FOR THEIR DESCRIPTION REFER TO SUBROUTINE SRFC1 (INCLUDED
C     IN THE SUBROUTINE PACKAGE 'SRFC').
C (10) THE DATA SPECIFYING THE DISTRIBUTION OF PARAMETERS OF THE MODEL
C     IN ALL NCB MATERIAL COMPLEX BLOCKS.  THE DATA ARE READ BY
C     SUBROUTINE PARM1.  FOR THEIR DESCRIPTION REFER TO SUBROUTINE
C     PARM1 (INCLUDED IN THE SUBROUTINE PACKAGE 'PARM').
C FOR AN EXAMPLE REFER TO THE SAMPLE INPUT DATA FOR THE MODEL.
C
C STORAGE IN THE MEMORY:
C     THE INPUT DATA (1) TO (8) DESCRIBING THE STRUCTURE OF THE MODEL
C     ARE STORED IN COMMON BLOCKS /MODELT/ AND /MODELC/ DEFINED IN THE
C     FOLLOWING SUBROUTINE:
C     ------------------------------------------------------------------
      BLOCK DATA MODELB
      CHARACTER*80 TEXTM
      COMMON/MODELT/TEXTM
      SAVE /MODELT/
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
      SAVE /MODELC/
      END
C     ------------------------------------------------------------------
C     TEXTM...THE NAME OF THE MODEL. STRING OF 80 CHARACTERS.
C     NEXPV,NEXPQ... SPECIFY EXPONENTS OF THE POWER OF VELOCITIES
C             (NEXPV) AND Q-FACTORS (NEXPQ) IN INPUT DATA.  FOR EXAMPLE,
C             UNIT VALUES OF NEXPV AND NEXPQ INDICATE THAT THE
C             PARAMETERS OF THE MEDIUM ARE VELOCITIES AND Q FACTORS,
C             INDICES EQUAL -1 INDICATE RECIPROCAL VALUES OF THESE
C             QUANTITIES, I.E. SLOWNESSES AND LOSS FACTORS.
C     BOUNDM..BOUNDARIES X1MIN,X1MAX,X2MIN,X2MAX,X3MIN,X3MAX OF THE
C             MODEL.
C     NSRFCS..NUMBER OF SMOOTH SURFACES IN THE MODEL. THE SURFACES
C             ARE INDEXED SEQUENTIALLY BY POSITIVE INTEGERS, FROM 1 TO
C             NSRFCS.  NSRFCS IS THE STORAGE LOCATION FOR NSRFC.
C     NSB...  NUMBER OF MATERIAL SIMPLE BLOCKS IN THE MODEL. THE BLOCKS
C             ARE INDEXED SEQUENTIALLY BY POSITIVE INTEGERS ISB FROM 1
C             TO NSB.  FREE-SPACE BLOCKS ARE NOT INDEXED.
```

```
C      KSB...  CONTAINS THE INDICES OF THE SURFACES BOUNDING INDIVIDUAL
C              SIMPLE BLOCKS.  KSB(ISB), FOR ISB = 1 TO NSB, SPECIFY THE
C              PARTITION OF ARRAY KSB(NSB+1:NSB+NS) AMONG THE SIMPLE
C              BLOCKS.  HERE NS IS THE TOTAL NUMBER OF ALL OCCURENCIES OF
C              THE INDICES OF THE SURFACES BOUNDING ALL INDIVIDUAL SIMPLE
C              BLOCKS IN THE INPUT DATA.  THE INDICES OF THE SURFACES
C              BOUNDING INDIVIDUAL SIMPLE BLOCKS ARE STORED FROM
C              KSB(NSB+1) TO KSB(NSB+NS). THE LOCATIONS KSB(NSB+NS+1:MSB)
C              ARE UNDEFINED.  IT MUST BE NSB+NS.LT.MSB.  THE INDICES OF
C              THE SURFACES BOUNDING THE SIMPLE BLOCK ISB ARE STORED IN
C              KSB(I1) TO KSB(I2), WITH
C                I1 = KSB(ISB-1)+1   ,
C                I2 = KSB(ISB)       ,
C              WHERE KSB(ISB-1)=NSB FOR ISB=1.  FOR EACH SIMPLE BLOCK
C              WITH INDEX ISB, THE INDICES OF THE SURFACES FORMING THE
C              SET F(+) ARE STORED WITH POSITIVE SIGNS, THE INDICES OF
C              SURFACES FROM F(-) WITH NEGATIVE SIGNS.  FOR AN EXAMPLE
C              REFER TO THE SAMPLE INPUT DATA FOR THE MODEL.
C      NCB...  NUMBER OF MATERIAL COMPLEX BLOCKS IN THE MODEL. THE BLOCKS
C              ARE INDEXED SEQUENTIALLY BY POSITIVE INTEGERS ICB FROM 1
C              TO NCB.  THE FREE-SPACE BLOCKS ARE NOT INDEXED.
C      KCB...  CONTAINS THE INDICES OF THE SIMPLE BLOCKS FORMING
C              INDIVIDUAL COMPLEX BLOCKS.  KCB(ICB), FOR ICB = 1 TO NCB,
C              SPECIFY THE PARTITION OF ARRAY KSB(NCB+1:NCB+NB) AMONG THE
C              COMPLEX BLOCKS.  HERE NB IS THE TOTAL NUMBER OF ALL
C              OCCURENCIES OF THE INDICES OF THE SIMPLE BLOCKS FORMING
C              INDIVIDUAL COMPLEX BLOCKS IN THE INPUT DATA.  THE INDICES
C              OF THE SIMPLE BLOCKS FORMING INDIVIDUAL COMPLEX BLOCKS ARE
C              STORED FROM KSB(NCB+1) TO KSB(NCB+NB). THE LOCATIONS
C              KSB(NCB+NB+1:MCB) ARE UNDEFINED. IT MUST BE NCB+NB.LT.MCB.
C              THE INDICES OF THE SIMPLE BLOCKS FORMING COMPLEX BLOCK ICB
C              ARE STORED IN KCB(I1) TO KCB(I2), WHERE
C                I1 = KCB(ICB-1)+1   ,
C                I2 = KCB(ICB)       .
C              HERE KCB(ICB-1)=NCB FOR ICB=1.  FOR AN EXAMPLE REFER TO
C              THE SAMPLE INPUT DATA FOR THE MODEL.
C      COMMON BLOCK /MODELT/ IS INCLUDED IN EXTERNAL PROCEDURE MODEL1.
C      COMMON BLOCK /MODELC/ IS INCLUDED IN EXTERNAL PROCEDURES MODEL1,
C      BLOCK, ISIDE, INTERF, AND MAY BE INCLUDED IN ANY OTHER SUBROUTINE.
C      ALL THE INPUT DATA ARE STORED SEQUENTIALLY IN THE SAME ORDER AS
C      THEY WERE READ.  THE ONLY EXCEPTION ARE LOCATIONS KSB(1) TO
C      KSB(NSB) AND KCB(1) TO KCB(NCB) WHICH ARE INSERTED WHEN THE INPUT
C      DATA ARE BEING READ .  THE INDEX OF THE LAST ALLOCATED NUMERIC
C      STORAGE UNIT OF ARRAY KSB IS NAMED MSB.  THE INDEX OF THE LAST
C      ALLOCATED NUMERIC STORAGE UNIT OF ARRAY KCB IS NAMED MCB.  THE
C      VALUES OF MSB AND MCB ARE GIVEN BY THE SIXTH AND SEVENTH STATEMENT
C      OF THE BLOCK DATA SUBROUTINE MODELB.  IF THE VALUE OF MSB OR MCB
C      IS CHANGED, IT MUST BE ADJUSTED IN ALL SUBROUTINES WHICH INCLUDE
C      THE COMMON BLOCK /MODELC/.
C
C DATE: 1990, FEBRUARY 6
C CODED BY LUDEK KLIMES
C
C=====================================================================
C
      SUBROUTINE MODEL1(LUN)
      INTEGER LUN
C
C SUBROUTINE MODEL1 READS THE INPUT DATA (1)-(8) FOR THE DESCRIPTION
C OF THE MODEL AND STORES THEM IN COMMON BLOCKS /MODELT/ AND /MODELC/.
```

```
C THEN IT CALLS SUBROUTINE SRFC1 (INCLUDED IN THE SUBROUTINE PACKAGE
C 'SRFC') TO READ THE INPUT DATA (9) FOR SMOOTH SURFACES AND TO STORE
C THEM IN THE MEMORY.  FINALY, IT CALLS SUBROUTINE PARM1 (INCLUDED IN
C THE SUBROUTINE PACKAGE 'PARM') TO READ THE INPUT DATA (10) FOR THE
C PARAMETERS OF THE MEDIUM AND TO STORE THEM IN THE MEMORY.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C THE INPUT PARAMETER IS NOT ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK /MODELT/:
      CHARACTER*80 TEXTM
      COMMON/MODELT/TEXTM
C COMMON BLOCK /MODELC/:
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCKS ARE DEFINED IN THIS
C SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL METR1,SRFC1,PARM1
C     SUBROUTINE METR1(KOOR)... SUBROUTINE PACKAGE 'METRIC'.
C     SRFC1 AND SUBSEQUENT ROUTINES... SUBROUTINE PACKAGE 'SRFC'.
C     PARM1 AND SUBSEQUENT ROUTINES... SUBROUTINE PACKAGE 'PARM'.
C
C ERROR MESSAGES:
C     310...  INSUFFICIENT MEMORY IN /MODELC/:
C             INSUFFICIENT MEMORY FOR THE INPUT DATA IN COMMON BLOCK
C             /MODELC/.  THE DIMENSIONS MSB OR MCB OF ARRAYS KSB OR KCB,
C             RESPECTIVELY, MUST BE ENLARGED.  SEE THE BLOCK DATA
C             SUBROUTINE MODELB.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,J,L
C
      READ(LUN,*) TEXTM
      READ(LUN,*) I,NEXPV,NEXPQ
      CALL METR1(I)
      READ(LUN,*) BOUNDM
      READ(LUN,*) NSRFCS
C
C     SIMPLE BLOCKS:
C     NUMBER OF SIMPLE BLOCKS
      READ(LUN,*) NSB
C     INITIALIZING MEMORY FOR INDICES OF SURFACES BOUNDING SIMPLE BLOCKS
      L=NSB+1
```

```
      DO 11 I=L,MSB
        KSB(I)=0
   11 CONTINUE
C     READING INDICES OF SURFACES BOUNDING SIMPLE BLOCKS:
      DO 14 J=1,NSB
        READ(LUN,*) (KSB(I),I=L,MSB)
        DO 12 I=L,MSB
          IF(KSB(I).EQ.0) THEN
            KSB(J)=I-1
            L=I
          GO TO 13
          END IF
   12   CONTINUE
        GO TO 99
   13   CONTINUE
   14 CONTINUE
C
C     COMPLEX BLOCKS:
C     NUMBER OF COMPLEX BLOCKS
      READ(LUN,*) NCB
C     INITIALIZING MEMORY FOR INDICES OF SIMPLE BLOCKS FORMING C. BLOCKS
      L=NCB+1
      DO 21 I=L,MCB
        KCB(I)=0
   21 CONTINUE
C     READING INDICES OF SIMPLE BLOCKS FORMING COMPLEX BLOCKS
      DO 24 J=1,NCB
        READ(LUN,*) (KCB(I),I=L,MCB)
        DO 22 I=L,MCB
          IF(KCB(I).EQ.0) THEN
            KCB(J)=I-1
            L=I
          GO TO 23
          END IF
   22   CONTINUE
        GO TO 99
   23   CONTINUE
   24 CONTINUE
C
C     SMOOTH SURFACES:
      CALL SRFC1(LUN,NSRFCS)
C
C     MATERIAL PARAMETERS:
      CALL PARM1(LUN,NCB)
      RETURN
C
   99 CONTINUE
      STOP 'ERROR 310 IN MODEL1: INSUFFICIENT MEMORY IN /MODELC/'
      END
C
C======================================================================
C
      INTEGER FUNCTION NSRFC()
C
C INTEGER FUNCTION NSRFC IS DESIGNED TO RETURN THE NUMBER OF SURFACES
C COVERING STRUCTURAL INTERFACES.
C
C NO INPUT.
C
C OUTPUT:
```

```
C      NSRFC...NUMBER OF SURFACES COVERING STRUCTURAL INTERFACES.
C
C COMMON BLOCK /MODELC/:
       INTEGER MSB,MCB
       PARAMETER (MSB=128)
       PARAMETER (MCB=128)
       INTEGER NEXPV,NEXPQ
       REAL BOUNDM(6)
       INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
       EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
       COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C      NO AUXILIARY STORAGE LOCATIONS.
C
       NSRFC=NSRFCS
       RETURN
       END
C
C=======================================================================
C
       SUBROUTINE BLOCK(COOR,ISRF1,ISB1,ISRF2,ISB2,ICB2,F)
       REAL COOR(3)
       INTEGER ISRF1,ISB1,ISB2,ICB2,ISRF2
       REAL F(10)
C
C THIS SUBROUTINE SEARCHES FOR THE SIMPLE BLOCK AND THE COMPLEX BLOCK IN
C WHICH A GIVEN POINT IS SITUATED.  THIS ROUTINE MAY BE ALSO USED TO
C DETERMINE THE INDEX OF A BLOCK TOUCHING A SPECIFIED BLOCK AT A GIVEN
C POINT SITUATED ON THE BOUNDARY OF THE SPECIFIED BLOCK (THE SITUATION
C WHICH MAY OCCUR WHEN A RAY IMPINGES ON A BOUNDARY OF A BLOCK).
C ANOTHER FUNCTION OF THE ROUTINE IS TO DETERMINE THE INDEX OF ANY OF
C THE SURFACES BOUNDING A BLOCK AND SEPARATING THE BLOCK FROM THE GIVEN
C POINT.
C
C INPUT:
C     COOR... ARRAY CONTAINING COORDINATES X1, X2, X3 OF A GIVEN POINT.
C     ISRF1...ISRF1.NE.0: INDEX OF THE SURFACE AT WHICH THE GIVEN POINT
C                IS SITUATED.  THE SIGN IS IGNORED.
C             ISRF1.EQ.0: THE GIVEN POINT IS NOT SITUATED AT ANY SURFACE
C     ISB1... FOR ISRF1.NE.0: INDEX OF A SIMPLE BLOCK BOUNDED BY THE
C                SURFACE IABS(ISRF1) - SEARCH FOR THE INDEX OF A
C                NEIGHBOURING SIMPLE BLOCK TOUCHING ISB1 AT THE GIVEN
C                POINT.
C             FOR ISRF1.EQ.0: INDEX OF AN ARBITRARY SIMPLE BLOCK OR
C                ISB1=0.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C     ISRF2...FOR THE GIVEN POINT NOT SITUATED INSIDE THE BLOCK ISB1 OR
C             AT ITS BOUNDARY, ISRF2 HAS THE MEANING OF THE INDEX OF ONE
C             OF THE SURFACES BOUNDING THE SIMPLE BLOCK ISB1 AND
C             SEPARATING THE GIVEN POINT FROM THE SIMPLE BLOCK ISB1,
```

```
C                SUPPLEMENTED BY A SIGN '+' OR '-' FOR THE SIMPLE BLOCK
C                ISB1 SITUATED AT THE POSITIVE OR NEGATIVE SIDE OF THE
C                SURFACE, RESPECTIVELY.
C                OTHERWISE ZERO.
C       ISB2...  FOR ISRF1.NE.0, ISRF2.EQ.0: INDEX OF A SIMPLE BLOCK
C                  NEIGHBOURING TO THE SIMPLE BLOCK ISB1 AND SEPARATED FROM
C                  THE SIMPLE BLOCK ISB1 BY SURFACE IABS(ISRF1) AT THE
C                  GIVEN POINT.
C                FOR ISRF1.NE.0, ISRF2.NE.0: INDEX OF A SIMPLE BLOCK IN
C                  WHICH THE GIVEN POINT IS SITUATED.  IN THIS CASE, THE
C                  GIVEN POINT MAY BE SITUATED EITHER INSIDE THE SIMPLE
C                  BLOCK ISB2 OR IN THE VICINITY OF ITS BOUNDARY FORMED BY
C                  THE SURFACE IABS(ISRF1).  FROM THE TWO POSSIBLE SIMPLE
C                  BLOCKS TOUCHING THE SURFACE IABS(ISRF1) AT THE GIVEN
C                  POINT, THAT BEING SITUATED AT THE SAME SIDE OF THE
C                  SURFACE ISRF1 AS THE SIMPLE BLOCK ISB1, IS SELECTED.
C                FOR ISRF1.EQ.0: INDEX OF THE SIMPLE BLOCK IN WHICH THE
C                  GIVEN POINT IS SITUATED.
C                IF THERE IS NO SIMPLE MATERIAL BLOCK OF THE ABOVE
C                PROPERTIES, ISB2=0.
C       ICB2...  INDEX OF THE COMPLEX BLOCK IN WHICH THE SIMPLE BLOCK ISB2
C                IS SITUATED.  ICB2=0 IF THE SIMPLE BLOCK ISB2 IS NOT
C                SITUATED IN ANY COMPLEX BLOCK.  FOR ISB2=0: ICB2=0.
C       F...     FOR ISRF2.NE.0: ARRAY CONTAINING THE VALUE AND PARTIAL
C                  DERIVATIVES F, F1, F3, F11, F12, F22, F13, F23, F33 OF
C                  THE FUNCTION DESCRIBING THE SURFACE IABS(ISRF2).
C                FOR ISRF2.EQ.0: UNDEFINED.
C
C COMMON BLOCK /MODELC/:
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL ISIDE,INTERF,SRFC2
      INTEGER ISIDE
C     FUNCTION ISIDE(ISRF,ISB)... THIS SUBROUTINE PACKAGE.
C     SUBROUTINE INTERF(COOR,ISRF1,ISB,ISRF2,F)... THIS SUBROUTINE
C               PACKAGE.
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C               SUBROUTINE PACKAGE 'SRFC'.
C
C ERROR MESSAGES:
C     311...  WRONG INDEX OF SURFACE:
C             ABSOLUTE VALUE OF THE INPUT PARAMETER ISRFC1 (INDEX OF THE
C             SURFACE) IS GREATER THAN THE NUMBER NSRFC OF THE SURFACES
C             COVERING STRUCTURAL INTERFACES.
C     312...  WRONG INDEX OF SIMPLE BLOCK:
C             PARAMETER ISB1 (INDEX OF THE SIMPLE BLOCK) IS EITHER
C             NEGATIVE OR GREATER THAN THE NUMBER NSB OF SIMPLE BLOCKS.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
```

```
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,J,I1,I2,ISIDE1,ISIDE2,ISRF
      REAL FAUX(10)
C
C     CHECKING INPUT VALUES:
      IF(ISRF1.LT.-NSRFCS.OR.ISRF1.GT.NSRFCS) THEN
        STOP 'ERROR 311 IN BLOCK: WRONG INDEX OF SURFACE'
      END IF
      IF(ISB1.LT.0.OR.ISB1.GT.NSB) THEN
        STOP 'ERROR 312 IN BLOCK: WRONG INDEX OF SIMPLE BLOCK'
      END IF
C
C     POSITION OF THE GIVEN POINT WITH RESPECT TO THE GIVEN SIMPLE BLOCK
C     ISB1:
      IF(ISB1.EQ.0) THEN
C       NO SIMPLE BLOCK SPECIFIED:
        ISRF2=0
      ELSE
        CALL INTERF(COOR,ISRF1,ISB1,ISRF2,F)
        IF(ISRF1.EQ.0) THEN
          IF(ISRF2.EQ.0) THEN
C           THE POINT IS INSIDE SIMPLE BLOCK ISB1:
            ISB2=ISB1
            GO TO 10
          END IF
        ELSE
          IF(ISRF2.NE.0) THEN
C           THE POINT BEING SITUATED AT SURFACE ISRF1 BOUNDING SIMPLE
C           BLOCK ISB1 IS NOT SITUATED AT THE BOUNDARY OF SIMPLE BLOCK
C           ISB1:
            ISIDE1=ISIDE(ISRF1,ISB1)
          ELSE
C           THE POINT IS SITUATED AT THE BOUNDARY OF SIMPLE BLOCK ISB1:
            ISIDE1=-ISIDE(ISRF1,ISB1)
          END IF
        END IF
      END IF
C
C     SEARCH FOR THE SIMPLE BLOCK IN WHICH THE GIVEN POINT IS
C     SITUATED:
      I2=MAX0(ISB1-1,NSB-ISB1)
      DO 2 J=1,I2
        DO 1 I=1,-1,-2
          ISB2=ISB1+I*J
          IF(ISB2.GT.0.AND.ISB2.LE.NSB) THEN
            IF(ISRF1.NE.0) THEN
              ISIDE2=ISIDE(ISRF1,ISB2)
            END IF
            IF(ISRF1.EQ.0.OR.ISIDE1.EQ.ISIDE2) THEN
              CALL INTERF(COOR,ISRF1,ISB2,ISRF,FAUX)
              IF(ISRF.EQ.0) GO TO 10
            END IF
          END IF
    1   CONTINUE
    2 CONTINUE
C     NO SIMPLE BLOCK HAS BEEN FOUND:
      ISB2=0
      ICB2=0
```

```
      RETURN
C
C     DETERMINATION OF THE COMPLEX BLOCK:
   10 CONTINUE
      DO 12 J=1,NCB
        I1=KCB(J-1)+1
        I2=KCB(J)
        DO 11 I=I1,I2
          IF(KCB(I).EQ.ISB2) THEN
            ICB2=J
            RETURN
          END IF
   11   CONTINUE
   12 CONTINUE
C     NO COMPLEX BLOCK:
      ICB2=0
C
      RETURN
      END
C
C=======================================================================
C
      INTEGER FUNCTION ISIDE(ISRF,ISB)
      INTEGER ISRF,ISB
C
C THIS IS AN AUXILIARY FUNCTION TO THE SUBROUTINE BLOCK.
C THIS FUNCTION DETERMINES THE MUTUAL POSITION OF A SURFACE AND A SIMPLE
C BLOCK.
C
C INPUT:
C     ISRF... INDEX OF THE SURFACE.  THE SIGN IS IGNORED.
C     ISB...  INDEX OF THE SIMPLE BLOCK.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C     ISIDE...ISIDE=-1: THE SIMPLE BLOCK IS BOUNDED BY THE SURFACE AND
C                IS SITUATED ON ITS NEGATIVE SIDE.
C             ISIDE= 1: THE SIMPLE BLOCK IS BOUNDED BY THE SURFACE AND
C                IS SITUATED ON ITS POSITIVE SIDE.
C             ISIDE= 2: THE SIMPLE BLOCK IS NOT BOUNDED BY THE SURFACE.
C
C COMMON BLOCK /MODELC/:
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
```

```
      INTEGER IS,LS,MS
C
      LS=KSB(ISB-1)+1
      MS=KSB(ISB)
C
C     LOOP FOR SURFACES BOUNDING SIMPLE BLOCK ISB:
      DO 1 IS=LS,MS
        IF(IABS(KSB(IS)).EQ.IABS(ISRF)) THEN
          ISIDE=ISIGN(1,KSB(IS))
          RETURN
        END IF
    1 CONTINUE
C
      ISIDE=2
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE INTERF(COOR,ISRF1,ISB,ISRF2,F)
      REAL COOR(3),F(10)
      INTEGER ISRF1,ISB,ISRF2
C
C THIS IS AN AUXILIARY SUBROUTINE TO THE SUBROUTINE BLOCK.
C THIS SUBROUTINE DETERMINES THE POSITION OF A GIVEN POINT WITH RESPECT
C TO A GIVEN SIMPLE BLOCK.
C
C INPUT:
C     COOR... ARRAY CONTAINING COORDINATES X1, X2, X3 OF A GIVEN POINT.
C     ISRF1...ISRF1.NE.0: INDEX OF THE SURFACE AT WHICH THE GIVEN POINT
C               IS SITUATED.  THE SIGN IS IGNORED.
C             ISRF1.EQ.0: THE GIVEN POINT IS NOT SITUATED AT ANY
C               SURFACE.
C     ISB...  INDEX OF THE GIVEN SIMPLE BLOCK.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C     ISRF2...INDEX OF A SURFACE SEPARATING THE GIVEN POINT AND THE
C               SIMPLE BLOCK ISB, SUPPLEMENTED BY A SIGN '+' OR '-' FOR
C               THE SIMPLE BLOCK ISB1 SITUATED AT THE POSITIVE OR
C               NEGATIVE SIDE OF THE SURFACE, RESPECTIVELY.
C             ISRF2=0  IF THE GIVEN POINT IS SITUATED INSIDE THE SIMPLE
C               BLOCK ISB.
C     F...    FOR ISRF2.NE.0: ARRAY CONTAINING THE VALUE AND PARTIAL
C               DERIVATIVES F, F1, F3, F11, F12, F22, F13, F23, F33 OF
C               THE FUNCTION DESCRIBING THE SURFACE IABS(ISRF2) AT THE
C               GIVEN POINT.
C             FOR ISRF2.EQ.0: UNDEFINED.
C
C COMMON BLOCK /MODELC/:
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
```

```
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SRFC2
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'SRFC'.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER IS,JS,KS,LS,MS
C
      LS=KSB(ISB-1)+1
      MS=KSB(ISB)
C
C     LOOP FOR SURFACES BOUNDING SIMPLE BLOCK ISB:
      DO 1 IS=LS,MS
        KS=KSB(IS)
        JS=IABS(KS)
        IF(JS.NE.IABS(ISRF1)) THEN
          CALL SRFC2(JS,COOR,F)
          IF(F(1)*FLOAT(KS).LT.0.) THEN
            ISRF2=KS
            RETURN
          END IF
        END IF
    1 CONTINUE
C
      ISRF2=0
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)
      INTEGER IWAVE
      REAL UP(10),US(10),QP,QS,VP,VS,VD(10),QL
C
C THIS SUBROUTINE TRANSFORMS THE VALUES OF PARAMETERS OF THE MEDIUM INTO
C VELOCITIES AND LOSS FACTORS.
C
C INPUT:
C     IWAVE...TYPE OF WAVE.
C             IWAVE.GE.0: P WAVE,
C             IWAVE.LT.0: S WAVE.
C     UP,US...POWERS OF P AND S WAVE VELOCITIES AND THEIR FIRST AND
C             SECOND PARTIAL DERIVATIVES (THE EXPONENT OF THE POWERS IS
C             NEXPV, SEE 'INPUT DATA FOR THE MODEL'), IN ORDER U, U1,
C             U2, U3, U11, U12, U22, U13, U23, U33.
C     QP,QS...POWERS OF THE LOSS FACTORS OF P AND S WAVES (THE EXPONENT
C             OF THE POWERS IS NEXPQ, SEE 'INPUT DATA FOR THE MODEL').
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C     VP,VS...P AND S WAVE VELOCITIES.
C     VD...   VELOCITY AND ITS FIRST AND SECOND PARTIAL DERIVATIVES
C             ORDERED AS UP, US, CORRESPONDING TO THE WAVE SPECIFIED BY
C             IWAVE, IN ORDER V, V1, V2, V3, V11, V12, V22, V13, V23,
C             V33.
```

```
C      QL...   LOSS FACTOR CORRESPONDING TO THE WAVE SPECIFIED BY IWAVE.
C
C COMMON BLOCK /MODELC/:
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL FPOWER
C     FPOWER... SUBROUTINE PACKAGE 'VAL'.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATION:
      REAL POWER
C
      POWER=FLOAT(NEXPV)
      IF(IWAVE.GE.0) THEN
        CALL FPOWER(10,UP,POWER,VD)
        VP=VD(1)
        CALL FPOWER(1,US,POWER,VS)
        CALL FPOWER(1,QP,FLOAT(NEXPQ),QL)
      ELSE
        CALL FPOWER(1,UP,POWER,VP)
        CALL FPOWER(10,US,POWER,VD)
        VS=VD(1)
        CALL FPOWER(1,QS,FLOAT(NEXPQ),QL)
      END IF
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'SRFC' FOR SPECIFICATION AND INTERPOLATION OF
C SMOOTH SURFACES IN THE MODEL IN RECTANGULAR GRIDS.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C     SRFC1...SUBROUTINE READING THE INPUT DATA FOR SMOOTH SURFACES.
C     SRFC2...SUBROUTINE EVALUATING THE FUNCTION VALUES AND THEIR FIRST
C             AND SECOND DERIVATIVES.  OUTSIDE THE SPECIFIED RECTANGULAR
C             GRID, THE FUNCTIONS ARE LINEAR ALONG STRAIGHT LINES
C             PERPENDICULAR TO THE BOUNDARY OF THE GRID.
C SUBROUTINES SRFC1 AND SRFC2 SUPPORTING THE COMPLETE RAY TRACING
C ALGORITHM ONLY MEDIATE THE WORK OF SUBROUTINES VAL1 AND VAL2 WHICH
C MUST BE APPENDED.  IN ADDITION, SUBROUTINES CURVN1 (OR ITS ALTERNATIVE
C CURVB1), CURV2D (OR ITS ALTERNATIVE CURVBD), SURFB1, SURFBD, VAL3B1,
C VAL3BD, VGEN, TERMS, SNHCSH, TRIDEC, TRISOL, DSPLNZ, INTRVL FROM THE
C SUBROUTINE PACKAGE 'FITPACK' BY ALAN KAYLOR CLINE, DEPARTMENT OF
C COMPUTER SCIENCES, UNIVERSITY OF TEXAS AT AUSTIN, ARE USED.  IN THE
C COMPLETE RAY TRACING, THIS SOFTWARE PACKAGE 'SRFC' MAY BE  REPLACED BY
C ANY USER-DEFINED PACKAGE CONTAINING SUBROUTINES SRFC1 AND SRFC2 WITH
C THE SAME NUMBER, TYPE AND MEANING OF THEIR PARAMETERS AS IN THIS
C PACKAGE.
C
C INPUT DATA (READ IN BY SUBROUTINE SRFC1):
C     THESE INPUT DATA DEFINE THE SURFACES.  THEY ARE READ IN BY
C     SUBROUTINE SRFC1.  THE NUMBER NSRFC OF THE SURFACES TO BE DEFINED
C     IS AN INPUT ARGUMENT OF SUBROUTINE SRFC1.  THE DATA ARE READ IN BY
C     THE LIST DIRECTED INPUT (FREE FORMAT).
C (1) NSRFC-TIMES (I.E. ONCE FOR EACH SURFACE) INPUT DATA (1A)+(1B):
C (1A) TEXTG,ISRFC
C     IDENTIFICATION OF THE SURFACE.
C     TEXTG...ANY STRING. ITS FIRST 3 CHARACTERS MUST DIFFER FROM 'END'.
C     ISRFC...INDEX OF THE SURFACE.
C (1B) 'INPUT DATA FOR ONE SURFACE', SEE BELOW.
C (2) TEXTE,AUX
C     END OF DATA.
C     TEXTE...STRING, THE FIRST 3 CHARACTERS OF WHICH MUST BE UPPER-CASE
C             'END'.
C     AUX...  ANY NUMBER OR A SLASH.
C FOR AN EXAMPLE REFER TO THE SAMPLE INPUT DATA FOR THE MODEL.
C
C INPUT DATA FOR ONE SURFACE:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C     THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C     I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C     INTEGER.  OTHERWISE, THE INPUT PARAMETER IS OF THE TYPE REAL.
C (1) IVAR1,IVAR2,IVAR3,SIGMA
C     THE FORM OF THE FUNCTION.
C     IVAR1,IVAR2,IVAR3... DENOTE THE FORM OF THE FUNCTION. THE FUNCTION
C             MUST BE OF THE FORM
C               F(X1,X2,X3) = W(A1,A2,A3)-B1-B2-B3   .
C             X1, X2, X3 ARE THE GENERAL COORDINATES. EACH OF A1, A2,
C             A3, B1, B2, B3 MUST BE EITHER:  (A) ONE OF GENERAL
C             COORDINATES X1, X2, X3,  OR (B) MUST BE LEFT OUT.  AT MOST
C             3 OF PARAMETERS A1-B3 MAY BE OF KIND (A).  NOTE THAT IVAR1
C             CONTROLS THE TYPE OF A1 AND B1, IVAR2 CONTROLS THE TYPE OF
C             A2 AND B2, IVAR3 CONTROLS THE TYPE OF A3 AND B3.
C             FOR IVAR1.EQ.0: A1, B1 ARE EMPTY (LEFT OUT),
```

```
C               FOR IVAR1.EQ.1: A1=X1, B1 IS EMPTY,
C               FOR IVAR1.EQ.2: A1=X2, B1 IS EMPTY,
C               FOR IVAR1.EQ.3: A1=X3, B1 IS EMPTY,
C               FOR IVAR1.EQ.-1: B1=X1, A1 IS EMPTY,
C               FOR IVAR1.EQ.-2: B1=X2, A1 IS EMPTY,
C               FOR IVAR1.EQ.-3: B1=X3, A1 IS EMPTY,
C               THE MEANING OF THE PARAMETERS IVAR2, IVAR3 IS SIMILAR.
C               EXAMPLES:
C               IVAR1: IVAR2: IVAR3: THE FORM OF THE FUNCTION:
C                 1      2      3     F(X1,X2,X3)=W(X1,X2,X3)
C                 3      1      2     F(X1,X2,X3)=W(X3,X1,X2)
C                 1      2      0     F(X1,X2,X3)=W(X1,X2)
C                 1      2     -3     F(X1,X2,X3)=W(X1,X2)-X3
C                 1     -3      2     F(X1,X2,X3)=W(X1,X2)-X3
C               FUNCTION W IS INTERPOLATED BY MEANS OF SPLINES UNDER
C               TENSION.
C      SIGMA...IS THE TENSION FACTOR (ITS SIGN IS IGNORED).  THIS VALUE
C               INDICATES THE CURVINESS DESIRED.  IF ABS(SIGMA) IS NEARLY
C               ZERO (E.G. 0.001), THE RESULTING SURFACE IS APPROXIMATELY
C               THE TENSOR PRODUCT OF CUBIC SPLINES.  IF ABS(SIGMA) IS
C               LARGE (E.G. 50.), THE RESULTING SURFACE IS APPROXIMATELY
C               TRI-LINEAR.  IF SIGMA EQUALS ZERO, TENSOR PRODUCTS OF
C               CUBIC SPLINES RESULT.  A RECOMMENDED VALUE FOR SIGMA IS
C               APPROXIMATELY 1. IN ABSOLUTE VALUE.
C (2) NX(1),...,NX(NVAR)
C      THE NUMBERS OF GRID COORDINATES FOR THE INTERPOLATION.
C      THIS INPUT IS PERFORMED IF AT LEAST ONE OF IVAR1, IVAR2, IVAR3 IS
C      POSITIVE.
C      EACH OF NX(1),...,NX(NVAR) CORRESPONDS TO ONE POSITIVE VALUE OF
C      IVAR1, IVAR2, IVAR3 AND SPECIFIES THE NUMBER OF GRID COORDINATES
C      CORRESPONDING TO THAT INDEPENDENT VARIABLE OF FUNCTION W, SEE (1).
C      THE SIGN OF NX(1),...,NX(NVAR) IS IGNORED. NVAR (.LE.3) IS THE
C      NUMBER OF POSITIVE VALUES OF THE ABOVE QUANTITIES IVAR1, IVAR2,
C      IVAR3, I.E. THE NUMBER OF INDEPENDENT VARIABLES OF FUNCTION W,
C      SEE (1).
C (3) X1(1),...,X1(NX(1))
C      THE GRID COORDINATES CORRESPONDING TO THE FIRST INDEPENDENT
C      VARIABLE OF FUNCTION W, SEE (1).
C      THIS INPUT IS PERFORMED IF NX(1) IS SPECIFIED, SEE (2), AND IS NOT
C      ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (4) X2(1),...,X2(NX(2))
C      THE GRID COORDINATES CORRESPONDING TO THE SECOND INDEPENDENT
C      VARIABLE OF FUNCTION W, SEE (1).
C      THIS INPUT IS PERFORMED IF NX(2) IS SPECIFIED, SEE (2), AND IS NOT
C      ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (5) X3(1),...,X3(NX(3))
C      THE GRID COORDINATES CORRESPONDING TO THE THIRD INDEPENDENT
C      VARIABLE OF FUNCTION W, SEE (1).
C      THIS INPUT IS PERFORMED IF NX(3) IS SPECIFIED, SEE (2), AND IS NOT
C      ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (6) (((W(I,J,K),I=1,MAX(NX(1),1)),J=1,MAX(NX(2),1)),K=1,MAX(NX(3),1))
C      THE VALUES OF FUNCTION W AT GRID POINTS. FUNCTION VALUE W(I,J,K)
C      CORRESPONDS TO POINT (X1(I),X2(J),X3(K)).
C
C DATE: 1988, JUNE 1
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
      SUBROUTINE SRFC1(LUN,NSRFC)
```

```
      INTEGER LUN,NSRFC
C
C THIS SUBROUTINE READS THE INPUT DATA FOR THE SMOOTH SURFACES,
C DETERMINES THE PARAMETERS NECESSARY TO COMPUTE AN INTERPOLATORY
C FUNCTION ON A THREE DIMENSIONAL RECTANGULAR GRID, AND STORES THEM IN
C THE MEMORY.  THE FUNCTION DETERMINED CAN BE REPRESENTED AS A TENSOR
C PRODUCT OF SPLINES UNDER TENSION.  FOR ACTUAL MAPPING OF POINTS IT IS
C NECESSARY TO CALL THE SUBROUTINE SRFC2, WHICH ALSO RETURNS THE FIRST
C AND SECOND PARTIAL DERIVATIVES.  SUBROUTINE SRFC1 MAY BE CALLED
C SEVERAL TIMES.  THE SURFACES ARE INDEXED SUCCESIVELY, FOLLOWING THE
C SURFACES DEFINED DURING THE PREVIOUS INVOCATIONS.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C     NSRFC...NUMBER OF THE SURFACES FOR WHICH THE INPUT DATA ARE
C             SPECIFIED DURING THE CURRENT INVOCATION OF SRFC1.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL VAL1
C     VAL1, SORTV, READV... SUBROUTINE PACKAGE 'VAL'.
C     CURVN1 OR CURVB1 (ALTERNATIVES), SURFB1, VAL3B1, SNHCSH, VGEN,
C               TERMS, TRIDEC, TRISOL... SUBROUTINE PACKAGE 'FITPACK'.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      CHARACTER*3 TFUNCT
      DATA TFUNCT/'   '/
C
      CALL VAL1(LUN,1,NSRFC,1,TFUNCT)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SRFC2(ISRFC,COOR,F)
      INTEGER ISRFC
      REAL COOR(3),F(10)
C
C THIS SUBROUTINE EVALUATES THE FUNCTIONS DESCRIBING VARIOUS SMOOTH
C SURFACES IN THE MODEL AT A GIVEN POINT. THE THREE FIRST AND SIX SECOND
C PARTIAL DERIVATIVES ARE ALSO EVALUATED.  THE SPECIFIED FUNCTIONS ARE
C REPRESENTED AS A TENSOR PRODUCT OF SPLINES UNDER TENSION.  THE
C COEFFICIENTS OF THESE FUNCTIONS ARE PREPARED IN SUBROUTINE SRFC1, IN
C WHICH THE INPUT DATA CONCERNING THE FUNCTION OF EACH SURFACE ARE READ
C IN.
C
C INPUT:
C     ISRFC...INDEX OF A SURFACE.
C     COOR... ARRAY CONTAINING COORDINATES X1, X2, X3 OF THE GIVEN
C             POINT.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
```

```
C OUTPUT:
C     F...      THE VALUE AND THE FIRST AND SECOND PARTIAL DERIVATIVES F,
C               F1, F2, F3, F11, F12, F22, F13, F23, F33 OF THE FUNCTION
C               F(X1,X2,X3) DETERMINING THE SURFACE ISRFC AT THE GIVEN
C               POINT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL VAL2
C     VAL2... SUBROUTINE PACKAGE 'VAL'.
C     CURV2D OR CURVBD (ALTERNATIVES), SURFBD, VAL3BD, SNHCSH, DSPLNZ,
C              INTRVL... SUBROUTINE PACKAGE 'FITPACK'.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATION:
      REAL POWER
C
      CALL VAL2(1,ISRFC,1,COOR,F,POWER)
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'PARM' FOR SPECIFICATION AND INTERPOLATION OF THE
C MATERIAL PARAMETERS OF THE MODEL IN RECTANGULAR GRIDS.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C     PARM1...SUBROUTINE READING THE INPUT DATA FOR THE MATERIAL
C             PARAMETERS OF THE MODEL.
C     PARM2...SUBROUTINE EVALUATING THE MATERIAL PARAMETERS INCLUDING
C             THEIR FIRST AND SECOND DERIVATIVES.  OUTSIDE THE SPECIFIED
C             RECTANGULAR GRID, THE FUNCTIONS ARE LINEAR ALONG STRAIGHT
C             LINES PERPENDICULAR TO THE BOUNDARY OF THE GRID.  THE
C             FUNCTIONS MAY BE EMBEDDED:  THE INDEPENDENT VARIABLE OF
C             THE FUNCTION MAY BE ANOTHER MATERIAL PARAMETER OF THE SAME
C             COMPLEX BLOCK FOREGOING IN THE INPUT DATA.
C SUBROUTINES PARM1 AND PARM2 SUPPORTING THE COMPLETE RAY TRACING
C ALGORITHM ONLY MEDIATE THE WORK OF SUBROUTINES VAL1, VAL2 AND FPOWER
C WHICH MUST BE APPENDED.  IN ADDITION, SUBROUTINES CURVN1 (OR ITS
C ALTERNATIVE CURVB1), CURV2D (OR ITS ALTERNATIVE CURVBD), SURFB1,
C SURFBD, VAL3B1, VAL3BD, VGEN, TERMS, SNHCSH, TRIDEC, TRISOL, DSPLNZ,
C INTRVL FROM THE SUBROUTINE PACKAGE 'FITPACK' BY ALAN KAYLOR CLINE,
C DEPARTMENT OF COMPUTER SCIENCES, UNIVERSITY OF TEXAS AT AUSTIN, ARE
C USED.  IN THE COMPLETE RAY TRACING, THIS SOFTWARE PACKAGE 'PARM' MAY
C BE REPLACED BY ANY USER-DEFINED PACKAGE CONTAINING SUBROUTINES PARM1
C AND PARM2 WITH THE SAME NUMBER, TYPE AND MEANING OF THEIR PARAMETERS
C AS IN THIS PACKAGE.
C
C INPUT DATA (READ IN BY SUBROUTINE PARM1):
C     THESE INPUT DATA DEFINE THE COMPLEX BLOCKS.  THEY ARE READ IN BY
C     SUBROUTINE PARM1.  THE NUMBER NCB OF THE COMPLEX BLOCKS TO BE
C     DEFINED IS AN INPUT ARGUMENT OF SUBROUTINE PARM1.  THE DATA ARE
C     READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).
C (1) NCB-TIMES (I.E. ONCE FOR EACH COMPLEX BLOCK) INPUT DATA (1A)+(1B):
C (1A) TEXTG,ICB
C     IDENTIFICATION OF THE COMPLEX BLOCK.
C     TEXTG...ANY STRING.  ITS FIRST 3 CHARACTERS MUST DIFFER FROM 'VP '
C             'VS ', 'DEN', 'QP ', 'QS ', 'END'.
C     ICB...  INDEX OF THE COMPLEX BLOCK.
C (1B) SEVERAL TIMES 'INPUT DATA FOR ONE MATERIAL PARAMETER', SEE BELOW.
C     AT LEAST ONE OF THE VELOCITIES MUST BE SPECIFIED.  THE DEFAULT
C     VALUES OF PARAMETERS NOT SPECIFIED ARE:
C     P WAVE VELOCITY: VP=1.73205*VS,
C     S WAVE VELOCITY: VS=0.57735*VP,
C     DENSITY: RHO=1.0,
C     P WAVE LOSS FACTOR (IF S WAVE LOSS FACTOR IS SPECIFIED):
C               QP=1.333333*QS*(US(1)/UP(1))**2,
C     S WAVE LOSS FACTOR (IF P WAVE LOSS FACTOR IS SPECIFIED):
C               QS=0.750000*QP*(UP(1)/US(1))**2,
C     P AND S WAVE LOSS FACTORS (IF NONE OF THEM IS SPECIFIED):
C               QP=0.0, QS=0.0.
C (2) TEXTE,AUX
C     END OF DATA.
C     TEXTE...STRING, THE FIRST 3 CHARACTERS OF WHICH MUST BE UPPER-CASE
C             'END'.
C     AUX...  ANY NUMBER OR A SLASH.
C FOR AN EXAMPLE REFER TO THE SAMPLE INPUT DATA FOR THE MODEL.
C
C INPUT DATA FOR ONE MATERIAL PARAMETER:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
```

```
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C     THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C     I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C     INTEGER.  OTHERWISE (EXCEPT TEXTF), THE INPUT PARAMETER IS OF THE
C     TYPE REAL.
C (1) TEXTF,POWER
C     PHYSICAL MEANING OF THE FUNCTION.
C     TEXTF...STRING IDENTIFYING WHICH MATERIAL PARAMETER THE FUNCTION
C             DESCRIBES. ONLY THE FIRST 3 CHARACTERS ARE SIGNIFICANT.
C             THE FIRST 3 CHARACTERS OF THE STRING MUST BE:
C             'VP ' FOR P WAVE VELOCITY,
C             'VS ' FOR S WAVE VELOCITY,
C             'DEN' FOR DENSITY,
C             'QP ' FOR P WAVE LOSS FACTOR,
C             'QS ' FOR S WAVE LOSS FACTOR.
C     POWER...THE SPECIFIED FUNCTION IS EQUAL TO THE POWER-TH POWER OF
C             THE MATERIAL PARAMETER.
C             EXAMPLES:
C               FOR P WAVE VELOCITY  TEXTF='VP ' AND  POWER=1.0,
C               FOR P WAVE SLOWNESS  TEXTF='VP ' AND  POWER=-1.0,
C               FOR S WAVE QUADRATIC SLOWNESS  TEXTF='VS ' AND
C                 POWER=-2.0,
C               FOR P WAVE LOSS FACTOR  TEXTF='QP ' AND  POWER=1.0,
C               FOR S WAVE QUALITY FACTOR  TEXTF='QS ' AND  POWER=-1.0.
C (2) IVAR1,IVAR2,IVAR3,SIGMA
C     THE FORM OF THE FUNCTION.
C     IVAR1,IVAR2,IVAR3... DENOTE THE FORM OF THE FUNCTION. THE FUNCTION
C             MUST BE OF THE FORM
C               F(X1,X2,X3) = W(A1,A2,A3)-B1-B2-B3   .
C             X1, X2, X3 ARE THE GENERAL COORDINATES. EACH OF A1, A2,
C             A3, B1, B2, B3 MUST BE EITHER:  (A) ONE OF GENERAL
C             COORDINATES X1, X2, X3, (B) ANOTHER PREVIOUSLY DEFINED
C             FUNCTION F(X1,X2,X3) OF THE SAME COMPLEX BLOCK,  OR (C)
C             MUST BE LEFT OUT.  AT MOST 3 OF PARAMETERS A1-B3 MAY BE OF
C             KIND (A) OR (B).  NOTE THAT IVAR1 CONTROLS THE TYPE OF A1
C             AND B1, IVAR2 CONTROLS THE TYPE OF A2 AND B2, IVAR3
C             CONTROLS THE TYPE OF A3 AND B3.
C             FOR IVAR1.EQ.0: A1, B1 ARE EMPTY (LEFT OUT).
C             FOR IVAR1.EQ.1: A1=X1, B1 IS EMPTY.
C             FOR IVAR1.EQ.2: A1=X2, B1 IS EMPTY.
C             FOR IVAR1.EQ.3: A1=X3, B1 IS EMPTY.
C             FOR IVAR1.GE.4: A1=F(X1,X2,X3), WHERE F(X1,X2,X3) IS
C               ANOTHER FUNCTION OF THE SAME COMPLEX BLOCK DEFINED IN
C               THE INPUT DATA AS THE (IVAR1-3)-TH FUNCTION OF THE
C               COMPLEX BLOCK. B1 IS EMPTY.
C               EXAMPLE:
C                 IF DENSITY=1.7+0.2*VP THEN THE INTERPOLATED FUNCTION
C                 IS W(A1,A2,A3)=1.7+0.2*A1 WITH THE INDEPENDENT
C                 VARIABLE A1=VP(X1,X2,X3).  THIS IS SPECIFIED BY
C                 IVAR1=4, IVAR2=0, IVAR3=0 IF VP IS THE FIRST READ IN
C                 PARAMETER, BY IVAR1=5, IVAR2=0, IVAR3=0 IF VP IS THE
C                 SECOND READ IN PARAMETER, ETC.
C                 THE POSSIBLE ALTERNATIVES ARE W(A1,A2,A3)=1.7+0.2*A2
C                 WITH A2=VP(X1,X2,X3) SPECIFIED BY IVAR1=0,
C                 IVAR2=(4 OR 5 OR THE LIKE), IVAR3=0, AND
C                 W(A1,A2,A3)=1.7+0.2*A3 WITH A3=VP(X1,X2,X3) SPECIFIED
C                 BY IVAR1=0, IVAR3=2, IVAR3=(4 OR 5 OR THE LIKE).
C             FOR IVAR1.EQ.-1: B1=X1, A1 IS EMPTY.
C             FOR IVAR1.EQ.-2: B1=X2, A1 IS EMPTY.
C             FOR IVAR1.EQ.-3: B1=X3, A1 IS EMPTY.
```

```
C                FOR IVAR1.GE.-4: B1=F(X1,X2,X3), WHERE F(X1,X2,X3) IS
C                  ANOTHER FUNCTION OF THE SAME COMPLEX BLOCK DEFINED IN
C                  THE INPUT DATA AS THE (-IVAR1-3)-TH FUNCTION OF THE
C                  COMPLEX BLOCK. A1 IS EMPTY.
C                THE MEANING OF THE PARAMETERS IVAR2, IVAR3 IS SIMILAR.
C                EXAMPLES:
C                IVAR1: IVAR2: IVAR3: THE FORM OF THE FUNCTION:
C                  1      2      3     F(X1,X2,X3)=W(X1,X2,X3)
C                  3      1      2     F(X1,X2,X3)=W(X3,X1,X2)
C                  1      2      0     F(X1,X2,X3)=W(X1,X2)
C                  5      0      0     F(X1,X2,X3)=W(F2(X1,X2,X3)), WHERE
C                F2(X1,X2,X3) IS THE SECOND MATERIAL PARAMETER OF THE
C                COMPLEX BLOCK DEFINED IN THE INPUT DATA. FUNCTION W IS
C                INTERPOLATED BY MEANS OF SPLINES UNDER TENSION.
C       SIGMA...IS THE TENSION FACTOR (ITS SIGN IS IGNORED).  THIS VALUE
C                INDICATES THE CURVINESS DESIRED.  IF ABS(SIGMA) IS NEARLY
C                ZERO (E.G. 0.001), THE RESULTING SURFACE IS APPROXIMATELY
C                THE TENSOR PRODUCT OF CUBIC SPLINES.  IF ABS(SIGMA) IS
C                LARGE (E.G. 50.), THE RESULTING SURFACE IS APPROXIMATELY
C                TRI-LINEAR.  IF SIGMA EQUALS ZERO, TENSOR PRODUCTS OF
C                CUBIC SPLINES RESULT.  A RECOMMENDED VALUE FOR SIGMA IS
C                APPROXIMATELY 1. IN ABSOLUTE VALUE.
C (3) NX(1),...,NX(NVAR)
C     THE NUMBERS OF GRID COORDINATES FOR THE INTERPOLATION.
C     THIS INPUT IS PERFORMED IF AT LEAST ONE OF IVAR1, IVAR2, IVAR3 IS
C     POSITIVE.
C     EACH OF NX(1),...,NX(NVAR) CORRESPONDS TO ONE POSITIVE VALUE OF
C     IVAR1, IVAR2, IVAR3 AND SPECIFIES THE NUMBER OF GRID COORDINATES
C     CORRESPONDING TO THAT INDEPENDENT VARIABLE OF FUNCTION W, SEE (2).
C     THE SIGN OF NX(1),...,NX(NVAR) IS IGNORED.  NVAR (.LE.3) IS THE
C     NUMBER OF POSITIVE VALUES OF THE ABOVE QUANTITIES IVAR1, IVAR2,
C     IVAR3, I.E. THE NUMBER OF INDEPENDENT VARIABLES OF FUNCTION W,
C     SEE (1).
C (4) X1(1),...,X1(NX(1))
C     THE GRID COORDINATES CORRESPONDING TO THE FIRST INDEPENDENT
C     VARIABLE OF FUNCTION W, SEE (2).
C     THIS INPUT IS PERFORMED IF NX(1) IS SPECIFIED, SEE (3), AND IS NOT
C     ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (5) X2(1),...,X2(NX(2))
C     THE GRID COORDINATES CORRESPONDING TO THE SECOND INDEPENDENT
C     VARIABLE OF FUNCTION W, SEE (2).
C     THIS INPUT IS PERFORMED IF NX(2) IS SPECIFIED, SEE (3), AND IS NOT
C     ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (6) X3(1),...,X3(NX(3))
C     THE GRID COORDINATES CORRESPONDING TO THE THIRD INDEPENDENT
C     VARIABLE OF FUNCTION W, SEE (2).
C     THIS INPUT IS PERFORMED IF NX(3) IS SPECIFIED, SEE (3), AND IS NOT
C     ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (7) (((W(I,J,K),I=1,MAX(NX(1),1)),J=1,MAX(NX(2),1)),K=1,MAX(NX(3),1))
C     THE VALUES OF FUNCTION W AT GRID POINTS. FUNCTION VALUE W(I,J,K)
C     CORRESPONDS TO POINT (X1(I),X2(J),X3(K)).
C
C DATE: 1988, JUNE 3
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
      SUBROUTINE PARM1(LUN,NCB)
      INTEGER LUN,NCB
C
```

```
C THIS SUBROUTINE READS THE INPUT DATA FOR THE DISTRIBUTIONS OF THE
C MATERIAL PARAMETERS (P AND S WAVE VELOCITIES, DENSITY, P AND S WAVE
C LOSS FACTORS) IN THE COMPLEX BLOCKS, DETERMINES THE PARAMETERS
C NECESSARY TO COMPUTE AN INTERPOLATORY FUNCTION ON A THREE DIMENSIONAL
C RECTANGULAR GRID, AND STORES THEM IN THE MEMORY.  THE FUNCTION
C DETERMINED CAN BE REPRESENTED AS A TENSOR PRODUCT OF SPLINES UNDER
C TENSION.  THE FUNCTIONS MAY BE EMBEDDED.  FOR ACTUAL MAPPING OF POINTS
C IT IS NECESSARY TO CALL THE SUBROUTINE PARM2, WHICH ALSO RETURNS THE
C FIRST AND SECOND PARTIAL DERIVATIVES.  SUBROUTINE PARM1 MAY BE CALLED
C SEVERAL TIMES.  THE COMPLEX BLOCKS ARE INDEXED SUCCESIVELY, FOLLOWING
C THE COMPLEX BLOCKS DEFINED DURING THE PREVIOUS INVOCATIONS.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C     NCB...  NUMBER OF THE MATERIAL COMPLEX BLOCKS FOR WHICH THE INPUT
C             DATA ARE SPECIFIED DURING THE CURRENT INVOCATION OF PARM1.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL VAL1
C     VAL1, SORTV, READV... SUBROUTINE PACKAGE 'VAL'.
C     CURVN1 OR CURVB1 (ALTERNATIVES), SURFB1, VAL3B1, SNHCSH, VGEN,
C               TERMS, TRIDEC, TRISOL... SUBROUTINE PACKAGE 'FITPACK'.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      CHARACTER*3 TFUNCT(5)
      DATA TFUNCT/'VP ','VS ','DEN','QP ','QS '/
C
      CALL VAL1(LUN,2,NCB,5,TFUNCT)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE PARM2(ICB,COOR,UP,US,RHO,QP,QS)
      INTEGER ICB
      REAL COOR(3),UP(10),US(10),RHO,QP,QS
C
C THIS SUBROUTINE EVALUATES P AND S WAVE VELOCITIES, DENSITY, AND P AND
C S WAVE LOSS FACTORS AT A GIVEN POINT.  THE THREE FIRST AND SIX SECOND
C PARTIAL DERIVATIVES OF THE VELOCITIES ARE ALSO EVALUATED.  THE
C SPECIFIED FUNCTIONS ARE REPRESENTED AS A TENSOR PRODUCT OF SPLINES
C UNDER TENSION.  THE PARAMETERS MAY BE DEPENDENT EITHER ON THE GENERAL
C COORDINATES OR ON THE DISTRIBUTION OF ANOTHER PARAMETER, E.G.
C VS=0.577*VP OR RHO=1.7+0.2*VP, WHERE VP, VS AND RHO ARE P AND S
C VELOCITIES AND DENSITY.  THE COEFFICIENTS OF THESE FUNCTIONS ARE
C PREPARED IN SUBROUTINE PARM1, IN WHICH THE INPUT DATA CONCERNING THE
C DISTRIBUTION OF INDIVIDUAL PARAMETERS WITHIN EACH COMPLEX BLOCK ARE
C READ IN.  THE DEFAULT VALUES OF PARAMETERS NOT SPECIFIED IN THE INPUT
C DATA ARE:
C     P WAVE VELOCITY: VP=1.73205*VS,
C     S WAVE VELOCITY: VS=0.57735*VP,
```

```
C       DENSITY: RHO=1.0,
C       P WAVE LOSS FACTOR (IF THE S WAVE LOSS FACTOR IS SPECIFIED):
C                 QP=1.333333*QS*(US(1)/UP(1))**2,
C       S WAVE LOSS FACTOR (IF THE P WAVE LOSS FACTOR IS SPECIFIED):
C                 QS=0.750000*QP*(UP(1)/US(1))**2,
C       P AND S WAVE LOSS FACTORS (IF NONE OF THEM IS SPECIFIED):
C                 QP=0.0, QS=0.0.
C NOTE THAT AT LEAST ONE OF THE VELOCITIES MUST BE SPECIFIED IN THE
C INPUT DATA. P WAVE VELOCITY MUST BE POSITIVE, OTHER MATERIAL
C PARAMETERS MUST BE NON-NEGATIVE.
C
C INPUT:
C       ICB...    INDEX OF A COMPLEX BLOCK.
C       COOR...   ARRAY CONTAINING COORDINATES X1, X2, X3 OF THE GIVEN
C                 POINT.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C       UP,US...POWERS OF THE P AND S WAVE VELOCITIES (THE EXPONENT OF THE
C                 POWER IS NEXPV, SEE THE INPUT DATA FOR THE MODEL) AND
C                 THEIR FIRST AND SECOND PARTIAL DERIVATIVES IN ORDER U, U1,
C                 U2, U3, U11, U12, U22, U13, U23, U33, AT THE GIVEN POINT.
C       RHO...    DENSITY AT THE GIVEN POINT.
C       QP,QS...POWERS OF THE P AND S WAVE LOSS FACTORS (THE EXPONENT OF
C                 THE POWER IS NEXPQ, SEE THE INPUT DATA FOR THE MODEL) AT
C                 THE GIVEN POINT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
        EXTERNAL VAL2,FPOWER
C       VAL2, FPOWER... SUBROUTINE PACKAGE 'VAL'.
C       CURV2D OR CURVBD (ALTERNATIVES), SURFBD, VAL3BD, SNHCSH, DSPLNZ,
C                  INTRVL... SUBROUTINE PACKAGE 'FITPACK'.
C
C ERROR MESSAGES:
C       321...    NO VELOCITY IS DEFINED:
C                 NEITHER P NOR S WAVE VELOCITY IN THE CURRENT COMPLEX BLOCK
C                 IS DEFINED IN THE INPUT DATA.
C       322...    PROHIBITED MATERIAL PARAMETER:
C                 P WAVE VELOCITY MUST BE POSITIVE, OTHER MATERIAL
C                 PARAMETERS MUST BE NON-NEGATIVE.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C       AUXILIARY STORAGE LOCATIONS:
        INTEGER I
        REAL FAUX(10,5),POWER(5),POWER1,POWER2,POWER3,POWER4,POWER5
        EQUIVALENCE (POWER(1),POWER1),(POWER(2),POWER2),(POWER(3),POWER3)
        EQUIVALENCE (POWER(4),POWER4),(POWER(5),POWER5)
C
        CALL VAL2(2,ICB,5,COOR,FAUX,POWER)
C
C       VELOCITIES:
        IF(POWER1.NE.0.) THEN
          IF(FAUX(1,1).LE.0.) GO TO 9
          CALL FPOWER(10,FAUX(1,1),POWER1,UP)
          IF(POWER2.NE.0.) THEN
            IF(FAUX(1,2).LT.0.) GO TO 9
```

```
          CALL FPOWER(10,FAUX(1,2),POWER2,US)
        ELSE
          DO 1 I=1,10
            US(I)=0.57735*UP(I)
    1     CONTINUE
        END IF
      ELSE
        IF(POWER2.NE.0.) THEN
          IF(FAUX(1,2).LE.0.) GO TO 9
          CALL FPOWER(10,FAUX(1,2),POWER2,US)
          DO 2 I=1,10
            UP(I)=1.73205*US(I)
    2     CONTINUE
        ELSE
          STOP 'ERROR 321 IN PARM2: NO VELOCITY IS DEFINED'
        END IF
      END IF
C
C     DENSITY:
      IF(POWER3.NE.0.) THEN
        IF(FAUX(1,3).LT.0.) GO TO 9
        CALL FPOWER(1,FAUX(1,3),POWER3,RHO)
      ELSE
        RHO=1.
      END IF
C
C     LOSS FACTORS:
      IF(POWER4.NE.0.) THEN
        IF(FAUX(1,4).LT.0.) GO TO 9
        CALL FPOWER(1,FAUX(1,4),POWER4,QP)
        IF(POWER5.NE.0.) THEN
          IF(FAUX(1,5).LT.0.) GO TO 9
          CALL FPOWER(1,FAUX(1,5),POWER5,QS)
          QS=0.750000*QP*(UP(1)/US(1))**2
        ELSE
        END IF
      ELSE
        IF(POWER5.NE.0.) THEN
          IF(FAUX(1,5).LT.0.) GO TO 9
          CALL FPOWER(1,FAUX(1,5),POWER5,QS)
          QP=1.333333*QS*(US(1)/UP(1))**2
        ELSE
          QP=0.
          QS=0.
        END IF
      END IF
      RETURN
C
    9 CONTINUE
      WRITE(*,'('' X='',F9.3,'' Y='',
     *        F9.3,'' Z='',F9.3/''   VP='',F7.3,'' VS='',F7.3,'' RO='',
     *        F7.3,'' QP='',F7.3,'' QS='',F7.3)') COOR,(FAUX(1,I),I=1,5)
      STOP 'ERROR 322 IN PARM2: PROHIBITED MATERIAL PARAMETER'
      END
C
C======================================================================
C
```

```
C SUBROUTINE PACKAGE 'VAL' FOR FUNCTION SPECIFICATION AND INTERPOLATION.
C DESIGNED TO PERFORM THE INTERPOLATION OF A SET OF FUNCTIONS IN A
C RECTANGULAR GRID.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C      VALB... BLOCK DATA SUBROUTINE DEFINING COMMON BLOCK /VALC/ TO
C              STORE THE DATA DESCRIBING THE INTERPOLATED FUNCTIONS.
C      VAL1... SUBROUTINE DESIGNED TO READ THE INPUT DATA FOR THE
C              FUNCTIONS, TO COMPUTE THE COEFFICIENTS OF THE EXPANSION
C              AND TO STORE THEM IN THE MEMORY.
C      SORTV...AUXILIARY SUBROUTINE TO SUBROUTINE VAL1.
C      READV...AUXILIARY SUBROUTINE TO SUBROUTINE VAL1.
C      VAL2... SUBROUTINE EVALUATING THE FUNCTIONS INCLUDING THEIR FIRST
C              AND SECOND DERIVATIVES.  THE FUNCTIONS MAY BE USED TO
C              SPECIFY VARIOUS SURFACES IN THE MODEL, THE SPACE
C              DISTRIBUTIONS OF VARIOUS PARAMETERS, E.T.C.  THE FUNCTIONS
C              ARE REPRESENTED AS A TENSOR PRODUCT OF SPLINES UNDER
C              TENSION OF AT MOST 3 INDEPENDENT VARIABLES (I.E.  A LINEAR
C              COMBINATION OF PRODUCTS OF B-SPLINES UNDER TENSION OF AT
C              MOST 3 INDEPENDENT VARIABLES).  OUTSIDE THE SPECIFIED
C              RECTANGULAR GRID, THE FUNCTIONS ARE LINEAR ALONG STRAIGHT
C              LINES PERPENDICULAR TO THE BOUNDARY OF THE GRID.  THE
C              FUNCTIONS MAY BE EMBEDDED:  THE INDEPENDENT VARIABLE OF
C              THE FUNCTION MAY BE ANOTHER FUNCTION OF THE SAME GROUP
C              (SEE BELOW) FOREGOING IN THE INPUT DATA.
C      FPOWER..SUBROUTINE EVALUATING THE VALUE AND, POSSIBLY, THE THREE
C              FIRST AND SIX SECOND PARTIAL DERIVATIVES OF A FUNCTION, IF
C              THE VALUE AND THE THREE FIRST AND SIX SECOND PARTIAL
C              DERIVATIVES OF THE POWER-TH POWER OF THE FUNCTION  (E.G.
C              THE OUTPUT OF VAL2) ARE KNOWN.
C THIS PACKAGE EMPLOYS ROUTINES CURVN1 (OR ITS ALTERNATIVE CURVB1),
C CURV2D (OR ITS ALTERNATIVE CURVBD), SURFB1, SURFBD, VAL3B1, VAL3BD,
C VGEN, TERMS, SNHCSH, TRIDEC, TRISOL, DSPLNZ, INTRVL FROM THE
C SUBROUTINE PACKAGE 'FITPACK' BY ALAN KAYLOR CLINE, DEPARTMENT OF
C COMPUTER SCIENCES, UNIVERSITY OF TEXAS AT AUSTIN.
C
C CLASSES OF FUNCTIONS:
C     THE INTERPOLATED FUNCTIONS ARE DIVIDED INTO SOME CLASSES, E.G.
C     FUNCTIONS DESCRIBING INTERFACES, FUNCTIONS DESCRIBING THE MEDIUM
C     PARAMETERS, FUNCTIONS DESCRIBING THE PROPERTIES OF THE SOURCE,
C     ETC.  THE NUMBER MCLASS OF THE DEFINED CLASSES IS STORED IN THE
C     MEMORY AND IS INITIALLY ZERO.  THE NEW CLASS MAY BE DEFINED BY
C     MEANS OF THE INVOCATION OF SUBROUTINE VAL1, SEE ITS INPUT ARGUMENT
C     ICLASS.  DURING ONE INVOCATION OF VAL1, ONLY THE GROUPS OF
C     FUNCTIONS RELEVANT TO ONE CLASS MAY BE DEFINED.  SUBROUTINE VAL1
C     MAY BE CALLED SEVERAL TIMES EVEN FOR ONE CLASS TO DEFINE ITS
C     GROUPS SUCCESSIVELY, STAGE BY STAGE.  IN THIS CASE, THE INPUT DATA
C     FOR THE GROUPS OF FUNCTIONS RELEVANT TO ONE CLASS MAY BE READ IN
C     FROM VARIOUS FILES.
C
C GROUPS OF FUNCTIONS:
C     THE INTERPOLATED FUNCTIONS OF EACH CLASS ARE DIVIDED INTO SOME
C     GROUPS. FOR INSTANCE, THE CLASS OF FUNCTIONS DESCRIBING THE MEDIUM
C     PARAMETERS IS DIVIDED INTO GROUPS CORRESPONDING TO INDIVIDUAL
C     COMPLEX BLOCKS. THE INDIVIDUAL GROUPS NEED NOT CONTAIN THE SAME
C     NUMBER OF FUNCTIONS. THE GROUP CORRESPONDING TO A COMPLEX BLOCK
C     MAY CONTAIN E.G. THE FUNCTIONS DESCRIBING P-WAVE VELOCITY, S-WAVE
C     VELOCITY, DENSITY, ETC. THE FUNCTIONS NOT SPECIFIED BY THE INPUT
```

```
C     DATA BUT REQUIRED BY THE PROGRAM ARE DEFINED AND ARE ZERO.
C
C INPUT DATA (READ IN BY SUBROUTINE VAL1):
C     THESE INPUT DATA DEFINE THE GROUPS OF FUNCTIONS FROM THE SPECIFIED
C     CLASS.  THE INDEX ICLASS OF THE CLASS IS AN INPUT ARGUMENT OF
C     SUBROUTINE VAL1.  IF THE CLASS IS NOT DEFINED BY A PREVIOUS
C     INVOCATION OF VAL1, IT IS CREATED.  THE NUMBER NGROUP OF THE
C     GROUPS TO BE DEFINED IS AN INPUT ARGUMENT OF SUBROUTINE VAL1.  THE
C     DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).
C (1) NGROUP-TIMES (I.E. ONCE FOR EACH GROUP OF FUNCTIONS) INPUT DATA
C     (1A)+(1B):
C (1A) TEXTG,IGROUP
C     IDENTIFICATION OF THE GROUP.
C     TEXTG...ANY STRING.  ITS FIRST 3 CHARACTERS MUST DIFFER FROM 'END'
C             AND FROM ANY STRING IDENTIFYING A PHYSICAL QUANTITY,
C             DEFINED BY INPUT ARRAY TFUNCT OF SUBROUTINE VAL1 (SEE
C             BELOW).
C     IGROUP..SEQUENTIAL NUMBER OF THE GROUP IN THE CLASS.
C (1B) SEVERAL TIMES 'INPUT DATA FOR ONE FUNCTION', SEE BELOW.
C     IF INPUT ARRAY TFUNCT OF SUBROUTINE VAL1 IS FULLY FILLED BY
C             SPACES, 'INPUT DATA FOR ONE FUNCTION' MUST BE INCLUDED
C             JUST NFUNCT-TIMES (NFUNCT IS AN INPUT ARGUMENT OF
C             SUBROUTINE VAL1).  IN THIS CASE, THE INPUT FUNCTIONS ARE
C             NOT IDENTIFIED BY A STRING (SEE (1) OF 'INPUT DATA FOR ONE
C             FUNCTION), THEIR NUMBER AND ORDER MUST BE APRIORI KNOWN.
C             THIS IS, FOR INSTANCE, THE CASE OF SMOOTH SURFACES: EACH
C             GROUP CORRESPONDS TO ONE SURFACE AND CONTAINS JUST ONE
C             FUNCTION.  THE INDEX OF THE SURFACE COINCIDES WITH THE
C             INDEX OF THE GROUP, AND NO IDENTIFICATION AND SORTING OF
C             FUNCTIONS INSIDE A GROUP IS NEEDED.
C     IF INPUT ARRAY TFUNCT OF SUBROUTINE VAL1 IS NOT FULLY FILLED BY
C             SPACES, 'INPUT DATA FOR ONE FUNCTION' MAY BE INCLUDED
C             N-TIMES, WHERE 0.LE.N.LE.NFUNCT.  IN THIS CASE, THE INPUT
C             FUNCTIONS OF INDIVIDUAL GROUPS ARE IDENTIFIED BY A STRING
C             (SEE (1) OF 'INPUT DATA FOR ONE FUNCTION) IN THE INPUT
C             DATA, THEIR NUMBER AND ORDER MAY BE ARBITRARY (NOTE THAT
C             THEIR NUMBER MUST BE LESS THAN OR EQUAL TO NFUNCT).  THIS
C             IS, FOR INSTANCE, THE CASE OF COMPLEX BLOCKS: EACH GROUP
C             CORRESPONDS TO ONE COMPLEX BLOCK AND CONTAINS SOME NUMBER
C             OF FUNCTIONS DESCRIBING MATERIAL PARAMETERS.  INDIVIDUAL
C             FUNCTIONS (MATERIAL PARAMETERS) ARE IDENTIFIED BY A STRING
C             IN THE INPUT DATA.
C (2) TEXTE,AUX
C     END OF DATA.
C     TEXTE...STRING, THE FIRST 3 CHARACTERS OF WHICH MUST BE UPPER-CASE
C             'END'.
C     AUX...  ANY NUMBER OR A SLASH.
C
C INPUT DATA FOR ONE FUNCTION:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C     THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C     I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C     INTEGER.  OTHERWISE (EXCEPT TEXTF), THE INPUT PARAMETER IS OF THE
C     TYPE REAL.
C (1) TEXTF,POWER
C     PHYSICAL MEANING OF THE FUNCTION. THIS INPUT IS NOT PERFORMED IF
C     INPUT CHARACTER ARRAY TFUNCT OF SUBROUTINE VAL1 (SEE BELOW) IS
C     FULLY FILLED BY SPACES.
```

```
C      TEXTF...STRING IDENTIFYING WHICH PHYSICAL QUANTITY THE FUNCTION
C              DESCRIBES. ONLY THE FIRST 3 CHARACTERS ARE SIGNIFICANT.
C              THE FIRST 3 CHARACTERS OF THE STRING MUST NOT BE EQUAL TO
C              'END'. THE SET OF MEANINGFUL STRINGS IS DEFINED BY INPUT
C              ARRAY TFUNCT OF SUBROUTINE VAL1 (SEE BELOW).
C      POWER...THE SPECIFIED FUNCTION IS EQUAL TO THE POWER-TH POWER OF
C              THE PHYSICAL QUANTITY.
C (2)  IVAR1,IVAR2,IVAR3,SIGMA
C      THE FORM OF THE FUNCTION.
C      IVAR1,IVAR2,IVAR3... DENOTE THE FORM OF THE FUNCTION. THE FUNCTION
C              MUST BE OF THE FORM
C                F(X1,X2,X3) =W(A1,A2,A3)-B1-B2-B3   .
C              X1, X2, X3 ARE THE GENERAL COORDINATES. EACH OF A1, A2,
C              A3, B1, B2, B3 MUST BE EITHER:  (A) ONE OF GENERAL
C              COORDINATES X1, X2, X3, (B) ANOTHER PREVIOUSLY DEFINED
C              FUNCTION F(X1,X2,X3) OF THE SAME GROUP, OR (C) MUST BE
C              LEFT OUT.  AT MOST 3 OF PARAMETERS A1-B3 MAY BE OF KIND
C              (A) OR (B).  NOTE THAT IVAR1 CONTROLS THE TYPE OF A1 AND
C              B1, IVAR2 CONTROLS THE TYPE OF A2 AND B2, IVAR3 CONTROLS
C              THE TYPE OF A3 AND B3.
C              FOR IVAR1.EQ.0: A1, B1 ARE EMPTY (LEFT OUT),
C              FOR IVAR1.EQ.1: A1=X1, B1 IS EMPTY,
C              FOR IVAR1.EQ.2: A1=X2, B1 IS EMPTY,
C              FOR IVAR1.EQ.3: A1=X3, B1 IS EMPTY,
C              FOR IVAR1.GE.4: A1=F(X1,X2,X3), WHERE F(X1,X2,X3) IS
C                ANOTHER FUNCTION OF THE SAME GROUP DEFINED IN THE INPUT
C                DATA AS THE (IVAR1-3)-TH FUNCTION OF THE GROUP. B1 IS
C                EMPTY,
C              FOR IVAR1.EQ.-1: B1=X1, A1 IS EMPTY,
C              FOR IVAR1.EQ.-2: B1=X2, A1 IS EMPTY,
C              FOR IVAR1.EQ.-3: B1=X3, A1 IS EMPTY,
C              FOR IVAR1.LE.-4: B1=F(X1,X2,X3), WHERE F(X1,X2,X3) IS
C                ANOTHER FUNCTION OF THE SAME GROUP DEFINED IN THE INPUT
C                DATA AS THE (-IVAR1-3)-TH FUNCTION OF THE GROUP. A1 IS
C                EMPTY.
C              THE MEANING OF THE PARAMETERS IVAR2, IVAR3 IS SIMILAR.
C              EXAMPLES:
C              IVAR1: IVAR2: IVAR3: THE FORM OF THE FUNCTION:
C                1      2      3     F(X1,X2,X3)=W(X1,X2,X3)
C                3      1      2     F(X1,X2,X3)=W(X3,X1,X2)
C                1      2      0     F(X1,X2,X3)=W(X1,X2)
C                1      2     -3     F(X1,X2,X3)=W(X1,X2)-X3
C                1     -3      2     F(X1,X2,X3)=W(X1,X2)-X3
C                5      0      0     F(X1,X2,X3)=W(F2(X1,X2,X3)), WHERE
C              F2(X1,X2,X3) IS THE SECOND FUNCTION OF THE GROUP DEFINED
C              IN THE INPUT DATA. FUNCTION W IS INTERPOLATED BY MEANS OF
C              SPLINES UNDER TENSION.
C      SIGMA...IS THE TENSION FACTOR (ITS SIGN IS IGNORED).  THIS VALUE
C              INDICATES THE CURVINESS DESIRED.  IF ABS(SIGMA) IS NEARLY
C              ZERO (E.G. 0.001), THE RESULTING SURFACE IS APPROXIMATELY
C              THE TENSOR PRODUCT OF CUBIC SPLINES.  IF ABS(SIGMA) IS
C              LARGE (E.G. 50.), THE RESULTING SURFACE IS APPROXIMATELY
C              TRI-LINEAR.  IF SIGMA EQUALS ZERO, TENSOR PRODUCTS OF
C              CUBIC SPLINES RESULT.  A RECOMMENDED VALUE FOR SIGMA IS
C              APPROXIMATELY 1. IN ABSOLUTE VALUE.
C (3)  NX(1),...,NX(NVAR)
C      THE NUMBERS OF GRID COORDINATES FOR THE INTERPOLATION.
C      THIS INPUT IS PERFORMED IF AT LEAST ONE OF IVAR1, IVAR2, IVAR3 IS
C      POSITIVE.
C      EACH OF NX(1),...,NX(NVAR) CORRESPONDS TO ONE POSITIVE VALUE OF
```

```
C     IVAR1, IVAR2, IVAR3 AND SPECIFIES THE NUMBER OF GRID COORDINATES
C     CORRESPONDING TO THAT INDEPENDENT VARIABLE OF FUNCTION W, SEE (2).
C     THE SIGN OF NX(1),...,NX(NVAR) IS IGNORED. NVAR (.LE.3) IS THE
C     NUMBER OF POSITIVE VALUES OF THE ABOVE QUANTITIES IVAR1, IVAR2,
C     IVAR3, I.E. THE NUMBER OF INDEPENDENT VARIABLES OF FUNCTION W,
C     SEE (2).
C (4) X1(1),...,X1(NX(1))
C     THE GRID COORDINATES CORRESPONDING TO THE FIRST INDEPENDENT
C     VARIABLE OF FUNCTION W, SEE (2).
C     THIS INPUT IS PERFORMED IF NX(1) IS SPECIFIED, SEE (3), AND IS NOT
C     ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (5) X2(1),...,X2(NX(2))
C     THE GRID COORDINATES CORRESPONDING TO THE SECOND INDEPENDENT
C     VARIABLE OF FUNCTION W, SEE (2).
C     THIS INPUT IS PERFORMED IF NX(2) IS SPECIFIED, SEE (3), AND IS NOT
C     ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (6) X3(1),...,X3(NX(3))
C     THE GRID COORDINATES CORRESPONDING TO THE THIRD INDEPENDENT
C     VARIABLE OF FUNCTION W, SEE (2).
C     THIS INPUT IS PERFORMED IF NX(3) IS SPECIFIED, SEE (3), AND IS NOT
C     ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (7) (((W(I,J,K),I=1,MAX(NX(1),1)),J=1,MAX(NX(2),1)),K=1,MAX(NX(3),1))
C     THE VALUES OF FUNCTION W AT GRID POINTS. FUNCTION VALUE W(I,J,K)
C     CORRESPONDS TO POINT (X1(I),X2(J),X3(K)).
C
C STORAGE IN THE MEMORY:
C     THE PARAMETERS DESCRIBING THE INTERPOLATED FUNCTIONS ARE STORED
C     IN COMMON BLOCK /VALC/ DEFINED IN THE FOLLOWING SUBROUTINE:
C     ----------------------------------------------------------------
      BLOCK DATA VALB
      INTEGER NPAR
      PARAMETER (NPAR=8191)
      INTEGER IPAR(0:NPAR)
      REAL RPAR(0:NPAR)
      EQUIVALENCE (IPAR,RPAR)
      COMMON/VALC/IPAR
      SAVE /VALC/
      DATA IPAR(0)/0/
      END
C     ----------------------------------------------------------------
C     COMMON BLOCK /VALC/ IS INCLUDED IN SUBROUTINES VAL1 AND VAL2.
C     THE PARAMETERS ARE STORED SEQUENTIALLY IN ONE ARRAY REGARDLESS OF
C     THE FACT WHETHER THEY ARE OF TYPE INTEGER OR REAL.  THE INDIVIDUAL
C     NUMERIC STORAGE UNITS OF THE ARRAY ARE INDEXED STARTING FROM 0 AND
C     ARE NAMED IPAR(I) OR RPAR(I) DEPENDING ON THE TYPE OF A PARAMETER.
C     THE INDEX OF THE LAST ALLOCATED NUMERIC STORAGE UNIT IS NPAR (SEE
C     (F) BELOW).  IF NPAR IS CHANGED (SEE THE THIRD STATEMENT OF THE
C     ABOVE BLOCK DATA SUBROUTINE VALB), IT MUST BE ADJUSTED IN
C     SUBROUTINES VAL1 AND VAL2, TOO.
C
C     COMMON BLOCK /VALC/ CAN BE DIVIDED INTO TWO PARTS.  THE FIRST PART
C     (SEE (A)-(D) BELOW) OF THE COMMON BLOCK /VALC/ CONTAINS INTEGERS.
C     THE NUMBER OF THESE INTEGERS IN THE FIRST PART EQUALS TO
C     1+MCLASS+NG+NF, WHERE MCLASS IS THE TOTAL NUMBER OF CLASSES, NG IS
C     THE TOTAL NUMBER OF ALL GROUPS OF ALL CLASSES, NF IS THE TOTAL
C     NUMBER OF ALL FUNCTIONS OF ALL GROUPS OF ALL CLASSES. THE FIRST
C     PART OF THE COMMON BLOCK SPECIFIES THE DIVISION OF THE COMMON
C     BLOCK /VALC/ INTO PARAMETERS DESCRIBING INDIVIDUAL FUNCTIONS.
C     THE SECOND PART (SEE (E) BELOW) OF THE COMMON BLOCK /VALC/
C     CONTAINS THE PARAMETERS DESCRIBING INDIVIDUAL FUNCTIONS, STORED
```

```
C       SUCCESSSIVELY FOR THE FIRST, SECOND, ..., LAST FUNCTION OF THE
C       FIRST GROUP OF THE FIRST CLASS, FOR THE FIRST, SECOND, ..., LAST
C       FUNCTION OF THE SECOND GROUP OF THE FIRST CLASS, ..., FOR THE
C       FIRST, SECOND, ..., LAST FUNCTION OF THE FIRST GROUP OF THE SECOND
C       CLASS, ..., FOR THE FIRST, SECOND, ..., LAST FUNCTION OF THE LAST
C       GROUP OF THE LAST CLASS.  THE SECOND PART OF THE COMMON BLOCK
C       /VALC/ CONTAINS BOTH INTEGER AND REAL PARAMETERS.
C       FOR AN EXAMPLE REFER TO THE SAMPLE INPUT DATA FOR THE MODEL.
C
C       (A) NUMERIC STORAGE UNIT IPAR(0) CONTAINS THE NUMBER MCLASS OF
C       CLASSES.
C
C       (B) CLASSES:
C       SUBSEQUENT NUMERIC STORAGE UNITS (IPAR(I),I=1,MCLASS) CORRESPOND
C       TO THE INDIVIDUAL CLASSES.  ONE NUMERIC STORAGE UNIT CORRESPONDS
C       TO ONE CLASS.  THE NUMERIC STORAGE UNIT CORRESPONDING TO A CLASS
C       CONTAINS THE INDEX OF THE NUMERIC STORAGE UNIT CORRESPONDING TO
C       THE LAST GROUP OF THE CLASS.
C
C       (C) GROUPS:
C       SUBSEQUENT NUMERIC STORAGE UNITS (IPAR(I),I=MCLASS+1,MCLASS+NG)
C       CORRESPOND TO THE INDIVIDUAL GROUPS.  THE TOTAL NUMBER OF THESE
C       STORAGE UNITS IS NG.  ONE NUMERIC STORAGE UNIT CORRESPONDS TO ONE
C       GROUP.  THE NUMERIC STORAGE UNIT CORRESPONDING TO A GROUP CONTAINS
C       THE INDEX OF THE NUMERIC STORAGE UNIT CORRESPONDING TO THE LAST
C       FUNCTION OF THE GROUP.
C
C       (D) FUNCTIONS:
C       SUBSEQUENT NUMERIC STORAGE UNITS (IPAR(I),I=MCLASS+NG+1,
C       MCLASS+NG+NF) CORRESPOND TO THE INDIVIDUAL FUNCTIONS.  THE TOTAL
C       NUMBER OF THESE STORAGE UNITS IS NF.  ONE NUMERIC STORAGE UNIT
C       CORRESPONDS TO ONE FUNCTION.  THE NUMERIC STORAGE UNIT
C       CORRESPONDING TO A FUNCTION CONTAINS THE INDEX OF THE NUMERIC
C       STORAGE UNIT CORRESPONDING TO THE LAST PARAMETER THAT DESCRIBES
C       THE FUNCTION.  THE FUNCTIONS ARE STORED IN THE ORDER IN WHICH THEY
C       HAVE BEEN SPECIFIED BY THE INPUT DATA.  SINCE THE NUMBER OF
C       FUNCTIONS IN THE INPUT DATA CORRESPODING TO ONE GROUP MAY BE LESS
C       THAN THEIR MAXIMUM NUMBER NFUNCT (INPUT PARAMETER OF SUBROUTINE
C       VAL1), SOME NUMERIC STORAGE UNITS RELEVANT TO FUNCTIONS MAY NOT
C       CORRESPOND TO THE FUNCTIONS SPECIFIED BY THE INPUT DATA.  EACH
C       SUCH NUMERIC STORAGE UNIT CONTAINS THE SAME ADDRESS AS THE
C       PREVIOUS NUMERIC STORAGE UNIT (I.E.  THE CORRESPONDING FUNCTION IS
C       SPECIFIED BY NO PARAMETER) AND HAS NO INFLUENCE ON THE FUNCTION
C       EVALUATION.
C
C       (E) PARAMETERS OF FUNCTIONS:
C       SUBSEQUENT NUMERIC STORAGE UNITS (IPAR(I),I=MCLASS+NG+NF+1,
C       MCLASS+NG+NF+NP) CONTAIN THE PARAMETERS DESCRIBING INDIVIDUAL
C       FUNCTIONS.  HERE WE HAVE DENOTED BY NP THE TOTAL NUMBER OF THESE
C       STORAGE UNITS.  ANY NUMBER OF NUMERIC STORAGE UNITS MAY CORRESPOND
C       TO THE PARAMETERS OF ONE FUNCTION.  THE FIRST NUMERIC STORAGE UNIT
C       CORRESPONDING TO A FUNCTION CONTAINS THE INTEGER THAT IDENTIFIES
C       THE PHYSICAL MEANING OF THE FUNCTION, E.G.  ITS VALUES MAY
C       IDENTIFY P-WAVE VELOCITY, S-WAVE VELOCITY, DENSITY, ETC.  THE
C       SECOND NUMERIC STORAGE UNIT CONTAINS THE POWER OF THE PHYSICAL
C       QUANTITY, SEE 'INPUT DATA FOR ONE FUNCTION' (1).  IF ITEM (1) OF
C       'INPUT DATA FOR ONE FUNCTION' IS OMITTED, THE TWO FIRST NUMERIC
C       STORAGE UNITS CORRESPONDING TO THE PARAMETERS OF THE FUNCTION
C       CONTAIN 1 (THE FIRST INTEGER, THE SECOND REAL).  THE SUBSEQUENT
C       NUMERIC STORAGE UNITS CONTAIN 'INPUT DATA FOR ONE FUNCTION' (2),
```

```
C       (3), (4), (5) AND (6), STORED IN THE SAME AMOUNT AS READ FROM THE
C       INPUT DATA.  DATA (4), (5), (6) AND (7) ARE REORDERED TO RENDER
C       THE GRID COORDINATES IN ASCENDING ORDER.  INSTEAD OF THE GRID
C       VALUES (7), THE COEFFICIENTS DESCRIBING THE FUNCTION IN TERMS OF A
C       B-SPLINE UNDER TENSION BASIS ARE STORED (ONLY IF SUBROUTINE CURVN1
C       (HERMITE REPRESENTATION) IS USED, THE GRID VALUES (7) ARE STORED).
C       THESE PARAMETERS ARE FOLLOWED BY ARRAYS OF LENGTH 5*NX(1), ...,
C       5*NX(NVAR) RESPECTIVELY, CONTAINING THE B-SPLINE UNDER TENSION
C       BASIS DATA COMPUTED FOR THE PROJECTIONS OF THE GRID ONTO THE AXES
C       OF INDIVIDUAL INDEPENDENT VARIABLES (ONLY IF SUBROUTINE CURVN1
C       (HERMITE REPRESENTATION) IS USED, THE FIRST NX(1) NUMERIC STORAGE
C       UNITS CONTAIN THE SECOND DERIVATIVES AND THE FOLLOWING 4*NX(1)
C       NUMERIC STORAGE UNITS ARE UNDEFINED.
C
C       (F) UNDEFINED PART OF THE COMMON BLOCK:
C       SUBSEQUENT NUMERIC STORAGE UNITS (IPAR(I),I=MCLASS+NG+NF+NP+1,
C       NPAR) ARE UNDEFINED.
C
C       MEMORY MODEL OF THE COMMON BLOCK /VALC/:
C       ------------------------------------------------------------------
C       ADDRESS:      0    JCLASS-1 JCLASS JGROUP-1 JGROUP JFUNCT-1 JFUNCT
C       VALUE:     MCLASS LGROUP-1 MGROUP LFUNCT-1 MFUNCT  LADR-1   MADR
C       ..................................................................
C       FOR THE MEANING OF INDIVIDUAL ITEMS IN THIS TABLE SEE BELOW.
C       ------------------------------------------------------------------
C       THE WAY OF ACCESS TO IFUNCT-TH FUNCTION OF IGROUP-TH GROUP OF THE
C       ICLASS-TH CLASS:
C       ADDRESS OF LAST CLASS..................... MCLASS=IPAR(0)
C       ADDRESS OF ICLASS-TH CLASS................ JCLASS=ICLASS
C               (IT MUST BE: 1.LE.JCLASS.LE.MCLASS)
C       ADDRESS OF FIRST GROUP OF THE CLASS....... LGROUP=IPAR(JCLASS-1)+1
C       ADDRESS OF LAST GROUP OF THE CLASS........ MGROUP=IPAR(JCLASS)
C       ADDRESS OF IGROUP-TH GROUP OF THE CLASS... JGROUP=LGROUP-1+IGROUP
C               (IT MUST BE: LGROUP.LE.JGROUP.LE.MGROUP)
C       ADDRESS OF FIRST FUNCTION OF THE GROUP.... LFUNCT=IPAR(JGROUP-1)+1
C       ADDRESS OF LAST FUNCTION OF THE GROUP..... MFUNCT=IPAR(JGROUP)
C       ADDRESS OF IFUNCT-TH FUNCTION OF THE GROUP JFUNCT=LFUNCT-1+IFUNCT
C               (IT MUST BE: LFUNCT.LE.JFUNCT.LE.MFUNCT)
C       ADDRESS OF FIRST FUNCTION PARAMETER....... LADR=IPAR(JFUNCT-1)+1
C       ADDRESS OF LAST FUNCTION PARAMETER........ MADR=IPAR(JFUNCT)
C       THE PARAMETERS OF THE FUNCTION ARE STORED IN IPAR(LADR) TO
C       IPAR(MADR).
C
C DATE: 1988, JUNE 1
C CODED BY LUDEK KLIMES
C
C=====================================================================
C
      SUBROUTINE VAL1(LUN,ICLASS,NGROUP,NFUNCT,TFUNCT)
      INTEGER LUN,ICLASS,NGROUP,NFUNCT
      CHARACTER*(*) TFUNCT(NFUNCT)
C
C THIS SUBROUTINE READS THE INPUT DATA FOR A SET OF FUNCTIONS,
C DETERMINES THE PARAMETERS NECESSARY TO COMPUTE AN INTERPOLATORY
C FUNCTION ON A THREE-DIMENSIONAL RECTANGULAR GRID, AND STORES THEM IN
C THE MEMORY.  THE FUNCTION DETERMINED CAN BE REPRESENTED AS A TENSOR
C PRODUCT OF SPLINES UNDER TENSION OF AT MOST 3 INDEPENDENT VARIABLES
C (I.E.  A LINEAR COMBINATION OF PRODUCTS OF B-SPLINES UNDER TENSION OF
C AT MOST 3 INDEPENDENT VARIABLES).  OUTSIDE THE SPECIFIED RECTANGULAR
C GRID, THE FUNCTIONS ARE LINEAR ALONG STRAIGHT LINES PERPENDICULAR TO
```

```
C THE BOUNDARY OF THE GRID.  THE FUNCTIONS MAY BE EMBEDDED:  THE
C INDEPENDENT VARIABLE OF THE FUNCTION MAY BE ANOTHER FUNCTION OF THE
C SAME GROUP FOREGOING IN THE INPUT DATA.  FOR ACTUAL MAPPING OF POINTS
C IT IS NECESSARY TO CALL THE SUBROUTINE VAL2, WHICH ALSO RETURNS THE
C FIRST AND SECOND PARTIAL DERIVATIVES. SUBROUTINE VAL1 MAY BE CALLED
C SEVERAL TIMES.  THE GROUPS IN THE CLASS ARE INDEXED SUCCESIVELY,
C FOLLOWING THE GROUPS OF THE CLASS DEFINED DURING THE PREVIOUS
C INVOCATIONS.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C     ICLASS..INDEX OF THE CLASS OF THE FUNCTIONS TO BE SPECIFIED. THE
C             CLASSES ARE INDEXED BY INTEGERS STARTING FROM 1.
C     NGROUP..NUMBER OF GROUPS OF FUNCTIONS TO BE SPECIFIED DURING THE
C             CURRENT INVOCATION OF VAL1.  THE GROUPS OF EACH CLASS ARE
C             INDEXED BY INTEGERS STARTING FROM 1. IF SOME GROUPS OF
C             FUNCTIONS OF THE ICLASS-TH CLASS WERE SPECIFIED IN THE
C             PREVIOUS INVOCATION OF VAL1, THE GROUPS OF FUNCTIONS NOW
C             READ IN ARE APPENDED TO THEM AND ARE INDEXED FOLLOWING
C             THEM.
C     NFUNCT..MAXIMUM NUMBER OF FUNCTIONS TO BE SPECIFIED FOR EACH
C             GROUP. THE ACTUAL NUMBER OF SPECIFIED FUNCTIONS MAY BE
C             DIFFERENT FOR DIFFERENT GROUPS.  HOWEVER, IT MUST BE LESS
C             THAN OR EQUAL TO NFUNCT.
C     TFUNCT..STRINGS IDENTIFYING THE FUNCTIONS SPECIFIED IN THE INPUT
C             DATA. THE FUNCTION IDENTIFIED IN THE INPUT DATA BY STRING
C             TFUNCT(I) IS ASSOCIATED WITH INTEGER I. THIS INTEGER
C             IDENTIFIES WHAT THE FUNCTION DESCRIBES.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK:
      INTEGER NPAR
      PARAMETER (NPAR=8191)
      INTEGER IPAR(0:NPAR)
      REAL RPAR(0:NPAR)
      EQUIVALENCE (IPAR,RPAR)
      COMMON/VALC/IPAR
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL CURVN1
*     EXTERNAL CURVB1
      EXTERNAL SURFB1,VAL3B1,SORTV,READV
C     CURVN1 OR CURVB1 (ALTERNATIVES), SURFB1, VAL3B1, SNHCSH, VGEN,
C               TERMS, TRIDEC, TRISOL... SUBROUTINE PACKAGE 'FITPACK'.
C     SORTV, READV... THIS SUBROUTINE PACKAGE.
C
C ERROR MESSAGES:
C     351...  END OF INPUT FUNCTIONS ENCOUNTERED:
C             END OF INPUT FUNCTIONS ENCOUNTERED BEFORE ALL NGROUP
C             GROUPS OF FUNCTIONS ARE DEFINED IN THE INPUT DATA.
C     352...  IMPROPER INDEX OF THE GROUP:
C             IMPROPER INDEX OF THE GROUP OF INPUT FUNCTIONS IN THE
C             INPUT DATA.
C     353...  STRANGE ERROR:
C             THIS ERROR IN THE INPUT FUNCTIONS SHOULD NOT APPEAR.
```

```
C                 CONTACT THE AUTHORS.
C       354...    INPUT FUNCTIONS NOT PROPERLY ENDED:
C                 READ IN INPUT DATA DESCRIBING FUNCTIONS ARE NOT PROPERLY
C                 ENDED.
C       355...    INSUFFICIENT MEMORY IN /VALC/:
C                 INSUFFICIENT MEMORY FOR THE INPUT DATA IN COMMON BLOCK
C                 /VALC/.  THE DIMENSION NPAR OF ARRAY IPAR (OR RPAR) MUST
C                 BE ENLARGED.  SEE THE BLOCK DATA SUBROUTINE VALB.
C       356...    IDENTICAL GRID COORDINATES:
C                 TWO IDENTICAL GRID COORDINATES ENCOUNTERED IN THE INPUT
C                 DATA.
C
C DATE: 1989, NOVEMBER 27
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      LOGICAL WHAT
      INTEGER MCLASS,MGROUP,MFUNCT,LADR,MADR,MAXADR
      INTEGER KCLASS,KGROUP,KFUNCT,KADR,NVAR
      CHARACTER*3 TEXT
      REAL GROUP,SIGMA
      INTEGER LX(3),LX1,LX2,LX3
      EQUIVALENCE (LX(1),LX1),(LX(2),LX2),(LX(3),LX3)
      INTEGER NX(3),NX1,NX2,NX3
      EQUIVALENCE (NX(1),NX1),(NX(2),NX2),(NX(3),NX3)
      INTEGER JADR(7),JADR1,JADR2,JADR3,JADR4,JADR5,JADR6,JADR7
      EQUIVALENCE (JADR(1),JADR1),(JADR(2),JADR2),(JADR(3),JADR3)
      EQUIVALENCE (JADR(4),JADR4),(JADR(5),JADR5),(JADR(6),JADR6)
      EQUIVALENCE (JADR(7),JADR7)
      INTEGER IGROUP,IFUNCT,IERR,I,J,L,N
C
C     WHAT... FLAG IF THE PHYSICAL MEANING OF THE FUNCTIONS IS INCLUDED
C             IN THE INPUT DATA.
C     MCLASS,MGROUP,MFUNCT,LADR,MADR,MAXADR... POSITIONS IN THE MEMORY.
C     KCLASS,KGROUP,KFUNCT,KADR... SHIFTS IN THE MEMORY.
C     NVAR... NUMBER OF THE INDEPENDENT VARIABLES A1, A2, A3 OF THE
C             INTERPOLATED FUNCTION W.
C     TEXT... STRING IDENTIFYING THE CURRENT GROUP OR THE CURRENT
C             FUNCTION.
C     GROUP...INDEX OF THE CURRENT GROUP OR POWER OF THE PHYSICAL
C             QUANTITY.
C     SIGMA...TENSION FACTOR.
C     LX=(LX1,LX2,LX3)... ADDRESSES OF AUXILIARY STORAGE LOCATIONS FOR
C             REORDERING THE GRID COORDINATES.
C     NX=(NX1,NX2,NX3)... NUMBERS OF GRID LINES.
C     JADR=(JADR1,JADR2,JADR3,JADR4,JADR5,JADR6,JADR7)... ADDRESSES OF
C             PARAMETERS DESCRIBING THE INTERPOLATED FUNCTION (GRID
C             COORDINATES, B-SPLINE COEFFICIENTS, B-SPLINE BASIS
C             FUNCTIONS).
C     IGROUP,IFUNCT... DO LOOP VARIABLES.
C     IERR,I,J,L,N... LOCAL AUXILIARY VARIABLES.
C
C.......................................................................
C
C     FLAG IF THE PHYSICAL MEANING OF THE FUNCTIONS IS INCLUDED IN THE
C     INPUT DATA:
      WHAT=.FALSE.
      DO 10 I=1,NFUNCT
        IF(TFUNCT(I).NE.'   ') WHAT=.TRUE.
```

```
   10 CONTINUE
C
C     POSITIONS IN THE MEMORY:
      IF(ICLASS.GT.IPAR(0)) THEN
C       NEW CLASS:
        MCLASS=IPAR(0)
        KCLASS=ICLASS-MCLASS
      ELSE
C       OLD CLASS:
        MCLASS=ICLASS
        KCLASS=0
      END IF
      MGROUP=IPAR(MCLASS)
      KGROUP=KCLASS+NGROUP
      MFUNCT=IPAR(MGROUP)
      KFUNCT=KGROUP+NGROUP*NFUNCT
      MADR=IPAR(MFUNCT)
      KADR=NPAR-IPAR(IPAR(IPAR(IPAR(0))))
      IF(KADR.LT.KFUNCT) GO TO 99
C     UPPER BOUND OF THE AVAILABLE MEMORY
      MAXADR=MADR+KADR
C
C     MOVEMENT IN THE MEMORY:
      DO 11 I=IPAR(IPAR(IPAR(IPAR(0)))),MADR+1,-1
        IPAR(I+KADR)=IPAR(I)
   11 CONTINUE
      DO 12 I=MADR,IPAR(IPAR(IPAR(0)))+1,-1
        IPAR(I+KFUNCT)=IPAR(I)
   12 CONTINUE
      DO 13 I=IPAR(IPAR(IPAR(0))),MFUNCT+1,-1
        IPAR(I+KFUNCT)=IPAR(I)+KADR
   13 CONTINUE
      MADR=MADR+KFUNCT
      DO 14 I=MFUNCT,MGROUP+1,-1
        IPAR(I+KGROUP)=IPAR(I)+KFUNCT
   14 CONTINUE
      MFUNCT=MFUNCT+KGROUP
      DO 15 I=MGROUP,MCLASS+1,-1
        IPAR(I+KCLASS)=IPAR(I)+KGROUP
   15 CONTINUE
      MGROUP=MGROUP+KCLASS
      DO 16 I=0,MCLASS
        IPAR(I)=IPAR(I)+KCLASS
   16 CONTINUE
C     NEW CLASSES:
      DO 17 I=MCLASS+1,ICLASS
        IPAR(I)=IPAR(MCLASS)
   17 CONTINUE
C     NUMBER OF PREVIOUSLY STORED GROUPS OF FUNCTIONS OF THE CLASS
      IGROUP=IPAR(ICLASS)-IPAR(ICLASS-1)
      IPAR(ICLASS)=IPAR(ICLASS)+NGROUP
C     NEW GROUPS:
      DO 18 I=MGROUP+1,MGROUP+NGROUP
        IPAR(I)=IPAR(I-1)+NFUNCT
   18 CONTINUE
C
C     LOOP FOR GROUPS OF FUNCTIONS:
      READ(LUN,*) TEXT,GROUP
      DO 90 IGROUP=IGROUP+1,IGROUP+NGROUP
        IF(TEXT.EQ.'END') THEN
```

```
          STOP 'ERROR 351 IN VAL1: END OF INPUT FUNCTIONS ENCOUNTERED'
        ELSE IF(INT(GROUP+0.5).NE.IGROUP) THEN
          STOP 'ERROR 352 IN VAL1: IMPROPER INDEX OF THE GROUP'
        END IF
C       LOOP FOR FUNCTIONS OF THE CURRENT GROUP:
        DO 80 IFUNCT=1,NFUNCT
C         PHYSICAL MEANING OF THE FUNCTION:
          IF(WHAT) THEN
            READ(LUN,*) TEXT,GROUP
            DO 21 I=1,NFUNCT
              IF(TFUNCT(I).EQ.TEXT) THEN
                MADR=MADR+2
                IF(MADR.GT.MAXADR) GO TO 99
                IPAR(MADR-1)=I
                RPAR(MADR)=GROUP
                GO TO 22
              END IF
   21       CONTINUE
            GO TO 81
   22       CONTINUE
          ELSE
            MADR=MADR+2
            IF(MADR.GT.MAXADR) GO TO 99
            IPAR(MADR-1)=IFUNCT
            RPAR(MADR)=1.
          END IF
C
C         FORM OF THE FUNCTION:
          LADR=MADR+1
          MADR=MADR+4
          IF(MADR.GT.MAXADR) GO TO 99
          READ(LUN,*) IPAR(LADR),IPAR(LADR+1),IPAR(LADR+2),RPAR(MADR)
          SIGMA=RPAR(MADR)
C         NUMBER OF INDEPENDENT VARIABLES:
          NVAR=0
          DO 23 I=LADR,LADR+2
            IF(IPAR(I).GT.0) NVAR=NVAR+1
   23     CONTINUE
C
C         NUMBERS OF GRID COORDINATES:
          LADR=MADR+1
          MADR=MADR+NVAR
          IF(MADR.GT.MAXADR) GO TO 99
          IF(LADR.LE.MADR) THEN
            READ(LUN,*) (IPAR(I),I=LADR,MADR)
          END IF
C
C         READING GRID COORDINATES:
          L=MAXADR+1
          NVAR=0
          DO 24 J=LADR,MADR
            N=IABS(IPAR(J))
            IF(N.GT.0) THEN
              LADR=MADR+1
              MADR=MADR+N
              IF(N.EQ.1) THEN
                IF(MADR.GE.L-1) GO TO 99
                READ(LUN,*) RPAR(LADR)
              ELSE
                L=L-N
```

```
                IF(MADR+N.GE.L-1) GO TO 99
                NVAR=NVAR+1
                NX(NVAR)=N
                LX(NVAR)=L
                JADR(NVAR)=LADR
                READ(LUN,*) (RPAR(I),I=MADR+1,MADR+N)
                CALL SORTV(N,RPAR(MADR+1),RPAR(LADR),IPAR(L))
              END IF
            END IF
   24     CONTINUE
          DO 25 I=NVAR+1,3
            NX(I)=1
            LX(I)=L-1
            IPAR(L-1)=1
   25     CONTINUE
C
C         READING GRID VALUES:
          JADR4=MADR+1
          MADR=MADR+NX1*NX2*NX3
          IF(MADR.GE.L) GO TO 99
          CALL READV(LUN,NX1,NX2,NX3,IPAR(LX1),IPAR(LX2),IPAR(LX3),
     *                                                       RPAR(JADR4))
C
C         COMPUTING B-SPLINE UNDER TENSION EXPANSION COEFFICIENTS:
          IF(NVAR.LE.0) THEN
C           NO INDEPENDENT VARIABLE:
            CONTINUE
          ELSE
C           SIZE OF THE TEMPORARY STORAGE LOCATION
            N=3*MAX0(NX1,NX2,NX3)
            JADR5=MADR+1
            MADR=MADR+5*NX1
            IF(MADR+N.GT.MAXADR) GO TO 99
            IF(NVAR.EQ.1) THEN
C             ONE INDEPENDENT VARIABLE:
C             TWO ALTERNATIVES: HERMITE OR B-SPLINE REPRESENTATIONS
C             MAY BE USED FOR THE 1-D INTERPOLATION. JUST ONE OF THE
C             FOLLOWING TWO STATEMENTS MUST BE SUPPLIED BY '*' IN THE
C             FIRST COLUMN:
C             FIRST STATAMENT - HERMITE REPRESENTATION:
                CALL CURVN1(NX1,RPAR(JADR1),RPAR(JADR4),
     *                        RPAR(JADR5),RPAR(MADR+1),SIGMA,IERR)
C             SECOND STATAMENT - B-SPLINE REPRESENTATION:
*               CALL CURVB1(NX1,RPAR(JADR1),RPAR(JADR4),RPAR(JADR4),
*    *                        RPAR(JADR5),RPAR(MADR+1),SIGMA,IERR)
C             DO NOT FORGET TO SUPPLY '*' INTO THE FIRST COLUMN OF THE
C             CORRESPONDING STATEMENT IN SUBROUTINE VAL2.
            ELSE
              JADR6=MADR+1
              MADR=MADR+5*NX2
              IF(MADR+N.GT.MAXADR) GO TO 99
              IF(NVAR.EQ.2) THEN
C               TWO INDEPENDENT VARIABLES:
                CALL SURFB1(NX1,NX2,RPAR(JADR1),RPAR(JADR2),
     *                  RPAR(JADR4),NX1,RPAR(JADR4),
     *                  RPAR(JADR5),RPAR(JADR6),RPAR(MADR+1),SIGMA,IERR)
              ELSE
C               THREE INDEPENDENT VARIABLES:
                JADR7=MADR+1
                MADR=MADR+5*NX3
```

```
                  IF(MADR+N.GT.MAXADR) GO TO 99
                  CALL VAL3B1(NX1,NX2,NX3,RPAR(JADR1),RPAR(JADR2),
     *                     RPAR(JADR3),RPAR(JADR4),NX1,NX2,RPAR(JADR4),
     *                     RPAR(JADR5),RPAR(JADR6),RPAR(JADR7),
     *                     RPAR(MADR+1),SIGMA,IERR)
                END IF
              END IF
            END IF
            IF(IERR.NE.0) THEN
              STOP 'ERROR 353 IN VAL1: STRANGE ERROR'
            END IF
C           COEFFICIENTS ARE EVALUATED
C
            MFUNCT=MFUNCT+1
            IPAR(MFUNCT)=MADR
   80     CONTINUE
          READ(LUN,*) TEXT,GROUP
C         END OF LOOP FOR FUNCTIONS
C
C         THE REMAINING FUNCTIONS OF THE CURRENT GROUP ARE NOT DEFINED BY
C         THE INPUT DATA:
   81     CONTINUE
          DO 82 I=IFUNCT,NFUNCT
            MFUNCT=MFUNCT+1
            IPAR(MFUNCT)=MADR
   82     CONTINUE
   90 CONTINUE
C     END OF LOOP FOR GROUPS OF FUNCTIONS
C
      IF(TEXT.NE.'END') THEN
        STOP 'ERROR 354 IN VAL1: INPUT FUNCTIONS NOT PROPERLY ENDED'
      END IF
C
C     MOVEMENT IN THE MEMORY:
      KADR=MAXADR-MADR
      DO 91 I=MAXADR+1,NPAR
        IPAR(I-KADR)=IPAR(I)
   91 CONTINUE
      RETURN
C
   99 CONTINUE
      STOP 'ERROR 355 IN VAL1: INSUFFICIENT MEMORY IN /VALC/'
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE SORTV(NX,X1,X2,IX)
      INTEGER NX,IX(NX)
      REAL X1(NX),X2(NX)
C
C THIS SUBROUTINE IS AN AUXILIARY ROUTINE TO VAL1. IT REORDERS THE
C INPUT GRID COORDINATES TO BE ASCENDING.
C
C     AUXILIARY STORAGE LOCATIONS
      INTEGER I,J
C
      DO 3 J=1,NX
        IX(J)=1
        DO 1 I=1,J-1
          IF(X1(J).EQ.X1(I)) GO TO 9
```

```
            IF(X1(J).GT.X1(I)) IX(J)=IX(J)+1
    1     CONTINUE
          DO 2 I=J+1,NX
            IF(X1(J).EQ.X1(I)) GO TO 9
            IF(X1(J).GT.X1(I)) IX(J)=IX(J)+1
    2     CONTINUE
    3   CONTINUE
        DO 4 J=1,NX
          X2(IX(J))=X1(J)
    4   CONTINUE
        RETURN
C
    9   CONTINUE
        STOP 'ERROR 356 IN SORTV IN VAL1: IDENTICAL GRID COORDINATES'
        END
C
C-----------------------------------------------------------------------
C
        SUBROUTINE READV(LUN,NX1,NX2,NX3,IX1,IX2,IX3,VAL)
        INTEGER LUN,NX1,NX2,NX3,IX1(NX1),IX2(NX2),IX3(NX3)
        REAL VAL(NX1,NX2,NX3)
C
C THIS SUBROUTINE IS AN AUXILIARY ROUTINE TO VAL1. IT READS FROM THE
C INPUT DATA THE VALUES GIVEN AT GRID POINTS.
C
C       AUXILIARY STORAGE LOCATIONS
        INTEGER I1,I2,I3
C
        READ(LUN,*) (((VAL(IX1(I1),IX2(I2),IX3(I3)),I1=1,NX1),
       *                                            I2=1,NX2),I3=1,NX3)
        RETURN
        END
C
C=======================================================================
C
        SUBROUTINE VAL2(ICLASS,IGROUP,NFUNCT,COOR,F,POWER)
        INTEGER ICLASS,IGROUP,NFUNCT
        REAL COOR(3),F(10,NFUNCT),POWER(NFUNCT)
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE THREE FIRST PARTIAL
C DERIVATIVES AND THE SIX SECOND PARTIAL DERIVATIVES OF A GIVEN FUNCTION
C AT A GIVEN POINT.
C
C INPUT:
C       ICLASS..INDEX OF THE CLASS OF THE REQUIRED FUNCTIONS. THE CLASSES
C               ARE INDEXED BY INTEGERS STARTING FROM 1.
C       IGROUP..INDEX OF THE GROUP OF THE REQUIRED FUNCTIONS. THE GROUPS
C               OF EACH CLASS ARE INDEXED BY INTEGERS STARTING FROM 1.
C       NFUNCT..NUMBER OF THE REQUIRED FUNCTIONS. ALL FUNCTIONS BELONGING
C               TO THE IGROUP-TH GROUP OF THE ICLASS-TH CLASS AND DEFINED
C               BY THE INPUT DATA MUST BE REQUIRED. THE FUNCTIONS DEFINED
C               BY THE INPUT DATA (SEE SUBROUTINE VAL1) ARE ONE-TO-ONE
C               CORRESPONDING TO THE INTEGERS WHICH IDENTIFY WHAT THE
C               FUNCTION DESCRIBES. THE POSITION OF EACH EVALUATED
C               FUNCTION IN THE OUTPUT ARRAY F (SEE BELOW) IS DETERMINED
C               BY THIS INTEGER. THAT IS WHY NFUNCT MUST BE GEATER THAN OR
C               EQUAL TO THE GREATEST OF THESE INTEGERS. THE REQUIRED
C               FUNCTIONS NOT DEFINED BY THE INPUT DATA ARE DEFINED ON THE
C               OUTPUT OF THIS SUBROUTINE AND ARE ZERO.
C       COOR... ARRAY CONTAINING COORDINATES X1, X2, X3 OF THE GIVEN POINT
```

```
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C     F...     ARRAY CONTAINING, IN EACH ITS COLUMN, FUNCTION VALUE, THE
C              FIRST AND SECOND PARTIAL DERIVATIVES OF THE CORRESPONDING
C              EVALUATED FUNCTION IN THE ORDER F, F1, F2, F3, F11, F12,
C              F22, F13, F23, F33.
C     POWER...THE SPECIFIED FUNCTION IS EQUAL TO THE POWER-TH POWER OF
C              THE CORRESPONDING PHYSICAL QUANTITY.  THE ZERO VALUE OF
C              THE POWER INDICATES THAT THE CORRESPONDING FUNCTION IS NOT
C              DEFINED BY THE INPUT DATA.
C
C COMMON BLOCK:
      INTEGER NPAR
      PARAMETER (NPAR=8191)
      INTEGER IPAR(0:NPAR)
      REAL RPAR(0:NPAR)
      EQUIVALENCE (IPAR,RPAR)
      COMMON/VALC/IPAR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL CURV2D
*     EXTERNAL CURVBD
      EXTERNAL SURFBD,VAL3BD
C     CURV2D OR CURVBD (ALTERNATIVES), SURFBD, VAL3BD, SNHCSH, DSPLNZ,
C              INTRVL... SUBROUTINE PACKAGE 'FITPACK'.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     THE EVALUATED FUNCTION HAS THE FORM OF
C     F(X1,X2,X3) = W(A1,A2,A3) - B1 - B2 - B3     ,
C
C     ITS FIRST DERIVATIVES ARE
C     DF     DW     DAK    DB1    DB2    DB3
C     --- =  --- *  ---  - ---  - ---  - ---    ,
C     DXI    DAK    DXI    DXI    DXI    DXI
C
C     ITS SECOND DERIVATIVES ARE
C      D2 F       D W  D2 AK      D2 W    DAK DAJ     D2 B1              D2 B3
C     ------- = ---*------- + -------*---*--- - ------- - ... - -------.
C     DXI DXM    DAK DXI DXM   DAK DAJ DXI DXM   DXI DXM            DXI DXM
C
C.......................................................................
C
      INTEGER JGROUP,LFUNCT,MFUNCT,JFUNCT,LADR,MADR,IADR,IVAL
      INTEGER NVAR,IVAR(3),JVAR,KVAR
      INTEGER NX(3),NX1,NX2,NX3
      EQUIVALENCE (NX(1),NX1),(NX(2),NX2),(NX(3),NX3)
      REAL XX(3),XX1,XX2,XX3,R1,R2,R3
      EQUIVALENCE (XX(1),XX1),(XX(2),XX2),(XX(3),XX3)
      INTEGER JADR(7),JADR1,JADR2,JADR3,JADR4,JADR5,JADR6,JADR7
      EQUIVALENCE (JADR(1),JADR1),(JADR(2),JADR2),(JADR(3),JADR3)
      EQUIVALENCE (JADR(4),JADR4),(JADR(5),JADR5),(JADR(6),JADR6)
      EQUIVALENCE (JADR(7),JADR7)
      REAL SIGMA,W(10),AUX1,AUX2
      INTEGER I,J,K,M,N,ISYM(3,3)
```

```
      DATA ISYM/5,6,8,6,7,9,8,9,10/
C
C     JGROUP..ADDRESS OF THE IGROUP-TH GROUP OF THE ICLASS-TH CLASS.
C     LFUNCT,MFUNCT,JFUNCT... ADDRESSES OF THE FIRST, LAST AND ARBITRARY
C             FUNCTIONS OF THE GROUP.
C     LADR,MADR,IADR... ADDRESSES OF THE FIRST, LAST AND ARBITRARY
C             PARAMETERS OF THE CURRENT FUNCTION.
C     IVAL... INDEX OF THE FUNCTION F BEING CURRENTLY EVALUATED.
C     NVAR,IVAR(3),JVAR,KVAR... NUMBER AND TYPES OF THE INDEPENDENT
C             VARIABLES A1, A2, A3 OF THE INTERPOLATED FUNCTION W.
C     NX=(NX1,NX2,NX3)... NUMBERS OF GRID LINES.
C     XX=(XX1,XX2,XX3),R1,R2,R3... VALUES OF INDEPENDENT VARIABLES A1,
C             A2, A3 OF FUNCTION W.
C     JADR=(JADR1,JADR2,JADR3,JADR4,JADR5,JADR6,JADR7)... ADDRESSES OF
C             PARAMETERS DESCRIBING THE INTERPOLATED FUNCTION (GRID
C             COORDINATES, B-SPLINE COEFFICIENTS, B-SPLINE BASIS
C             FUNCTIONS).
C     SIGMA...TENSION FACTOR.
C     W...    ARRAY FOR THE VALUE, THE FIRST AND SECOND PARTIAL
C             DERIVATIVES OF FUNCTION W.
C     AUX1,AUX2,I,J,K,M,N... LOCAL AUXILIARY VARIABLES.
C     ISYM... STORAGE OF THE SYMMETRIC 3*3 MATRIX.
C
C.......................................................................
C
C     THE DEFAULT VALUE OF THE FUNCTION IS THE ZERO FUNCTION.
C     LOOP FOR THE FUNCTIONS TO BE EVALUATED:
      DO 12 J=1,NFUNCT
        DO 11 I=1,10
          F(I,J)=0.
   11   CONTINUE
      POWER(J)=0.
   12 CONTINUE
C
      JGROUP=IPAR(ICLASS-1)+IGROUP
      LFUNCT=IPAR(JGROUP-1)+1
      MFUNCT=IPAR(JGROUP)
      MADR  =IPAR(LFUNCT-1)
C
C     LOOP FOR FUNCTIONS F BEING EVALUATED:
      DO 90 JFUNCT=LFUNCT,MFUNCT
C       STARTING AND END ADDRESSES OF THE PARAMETERS DESCRIBING THE
C       FUNCTION
        LADR=MADR+1
        MADR=IPAR(JFUNCT)
        IF(LADR.LE.MADR) THEN
C         INDEX OF FUNCTION F BEING CURRENTLY EVALUATED
          IVAL=IPAR(LADR)
C         POWER OF THE CORRESPONDING PHYSICAL QUANTITY
          POWER(IVAL)=RPAR(LADR+1)
C         TENSION FACTOR
          SIGMA=RPAR(LADR+5)
C
C         THE NUMBER, TYPES AND VALUES OF THE INDEPENDENT VARIABLES AI
C         OF FUNCTION W BEING INTERPOLATED, AND THE FUNCTIONS BI BEING
C         SUBTRACTED FROM THE EVALUATED FUNCTION:
C         INITIAL ADDRESS
          IADR=LADR+6
C         INITIAL NUMBER OF THE INDEPENDENT VARIABLES
          NVAR=0
```

```
      JADR1=0
      JADR2=0
      JADR3=0
      JADR4=0
C     LOOP FOR THE POSSIBLE INDEPENDENT VARIABLES:
      DO 20 M=LADR+2,LADR+4
C       TYPE OF THE POSSIBLE INDEPENDENT VARIABLE:
        J=IPAR(M)
        IF(J.NE.0) THEN
          IF(J.GT.0) THEN
            N=IABS(IPAR(IADR))
            IF(N.GE.2) THEN
              NVAR=NVAR+1
              NX(NVAR)=N
              IF(J.LE.3) THEN
                IVAR(NVAR)=J
                XX(NVAR)=COOR(J)
              ELSE
                K=IPAR(IPAR(LFUNCT+J-5)+1)
                IVAR(NVAR)=K+3
                XX(NVAR)=F(1,K)
              END IF
            ELSE IF(N.EQ.1) THEN
              JADR(NVAR+1)=JADR(NVAR+1)+1
            END IF
            IADR=IADR+1
          ELSE
C           SUBTRACTING CERTAIN FUNCTIONS FROM FUNCTION F BEING
C           EVALUATED:
            IF(J.GE.-3) THEN
C             SUBTRACTING A COORDINATE:
              F(1,IVAL)=F(1,IVAL)-COOR(-J)
              F(1-J,IVAL)=F(1-J,IVAL)-1.
            ELSE
C             SUBTRACTING ANOTHER FUNCTION F:
              K=IPAR(IPAR(LFUNCT-J-5))
              DO 19 I=1,10
                F(I,IVAL)=F(I,IVAL)-F(I,K)
   19         CONTINUE
            END IF
          END IF
        END IF
   20 CONTINUE
C
C     INTERPOLATION OF FUNCTION W:
      JADR1=IADR+JADR1
      IF(NVAR.LE.0) THEN
C       NO INDEPENDENT VARIABLE:
        W(1)=RPAR(JADR1)
      ELSE
        JADR2=JADR1+NX1+JADR2
        IF(XX1.LT.RPAR(JADR1)) THEN
          R1=RPAR(JADR1)
        ELSE IF(XX1.GT.RPAR(JADR2-1)) THEN
          R1=RPAR(JADR2-1)
        ELSE
          R1=XX1
        END IF
        IF(NVAR.EQ.1) THEN
C         ONE INDEPENDENT VARIABLE:
```

```
                 JADR3=JADR2+NX1
C                TWO ALTERNATIVES: HERMITE OR B-SPLINE REPRESENTATIONS
C                MAY BE USED FOR THE 1-D INTERPOLATION. JUST ONE OF THE
C                FOLLOWING TWO STATEMENTS MUST BE SUPPLIED BY '*' IN THE
C                FIRST COLUMN:
C                FIRST STATAMENT - HERMITE REPRESENTATION:
                   CALL CURV2D(R1,W(1),W(2),W(5),NX1,
     *                         RPAR(JADR1),RPAR(JADR2),RPAR(JADR3),SIGMA)
C                SECOND STATAMENT - B-SPLINE REPRESENTATION:
*                  CALL CURVBD(R1,W(1),W(2),W(5),NX1,
*    *                         RPAR(JADR1),RPAR(JADR2),RPAR(JADR3),SIGMA)
C                DO NOT FORGET TO SUPPLY '*' INTO THE FIRST COLUMN OF THE
C                CORRESPONDING STATEMENT IN SUBROUTINE VAL1.
               ELSE
                 JADR3=JADR2+NX2+JADR3
                 IF(XX2.LT.RPAR(JADR2)) THEN
                   R2=RPAR(JADR2)
                 ELSE IF(XX2.GT.RPAR(JADR3-1)) THEN
                   R2=RPAR(JADR3-1)
                 ELSE
                   R2=XX2
                 END IF
                 IF(NVAR.EQ.2) THEN
C                  TWO INDEPENDENT VARIABLES:
                   JADR4=JADR3+NX1*NX2
                   JADR5=JADR4+5*NX1
                   CALL SURFBD(R1,R2,W(1),W(2),W(3),W(5),W(6),W(7),NX1,NX2,
     *                     RPAR(JADR1),RPAR(JADR2),RPAR(JADR3),RPAR(JADR4),
     *                     RPAR(JADR5),SIGMA)
                 ELSE
C                  THREE INDEPENDENT VARIABLES:
                   JADR4=JADR3+NX3+JADR4
                   JADR5=JADR4+NX1*NX2*NX3
                   JADR6=JADR5+5*NX1
                   JADR7=JADR6+5*NX2
                   IF(XX3.LT.RPAR(JADR3)) THEN
                     R3=RPAR(JADR3)
                   ELSE IF(XX3.GT.RPAR(JADR4-1)) THEN
                     R3=RPAR(JADR4-1)
                   ELSE
                     R3=XX3
                   END IF
                   CALL VAL3BD(R1,R2,R3,W(1),W(2),W(3),W(4),W(5),W(6),W(7),
     *                     W(9),W(10),W(8),NX1,NX2,NX3,
     *                     RPAR(JADR1),RPAR(JADR2),RPAR(JADR3),RPAR(JADR4),
     *                     RPAR(JADR5),RPAR(JADR6),RPAR(JADR7),SIGMA)
                   W(1)=W(1)+W(4)*(XX3-R3)
                 END IF
                 W(1)=W(1)+W(3)*(XX2-R2)
               END IF
               W(1)=W(1)+W(2)*(XX1-R1)
             END IF
C            FUNCTION W IS EVALUATED
C
C            EVALUATION OF FUNCTION F:
C            FUNCTIONAL VALUE (ZERO DERIVATIVE)
             F(1,IVAL)=F(1,IVAL)+W(1)
C            LOOP FOR THE SUMMATION INDEX K:
             DO 39 K=1,NVAR
               KVAR=IVAR(K)
```

```
              IF(KVAR.LE.3) THEN
C               FIRST DERIVATIVES - FIRST TERM ON R.H.S.
                F(1+KVAR,IVAL)=F(1+KVAR,IVAL)+W(1+K)
C               SECOND DERIVATIVES - SECOND TERM ON R.H.S. (THE FIRST TERM
C               VANISHES IN THIS CASE) - LOOP FOR THE SUMMATION INDEX J:
                DO 32 J=1,NVAR
                  JVAR=IVAR(J)
                  IF(JVAR.LE.3) THEN
                    IF(JVAR.LE.KVAR) THEN
                      N=ISYM(JVAR,KVAR)
                      F(N,IVAL)=F(N,IVAL)+W(ISYM(J,K))
                    END IF
                  ELSE
                    JVAR=JVAR-3
                    AUX1=W(ISYM(J,K))
                    DO 31 I=1,JVAR
                      N=ISYM(I,JVAR)
                      F(N,IVAL)=F(N,IVAL)+AUX1*F(1+I,JVAR)
   31               CONTINUE
                  END IF
   32           CONTINUE
              ELSE
                KVAR=KVAR-3
C               FIRST AND SECOND DERIVATIVES - FIRST TERMS ON R.H.S.
                DO 33 I=2,10
                  F(I,IVAL)=F(I,IVAL)+W(1+K)*F(I,KVAR)
   33           CONTINUE
C               SECOND DERIVATIVES - SECOND TERM ON R.H.S. -
C               LOOP FOR THE SUMMATION INDEX J:
                DO 38 J=1,NVAR
                  JVAR=IVAR(J)
                  IF(JVAR.LE.3) THEN
                    AUX1=W(ISYM(J,K))
                    DO 34 I=1,KVAR
                      N=ISYM(I,KVAR)
                      F(N,IVAL)=F(N,IVAL)+AUX1*F(1+I,KVAR)
   34               CONTINUE
                  ELSE
                    JVAR=JVAR-3
                    AUX1=W(ISYM(J,K))
                    DO 36 M=1,3
                      AUX2=AUX1*F(1+M,JVAR)
                      DO 35 I=1,M
                        N=ISYM(I,M)
                        F(N,IVAL)=F(N,IVAL)+AUX2*F(1+I,KVAR)
   35                 CONTINUE
   36               CONTINUE
                  END IF
   38           CONTINUE
              END IF
   39       CONTINUE
C
          END IF
   90 CONTINUE
C     END OF LOOP FOR EVALUATED FUNCTIONS F
C
      RETURN
      END
C
C=====================================================================
```

```
C
      SUBROUTINE FPOWER(N,FINP,POWER,FOUT)
      INTEGER N
      REAL FINP(N),POWER,FOUT(N)
C
C THIS SUBROUTINE EVALUATES THE VALUE AND, POSSIBLY, THE THREE FIRST AND
C SIX SECOND PARTIAL DERIVATIVES OF A FUNCTION IF THE VALUE AND THE
C THREE FIRST AND SIX SECOND PARTIAL DERIVATIVES OF THE POWER-TH POWER
C OF THE FUNCTION (E.G. THE OUTPUT OF VAL2) ARE KNOWN.
C
C INPUT:
C     N...     FOR N=1: ONLY THE FUNCTION VALUE IS EVALUATED. THE
C                DERIVATIVES ARE IGNORED.
C              FOR N=10: THE VALUE AND THE THREE FIRST AND SIX SECOND
C                PARTIAL DERIVATIVES ARE EVALUATED.
C     FINP...  ARRAY CONTAINING THE VALUE, THE FIRST AND SECOND PARTIAL
C              DERIVATIVES OF THE POWER-TH POWER OF THE FUNCTION TO BE
C              EVALUATED, IN THE ORDER F, F1, F2, F3, F11, F12, F22, F13,
C              F23, F33.  FOR N=1, ONLY THE FUNCTION VALUE IS REQUIRED.
C     POWER...THE SPECIFIED FUNCTION IS EQUAL TO THE POWER-TH POWER OF
C              THE CORRESPONDING PHYSICAL QUANTITY.
C NONE OF THE INPUT PARAMETERS ARE ALTERED (EXCEPT FINP IF THIS
C PARAMETER AND FOUT ARE IDENTICAL IN THE CALLING SEQUENCE).
C
C OUTPUT:
C     FOUT...  ARRAY CONTAINING THE VALUE, THE FIRST AND SECOND PARTIAL
C              DERIVATIVES OF THE EVALUATED FUNCTION, IN THE ORDER F, F1,
C              F2, F3, F11, F12, F22, F13, F23, F33.  THIS PARAMETER MAY
C              COINCIDE WITH FINP, IN WHICH CASE FINP IS DESTROYED ON
C              OUTPUT.  NOTE THAT THIS COINCIDENCE IS AN EXCEPTION TO
C              ANSI STANDARD OF FORTRAN 77.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1988, MAY 26
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I
      REAL F,AUX1,AUX2
C
      IF(0.999.LT.POWER.AND.POWER.LT.1.001) THEN
        DO 1 I=1,N
          FOUT(I)=FINP(I)
    1   CONTINUE
      ELSE
        IF(-1.001.LT.POWER.AND.POWER.LT.-0.999) THEN
          F=1./FINP(1)
        ELSE
          F=FINP(1)**(1./POWER)
        END IF
        FOUT(1)=F
        IF(N.GT.1) THEN
          AUX1= F/(FINP(1)*POWER)
          AUX2= (POWER-1.)/F
          FOUT(2)=AUX1*FINP(2)
          FOUT(3)=AUX1*FINP(3)
          FOUT(4)=AUX1*FINP(4)
```

```
          FOUT(5)=AUX1*FINP(5)-AUX2*FOUT(2)*FOUT(2)
          FOUT(6)=AUX1*FINP(6)-AUX2*FOUT(2)*FOUT(3)
          FOUT(7)=AUX1*FINP(7)-AUX2*FOUT(3)*FOUT(3)
          FOUT(8)=AUX1*FINP(8)-AUX2*FOUT(2)*FOUT(4)
          FOUT(9)=AUX1*FINP(9)-AUX2*FOUT(3)*FOUT(4)
          FOUT(10)=AUX1*FINP(10)-AUX2*FOUT(4)*FOUT(4)
        END IF
      END IF
      RETURN
      END
C
C======================================================================
C
```

```
C SUBROUTINES OF THE SOFTWARE PACKAGE 'FITPACK' BY A.K. CLINE
C USED IN THE COMPLETE RAY TRACING ALGORITHM
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING PARTS:
C     (0) AUXILIARY SUBROUTINE
C              SNHCSH
C         COMMON TO ALL THE FOLLOWING PARTS.
C     (1) THE SUBROUTINES PREPARING THE PARAMETERS NECESSARY TO COMPUTE
C         AN INTERPOLATORY FUNCTION:
C              CURVN1 (HERMITE REPRESENTATION OF 1-D FUNCTION),
C              CURVB1 (B-SPLINE REPRESENTATION OF 1-D FUNCTION),
C              SURFB1 (B-SPLINE REPRESENTATION OF 2-D FUNCTION),
C              VAL3B1 (B-SPLINE REPRESENTATION OF 3-D FUNCTION),
C              VGEN (AUXILIARY SUBROUTINE),
C              TERMS (AUXILIARY SUBROUTINE),
C              TRIDEC (AUXILIARY SUBROUTINE),
C              TRISOL (AUXILIARY SUBROUTINE).
C         SUBROUTINES CURVN1 AND CURVB1 ARE ALTERNATIVES.
C     (2) THE SUBROUTINES EVALUATING THE VALUE, FIRST AND SECOND PARTIAL
C         DERIVATIVES OF THE INTERPOLATORY FUNCTION AT A GIVEN POINT:
C              CURV2D (HERMITE REPRESENTATION OF 1-D FUNCTION),
C              CURVBD (B-SPLINE REPRESENTATION OF 1-D FUNCTION),
C              SURFBD (B-SPLINE REPRESENTATION OF 2-D FUNCTION),
C              VAL3BD (B-SPLINE REPRESENTATION OF 3-D FUNCTION),
C              DSPLNZ (AUXILIARY SUBROUTINE),
C              INTRVL (AUXILIARY EXTERNAL FUNCTION).
C         SUBROUTINES CURV2D AND CURVBD ARE ALTERNATIVES.
C     (3) THE SUBROUTINES EVALUATING THE VALUE, FIRST AND SECOND PARTIAL
C         DERIVATIVES, AND THE VARIATIONS OF THE FUNCTION VALUE WITH
C         RESPECT TO THE B-SPLINE EXPANSION COEFFICIENTS, AT A GIVEN
C         POINT. THIS PART IS AN ALTERNATIVE TO PART (2) AND CONTAINS
C         SUBROUTINES
C              CURVBV (B-SPLINE REPRESENTATION OF 1-D FUNCTION),
C              SURFBV (B-SPLINE REPRESENTATION OF 2-D FUNCTION),
C              VAL3BV (B-SPLINE REPRESENTATION OF 3-D FUNCTION).
C         THESE SUBROUTINES ARE THE MODIFICATIONS OF CURVBD, SURFBD,
C         AND VAL3BD DESIGNED FOR THE SOLUTION OF INVERSE PROBLEMS. THEY
C         REFER PROCEDURES DSPLNZ AND INTRVL FROM PART (2). THIS PART
C         CORRESPONDS TO THE ALTERNATIVES WITH CURVB1 AND CURVBD OF
C         PARTS (1) AND (2), RESPECTIVELY.
C
C TAKEN FROM:
C     FITPACK - A SOFTWARE PACKAGE FOR CURVE AND SURFACE FITTING
C     EMPLOYING SPLINES UNDER TENSION
C     BY ALAN KAYLOR CLINE, DEPARTMENT OF COMPUTER SCIENCES,
C     THE UNIVERSITY OF TEXAS AT AUSTIN, AUGUST 31, 1981.
C
C=======================================================================
C
C PART 0:
C
C=======================================================================
C
      SUBROUTINE SNHCSH (SINHM,COSHM,X,ISW)
C
      INTEGER ISW
      REAL SINHM,COSHM,X
C
C                              FROM FITPACK -- AUGUST 31, 1981
C                          CODED BY A. K. CLINE AND R. J. RENKA
```

```
C                                   DEPARTMENT OF COMPUTER SCIENCES
C                                     UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE RETURNS APPROXIMATIONS TO
C       SINHM(X) = SINH(X)-X
C       COSHM(X) = COSH(X)-1
C AND
C       COSHMM(X) = COSH(X)-1-X*X/2
C WITH RELATIVE ERROR LESS THAN 6.16E-6
C
C ON INPUT--
C
C   X CONTAINS THE VALUE OF THE INDEPENDENT VARIABLE.
C
C   ISW INDICATES THE FUNCTION DESIRED
C           = -1 IF ONLY SINHM IS DESIRED,
C           =  0 IF BOTH SINHM AND COSHM ARE DESIRED,
C           =  1 IF ONLY COSHM IS DESIRED,
C           =  2 IF ONLY COSHMM IS DESIRED,
C           =  3 IF BOTH SINHM AND COSHMM ARE DESIRED.
C
C ON OUTPUT--
C
C   SINHM CONTAINS THE VALUE OF SINHM(X) IF ISW .LE. 0 OR
C   ISW .EQ. 3 (SINHM IS UNALTERED IF ISW .EQ.1 OR ISW .EQ.
C   2).
C
C   COSHM CONTAINS THE VALUE OF COSHM(X) IF ISW .EQ. 0 OR
C   ISW .EQ. 1 AND CONTAINS THE VALUE OF COSHMM(X) IF ISW
C   .GE. 2 (COSHM IS UNALTERED IF ISW .EQ. -1).
C
C AND
C
C   X AND ISW ARE UNALTERED.
C
C-----------------------------------------------------------------
C
      DATA SP2/5.04850926418006E-04/,
     *     SP1/3.62841692246321E-02/,
     *     SQ1/-1.37157937097122E-02/
      DATA CP2/1.31625490355985E-03/,
     *     CP1/6.57866547762733E-02/,
     *     CQ1/-1.75465241841312E-02/
      DATA ZP2/1.40048186158693E-04/,
     *     ZP1/1.67309141907440E-02/,
     *     ZQ2/9.82154460147143E-05/,
     *     ZQ1/-1.66024148976133E-02/
      XX = X
      AX = ABS(XX)
      XS = XX*XX
      IF ((AX .GE. 2.20) .OR. (AX .GE. 1.17 .AND.
     *     ISW .NE. 2)) EXPX = EXP(AX)
C
C SINHM APPROXIMATION
C
      IF (ISW .EQ. 1 .OR. ISW .EQ. 2) GO TO 2
      IF (AX .GE. 1.17) GO TO 1
      SINHM = (((SP2*XS+SP1)*XS+1.)*XS*XX)/((SQ1*XS+1.)*6.)
      GO TO 2
    1 SINHM = (EXPX-1./EXPX)/2.-AX
```

```
      IF (XX .LT. 0.) SINHM = -SINHM
C
C COSHM APPROXIMATION
C
    2 IF (ISW .NE. 0 .AND. ISW .NE. 1) GO TO 4
      IF (AX .GE. 1.17) GO TO 3
      COSHM = (((CP2*XS+CP1)*XS+1.)*XS)/((CQ1*XS+1.)*2.)
      GO TO 4
    3 COSHM = (EXPX+1./EXPX)/2.-1.
C
C COSHMM APPROXIMATION
C
    4 IF (ISW .LE. 1) RETURN
      IF (AX .GE. 2.20) GO TO 5
      COSHM = (((ZP2*XS+ZP1)*XS+1.)*XS*XS)/(((ZQ2*XS+ZQ1)*XS
     *        +1.)*24.)
      RETURN
    5 COSHM = (EXPX+1./EXPX)/2.-1.-XS/2.
      RETURN
      END
C
C=======================================================================
C
C PART 1:
C
C=======================================================================
C
      SUBROUTINE CURVN1 (N,X,Y,YP,TEMP,SIGMA,IERR)
C
      INTEGER N,IERR
      REAL X(N),Y(N),YP(N),TEMP(N),SIGMA
C
C                                  FROM FITPACK -- AUGUST 31, 1981
C                         CODED BY A. K. CLINE AND S. E. GALINSKY
C                                  DEPARTMENT OF COMPUTER SCIENCES
C                                    UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE A NATURAL INTERPOLATORY SPLINE UNDER TENSION
C THROUGH A SEQUENCE OF FUNCTIONAL VALUES. FOR ACTUAL
C COMPUTATION OF POINTS ON THE CURVE IT IS NECESSARY TO CALL
C THE FUNCTION CURV2.
C
C ON INPUT--
C
C   N IS THE NUMBER OF VALUES TO BE INTERPOLATED (N.GE.2).
C
C   X IS AN ARRAY OF THE N INCREASING ABSCISSAE OF THE
C   FUNCTIONAL VALUES.
C
C   Y IS AN ARRAY OF THE N ORDINATES OF THE VALUES, (I. E.
C   Y(K) IS THE FUNCTIONAL VALUE CORRESPONDING TO X(K) ).
C
C   YP IS AN ARRAY OF LENGTH AT LEAST N.
C
C   TEMP IS AN ARRAY OF LENGTH AT LEAST N WHICH IS USED FOR
C   SCRATCH STORAGE.
C
C AND
C
```

```
C    SIGMA CONTAINS THE TENSION FACTOR. THIS VALUE INDICATES
C    THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO
C    (E.G. .001) THE RESULTING CURVE IS APPROXIMATELY A
C    CUBIC SPLINE. IF ABS(SIGMA) IS LARGE (E.G. 50.) THE
C    RESULTING CURVE IS NEARLY A POLYGONAL LINE. IF SIGMA
C    EQUALS ZERO A CUBIC SPLINE RESULTS.  A STANDARD VALUE
C    FOR SIGMA IS APPROXIMATELY 1. IN ABSOLUTE VALUE.
C
C ON OUTPUT--
C
C    YP CONTAINS THE VALUES OF THE SECOND DERIVATIVE OF THE
C    CURVE AT THE GIVEN NODES.
C
C    IERR CONTAINS AN ERROR FLAG,
C         = 0 FOR NORMAL RETURN,
C         = 1 IF N IS LESS THAN 2,
C         = 2 IF X-VALUES ARE NOT STRICTLY INCREASING.
C
C AND
C
C    N, X, Y, AND SIGMA ARE UNALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES SNHCSH.
C
C-----------------------------------------------------------
C
      NM1 = N-1
      NP1 = N+1
      IERR = 0
      IF (N .LE. 1) GO TO 4
      IF (X(N) .LE. X(1)) GO TO 5
C
C DENORMALIZE TENSION FACTOR
C
      SIGMAP = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
C
C SET UP RIGHT HAND SIDE AND TRIDIAGONAL SYSTEM FOR YP AND
C PERFORM FORWARD ELIMINATION
C
      DELX1 = X(2)-X(1)
      IF (DELX1 .LE. 0.) GO TO 5
      DX1 = (Y(2)-Y(1))/DELX1
      CALL TERMS (DIAG1,SDIAG1,SIGMAP,DELX1)
      SDIAG1 = 0.
      YP(1) = 0.
      TEMP(1) = 0.
      IF (N .EQ. 2) GO TO 2
      DO 1 I = 2,NM1
        DELX2 = X(I+1)-X(I)
        IF (DELX2 .LE. 0.) GO TO 5
        DX2 = (Y(I+1)-Y(I))/DELX2
        CALL TERMS (DIAG2,SDIAG2,SIGMAP,DELX2)
        DIAG = DIAG1+DIAG2-SDIAG1*TEMP(I-1)
        YP(I) = (DX2-DX1-SDIAG1*YP(I-1))/DIAG
        TEMP(I) = SDIAG2/DIAG
        DX1 = DX2
        DIAG1 = DIAG2
    1   SDIAG1 = SDIAG2
    2 YP(N) = 0.
      TEMP(N-1) = 0.
```

```
C
C PERFORM BACK SUBSTITUTION
C
      DO 3 I = 2,N
        IBAK = NP1-I
    3   YP(IBAK) = YP(IBAK)-TEMP(IBAK)*YP(IBAK+1)
      RETURN
C
C TOO FEW POINTS
C
    4 IERR = 1
      RETURN
C
C X-VALUES NOT STRICTLY INCREASING
C
    5 IERR = 2
      RETURN
      END
C
C======================================================================
C
      SUBROUTINE CURVB1 (NX,X,W,C,VX,TEMP,SIGMA,IERR)
C
      INTEGER NX,IERR
      REAL X(NX),W(NX),C(NX),VX(5,NX),TEMP(1),SIGMA
C
C                                            COMPLEMENT TO FITPACK
C                                            BY ALAN KAYLOR CLINE
C                                        CODED -- OCTOBER 9, 1986
C                                                  BY LUDEK KLIMES
C                                            INST. GEOL. GEOTECHN.
C                                    CZECHOSL. ACAD. SCI., PRAGUE
C
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPOLATORY FUNCTION ON A ONE DIMENSIONAL
C GRID. THE FUNCTION DETERMINED CAN BE
C REPRESENTED BY SPLINES UNDER TENSION.  FOR ACTUAL
C MAPPING OF POINTS IT IS NECESSARY TO CALL THE SUBROUTINE
C CURVBD, WHICH ALSO RETURNS FIRST AND SECOND DERIVATIVES.
C
C ON INPUT--
C
C   NX IS THE NUMBER OF GRID POINTS.
C   (NX SHOULD BE AT LEAST 2)
C
C   X IS ARRAY OF THE NX COORDINATES OF THE GRID POINTS.
C   THESE SHOULD BE STRICTLY INCREASING.
C
C   W IS AN ARRAY OF THE NX FUNCTIONAL VALUES AT THE
C   THE GRID POINTS, I. E. W(I,J) CONTAINS THE FUNCTIONAL
C   VALUE AT X(I) FOR I = 1,...,NX .
C
C   C IS AN ARRAY OF AT LEAST NX LOCATIONS. THIS
C   PARAMETER MAY COINCIDE WITH W IN WHICH CASE W IS
C   DESTROYED ON OUTPUT.
C
C   VX IS THE ARRAY OF AT LEAST 5 * NX  LOCATIONS.
C
C   TEMP IS AN ARRAY OF AT LEAST 3 * NX LOCATIONS
C   WHICH IS USED FOR SCRATCH STORAGE.
```

```
C
C    SIGMA CONTAINS THE TENSION FACTOR. THIS VALUE INDICATE
C    THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO
C    (E. G. .001) THE RESULTING SURFACE IS APPROXIMATELY THE
C    TENSOR PRODUCT OF CUBIC SPLINES. IF ABS(SIGMA) IS LARGE
C    (E. G. 50.) THE RESULTING SURFACE IS APPROXIMATELY
C    BI-LINEAR. IF SIGMA EQUALS ZERO TENSOR PRODUCTS OF CUBIC
C    SPLINES RESULT. A STANDARD VALUE FOR SIGMA IS
C    APPROXIMATELY 1. IN ABSOLUTE VALUE.
C
C ON OUTPUT--
C
C    C CONTAINS THE COEFFICIENTS OF A REPRESENTATION OF THE
C    INTERPOLATED FUNCTION IN A B-SPLINE FORM.
C
C    VX CONTAINS B-SPLINE UNDER TENSION BASIS DATA.
C
C    IERR CONTAINS AN ERROR FLAG.
C         = 0 FOR NORMAL RETURN,
C         = 1 IF NX IS LESS THAN 2,
C         = 2 IF THE X-ARRAY IS NOT STRICTLY
C             INCREASING.
C
C AND
C
C    NONE OF THE INPUT PARAMETERS ARE ALTERED (EXCEPT W IF
C    THIS PARAMETER AND C ARE IDENTICAL IN THE CALLING
C    SEQUENCE).
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES VGEN, TERMS,
C SNHCSH, TRIDEC, AND TRISOL.
C
C-----------------------------------------------------------
C
C COPY W INTO C
C
      DO 1 I = 1,NX
    1   C(I) = W(I)
C
C GENERATE BASIS FUNCTIONS ALONG X-GRID
C SET UP TRIDIAGONAL SYSTEM AND SOLVE
C
      CALL VGEN (NX,X,SIGMA,VX,IERR)
      IF (IERR .NE. 0) RETURN
      DO 2 I = 2,NX
    2   TEMP(I) = VX(5,I-1)
      NXPI = NX
      DO 3 I = 1,NX
        NXPI = NXPI+1
    3   TEMP(NXPI) = 1.
      DO 4 I = 2,NX
        NXPI = NXPI+1
    4   TEMP(NXPI) = VX(4,I)
      CALL TRIDEC (NX,TEMP(1),TEMP(NX+1),TEMP(2*NX+1),
     *             TEMP(1),TEMP(NX+1),IERR)
      CALL TRISOL (NX,TEMP(1),TEMP(NX+1),TEMP(2*NX+1),C,NX,
     *             1,1)
      RETURN
      END
C
```

```
C=====================================================================
C
      SUBROUTINE SURFB1 (NX,NY,X,Y,W,NW1,C,VX,VY,TEMP,SIGMA,
     *                   IERR)
C
      INTEGER NX,NY,NW1,IERR
      REAL X(NX),Y(NY),W(NW1,NY),C(NX,NY),VX(5,NX),VY(5,NY),
     *     TEMP(1),SIGMA
C
C                                  FROM FITPACK -- AUGUST 31, 1981
C                                       CODED BY ALAN KAYLOR CLINE
C                                  DEPARTMENT OF COMPUTER SCIENCES
C                                    UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPOLATORY FUNCTION ON A TWO DIMENSIONAL
C RECTANGULAR GRID. THE FUNCTION DETERMINED CAN BE
C REPRESENTED AS A TENSOR PRODUCT OF SPLINES UNDER TENSION
C FOR ACTUAL MAPPING OF POINTS IT IS NECESSARY TO CALL THE
C SUBROUTINE SURFBD, WHICH ALSO RETURNS FIRST AND SECOND
C PARTIAL DERIVATIVES.
C
C ON INPUT--
C
C   NX AND NY ARE THE NUMBER OF GRID LINES IN THE X- AND Y
C   DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR GRID. (NX
C   AND NY SHOULD BE AT LEAST 2.)
C
C   X AND Y ARE ARRAYS OF THE NX AND NY COORDINATES OF THE
C   GRID LINES IN X- AND Y-DIRECTIONS, RESPECTIVELY. THESE
C   SHOULD BE STRICTLY INCREASING.
C
C   W IS AN ARRAY OF THE NX * NY FUNCTIONAL VALUES AT THE
C   THE GRID POINTS, I. E. W(I,J) CONTAINS THE FUNCTIONAL
C   VALUE AT (X(I),Y(J)) FOR I = 1,...,NX, AND J = 1,...,NY.
C
C   NW1 IS THE FIRST DIMENSION OF THE ARRAY W USED IN THE
C   CALLING PROGRAM (NW1 .GE. NX).
C
C   C IS AN ARRAY OF AT LEAST NX * NY LOCATIONS. THIS
C   PARAMETER MAY COINCIDE WITH W IN WHICH CASE W IS
C   DESTROYED ON OUTPUT.
C
C   VX AND VY ARE ARRAYS OF AT LEAST 5 * NX AND 5 * NY
C   LOCATIONS, RESPECTIVELY.
C
C   TEMP IS AN ARRAY OF AT LEAST 3 * MAX(NX, NY) LOCATIONS
C   WHICH IS USED FOR SCRATCH STORAGE.
C
C AND
C
C   SIGMA CONTAINS THE TENSION FACTOR. THIS VALUE INDICATE
C   THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO
C   (E. G. .001) THE RESULTING SURFACE IS APPROXIMATELY THE
C   TENSOR PRODUCT OF CUBIC SPLINES. IF ABS(SIGMA) IS LARGE
C   (E. G. 50.) THE RESULTING SURFACE IS APPROXIMATELY
C   BI-LINEAR. IF SIGMA EQUALS ZERO TENSOR PRODUCTS OF CUBIC
C   SPLINES RESULT. A STANDARD VALUE FOR SIGMA IS
C   APPROXIMATELY 1. IN ABSOLUTE VALUE.
C
```

```
C ON OUTPUT--
C
C   C CONTAINS THE COEFFICIENTS OF A REPRESENTATION OF THE
C   INTERPOLATED FUNCTION IN A B-SPLINE TENSOR PRODUCTION
C   FORM.
C
C   VX AND VY CONTAIN B-SPLINE UNDER TENSION BASIS DATA.
C
C   IERR CONTAINS AN ERROR FLAG.
C        = 0 FOR NORMAL RETURN,
C        = 1 IF NX OR NY IS LESS THAN 2,
C        = 2 IF THE X- OR Y-ARRAYS ARE NOT STRICTLY
C            INCREASING.
C
C AND
C
C   NONE OF THE INPUT PARAMETERS ARE ALTERED (EXCEPT W IF
C   THIS PARAMETER AND C ARE IDENTICAL IN THE CALLING
C   SEQUENCE).
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES VGEN, TERMS,
C SNHCSH, TRIDEC, AND TRISOL.
C
C-------------------------------------------------------------- -
C
C COPY W INTO C
C
      DO 1 J = 1,NY
        DO 1 I = 1,NX
    1     C(I,J) = W(I,J)
C
C GENERATE BASIS FUNCTIONS ALONG X-GRID
C SET UP TRIDIAGONAL SYSTEM AND SOLVE
C
      CALL VGEN (NX,X,SIGMA,VX,IERR)
      IF (IERR .NE. 0) RETURN
      DO 2 I = 2,NX
    2   TEMP(I) = VX(5,I-1)
      NXPI = NX
      DO 3 I = 1,NX
        NXPI = NXPI+1
    3   TEMP(NXPI) = 1.
      DO 4 I = 2,NX
        NXPI = NXPI+1
    4   TEMP(NXPI) = VX(4,I)
      CALL TRIDEC (NX,TEMP(1),TEMP(NX+1),TEMP(2*NX+1),
     *             TEMP(1),TEMP(NX+1),IERR)
      CALL TRISOL (NX,TEMP(1),TEMP(NX+1),TEMP(2*NX+1),C,NX,
     *             NY,1)
C
C GENERATE BASIS FUNCTIONS ALONG Y-GRID
C SET UP TRIDIAGONAL SYSTEM AND SOLVE
C
      CALL VGEN (NY,Y,SIGMA,VY,IERR)
      IF (IERR .NE. 0) RETURN
      DO 5 J = 2,NY
    5   TEMP(J) = VY(5,J-1)
      NYPJ = NY
      DO 6 J = 1,NY
        NYPJ = NYPJ+1
```

```
    6   TEMP(NYPJ) = 1.
      DO 7 J = 2,NY
        NYPJ = NYPJ+1
    7   TEMP(NYPJ) = VY(4,J)
      CALL TRIDEC (NY,TEMP(1),TEMP(NY+1),TEMP(2*NY+1),
     *             TEMP(1),TEMP(NY+1),IERR)
      CALL TRISOL (NY,TEMP(1),TEMP(NY+1),TEMP(2*NY+1),C,1,
     *            NX,NX)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE VAL3B1 (NX,NY,NZ,X,Y,Z,W,NW1,NW2,C,VX,VY,
     *                   VZ,TEMP,SIGMA,IERR)
C
      INTEGER NX,NY,NZ,NW1,NW2,IERR
      REAL X(NX),Y(NY),Z(NZ),W(NW1,NW2,NZ),C(NX,NY,NZ),
     *     VX(5,NX),VY(5,NY),VZ(5,NZ),TEMP(1),SIGMA
C
C                               FROM FITPACK -- AUGUST 31, 1981
C                                    CODED BY ALAN KAYLOR CLINE
C                               DEPARTMENT OF COMPUTER SCIENCES
C                                 UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPOLATORY FUNCTION ON A THREE DIMENSIONAL
C RECTANGULAR GRID. THE FUNCTION DETERMINED CAN BE
C REPRESENTED AS A TENSOR PRODUCT OF SPLINES UNDER TENSION.
C FOR ACTUAL MAPPING OF POINTS IT IS NECESSARY TO CALL THE
C SUBROUTINE VAL3BD, WHICH ALSO RETURNS FIRST AND SECOND
C PARTIAL DERIVATIVES.
C
C ON INPUT--
C
C   NX, NY, AND NZ ARE THE NUMBER OF GRID LINES IN THE X-,
C   Y-, AND Z-DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR
C   GRID. (NX, NY, AND NZ SHOULD BE AT LEAST 2.)
C
C   X, Y, AND Z ARE ARRAYS OF THE NX, NY, AND NZ COORDINATES
C   OF THE GRID LINES IN THE X-, Y-, AND Z-DIRECTIONS,
C   RESPECTIVELY. THESE SHOULD BE STRICTLY INCREASING.
C
C   W IS AN ARRAY OF THE NX * NY * NZ FUNCTIONAL VALUES AT
C   THE GRID POINTS, I. E. W(I,J,K) CONTAINS THE FUNCTIONAL
C   VALUE AT (X(I),Y(J),Z(K)) FOR I = 1,...,NX,
C   J = 1,...,NY, AND K = 1,...,NZ.
C
C   NW1 AND NW2 ARE THE FIRST TWO DIMENSIONS OF THE ARRAY W
C   USED IN THE CALLING PROGRAM (NW1 .GE. NX AND NW2 .GE.
C   NY).
C
C   C IS AN ARRAY OF AT LEAST NX * NY * NZ LOCATIONS. THIS
C   PARAMETER MAY COINCIDE WITH W IN WHICH CASE W IS
C   DESTROYED ON OUTPUT.
C
C   VX, VY, AND VZ ARE ARRAYS OF AT LEAST 5 * NX, 5 * NY,
C   AND 5 * NZ LOCATIONS, RESPECTIVELY.
C
C   TEMP IS AN ARRAY OF AT LEAST 3 * MAX(NX, NY, NZ)
```

```
C    LOCATIONS WHICH IS USED FOR SCRATCH STORAGE.
C
C AND
C
C    SIGMA CONTAINS THE TENSION FACTOR. THIS VALUE INDICATES
C    THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO
C    (E. G. .001) THE RESULTING SURFACE IS APPROXIMATELY THE
C    TENSOR PRODUCT OF CUBIC SPLINES. IF ABS(SIGMA) IS LARGE
C    (E. G. 50.) THE RESULTING SURFACE IS APPROXIMATELY
C    TRI-LINEAR. IF SIGMA EQUALS ZERO TENSOR PRODUCTS OF
C    CUBIC SPLINES RESULT. A STANDARD VALUE FOR SIGMA IS
C    APPROXIMATELY 1. IN ABSOLUTE VALUE.
C
C ON OUTPUT--
C
C    C CONTAINS THE COEFFICIENTS OF A REPRESENTATION OF THE
C    INTERPOLATED FUNCTION IN A B-SPLINE TENSOR PRODUCTION
C    FORM.
C
C    VX, VY, AND VZ CONTAIN B-SPLINE UNDER TENSION BASIS
C    DATA.
C
C    IERR CONTAINS AN ERROR FLAG.
C         = 0 FOR NORMAL RETURN,
C         = 1 IF NX, NY, OR NZ IS LESS THAN 2,
C         = 2 IF THE X-, Y-, OR Z-ARRAYS ARE NOT STRICTLY
C             INCREASING.
C
C AND
C
C    NONE OF THE INPUT PARAMETERS ARE ALTERED (EXCEPT W IF
C    THIS PARAMETER AND C ARE IDENTICAL IN THE CALLING
C    SEQUENCE).
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES VGEN, TERMS,
C SNHCSH, TRIDEC, AND TRISOL.
C
C-----------------------------------------------------------
C
C COPY W INTO C
C
      DO 1 K = 1,NZ
        DO 1 J = 1,NY
          DO 1 I = 1,NX
    1       C(I,J,K) = W(I,J,K)
C
C GENERATE BASIS FUNCTIONS ALONG X-GRID
C SET UP TRIDIAGONAL SYSTEM AND SOLVE
C
      CALL VGEN (NX,X,SIGMA,VX,IERR)
      IF (IERR .NE. 0) RETURN
      DO 2 I = 2,NX
    2   TEMP(I) = VX(5,I-1)
      NXPI = NX
      DO 3 I = 1,NX
        NXPI = NXPI+1
    3   TEMP(NXPI) = 1.
      DO 4 I = 2,NX
        NXPI = NXPI+1
    4   TEMP(NXPI) = VX(4,I)
```

```
      CALL TRIDEC (NX,TEMP(1),TEMP(NX+1),TEMP(2*NX+1),
     *             TEMP(1),TEMP(NX+1),IERR)
      CALL TRISOL (NX,TEMP(1),TEMP(NX+1),TEMP(2*NX+1),C,NX,
     *            NY*NZ,1)
C
C GENERATE BASIS FUNCTIONS ALONG Y-GRID
C SET UP TRIDIAGONAL SYSTEM AND SOLVE
C
      CALL VGEN (NY,Y,SIGMA,VY,IERR)
      IF (IERR .NE. 0) RETURN
      DO 5 J = 2,NY
    5   TEMP(J) = VY(5,J-1)
      NYPJ = NY
      DO 6 J = 1,NY
        NYPJ = NYPJ+1
    6   TEMP(NYPJ) = 1.
      DO 7 J = 2,NY
        NYPJ = NYPJ+1
    7   TEMP(NYPJ) = VY(4,J)
      CALL TRIDEC (NY,TEMP(1),TEMP(NY+1),TEMP(2*NY+1),
     *             TEMP(1),TEMP(NY+1),IERR)
      DO 8 K = 1,NZ
    8   CALL TRISOL (NY,TEMP(1),TEMP(NY+1),TEMP(2*NY+1),C(1,1,K),
     *               1,NX,NX)
C
C GENERATE BASIS FUNCTIONS ALONG Z-GRID
C SET UP TRIDIAGONAL SYSTEM AND SOLVE
C
      CALL VGEN (NZ,Z,SIGMA,VZ,IERR)
      IF (IERR .NE. 0) RETURN
      DO 9 K = 2,NZ
    9   TEMP(K) = VZ(5,K-1)
      NZPK = NZ
      DO 10 K = 1,NZ
        NZPK = NZPK+1
   10   TEMP(NZPK) = 1.
      DO 11 K = 2,NZ
        NZPK = NZPK+1
   11   TEMP(NZPK) = VZ(4,K)
      CALL TRIDEC (NZ,TEMP(1),TEMP(NZ+1),TEMP(2*NZ+1),
     *             TEMP(1),TEMP(NZ+1),IERR)
      CALL TRISOL (NZ,TEMP(1),TEMP(NZ+1),TEMP(2*NZ+1),C,1,
     *            NX*NY,NX*NY)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE VGEN (N,X,SIGMA,V,IERR)
C
      INTEGER N,IERR
      REAL X(N),SIGMA,V(5,N)
C
C                                  FROM FITPACK -- AUGUST 31, 1981
C                                       CODED BY ALAN KAYLOR CLINE
C                                  DEPARTMENT OF COMPUTER SCIENCES
C                                    UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE GENERATES AN ARRAY OF COEFFICIENTS USED BY
C OTHER SUBROUTINES FOR THE DETERMINATION OF A B-SPLINE
```

```
C UNDER TENSION BASIS.
C
C ON INPUT--
C
C   N IS THE NUMBER OF KNOTS DEFINING THE BASIS (N .GE. 2).
C
C   X IS THE ARRAY OF THE N INCREASING KNOTS. ANY LINEAR
C   COMBINATION OF THE RESULTING BASIS WILL HAVE THIRD
C   DERIVATIVE DISCONTINUITIES ONLY AT THE INTERIOR KNOTS,
C   (I. E. X(2),...,X(N-1) ).
C
C   SIGMA CONTAINS THE TENSION FACTOR. THIS VALUE INDICATES
C   THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO
C   (E. G. .001) THE BASIS FUNCTIONS ARE APPROXIMATELY CUBIC
C   SPLINES. IF ABS(SIGMA) IS LARGE (E. G. 50.) THE BASIS
C   FUNCTIONS ARE NEARLY PIECEWISE LINEAR. IF SIGMA EQUALS
C   ZERO A CUBIC SPLINE BASIS RESULTS. A STANDARD VALUE FOR
C   SIGMA IS APPROXIMATELY 1. IN ABSOLUTE VALUE.
C
C AND
C
C   V IS AN ARRAY OF AT LEAST 5*N LOCATIONS.
C
C ON OUTPUT--
C
C   V CONTAINS CERTAIN COEFFICIENTS TO BE USED BY OTHER
C   SUBPROGRAMS FOR THE DETERMINATION OF THE B-SPLINE UNDER
C   TENSION BASIS. CONSIDERED AS A 5 BY N ARRAY, FOR I = 1,
C   ... , N, B-SPLINE BASIS FUNCTION I IS SPECIFIED BY--
C     V(1,I) = SECOND DERIVATIVE AT X(I-1), FOR I .NE. 1,
C     V(2,I) = SECOND DERIVATIVE AT X(I),   FOR ALL I,
C     V(3,I) = SECOND DERIVATIVE AT X(I+1), FOR I .NE. N,
C     V(4,I) = FUNCTION VALUE AT X(I-1),    FOR I .NE. 1,
C     V(5,I) = FUNCTION VALUE AT X(I+1),    FOR I .NE. N,
C   AND THE PROPERTIES THAT IT HAS--
C     1. FUNCTION VALUE 1 AT X(I),
C     2. FUNCTION VALUE AND SECOND DERIVATIVE = 0 AT
C          X(1), ... , X(I-2), AND X(I+2), ... , X(N).
C   IN V(5,N) AND V(3,N) ARE CONTAINED FUNCTION VALUE AND
C   SECOND DERIVATIVE OF BASIS FUNCTION ZERO AT X(1),
C   RESPECTIVELY. IN V(4,1) AND V(1,1) ARE CONTAINED
C   FUNCTION VALUE AND SECOND DERIVATIVE OF BASIS FUNCTION
C   N+1 AT X(N), RESPECTIVELY. FUCTION VALUE AND SECOND
C   DERIVATIVE OF THESE TWO BASIS FUNCTIONS ARE ZERO AT ALL
C   OTHER KNOTS. ONLY BASIS FUNCTION ZERO HAS NON-ZERO
C   SECOND DERIVATIVE VALUE AT X(1) AND ONLY BASIS
C   FUNCTION N+1 HAS NON-ZERO SECOND DERIVATIVE AT X(N).
C
C   IERR CONTAINS AN ERROR FLAG,
C        = 0 FOR NORMAL RETURN,
C        = 1 IF N IS LESS THAN 2,
C        = 2 IF X-VALUES ARE NOT STRICTLY INCREASING.
C
C AND
C
C   N, X, AND SIGMA ARE UNALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES TERMS AND
C SNHCSH.
C
```

```
C-----------------------------------------------------------------
C
      NM1 = N-1
      IERR = 0
      IF (N .LE. 1) GO TO 3
      IF (X(N) .LE. X(1)) GO TO 4
C
C DENORMALIZE TENSION FACTOR
C
      SIGMAP = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
C
C GENERATE COEFFICIENTS FOR LEFT END BASIS FUNCTIONS
C
      D3 = X(2)-X(1)
      IF (D3 .LE. 0.) GO TO 4
      CALL TERMS (DIAG3,SDIAG3,SIGMAP,D3)
      D4 = D3
      IF (N .GE. 3) D4 = X(3)-X(2)
      IF (D4 .LE. 0.) GO TO 4
      CALL TERMS (DIAG4,SDIAG4,SIGMAP,D4)
      A22 = DIAG3+SDIAG3
      A23 = DIAG3+DIAG4+SDIAG3+SDIAG4
      V(2,1) = 0.
      V(3,1) = 1./(D3*(DIAG3+DIAG4)+(D3+D4)*SDIAG4)
      V(5,1) = SDIAG4*D4*V(3,1)
      IF (N .EQ. 2) GO TO 2
      A22 = 2.*A22
      D1 = D3
      D2 = D3
      D3 = D4
      DIAG1 = DIAG3
      DIAG2 = DIAG3
      DIAG3 = DIAG4
      SDIAG1 = SDIAG3
      SDIAG2 = SDIAG3
      SDIAG3 = SDIAG4
C
C GENERATE COEFFICIENTS FOR INTERIOR BASIS FUNCTIONS
C
      DO 1 I = 2,NM1
        IF (I .NE. NM1) D4 = X(I+2)-X(I+1)
        IF (D4 .LE. 0.) GO TO 4
        IF (D4 .NE. D3) CALL TERMS (DIAG4,SDIAG4,SIGMAP,D4)
        A11 = DIAG1+DIAG2+SDIAG1*(1.+D1/D2)
        A12 = SDIAG2/A11
        B1 = 1./(D2*A11)
        A33 = DIAG3+DIAG4+SDIAG4*(1.+D4/D3)
        A32 = SDIAG3/A33
        B3 = 1./(D3*A33)
        A21 = A22
        A22 = A23
        A23 = DIAG3+DIAG4+SDIAG3+SDIAG4
        V(2,I) = -(A21*B1+A23*B3)/(A22-A21*A12-A23*A32)
        V(1,I) = B1-A12*V(2,I)
        V(3,I) = B3-A32*V(2,I)
        V(4,I) = SDIAG1*D1*V(1,I)
        V(5,I) = SDIAG4*D4*V(3,I)
C
C SAVE CONSTANTS FOR NEXT ITERATION
C
```

```
          D1 = D2
          D2 = D3
          D3 = D4
          DIAG1 = DIAG2
          DIAG2 = DIAG3
          DIAG3 = DIAG4
          SDIAG1 = SDIAG2
          SDIAG2 = SDIAG3
    1     SDIAG3 = SDIAG4
C
C GENERATE COEFFICIENTS FOR RIGHT END BASIS FUNCTIONS
C
      V(2,N) = 0.
      V(1,N) = 1./(D2*(DIAG1+DIAG2)+(D2+D1)*SDIAG1)
      V(4,N) = SDIAG1*D1*V(1,N)
      V(3,N) = V(1,3)
      V(5,N) = V(4,3)
      V(1,1) = V(3,N-2)
      V(4,1) = V(5,N-2)
C
C ADJUST BASIS FOR NATURAL END CONDITIONS
C
      V(4,2) = V(4,2)-V(1,2)*V(5,N)/V(3,N)
      V(1,2) = 0.
      V(5,NM1) = V(5,NM1)-V(3,NM1)*V(4,1)/V(1,1)
      V(3,NM1) = 0.
      RETURN
C
C N EQUAL TO 2
C
    2 V(4,1) = V(5,1)
      V(1,1) = V(3,1)
      V(3,1) = 0.
      V(5,1) = 0.
      V(1,2) = 0.
      V(2,2) = 0.
      V(3,2) = V(1,1)
      V(4,2) = 0.
      V(5,2) = V(4,1)
      RETURN
C
C TOO FEW KNOTS
C
    3 IERR = 1
      RETURN
C
C X-VALUES NOT STRICTLY INCREASING
C
    4 IERR = 2
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE TERMS (DIAG,SDIAG,SIGMA,DEL)
C
      REAL DIAG,SDIAG,SIGMA,DEL
C
C                                   FROM FITPACK -- AUGUST 31, 1981
C                              CODED BY A. K. CLINE AND R. J. RENKA
```

```
C                                      DEPARTMENT OF COMPUTER SCIENCES
C                                        UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE COMPUTES THE DIAGONAL AND SUPERDIAGONAL
C TERMS OF THE TRIDIAGONAL LINEAR SYSTEM ASSOCIATED WITH
C SPLINE UNDER TENSION INTERPOLATION.
C
C ON INPUT--
C
C   SIGMA CONTAINS THE TENSION FACTOR.
C
C AND
C
C   DEL CONTAINS THE STEP SIZE.
C
C ON OUTPUT--
C
C                (SIGMA*DEL*COSH(SIGMA*DEL) - SINH(SIGMA*DEL)
C   DIAG = DEL*--------------------------------------------.
C                      (SIGMA*DEL)**2 * SINH(SIGMA*DEL)
C
C                     SINH(SIGMA*DEL) - SIGMA*DEL
C   SDIAG = DEL*----------------------------------.
C                 (SIGMA*DEL)**2 * SINH(SIGMA*DEL)
C
C AND
C
C   SIGMA AND DEL ARE UNALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULE SNHCSH.
C
C-----------------------------------------------------------
C
      IF (SIGMA .NE. 0.) GO TO 1
      DIAG = DEL/3.
      SDIAG = DEL/6.
      RETURN
    1 SIGDEL = SIGMA*DEL
      CALL SNHCSH (SINHM,COSHM,SIGDEL,0)
      DENOM = DEL/((SINHM+SIGDEL)*SIGDEL*SIGDEL)
      DIAG = DENOM*(SIGDEL*COSHM-SINHM)
      SDIAG = DENOM*SINHM
      RETURN
      END
C
C==================================================================
C
      SUBROUTINE TRIDEC (N,SUBDI,DIAGI,SUPD,SUBDO,DIAGO,
     *                   IERR)
C
      INTEGER N,IERR
      REAL SUBDI(N),DIAGI(N),SUPD(N),SUBDO(N),DIAGO(N)
C
C                                     FROM FITPACK -- AUGUST 31, 1981
C                                          CODED BY ALAN KAYLOR CLINE
C                                     DEPARTMENT OF COMPUTER SCIENCES
C                                       UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE FACTORIZES A TRIDIAGONAL MATRIX IN ORDER
C TO SOLVE SYSTEMS OF LINEAR EQUATIONS. THE FACTORIZATION
```

```
C EMPLOYS GAUSSIAN ELIMINATION WITHOUT ANY INTERCHANGING OF
C COLUMNS OR ROWS. THE SUBROUTINE TRISOL MAY BE CALLED TO
C ACTUALLY SOLVE THE SYSTEM ONCE THE FACTORIZATION HAS BEEN
C PERFORMED.
C
C ON INPUT--
C
C    N CONTAINS THE ORDER OF THE MATRIX (N .GE. 1).
C
C    SUBDI IS AN ARRAY CONTAINING THE SUBDIAGONAL ELEMENTS OF
C    THE MATRIX IN POSITIONS 2, ... , N.
C
C    DIAGI IS AN ARRAY CONTAINING THE DIAGONAL ELEMENTS OF
C    THE MATRIX.
C
C    SUPD IS AN ARRAY CONTAINING THE SUPERDIAGONAL ELEMENTS
C    OF THE MATRIX IN POSITIONS 1, ... , N-1.
C
C AND
C
C    SUBDO AND DIAGO ARE ARRAYS OF LENGTH N. (THE STORAGE
C    FOR THESE MAY COINCIDE WITH THAT FOR SUBDI AND DIAGI,
C    RESPECTIVELY, IN WHICH CASE THE ORIGINAL CONTENTS OF
C    SUBDI AND DIAGI WILL BE DESTROYED.)
C
C ON OUTPUT--
C
C    SUBDO AND DIAGO CONTAIN THE SUBDIAGONAL AND DIAGONAL OF
C    THE FACTORIZATION MATRIX.
C
C    IERR CONTAINS AN ERROR FLAG,
C         = 0 FOR NORMAL RETURN,
C         = 1 IF N IS LESS THAN 1,
C         = 2 IF THE SYSTEM IS SINGULAR.
C
C AND
C
C    N, SUBDI, DIAGI, AND SUPD ARE UNALTERED (UNLESS STORAGE
C    FOR SUBDI OR DIAGI COINCIDED WITH THAT FOR SUBDO
C    OR DIAGO, RESPECTIVELY).
C
C-----------------------------------------------------------
C
      IF (N .LE. 0) GO TO 3
      IERR = 2
      DIAGO(1) = DIAGI(1)
      IF (N .EQ. 1) GO TO 2
C
C FORWARD ELIMINATION
C
      DO 1 I = 2,N
        IM1 = I-1
        IF (DIAGO(IM1) .EQ. 0.) RETURN
        DIAGO(IM1) = 1./DIAGO(IM1)
        SUBDO(I) = SUBDI(I)*DIAGO(IM1)
    1   DIAGO(I) = DIAGI(I)-SUBDO(I)*SUPD(IM1)
    2 IF (DIAGO(N) .EQ. 0.) RETURN
      DIAGO(N) = 1./DIAGO(N)
      IERR = 0
      RETURN
```

```
C
C N LESS THAN 1
C
    3 IERR = 1
      RETURN
      END
C
C===========================================================================
C
      SUBROUTINE TRISOL (N,SUBD,DIAG,SUPD,RHS,MRHS,NUMRHS,
     *                   INCRHS)
C
      INTEGER N,MRHS,NUMRHS,INCRHS
      REAL SUBD(N),DIAG(N),SUPD(N),RHS(MRHS,NUMRHS)
C
C                                FROM FITPACK -- AUGUST 31, 1981
C                                   CODED BY ALAN KAYLOR CLINE
C                                DEPARTMENT OF COMPUTER SCIENCES
C                                  UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE SOLVES TRIDIAGONAL SYSTEMS OF LINEAR
C EQUATIONS WITH MULTIPLE RIGHT HAND SIDES. THE RIGHT HAND
C SIDES MAY BE STORED ROW-WISE OR COLUMN-WISE. THE
C SUBROUTINE TRIDEC SHOULD BE CALLED EARLIER TO DETERMINE A
C FACTORIZATION OF THE TRIDIAGONAL MATRIX. THE SOLUTION
C VECTORS OVER-WRITE THE RIGHT HAND SIDES.
C
C ON INPUT--
C
C   N CONTAINS THE ORDER OF THE MATRIX (N .GE. 1).
C
C   SUBD, DIAG, AND SUPD ARE ARRAYS OF LENGTH N CONTAINING
C   THE SUBDIAGONAL, DIAGONAL, AND SUPERDIAGONAL OF THE
C   FACTORIZATION, RESPECTIVELY.
C
C   RHS IS AN ARRAY CONTAINING THE RIGHT HAND SIDES OF THE
C   TRIDIAGONAL SYSTEM.
C
C   MRHS IS THE INCREMENT BETWEEN THE FIRST COMPONENTS OF
C   EACH OF THE RIGHT HAND SIDE VECTORS IN STORAGE (MRHS
C   .GE. 1).
C
C   NUMRHS IS THE NUMBER OF RIGHT HAND SIDES TO BE SOLVED.
C
C AND
C
C   INCRHS IS THE INCREMENT BETWEEN COMPONENTS WITHIN EACH
C   OF THE RIGHT HAND SIDE VECTORS IN STORAGE (INCRHS .GE.
C   1).
C
C THE PARAMETERS N, SUBD, DIAG, AND SUPD MAY BE INPUT AS THE
C PARAMETERS N, SUBDO, DIAGO, AND SUPD OUTPUT BY SUBROUTINE
C TRIDEC, RESPECTIVELY.
C
C ON OUTPUT--
C
C   RHS CONTAINS THE SOLUTION VECTORS IN THE SAME STORAGE
C   STRUCTURE AS FOR THE RIGHT HAND SIDES.
C
C AND
```

```
C
C   N, SUBD, DIAG, SUPD, MRHS, NUMRHS, AND INCRHS ARE
C   UNALTERED.
C
C-----------------------------------------------------------------
C
      NP1 = N+1
C
C LOOP ON RIGHT HAND SIDES
C
      DO 4 K = 1,NUMRHS
C
C FORWARD ELIMINATION
C
        IRHS = 1
        IF (N .EQ. 1) GO TO 2
        DO 1 I = 2,N
          IM1RHS = IRHS
          IRHS = IRHS+INCRHS
    1     RHS(IRHS,K) = RHS(IRHS,K)-SUBD(I)*RHS(IM1RHS,K)
C
C BACK SUBSTITUTION
C
    2   RHS(IRHS,K) = DIAG(N)*RHS(IRHS,K)
        IF (N .EQ. 1) GO TO 4
        DO 3 IBAK = 2,N
          I = NP1-IBAK
          RHS(IM1RHS,K) = DIAG(I)*(RHS(IM1RHS,K)-SUPD(I)
     *                                         *RHS(IRHS,K))
          IRHS = IM1RHS
    3     IM1RHS = IM1RHS-INCRHS
    4   CONTINUE
      RETURN
      END
C
C======================================================================
C
C PART 2:
C
C======================================================================
C
      SUBROUTINE CURV2D (T,YY,YX,YXX,N,X,Y,YP,SIGMA)
C
      INTEGER N
      REAL T,YY,YX,YXX,X(N),Y(N),YP(N),SIGMA
C
C                               FROM FITPACK -- AUGUST 31, 1981
C                                   CODED BY ALAN KAYLOR CLINE
C                               DEPARTMENT OF COMPUTER SCIENCES
C                                  UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE DETERMINES FUNCTION VALUE, FIRST, AND
C SECOND DERIVATIVES OF A CURVE AT A GIVEN POINT USING A
C SPLINE UNDER TENSION. THE SUBROUTINE CURV1 SHOULD BE
C CALLED EARLIER TO DETERMINE CERTAIN NECESSARY PARAMETERS.
C
C ON INPUT--
C
C   T CONTAINS A REAL VALUE AT WHICH THE FUNCTION AND
C   DERIVATIVES ARE TO BE EVALUATED.
```

```
C
C   N CONTAINS THE NUMBER OF POINTS WHICH WERE SPECIFIED TO
C   DETERMINE THE CURVE.
C
C   X AND Y ARE ARRAYS CONTAINING THE ABSCISSAE AND
C   ORDINATES, RESPECTIVELY, OF THE SPECIFIED POINTS.
C
C   YP IS AN ARRAY OF SECOND DERIVATIVE VALUES OF THE CURVE
C   AT THE NODES.
C
C AND
C
C   SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C THE PARAMETERS N, X, Y, YP, AND SIGMA SHOULD BE INPUT
C UNALTERED FROM THE OUTPUT OF CURV1.
C
C ON OUTPUT--
C
C   YY, YX, AND YXX CONTAIN THE FUNCTION VALUE, FIRST AND
C   SECOND DERIVATIVES, RESPECTIVELY.
C
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES INTRVL AND
C SNHCSH.
C
C-----------------------------------------------------------------
C
C DETERMINE INTERVAL
C
      IM1 = INTRVL(T,X,N)
      I = IM1+1
C
C DENORMALIZE TENSION FACTOR
C
      SIGMAP = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
C
C SET UP AND PERFORM INTERPOLATION
C
      DEL1 = T-X(IM1)
      DEL2 = X(I)-T
      DELS = X(I)-X(IM1)
      YY = (Y(I)*DEL1+Y(IM1)*DEL2)/DELS
      YX = (Y(I)-Y(IM1))/DELS
      IF (SIGMAP .NE. 0.) GO TO 1
      YY = YY-DEL1*DEL2*(YP(I)*(DEL1+DELS)+YP(IM1)*
     *        (DEL2+DELS))/(6.*DELS)
      YX = YX+(YP(I)*(2.*DEL1*DEL1-DEL2*(DEL1+DELS))-
     *         YP(IM1)*(2.*DEL2*DEL2-DEL1*(DEL2+DELS)))
     *         /(6.*DELS)
      YXX = (YP(I)*DEL1+YP(IM1)*DEL2)/DELS
      RETURN
    1 DELP1 = SIGMAP*(DEL1+DELS)/2.
      DELP2 = SIGMAP*(DEL2+DELS)/2.
      CALL SNHCSH (SINHM1,COSHM1,SIGMAP*DEL1,0)
      CALL SNHCSH (SINHM2,COSHM2,SIGMAP*DEL2,0)
      CALL SNHCSH (SINHMS,DUMMY,SIGMAP*DELS,-1)
      CALL SNHCSH (SINHP1,DUMMY,SIGMAP*DEL1/2.,-1)
      CALL SNHCSH (SINHP2,DUMMY,SIGMAP*DEL2/2.,-1)
```

```
      CALL SNHCSH (DUMMY,COSHP1,DELP1,1)
      CALL SNHCSH (DUMMY,COSHP2,DELP2,1)
      YY = YY+(YP(I)*(SINHM1*DEL2-DEL1*(2.*(COSHP1+1.)*
     *         SINHP2+SIGMAP*COSHP1*DEL2))+YP(IM1)*(SINHM2*
     *         DEL1-DEL2*(2.*(COSHP2+1.)*SINHP1+SIGMAP*
     *         COSHP2*DEL1)))/(SIGMAP*SIGMAP*DELS*(SINHMS+
     *         SIGMAP*DELS))
      YX = YX+(YP(I)*(DELS*SIGMAP*COSHM1-SINHMS)-
     *     YP(IM1)*(DELS*SIGMAP*COSHM2-SINHMS))/
     *     (SIGMAP*SIGMAP*DELS*(SINHMS+SIGMAP*DELS))
      YXX = (YP(I)*(SINHM1+SIGMAP*DEL1)+YP(IM1)*(SINHM2+
     *       SIGMAP*DEL2))/(SINHMS+SIGMAP*DELS)
      RETURN
      END
C
C=====================================================================
C
      SUBROUTINE CURVBD (XX,W,WX,WXX,NX,X,C,VX,SIGMA)
C
      INTEGER NX
      REAL XX,W,WX,WXX,X(NX),VX(5,NX),C(NX),SIGMA
C
C                                       COMPLEMENT TO FITPACK
C                                        BY ALAN KAYLOR CLINE
C                                    CODED -- OCTOBER 9, 1986
C                                            BY LUDEK KLIMES
C                                      INST. GEOL. GEOTECHN.
C                                 CZECHOSL. ACAD. SCI., PRAGUE
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE
C FIRST PARTIAL DERIVATIVE, AND THE SECOND PARTIAL
C DERIVATIVE OF A SPLINE UNDER TENSION IN ONE VARIABLE.
C
C ON INPUT--
C
C   XX CONTAINS THE X-COORDINATE OF THE POINT
C   AT WHICH THE INTERPOLATION IS TO BE PERFORMED
C
C   NX IS THE NUMBER OF GRID POINTS
C
C   X IS ARRAY CONTAINING THE X-GRID VALUES.
C
C   C IS AN ARRAY OF COEFFICIENTS DESCRIBING THE FUNCTION IN
C   TERMS OF A B-SPLINE UNDER TENSION BASIS. IN THE
C   EXPANSION OF THE FUNCTION, FOR I = 1,...,NX ,
C   THE COEFFICIENT MULTIPLYING THE BASIS
C   FUNCTION I IS STORED IN C(I).
C
C   VX IS THE ARRAY OF LENGTH 5*NX
C   CONTAINING THE B-SPLINE BASIS DATA
C
C   SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C THE PARAMETERS NX, X, C, VX, AND SIGMA
C SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF CURVB1.
C
C ON OUTPUT--
C
C   W CONTAINS THE INTERPOLATED FUNCTION VALUE.
C
```

```
C   WX CONTAINS THE FIRST DERIVATIVE .
C
C   WXX CONTAINS THE SECOND DERIVATIVE .
C
C AND
C
C   NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES DSPLNZ, INTRVL,
C AND SNHCSH.
C
C-----------------------------------------------------------------------
C
      REAL BX(3,4)
C
C EVALUATE BASIS FUNCTIONS AT XX
C
      CALL DSPLNZ (XX,NX,X,VX,SIGMA,ISTART,BX)
C
C ACCUMULATE BASIS FUNCTIONS
C
      SUM = 0.
      SUMX = 0.
      SUMXX = 0.
      DO 1 I = 1,4
        II = ISTART+I-1
        IF (II .EQ. 0 .OR. II .GT. NX) GO TO 1
        BX1I = BX(1,I)
        CI  = C(II)
        SUM = SUM+CI*BX1I
        SUMX = SUMX+CI*BX(2,I)
        SUMXX = SUMXX+CI*BX(3,I)
    1   CONTINUE
      W = SUM
      WX = SUMX
      WXX = SUMXX
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SURFBD (XX,YY,W,WX,WY,WXX,WXY,WYY,NX,NY,X,
     *                   Y,C,VX,VY,SIGMA)
C
      INTEGER NX,NY
      REAL XX,YY,W,WX,WY,WXX,WXY,WYY,X(NX),Y(NY),VX(5,NX),
     *     VY(5,NY),C(NX,NY),SIGMA
C
C                                  FROM FITPACK -- AUGUST 31, 1981
C                                       CODED BY ALAN KAYLOR CLINE
C                                  DEPARTMENT OF COMPUTER SCIENCES
C                                    UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE TWO
C FIRST PARTIAL DERIVATIVES, AND THE SIX SECOND PARTIAL
C DERIVATIVES OF A TENSOR PRODUCT SPLINE UNDER TENSION IN
C TWO VARIABLES.
C
C ON INPUT--
C
```

```
C   XX AND YY CONTAIN THE X- AND Y-COORDINES OF THE POINT
C   AT WHICH THE INTERPOLATION IS TO BE PERFORME .D.
C
C   NX AND NY ARE THE NUMBER OF GRID LINES IN THE X- AND Y-
C   DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR GRID WHICH
C   SPECIFIED THE FUNCTION.
C
C   X AND Y ARE ARRAYS CONTAINING THE X- AND Y-GRID VALUES,
C   RESPECTIVELY.
C
C   C IS AN ARRAY OF COEFFICIENTS DESCRIBING THE FUNCTION IN
C   TERMS OF A B-SPLINE UNDER TENSION BASIS. IN THE
C   EXPANSION OF THE FUNCTION, FOR I = 1,...,NX AND J = 1,
C   ...,NY, THE COEFFICIENT MULTIPLYING THE PRODUCT OF BASIS
C   FUNCTION I IN X AND BASIS FUNCTION J IN Y IS STORED IN
C   C(I,J).
C
C   VX AND VY VZ ARE ARRAYS OF LENGTH 5*NX AND 5*NY,
C   RESPECTIVELY, CONTAINING THE B-SPLINE BASIS DATA FOR THE
C   X- AND Y-GRIDS.
C
C AND
C
C   SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C THE PARAMETERS NX, NY, X, Y, Z, C, VX, VY, AND SIGMA
C SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF SURFB1.
C
C ON OUTPUT--
C
C   W CONTAINS THE INTERPOLATED FUNCTION VALUE.
C
C   WX AND WY CONTAIN THE X- AND Y-PARTIAL DERIVATIVES,
C   RESPECTIVELY.
C
C   WXX, WXY, AND WYY CONTAIN THE XX-, XY-, AND YY-PARTIAL
C   DERIVATIVES, RESPECTIVELY.
C
C AND
C
C   NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES DSPLNZ, INTRVL,
C AND SNHCSH.
C
C-----------------------------------------------------------------------
C
      REAL BX(3,4),BY(3,4)
C
C EVALUATE BASIS FUNCTIONS AT XX AND YY
C
      CALL DSPLNZ (XX,NX,X,VX,SIGMA,ISTART,BX)
      CALL DSPLNZ (YY,NY,Y,VY,SIGMA,JSTART,BY)
C
C ACCUMULATE TENSOR PRODUCTS
C
      SUM = 0.
      SUMX = 0.
      SUMY = 0.
      SUMXX = 0.
```

```
      SUMXY = 0.
      SUMYY = 0.
      DO 2 J = 1,4
        JJ = JSTART+J-1
        IF (JJ .EQ. 0 .OR. JJ .GT. NY) GO TO 2
        BY1J = BY(1,J)
        BY2J = BY(2,J)
        BY3J = BY(3,J)
        DO 1 I = 1,4
          II = ISTART+I-1
          IF (II .EQ. 0 .OR. II .GT. NX) GO TO 1
          BX1I = BX(1,I)
          BX2I =  BX(2,I)
          CIJ = C(II,JJ)
          SUM = SUM+CIJ*BX1I*BY1J
          SUMX = SUMX+CIJ*BX2I*BY1J
          SUMY = SUMY+CIJ*BX1I*BY2J
          SUMXX = SUMXX+CIJ*BX(3,I)*BY1J
          SUMXY = SUMXY+CIJ*BX2I*BY2J
          SUMYY = SUMYY+CIJ*BX1I*BY3J
    1     CONTINUE
    2   CONTINUE
      W = SUM
      WX = SUMX
      WY = SUMY
      WXX = SUMXX
      WXY = SUMXY
      WYY = SUMYY
      RETURN
      END
C
C=====================================================================
C
      SUBROUTINE VAL3BD (XX,YY,ZZ,W,WX,WY,WZ,WXX,WXY,WYY,
     *                   WYZ,WZZ,WXZ,NX,NY,NZ,X,Y,Z,C,VX,VY,
     *                   VZ,SIGMA)
C
      INTEGER NX,NY,NZ
      REAL XX,YY,ZZ,W,WX,WY,WZ,WXX,WXY,WYY,WYZ,WZZ,WXZ,
     *     X(NX),Y(NY),Z(NZ),VX(5,NX),VY(5,NY),VZ(5,NZ),
     *     C(NX,NY,NZ),SIGMA
C
C                                 FROM FITPACK -- AUGUST 31, 1981
C                                      CODED BY ALAN KAYLOR CLINE
C                                 DEPARTMENT OF COMPUTER SCIENCES
C                                   UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE THREE
C FIRST PARTIAL DERIVATIVES, AND THE SIX SECOND PARTIAL
C DERIVATIVES OF A TENSOR PRODUCT SPLINE UNDER TENSION IN
C THREE VARIABLES.
C
C ON INPUT--
C
C   XX, YY, AND ZZ CONTAIN THE X-, Y-, AND Z-COORDINATES OF
C   THE POINT AT WHICH THE INTERPOLATION IS TO BE PERFORMED.
C
C   NX, NY, AND NZ ARE THE NUMBER OF GRID LINES IN THE X-,
C   Y-, AND Z-DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR
C   GRID WHICH SPECIFIED THE FUNCTION.
```

```
C
C    X, Y, AND Z ARE ARRAYS CONTAINING THE X-, Y-, AND Z-GRID
C    VALUES, RESPECTIVELY.
C
C    C IS AN ARRAY OF COEFFICIENTS DESCRIBING THE FUNCTION IN
C    TERMS OF A B-SPLINE UNDER TENSION BASIS. IN THE
C    EXPANSION OF THE FUNCTION, FOR I = 1,...,NX, J = 1,...,
C    NY, AND K = 1,...,NZ, THE COEFFICIENT MULTIPLYING THE
C    PRODUCT OF BASIS FUNCTION I IN X, BASIS FUNCTION J IN Y,
C    AND BASIS FUNCTION K IN Z IS STORED IN C(I,J,K).
C
C    VX, VY, AND VZ ARE ARRAYS OF LENGTH 5*NX, 5*NY, AND
C    5*NZ, RESPECTIVELY, CONTAINING THE B-SPLINE BASIS DATA
C    FOR THE X-, Y-, AND Z-GRIDS.
C
C AND
C
C    SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C THE PARAMETERS NX, NY, NZ, X, Y, Z, C, VX, VY, VZ, AND
C SIGMA SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF
C VAL3B1.
C
C ON OUTPUT--
C
C    W CONTAINS THE INTERPOLATED FUNCTION VALUE.
C
C    WX, WY, AND WZ CONTAIN THE X-, Y-, AND Z-PARTIAL
C    DERIVATIVES, RESPECTIVELY.
C
C    WXX, WXY, WYY, WYZ, WZZ, AND WXZ CONTAIN THE XX-, XY-
C    YY-, YZ-, ZZ-, AND XZ-PARTIAL DERIVATIVES, RESPECTIVELY.
C
C AND
C
C    NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES DSPLNZ, INTRVL,
C AND SNHCSH.
C
C-----------------------------------------------------------------
C
      REAL BX(3,4),BY(3,4),BZ(3,4)
C
C EVALUATE BASIS FUNCTIONS AT XX, YY, AND ZZ
C
      CALL DSPLNZ (XX,NX,X,VX,SIGMA,ISTART,BX)
      CALL DSPLNZ (YY,NY,Y,VY,SIGMA,JSTART,BY)
      CALL DSPLNZ (ZZ,NZ,Z,VZ,SIGMA,KSTART,BZ)
C
C ACCUMULATE TENSOR PRODUCTS
C
      SUM = 0.
      SUMX = 0.
      SUMY = 0.
      SUMZ = 0.
      SUMXX = 0.
      SUMXY = 0.
      SUMYY = 0.
      SUMYZ = 0.
```

```
      SUMZZ = 0.
      SUMXZ = 0.
      DO 3 K = 1,4
        KK = KSTART+K-1
        IF (KK .EQ. 0 .OR. KK .GT. NZ) GO TO 3
        BZ1K = BZ(1,K)
        BZ2K = BZ(2,K)
        BZ3K = BZ(3,K)
        DO 2 J = 1,4
          JJ = JSTART+J-1
          IF (JJ .EQ. 0 .OR. JJ .GT. NY) GO TO 2
          BY1J = BY(1,J)
          BY2J = BY(2,J)
          BY3J = BY(3,J)
          DO 1 I = 1,4
            II = ISTART+I-1
            IF (II .EQ. 0 .OR. II .GT. NX) GO TO 1
            BX1I = BX(1,I)
            BX2I =  BX(2,I)
            CIJK = C(II,JJ,KK)
            SUM = SUM+CIJK*BX1I*BY1J*BZ1K
            SUMX = SUMX+CIJK*BX2I*BY1J*BZ1K
            SUMY = SUMY+CIJK*BX1I*BY2J*BZ1K
            SUMZ = SUMZ+CIJK*BX1I*BY1J*BZ2K
            SUMXX = SUMXX+CIJK*BX(3,I)*BY1J*BZ1K
            SUMXY = SUMXY+CIJK*BX2I*BY2J*BZ1K
            SUMYY = SUMYY+CIJK*BX1I*BY3J*BZ1K
            SUMYZ = SUMYZ+CIJK*BX1I*BY2J*BZ2K
            SUMZZ = SUMZZ+CIJK*BX1I*BY1J*BZ3K
            SUMXZ = SUMXZ+CIJK*BX2I*BY1J*BZ2K
    1       CONTINUE
    2     CONTINUE
    3   CONTINUE
      W = SUM
      WX = SUMX
      WY = SUMY
      WZ = SUMZ
      WXX = SUMXX
      WXY = SUMXY
      WYY = SUMYY
      WYZ = SUMYZ
      WZZ = SUMZZ
      WXZ = SUMXZ
      RETURN
      END
C
C=====================================================================
C
      SUBROUTINE DSPLNZ (T,N,X,V,SIGMA,ISTART,B)
C
      INTEGER N,ISTART
      REAL T,X(N),V(5,N),SIGMA,B(3,4)
C
C                                  FROM FITPACK -- AUGUST 31, 1981
C                                       CODED BY ALAN KAYLOR CLINE
C                                  DEPARTMENT OF COMPUTER SCIENCES
C                                    UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS SUBROUTINE EVALUATES AT A GIVEN POINT THE FOUR NON-
C ZERO BASIS FUNCTIONS OF A B-SPLINE UNDER TENSION BASIS AND
```

```
C THEIR FIRST AND SECOND DERIVATIVES. THE INDEX OF THE FIRST
C NON-ZERO BASIS FUNCTION IS ALSO DETERMINED. (THE SENSE OF
C THE WORD NON-ZERO IS EXTENDED TO INCLUDE THE SPECIAL CASE
C WHERE THE GIVEN POINT COINCIDES WITH A KNOT IN WHICH CASE
C THE LAST OF THE FOUR RETURNED FUNCTION VALUES MAY BE ZERO.
C ) THE SUBROUTINE VGEN SHOULD BE CALLED EARLIER TO
C DETERMINE CERTAIN NECESSARY COEFFICIENTS.
C
C ON INPUT--
C
C    T CONTAINS A REAL VALUE AT WHICH THE BASIS FUNCTIONS ARE
C    TO BE EVALUATED.
C
C    N CONTAINS THE NUMBER OF KNOTS DEFINING THE BASIS.
C
C    X CONTAINS THE ARRAY OF KNOTS.
C
C    V CONTAINS THE ARRAY OF COEFFICIENTS DETERMINED BY VGEN
C    FOR CALCULATION OF BASIS FUNCTIONS.
C
C    SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C    ISTART IS AN INTEGER VARIABLE.
C
C AND
C
C    B IS A REAL ARRAY WITH 3 ROWS AND 4 COLUMNS.
C
C THE PARAMETERS N, X, V, AND SIGMA SHOULD BE INPUT
C UNALTERED FROM THE OUTPUT OF VGEN.
C
C ON OUTPUT--
C
C    ISTART CONTAINS THE INDEX OF THE FIRST NON-ZERO BASIS
C    FUNCTION. THUS 0 .LE. ISTART .LE. N-2 AND THE NON-ZERO
C    BASIS FUNCTIONS HAVE INDICES ISTART, ... , ISTART+3.
C
C    B CONTAINS THE VALUES AT T OF BASIS FUNCTIONS ISTART,
C    ... , ISTART+3 IN B(1,1), ... , B(1,4), RESPECTIVELY.
C    FIRST AND SECOND DERIVATIVES OF THE CORRESPONDING
C    FUNCTIONS ARE CONTAINED IN B(2,1), ... , B(2,4), AND
C    B(3,1), ... , B(3,4), RESPECTIVELY.
C
C    T, N, X, V, AND SIGMA ARE UNALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES INTRVL AND
C SNHCSH.
C
C-----------------------------------------------------------
C
C DENORMALIZE TENSION FACTOR
C
      SIGMAP = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
C
C DETERMINE INDEX OF FIRST NON-ZERO BASIS FUNCTION
C
      I = INTRVL (T,X,N)-1
C
C COMPUTE DISTANCES TO ADJACENT KNOTS AND LAGRANGIAN
C WEIGHTS
```

```
C
      DEL1 = T-X(I+1)
      DEL2 = X(I+2)-T
      DELS = X(I+2)-X(I+1)
      C10 = DEL2/DELS
      C20 = DEL1/DELS
      C11 = -1./DELS
      C21 = 1./DELS
      IF (SIGMAP .NE. 0.) GO TO 1
      FAC = -DEL1*DEL2/(6.*DELS)
      CP10 = FAC*(DEL2+DELS)
      CP20 = FAC*(DEL1+DELS)
      CP11 = -(2.*DEL2*DEL2-DEL1*(DEL2+DELS))/(6.*DELS)
      CP21 = (2.*DEL1*DEL1-DEL2*(DEL1+DELS))/(6.*DELS)
      CP12 = C10
      CP22 = C20
      GO TO 2
    1 DELP1 = SIGMAP*(DEL1+DELS)/2.
      DELP2 = SIGMAP*(DEL2+DELS)/2.
      CALL SNHCSH (SINHM1,COSHM1,SIGMAP*DEL1,0)
      CALL SNHCSH (SINHM2,COSHM2,SIGMAP*DEL2,0)
      CALL SNHCSH (SINHMS,DUMMY,SIGMAP*DELS,-1)
      CALL SNHCSH (SINHP1,DUMMY,SIGMAP*DEL1/2.,-1)
      CALL SNHCSH (SINHP2,DUMMY,SIGMAP*DEL2/2.,-1)
      CALL SNHCSH (DUMMY,COSHP1,DELP1,1)
      CALL SNHCSH (DUMMY,COSHP2,DELP2,1)
      SINHS = SINHMS+SIGMAP*DELS
      DENOM = SIGMAP*SIGMAP*DELS*SINHS
      CP10 = (SINHM2*DEL1-DEL2*(2.*(COSHP2+1.)*SINHP1
     *        +SIGMAP*COSHP2*DEL1))/DENOM
      CP20 = (SINHM1*DEL2-DEL1*(2.*(COSHP1+1.)*SINHP2
     *        +SIGMAP*COSHP1*DEL2))/DENOM
      CP11 = -(DELS*SIGMAP*COSHM2-SINHMS)/DENOM
      CP21 = (DELS*SIGMAP*COSHM1-SINHMS)/DENOM
      CP12 = (SINHM2+SIGMAP*DEL2)/SINHS
      CP22 = (SINHM1+SIGMAP*DEL1)/SINHS
C
C COMPUTE BASIS FUNCTION VALUES
C
    2 II = I
      IF (II .EQ. 0) II = N
      IIP1 = I+1
      IIP2 = I+2
      IIP3 = I+3
      IF (IIP2 .EQ. N) IIP3 = 1
      B(1,1) = C10*V(5,II)+CP10*V(3,II)
      B(1,2) = C10+C20*V(5,IIP1)+CP10*V(2,IIP1)+
     *         CP20*V(3,IIP1)
      B(1,3) = C10*V(4,IIP2)+C20+CP10*V(1,IIP2)+
     *         CP20*V(2,IIP2)
      B(1,4) = C20*V(4,IIP3)+CP20*V(1,IIP3)
      B(2,1) = C11*V(5,II)+CP11*V(3,II)
      B(2,2) = C11+C21*V(5,IIP1)+CP11*V(2,IIP1)+
     *         CP21*V(3,IIP1)
      B(2,3) = C11*V(4,IIP2)+C21+CP11*V(1,IIP2)+
     *         CP21*V(2,IIP2)
      B(2,4) = C21*V(4,IIP3)+CP21*V(1,IIP3)
      B(3,1) = CP12*V(3,II)
      B(3,2) = CP12*V(2,IIP1)+CP22*V(3,IIP1)
      B(3,3) = CP12*V(1,IIP2)+CP22*V(2,IIP2)
```

```
      B(3,4) = CP22*V(1,IIP3)
      ISTART = I
      RETURN
      END
C
C==========================================================================
C
      FUNCTION INTRVL (T,X,N)
C
      INTEGER N
      REAL T,X(N)
C
C                                     FROM FITPACK -- AUGUST 31, 1981
C                                CODED BY A. K. CLINE AND R. J. RENKA
C                                     DEPARTMENT OF COMPUTER SCIENCES
C                                       UNIVERSITY OF TEXAS AT AUSTIN
C
C THIS FUNCTION DETERMINES THE INDEX OF THE INTERVAL
C (DETERMINED BY A GIVEN INCREASING SEQUENCE) IN WHICH
C A GIVEN VALUE LIES.
C
C ON INPUT--
C
C   T IS THE GIVEN VALUE.
C
C   X IS A VECTOR OF STRICTLY INCREASING VALUES.
C
C AND
C
C   N IS THE LENGTH OF X (N .GE. 2).
C
C ON OUTPUT--
C
C   INTRVL RETURNS AN INTEGER I SUCH THAT
C
C          I = 1        IF              T .LT. X(2)    ,
C          I = N-1      IF X(N-1) .LE. T               ,
C          OTHERWISE       X(I)   .LE. T .LT. X(I+1),
C
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C-----------------------------------------------------------------
C
      TT = T
      IF (TT .LT. X(2)) GO TO 4
      IF (TT .GE. X(N-1)) GO TO 5
      IL = 2
      IH = N-1
C
C LINEAR INTERPOLATION
C
    1 I = MIN0(IL+IFIX(FLOAT(IH-IL)*(TT-X(IL))/(X(IH)-X(IL))),
     *          IH-1)
      IF (TT .LT. X(I)) GO TO 2
      IF (TT .LT. X(I+1)) GO TO 3
C
C TOO HIGH
C
      IL = I+1
      GO TO 1
```

```
C
C TOO LOW
C
    2 IH = I
      GO TO 1
    3 INTRVL = I
      RETURN
C
C LEFT END
C
    4 INTRVL = 1
      RETURN
C
C RIGHT END
C
    5 INTRVL = N-1
      RETURN
      END
C
C==========================================================================
C
C PART 3:
C
C==========================================================================
C
      SUBROUTINE CURVBV (XX,W,WX,WXX,NX,X,C,VX,SIGMA,
     *                   NB,IB,BB)
C
      INTEGER NX
      REAL XX,W,WX,WXX,X(NX),VX(5,NX),C(NX),SIGMA
      INTEGER NB,IB(4)
      REAL    BB(4)
C
C                                                 COMPLEMENT TO FITPACK
C                                                  BY ALAN KAYLOR CLINE
C                                              CODED -- OCTOBER 9, 1986
C                                                       BY LUDEK KLIMES
C                                                 INST. GEOL. GEOTECHN.
C                                          CZECHOSL. ACAD. SCI., PRAGUE
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE
C FIRST PARTIAL DERIVATIVE, AND THE SECOND PARTIAL
C DERIVATIVE OF A SPLINE UNDER TENSION IN ONE VARIABLE.
C IT ALSO RETURNS THE VARIATIONS OF THE FUNCTION VALUE WITH
C RESPECT TO THE B-SPLINE EXPANSION COEFFICIENTS.
C
C ON INPUT--
C
C   XX CONTAINS THE X-COORDINATE OF THE POINT
C   AT WHICH THE INTERPOLATION IS TO BE PERFORMED
C
C   NX IS THE NUMBER OF GRID POINTS
C
C   X IS ARRAY CONTAINING THE X-GRID VALUES.
C
C   C IS AN ARRAY OF COEFFICIENTS DESCRIBING THE FUNCTION IN
C   TERMS OF A B-SPLINE UNDER TENSION BASIS. IN THE
C   EXPANSION OF THE FUNCTION, FOR I = 1,...,NX ,
C   THE COEFFICIENT MULTIPLYING THE BASIS
C   FUNCTION I IS STORED IN C(I).
```

```
C
C     VX IS THE ARRAY OF LENGTH 5*NX
C     CONTAINING THE B-SPLINE BASIS DATA
C
C     SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C     IB IS AN ARRAY OF AT LEAST 4 LOCATIONS.
C
C     BB IS AN ARRAY OF AT LEAST 4 LOCATIONS.
C
C THE PARAMETERS NX, X, C, VX, AND SIGMA
C SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF CURVB1.
C
C ON OUTPUT--
C
C     W CONTAINS THE INTERPOLATED FUNCTION VALUE.
C
C     WX CONTAINS THE FIRST DERIVATIVE .
C
C     WXX CONTAINS THE SECOND DERIVATIVE .
C
C     NB CONTAINS THE NUMBER OF THE B-SPLINE EXPANSION
C     COEFFICIENTS (SEE INPUT ARRAY C) INFLUENCING THE
C     FUNCTION VALUE AT THE GIVEN POINT.
C
C     IB CONTAINS NB INDICES IN ARRAY C OF THE B-SPLINE
C     EXPANSION COEFFICIENTS INFLUENCING THE FUNCTION VALUE AT
C     THE GIVEN POINT.
C
C     BB CONTAINS NB VARIATIONS OF THE FUNCTION VALUE WITH
C     RESPECT TO THE B-SPLINE EXPANSION COEFFICIENTS SPECIFIED
C     IN ARRAY IB.
C
C AND
C
C     NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES DSPLNZ, INTRVL,
C AND SNHCSH.
C
C-----------------------------------------------------------------
C
      REAL BX(3,4)
C
C EVALUATE BASIS FUNCTIONS AT XX
C
      CALL DSPLNZ (XX,NX,X,VX,SIGMA,ISTART,BX)
C
C ACCUMULATE BASIS FUNCTIONS
C
      NB = 0
      SUM = 0.
      SUMX = 0.
      SUMXX = 0.
      DO 1 I = 1,4
        II = ISTART+I-1
        IF (II .EQ. 0 .OR. II .GT. NX) GO TO 1
        BX1I = BX(1,I)
        CI  = C(II)
        NB    = NB+1
```

```
         JB    = II
         IB(NB)= JB
         BB(NB)= BX1I
         SUM = SUM+CI*BX1I
         SUMX = SUMX+CI*BX(2,I)
         SUMXX = SUMXX+CI*BX(3,I)
    1    CONTINUE
      W = SUM
      WX = SUMX
      WXX = SUMXX
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SURFBV (XX,YY,W,WX,WY,WXX,WXY,WYY,NX,NY,X,
     *                   Y,C,VX,VY,SIGMA,NB,IB,BB)
C
      INTEGER NX,NY
      REAL XX,YY,W,WX,WY,WXX,WXY,WYY,X(NX),Y(NY),VX(5,NX),
     *     VY(5,NY),C(NX,NY),SIGMA
      INTEGER NB,IB(16)
      REAL    BB(16)
C
C                              FROM FITPACK -- AUGUST 31, 1981
C                                 CODED BY ALAN KAYLOR CLINE
C                              DEPARTMENT OF COMPUTER SCIENCES
C                                UNIVERSITY OF TEXAS AT AUSTIN
C                                  MODIFIED -- OCTOBER 9, 1986
C                                             BY LUDEK KLIMES
C                                       INST. GEOL. GEOTECHN.
C                                 CZECHOSL. ACAD. SCI., PRAGUE
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE TWO
C FIRST PARTIAL DERIVATIVES, AND THE SIX SECOND PARTIAL
C DERIVATIVES OF A TENSOR PRODUCT SPLINE UNDER TENSION IN
C TWO VARIABLES. IT ALSO RETURNS THE VARIATIONS OF THE
C FUNCTION VALUE WITH RESPECT TO THE B-SPLINE EXPANSION
C COEFFICIENTS.
C
C ON INPUT--
C
C   XX AND YY CONTAIN THE X- AND Y-COORDINES OF THE POINT
C   AT WHICH THE INTERPOLATION IS TO BE PERFORME .D.
C
C   NX AND NY ARE THE NUMBER OF GRID LINES IN THE X- AND Y-
C   DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR GRID WHICH
C   SPECIFIED THE FUNCTION.
C
C   X AND Y ARE ARRAYS CONTAINING THE X- AND Y-GRID VALUES,
C   RESPECTIVELY.
C
C   C IS AN ARRAY OF COEFFICIENTS DESCRIBING THE FUNCTION IN
C   TERMS OF A B-SPLINE UNDER TENSION BASIS. IN THE
C   EXPANSION OF THE FUNCTION, FOR I = 1,...,NX AND J = 1,
C   ...,NY, THE COEFFICIENT MULTIPLYING THE PRODUCT OF BASIS
C   FUNCTION I IN X AND BASIS FUNCTION J IN Y IS STORED IN
C   C(I,J).
C
C   VX AND VY VZ ARE ARRAYS OF LENGTH 5*NX AND 5*NY,
```

```
C    RESPECTIVELY, CONTAINING THE B-SPLINE BASIS DATA FOR THE
C    X- AND Y-GRIDS.
C
C    SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C    IB IS AN ARRAY OF AT LEAST 16 LOCATIONS.
C
C    BB IS AN ARRAY OF AT LEAST 16 LOCATIONS.
C
C THE PARAMETERS NX, NY, X, Y, Z, C, VX, VY, AND SIGMA
C SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF SURFB1.
C
C ON OUTPUT--
C
C    W CONTAINS THE INTERPOLATED FUNCTION VALUE.
C
C    WX AND WY CONTAIN THE X- AND Y-PARTIAL DERIVATIVES,
C    RESPECTIVELY.
C
C    WXX, WXY, AND WYY CONTAIN THE XX-, XY-, AND YY-PARTIAL
C    DERIVATIVES, RESPECTIVELY.
C
C    NB CONTAINS THE NUMBER OF THE B-SPLINE EXPANSION
C    COEFFICIENTS (SEE INPUT ARRAY C) INFLUENCING THE
C    FUNCTION VALUE AT THE GIVEN POINT.
C
C    IB CONTAINS NB INDICES IN ARRAY C OF THE B-SPLINE
C    EXPANSION COEFFICIENTS INFLUENCING THE FUNCTION VALUE AT
C    THE GIVEN POINT. THE ELEMENTS OF ARRAY C ARE INDEXED AS
C    IF IT WAS A ONE-DIMENSIONAL ARRAY.
C
C    BB CONTAINS NB VARIATIONS OF THE FUNCTION VALUE WITH
C    RESPECT TO THE B-SPLINE EXPANSION COEFFICIENTS SPECIFIED
C    IN ARRAY IB.
C
C AND
C
C    NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES DSPLNZ, INTRVL,
C AND SNHCSH.
C
C-----------------------------------------------------------------
C
      REAL BX(3,4),BY(3,4)
C
C EVALUATE BASIS FUNCTIONS AT XX AND YY
C
      CALL DSPLNZ (XX,NX,X,VX,SIGMA,ISTART,BX)
      CALL DSPLNZ (YY,NY,Y,VY,SIGMA,JSTART,BY)
C
C ACCUMULATE TENSOR PRODUCTS
C
      NB = 0
      SUM = 0.
      SUMX = 0.
      SUMY = 0.
      SUMXX = 0.
      SUMXY = 0.
      SUMYY = 0.
```

```
      DO 2 J = 1,4
        JJ = JSTART+J-1
        IF (JJ .EQ. 0 .OR. JJ .GT. NY) GO TO 2
        BY1J = BY(1,J)
        BY2J = BY(2,J)
        BY3J = BY(3,J)
        DO 1 I = 1,4
          II = ISTART+I-1
          IF (II .EQ. 0 .OR. II .GT. NX) GO TO 1
          BX1I = BX(1,I)
          BX2I =  BX(2,I)
          CIJ = C(II,JJ)
          NB    = NB+1
          JB    = II+NX*(JJ-1)
          IB(NB)= JB
          BB(NB)= BX1I*BY1J
          SUM = SUM+CIJ*BX1I*BY1J
          SUMX = SUMX+CIJ*BX2I*BY1J
          SUMY = SUMY+CIJ*BX1I*BY2J
          SUMXX = SUMXX+CIJ*BX(3,I)*BY1J
          SUMXY = SUMXY+CIJ*BX2I*BY2J
          SUMYY = SUMYY+CIJ*BX1I*BY3J
    1     CONTINUE
    2   CONTINUE
      W = SUM
      WX = SUMX
      WY = SUMY
      WXX = SUMXX
      WXY = SUMXY
      WYY = SUMYY
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE VAL3BV (XX,YY,ZZ,W,WX,WY,WZ,WXX,WXY,WYY,
     *                   WYZ,WZZ,WXZ,NX,NY,NZ,X,Y,Z,C,VX,VY,
     *                   VZ,SIGMA,NB,IB,BB)
C
      INTEGER NX,NY,NZ
      REAL XX,YY,ZZ,W,WX,WY,WZ,WXX,WXY,WYY,WYZ,WZZ,WXZ,
     *     X(NX),Y(NY),Z(NZ),VX(5,NX),VY(5,NY),VZ(5,NZ),
     *     C(NX,NY,NZ),SIGMA
      INTEGER NB,IB(64)
      REAL    BB(64)
C
C                                 FROM FITPACK -- AUGUST 31, 1981
C                                      CODED BY ALAN KAYLOR CLINE
C                                 DEPARTMENT OF COMPUTER SCIENCES
C                                   UNIVERSITY OF TEXAS AT AUSTIN
C                                     MODIFIED -- OCTOBER 9, 1986
C                                                BY LUDEK KLIMES
C                                          INST. GEOL. GEOTECHN.
C                                    CZECHOSL. ACAD. SCI., PRAGUE
C
C THIS SUBROUTINE EVALUATES THE FUNCTION VALUE, THE THREE
C FIRST PARTIAL DERIVATIVES, AND THE SIX SECOND PARTIAL
C DERIVATIVES OF A TENSOR PRODUCT SPLINE UNDER TENSION IN
C THREE VARIABLES. IT ALSO RETURNS THE VARIATIONS OF THE
C FUNCTION VALUE WITH RESPECT TO THE B-SPLINE EXPANSION
```

```
C COEFFICIENTS.
C
C ON INPUT--
C
C   XX, YY, AND ZZ CONTAIN THE X-, Y-, AND Z-COORDINATES OF
C   THE POINT AT WHICH THE INTERPOLATION IS TO BE PERFORMED.
C
C   NX, NY, AND NZ ARE THE NUMBER OF GRID LINES IN THE X-,
C   Y-, AND Z-DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR
C   GRID WHICH SPECIFIED THE FUNCTION.
C
C   X, Y, AND Z ARE ARRAYS CONTAINING THE X-, Y-, AND Z-GRID
C   VALUES, RESPECTIVELY.
C
C   C IS AN ARRAY OF COEFFICIENTS DESCRIBING THE FUNCTION IN
C   TERMS OF A B-SPLINE UNDER TENSION BASIS. IN THE
C   EXPANSION OF THE FUNCTION, FOR I = 1,...,NX, J = 1,...,
C   NY, AND K = 1,...,NZ, THE COEFFICIENT MULTIPLYING THE
C   PRODUCT OF BASIS FUNCTION I IN X, BASIS FUNCTION J IN Y,
C   AND BASIS FUNCTION K IN Z IS STORED IN C(I,J,K).
C
C   VX, VY, AND VZ ARE ARRAYS OF LENGTH 5*NX, 5*NY, AND
C   5*NZ, RESPECTIVELY, CONTAINING THE B-SPLINE BASIS DATA
C   FOR THE X-, Y-, AND Z-GRIDS.
C
C   SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C   IB IS AN ARRAY OF AT LEAST 64 LOCATIONS.
C
C   BB IS AN ARRAY OF AT LEAST 64 LOCATIONS.
C
C THE PARAMETERS NX, NY, NZ, X, Y, Z, C, VX, VY, VZ, AND
C SIGMA SHOULD BE INPUT UNALTERED FROM THE OUTPUT OF
C VAL3B1.
C
C ON OUTPUT--
C
C   W CONTAINS THE INTERPOLATED FUNCTION VALUE.
C
C   WX, WY, AND WZ CONTAIN THE X-, Y-, AND Z-PARTIAL
C   DERIVATIVES, RESPECTIVELY.
C
C   WXX, WXY, WYY, WYZ, WZZ, AND WXZ CONTAIN THE XX-, XY-
C   YY-, YZ-, ZZ-, AND XZ-PARTIAL DERIVATIVES, RESPECTIVELY.
C
C   NB CONTAINS THE NUMBER OF THE B-SPLINE EXPANSION
C   COEFFICIENTS (SEE INPUT ARRAY C) INFLUENCING THE
C   FUNCTION VALUE AT THE GIVEN POINT.
C
C   IB CONTAINS NB INDICES IN ARRAY C OF THE B-SPLINE
C   EXPANSION COEFFICIENTS INFLUENCING THE FUNCTION VALUE AT
C   THE GIVEN POINT. THE ELEMENTS OF ARRAY C ARE INDEXED AS
C   IF IT WAS A ONE-DIMENSIONAL ARRAY.
C
C   BB CONTAINS NB VARIATIONS OF THE FUNCTION VALUE WITH
C   RESPECT TO THE B-SPLINE EXPANSION COEFFICIENTS SPECIFIED
C   IN ARRAY IB.
C
C AND
C
```

```
C   NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES DSPLNZ, INTRVL,
C AND SNHCSH.
C
C-----------------------------------------------------------------
C
      REAL BX(3,4),BY(3,4),BZ(3,4)
C
C EVALUATE BASIS FUNCTIONS AT XX, YY, AND ZZ
C
      CALL DSPLNZ (XX,NX,X,VX,SIGMA,ISTART,BX)
      CALL DSPLNZ (YY,NY,Y,VY,SIGMA,JSTART,BY)
      CALL DSPLNZ (ZZ,NZ,Z,VZ,SIGMA,KSTART,BZ)
C
C ACCUMULATE TENSOR PRODUCTS
C
      NB = 0
      SUM = 0.
      SUMX = 0.
      SUMY = 0.
      SUMZ = 0.
      SUMXX = 0.
      SUMXY = 0.
      SUMYY = 0.
      SUMYZ = 0.
      SUMZZ = 0.
      SUMXZ = 0.
      DO 3 K = 1,4
        KK = KSTART+K-1
        IF (KK .EQ. 0 .OR. KK .GT. NZ) GO TO 3
        BZ1K = BZ(1,K)
        BZ2K = BZ(2,K)
        BZ3K = BZ(3,K)
        DO 2 J = 1,4
          JJ = JSTART+J-1
          IF (JJ .EQ. 0 .OR. JJ .GT. NY) GO TO 2
          BY1J = BY(1,J)
          BY2J = BY(2,J)
          BY3J = BY(3,J)
          DO 1 I = 1,4
            II = ISTART+I-1
            IF (II .EQ. 0 .OR. II .GT. NX) GO TO 1
            BX1I = BX(1,I)
            BX2I =  BX(2,I)
            CIJK = C(II,JJ,KK)
            NB    = NB+1
            JB    = II+NX*(JJ-1+NY*(KK-1))
            IB(NB)= JB
            BB(NB)= BX1I*BY1J*BZ1K
            SUM = SUM+CIJK*BX1I*BY1J*BZ1K
            SUMX = SUMX+CIJK*BX2I*BY1J*BZ1K
            SUMY = SUMY+CIJK*BX1I*BY2J*BZ1K
            SUMZ = SUMZ+CIJK*BX1I*BY1J*BZ2K
            SUMXX = SUMXX+CIJK*BX(3,I)*BY1J*BZ1K
            SUMXY = SUMXY+CIJK*BX2I*BY2J*BZ1K
            SUMYY = SUMYY+CIJK*BX1I*BY3J*BZ1K
            SUMYZ = SUMYZ+CIJK*BX1I*BY2J*BZ2K
            SUMZZ = SUMZZ+CIJK*BX1I*BY1J*BZ3K
            SUMXZ = SUMXZ+CIJK*BX2I*BY1J*BZ2K
```

```
    1          CONTINUE
    2        CONTINUE
    3      CONTINUE
        W = SUM
        WX = SUMX
        WY = SUMY
        WZ = SUMZ
        WXX = SUMXX
        WXY = SUMXY
        WYY = SUMYY
        WYZ = SUMYZ
        WZZ = SUMZZ
        WXZ = SUMXZ
        RETURN
        END
C
C======================================================================
C
```

```
C SUBROUTINE PACKAGE 'CODE' - CODES FOR ELEMENTARY WAVES.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF:
C      GENERAL DESCRIPTION OF THE CODES FOR ELEMENTARY WAVES.
C      CODEB...BLOCK DATA SUBROUTINE DEFINING COMMON BLOCK /COD/ TO STORE
C              THE CODE OF THE COMPUTED ELEMENTARY WAVE.
C      CODE1...SUBROUTINE DESIGNED TO READ THE INPUT DATA FOR THE CODES
C              OF ELEMENTARY WAVES AND TO STORE THEM IN THE COMMON BLOCK
C              /COD/.
C      LCODE...INTEGER FUNCTION RETURNING THE LENGTH OF THE CODE OF THE
C              CURRENT ELEMENTARY WAVE.
C      NELEM...INTEGER FUNCTION RETURNING THE NUMBER OF RAY ELEMENTS
C              CORRESPONDING TO THE GIVEN POSITION IN THE CODE OF THE
C              ELEMENTARY WAVE.
C      CODE... SUBROUTINE DESIGNED TO TRANSFORM THE USED NUMERICAL CODE
C              OF THE ELEMENTARY WAVE UNDER CONSIDERATION INTO
C              INSTRUCTIONS SPECIFYING THE BEHAVIOUR OF THE WAVE AT THE
C              INITIAL POINTS OF RAYS AND AT ALL POINTS OF INCIDENCE OF
C              THE RAYS AT INTERFACES (BOUNDARIES OF COMPLEX BLOCKS).
C
C INPUT DATA FOR THE CODES OF ELEMENTARY WAVES:
C      THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C      THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C      THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C      THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C      I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C      INTEGER.  OTHERWISE (EXCEPT TEXTC), THE INPUT PARAMETER IS OF THE
C      TYPE REAL.
C (1)  TEXTC
C      STRING DESCRIBING THE DATA.  ONLY THE FIRST 80 CHARACTERS OF THE
C      STRING ARE SIGNIFICANT.
C (2)  NSKIP
C      THE NUMBER OF ELEMENTARY WAVE TO BE SKIPPED OVER.  THE
C      (NSKIP+1)-TH ELEMENTARY WAVE IN THESE INPUT DATA WILL BE THE FIRST
C      COMPUTED WAVE.  DEFAULT VALUE: 0.
C FOR EACH ELEMENTARY WAVE IWAVE THE FOLLOWING DATA (3):
C (3)  KODTYP,KODE
C      KODTYP..DETERMINES THE TYPE OF THE CODE
C              1... BLOCK-SURFACE CODE,
C              2... BASIC BLOCK CODE,
C              3... COMPOUND-ELEMENT BLOCK CODE,
C              4... GENERALIZED LAYER CODE,
C              5... TRANSMISSION-REFLECTION CODE.
C      KODE... THE CODE OF THE ELEMENTARY WAVE IN THE FORM OF A FINITE
C              SEQUENCE OF INTEGERS TERMINATED BY A SLASH.
C (4)  A SLASH.
C
C STORAGE IN THE MEMORY:
C      THE INPUT DATA (2) FOR THE COMPUTED ELEMENTARY WAVE ARE STORED IN
C      THE COMMON BLOCK /COD/ DEFINED IN THE FOLLOWING SUBROUTINE:
C      ----------------------------------------------------------------
      BLOCK DATA CODEB
      INTEGER MKODE
      PARAMETER (MKODE=128)
      INTEGER KODTYP,KODE(MKODE)
      COMMON /COD/ KODTYP,KODE
      SAVE /COD/
      END
```

```
C     -----------------------------------------------------------------------
C     KODTYP..DETERMINES THE TYPE OF THE CODE, SEE THE INPUT DATA (2).
C     KODE... ARRAY CONTAINING THE CODE OF THE ELEMENTARY WAVE.
C             THE CODE IS A FINITE SEQUENCE OF NONZERO INTEGERS.
C             ZERO INDICATES THE END OF A CODE.
C     COMMON BLOCK /COD/ IS INCLUDED IN EXTERNAL PROCEDURES COD1, CODE,
C     WRIT1,  AND MAY BE INCLUDED IN ANY OTHER SUBROUTINE. IF MKODE IS
C     CHANGED, IT MUST BE ADJUSTED IN ALL SUBROUTINES WHICH INCLUDE
C     COMMON BLOCK /COD/.
C
C DATE: 1989, NOVEMBER 19
C CODED BY LUDEK KLIMES
C
C=========================================================================
C
C CODES FOR ELEMENTARY WAVES - GENERAL DESCRIPTION
C
C     WE CONSIDER ORDINARY SEISMIC BODY WAVES, SUCH AS REFRACTED,
C     PRIMARILY OR MULTIPLY REFLECTED, POSSIBLY CONVERTED WAVES.  IN
C     GENERAL, INCIDENCE OF A WAVE AT AN INTERFACE (BOUNDARY OF A
C     COMPLEX BLOCK) PRODUCES FOUR WAVES, REFLECTED P AND S, AND
C     TRANSMITTED P AND S WAVES.  WHEN PERFORMING COMPLETE RAY TRACING,
C     WE MUST KNOW A PRIORI WHICH OF THE FOUR GENERATED WAVES TO FOLLOW.
C     THIS DECISION MUST BE MADE AT EACH INTERFACE.  THE ALPHANUMERIC
C     STRING SPECIFYING THE BEHAVIOUR OF A RAY FROM ITS INITIAL POINT TO
C     ITS ENDPOINT IS A 'CODE'.  IN THE FOLLOWING, WE SHALL CONSIDER THE
C     CODE TO BE A SEQUENCE OF NONZERO INTEGERS.
C
C     THE TERM 'ELEMENTARY WAVE' DOES NOT HAVE UNIQUE MEANING IN THE
C     LITERATURE.  HERE WE APPLY THE TERM ELEMENTARY WAVE TO THAT PART
C     OF THE WAVEFIELD THAT IS DESCRIBED BY ONE SPECIFIC CODE.  SINCE
C     THERE MAY BE VARIOUS TYPES OF CODES, THERE IS ALSO A VARIETY OF
C     DIVISIONS OF THE WAVE FIELD INTO ELEMENTARY WAVES.
C
C     WE INTRODUCE THE TERM 'ELEMENT OF A RAY', WHICH HAS AN IMPORTANT
C     MEANING IN THE CONSTRUCTION OF CODES.  BY AN ELEMENT OF A RAY, WE
C     DENOTE THAT PART OF THE RAY THAT IS SITUATED IN ONE COMPLEX BLOCK
C     BETWEEN TWO SUCCESSIVE POINTS OF REFLECTION/TRANSMISSION, OR
C     BETWEEN THE INITIAL POINT OR ENDPOINT OF THE RAY AND THE CLOSEST
C     POINT OF REFLECTION/TRANSMISSION, OR BETWEEN THE INITIAL POINT AND
C     THE ENDPOINT OF THE RAY, IF THE RAY IS ENTIRELY SITUATED IN ONE
C     COMPLEX BLOCK.
C
C     IN THE FOLLOWING, WE PRESENT SEVERAL POSSIBLE TYPES OF NUMERICAL
C     CODES OF ELEMENTARY WAVES.
C
C EXAMPLES OF CODE TYPES
C     (1) BLOCK-SURFACE CODE
C             TWO INTEGERS ARE USED FOR ANY ELEMENT OF THE RAY.  THE
C             ABSOLUTE VALUE OF THE FIRST INTEGER SPECIFIES THE INDEX OF
C             THE COMPLEX BLOCK, IN WHICH THE ELEMENT OF THE RAY IS
C             SITUATED, THE SIGN SPECIFIES THE TYPE OF THE WAVE (PLUS
C             SIGN...  P WAVE, MINUS SIGN...  S WAVE).  THE SECOND
C             NUMBER IS THE INDEX OF THE SMOOTH SURFACE, AT WHICH THE
C             ELEMENT TERMINATES.  THE CODE OF THE RAY IS A CHAIN OF THE
C             ABOVE DOUBLETS DESCRIBING THE RAY FROM ITS INITIAL POINT
C             TO ITS ENDPOINT.  THE INDEX OF A SURFACE, AT WHICH THE
C             LAST ELEMENT OF A RAY TERMINATES, NEED NOT BE SPECIFIED.
C             IN THIS CASE THE RAY IS ALLOWED TO TERMINATE ON ANY
C             SURFACE BOUNDING THE COMPLEX BLOCK IN WHICH THE LAST
```

```
C              ELEMENT OF THE RAY IS SITUATED.  APPLICATION OF THIS CODE
C              IS STRAIGHTFORWARD.
C
C              IF ANY COMPLEX BLOCK IS BOUNDED BY SEVERAL SMOOTH
C              SURFACES, THE BLOCK-SURFACE CODE MAY DIVIDE ARTIFICIALLY
C              THE WAVE FIELD INTO SEVERAL ELEMENTARY WAVES.  THIS WOULD
C              RESULT IN REPEATED APPLICATION OF THE COMPLETE RAY TRACING
C              TO ALL OF THE ELEMENTARY WAVES, WHAT MAY BE TIME
C              CONSUMING.  IN SUCH SITUATIONS, IT IS THEREFORE DESIRABLE
C              TO USE OTHER CODES, FOR WHICH THE ELEMENTARY WAVE HAS A
C              MORE GENERAL MEANING.
C
C        (2) BASIC BLOCK CODE
C              THIS CODE MAY BE OBTAINED FROM THE BLOCK-SURFACE CODE IF
C              THE INDICES OF THE SURFACES, AT WHICH INDIVIDUAL ELEMENTS
C              OF A RAY TERMINATE, ARE OMITTED.  THEN, ANY ELEMENT OF THE
C              RAY IS DESCRIBED BY A SINGLE INTEGER SPECIFYING THE
C              COMPLEX BLOCK AND WAVE TYPE, AND IT MAY TERMINATE AT ANY
C              SURFACE BOUNDING THE COMPLEX BLOCK.  THE CODE OF THE RAY
C              IS AGAIN A CHAIN OF THE NUMBERS DESCRIBING SUCCESIVELY ITS
C              INDIVIDUAL ELEMENTS.
C
C              ELEMENTARY WAVES SPECIFIED BY THE BLOCK-SURFACE CODE,
C              PASSING THROUGH THE SAME COMPLEX BLOCKS, AND REFLECTED
C              FROM DIFFERENT SURFACES BOUNDING A COMPLEX BLOCK, ARE
C              UNITED INTO ONE ELEMENTARY WAVE IN THE BASIC BLOCK CODE.
C              SIMILARLY, ELEMENTARY WAVES SPECIFIED BY THE BLOCK-SURFACE
C              CODE AND TRANSMITTED FROM A BLOCK INTO THE NEIGHBOURING
C              BLOCK ACROSS DIFFERENT SURFACES SEPARATING THESE BLOCKS,
C              ARE UNITED INTO ONE ELEMENTARY WAVE BY THE BASIC BLOCK
C              CODE.
C
C        (3) COMPOUND-ELEMENT BLOCK CODE
C              LET US INTRODUCE THE TERMS SIMPLE AND COMPOUND ELEMENT OF
C              A RAY.  FOR THIS PURPOSE, WE CALL THE 'LOWER-INDEX
C              BOUNDARY' OF A COMPLEX BLOCK THAT PART OF ITS BOUNDARY,
C              WHICH SEPARATES THE COMPLEX BLOCK EITHER FROM COMPLEX
C              BLOCKS WITH LOWER INDICES OR FROM A FREE SPACE.  THE
C              REMAINING PART OF THE BOUNDARY IS CALLED THE
C              'HIGHER-INDEX BOUNDARY' OF THE COMPLEX BLOCK.  THE INITIAL
C              POINT OF A RAY IS TREATED IN THE SAME WAY AS THE POINTS
C              SITUATED AT A LOWER-INDEX BOUNDARY.
C
C              AN ELEMENT OF A RAY IS CALLED 'SIMPLE ELEMENT' IF ITS
C              INITIAL POINT (E.G. THE POINT WHERE THE WAVE ENTERS THE
C              CORRESPONDING COMPLEX BLOCK) AND ITS ENDPOINT (E.G. THE
C              POINT AT WHICH THE WAVE LEAVES THE COMPLEX BLOCK) ARE
C              SITUATED ONE ON THE LOWER-INDEX AND THE OTHER ON THE
C              HIGHER-INDEX BOUNDARY OF THE COMPLEX BLOCK.  THE
C              'COMPOUND ELEMENT' IS SUCH AN ELEMENT FOR WHICH ITS
C              INITIAL POINT AND ITS ENDPOINT ARE BOTH SITUATED EITHER
C              ON THE LOWER-INDEX OR ON THE HIGHER-INDEX BOUNDARY OF THE
C              COMPLEX BLOCK.  THE COMPOUND ELEMENT IS FORMALLY
C              CONSIDERED AS TWO SIMPLE ELEMENTS.
C
C              OFTEN IT IS CONVENIENT TO WORK WITH WAVES, THE RAYS OF
C              WHICH HAVE EITHER A COMPOUND ELEMENT IN A COMPLEX BLOCK OR
C              TWO SIMPLE ELEMENTS OF THE SAME WAVE TYPE (UNCONVERTED
C              REFLECTION) IN THE SAME BLOCK, AS WITH ONE ELEMENTARY
C              WAVE.  THE COMPOUND-ELEMENT BLOCK CODE MAKES A DIVISION OF
```

```
C             THE WAVE FIELD INTO SUCH ELEMENTARY WAVES POSSIBLE.
C
C             WE INTRODUCE THE COMPOUND-ELEMENT BLOCK CODE AS FOLLOWS.
C             FOR ANY SIMPLE ELEMENT OF A RAY ONE NUMBER IS USED.  THE
C             ABSOLUTE VALUE OF THIS NUMBER SPECIFIES THE INDEX OF THE
C             COMPLEX BLOCK, IN WHICH THE SIMPLE ELEMENT IS SITUATED.
C             THE SIGN OF THIS NUMBER SPECIFIES THE TYPE OF THE WAVE
C             (PLUS SIGN...  P WAVE, MINUS SIGN...  S WAVE).  FOR ANY
C             COMPOUND ELEMENT OF A RAY, TWO IDENTICAL NUMBERS ARE USED.
C             A COMPOUND ELEMENT AND A DOUBLET OF SIMPLE ELEMENTS,
C             DESCRIBED BY THE SAME NUMBERS, ARE NOT DISTINGUISHED.  THE
C             CODE OF A RAY IS A CHAIN OF THESE NUMBERS DESCRIBING THE
C             RAY FROM ITS INITIAL POINT TO ITS ENDPOINT.  THE USE OF
C             THE COMPOUND-ELEMENT BLOCK CODE LEADS TO MORE GENERAL
C             ELEMENTARY WAVES.
C
C       (4) GENERALIZED LAYER CODE
C             ANOTHER CODE MAY BE OBTAINED FROM THE COMPOUND-ELEMENT
C             BLOCK CODE IF THE MODIFIED INTERPRETATION OF THE MODEL
C             STRUCTURE IS USED.  THE MODIFIED INTERPRETATION CONSISTS
C             IN ASSUMING AN EXISTENCE OF FICTITIOUS PARTS OF BLOCKS OF
C             A ZERO THICKNESS SITUATED BETWEEN EVERY NEIGHBOURING
C             COMPLEX BLOCKS, THE INDICES OF WHICH ARE NOT SEQUENTIALLY
C             ORDERED.  THE NUMBER OF FICTITIOUS BLOCKS IS CONSIDERED TO
C             BE JUST THE NUMBER, WHICH IS NECESSARY TO FILL THE GAP
C             BETWEEN INDICES OF THE TWO NEIGHBOURING COMPLEX BLOCKS.
C             THE RAY CAN ONLY PASS THROUGH THE FICTITIOUS BLOCKS, NO
C             REFLECTION IN FICTITIOUS BLOCKS BEING ALLOWED.  SIMILARLY,
C             NO CONVERSION IS ALLOWED NEITHER AT INTERFACES BETWEEN
C             FICTITIOUS BLOCKS NOR AT THE INTERFACE AT WHICH THE RAY
C             LEAVES A FICTITIOUS BLOCK (CONVERSION MAY TAKE PLACE ONLY
C             AT THE INTERFACE AT WHICH THE RAY ENTERS THE FICTITIOUS
C             BLOCK).  IF ANY OF THESE PROHIBITED SITUATIONS IS
C             SPECIFIED BY THE CODE, IT LEADS TO THE TERMINATION OF
C             COMPUTATIONS OF THE CORRESPONDING RAY.
C
C             THIS CODE IS A GENERALIZATION OF THE CODE USED IN THE 2-D
C             PROGRAM PACKAGE 'SEIS83' AND IS DESCRIBED IN CERVENY,
C             MOLOTKOV AND PSENCIK: RAY METHOD IN SEISMOLOGY, CHARLES
C             UNIVERSITY, PRAGUE 1977.  IT ENABLES ANY BLOCK STRUCTURE
C             TO BE INTERPRETED AS LOCALLY LAYERED STRUCTURE WITH
C             FICTITIOUS LAYERS.  THUS, THE ROUTINES FOR AUTOMATIC OR
C             SEMIAUTOMATIC GENERATION OF RAY CODES FOR LAYERED
C             STRUCTURES MAY BE USED FOR THIS CODE EVEN IN GENERAL BLOCK
C             STRUCTURES.  ITS EFFECTIVE USE IS CONDITIONED BY PROPER
C             INDEXING OF COMPLEX BLOCKS, WHICH MINIMIZES THE NUMBER OF
C             FICTITIOUS LAYERS.  FOR EXAMPLE, IN A LAYERED STRUCTURE,
C             THE BLOCKS (LAYERS) SHOULD BE INDEXED SEQUENTIALLY FROM
C             THE TOP TO THE BOTTOM.
C
C       (5) REFLECTION-TRANSMISSION CODE
C             FOR ANY ELEMENT OF A RAY ONE INTEGER IS USED.  THE FIRST
C             ELEMENT OF THE RAY, I.E.  THE ELEMENT CONTAINING THE
C             INITIAL POINT OF THE RAY, IS DENOTED BY 1 FOR P AND BY -1
C             FOR S WAVE.  ANY OTHER ELEMENT IS DENOTED BY 1 OR -1 (P OR
C             S WAVE) IF THE RAY IS TRANSMITTED AT THE INITIAL POINT OF
C             THE ELEMENT, AND BY 2 OR -2 (P OR S WAVE) IF THE RAY IS
C             REFLECTED AT THE INITIAL POINT OF THE ELEMENT.  THE CODE
C             OF A RAY IS A CHAIN OF THE ABOVE NUMBERS CORRESPONDING TO
C             INDIVIDUAL ELEMENTS OF THE RAY FROM ITS INITIAL POINT TO
```

```
C               ITS ENDPOINT.  IN A LAYERED MODEL, THIS CODE IS VERY
C               CONVENIENT FOR THE DESCRIPTION OF REFRACTED AND PRIMARY
C               REFLECTED WAVES.  THE APPLICATION OF THIS CODE IS
C               STRAIGHTFORWARD.
C
C SPECIFICATION OF AN ELEMENTARY WAVE
C     AN ELEMENTARY WAVE IS SPECIFIED BY THE FOLLOWING DATA:
C     (1) INTEGER KODTYP, WHICH DETERMINES THE TYPE OF THE CODE
C               1... BLOCK-SURFACE CODE,
C               2... BASIC BLOCK CODE,
C               3... COMPOUND-ELEMENT BLOCK CODE,
C               4... GENERALIZED LAYER CODE,
C               5... TRANSMISSION-REFLECTION CODE.
C     (2) INTEGER ARRAY KODE CONTAINING THE CODE OF THE ELEMENTARY
C               WAVE.  THE CODE IS A FINITE SEQUENCE OF NONZERO INTEGERS.
C               A ZERO INDICATES THE END OF A CODE IN THE INPUT DATA.
C     THE DATA (1) AND (2) SHOULD BE STORED IN COMMON BLOCK
C     --------------------------------
C     COMMON /COD/ KODTYP,KODE(MKODE).
C     --------------------------------
C     THE DIMENSION MKODE OF THE ARRAY KODE MAY BE ADJUSTED BY THE USER.
C
C DATE: 1988, JUNE 3
C WRITTEN BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C=======================================================================
C
      SUBROUTINE CODE1(LUN,IWAVE,IWAVE0,IKODE)
      INTEGER LUN,IWAVE,IWAVE0,IKODE
C
C THIS SUBROUTINE IS CALLED WHEN STARTING THE COMPUTATION OF A NEW
C ELEMENTARY WAVE.  IT STORES THE CODE OF THE ELEMENTARY WAVE INTO THE
C COMMON BLOCK /COD/.
C
C INPUT:
C     LUN...   LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C              CONTAINING THE INPUT DATA.
C     IWAVE...ZERO WHEN STARTING THE COMPLETE RAY TRACING PROGRAM,
C              OTHERWISE THE INDEX OF THE LAST ELEMENTARY WAVE WHICH HAS
C              BEEN COMPUTED (I.E. THE OUTPUT FROM THE PREVIOUS
C              INVOCATION OF THE SUBROUTINE CODE1).
C
C OUTPUT:
C     IWAVE...ZERO IF ALL REQUIRED ELEMENTARY WAVES ARE COMPUTED AND THE
C              COMPLETE RAY TRACING PROGRAM WILL BE TERMINATED.
C              OTHERWISE, THE INDEX OF THE ELEMENTARY WAVE WHICH WILL BE
C              COMPUTED (I.E. NSKIP+1 FOR THE FIRST COMPUTED ELEMENTARY
C              WAVE, WHERE NSKIP IS THE NUMBER OF ELEMENTARY WAVES HAVING
C              BEEN SKIPPED OVER, OTHERWISE THE INPUT VALUE INCREASED BY
C              ONE).
C     IWAVE0..INDEX OF THE ALREADY COMPUTED ELEMENTARY WAVE HAVING THE
C              MOST NUMEROUS COMMON ELEMENTS WITH THE CURRENT ELEMENTARY
C              WAVE.  IN THE CASE OF SEVERAL POSSIBILITIES, THE FIRST
C              COMPUTED WAVE OF THEM IS TAKEN.
C              NOTE: FOR FUTURE EXETENSION, IWAVE0=0 IN THIS VERSION.
C     IKODE...THE LENGTH OF THE COMMON PART OF THE CODES OF THE IWAVE-TH
C              AND IWAVE0-TH ELEMENTARY WAVES.
C              NOTE: FOR FUTURE EXETENSION, IKODE=0 IN THIS VERSION.
C
C COMMON BLOCK /COD/:
```

```
      INTEGER MKODE
      PARAMETER (MKODE=128)
      INTEGER KODTYP,KODE(MKODE)
      COMMON /COD/ KODTYP,KODE
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, NOVEMBER 14
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      CHARACTER*80 TEXTC
      INTEGER NSKIP,I
C
C     TEXTC...THE NAME OF THE DATA. STRING OF 80 CHARACTERS.
C     NSKIP...THE NUMBER OF ELEMENTARY WAVES TO BE SKIPPED OVER.
C     I...    AUXILIARY LOOP VARIABLE
C
      IF(IWAVE.EQ.0) THEN
C       READING THE NAME OF THE INPUT DATA
        READ(LUN,*) TEXTC
C
C       READING THE NUMBER OF ELEMENTARY WAVES TO BE SKIPPED OVER.
        NSKIP=0
        READ(LUN,*) NSKIP
        IWAVE=NSKIP
      END IF
C
C     READING THE CODE OF THE ELEMENTARY WAVE
      KODTYP=0
      DO 10 I=1,MKODE
        KODE(I)=0
   10 CONTINUE
      READ(LUN,*) KODTYP,KODE
      IF(KODTYP.EQ.0) THEN
        IWAVE=0
      ELSE
        IWAVE=IWAVE+1
      END IF
C
      IKODE=0
      IWAVE0=0
      RETURN
      END
C
C=======================================================================
C
      INTEGER FUNCTION LCODE()
C
C INTEGER FUNCTION KOOR IS DESIGNED TO RETURN THE LENGTH OF THE CODE OF
C THE CURRENT ELEMENTARY WAVE.
C
C NO INPUT.
C
C OUTPUT:
C     LCODE...LENGTH OF THE CODE OF THE CURRENT ELEMENTARY WAVE, I.E.
```

```
C              THE COUNT OF THE NUMERIC ITEMS IN THE ARRAY KODE WHICH
C              DESCRIBES THE BEHAVIOUR OF A RAY AT INTERFACES.  THE ARRAY
C              IS CALLED HERE THE CODE OF ELEMENTARY WAVE.
C
C COMMON BLOCK /COD/:
      INTEGER MKODE
      PARAMETER (MKODE=128)
      INTEGER KODTYP,KODE(MKODE)
      COMMON /COD/ KODTYP,KODE
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I
C
      DO 1 I=1,MKODE
        IF(KODE(I).EQ.0) THEN
          LCODE=I-1
          GO TO 2
        END IF
    1 CONTINUE
      LCODE=MKODE
    2 CONTINUE
      RETURN
      END
C
C=======================================================================
C
      INTEGER FUNCTION NELEM(KODIND)
      INTEGER KODIND
C
C THIS FUNCTION IS DESIGNED TO RETURN THE MAXIMUM POSSIBLE NUMBER OF RAY
C ELEMENTS CORRESPONDING TO THE GIVEN POSITION IN THE CODE OF THE
C ELEMENTARY WAVE.
C
C INPUT:
C     KODIND..POSITION IN THE CODE (INDEX IN ARRAY KODE)
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
C     NELEM...NUMBER OF RAY ELEMENTS BETWEEN THE INITIAL POINT OF THE
C             RAY AND THE END OF THE ELEMENT CORRESPONDING TO THE GIVEN
C             POSITION IN THE CODE.  NOTE THAT, FOR A PARTICULAR RAY,
C             SOME OF THESE POSSIBLE RAY ELEMENTS MAY BE LEFT OUT OR
C             COUPLED INTO ONE VIRTUAL ELEMENT.
C
C COMMON BLOCK /COD/:
      INTEGER MKODE
      PARAMETER (MKODE=128)
      INTEGER KODTYP,KODE(MKODE)
      COMMON /COD/ KODTYP,KODE
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
```

```
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C      NO AUXILIARY STORAGE LOCATIONS.
C
      IF(KODTYP.EQ.1) THEN
        NELEM=(KODIND+1)/2
      ELSE
        NELEM=KODIND
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE CODE(IY,KODIND,ICBNEW,IEND)
      INTEGER IY(12),KODIND,ICBNEW,IEND
C
C THIS SUBROUTINE TRANSFORMS THE USED NUMERICAL CODE OF THE ELEMENTARY
C WAVE UNDER CONSIDERATION INTO INSTRUCTIONS SPECIFYING THE BEHAVIOUR
C OF THE WAVE AT THE INITIAL POINTS OF RAYS AND AT ALL POINTS OF
C INCIDENCE OF THE RAYS AT INTERFACES (BOUNDARIES OF COMPLEX BLOCKS).
C THUS, IT IS FIRST CALLED AT THE INITIAL POINT OF A RAY AND THEN
C SUCCESSIVELY AT ALL POINTS OF INCIDENCE.
C
C INPUT:
C     IY...      INTEGER ARRAY. ITS NUMERICAL STORAGE UNITS 2, 3, 5, 6, 8
C                MUST BE DEFINED AS FOLLOWS:
C                IY(2)... POSITION IN THE CODE (INDEX IN ARRAY KODE)
C                  CORRESPONDING TO THE LAST COMPUTED ELEMENT OF A RAY.
C                  IY(2)=0 AT THE INITIAL POINT OF THE RAY,
C                  IY(2)=KODIND FROM THE LAST INVOCATION OF SUBROUTINE CODE
C                    AT OTHER POINTS OF THE RAY.
C                IY(3)=ICB0... INDEX OF THE NEIGHBOURING COMPLEX BLOCK,
C                  FROM WHICH THE RAY ENTERED THE COMPLEX BLOCK IN WHICH
C                  THE LAST COMPUTED ELEMENT OF THE RAY IS SITUATED.
C                  IY(3)=0 BEFORE LEAVING THE COMPLEX BLOCK, IN WHICH THE
C                  INITIAL POINT OF THE RAY IS SITUATED.
C                  AT THE INITIAL POINT OF THE RAY (IY(2)=0), IY(3) IS
C                  IGNORED.
C                IY(5)=ICB1... INDEX OF THE COMPLEX BLOCK CONTAINING THE
C                  COMPUTED ELEMENT OF THE RAY, SUPPLEMENTED BY THE SIGN
C                  '+' FOR P WAVE AND SIGN '-' FOR S WAVE.
C                  ICB1 IS IGNORED AT THE INITIAL POINT OF THE RAY
C                  (IY(2)=0).
C                IY(6)=ISRF... INDEX OF THE SURFACE AT WHICH THE ENDPOINT
C                  OF THE LAST COMPUTED ELEMENT OF THE RAY IS SITUATED.
C                  THE SIGN OF IY(6) IS IGNORED.
C                  IY(6)=0 AT THE INITIAL POINT OF THE RAY.
C                IY(8)=ICB2... INDEX OF THE COMPLEX BLOCK TOUCHING THE
C                  COMPLEX BLOCK ICB1 FROM THE OTHER SIDE OF THE SURFACE
C                  ISRF AT THE ENDPOINT OF THE LAST COMPUTED ELEMENT OF THE
C                  RAY. IY(8)=0 FOR A FREE SPACE.
C                  AT THE INITIAL POINT OF THE RAY, ICB2 IS THE INDEX OF
C                  THE COMPLEX BLOCK CONTAINING THE INITIAL POINT.
C THE INPUT PARAMETER IS NOT ALTERED.
C
```

```
C OUTPUT:
C     KODIND..POSITION IN THE CODE (INDEX IN THE ARRAY KODE) CORRES-
C             PONDING TO THE NEXT ELEMENT OF THE RAY.
C     ICBNEW..THE INDEX OF THE COMPLEX BLOCK IN WHICH THE NEXT
C             ELEMENT OF THE RAY IS TO BE SITUATED, SUPPLEMENTED
C             BY THE SIGN "+" FOR P WAVE OR "-" FOR S WAVE.
C     IEND... INFORMATION ON THE PROCESS OF THE INTERPRETATION
C             OF THE CODE:
C             IEND.EQ.0... COMPUTATION OF THE RAY CONTINUES.
C             IEND.NE.0... THE COMPUTATION OF THE RAY TERMINATES.
C               DIFFERENT VALUES OF IEND SPECIFY THE REASON FOR THE
C               TERMINATION:
C               10... RAY SATISFIES THE WHOLE CODE (REGULAR TERMINATION
C                 OF THE RAY).
C               21... THE POINT OF INCIDENCE IS SITUATED AT A DIFFERENT
C                 SURFACE THAN THAT REQUIRED BY THE CODE.
C               22... THE NEXT ELEMENT OF THE RAY IS REQUIRED BY THE
C                 CODE TO BE SITUATED IN A COMPLEX BLOCK THAT DOES NOT
C                 TOUCH THE POINT OF INCIDENCE.
C               23... TRANSMISSION IS REQUIRED BY THE CODE AT A FREE
C                 SURFACE.
C               30... REFLECTION OR WAVE CONVERSION AT THE FICTITIOUS
C                 PART OF THE INTERFACE.
C
C COMMON BLOCK /COD/:
C     ------------------------------------------------------------
      INTEGER MKODE
      PARAMETER (MKODE=128)
      INTEGER KODTYP,KODE(MKODE)
      COMMON /COD/ KODTYP,KODE
C     ------------------------------------------------------------
C     KODTYP..DETERMINES THE TYPE OF THE CODE:
C             1... BLOCK-SURFACE CODE,
C             2... BASIC BLOCK CODE,
C             3... COMPOUND-ELEMENT BLOCK CODE,
C             4... GENERALIZED LAYER CODE,
C             5... TRANSMISSION-REFLECTION CODE.
C     KODE... ARRAY CONTAINING THE CODE OF THE ELEMENTARY WAVE.
C             THE CODE IS A FINITE SEQUENCE OF NONZERO INTEGERS.
C             ZERO INDICATES THE END OF A CODE.
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1990, JANUARY 23
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,J
C
      KODIND=IY(2)
      GO TO (10,20,30,30,50),KODTYP
C
C     BLOCK-SURFACE CODE:
   10 CONTINUE
      IF(KODIND.GT.0) THEN
C       CHECK ON THE SURFACE:
        KODIND=KODIND+1
```

```
         IF(KODE(KODIND).EQ.0) THEN
C          END OF CODE
           IEND=10
           RETURN
         ELSE IF(KODE(KODIND).NE.IABS(IY(6))) THEN
C          WRONG SURFACE
           IEND=21
           RETURN
         END IF
       END IF
C      THE REST OF INTERPRETATION IS THE SAME AS IN THE BASIC BLOCK CODE
C
C      BASIC BLOCK CODE:
   20  CONTINUE
       KODIND=KODIND+1
       ICBNEW=KODE(KODIND)
       IF(ICBNEW.EQ.0) THEN
C        END OF CODE
         IEND=10
       ELSE IF(IABS(ICBNEW).EQ.IY(8)) THEN
C        TRANSMISSION
         IEND=0
       ELSE IF(KODIND.GT.1.AND.IABS(ICBNEW).EQ.IABS(IY(5))) THEN
C        REFLECTION
         IEND=0
       ELSE
C        REQUIRED COMPLEX BLOCK IS NOT ATTAINABLE
         IEND=22
       END IF
       RETURN
C
C      COMPOUND-ELEMENT BLOCK CODE:
   30  CONTINUE
       IF(KODIND.GT.0) THEN
         I=IABS(IY(5))
         J=(I-IABS(IY(3)))*(I-IY(8))
         DO 31 I=KODIND-1,1,-1
           IF(KODE(I).NE.KODE(KODIND)) GO TO 32
           J=-J
   31    CONTINUE
   32    CONTINUE
         IF(J.GT.0) THEN
C          COMPOUND ELEMENT:
           KODIND=KODIND+1
           IF(KODE(KODIND).NE.IY(5)) THEN
C            WRONG SURFACE
             IEND=21
             RETURN
           END IF
         END IF
       END IF
C      REST OF INTERPRETATION IS THE SAME AS IN BASIC BLOCK CODE
       IF(KODTYP.EQ.3) GO TO 20
C
C      GENERALIZED LAYER CODE:
C      INTERPRETATION OF COMPOUND ELEMENTS IS THE SAME AS IN
C      COMPOUND-ELEMENT BLOCK CODE, THEN:
       KODIND=KODIND+1
       ICBNEW=KODE(KODIND)
       IF(ICBNEW.EQ.0) THEN
```

```
C         END OF CODE
          IEND=10
        ELSE IF(KODNEW.EQ.1) THEN
C         INITIAL POINT OF A RAY:
          IF(IABS(ICBNEW).EQ.IY(8)) THEN
C           INITIAL POINT OF A RAY IS SITUATED IN THE REQUIRED C.B.
            IEND=0
          ELSE
C           INITIAL POINT OF A RAY IS NOT SITUATED IN THE REQUIRED C.B.
            IEND=22
          END IF
        ELSE IF(IABS(ICBNEW).EQ.IABS(IY(5))) THEN
C         REFLECTION
          IEND=0
        ELSE
C         POSSIBLE TRANSMISSION:
          J=ISIGN(1,IY(8)-IABS(IY(5)))
          IF(IABS(ICBNEW).EQ.IABS(IY(5))+J) THEN
C           LOOP FOR FICTITIOUS PARTS OF BLOCKS:
            DO 41 I=IABS(IY(5))+J+J,IY(8),J
              KODIND=KODIND+1
              ICBNEW=KODE(KODIND)
              IF(ICBNEW.EQ.0) THEN
C               END OF CODE IN THE FICTITIOUS PART OF BLOCK
                IEND=10
                RETURN
              ELSE IF(ICBNEW.EQ.ISIGN(I,KODE(KODIND-1))) THEN
C               TRANSMISSION FROM THE FICTITIOUS PART OF BLOCK
                IEND=0
              ELSE
C               TERMINATION OF THE RAY COMPUTATION IN THE FICTITIOUS PART
C               OF THE BLOCK
                IEND=30
                RETURN
              END IF
   41       CONTINUE
          ELSE
C           REQUIRED COMPLEX BLOCK IS NOT ATTAINABLE
            IEND=22
          END IF
        END IF
        RETURN
C
C       TRAMSMISSION-REFLECTION CODE:
   50 CONTINUE
        KODIND=KODIND+1
        I=KODE(KODIND)
        IF(IABS(I).EQ.1) THEN
C         TRANSMISSION:
          IF(IY(8).GT.0) THEN
C           TRANSMISSION INTO MATERIAL BLOCK
            ICBNEW=ISIGN(IY(8),I)
            IEND=0
          ELSE
C           TRANSMISSION INTO FREE SPACE
            IEND=23
          END IF
        ELSE IF(IABS(I).EQ.2.AND.KODIND.GT.1) THEN
C         REFLECTION
          ICBNEW=ISIGN(IABS(IY(5)),I)
```

```
          IEND=0
        ELSE
C         END OF CODE
          IEND=10
        END IF
        RETURN
        END
C
C=========================================================================
C
```

```
C PACKAGE 'RAY' FOR THE COMPLETE TRACING OF A RAY FROM A GIVEN POINT
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF:
C     RAYB... BLOCK DATA SUBROUTINE DEFINING COMMON BLOCKS /RAYT/ AND
C             /DCRT/ TO STORE THE INPUT DATA FOR COMPLETE RAY TRACING.
C     DESCRIPTION OF THE QUANTITIES COMPUTED ALONG A RAY (SEE
C             C.R.T.5.2).
C     RAY1... SUBROUTINE DESIGNED TO READ THE INPUT DATA FOR COMPLETE
C             RAY TRACING AND TO STORE THEM IN THE COMMON BLOCKS /RAYT/
C             AND /DCRT/ (SEE C.R.T.5.6).
C     RAY2... SUBROUTINE DESIGNED TO CONTINUE IN THE COMPLETE RAY
C             TRACING OF A RAY FROM THE GIVEN POINT (SEE C.R.T.5.7).
C
C INPUT DATA FOR COMPLETE RAY TRACING:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C     THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C     I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C     INTEGER.  OTHERWISE (EXCEPT TEXTR), THE INPUT PARAMETER IS OF THE
C     TYPE REAL.
C (1) TEXTR
C     STRING DESCRIBING THE DATA.  ONLY THE FIRST 80 CHARACTERS OF THE
C     STRING ARE SIGNIFICANT.
C (2) KSTORE,NEXPS,NHLF,MODCRT
C     QUANTITIES CONTROLING NUMERICAL INTEGRATION.
C     KSTORE..SPECIFIES WHETHER THE CONVERSION COEFFICIENTS ARE TO BE
C             CONSIDERED (SEE C.R.T.5.5.4 AND 5.6E):
C             KSTORE.LE.1... NO AMPLITUDE CONVERSION COEFFICIENTS ARE
C               APPLIED AT THE POINT OF INTERSECTION WITH AN INTERFACE.
C             KSTORE.GE.2... AMPLITUDE CONVERSION COEFFICIENTS ARE
C               APPLIED AT THE POINT OF INTERSECTION WITH AN INTERFACE.
C     NEXP... SPECIFIES INDEPENDENT VARIABLE ALONG RAYS, SEE
C             C.R.T.(5.1).
C     NHLF... MAXIMUM ALLOWED NUMBER OF HALVINGS (BISECTIONS) OF INITIAL
C             INCREMENT OF INDEPENDENT VARIABLE DURING NUMERICAL
C             INTEGRATION.
C     MODCRT..SPECIFIES THE MODE OF COMPLETE RAY TRACING, AND THE RAY
C             ELEMENTS ALONG WHICH THE COMPUTED QUANTITIES ARE STORED.
C             0...INITIAL MODE,
C               STORING THE COMPUTED QUANTITIES ALONG THE WHOLE RAYS,
C               NO STORING AT THE ENDPOINTS OF THE RAY ELEMENTS.
C             1...INITIAL MODE,
C               STORING THE QUANTITIES JUST ALONG NEW RAY ELEMENTS,
C               NO STORING AT THE ENDPOINTS OF THE RAY ELEMENTS.
C             2...INITIAL MODE,
C               STORING THE QUANTITIES JUST ALONG NEW RAY ELEMENTS,
C               STORING AT THE ENDPOINTS OF THE NEW RAY ELEMENTS.
C             3...INTERPOLATION MODE,
C               STORING THE QUANTITIES JUST ALONG NEW RAY ELEMENTS,
C               STORING AT THE ENDPOINTS OF THE NEW RAY ELEMENTS.
C (3) STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT
C     QUANTITIES CONTROLING NUMERICAL INTEGRATION.
C     STORE...STEP OF INDEPENDENT VARIABLE FOR STORING THE COMPUTED
C             QUANTITIES ALONG A RAY (C.R.T.5.5.1).  FOR STORE=0 THE
C             QUANTITIES ARE NOT STORED ALONG RAYS.
C     STEP... INITIAL INCREMENT OF INDEPENDENT VARIABLE FOR NUMERICAL
C             INTEGRATION.
```

```
C     UEB...  UPPER ERROR BOUND OF TRAVEL TIME PER ONE STEP OF NUMERICAL
C             INTEGRATION.  ERRORS IN THE COORDINATES OF POINTS ALONG
C             THE RAY ARE APPROXIMATELY TRANSFORMED TO UNITS OF TRAVEL
C             TIME AND ARE ALSO BOUNDED BY UEB.  THE ERROR PER STEP OF
C             NUMERICAL INTEGRATION IS AUTOMATICALLY KEPT WITHIN THE
C             LIMIT UEB IF THIS DOES NOT REQUIRE MORE THAN NHLF
C             BISECTIONS OF THE INITIAL INCREMENT STEP.  IN THE OPPOSITE
C             CASE THE UPPER ERROR BOUND IS 2,4,8... TIMES GREATER FOR
C             THE COMPUTATION OF THE REST OF THE RAY.  THUS THE
C             COMPUTATION OF EACH RAY IS COMPLETED.
C     UEBPP,UEBPH,UEBHH... MAXIMUM ALLOWED ACCUMULATED DEVIATIONS OF THE
C             TWO POLARIZATION VECTORS FROM THE CONDITIONS OF
C             ORTHONORMALITY (C.R.T.5.8.2D AND 5.8.3I).  THE ACCUMULATED
C             DEVIATIONS ARE EXPRESSED IN TIME UNITS.  IF ANY OF THE
C             ACCUMULATED DEVIATIONS OVERRFLOWS ITS UPPER BOUND, A
C             WARNING IS GENERATED.
C     UEBDRT..MAXIMUM ALLOWED DEVIATION OF L.H.S. AND R.H.S. IN
C             C.R.T.,EQ.(5.11).  IF THE DEVIATION OF ANY COMPONENT OF TH
C             4*4 MATRIX EXCEEDS UEBDRT, A WARNING IS GENERATED.
C (4) X1MIN,X1MAX,X2MIN,X2MAX,X3MIN,X3MAX,TMAX
C     THE BOUNDARIES OF THE COMPUTATIONAL VOLUME.
C     X1MIN,X1MAX,X2MIN,X2MAX,X3MIN,X3MAX... COORDINATES SPECIFYING THE
C             COORDINATE PLANES BOUNDING THE COMPUTATIONAL VOLUME.  THE
C             COORDINATE PLANES ARE INDEXED 101, 102, 103, 104, 105 AND
C             106.  DEFAULT VALUES ARE THE BOUNDARIES OF THE MODEL, SEE
C             SUBROUTINE MODEL1 AND THE INPUT DATA FOR THE MODEL.
C     TMAX... MAXIMUM TRAVEL TIME FOR THE COMPUTATION OF RAYS. THE
C             CORRESPONDING ISOCHRONE IS INDEXED BY 107.  DEFAULT VALUE
C             IS 999999.
C (5) NSRFCA
C     NUMBER OF AUXILIARY SURFACES, SEE C.R.T.5.3.  THE SURFACES ARE
C     INDEXED SEQUENTIALLY BY POSITIVE INTEGERS FROM NSRFC+1 TO
C     NSRFC+NSRFCA.  DEFAULT VALUE IS 0.
C (6) THE INDICES OF END SURFACES.
C     THE INDICES MAY BE SPECIFIED IN AN ARBITRARY ORDER AND MUST BE
C     TERMINATED BY A SLASH.  SEE C.R.T.5.4F.
C (7) THE INDICES OF SURFACES FOR STORING COMPUTED QUANTITIES.
C     THE INDICES OF SURFACES 1 TO NSRFC COVERING STRUCTURAL INTERFACES
C     MUST INCLUDE THE PROPER SIGN.  THE INDICES MAY BE SPECIFIED IN AN
C     ARBITRARY ORDER AND MUST BE TERMINATED BY A SLASH.
C     SEE C.R.T.5.5.2.
C (8) THE DATA SPECIFYING NSRFCA FUNCTIONS DESCRIBING THE AUXIIARY
C     SURFACES. THE DATA ARE READ BY SUBROUTINE SRFC1.  FOR THEIR
C     DESCRIPTION REFER TO SUBROUTINE SRFC1.
C
C STORAGE IN THE MEMORY:
C     THE INPUT DATA (1) TO (7) ARE STORED IN COMMON BLOCKS /RAYT/ AND
C     /DCRT/ DEFINED IN THE FOLLOWING SUBROUTINE:
C     ---------------------------------------------------------------
      BLOCK DATA RAYB
      CHARACTER*80 TEXTR
      COMMON/RAYT/TEXTR
      SAVE /RAYT/
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
```

```
     *                UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
      SAVE /DCRT/
      END
C     ------------------------------------------------------------------
C     TEXTR...THE NAME OF THE DATA. STRING OF 80 CHARACTERS.
C     KSTORE,NEXPS,NHLF,MODCRT... INPUT DATA (2).
C     STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT... INPUT DATA (3).
C     BOUNDR..BOUNDARIES X1MIN,X1MAX,X2MIN,X2MAX,X3MIN,X3MAX,TMAX OF THE
C             COMPUTATIONAL VOLUME, SEE INPUT DATA (4).
C     NSRFCA..NUMBER OF AUXILIARY SURFACES, SEE INPUT DATA (5).
C     NEND... NUMBER OF END SURFACES, SEE INPUT DATA (6).
C     KEND... CONTAINS THE INDICES OF END SURFACES, SEE INPUT DATA (6).
C     NSTOR...NUMBER OF SURFACES FOR STORING COMPUTED QUANTITIES, SEE
C             INPUT DATA (7).
C     KSTOR...CONTAINS THE INDICES OF SURFACES FOR STORING COMPUTED
C             QUANTITIES, SEE INPUT DATA (7).
C     COMMON BLOCK /DCRT/ IS ALSO INCLUDED IN SUBROUTINE PACKAGES
C     'RAYCB', 'INIT', 'WRIT' AND 'SCROPC'.  ALL THE INPUT DATA ARE
C     STORED SEQUENTIALLY IN THE SAME ORDER AS THEY WERE READ.  THE ONLY
C     EXCEPTION ARE LOCATIONS NEND AND NSTOR WHICH ARE INSERTED WHEN
C     READING THE INPUT DATA.  THE INDEX OF THE LAST ALLOCATED NUMERIC
C     UNIT OF ARRAY KEND IS NAMED MEND.  THE INDEX OF THE LAST ALLOCATED
C     NUMERIC UNIT OF ARRAY KSTOR IS NAMED MSTOR.  IF MEND OR MSTOR IS
C     CHANGED, IT MUST BE ADJUSTED IN ALL SUBROUTINES WHICH INCLUDE
C     COMMON BLOCK /DCRT/.
C
C DATE: 1990, JANUARY 23
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
C DESCRIPTION OF THE QUANTITIES COMPUTED ALONG A RAY (SEE C.R.T.5.2)
C
C LOCAL QUANTITIES (C.R.T.5.5.4):
C     YL(1)=VP... VELOCITY OF P-WAVES AT THE POINT.
C     YL(2)=VS... VELOCITY OF S-WAVES AT THE POINT.
C     YL(3)=RO... DENSITY AT THE POINT.
C     YL(4)=V1,YL(5)=V2,YL(6)=V3... VELOCITY DERIVATIVES IN GENERAL
C             COORDINATES.
C
C QUANTITIES COMPUTED ALONG A RAY (C.R.T.5.2):
C BASIC QUANTITIES COMPUTED ALONG A RAY (C.R.T.5.2.1):
C     Y(1)... TRAVEL TIME.
C     Y(2)... IMAGINARY PART OF THE COMPLEX-VALUED TRAVEL TIME.
C     Y(3),Y(4),Y(5)... COORDINATES OF POINTS ALONG THE RAY.
C     Y(6),Y(7),Y(8)... COVARIANT COMPONENTS OF THE SLOWNESS VECTOR.
C     Y(9),Y(10),Y(11)... COVARIANT COMPONENTS OF THE POLARIZATION
C             VECTOR E1 PERPENDICALAR TO THE RAY.
C     ( Y(12),Y(16),Y(20),Y(24) )    RAY PROPAGATOR MATRIX (I.E. THE
C     ( Y(13),Y(17),Y(21),Y(25) ) ...MATRIX OF FUNDAMENTAL SOLUTIONS
C     ( Y(14),Y(18),Y(22),Y(26) )    OF THE DYNAMIC RAY TRACING SYSTEM.
C     ( Y(15),Y(19),Y(23),Y(27) )
C     Y(28) TO Y(NY), WHERE NY=27+NAMPL... NAMPL REAL QUANTITIES
C             REPRESENTING COMPLEX-VALUED VECTORIAL REDUCED AMPLITUDES.
C             THE VECTORIAL REDUCED AMPLITUDES ARE SPECIFIED IN THE
C             RAY-CENTRED COORDINATE SYSTEM.
C             P WAVE AT THE INITIAL POINT OF THE RAY,
C             P WAVE AT THE POINT UNDER CONSIDERATION:
C               NAMPL=2,
C               Y(28)=REAL(A33), Y(29)=AIMAG(A33).
```

```
C              P WAVE AT THE INITIAL POINT OF THE RAY,
C              S WAVE AT THE POINT UNDER CONSIDERATION:
C                NAMPL=4,
C                Y(28)=REAL(A13), Y(29)=AIMAG(A13),
C                Y(30)=REAL(A23), Y(31)=AIMAG(A23).
C              S WAVE AT THE INITIAL POINT OF THE RAY,
C              P WAVE AT THE POINT UNDER CONSIDERATION:
C                NAMPL=4,
C                Y(28)=REAL(A31), Y(29)=AIMAG(A31),
C                Y(30)=REAL(A32), Y(31)=AIMAG(A32).
C              S WAVE AT THE INITIAL POINT OF THE RAY,
C              S WAVE AT THE POINT UNDER CONSIDERATION:
C                NAMPL=8,
C                Y(28)=REAL(A11), Y(29)=AIMAG(A11),
C                Y(30)=REAL(A21), Y(31)=AIMAG(A21),
C                Y(32)=REAL(A12), Y(33)=AIMAG(A12),
C                Y(34)=REAL(A22), Y(35)=AIMAG(A22).
C       Y(NY+1) TO Y(35), WHERE NY=27+NAMPL... UNDEFINED.
C AUXILIARY QUANTITIES COMPUTED ALONG A RAY (C.R.T.5.2.2):
C       YY(1)...INDEPENDENT VARIABLE ALONG A RAY.
C       YY(2)=UEBRAY... UPPER ERROR BOUND FOR RAY TRACING, WHICH IS EQUAL
C              TO THE INPUT VALUE UEB AT THE INITIAL POINT OF THE RAY
C              (C.R.T.5.6J).  IT IS ALWAYS DOUBLED WHEN THE NUMERICAL
C              INTEGRATION REQUIRES MORE THAN NHLF BISECTIONS OF THE
C              INITIAL INCREMENT STEP (C.R.T.5.6G).  UEBRAY.GT.UEB AT
C              THE ENDPOINT OF THE RAY INDICATES A DECREASED ACCURACY
C              OF COMPUTATION.
C       YY(3)=ERRPP,YY(4)=ERRPH,YY(5)=ERRHH... DEVIATIONS (IN ABSOLUTE
C              VALUES OF THE TWO COMPUTED BASIS VECTORS OF THE
C              RAY-CENTRED COORDINATE SYSTEM FROM THE CONDITIONS OF
C              ORTHONORMALITY (C.R.T.5.8.2D AND 5.8.3I), ACCUMULATED
C              ALONG THE RAY.  ANY OF THEM MAY BE COMPARED WITH THE
C              CORRESPONDING SPECIFIED LIMIT UEBPP, UEBPH, UEBHH AT THE
C              ENDPOINT OF THE RAY.
C       IY(1)=NY=27+NAMPL... NUMBER OF THE BASIC QUANTITIES
C              DESCRIBING THE POINT OF A RAY, SEE ABOVE.
C       IY(2)=KODIND... POSITION IN THE CODE (INDEX IN ARRAY KODE)
C              CORRESPONDING TO THE CONSIDERED ELEMENT OF A RAY. ITS
C              VALUE IS DETERMINED BY SUBROUTINE CODE (C.R.T.4.3).
C       IY(3)=ICB0... INDEX OF THE COMPLEX BLOCK, FROM WHICH THE RAY
C              ENTERED THE COMPLEX BLOCK IN WHICH THE COMPUTED
C              ELEMENT OF THE RAY IS SITUATED.
C              IY(3)=0 BEFORE LEAVING THE COMPLEX BLOCK, IN WHICH THE
C              INITIAL POINT OF THE RAY IS SITUATED.
C       IY(4)=ISB1... INDEX OF THE SIMPLE BLOCK CONTAINING THE
C              COMPUTED ELEMENT OF THE RAY.
C       IY(5)=ICB1... INDEX OF THE COMPLEX BLOCK CONTAINING THE
C              COMPUTED ELEMENT OF THE RAY, SUPPLEMENTED BY A SIGN '+'
C              FOR P WAVE AND SIGN '-' FOR S WAVE.
C       IY(6)=ISRF... INDEX OF THE SURFACE AT WHICH THE ENDPOINT OF THE
C              COMPUTED ELEMENT OF THE RAY IS SITUATED, SUPPLEMENTED BY
C              A SIGN '+' OR '-' FOR THE ENDPOINT SITUATED AT THE
C              POSITIVE OR NEGATIVE SIDE OF THE SURFACE, RESPECTIVELY.
C              ZERO INSIDE THE COMPLEX BLOCK.  NOTE: THE SIGN OF THIS
C              QUANTITY IS THE EXEPTION TO THE DEFINITION IN THE ORIGINAL
C              PAPER ON C.R.T.
C       IY(7)=ISB2... INDEX OF THE SIMPLE BLOCK TOUCHING THE COMPLEX
C              BLOCK ICB1 FROM THE OTHER SIDE OF THE SURFACE ISRF AT
C              THE ENDPOINT OF THE COMPUTED ELEMENT OF THE RAY.
C              ISB2=0 FOR A FREE SPACE ON THE OTHER SIDE OF ISRF.
```

```
C               UNDEFINED INSIDE THE COMPLEX BLOCK, DEFINED ONLY AT
C               THE ENDPOINT OF THE ELEMENT OF THE RAY.
C       IY(8)=ICB2... INDEX OF THE COMPLEX BLOCK TOUCHING THE COMPLEX
C               BLOCK ICB1 FROM THE OTHER SIDE OF THE SURFACE ISRF AT
C               THE ENDPOINT OF THE COMPUTED ELEMENT OF THE RAY.
C               ICB2=0 FOR A FREE SPACE ON THE OTHER SIDE OF ISRF.
C               UNDEFINED INSIDE THE COMPLEX BLOCK, DEFINED ONLY AT
C               THE ENDPOINT OF THE ELEMENT OF THE RAY.
C       IY(9)=IFCT... NUMBER OF INVOCATIONS OF SUBROUTINE FCT EVALUATING
C               THE RIGHT-HAND SIDES OF THE ORDINARY DIFFERENTIAL
C               EQUATIONS, ALONG THE COMPUTED PART OF THE RAY.
C       IY(10)=IOUTP... NUMBER OF SUCCESSFUL STEPS OF THE NUMERICAL
C               INTEGRATION, ALONG THE RAY (I.E. THE NUMBER OF INVOCATIONS
C               OF SUBROUTINE OUTP DECREASED BY THE NUMBER OF INVOCATIONS
C               OF SUBROUTINE HPCG).
C       IY(11)=ITRANS... NUMBER OF TRANSFORMATIONS AT AN INTERFACE (I.E.
C               THE NUMBER OF INVOCATIONS OF SUBROUTINE TRANS).
C       IY(12)=KMAH... NUMBER OF CAUSTIC POINTS ALONG THE RAY (THE INDEX
C               OF THE RAY TRAJECTORY).
C
C DATE: 1990, JANUARY 23
C WRITTEN BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C=======================================================================
C
      SUBROUTINE RAY1(LUN)
      INTEGER LUN
C
C SUBROUTINE RAY1 READS THE INPUT DATA FOR COMPLETE RAY TRACING (SEE
C C.R.T.5.6) AND STORES THEM IN COMMON BLOCKS /RAYT/ AND /DCRT/.  IT IS
C CALLED ONCE, BEFORE THE COMPLETE TRACING OF INDIVIDUAL RAYS BEGINS.
C SUBROUTINE RAY1 CALLS SUBROUTINE SRFC1 IN ORDER TO READ THE INPUT DATA
C FOR THE AUXILIARY SURFACES, AND TO STORE THEM IN THE MEMORY.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C THE INPUT PARAMETER IS NOT ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK /MODELC/ (SUBROUTINE PACKAGE 'MODEL'):
      INTEGER MSB,MCB
      PARAMETER (MSB=128)
      PARAMETER (MCB=128)
      INTEGER NEXPV,NEXPQ
      REAL BOUNDM(6)
      INTEGER NSRFCS,NSB,KSB(0:MSB),NCB,KCB(0:MSB)
      EQUIVALENCE (KSB(0),NSB),(KCB(0),NCB)
      COMMON/MODELC/NEXPV,NEXPQ,BOUNDM,NSRFCS,KSB,KCB
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /RAYT/:
      CHARACTER*80 TEXTR
      COMMON/RAYT/TEXTR
C COMMON BLOCK /DCRT/:
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
```

```
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCKS ARE DEFINED IN THIS
C SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL NSRFC,SRFC1
      INTEGER NSRFC
C     FUNCTION NSRFC()... SUBROUTINE PACKAGE 'MODEL'.
C     SRFC1 AND SUBSEQUENT ROUTINES... SUBROUTINE PACKAGE 'SRFC'.
C
C ERROR MESSAGES:
C     561...  INSUFFICIENT MEMORY IN /DCRT/:
C             INSUFFICIENT MEMORY FOR THE INPUT DATA IN COMMON BLOCK
C             /DCRT/.  THE DIMENSION MEND OF ARRAY KEND MUST BE
C             ENLARGED.  SEE THE BLOCK DATA SUBROUTINE RAYB.
C     562...  INSUFFICIENT MEMORY IN /DCRT/:
C             INSUFFICIENT MEMORY FOR THE INPUT DATA IN COMMON BLOCK
C             /DCRT/.  THE DIMENSION MSTOR OF ARRAY KSTOR MUST BE
C             ENLARGED.  SEE THE BLOCK DATA SUBROUTINE RAYB.
C     563...  INADMISSIBLE END SURFACE:
C             THE INDEX OF AN END SURFACE IN INPUT DATA IS NEGATIVE OR
C             GREATER THAN THE NUMBER OF SPECIFIED SURFACES.
C     564...  INADMISSIBLE STORE SURFACE:
C             THE INDEX OF A SURFACE FOR STORING THE COMPUTED QUANTITIES
C             IS GREATER THAN THE NUMBER OF SPECIFIED SURFACES, OR THE
C             INDEX OF AN AUXILIARY SURFACE FOR STORING IS NEGATIVE.
C
C DATE: 1990, JANUARY 23
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I
C
C     (1) STRING DESCRIBING THE DATA
      READ(LUN,*) TEXTR
C     (2,3) QUANTITIES CONTROLING NUMERICAL INTEGRATION
      READ(LUN,*) KSTORE,NEXPS,NHLF
      READ(LUN,*) STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT
C     (4) COORDINATE PLANES AND ISOCHRONE BOUNDING THE COMPUTATIONAL
C     VOLUME
      DO 10 I=1,6
        BOUNDR(I)=BOUNDM(I)
   10 CONTINUE
      BOUNDR(7)=999999.
      READ(LUN,*) BOUNDR
C     (5) NUMBER OF AUXILIARY SURFACES
      NSRFCA=0
      READ(LUN,*) NSRFCA
C
C     (6) END SURFACES:
C     INITIALIZING MEMORY FOR INDICES OF END SURFACES
      DO 11 I=1,MEND
        KEND(I)=0
   11 CONTINUE
C     READING INDICES OF END SURFACES:
```

```
      READ(LUN,*) (KEND(I),I=1,MEND)
      DO 12 I=1,MEND
        IF(KEND(I).LT.0.OR.KEND(I).GT.NSRFC()+NSRFCA) THEN
          STOP 'ERROR 563 IN RAY1: INADMISSIBLE END SURFACE'
        ELSE IF(KEND(I).EQ.0) THEN
          NEND=I-1
          GO TO 13
        END IF
   12 CONTINUE
      STOP 'ERROR 561 IN RAY1: INSUFFICIENT MEMORY IN /DCRT/'
   13 CONTINUE
C
C     (7) SURFACES FOR STORING COMPUTED QUANTITIES:
C     INITIALIZING MEMORY FOR INDICES OF THE SURFACES
      DO 21 I=1,MSTOR
        KSTOR(I)=0
   21 CONTINUE
C     READING INDICES OF THE SURFACES:
      READ(LUN,*) (KSTOR(I),I=1,MSTOR)
      DO 22 I=1,MSTOR
        IF(KSTOR(I).LT.-NSRFC().OR.KSTOR(I).GT.NSRFC()+NSRFCA) THEN
          STOP 'ERROR 564 IN RAY1: INADMISSIBLE STORE SURFACE'
        ELSE IF(KSTOR(I).EQ.0) THEN
          NSTOR=I-1
          GO TO 23
        END IF
   22 CONTINUE
      STOP 'ERROR 562 IN RAY1: INSUFFICIENT MEMORY IN /DCRT/'
   23 CONTINUE
C
C     AUXILIARY SURFACES:
      IF(NSRFCA.GT.0) THEN
        CALL SRFC1(LUN,NSRFCA)
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE RAY2(YL,Y,YY,IY,IEND)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12),IEND
C
C THIS SUBROUTINE CONTINUES IN THE COMPLETE TRACING OF A RAY FROM THE
C GIVEN POINT (SEE C.R.T.5.7).  FOR MODCRT.LE.2 (SEE THE INPUT DATA
C (2)), THE WHOLE RAY IS COMPUTED.  FOR MODCRT.GE.3, JUST ONE RAY
C ELEMENT IS COMPUTED.
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE GIVEN POINT.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY
C             AT THE GIVEN POINT.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY AT THE GIVEN POINT.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY AT THE GIVEN POINT.
C
C OUTPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE ENDPOINT OF THE
C             RAY.
```

```
C       Y...     ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY
C                AT THE ENDPOINT OF THE RAY.
C       YY...    ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C                THE RAY AT THE ENDPOINT OF THE RAY.
C       IY...    ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C                ALONG THE RAY AT THE ENDPOINT OF THE RAY.
C       IEND...  REASON OF THE TERMINATION OF THE COMPUTATION OF A RAY (SEE
C                  C.R.T.5.4).  IEND=10,21,22,23,30 ARE INDICATED IN
C                  SUBROUTINE CODE.  IEND=24,25,26,32,33 ARE INDICATED IN
C                  SUBROUTINE TRANS.  THE REASONS IEND=40,50,60 ARE CHECKED
C                  AND INDICATED BY IY(6) IN SUBROUTINE RAYCB.  IEND=61 IS
C                  INDICATED IN THIS SUBROUTINE RAY2 USING IY(6).
C                0...  THE COMPUTATION OF THE RAY MAY CONTINUE.
C                10... RAY SATISFIES THE WHOLE CODE.
C                21... THE POINT OF INCIDENCE IS SITUATED AT A DIFFERENT
C                  SURFACE THAN THAT SPECIFIED BY THE CODE.
C                22... THE NEXT ELEMENT OF THE RAY IS REQUIRED BY THE CODE
C                  TO BE SITUATED IN A COMPLEX BLOCK THAT DOES NOT TOUCH
C                  THE POINT OF INCIDENCE.
C                23,24... TRANSMISSION IS REQUIRED BY THE CODE AT A FREE
C                  SURFACE.
C                25... RAY OF THE REQUIRED REFLECTED OR TRANSMITTED WAVE
C                  IS NOT REAL-VALUED (OVERCRITICAL REFLECTION OR
C                  TRANSMISSION).
C                26... S WAVE IN A LIQUID BLOCK IS REQUIRED BY THE CODE.
C                30,32... REFLECTION OR TYPE CONVERSION AT THE FICTITIOUS
C                  PART OF THE INTERFACE.
C                33... ZERO REFLECTION/TRANSMISSION COEFFICIENT.
C                40... TRAVEL TIME GREATER THAN THE SPECIFIED LIMIT.
C                50... RAY HAS INTERSECTED A COORDINATE PLANE LIMITING THE
C                  COMPUTATIONAL VOLUME FOR COMPLETE RAY TRACING.
C                60,61... RAY HAS INTERSECTED A SMOOTH SURFACE LIMITING THE
C                  COMPUTATIONAL VOLUME FOR COMPLETE RAY TRACING.
C                OTHER VALUES OF IEND MAY INDICATE THE PROGRAM FAILURE.
C
C COMMON BLOCK /DCRT/:
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL NSRFC,KOOR,CODE,RAYCB,TRANS,CONV,RPAR31,RPAR32,RPAR33
      EXTERNAL WRIT31,WRIT32,WRIT33
      INTEGER NSRFC,KOOR
C     FUNCTION NSRFC()... SUBROUTINE PACKAGE 'MODEL'.
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)... SUBROUTINE PACKAGE
C             'METRIC'.
C     FUNCTION KOOR()... SUBROUTINE PACKAGE 'METRIC'.
C     SUBROUTINE CODE(IY,KODIND,ICBNEW,IEND) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'CODE'.
C     SUBROUTINE RAYCB(YL,Y,YY,IY) AND SUBSEQUENT ROUTINES... SUBROUTINE
C             PACKAGE 'RAYCB'.
C     SUBROUTINE HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)... IBM
C             SCIENTIFIC SUBROUTINE PACKAGE.
```

```
C      SUBROUTINE TRANS(YL,Y,YY,IY,KODIND,ICBNEW,IEND) AND SUBSEQUENT
C              ROUTINES... SUBROUTINE PACKAGE 'TRANS'.
C      SUBROUTINE CONV(KSTORE,YL,Y,YY,IY,NAMPL,YC)... SAMPLE SUBROUTINE
C              PACKAGE 'TRANS'.
C      SUBROUTINE RPAR31(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'RPAR'.
C      SUBROUTINE RPAR32(ISTOR,YL,Y,YY,IY,NAMPL,YC)... SAMPLE SUBROUTINE
C              PACKAGE 'RPAR'.
C      SUBROUTINE RPAR33(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'RPAR'.
C      SUBROUTINE WRIT31(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'WRIT'.
C      SUBROUTINE WRIT32(ISTOR,YL,Y,YY,IY,NAMPL,YC)... SAMPLE SUBROUTINE
C              PACKAGE 'WRIT'.
C      SUBROUTINE WRIT33(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'WRIT'.
C      SUBROUTINE BLOCK(COOR,ISRF1,ISB1,ISRF2,ISB2,ICB2)... SUBROUTINE
C              PACKAGE 'MODEL'.
C      SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)... SUBROUTINE
C              PACKAGE 'MODEL'.
C      SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C              SUBROUTINE PACKAGE 'SRFC'.
C      SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C              ... SUBROUTINE PACKAGE 'PARM'.
C
C ERROR MESSAGES:
C      570...  WRONG FUNCTION OF SUBROUTINE CODE:
C              REASON 37 OR 38 OF THE TERMINATION OF THE RAY COMPUTATION
C              IS REPORTED BY THE SUBROUTINE TRANS, WHILE IT SHOULD BE
C              REPORTED PREVIOUSLY BY THE SUBROUTINE CODE AS THE REASON
C              22 OR 23, RESPECTIVELY.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C      AUXILIARY STORAGE LOCATIONS:
       INTEGER KODIND,ICBNEW,I,NAMPL
       REAL YC(12)
C
C      KODIND..POSITION IN THE CODE (INDEX IN THE ARRAY KODE) CORRES-
C              PONDING TO THE NEXT ELEMENT OF THE RAY.
C      ICBNEW..THE INDEX OF THE COMPLEX BLOCK IN WHICH THE NEXT
C              ELEMENT OF THE RAY IS TO BE SITUATED, SUPPLEMENTED
C              BY THE SIGN "+" FOR P WAVE OR "-" FOR S WAVE.
C      I...    AUXILIARY LOOP VARIABLE.
C      NAMPL...NUMBER OF REAL QUANTITIES REPRESENTING COMPLEX-VALUED
C              VECTORIAL REDUCED AMPLITUDES.  SEE THE SUBROUTINE CONV OF
C              THE SUBROUTINE PACKAGE 'TRANS'.
C      YC...   ARRAY CONTAINING REAL QUANTITIES REPRESENTING COMPLEX-
C              -VALUED VECTORIAL REDUCED AMPLITUDES.  SEE C.R.T.5.5.4 AND
C              THE SUBROUTINE CONV OF THE SUBROUTINE PACKAGE 'TRANS'.
C
C.......................................................................
C
   10 CONTINUE
      IF(IY(6).EQ.0) THEN
C
C        (A1) ELEMENT OF THE RAY INSIDE A COMPLEX BLOCK
         CALL RAYCB(YL,Y,YY,IY)
C
C        (B1) STORAGE OF THE COMPUTED QUANTITIES AT THE INTERFACE
         IF(STORE.NE.0) THEN
```

```
            CALL RPAR31(YL,Y,YY,IY)
            CALL WRIT31(YL,Y,YY,IY)
          END IF
          DO 21 I=1,NSTOR
            IF(KSTOR(I).EQ.IY(6)) THEN
              CALL CONV(KSTORE,YL,Y,IY,NAMPL,YC)
              CALL RPAR32(I,YL,Y,YY,IY,NAMPL,YC)
              CALL WRIT32(I,YL,Y,YY,IY,NAMPL,YC)
            END IF
   21     CONTINUE
          IF(IABS(IY(6)).LE.NSRFC()) THEN
            CALL RPAR33(YL,Y,YY,IY)
            CALL WRIT33(YL,Y,YY,IY)
          END IF
C
C         (A2) CHECK FOR THE BOUNDARY OF THE COMPUTATIONAL VOLUME
          IF(IABS(IY(6)).GE.107) THEN
            IEND=40
            GO TO 99
          ELSE IF(IABS(IY(6)).GE.101) THEN
            IEND=50
            GO TO 99
          ELSE IF(IABS(IY(6)).GT.NSRFC()) THEN
            IEND=60
            GO TO 99
          ELSE IF(MODCRT.GE.3) THEN
            IEND=0
            GO TO 99
          END IF
C
        ELSE
C
C         (B2) INTERPRETATION OF THE ELEMENTARY WAVES CODE
          CALL CODE(IY,KODIND,ICBNEW,IEND)
          IF(IEND.NE.0) THEN
            GO TO 99
          END IF
C
C         (B3) CHECK FOR THE END SURFACES
C         NOTE: REFLECTION FROM AN END SURFACE IS ALLOWED.
          IF(IABS(IY(5)).NE.IABS(ICBNEW)) THEN
            DO 23 I=1,NEND
              IF(IABS(KEND(I)).EQ.IABS(IY(6))) THEN
                IEND=61
                GO TO 99
              END IF
   23       CONTINUE
          END IF
C
C         (B4) TRANSFORMATION ACROSS THE INTERFACE
          CALL TRANS(YL,Y,YY,IY,KODIND,ICBNEW,IEND)
          IF(IEND.GE.37) THEN
            STOP 'ERROR 570 IN RAY2: WRONG FUNCTION OF SUBROUTINE CODE'
          ELSE IF(IEND.NE.0) THEN
            GO TO 99
          END IF
C
C         (B5) STORAGE OF THE COMPUTED QUANTITIES AT THE INTERFACE
          IF(STORE.NE.0) THEN
            CALL RPAR31(YL,Y,YY,IY)
```

```
            CALL WRIT31(YL,Y,YY,IY)
          END IF
          DO 25 I=1,NSTOR
            IF(KSTOR(I).EQ.IY(6)) THEN
              CALL CONV(KSTORE,YL,Y,IY,NAMPL,YC)
              CALL RPAR32(I,YL,Y,YY,IY,NAMPL,YC)
              CALL WRIT32(I,YL,Y,YY,IY,NAMPL,YC)
            END IF
   25     CONTINUE
C
C         (B6) THE RAY MAY CONTINUE IN THE NEXT COMPLEX BLOCK
          IY(6)=0
C
        END IF
        GO TO 10
C
   99   CONTINUE
        RETURN
        END
C
C======================================================================
C
```

```
C SUBROUTINE PACKAGE 'RAYCB' FOR COMPLETE RAY TRACING WITHIN ONE COMPLEX
C BLOCK
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C     RAYCB...SUBROUTINE TRANSFERING THE QUANTITIES GIVEN AT THE INITIAL
C             POINT OF THE ELEMENT OF A RAY INTO THE CORRESPONDING
C             QUANTITIES AT THE ENDPOINT OF THE ELEMENT (SEE
C             C.R.T.5.8.1).
C     FCT...  SUBROUTINE EVALUATING THE RIGHT-HAND SIDES OF THE SYSTEM
C             OF ORDINARY DIFFERENTIAL EQUATIONS FOR Y(1) TO Y(27) (SEE
C             C.R.T.5.8.2).
C     OUTP... SUBROUTINE CONTAINING VARIOUS TESTS OF THE POSITION OF THE
C             NEWLY COMPUTED POINT OF THE RAY, TESTS FOR POSSIBLE
C             CAUSTIC POINTS, ETC. (SEE C.R.T.5.8.3).
C     SMVPRD..AUXILIARY SUBROUTINE TO FCT, EVALUATES SYMMETRIC MATRIX BY
C             VECTOR PRODUCT.
C     PSHIFT..AUXILIARY SUBROUTINE TO OUTP. IT CORRECTS REDUCED
C             AMPLITUDES WITH REGARD TO PHASE SHIFT DUE TO CAUSTICS (SEE
C             C.R.T.5.8.3F).
C     CROSS...SUBROUTINE DESIGNED TO FIND THE POINT OF INTERSECTION OF A
C             CURVE WITH A SURFACE (SEE C.R.T.5.8.4B).
C     HIVD2...SUBROUTINE PERFORMING THE HERMITE INTERPOLATION OF A
C             VECTOR AND ITS DERIVATIVES USING FUNCTIONAL VALUES AND
C             DERIVATIVES AT 2 GIVEN POINTS (SEE C.R.T.5.8.4A).
C
C DATE: 1989, DECEMBER 6
C
C=======================================================================
C
      SUBROUTINE RAYCB(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SUBROUTINE TRANSFERS THE QUANTITIES GIVEN AT THE INITIAL POINT OF
C THE ELEMENT OF A RAY INTO THE CORRESPONDING QUANTITIES AT THE ENDPOINT
C OF THE ELEMENT (SEE C.R.T.5.8.1).
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE INITIAL POINT OF
C             THE ELEMENT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY
C             AT THE INITIAL POINT OF THE ELEMENT OF THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY AT THE INITIAL POINT OF THE ELEMENT OF THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY AT THE INITIAL POINT OF THE ELEMENT OF THE
C             RAY.
C
C OUTPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE ENDPOINT OF THE
C             ELEMENT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY
C             AT THE ENDPOINT OF THE ELEMENT OF THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY AT THE ENDPOINT OF THE ELEMENT OF THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY AT THE ENDPOINT OF THE ELEMENT OF THE RAY.
C
```

```
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /INITC/ (SEE SUBROUTINE PACKAGE 'INIT') IS REQUIRED IN
C     SUBROUTINE OUTP.  NONE OF THE STORAGE LOCATIONS OF THE COMMON
C     BLOCK ARE ALTERED THERE.
C
C COMMON BLOCK /RAYC/ -  COMMUNICATION BETWEEN RAYCB, FCT, AND OUTP:
C     -----------------------------------------------------------------
      REAL YLRC(6),YYRC(5),DELT11,DELT13,DELT33,VNEXPS
      INTEGER IYRC(12)
      COMMON/RAYC/YLRC,YYRC,IYRC,DELT11,DELT13,DELT33,VNEXPS
C     -----------------------------------------------------------------
C     YLRC,YYRC,IYRC... WORKING COPIES OF THE ARRAYS YL,YY,YI.
C     DELT11,DELT13,DELT33... DEVIATIONS (IN ABSOLUTE VALUES OF THE TWO
C             COMPUTED BASIS VECTORS OF THE RAY-CENTRED COORDINATE
C             SYSTEM FROM THE CONDITIONS OF ORTHONORMALITY, SEE
C             C.R.T.5.8.2D.
C     VNEXPS..VELOCITY**(-NEXPS).
C YLRC,YYRC,IYRC ARE DEFINED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL METRIC,HPCG
C     FUNCTION NSRFC()... SUBROUTINE PACKAGE 'MODEL'.
C     SUBROUTINE FCT(X,Y,D)... THIS PACKAGE.
C     SUBROUTINE OUTP(X,Y,D,IHLF,NDIM,PRMT)... THIS PACKAGE.
C     SUBROUTINE SMVPRD(G,A1,A2,A3,B1,B2,B3)... THIS PACKAGE.
C     SUBROUTINE PSHIFT(NY,Y,YI,Q111,Q121,Q112,Q122,Q1DET,Q211,Q221,
C             Q212,Q222,Q2DET,ISHIFT)... THIS PACKAGE.
C     SUBROUTINE CROSS(ISRF,NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,F)... THIS
C             PACKAGE.
C     SUBROUTINE HIVD2(NDIM,X1,Y1,D1,X2,Y2,D2,X,Y,D)... THIS PACKAGE.
C     SUBROUTINE HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)... IBM
C             SCIENTIFIC SUBROUTINE PACKAGE.
C     SUBROUTINE RPAR31(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'RPAR'.
C     SUBROUTINE RPAR32(ISTOR,YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE
C             'RPAR'.
C     SUBROUTINE WRIT31(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'WRIT'.
C     SUBROUTINE WRIT32(ISTOR,YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE
C             'WRIT'.
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)
C     SUBROUTINE BLOCK(COOR,ISRF1,ISB1,ISRF2,ISB2,ICB2)... SUBROUTINE
C             PACKAGE 'MODEL'.
C     SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)... SUBROUTINE
C             PACKAGE 'MODEL'.
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'SRFC'.
C     SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C             ... SUBROUTINE PACKAGE 'PARM'.
C
C DATE: 1990, JANUARY 25
C CODED BY LUDEK KLIMES
```

```
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS FOR LOCAL MODEL PARAMETERS:
      REAL G(12),GAMMA(18),GSQRD,FAUX(10)
      REAL UP(10),US(10),RO,QP,QS,VP,VS,VD(10),QL
C     THESE AUXILIARY VARIABLES AND ARRAYS NEED NOT BE LOCATED IN A
C     COMMON BLOCK.  THERE IS NO REASON TO LOCATE THEM IN THE AUXILIARY
C     COMMON BLOCK /AUXMOD/ BUT TO SHARE THE MEMORY.
      COMMON/AUXMOD/ G,GAMMA,GSQRD,FAUX,UP,US,RO,QP,QS,VP,VS,VD,QL
C
C     G,GAMMA,GSQRD... SEE SUBROUTINE METRIC OF THE PACKAGE 'METRIC'.
C     FAUX... AUXILIARY ARRAY TO STORE A FUNCTIONAL VALUE AND ITS
C             DERIVATIVES.
C     UP,US,RO,QP,QS... SEE SUBROUTINE PARM2 OF THE PACKAGE 'PARM'.
C     UP,US,QP,QS,VP,VS,VD,QL... SEE SUBROUTINE VELOC OF THE PACKAGE
C             'MODEL'.
C
C.......................................................................
C
C     EXTERNAL PROCEDURE NAMES:
      EXTERNAL FCT,OUTP
C     THESE SUBROUTINES ARE CALLED BY THE SUBROUTINE HPCG.
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER NDIM,IHLF,I
      PARAMETER (NDIM=27)
      REAL PRMT(6),DERY(NDIM),AUX(16,NDIM),VRED,AUX1
C
C     NDIM,IHLF,PRMT,DERY,AUX... SEE SUBROUTINE HPCG OF THE IBM
C             SCIENTIFIC SUBROUTINE PACKAGE.
C     I...    AUXILIARY LOOP VARIABLE.
C     VRED... APPROXIMATE REFERENCE VELOCITY TO ESTIMATE ERROR WEIGHTS.
C     AUX1... AUXILIARY STORAGE LOCATION.
C
C.......................................................................
C
C     (A) STORING AUXILIARY INPUT QUANTITIES IN COMMON BLOCK /RAYC/:
      DO 11 I=1,6
        YLRC(I)=YL(I)
   11 CONTINUE
      DO 12 I=1,5
        YYRC(I)=YY(I)
   12 CONTINUE
      DO 13 I=1,12
        IYRC(I)=IY(I)
   13 CONTINUE
C
C     (B) INITIAL VALUES FOR NUMERICAL INTEGRATION:
   20 PRMT(1)=YYRC(1)
      PRMT(2)=PRMT(1)+999999.
      PRMT(3)=STEP
      PRMT(4)=13.444*YYRC(2)
      CALL METRIC(Y(3),GSQRD,G,GAMMA)
      IF(IYRC(5).GE.0) THEN
        VRED=YLRC(1)
      ELSE
        VRED=YLRC(2)
      END IF
      DERY(1)=1.
```

```
      DERY(2)=1.
      DERY(3)=SQRT(G(1))/VRED
      DERY(4)=SQRT(G(3))/VRED
      DERY(5)=SQRT(G(6))/VRED
      AUX1=STEP*VRED**(1-NEXPS)
      DERY(6)=SQRT(G(7))*AUX1
      DERY(7)=SQRT(G(9))*AUX1
      DERY(8)=SQRT(G(12))*AUX1
      DO 21 I=9,NDIM
        DERY(I)=0.
   21 CONTINUE
C
C     (C) NUMERICAL INTEGRATION:
      CALL HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C
C     (D) GREAT NUMBER OF BISECTIONS:
      IF(PRMT(5).LT.0.) THEN
        YYRC(2)=2.*YYRC(2)
        GO TO 20
      END IF
C
C     (E) RECALLING AUXILIARY INPUT QUANTITIES FROM COMMON BLOCK /RAYC/:
      DO 51 I=1,6
        YL(I)=YLRC(I)
   51 CONTINUE
      DO 52 I=1,5
        YY(I)=YYRC(I)
   52 CONTINUE
      DO 53 I=1,12
        IY(I)=IYRC(I)
   53 CONTINUE
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE FCT(X,Y,D)
      INTEGER NDIM
      PARAMETER (NDIM=27)
      REAL X,Y(NDIM),D(NDIM)
C
C THIS SUBROUTINE EVALUATES THE RIGHT-HAND SIDES OF THE SYSTEM OF
C ORDINARY DIFFERENTIAL EQUATIONS FOR Y(1) TO Y(27). SEE C.R.T., SECTION
C 5.8.2.
C
C INPUT:
C     X...     VALUE OF THE INDEPENDENT VARIABLE ALONG THE RAY.
C     Y...     ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C NONE OF THE INPUT PARAMETERS EXCEPT Y(3)-Y(8) IS ALTERED.
C
C OUTPUT:
C     D...     ARRAY CONTAINING DERIVATIVES OF THE BASIC QUANTITIES
C              COMPUTED ALONG THE RAY, WITH RESPECT TO THE INDEPENDENT
C              VARIABLE ALONG THE RAY.
C     Y(3:8)..RENORMALIZED ORTHOGONAL VECTORS.  THE MODIFICATION SHOULD
C              BE NEGLIGIBLE.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
```

```
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /RAYC/ -  COMMUNICATION BETWEEN RAYCB, FCT, AND OUTP:
      REAL YLRC(6),YYRC(5),DELT11,DELT13,DELT33,VNEXPS
      INTEGER IYRC(12)
      COMMON/RAYC/YLRC,YYRC,IYRC,DELT11,DELT13,DELT33,VNEXPS
C YLRC,IYRC ARE MODIFIED IN THIS SUBROUTINE.  DELT11,DELT13,DELT33 AND
C VNEXPS ARE DEFINED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL KOOR,METRIC,PARM2,VELOC,SMVPRD
      INTEGER KOOR
C     SUBROUTINE SMVPRD(G,A1,A2,A3,B1,B2,B3)... THIS PACKAGE.
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)
C     FUNCTION KOOR()... SUBROUTINE PACKAGE 'METRIC'.
C     SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)... SUBROUTINE
C             PACKAGE 'MODEL'.
C     SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C             ... SUBROUTINE PACKAGE 'PARM'.
C
C DATE: 1990, JANUARY 25
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS FOR LOCAL MODEL PARAMETERS:
      REAL G(12),GAMMA(18),GSQRD,FAUX(10)
      REAL UP(10),US(10),RO,QP,QS,VP,VS,VD(10),QL
C     THESE AUXILIARY VARIABLES AND ARRAYS NEED NOT BE LOCATED IN A
C     COMMON BLOCK.  THERE IS NO REASON TO LOCATE THEM IN THE AUXILIARY
C     COMMON BLOCK /AUXMOD/ BUT TO SHARE THE MEMORY.
      COMMON/AUXMOD/ G,GAMMA,GSQRD,FAUX,UP,US,RO,QP,QS,VP,VS,VD,QL
C
C     G,GAMMA,GSQRD... SEE SUBROUTINE METRIC OF THE PACKAGE 'METRIC'.
C     FAUX... AUXILIARY ARRAY TO STORE A FUNCTIONAL VALUE AND ITS
C             DERIVATIVES.
C     UP,US,RO,QP,QS... SEE SUBROUTINE PARM2 OF THE PACKAGE 'PARM'.
C     UP,US,QP,QS,VP,VS,VD,QL... SEE SUBROUTINE VELOC OF THE PACKAGE
C             'MODEL'.
C
C.......................................................................
C
C     AUXILIARY STORAGE LOCATIONS:
C
      REAL H11,H21,H31,H41,H51,H61,H42,H52,H62,H13,H23,H33,H43,H53,H63
      REAL D11,D13,D33,V,V1,V11,V12,V22,AUX1,AUX2,AUX3,AUX4
      REAL AUX11,AUX21,AUX31,AUX12,AUX22,AUX32,AUX13,AUX23,AUX33
C
C     H11,H21,H31,H13,H23,H33... COVARIANT COMPONENTS OF THE BASIS
C             VECTORS OF RAY-CENTRED COORDINATE SYSTEM.
C     H41,H51,H61,H42,H52,H62,H43,H53,H63... CONTRAVARIANT COMPONENTS OF
C             THE BASIS VECTORS OF RAY-CENTRED COORDINATE SYSTEM.  H41,
C             H51,H61,H43,H53,H63 ARE NOT USED IN CARTESIAN COORDINATES.
```

```
C       D11,D13,D33... NORMS AND SCALAR PRODUCT OF BASIS VECTORS H1, H2.
C       V,V1,V11,V12,V22,... VELOCITY AND VELOCITY DERIVATIVES IN
C               RAY-CENTRED COORDINATE SYSTEM.
C       AUX1,AUX2,AUX3,AUX4... AUXILIARY STORAGE LOCATIONS.
C       AUX11,AUX21,AUX31,AUX12,AUX22,AUX32,AUX13,AUX23,AUX33... AUXILIARY
C               STORAGE LOCATIONS.  NOT USED IN CARTESIAN COORDINATES.
C
C.......................................................................
C
C       (A) NUMBER OF INVOCATIONS OF FCT:
        IYRC(9)=IYRC(9)+1
C
C       (B) MATERIAL PARAMETERS:
        CALL PARM2(IABS(IYRC(5)),Y(3),UP,US,YLRC(3),QP,QS)
        CALL VELOC(IYRC(5),UP,US,QP,QS,YLRC(1),YLRC(2),VD,QL)
        YLRC(4)=VD(2)
        YLRC(5)=VD(3)
        YLRC(6)=VD(4)
        V=VD(1)
        VNEXPS=V**(-NEXPS)
C
C       (C) BASIS OF THE RAY-CENTRED COORDINATE SYSTEM:
C       (C1) NON-UNIT VECTORS - COVARIANT COMPONENTS
        H13=Y(6)
        H23=Y(7)
        H33=Y(8)
        H11=Y(9)
        H21=Y(10)
        H31=Y(11)
        IF(KOOR().NE.0) THEN
C         CURVILINEAR COORDINATES:
          CALL METRIC(Y(3),GSQRD,G,GAMMA)
C         (C2) NON-UNIT VECTORS - CONTRAVARIANT COMPONENTS
          CALL SMVPRD(G(7),H13,H23,H33,H43,H53,H63)
          CALL SMVPRD(G(7),H11,H21,H31,H41,H51,H61)
C         (C3) NORMS:
          D33=SQRT(H13*H43+H23*H53+H33*H63)
          D11=SQRT(H11*H41+H21*H51+H31*H61)
          D13=(H11*H43+H21*H53+H31*H63)/(D11*D33)
C         (C4) ORTHONORMAL VECTORS:
C         COVARIANT COMPONENTS
          H13=H13/D33
          H23=H23/D33
          H33=H33/D33
          H11=H11/D11-H13*D13
          H21=H21/D11-H23*D13
          H31=H31/D11-H33*D13
C         CONTRAVARIANT COMPONENTS
          H43=H43/D33
          H53=H53/D33
          H63=H63/D33
          H41=H41/D11-H43*D13
          H51=H51/D11-H53*D13
          H61=H61/D11-H63*D13
          H42=(H23*H31-H33*H21)/GSQRD
          H52=(H33*H11-H13*H31)/GSQRD
          H62=(H13*H21-H23*H11)/GSQRD
C
C         (D) QUANTITIES USEFUL TO TEST THE ACCURACY OF COMPUTATIONS:
          DELT11=ABS(D11-1.)
```

```
         DELT33=ABS(V*D33-1.)
         DELT11=ABS(D13)
C
C        (E) FIRST AND SECOND DERIVATIVES OF THE VELOCITY IN RAY-CENTRED
C        COORDINATE SYSTEM:
C        SECOND COVARIANT VELOCITY DERIVATIVES:
         VD(5)=VD(5)-GAMMA(1)*VD(2)-GAMMA(7)*VD(3)-GAMMA(13)*VD(4)
         VD(6)=VD(6)-GAMMA(2)*VD(2)-GAMMA(8)*VD(3)-GAMMA(14)*VD(4)
         VD(7)=VD(7)-GAMMA(3)*VD(2)-GAMMA(9)*VD(3)-GAMMA(15)*VD(4)
         VD(8)=VD(8)-GAMMA(4)*VD(2)-GAMMA(10)*VD(3)-GAMMA(16)*VD(4)
         VD(9)=VD(9)-GAMMA(5)*VD(2)-GAMMA(11)*VD(3)-GAMMA(17)*VD(4)
         VD(10)=VD(10)-GAMMA(6)*VD(2)-GAMMA(12)*VD(3)-GAMMA(18)*VD(4)
C        RAY-CENTRED COORDINATE SYSTEM:
         V1=VD(2)*H41+VD(3)*H51+VD(4)*H61
         CALL SMVPRD(VD(5),H41,H51,H61,AUX1,AUX2,AUX3)
         V11=AUX1*H41+AUX2*H51+AUX3*H61
         V12=AUX1*H42+AUX2*H52+AUX3*H62
         CALL SMVPRD(VD(5),H42,H52,H62,AUX1,AUX2,AUX3)
         V22=AUX1*H42+AUX2*H52+AUX3*H62
C
C        (F) CORRECTION OF THE COMPUTED QUANTITIES:
         Y(6)=H13/V
         Y(7)=H23/V
         Y(8)=H33/V
         Y(9)=H11
         Y(10)=H21
         Y(11)=H31
C
C        (G) RIGHT-HAND SIDES OF THE DIFFERENTIAL EQUATIONS:
         AUX1=V*VNEXPS
         AUX2=V*AUX1
         AUX3=VNEXPS/V
         AUX4=V1*VNEXPS
         D(1)=VNEXPS
         D(2)=0.5*QL*VNEXPS
         D(3)=H43*AUX1
         D(4)=H53*AUX1
         D(5)=H63*AUX1
         CALL SMVPRD(GAMMA(1),H43,H53,H63,AUX11,AUX21,AUX31)
         CALL SMVPRD(GAMMA(7),H43,H53,H63,AUX12,AUX22,AUX32)
         CALL SMVPRD(GAMMA(13),H43,H53,H63,AUX13,AUX23,AUX33)
         D(6)=-VD(2)*AUX3+(AUX11*H13+AUX12*H23+AUX13*H33)*VNEXPS
         D(7)=-VD(3)*AUX3+(AUX21*H13+AUX22*H23+AUX23*H33)*VNEXPS
         D(8)=-VD(4)*AUX3+(AUX31*H13+AUX32*H23+AUX33*H33)*VNEXPS
         D(9) =H13*AUX4+(AUX11*H11+AUX12*H21+AUX13*H31)*AUX1
         D(10)=H23*AUX4+(AUX21*H11+AUX22*H21+AUX23*H31)*AUX1
         D(11)=H33*AUX4+(AUX31*H11+AUX32*H21+AUX33*H31)*AUX1
       ELSE
C        CARTESIAN COORDINATES (20 OF THESE STATEMENT LINES ARE THE SAME
C        AS FOR CURVILINEAR COORDINATES):
C        (C2) NON-UNIT VECTORS - CONTRAVARIANT COMPONENTS
C        H43=H13, H53=H23, H63=H33, H41=H11, H51=H21, H61=H31
C        (C3) NORMS:
         D33=SQRT(H13*H13+H23*H23+H33*H33)
         D11=SQRT(H11*H11+H21*H21+H31*H31)
         D13=(H11*H13+H21*H23+H31*H33)/(D11*D33)
C        (C4) ORTHONORMAL VECTORS:
C        COVARIANT=CONTRAVARIANT COMPONENTS
         H13=H13/D33
         H23=H23/D33
```

```
      H33=H33/D33
      H11=H11/D11-H13*D13
      H21=H21/D11-H23*D13
      H31=H31/D11-H33*D13
      H12=(H23*H31-H33*H21)
      H22=(H33*H11-H13*H31)
      H32=(H13*H21-H23*H11)
C
C     (D) QUANTITIES USEFUL TO TEST THE ACCURACY OF COMPUTATIONS:
      DELT11=ABS(D11-1.)
      DELT33=ABS(V*D33-1.)
      DELT11=ABS(D13)
C
C     (E) FIRST AND SECOND DERIVATIVES OF THE VELOCITY IN RAY-CENTRED
C     COORDINATE SYSTEM:
      V1=VD(2)*H11+VD(3)*H21+VD(4)*H31
      CALL SMVPRD(VD(5),H11,H21,H31,AUX1,AUX2,AUX3)
      V11=AUX1*H11+AUX2*H21+AUX3*H31
      V12=AUX1*H12+AUX2*H22+AUX3*H32
      CALL SMVPRD(VD(5),H12,H22,H32,AUX1,AUX2,AUX3)
      V22=AUX1*H12+AUX2*H22+AUX3*H32
C
C     (F) CORRECTION OF THE COMPUTED QUANTITIES:
      Y(6)=H13/V
      Y(7)=H23/V
      Y(8)=H33/V
      Y(9)=H11
      Y(10)=H21
      Y(11)=H31
C
C     (G) RIGHT-HAND SIDES OF THE DIFFERENTIAL EQUATIONS:
      AUX1=V*VNEXPS
      AUX2=V*AUX1
      AUX3=VNEXPS/V
      AUX4=V1*VNEXPS
      D(1)=VNEXPS
      D(2)=0.5*QL*VNEXPS
      D(3)=H13*AUX1
      D(4)=H23*AUX1
      D(5)=H33*AUX1
      D(6)=-VD(2)*AUX3
      D(7)=-VD(3)*AUX3
      D(8)=-VD(4)*AUX3
      D(9)=H13*AUX4
      D(10)=H23*AUX4
      D(11)=H33*AUX4
    END IF
    V11=-V11*AUX3
    V12=-V12*AUX3
    V22=-V22*AUX3
    D(12)=AUX2*Y(14)
    D(13)=AUX2*Y(15)
    D(14)=V11*Y(12)+V12*Y(13)
    D(15)=V12*Y(12)+V22*Y(13)
    D(16)=AUX2*Y(18)
    D(17)=AUX2*Y(19)
    D(18)=V11*Y(16)+V12*Y(17)
    D(19)=V12*Y(16)+V22*Y(17)
    D(20)=AUX2*Y(22)
    D(21)=AUX2*Y(23)
```

```
      D(22)=V11*Y(20)+V12*Y(21)
      D(23)=V12*Y(20)+V22*Y(21)
      D(24)=AUX2*Y(26)
      D(25)=AUX2*Y(27)
      D(26)=V11*Y(24)+V12*Y(25)
      D(27)=V12*Y(24)+V22*Y(25)
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE OUTP(X,Y,D,IHLF,NDIM,PRMT)
      INTEGER IHLF,NDIM,MY
      PARAMETER (MY=35)
      REAL X,Y(MY),D(MY),PRMT(5)
C
C THIS SUBROUTINE INCLUDES VARIOUS TESTS OF THE POSITION OF THE NEWLY
C COMPUTED POINT OF THE RAY, TESTS FOR POSSIBLE CAUSTIC POINTS, ETC.  IT
C ALSO STORES THE COMPUTED QUANTITIES INTO PROPER FILES IF REQUIRED.
C
C INPUT:
C     X...    VALUE OF THE INDEPENDENT VARIABLE ALONG THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     D...    ARRAY CONTAINING DERIVATIVES OF THE BASIC QUANTITIES
C             COMPUTED ALONG THE RAY, WITH RESPECT TO THE INDEPENDENT
C             VARIABLE ALONG THE RAY.
C     IHLF... NUMBER OF BISECTIONS OF THE INITIAL INCREMENT OF
C             INDEPENDENT VARIABLE.
C     NDIM... NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS.
C     PRMT... ARRAY CONTAINING PARAMETERS OF THE INTEGRATION.
C NONE OF THE INPUT PARAMETERS EXCEPT X,Y,D,PRMT(5) IS ALTERED.
C
C OUTPUT:
C     X,Y,D...VALUES CORRESPONDING TO THE POINT OF INTERSECTION OF THE
C             RAY WITH THE BOUNDARY OF THE COMPLEX BLOCK OR THE
C             COMPUTATIONAL VOLUME IF THE RAY INTERSECTS THE BOUNDARY.
C             UNALTERED IF THE RAY DOES NOT INTERSECT THE BOUNDARY.
C     PRMT(5)=1.0... THE RAY HAS INTERSECTED THE BOUNDARY OF THE COMPLEX
C               BLOCK OR THE COMPUTATIONAL VOLUME,
C             -1.0... NUMBER IHLF OF BISECTIONS OF THE INITIAL INCREMENT
C               GREATER THAN THE SPECIFIED LIMIT NHLF,
C             OTHERWISE UNALTERED.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /INITC/ (SEE SUBROUTINE PACKAGE 'INIT'):
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
```

```
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /RAYC/ -  COMMUNICATION BETWEEN RAYCB, FCT, AND OUTP:
      REAL YLRC(6),YYRC(5),DELT11,DELT13,DELT33,VNEXPS
      INTEGER IYRC(12)
      COMMON/RAYC/YLRC,YYRC,IYRC,DELT11,DELT13,DELT33,VNEXPS
C YLRC,YYRC,IYRC ARE MODIFIED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SRFC2,PARM2,BLOCK,VELOC,RPAR31,RPAR32,WRIT31,WRIT32
      EXTERNAL NSRFC,CROSS,HIVD2,PSHIFT
      INTEGER NSRFC
C     FUNCTION NSRFC()... SUBROUTINE PACKAGE 'MODEL'.
C     SUBROUTINE PSHIFT(NY,Y,YI,Q111,Q121,Q112,Q122,Q1DET,Q211,Q221,
C             Q212,Q222,Q2DET,ISHIFT)... THIS PACKAGE.
C     SUBROUTINE CROSS(ISRF,NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,F)... THIS
C             PACKAGE.
C     SUBROUTINE HIVD2(NDIM,X1,Y1,D1,X2,Y2,D2,X,Y,D)... THIS PACKAGE.
C     SUBROUTINE RPAR31(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'RPAR'.
C     SUBROUTINE RPAR32(ISTOR,YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE
C             'RPAR'.
C     SUBROUTINE WRIT31(YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE 'WRIT'.
C     SUBROUTINE WRIT32(ISTOR,YL,Y,YY,IY)... SAMPLE SUBROUTINE PACKAGE
C             'WRIT'.
C     SUBROUTINE BLOCK(COOR,ISRF1,ISB1,ISRF2,ISB2,ICB2)... SUBROUTINE
C             PACKAGE 'MODEL'.
C     SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)... SUBROUTINE
C             PACKAGE 'MODEL'.
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'SRFC'.
C     SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C             ... SUBROUTINE PACKAGE 'PARM'.
C
C ERROR MESSAGES:
C     581...  TOO MANY FICTITIOUS INTERFACES:
C             MORE THAN 100 FICTITIOUS INTERFACES CROSSED DURING ONE
C             STEP OF THE NUMERICAL INTEGRATION.
C             THIS ERROR SHOULD NOT APPEAR.  CONTACT THE AUTHORS.
C     582,583... EXCLUDED PROGRAM BRANCH:
C             THIS ERROR SHOULD NOT APPEAR.  CONTACT THE AUTHORS.
C     584...  TOO THIN COMPLEX BLOCK:
C             THIS ERROR SHOULD NOT APPEAR.  CONTACT THE AUTHORS.
C     585,586... THIN COMPLEX BLOCK SKIPPED:
C             A THIN COMPLEX BLOCK HAS BEEN SKIPPED OVER WITHIN THE LAST
C             STEP OF THE NUMERICAL INTEGRATION.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS FOR LOCAL MODEL PARAMETERS:
      REAL G(12),GAMMA(18),GSQRD,FAUX(10)
      REAL UP(10),US(10),RO,QP,QS,VP,VS,VD(10),QL
C     THESE AUXILIARY VARIABLES AND ARRAYS NEED NOT BE LOCATED IN A
C     COMMON BLOCK.  THERE IS NO REASON TO LOCATE THEM IN THE AUXILIARY
C     COMMON BLOCK /AUXMOD/ BUT TO SHARE THE MEMORY.
      COMMON/AUXMOD/ G,GAMMA,GSQRD,FAUX,UP,US,RO,QP,QS,VP,VS,VD,QL
C
```

```
C      G,GAMMA,GSQRD... SEE SUBROUTINE METRIC OF THE PACKAGE 'METRIC'.
C      FAUX... AUXILIARY ARRAY TO STORE A FUNCTIONAL VALUE AND ITS
C              DERIVATIVES.
C      UP,US,RO,QP,QS... SEE SUBROUTINE PARM2 OF THE PACKAGE 'PARM'.
C      UP,US,QP,QS,VP,VS,VD,QL... SEE SUBROUTINE VELOC OF THE PACKAGE
C              'MODEL'.
C
C.......................................................................
C
C      AUXILIARY STORAGE LOCATIONS:
C
       REAL Q111,Q121,Q112,Q122,Q1DET,Q211,Q221,Q212,Q222,Q2DET
       SAVE Q111,Q121,Q112,Q122,Q1DET,Q211,Q221,Q212,Q222,Q2DET
       REAL X1,X2,Y1(MY),Y2(MY),D1(MY),D2(MY)
       SAVE X1,X2,Y1,Y2,D1,D2
       REAL XAUX,YAUX(MY),DAUX(MY),AUX,ERR
       INTEGER IYAUX(12),IYRC12,ISRF2,ISB2,ICB2,K1,K2,K,I
C
C      Q111,Q121,Q112,Q122,Q1DET... MATRIX OF RAY GEOMETRICAL SPREADING
C              AND ITS DETERMINANT AT THE LAST BUT ONE COMPUTED POINT OF
C              THE RAY.
C      Q211,Q221,Q212,Q222,Q2DET... MATRIX OF RAY GEOMETRICAL SPREADING
C              AND ITS DETERMINANT AT THE LAST COMPUTED POINT OF THE RAY.
C      X1,Y1,D1... INDEPENDENT VARIABLE, COMPUTED QUANTITIES AND THEIR
C              DERIVATIVES AT THE LAST BUT ONE COMPUTED POINT OF THE RAY.
C      X2,Y2,D2... INDEPENDENT VARIABLE, COMPUTED QUANTITIES AND THEIR
C              DERIVATIVES AT THE LAST COMPUTED POINT OF THE RAY.
C      IYRC12..VALUE OF IYRC(12)=KMAH AT THE LAST BUT ONE COMPUTED POINT
C              OF THE RAY.
C      XAUX,YAUX,DAUX,AUX,IYAUX,ISRF2,ISB2,ICB2,K1,K2,K,I...  AUXILIARY
C              STORAGE LOCATIONS.
C      ERR...  MAXIMUM ERROR IN INDEPENDENT VARIABLE FOR THE
C              DETERMINATION OF THE POINT OF INTERSECTION.
C
C.......................................................................
C
C      (A) COPYING THE COMPUTED QUANTITIES:
       X1=X2
       X2=X
       DO 11 I=1,NDIM
         Y1(I)=Y2(I)
         Y2(I)=Y(I)
         D1(I)=D2(I)
         D2(I)=D(I)
   11  CONTINUE
       DO 12 I=NDIM+1,IYRC(1)
         Y1(I)=Y(I)
   12  CONTINUE
       IF(X.EQ.PRMT(1)) THEN
C        BEGINNIG OF THE NUMERICAL INTEGRATION
         CALL PSHIFT(27,Y,YI,0.00,0.00,0.00,0.00,0.000,
     *                     Q211,Q221,Q212,Q222,Q2DET,ISHIFT)
         RETURN
       END IF
C
C      (B) NUMBER OF INVOCATIONS OF OUTP
       IYRC(10)=IYRC(10)+1
C      INITIAL VALUE OF THE INDEX OF CROSSED SURFACE BOUNDING THE COMPLEX
C      BLOCK
       IYRC(6)=0
```

```
C
C     (C) CHECK FOR CROSSING THE COORDINATE BOUNDARIES OF THE
C     COMPUTATIONAL VOLUME:
      ERR=YYRC(2)*VNEXPS
      DO 31 I=3,5
        IF(Y(I).LT.BOUNDR(I+I-5)) THEN
          IYRC(6)=95+I+I
          FAUX(1)=Y(I)-BOUNDR(I+I-5)
          FAUX(2)=0.
          FAUX(3)=0.
          FAUX(4)=0.
          FAUX(I-1)=1.
          CALL CROSS(IYRC(6),NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,FAUX)
        END IF
        IF(Y(I).GT.BOUNDR(I+I-4)) THEN
          IYRC(6)=96+I+I
          FAUX(1)=Y(I)-BOUNDR(I+I-4)
          FAUX(2)=0.
          FAUX(3)=0.
          FAUX(4)=0.
          FAUX(I-1)=1.
          CALL CROSS(IYRC(6),NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,FAUX)
        END IF
   31 CONTINUE
      IF(Y(1).GT.BOUNDR(7)) THEN
        IYRC(6)=107
        FAUX(1)=Y(I)-BOUNDR(7)
        FAUX(2)=Y(6)*D(I)/D(1)
        FAUX(3)=Y(7)*D(I)/D(1)
        FAUX(4)=Y(8)*D(I)/D(1)
        CALL CROSS(IYRC(6),NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,FAUX)
      END IF
      IF(IYRC(6).GT.100) THEN
        IYRC(7)=IYRC(4)
        IYRC(8)=IABS(IYRC(5))
      END IF
C
C     (D) CHECK FOR CROSSING THE BOUNDARY OF THE COMPLEX BLOCK:
C     NOTE: IYRC(4)=ISB1, IYRC(5)=ICB1, IYRC(6)=ISRF.
      CALL BLOCK(Y(3),0,IYRC(4),ISRF2,ISB2,ICB2,FAUX)
      IF(ISRF2.NE.0) THEN
C       BOUNDARY OF THE SIMPLE BLOCK IS CROSSED
C       NOTE: IN THIS ROUTINE, UNLIKE IN THE PAPER ON C.R.T., THE POINT
C       OF INTERSECTION WITH THE BOUNDARY OF THE SIMPLE BLOCK IS FOUND
C       EVEN IF THE BOUNDARY OF THE COMPLEX BLOCK IS NOT CROSSED.
C       (D1)
        ISBAUX=ISB2
        XAUX=X
        DO 41 I=1,NDIM
          YAUX(I)=Y(I)
          DAUX(I)=D(I)
   41   CONTINUE
C       FOLLOWING LOOP IS INCLUDED TO AVOID INFINITE REPEATING OF THE
C       STEPS (D2) AND (D3) OF THE ALGORITHM
        DO 47 K=1,100
C         (D2)
          IYRC(6)=ISRF2
C         (D3)
          CALL CROSS(IABS(IYRC(6)),NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,
     *                                                            FAUX)
```

```
C          X,Y IS THE POINT OF INTERSECTION WITH THE SURFACE IYRC(6)
           CALL BLOCK(Y(3),IYRC(6),IYRC(4),ISRF2,ISB2,ICB2,FAUX)
   43      CONTINUE
           IF(ISRF2.NE.0) THEN
C            (D3-I)
C            X,Y IS NOT SITUATED AT THE BOUNDARY OF THE SIMPLE BLOCK
             IF(FAUX(1)*(Y(6)*FAUX(2)+Y(7)*FAUX(3)+Y(8)*FAUX(4)).LT.0.)
     *                                                              THEN
C              SINCE, IN THE CURRENT VERSION OF THE SUBROUTINE CROSS, THE
C              THE POINT OF INTERSECTION WITH THE SURFACE IYRC(6) IS
C              ALWAYS FOUND AFTER (NOT BEFORE) THE SURFACE, THIS PROGRAM
C              BRANCH SHOULD BE EXCLUDED:
               STOP 'ERROR 582 IN OUTP: EXCLUDED PROGRAM BRANCH'
C              THE POINT OF INTERSECTION WITH THE SURFACE IYRC(6),
C              SEPARATED FROM THE SIMPLE BLOCK IYRC(4) BY THE SURFACE
C              ISRF2, IS SITUATED BEFORE (NOT AFTER) THE SIMPLE BLOCK
C              IYRC(4).  THIS IS LIKELY DUE TO A VERY THIN SIMPLE BLOCK
C              IYRC(4) IN THE VICINITY OF THE POINT OF INTERSECTION.
C              THIS SITUATION IS NOT CONSIDERED IN THE ORIGINAL PAPER
C              ON C.R.T. AND MAY CAUSE SOME DIFFICULTIES:
C              IF(IABS(IYRC(5)).EQ.ICB2) THEN
C                ISRF2=IYRC(6)
C                IYRC(4)=ISB2
C              ELSE
C                STOP 'ERROR 584 IN OUTP: TOO THIN COMPLEX BLOCK'
C              END IF
             END IF
C            THE POINT OF INTERSECTION WITH THE SURFACE IYRC(6) IS
C            FOUND WITHIN THE SPECIFIED ERROR ERR.  IT MAY BE SEPARATED
C            FROM THE SIMPLE BLOCK IYRC(4) BY A SURFACE ISRF2 SITUATED
C            WITHIN THE LAST INTERVAL OF THE LENGTH ERR.  IN SUCH A
C            CASE, THE SURFACE ISRF2 NEED NOT BE TAKEN INTO ACCOUNT.
C            THIS SITUATION IS NOT CONSIDERED IN THE ORIGINAL PAPER
C            ON C.R.T. AND IS EXAMINED IN THE FOLLOWING LINES:
             K2=ISRF2
             DO 44 I=3,5
               Y(I)=Y(I)-D(I)*ERR
   44        CONTINUE
             CALL BLOCK(Y(3),IYRC(6),IYRC(4),ISRF2,ISB2,ICB2,FAUX)
             DO 45 I=3,5
               Y(I)=Y(I)+D(I)*ERR
   45        CONTINUE
             IF(ISRF2.EQ.0) THEN
C              SINCE, IN THE CURRENT VERSION OF THE SUBROUTINE CROSS, THE
C              THE POINT OF INTERSECTION WITH THE SURFACE IYRC(6) IS
C              ALWAYS FOUND AFTER (NOT BEFORE) THE SURFACE, THIS PROGRAM
C              BRANCH SHOULD BE EXCLUDED:
               STOP 'ERROR 583 IN OUTP: EXCLUDED PROGRAM BRANCH'
C              THE FORMERLY FOUND SURFACE ISRF2 WAS SITUATED WITHIN THE
C              LAST INTERVAL OF THE LENGTH ERR.  THUS IT NEED NOT BE
C              TAKEN INTO ACCOUNT:
C              GO TO 43
             ELSE IF(ISRF2.NE.K2) THEN
               IF(FAUX(1)*(Y(6)*FAUX(2)+Y(7)*FAUX(3)+Y(8)*FAUX(4)).LT.0.)
     *                                                              THEN
C                THE POINT OF INTERSECTION WITH THE SURFACE ISRF2 IS
C                IS SITUATED BEFORE (NOT AFTER) THE SIMPLE BLOCK
C                IYRC(4) AND WILL NOT BE CONSIDERED
                 ISRF2=K2
               END IF
```

```
              END IF
C             GO TO (D2)
            ELSE IF(ICB2.EQ.IABS(IYRC(5))) THEN
C             (D3-II)
C             X,Y IS SITUATED AT THE BOUNDARY OF THE SIMPLE BLOCK
C             BUT NOT SITUATED AT THE BOUNDARY OF THE COMPLEX BLOCK
              IYRC(4)=ISB2
              X=XAUX
              DO 46 I=1,NDIM
                Y(I)=YAUX(I)
                D(I)=DAUX(I)
   46         CONTINUE
              IF(ISB2.EQ.ISBAUX) THEN
C               BOUNDARY OF THE COMPLEX BLOCK HAS NOT BEEN CROSSED
C               DURING THE LAST STEP OF NUMERICAL INTEGRATION
                IYRC(6)=0
                GO TO 49
              END IF
              CALL BLOCK(YAUX(3),0,IYRC(4),ISRF2,ISB2,ICB2,FAUX)
C             GO TO (D2)
            ELSE
C             (D3-III)
C             X,Y IS SITUATED AT THE BOUNDARY OF THE SIMPLE BLOCK AND
C             X,Y IS SITUATED AT THE BOUNDARY OF THE COMPLEX BLOCK
              GO TO 48
            END IF
   47     CONTINUE
          STOP 'ERROR 581 IN OUTP: TOO MANY FICTITIOUS INTERFACES'
   48     CONTINUE
          IYRC(7)=ISB2
          IYRC(8)=ICB2
        END IF
   49 CONTINUE
C
C     (E) CHECK FOR CROSSING THE END SURFACES
      DO 51 I=1,NEND
        IF(KEND(I).GT.NSRFC()) THEN
          CALL SRFC2(KEND(I),Y(3),FAUX)
          IF(FAUX(1)*FSRFCA(KEND(I)-NSRFC()).LE.0.)  THEN
            IYRC(6)=KEND(I)
            CALL CROSS(IYRC(6),NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,FAUX)
          END IF
        END IF
   51 CONTINUE
C
C     (F) PHASE SHIFT DUE TO CAUSTICS
      Q111=Q211
      Q121=Q221
      Q112=Q212
      Q122=Q222
      Q1DET=Q2DET
      CALL PSHIFT(IYRC(1),Y,YI,Q111,Q121,Q112,Q122,Q1DET,
     *                     Q211,Q221,Q212,Q222,Q2DET,I)
      IYRC12=IYRC(12)
      IYRC(12)=IYRC(12)+I
C
C     QUANTITIIES DESCRIBING LOCAL PROPERTIES OF THE MODEL AT POINT X,Y:
      IF(IYRC(6).NE.0) THEN
        CALL PARM2(IABS(IYRC(5)),Y(3),UP,US,YLRC(3),QP,QS)
        CALL VELOC(IYRC(5),UP,US,QP,QS,YLRC(1),YLRC(2),VD,QL)
```

```
          YLRC(4)=VD(2)
          YLRC(5)=VD(3)
          YLRC(6)=VD(4)
        END IF
C
C       (G) STORAGE OF THE COMPUTED QUANTITIES AT GIVEN SURFACES:
        DO 70 I=1,3
          IYAUX(I)=IYRC(I)
          IYAUX(I+5)=0.0
          IYAUX(I+8)=IYRC(I+8)
   70   CONTINUE
        IYAUX(5)=IYRC(5)
        DO 72 K=1,NSTOR
          IF(KSTOR(K).GT.NSRFC()) THEN
            IF(KSTOR(K).EQ.IYRC(6)) THEN
              YYRC(1)=X
              CALL RPAR32(K,YLRC,Y,YYRC,IYRC)
              CALL WRIT32(K,YLRC,Y,YYRC,IYRC)
            ELSE
              CALL SRFC2(KSTOR(K),Y(3),FAUX)
              IF(FAUX(1)*FSRFCA(KSTOR(K)-NSRFC()).LE.0.)  THEN
                FSRFCA(KSTOR(K)-NSRFC())=FAUX(1)
                CALL CROSS(KSTOR(K),NDIM,ERR,X1,Y1,D1,X2,Y2,D2,
     *                                          XAUX,YAUX,DAUX,FAUX)
                DO 71 I=NDIM+1,IYRC(1)
                  YAUX(I)=Y1(I)
   71           CONTINUE
                CALL PSHIFT(IYRC(1),YAUX,YI,Q111,Q121,Q112,Q122,Q1DET,
     *                      DAUX(1),DAUX(2),DAUX(3),DAUX(4),AUX,I)
                IYAUX(12)=IYRC12+I
                CALL BLOCK(YAUX(3),0,IYRC(4),ISRF2,IYAUX(4),ICB2,FAUX)
                IF(ICB2.NE.IABS(IYRC(5))) THEN
                  STOP 'ERROR 585 IN OUTP: THIN COMPLEX BLOCK SKIPPED'
                END IF
                CALL PARM2(IABS(IYRC(5)),YAUX(3),UP,US,DAUX(3),QP,QS)
                CALL VELOC(IYRC(5),UP,US,QP,QS,DAUX(1),DAUX(2),VD,QL)
                DAUX(4)=VD(2)
                DAUX(5)=VD(3)
                DAUX(6)=VD(4)
                YYRC(1)=XAUX
                CALL RPAR32(K,DAUX,YAUX,YYRC,IYAUX)
                CALL WRIT32(K,DAUX,YAUX,YYRC,IYAUX)
              END IF
            END IF
          END IF
   72   CONTINUE
C
C       (H) STORAGE OF THE COMPUTED QUANTITIES ALONG THE RAY:
        IF(STORE.GT.0.) THEN
          K1=INT(X1/STORE+0.000001)+1
          K2=INT(X/STORE+0.000001)
          DO 82 K=K1,K2
            XAUX=FLOAT(K)*STORE
            CALL HIVD2(NDIM,X1,Y1,D1,X2,Y2,D2,XAUX,YAUX,DAUX)
            DO 81 I=NDIM+1,IYRC(1)
              YAUX(I)=Y1(I)
   81       CONTINUE
            CALL PSHIFT(IYRC(1),YAUX,YI,Q111,Q121,Q112,Q122,Q1DET,
     *                  DAUX(1),DAUX(2),DAUX(3),DAUX(4),AUX,I)
            IYAUX(12)=IYRC12+I
```

```
            CALL BLOCK(YAUX(3),0,IYRC(4),ISRF2,IYAUX(4),ICB2,FAUX)
            IF(ICB2.NE.IABS(IYRC(5))) THEN
              STOP 'ERROR 586 IN OUTP: THIN COMPLEX BLOCK SKIPPED'
            END IF
            CALL PARM2(IABS(IYRC(5)),YAUX(3),UP,US,DAUX(3),QP,QS)
            CALL VELOC(IYRC(5),UP,US,QP,QS,DAUX(1),DAUX(2),VD,QL)
            DAUX(4)=VD(2)
            DAUX(5)=VD(3)
            DAUX(6)=VD(4)
            YYRC(1)=XAUX
            CALL RPAR31(DAUX,YAUX,YYRC,IYAUX)
            CALL WRIT31(DAUX,YAUX,YYRC,IYAUX)
   82     CONTINUE
        END IF
C
C       (I) ACCUMULATION OF THE RENORMALIZATION ERRORS FOR A TEST OF
C       ACCURACY:
        AUX=0.5*(Y(1)-Y1(1))
        YYRC(3)=YYRC(3)+DELT33*AUX
        YYRC(4)=YYRC(4)+DELT13*AUX
        YYRC(5)=YYRC(5)+DELT11*AUX
C
C       (J) GREAT NUMBER OF BISECTIONS OF THE INITIAL INCREMENT:
        IF(IHLF.GT.NHLF) PRMT(5)=-1.
C
C       (K) FINAL OPERATIONS:
        IF(IYRC(6).NE.0) PRMT(5)=1.
        YYRC(1)=X
        RETURN
        END
C
C=======================================================================
C
      SUBROUTINE SMVPRD(G,A1,A2,A3,B1,B2,B3)
      REAL G(6),A1,A2,A3,B1,B2,B3
C
C THIS SUBROUTINE IS AN AUXILIARY ROUTINE TO FCT. IT EVALUATES SYMMETRIC
C MATRIX BY VECTOR PRODUCT.
C
C INPUT:
C   G...        ARRAY CONTAINING COMPONENTS G11, G12, G22, G13, G23, G33
C               OF THE 3*3 SYMMETRIC MATRIX.
C   A1,A2,A3... COMPONENTS OF THE 3-VECTOR.
C
C OUTPUT:
C   B1,B2,B3... COMPONENTS OF VECTOR B=G*A.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, SEPTEMBER 5
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      B1=G(1)*A1+G(2)*A2+G(4)*A3
      B2=G(2)*A1+G(3)*A2+G(5)*A3
      B3=G(4)*A1+G(5)*A2+G(6)*A3
      RETURN
      END
C
```

```
C=======================================================================
C
      SUBROUTINE PSHIFT(NY,Y,YI,Q111,Q121,Q112,Q122,Q1DET,
     *                          Q211,Q221,Q212,Q222,Q2DET,ISHIFT)
      INTEGER NY,ISHIFT
      REAL Y(NY),YI(21)
      REAL Q111,Q121,Q112,Q122,Q1DET,Q211,Q221,Q212,Q222,Q2DET
C
C THIS SUBROUTINE IS AN AUXILIARY ROUTINE TO OUTP. IT CORRECTS REDUCED
C AMPLITUDES WITH REGARD TO PHASE SHIFT DUE TO CAUSTICS
C (SEE C.R.T.5.8.3F).
C
C INPUT:
C     NY...   NUMBER OF BASIC QUANTITIES DESCRIBING THE POINT OF A RAY.
C     Y(1:11)... ANYTHING.
C     Y(12:27)... RAY PROPAGATOR MATRIX AT THE POINT 2 OF THE RAY.
C     Y(28:NY)... REDUCED AMPLITUDES AT THE POINT 1 OF THE RAY.
C     YI...   ARRAY CONTAINING QUANTITIES AT THE INITIAL POINT OF THE
C             RAY (SEE C.R.T.6.1)
C     Q111,Q121,Q112,Q122,Q1DET... MATRIX OF RAY GEOMETRICAL SPREADING
C             AND ITS DETERMINANT AT THE POINT 1 OF THE RAY.
C NONE OF THE INPUT PARAMETERS EXCEPT Y(28:NY) ARE ALTERED.
C
C OUTPUT:
C     Y(28:NY)... REDUCED AMPLITUDES AT THE POINT 2 OF THE RAY.
C     Q211,Q221,Q212,Q222,Q2DET... MATRIX OF RAY GEOMETRICAL SPREADING
C             AND ITS DETERMINANT AT THE POINT 2 OF THE RAY.
C     ISHIFT..ORDER OF A CAUSTIC BETWEEN POINTS 1 AND 2 (INCREMENT OF
C             THE KMAH INDEX).
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, SEPTEMBER 6
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATION:
      REAL AUX
C
      Q211=Y(12)*YI(12)+Y(16)*YI(13)+Y(20)*YI(14)+Y(24)*YI(15)
      Q221=Y(13)*YI(12)+Y(17)*YI(13)+Y(21)*YI(14)+Y(25)*YI(15)
      Q212=Y(12)*YI(16)+Y(16)*YI(17)+Y(20)*YI(18)+Y(24)*YI(19)
      Q222=Y(13)*YI(16)+Y(17)*YI(17)+Y(21)*YI(18)+Y(25)*YI(19)
      Q2DET=Q211*Q222-Q212*Q221
      IF(Q1DET*Q2DET.LT.0.) THEN
C       CAUSTIC OF THE FIRST ORDER (LINE CAUSTIC)
        ISHIFT=1
        DO 10 I=28,NY,2
          AUX=Y(I+1)
          Y(I+1)=-Y(I)
          Y(I)=AUX
   10   CONTINUE
      ELSE IF((Q111*Q222-Q112*Q221+Q122*Q211-Q121*Q212)*Q2DET.LT.0.)
     *      THEN
C       CAUSTIC OF THE SECOND ORDER (POINT CAUSTIC)
        ISHIFT=2
        DO 20 I=28,NY
          Y(I)=-Y(I)
   20   CONTINUE
```

```
      ELSE
C       NO CAUSTIC
        ISHIFT=0
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE CROSS(ISRF,NDIM,ERR,X1,Y1,D1,X2,Y2,D2,X,Y,D,F)
      INTEGER ISRF,NDIM
      REAL ERR,X1,Y1(NDIM),D1(NDIM),X2,Y2(NDIM),D2(NDIM)
      REAL X,Y(NDIM),D(NDIM),F(10)
C
C THIS SUBROUTINE FINDS THE POINT OF INTERSECTION OF A CURVE WITH
C A SURFACE (SEE C.R.T.5.8.4B).  THE CURVE IS PARAMETRIZED BY AN
C INDEPENDENT VARIABLE X AND EVALUATED BY THE HERMITE INTERPOLATION FROM
C THE TWO GIVEN POINTS.  THE SURFACE IS SPECIFIED IN AN IMPLICIT WAY BY
C SUBROUTINE SRFC2 WHICH IS DESCRIBED ELSWHERE.
C
C INPUT:
C     ISRF... INDEX OF THE SURFACE. FOR ISRF = 101 TO 107, THE SURFACE
C             COINCIDES WITH A COORDINATE SURFACE BOUNDING A VOLUME FOR
C             COMPUTATION.
C     NDIM... DIMENSION OF ARRAYS Y1,D1,Y2,D2,Y,D.
C     ERR...  MAXIMUM ERROR IN INDEPENDENT VARIABLE FOR THE
C             DETERMINATION OF THE POINT OF INTERSECTION.
C     X1...   INDEPENDENT VARIABLE CORRESPONDING TO THE FIRST POINT
C             GIVEN FOR THE INTERPOLATION OF THE CURVE.
C     Y1...   ARRAY CONTAINING DEPENDENT VARIABLES AT POINT X1.
C             Y1(3), Y1(4), Y1(5) MUST CONTAIN THE COORDINATES OF POINT
C             X1, Y1(6), Y1(7), Y1(8) MUST CONTAIN THE SLOWNESS VECTOR.
C     D1...   ARRAY CONTAINING THE DERIVATIVES OF THE DEPENDENT
C             VARIABLES AT POINT X1.
C     X2...   INDEPENDENT VARIABLE CORRESPONDING TO THE SECOND POINT
C             GIVEN FOR THE INTERPOLATION OF THE CURVE.
C     Y2...   ARRAY CONTAINING DEPENDENT VARIABLES AT POINT X2.
C             Y2(3), Y2(4), Y2(5) MUST CONTAIN THE COORDINATES OF POINT
C             X2, Y2(6), Y2(7), Y2(8) MUST CONTAIN THE SLOWNESS VECTOR.
C     D2...   ARRAY CONTAINING THE DERIVATIVES OF THE DEPENDENT
C             VARIABLES AT POINT X2.
C     X...    INDEPENDENT VARIABLE CORRESPONDING TO THE POINT OF THE
C             CURVE AT WHICH THE FUNCTION SPECIFYING THE SURFACE HAS THE
C             OPPOSITE SIGN THAN AT X1. THEN THE POINT OF INTERSECTION
C             IS BEING FOUND BETWEEN THE POINTS X1 AND X.  THE FOUND
C             APPROXIMATION OF THE POINT OF INTERSECTION IS SITUATED
C             CLOSE TO THE SURFACE, AT THE OPOSITE SIDE THAN THE GIVEN
C             POINT X1.
C     Y...    ARRAY OF THE DIMENSION AT LEAST NDIM.
C     D...    ARRAY CONTAINING THE DERIVATIVES OF THE DEPENDENT
C             VARIABLES AT POINT X.  D(3), D(4), D(5) MUST CONTAIN THE
C             DERIVATIVES OF COORDINATES WITH RESPECT TO X, AT THE POINT
C             X.  OTHER STORAGE LOCATIONS MAY BE UNDEFINED.
C     F...    ARRAY CONTAINING THE VALUE, FIRST AND SECOND DERIVATIVES
C             OF THE FUNCTION SPECIFYING THE SURFACE AT POINT X.
C ISRF, NDIM, ERR, X1, Y1, D1, X2, Y2, D2 ARE UNALTERED.
C
C OUTPUT:
C     X...    INDEPENDENT VARIABLE CORRESPONDING TO THE POINT OF
C             INTERSECTION.
```

```
C     Y...    ARRAY CONTAINING DEPENDENT VARIABLES AT THE POINT OF
C             INTERSECTION.
C     D...    ARRAY CONTAINING THE DERIVATIVES OF THE DEPENDENT
C             VARIABLES AT THE POINT OF INTERSECTION.
C     F...    UNDEFINED.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SRFC2,HIVD2
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'SRFC'.
C     SUBROUTINE HIVD2(NDIM,X1,Y1,D1,X2,Y2,D2,X,Y,D)... THIS SUBROUTINE
C             PACKAGE.
C
C ERROR MESSAGES:
C     588...  NO POINT OF INTERSECTION:
C             THE FUNCTION SPECIFYING THE SURFACE HAS THE SAME SIGN AT
C             THE GIVEN POINTS X1 AND X.  THUS, THE BOTH POINTS ARE
C             SITUATED AT THE SAME SIDE OF THE SURFACE AND THERE IS NO
C             GUARANTY TO FIND A POINT OF INTERSECTION BETWEEN THEM.
C     589...  TOO MANY ITERATIONS:
C             MORE THAN 20 ITERATIONS WHEN DETERMINING THE POINT OF
C             INTERSECTION OF THE RAY WITH A SURFACE.
C             THIS ERROR SHOULD NOT APPEAR.  CONTACT THE AUTHORS.
C
C DATE: 1990, JANUARY 15
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,ITER
      REAL XA,FA,DA,XB,FB,DB,XC,XCA,XCB,FOLD,RATE
C
C
C     INITIAL VALUES:
      XA=X
      FA=F(1)
      DA=F(2)*D(3)+F(3)*D(4)+F(4)*D(5)
      XB=X1
      X =X1
      DO 1 I=1,NDIM
        Y(I)=Y1(I)
        D(I)=D1(I)
    1 CONTINUE
      FOLD=FA
C     RATE OF THE GIVEN MAXIMUM ERROR AND A DESIRED ERROR
      RATE=5.
C
C     ITERATIONS:
```

```
      DO 2 ITER=1,20
C
C       FUNCTIONAL VALUE AT X:
        IF(ISRF.LE.100) THEN
          CALL SRFC2(ISRF,Y(3),F)
        ELSE IF(ISRF.LE.106) THEN
          I=(ISRF-97)/2
          F(1)=Y(I+1)-BOUNDR(ISRF-100)
          F(2)=0.
          F(3)=0.
          F(4)=0.
          F(I)=1.
        ELSE
          I=ISRF-106
          F(1)=Y(I)-BOUNDR(ISRF-100)
          F(2)=Y(6)*D(I)/D(1)
          F(3)=Y(7)*D(I)/D(1)
          F(4)=Y(8)*D(I)/D(1)
        END IF
C
C       SELECTION OF POINTS:
        IF(FA*F(1).GT.0.) THEN
          IF(ITER.EQ.1) THEN
            STOP 'ERROR 588 IN CROSS: NO POINT OF INTERSECTION'
          END IF
          XA=XB
          FA=FB
          DA=DB
        END IF
C
C       NEW POINT OR END OF ITERATIONS:
        IF(ABS(X-XA).LE.ERR) THEN
C         POINT OF INTERSECTION IS FOUND WITHIN THE SPECIFIED ERROR ERR
          IF(F(1)*FOLD.GT.0.) THEN
C           POINT X IS SITUATED AT THE OTHER SIDE OF THE SURFACE THAN X1
            GO TO 3
          ELSE IF(ABS(X-XA)*RATE.LE.ERR) THEN
C           POINT X IS SITUATED AT THE SAME SIDE OF THE SURFACE AS X1
            X=XA
            GO TO 3
          ELSE
C           ALLOWING FOR ONE MORE ITERATION
            RATE=0.
          END IF
        END IF
        XB=X
        FB=F(1)
        DB=F(2)*D(3)+F(3)*D(4)+F(4)*D(5)
C
C       NEW APPROXIMATION:
        IF(MOD(ITER,2).EQ.1) THEN
C         REGULA FALSI:
          X=(FA*XB-FB*XA)/(FA-FB)
        ELSE
C         MODIFIED NEWTON-RAPHSON:
          XCA=XA-FA/DA
          XCB=XB-FB/DB
          XC=(XA+XB)/2.
          IF((XCA-XC)*(XCA-XB).LT.0.) THEN
            IF((XCB-XC)*(XCB-XB).LT.0.) THEN
```

```
                IF(ABS(XCA-XB).LT.ABS(XCB-XB)) THEN
                  X=XCA
                ELSE
                  X=XCB
                END IF
              ELSE
                X=XCA
              END IF
            ELSE
              IF((XCB-XC)*(XCB-XB).LT.0.) THEN
                X=XCB
              ELSE
                X=XC
              END IF
            END IF
          END IF
C
C         MODIFICATION FOR ROUNDING ERRORS:
          X=X+SIGN(ERR/50.,XA-XB)
C
C         INTERPOLATION OF THE RAY:
          CALL HIVD2(3,X1,Y1(3),D1(3),X2,Y2(3),D2(3),X,Y(3),D(3))

    2 CONTINUE
C     END OF LOOP FOR ITERATIONS
      STOP 'ERROR 589 IN CROSS: TOO MANY ITERATIONS'
C
    3 CONTINUE
      CALL HIVD2(NDIM,X1,Y1,D1,X2,Y2,D2,X,Y,D)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE HIVD2(NDIM,X1,Y1,D1,X2,Y2,D2,X,Y,D)
      INTEGER NDIM
      REAL X1,Y1(NDIM),D1(NDIM),X2,Y2(NDIM),D2(NDIM)
      REAL X,Y(NDIM),D(NDIM)
C
C THIS SUBROUTINE PERFORMS HERMITE INTERPOLATION OF A VECTOR AND ITS
C DERIVATIVES USING FUNCTIONAL VALUES AND DERIVATIVES AT 2 GIVEN POINTS.
C
C INPUT:
C     NDIM... DIMENSION OF ARRAYS Y1,D1,Y2,D2,Y,D (THE NUMBER OF
C             INDEPENDENT VARIABLES).
C     X1...   INDEPENDENT VARIABLE CORRESPONDING TO THE FIRST GIVEN
C             POINT.
C     Y1...   ARRAY CONTAINING FUNCTIONAL VALUES AT THE FIRST GIVEN
C             POINT.
C     D1...   ARRAY CONTAINING THE DERIVATIVES AT THE FIRST GIVEN POINT.
C     X2...   INDEPENDENT VARIABLE CORRESPONDING TO THE SECOND GIVEN
C             POINT.
C     Y2...   ARRAY CONTAINING FUNCTIONAL VALUES AT THE SECOND GIVEN
C             POINT.
C     D2...   ARRAY CONTAINING THE DERIVATIVES AT THE SECOND GIVEN POINT
C     X...    INDEPENDENT VARIABLE OF THE POINT AT WHICH THE
C             INTERPOLATED VECTOR IS TO BE EVALUATED.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C OUTPUT:
```

```
C     Y...     ARRAY CONTAINING INTERPOLATED FUNCTIONAL VALUES AT X.
C     D...     ARRAY CONTAINING THE DERIVATIVES OF THE INTERPOLATED
C              FUNCTIONAL VALUES AT X.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, OCTOBER 20
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I
      REAL A,B,A1,A2,B1,B2,DA1,DB1,DB2
C
C     SUBSTITUTIONS:
      A=(X-X2)/(X1-X2)
      B=(A-1.)*A
C
C     BASIC FUNCTIONS:
      A1=(A-B-B)*A
      A2=1.-A1
      B1=B*(X-X2)
      B2=B*(X-X1)
C     DERIVATIVES OF BASIC FUNCTIONS:
      DA1=6.*B/(X2-X1)
      DB1=3.*B+A
      DB2=3.*B+1.-A
C
C     INTERPOLATION:
      DO 1 I=1,NDIM
        Y(I)=A1*Y1(I)+A2*Y2(I)+B1*D1(I)+B2*D2(I)
        D(I)=DA1*(Y1(I)-Y2(I))+DB1*D1(I)+DB2*D2(I)
    1 CONTINUE
C
      RETURN
      END
C
C=======================================================================
C
```

```
C          SUBROUTINE HPCG
C
C          PURPOSE
C             TO SOLVE A SYSTEM OF FIRST ORDER ORDINARY GENERAL
C             DIFFERENTIAL EQUATIONS WITH GIVEN INITIAL VALUES.
C
C          USAGE
C             CALL HPCG (PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C             PARAMETERS FCT AND OUTP REQUIRE AN EXTERNAL STATEMENT.
C
C          DESCRIPTION OF PARAMETERS
C             PRMT   - AN INPUT AND OUTPUT VECTOR WITH DIMENSION GREATER
C                      OR EQUAL TO 5, WHICH SPECIFIES THE PARAMETERS OF
C                      THE INTERVAL AND OF ACCURACY AND WHICH SERVES FOR
C                      COMMUNICATION BETWEEN OUTPUT SUBROUTINE (FURNISHED
C                      BY THE USER) AND SUBROUTINE HPCG. EXCEPT PRMT(5)
C                      THE COMPONENTS ARE NOT DESTROYED BY SUBROUTINE
C                      HPCG AND THEY ARE
C             PRMT(1)- LOWER BOUND OF THE INTERVAL (INPUT),
C             PRMT(2)- UPPER BOUND OF THE INTERVAL (INPUT),
C             PRMT(3)- INITIAL INCREMENT OF THE INDEPENDENT VARIABLE
C                      (INPUT),
C             PRMT(4)- UPPER ERROR BOUND (INPUT). IF ABSOLUTE ERROR IS
C                      GREATER THAN PRMT(4), INCREMENT GETS HALVED.
C                      IF INCREMENT IS LESS THAN PRMT(3) AND ABSOLUTE
C                      ERROR LESS THAN PRMT(4)/50, INCREMENT GETS DOUBLED.
C                      THE USER MAY CHANGE PRMT(4) BY MEANS OF HIS
C                      OUTPUT SUBROUTINE.
C             PRMT(5)- NO INPUT PARAMETER. SUBROUTINE HPCG INITIALIZES
C                      PRMT(5)=0. IF THE USER WANTS TO TERMINATE
C                      SUBROUTINE HPCG AT ANY OUTPUT POINT, HE HAS TO
C                      CHANGE PRMT(5) TO NON-ZERO BY MEANS OF SUBROUTINE
C                      OUTP. FURTHER COMPONENTS OF VECTOR PRMT ARE
C                      FEASIBLE IF ITS DIMENSION IS DEFINED GREATER
C                      THAN 5. HOWEVER SUBROUTINE HPCG DOES NOT REQUIRE
C                      AND CHANGE THEM. NEVERTHELESS THEY MAY BE USEFUL
C                      FOR HANDING RESULT VALUES TO THE MAIN PROGRAM
C                      (CALLING HPCG) WHICH ARE OBTAINED BY SPECIAL
C                      MANIPULATIONS WITH OUTPUT DATA IN SUBROUTINE OUTP.
C             Y      - INPUT VECTOR OF INITIAL VALUES.  (DESTROYED)
C                      LATERON Y IS THE RESULTING VECTOR OF DEPENDENT
C                      VARIABLES COMPUTED AT INTERMEDIATE POINTS X.
C             DERY   - INPUT VECTOR OF ERROR WEIGHTS.  (DESTROYED)
C                      THE SUM OF ITS COMPONENTS MUST BE EQUAL TO 1.
C                      LATERON DERY IS THE VECTOR OF DERIVATIVES, WHICH
C                      BELONG TO FUNCTION VALUES Y AT A POINT X.
C             NDIM   - AN INPUT VALUE, WHICH SPECIFIES THE NUMBER OF
C                      EQUATIONS IN THE SYSTEM.
C             IHLF   - AN OUTPUT VALUE, WHICH SPECIFIES THE NUMBER OF
C                      BISECTIONS OF THE INITIAL INCREMENT. IF IHLF GETS
C                      GREATER THAN 10, SUBROUTINE HPCG RETURNS WITH
C                      ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM.
C                      ERROR MESSAGE IHLF=12 OR IHLF=13 APPEARS IN CASE
C                      PRMT(3)=0 OR IN CASE SIGN(PRMT(3)).NE.SIGN(PRMT(2)-
C                      PRMT(1)) RESPECTIVELY.
C             FCT    - THE NAME OF AN EXTERNAL SUBROUTINE USED. IT
C                      COMPUTES THE RIGHT HAND SIDES DERY OF THE SYSTEM
C                      TO GIVEN VALUES OF X AND Y. ITS PARAMETER LIST
C                      MUST BE X,Y,DERY. THE SUBROUTINE SHOULD NOT
C                      DESTROY X AND Y.
```

```
C           OUTP    - THE NAME OF AN EXTERNAL OUTPUT SUBROUTINE USED.
C                     ITS PARAMETER LIST MUST BE X,Y,DERY,IHLF,NDIM,PRMT.
C                     NONE OF THESE PARAMETERS (EXCEPT, IF NECESSARY,
C                     PRMT(4),PRMT(5),...) SHOULD BE CHANGED BY
C                     SUBROUTINE OUTP. IF PRMT(5) IS CHANGED TO NON-ZERO,
C                     SUBROUTINE HPCG IS TERMINATED.
C           AUX     - AN AUXILIARY STORAGE ARRAY WITH 16 ROWS AND NDIM
C                     COLUMNS.
C
C        REMARKS
C           THE PROCEDURE TERMINATES AND RETURNS TO CALLING PROGRAM, IF
C           (1) MORE THAN 10 BISECTIONS OF THE INITIAL INCREMENT ARE
C               NECESSARY TO GET SATISFACTORY ACCURACY (ERROR MESSAGE
C               IHLF=11),
C           (2) INITIAL INCREMENT IS EQUAL TO 0 OR HAS WRONG SIGN
C               (ERROR MESSAGES IHLF=12 OR IHLF=13),
C           (3) THE WHOLE INTEGRATION INTERVAL IS WORKED THROUGH,
C           (4) SUBROUTINE OUTP HAS CHANGED PRMT(5) TO NON-ZERO.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL SUBROUTINES FCT(X,Y,DERY) AND
C           OUTP(X,Y,DERY,IHLF,NDIM,PRMT) MUST BE FURNISHED BY THE USER.
C
C        METHOD
C           EVALUATION IS DONE BY MEANS OF HAMMINGS MODIFIED PREDICTOR-
C           CORRECTOR METHOD. IT IS A FOURTH ORDER METHOD, USING 4
C           PRECEEDING POINTS FOR COMPUTATION OF A NEW VECTOR Y OF THE
C           DEPENDENT VARIABLES.
C           FOURTH ORDER RUNGE-KUTTA METHOD SUGGESTED BY RALSTON IS
C           USED FOR ADJUSTMENT OF THE INITIAL INCREMENT AND FOR
C           COMPUTATION OF STARTING VALUES.
C           SUBROUTINE HPCG AUTOMATICALLY ADJUSTS THE INCREMENT DURING
C           THE WHOLE COMPUTATION BY HALVING OR DOUBLING.
C           TO GET FULL FLEXIBILITY IN OUTPUT, AN OUTPUT SUBROUTINE
C           MUST BE CODED BY THE USER.
C           FOR REFERENCE, SEE
C           (1)  RALSTON/WILF, MATHEMATICAL METHODS FOR DIGITAL
C                COMPUTERS, WILEY, NEW YORK/LONDON, 1960, PP.95-109.
C           (2)  RALSTON, RUNGE-KUTTA METHODS WITH MINIMUM ERROR BOUNDS,
C                MTAC, VOL.16, ISS.80 (1962), PP.431-437.
C
C     ..................................................................
C
      SUBROUTINE HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C
      DIMENSION PRMT(1),Y(1),DERY(1),AUX(16,1)
      N=1
      IHLF=0
      X=PRMT(1)
      H=PRMT(3)
      PRMT(5)=0.
      DO 1 I=1,NDIM
      AUX(16,I)=0.
      AUX(15,I)=DERY(I)
    1 AUX(1,I)=Y(I)
      IF(H*(PRMT(2)-X))3,2,4
C
C     ERROR RETURNS
    2 IHLF=12
      GOTO 4
```

```
    3 IHLF=13
C
C     COMPUTATION OF DERY FOR STARTING VALUES
    4 CALL FCT(X,Y,DERY)
C
C     RECORDING OF STARTING VALUES
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)
      IF(PRMT(5))6,5,6
    5 IF(IHLF)7,7,6
    6 RETURN
    7 DO 8 I=1,NDIM
    8 AUX(8,I)=DERY(I)
C
C     COMPUTATION OF AUX(2,I)
      ISW=1
      GOTO 100
C
    9 X=X+H
      DO 10 I=1,NDIM
   10 AUX(2,I)=Y(I)
C
C     INCREMENT H IS TESTED BY MEANS OF BISECTION
   11 IHLF=IHLF+1
      X=X-H
      DO 12 I=1,NDIM
   12 AUX(4,I)=AUX(2,I)
      H=.5*H
      N=1
      ISW=2
      GOTO 100
C
   13 X=X+H
      CALL FCT(X,Y,DERY)
      N=2
      DO 14 I=1,NDIM
      AUX(2,I)=Y(I)
   14 AUX(9,I)=DERY(I)
      ISW=3
      GOTO 100
C
C     COMPUTATION OF TEST VALUE DELT
   15 DELT=0.
      DO 16 I=1,NDIM
   16 DELT=DELT+AUX(15,I)*ABS(Y(I)-AUX(4,I))
      DELT=.06666667*DELT
      IF(DELT-PRMT(4))19,19,17
   17 IF(IHLF-10)11,18,18
C
C     NO SATISFACTORY ACCURACY AFTER 10 BISECTIONS. ERROR MESSAGE.
   18 IHLF=11
      X=X+H
      GOTO 4
C
C     THERE IS SATISFACTORY ACCURACY AFTER LESS THAN 11 BISECTIONS.
   19 X=X+H
      CALL FCT(X,Y,DERY)
      DO 20 I=1,NDIM
      AUX(3,I)=Y(I)
   20 AUX(10,I)=DERY(I)
      N=3
```

```
      ISW=4
      GOTO 100
C
   21 N=1
      X=X+H
      CALL FCT(X,Y,DERY)
      X=PRMT(1)
      DO 22 I=1,NDIM
      AUX(11,I)=DERY(I)
   22 Y(I)=AUX(1,I)+H*(.375*AUX(8,I)+.7916667*AUX(9,I)
     1-.2083333*AUX(10,I)+.04166667*DERY(I))
   23 X=X+H
      N=N+1
      CALL FCT(X,Y,DERY)
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)
      IF(PRMT(5))6,24,6
   24 IF(N-4)25,200,200
   25 DO 26 I=1,NDIM
      AUX(N,I)=Y(I)
   26 AUX(N+7,I)=DERY(I)
      IF(N-3)27,29,200
C
   27 DO 28 I=1,NDIM
      DELT=AUX(9,I)+AUX(9,I)
      DELT=DELT+DELT
   28 Y(I)=AUX(1,I)+.3333333*H*(AUX(8,I)+DELT+AUX(10,I))
      GOTO 23
C
   29 DO 30 I=1,NDIM
      DELT=AUX(9,I)+AUX(10,I)
      DELT=DELT+DELT+DELT
   30 Y(I)=AUX(1,I)+.375*H*(AUX(8,I)+DELT+AUX(11,I))
      GOTO 23
C
C     THE FOLLOWING PART OF SUBROUTINE HPCG COMPUTES BY MEANS OF
C     RUNGE-KUTTA METHOD STARTING VALUES FOR THE NOT SELF-STARTING
C     PREDICTOR-CORRECTOR METHOD.
  100 DO 101 I=1,NDIM
      Z=H*AUX(N+7,I)
      AUX(5,I)=Z
  101 Y(I)=AUX(N,I)+.4*Z
C     Z IS AN AUXILIARY STORAGE LOCATION
C
      Z=X+.4*H
      CALL FCT(Z,Y,DERY)
      DO 102 I=1,NDIM
      Z=H*DERY(I)
      AUX(6,I)=Z
  102 Y(I)=AUX(N,I)+.2969776*AUX(5,I)+.1587596*Z
C
      Z=X+.4557372*H
      CALL FCT(Z,Y,DERY)
      DO 103 I=1,NDIM
      Z=H*DERY(I)
      AUX(7,I)=Z
  103 Y(I)=AUX(N,I)+.2181004*AUX(5,I)-3.050965*AUX(6,I)+3.832865*Z
C
      Z=X+H
      CALL FCT(Z,Y,DERY)
      DO 104 I=1,NDIM
```

```
 1040Y(I)=AUX(N,I)+.1747603*AUX(5,I)-.5514807*AUX(6,I)
     1+1.205536*AUX(7,I)+.1711848*H*DERY(I)
      GOTO(9,13,15,21),ISW
C
C     POSSIBLE BREAK-POINT FOR LINKAGE
C
C     STARTING VALUES ARE COMPUTED.
C     NOW START HAMMINGS MODIFIED PREDICTOR-CORRECTOR METHOD.
  200 ISTEP=3
  201 IF(N-8)204,202,204
C
C     N=8 CAUSES THE ROWS OF AUX TO CHANGE THEIR STORAGE LOCATIONS
  202 DO 203 N=2,7
      DO 203 I=1,NDIM
      AUX(N-1,I)=AUX(N,I)
  203 AUX(N+6,I)=AUX(N+7,I)
      N=7
C
C     N LESS THAN 8 CAUSES N+1 TO GET N
  204 N=N+1
C
C     COMPUTATION OF NEXT VECTOR Y
      DO 205 I=1,NDIM
      AUX(N-1,I)=Y(I)
  205 AUX(N+6,I)=DERY(I)
      X=X+H
  206 ISTEP=ISTEP+1
      DO 207 I=1,NDIM
     0DELT=AUX(N-4,I)+1.333333*H*(AUX(N+6,I)+AUX(N+6,I)-AUX(N+5,I)+
     1AUX(N+4,I)+AUX(N+4,I))
      Y(I)=DELT-.9256198*AUX(16,I)
  207 AUX(16,I)=DELT
C     PREDICTOR IS NOW GENERATED IN ROW 16 OF AUX, MODIFIED PREDICTOR
C     IS GENERATED IN Y. DELT MEANS AN AUXILIARY STORAGE.
C
      CALL FCT(X,Y,DERY)
C     DERIVATIVE OF MODIFIED PREDICTOR IS GENERATED IN DERY
C
      DO 208 I=1,NDIM
     0DELT=.125*(9.*AUX(N-1,I)-AUX(N-3,I)+3.*H*(DERY(I)+AUX(N+6,I)+
     1AUX(N+6,I)-AUX(N+5,I)))
      AUX(16,I)=AUX(16,I)-DELT
  208 Y(I)=DELT+.07438017*AUX(16,I)
C
C     TEST WHETHER H MUST BE HALVED OR DOUBLED
      DELT=0.
      DO 209 I=1,NDIM
  209 DELT=DELT+AUX(15,I)*ABS(AUX(16,I))
      IF(DELT-PRMT(4))210,222,222
C
C     H MUST NOT BE HALVED. THAT MEANS Y(I) ARE GOOD.
  210 CALL FCT(X,Y,DERY)
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)
      IF(PRMT(5))212,211,212
  211 IF(IHLF-11)213,212,212
  212 RETURN
  213 IF(H*(X-PRMT(2)))214,212,212
  214 IF(ABS(X-PRMT(2))-.1*ABS(H))212,215,215
  215 IF(DELT-.02*PRMT(4))216,216,201
C
```

```
C
C     H COULD BE DOUBLED IF ALL NECESSARY PRECEEDING VALUES ARE
C     AVAILABLE
  216 IF(IHLF)201,201,217
  217 IF(N-7)201,218,218
  218 IF(ISTEP-4)201,219,219
  219 IMOD=ISTEP/2
      IF(ISTEP-IMOD-IMOD)201,220,201
  220 H=H+H
      IHLF=IHLF-1
      ISTEP=0
      DO 221 I=1,NDIM
      AUX(N-1,I)=AUX(N-2,I)
      AUX(N-2,I)=AUX(N-4,I)
      AUX(N-3,I)=AUX(N-6,I)
      AUX(N+6,I)=AUX(N+5,I)
      AUX(N+5,I)=AUX(N+3,I)
      AUX(N+4,I)=AUX(N+1,I)
      DELT=AUX(N+6,I)+AUX(N+5,I)
      DELT=DELT+DELT+DELT
  2210AUX(16,I)=8.962963*(Y(I)-AUX(N-3,I))-3.361111*H*(DERY(I)+DELT
     1+AUX(N+4,I))
      GOTO 201
C
C
C     H MUST BE HALVED
  222 IHLF=IHLF+1
      IF(IHLF-10)223,223,210
  223 H=.5*H
      ISTEP=0
      DO 224 I=1,NDIM
     0Y(I)=.00390625*(80.*AUX(N-1,I)+135.*AUX(N-2,I)+40.*AUX(N-3,I)+
     1AUX(N-4,I))-.1171875*(AUX(N+6,I)-6.*AUX(N+5,I)-AUX(N+4,I))*H
     0AUX(N-4,I)=.00390625*(12.*AUX(N-1,I)+135.*AUX(N-2,I)+
     1108.*AUX(N-3,I)+AUX(N-4,I))-.0234375*(AUX(N+6,I)+18.*AUX(N+5,I)-
     29.*AUX(N+4,I))*H
      AUX(N-3,I)=AUX(N-2,I)
  224 AUX(N+4,I)=AUX(N+5,I)
      X=X-H
      DELT=X-(H+H)
      CALL FCT(DELT,Y,DERY)
      DO 225 I=1,NDIM
      AUX(N-2,I)=Y(I)
      AUX(N+5,I)=DERY(I)
  225 Y(I)=AUX(N-4,I)
      DELT=DELT-(H+H)
      CALL FCT(DELT,Y,DERY)
      DO 226 I=1,NDIM
      DELT=AUX(N+5,I)+AUX(N+4,I)
      DELT=DELT+DELT+DELT
     0AUX(16,I)=8.962963*(AUX(N-1,I)-Y(I))-3.361111*H*(AUX(N+6,I)+DELT
     1+DERY(I))
  226 AUX(N+3,I)=DERY(I)
      GOTO 206
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'TRANS' TO TRANSFORM THE QUANTITIES AT AN INTERFACE
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C     TRANS...SUBROUTINE TRANSFORMING THE COMPUTED QUANTITIES ACROSS A
C             CURVED INTERFACE, SEE C.R.T.5.9.
C     CONV... SUBROUTINE DESIGNED TO REPLACE THE AMPLITUDES Y(28) TO
C             Y(IY(1)) EXPRESSED IN THE RAY-CENTRED COORDINATE SYSTEM BY
C             THE AMPLITUDES INVOLVING APPROPRIATE CONVERSION
C             COEFFICIENTS, SEE C.R.T.5.5.4.
C
C=======================================================================
C
      SUBROUTINE TRANS(YL,Y,YY,IY,KODNEW,ICBNEW,IEND)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12),ICBNEW,IEND
C
C THIS SUBROUTINE TRANSFORMS THE COMPUTED QUANTITIES ACROSS A CURVED
C INTERFACE, SEE C.R.T.5.9.
C
C INPUT:
C     YL...    ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF
C              INCIDENCE.
C     Y...     ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY,
C              CORRESPONDING TO THE INCIDENT WAVE.
C     YY...    ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C              THE RAY, CORRESPONDING TO THE INCIDENT WAVE.
C     IY...    ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C              ALONG THE RAY, CORRESPONDING TO THE INCIDENT WAVE.
C     KODNEW..POSITION IN THE CODE (INDEX IN THE ARRAY KODE)
C              CORRESPONDING TO THE NEXT ELEMENT OF THE RAY.  OUTPUT FROM
C              SUBROUTINE CODE.
C     ICBNEW..THE INDEX OF THE COMPLEX BLOCK IN WHICH THE NEXT ELEMENT
C              OF THE RAY IS TO BE SITUATED, SUPPLEMENTED BY THE SIGN '+'
C              FOR P WAVE OR '-' FOR S WAVE. OUTPUT FROM SUBROUTINE CODE.
C
C OUTPUT:
C     YL...    ARRAY CONTAINING LOCAL QUANTITIES AT THE R/T POINT,
C              CORRESPONDING TO THE GENERATED WAVE.
C     Y...     ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY,
C              CORRESPONDING TO THE GENERATED WAVE.
C     YY...    ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C              THE RAY, CORRESPONDING TO THE GENERATED WAVE.  UNCHANGED
C              INPUT VALUES.
C     IY...    ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C              ALONG THE RAY, CORRESPONDING TO THE GENERATED WAVE.
C     IEND...  INDICATION OF THE TERMINATION OF THE RAY COMPUTATION (SEE
C                C.R.T.5.4).
C              IEND.EQ.0... COMPUTATION OF THE RAY CONTINUES.
C              IEND.NE.0... THE COMPUTATION OF THE RAY TERMINATES,
C                DIFFERENT VALUES OF IEND MAY SPECIFY THE REASON OF THE
C                TERMINATION:
C              24... TRANSMISSION IS REQUIRED BY THE CODE AT A FREE
C                SURFACE.
C              25... RAY OF THE REQUIRED REFLECTED OR TRANSMITTED WAVE
C                IS NOT REAL-VALUED (OVERCRITICAL REFLECTION OR
C                TRANSMISSION).
C              26... S WAVE IN A LIQUID BLOCK IS REQUIRED BY THE CODE.
C              32... REFLECTION OR TYPE CONVERSION AT THE FICTITIOUS
```

```
C                 PART OF THE INTERFACE.  HERE, THE INTERFACE IS
C                 CONSIDERED TO BE FICTITIOUS IF P AND S WAVE VELOCITIES
C                 AND THE DENSITY ARE THE SAME AT BOTH SIDES OF THE
C                 INTERFACE AT THE POINT OF INCIDENCE.
C              33... ZERO REFLECTION/TRANSMISSION COEFFICIENT.
C              37... THE NEXT ELEMENT OF THE RAY IS REQUIRED BY THE CODE
C                 TO BE SITUATED IN A COMPLEX BLOCK THAT DOES NOT TOUCH
C                 THE POINT OF INCIDENCE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL KOOR,METRIC,SRFC2,PARM2,VELOC,SMVPRD,COEF50,COEFSH
      INTEGER KOOR
C     FUNCTION KOOR()... SUBROUTINE PACKAGE 'METRIC'.
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)... SUBROUTINE PACKAGE
C           'METRIC'.
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C           SUBROUTINE PACKAGE 'SRFC'.
C     SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C           ... SUBROUTINE PACKAGE 'PARM'.
C     SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)
C     SUBROUTINE SMVPRD(G,A1,A2,A3,B1,B2,B3)... SUBROUTINE PACKAGE
C           'RAYCB'.
C     SUBROUTINE COEF50(P,VP1,VS1,RO1,VP2,VS2,RO2,NCODE,ND,RMOD,RPH)...
C           SUBROUTINE PACKAGE 'COEF'.
C     SUBROUTINE COEFSH(P,VS1,RO1,VS2,RO2,NCODE,RMOD,RPH)... SUBROUTINE
C           PACKAGE 'COEF'.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS FOR LOCAL MODEL PARAMETERS:
      REAL G(12),GAMMA(18),GSQRD,FAUX(10)
      REAL UP(10),US(10),RO,QP,QS,VP,VS,VD(10),QL
C     THESE AUXILIARY VARIABLES AND ARRAYS NEED NOT BE LOCATED IN A
C     COMMON BLOCK.  THERE IS NO REASON TO LOCATE THEM IN THE AUXILIARY
C     COMMON BLOCK /AUXMOD/ BUT TO SHARE THE MEMORY.
      COMMON/AUXMOD/ G,GAMMA,GSQRD,FAUX,UP,US,RO,QP,QS,VP,VS,VD,QL
C
C     G,GAMMA,GSQRD... SEE SUBROUTINE METRIC OF THE PACKAGE 'METRIC'.
C     FAUX... AUXILIARY ARRAY TO STORE A FUNCTIONAL VALUE AND ITS
C           DERIVATIVES.
C     UP,US,RO,QP,QS... SEE SUBROUTINE PARM2 OF THE PACKAGE 'PARM'.
C     UP,US,QP,QS,VP,VS,VD,QL... SEE SUBROUTINE VELOC OF THE PACKAGE
C           'MODEL'.
C
C.......................................................................
C
C     AUXILIARY STORAGE LOCATIONS:
C
      INTEGER NAMPL1,NAMPL2,ICB1,ICB2,I,J,NCODE,ND
      REAL VP1,VS1,RO1,VP2,VS2,RO2,V1,VD1(4),V2,VD2(10)
      REAL H11,H13,H43,HHH,Z11,Z12,Z13,Z41,Z42,Z43,ZZZ
      REAL H21,H23,H53,    Z21,Z22,Z23,Z51,Z52,Z53
      REAL H31,H33,H63,    Z31,Z32,Z33,Z61,Z62,Z63
      REAL E11,E21,E31,E13,E23,E33,C1,S1,C2,S2,C,S
      REAL V11,V21,V31,V12,V22,V32,D11,D13,D33,AUX1,AUX2,AUX3
      REAL P,AUX,E12,CFC11,CFC12,CFC22,Q1,Q2,P1,P2
      REAL COEF,RMOD,RPH,RMODSH,RPHSH,RR,YR
```

```
      REAL R1A1,Y1A1,R2A1,Y2A1,R1B1,Y1B1,R2B1,Y2B1
      REAL R1A2,Y1A2,R2A2,Y2A2,R1B2,Y1B2,R2B2,Y2B2

C     NAMPL1..NUMBER OF THE APMLITUDE COEFFICIENTS OF THE INCIDENT WAVE
C             (5.9.1A).
C     NAMPL2..NUMBER OF THE APMLITUDE COEFFICIENTS OF THE R/T WAVE
C             (5.9.1B).
C     ICB1... INDEX OF A COMPLEX BLOCK OF THE INCIDENT WAVE (5.9.1E).
C     ICB2... INDEX OF A COMPLEX BLOCK AT THE OTHER SIDE (5.9.1E).
C     I,J...  TEMPORARY STORAGE LOCATIONS (5.9.5).
C     NCODE,ND... TYPE OF THE R/T COEFFICIENT (5.9.6).
C     VP1,VS1,RO1... PARAMETERS OF THE COMPLEX BLOCK IABS(ICB1) (5.9.2).
C     VP2,VS2,RO2... PARAMETERS OF THE COMPLEX BLOCK ICB2 (5.9.2).
C     V1,VD1(2:4)... INCIDENT WAVE VELOCITY AND ITS DERIVATIVES (5.9.2).
C     V2,VD2(2:4)...R/T WAVE VELOCITY AND ITS DERIVATIVES (5.9.2).
C     VD2(1),VD2(5:10)... TEMPORARY STORAGE LOCATIONS (5.9.2).
C     H11,H21,H31,H13,H23,H33... COVARIANT COMPONENTS OF THE BASIS
C             VECTORS OF RAY-CENTRED COORDINATE SYSTEM OF THE INCIDENT
C             WAVE (5.9.3).
C     H43,H53,H63... CONTRAVARIANT COMPONENTS OF THE SLOWNESS VECTOR OF
C             THE INCIDENT WAVE (5.9.3).  NOT USED IN CARTESIAN
C             COORDINATES.
C     HHH...  NORM OF THE SLOWNESS VECTOR (5.9.3).
C     Z11,Z21,Z31,Z12,Z22,Z32,Z13,Z23,Z33... COVARIANT COMPONENTS OF THE
C             LOCAL CARTESIAN COORDINATE SYSTEM (5.9.3).
C     Z41,Z51,Z61,Z42,Z52,Z62,Z43,Z53,Z63... CONTRAVARIANT COMPONENTS OF
C             THE LOCAL CARTESIAN COORDINATE SYSTEM (5.9.3).
C     ZZZ...  NORM OF THE GRADIENT OF THE FUNCTION DESCRIBING THE
C             INTERFACE (5.9.3,5.9.4).
C     E11,E21,E31... COVARIANT COMPONENTS OF THE UNIT VECTOR
C             PERPENDICULAR TO THE R/T RAY IN THE PLANE OF INCIDENCE
C             (5.9.3).
C             E11 IS ALSO A TEMPORARY STORAGE LOCATION IN (5.9.4).
C     E13,E23,E33... COVARIANT COMPONENTS OF THE UNIT VECTOR TANGENT TO
C             THE R/T RAY (5.9.3).
C     C1,S1...COSINE AND SINE OF THE ANGLE OF INCIDENCE (5.9.3,5.9.5).
C     C2,S2...COSINE AND SINE OF THE R/T ANGLE (5.9.3,5.9.5).
C     C,S...  COSINE AND SINE OF THE REVOLUTION ANGLE (5.9.3,5.9.5,5.9.6
C     V11,V21,V31... VELOCITY GRADIENT IN LOCAL CARTESIAN COORDINATES
C             CORRESPONDING TO THE INCIDENT WAVE (5.9.4).
C     V12,V22,V32... VELOCITY GRADIENT IN LOCAL CARTESIAN COORDINATES
C             CORRESPONDING TO THE INCIDENT WAVE (5.9.4).
C     D11,D13,D33... MATRIX OF THE CURVATURE OF THE INTERFACE (5.9.4).
C     AUX1,AUX2,AUX3... TEMPORARY STORAGE LOCATIONS (5.9.4).
C     P...    SNELL RAY PARAMETER(5.9.5,5.9.6)
C     AUX,E12,CFC11,CFC12,CFC22,Q1,Q2,P1,P2... TEMPORARY STORAGE
C             LOCATIONS (5.9.5).
C     COEF... (5.9.6).
C     RMOD,RPH... P-SV R/T COEFFICIENT (5.9.6).
C     RMODSH,RPHSH... SH R/T COEFFICIENT (5.9.6).
C     RR,YR...REAL AND IMAGINARY PART OF A COMPLEX-VALUED R/T
C             COEFFICIENT (5.9.6).
C     ( R1A1,Y1A1,R2A1,Y2A1 )
C     ( R1B1,Y1B1,R2B1,Y2B1 )... AMPLIDUDE COEFFICIENTS (5.9.6):
C     ( R1A2,Y1A2,R2A2,Y2A2 )
C     ( R1B2,Y1B2,R2B2,Y2B2 )                INITIAL
C             PART:           COMPONENT:     POLARISATION:   WAVE:
C             R...REAL        1...P-SV       A...FIRST       1...INCIDENT
C             Y...IMAGINARY   2...SH         B...SECOND      2...R/T
C
```

```
C.......................................................................
C
C     (5.9.1) TRANSFORMATION OF AUXILIARY QUANTITIES, TRAVEL TIME AND
C     COORDINATES:
C     (A) NUMBER OF REAL-VALUED QUANTITIES DESCRIBING THE REDUCED
C     AMPLITUDES OF THE INCIDENT WAVE:
      NAMPL1=IY(1)-27
C     (B) NUMBER OF REAL-VALUED QUANTITIES DESCRIBING THE REDUCED
C     AMPLITUDES OF THE GENERATED WAVE:
      IF(IY(5).GT.0.AND.ICBNEW.LT.0) THEN
        NAMPL2=NAMPL1*2
      ELSE IF(IY(5).LT.0.AND.ICBNEW.GT.0) THEN
        NAMPL2=NAMPL1/2
      ELSE
        NAMPL2=NAMPL1
      END IF
C     (C) OUTPUT NUMBER OF BASIC QUANTITIES:
      IY(1)=27+NAMPL2
C     (D) NEW POSITION IN THE CODE:
      IY(2)=KODNEW
C     (E) INDICES OF SIMPLE AND COMPLEX BLOCKS:
      ICB1=IY(5)
      ICB2=IABS(IY(8))
      IF(IABS(ICBNEW).EQ.ICB2) THEN
C       TRANSMISSION:
        IY(3)=IABS(IY(5))
        IY(4)=IY(7)
        IY(5)=ICBNEW
        IY(6)=-IY(6)
      ELSE IF(IABS(ICBNEW).EQ.IABS(IY(5))) THEN
C       REFLECTION:
        IY(5)=ICBNEW
      ELSE
C       REQUIRED COMPLEX BLOCK IS NOT ATTAINABLE:
        IEND=37
        RETURN
      END IF
C     (F) IY(7), IY(8) MAY BE UNDEFINED
      IY(7)=0
      IY(8)=0
C     (G) NUMBER OF TRANSFORMATIONS AT AN INTERFACE:
      IY(11)=IY(11)+1
C     (H) IY(9), IY(10), IY(12), YY(1:5) ARE UNCHANGED
C     (I) TRAVEL TIME AND COORDINATES Y(1:5) REMAIN UNCHANGED
C
C     (5.9.2) VELOCITIES (METRIC TENSOR IS DETERMINED LATER ON)
C     (B) MATERIAL PARAMETERS CORRESPONDING TO THE INCIDENT WAVE:
      VP1=YL(1)
      VS1=YL(2)
      RO1=YL(3)
      VD1(2)=YL(4)
      VD1(3)=YL(5)
      VD1(4)=YL(6)
      IF(ICB1.GE.0) THEN
        V1=VP1
      ELSE
        V1=VS1
      END IF
C     (C) MATERIAL PARAMETERS ON THE OTHER SIDE OF THE INTERFACE:
      IF(ICB2.NE.0) THEN
```

```
          CALL PARM2(ICB2,Y(3),UP,US,RO2,QP2,QS2)
          CALL VELOC(ISIGN(ICB2,ICBNEW),UP,US,QP2,QS2,VP2,VS2,VD2,QL)
        ELSE
C         FREE SPACE:
          RO2=0.
        END IF
        IF(VP1.EQ.VP2) THEN
          IF(ICBNEW.NE.ISIGN(ICB2,ICB1)) THEN
C           TRANSMISSION WITHOUT CONVERSION IS EXCLUDED
            IF(VS1.EQ.VS2.AND.RO1.EQ.RO2) THEN
C             REFLECTION OR WAVE TYPE CONVERSION AT A FICTITIOUS INTERFACE
              IEND=32
              RETURN
            END IF
          END IF
        END IF
C       (D) MATERIAL PARAMETERS CORRESPONDING TO THE GENERATED WAVE:
        IF(ICBNEW.EQ.ICB1) THEN
C         REFLECTION WITHOUT CONVERSION:
          V2=V1
          VD2(2)=VD1(2)
          VD2(3)=VD1(3)
          VD2(4)=VD1(4)
        ELSE IF(ICBNEW.EQ.-ICB1) THEN
C         REFLECTION WITH CONVERSION:
          CALL PARM2(IABS(ICBNEW),Y(3),UP,US,AUX,QP,QS)
          CALL VELOC(ICBNEW,UP,US,QP,QS,AUX,AUX,VD2,QL)
          V2=VD2(1)
        ELSE
C         TRANSMISSION:
          V2=VD2(1)
          IF(RO2.EQ.0.) THEN
C           TRANSMISSION INTO FREE SPACE:
            IEND=24
            RETURN
          END IF
          IF(V2.EQ.0.) THEN
C           ZERO VELOCITY (S-WAVE IN LIQUID):
            IEND=26
            RETURN
          END IF
          YL(1)=VP2
          YL(2)=VS2
          YL(3)=RO2
        END IF
        YL(4)=VD2(2)
        YL(5)=VD2(3)
        YL(6)=VD2(4)
C
C       (5.9.3) TRANSFORMATION OF THE SLOWNESS VECTOR AND OF THE
C       BASIS OF THE RAY-CENTRED COORDINATE SYSTEM:
        CALL SRFC2(IABS(IY(6)),Y(3),FAUX)
C       NON-UNIT VECTORS - COVARIANT COMPONENTS
        H13=Y(6)
        H23=Y(7)
        H33=Y(8)
        H11=Y(9)
        H21=Y(10)
        H31=Y(11)
        Z13=FAUX(2)
```

```
      Z23=FAUX(3)
      Z33=FAUX(4)
C     (C) NON-UNIT VECTORS - CONTRAVARIANT COMPONENTS
      IF(KOOR().EQ.0) THEN
C       CARTESIAN COORDINATES:
C       NORM OF THE SLOWNESS VECTOR
        HHH=SQRT(H13*H13+H23*H23+H33*H33)
C       NORM OF THE GRADIENT OF THE FUNCTION DESCRIBING THE INTERFACE
        ZZZ=SQRT(Z13*Z13+Z23*Z23+Z33*Z33)
C       UNIT NORMAL TO THE INTERFACE:
C       COVARIANT COMPONENTS
        Z13=Z13/ZZZ
        Z23=Z23/ZZZ
        Z33=Z33/ZZZ
C       CONTRAVARIANT COMPONENTS
        Z43=Z13
        Z53=Z23
        Z63=Z33
      ELSE
C       CURVILINEAR COORDINATES:
        CALL METRIC(Y(3),GSQRD,G,GAMMA)
C       SECOND COVARIANT DERIVATIVES OF THE INTERFACE FUNCTION
        FAUX(5)=FAUX(5)
     *          -GAMMA(1)*FAUX(2)-GAMMA(7)*FAUX(3)-GAMMA(13)*FAUX(4)
        FAUX(6)=FAUX(6)
     *          -GAMMA(2)*FAUX(2)-GAMMA(8)*FAUX(3)-GAMMA(14)*FAUX(4)
        FAUX(7)=FAUX(7)
     *          -GAMMA(3)*FAUX(2)-GAMMA(9)*FAUX(3)-GAMMA(15)*FAUX(4)
        FAUX(8)=FAUX(8)
     *          -GAMMA(4)*FAUX(2)-GAMMA(10)*FAUX(3)-GAMMA(16)*FAUX(4)
        FAUX(9)=FAUX(9)
     *          -GAMMA(5)*FAUX(2)-GAMMA(11)*FAUX(3)-GAMMA(17)*FAUX(4)
        FAUX(10)=FAUX(10)
     *          -GAMMA(6)*FAUX(2)-GAMMA(12)*FAUX(3)-GAMMA(18)*FAUX(4)
C       SLOWNESS VECTOR - CONTRAVARIANT COMPONENTS
        CALL SMVPRD(G(7),H13,H23,H33,H43,H53,H63)
C       NORM OF THE SLOWNESS VECTOR
        HHH=SQRT(H13*H43+H23*H53+H33*H63)
C       CONTRAVARIANT GRADIENT OF THE FUNCTION DESCRIBING THE INTERFACE
        CALL SMVPRD(G(7),Z13,Z23,Z33,Z43,Z53,Z63)
C       NORM OF THE GRADIENT OF THE FUNCTION DESCRIBING THE INTERFACE
        ZZZ=SQRT(Z13*Z43+Z23*Z53+Z33*Z63)
C       UNIT NORMAL TO THE INTERFACE:
C       COVARIANT COMPONENTS
        Z13=Z13/ZZZ
        Z23=Z23/ZZZ
        Z33=Z33/ZZZ
C       CONTRAVARIANT COMPONENTS
        Z43=Z43/ZZZ
        Z53=Z53/ZZZ
        Z63=Z63/ZZZ
      END IF
C     UNIT VECTOR TANGENT TO THE INCIDENT RAY
      H13=H13/HHH
      H23=H23/HHH
      H33=H33/HHH
C     COSINE OF THE ANGLE OF INCIDENCE
      C1=H13*Z43+H23*Z53+H33*Z63
C     VECTOR TANGENT TO THE INTERFACE IN THE PLANE OF INCIDENCE
      Z11=H13-Z13*C1
```

```
      Z21=H23-Z23*C1
      Z31=H33-Z33*C1
      IF(KOOR().EQ.0) THEN
C       CARTESIAN COORDINATES:
        AUX=Z11*Z11+Z21*Z21+Z31*Z31
      ELSE
C       CURVILINEAR COORDINATES:
        Z41=H43/HHH-Z43*C1
        Z51=H53/HHH-Z53*C1
        Z61=H63/HHH-Z63*C1
        AUX=Z11*Z41+Z21*Z51+Z31*Z61
      END IF
C     SINE OF THE ANGLE OF INCIDENCE
      IF(AUX.LT.0.5) THEN
        S1=SQRT(AUX)
      ELSE
        S1=SQRT(1.-C1*C1)
      END IF
      IF(S1.LT.0.0001) THEN
C       CORRECTION FOR NEARLY NORMAL INCIDENCE
        Z11=H11
        Z21=H21
        Z31=H31
      ELSE
        Z11=Z11/S1
        Z21=Z21/S1
        Z31=Z31/S1
      END IF
C     ANGLE OF REFLECTION/TRANSMISSION
      S2=S1*V2/V1
      C2=1.-S2*S2
      IF(C2.LE.0.) THEN
C       OVERCRITICAL RAY
        IEND=25
        RETURN
      ELSE
        C2=SIGN(SQRT(C2),C1)
      END IF
      IF(IABS(ICBNEW).EQ.IABS(ICB1)) C2=-C2
      IF(KOOR().EQ.0) THEN
C       CARTESIAN COORDINATES:
C       VECTORS TANGENT TO THE INTERFACE:
C       VECTOR IN THE PLANE OF INCIDENCE - CONTRAVARIANT COMPONENTS
        Z41=Z11
        Z51=Z21
        Z61=Z31
C       VECTOR PERPENDICULAR TO THE PLANE OF INCIDENCE - COVARIANT COMP.
        Z12=Z23*Z31-Z33*Z21
        Z22=Z33*Z11-Z13*Z31
        Z32=Z13*Z21-Z23*Z11
C       VECTOR PERPENDICULAR TO THE PLANE OF INCIDENCE - CONTRAVARIANT
        Z42=Z12
        Z52=Z22
        Z62=Z32
      ELSE
C       CURVILINEAR COORDINATES:
C       VECTORS TANGENT TO THE INTERFACE:
C       VECTOR IN THE PLANE OF INCIDENCE - CONTRAVARIANT COMPONENTS
        CALL SMVPRD(G(7),Z11,Z21,Z31,Z41,Z51,Z61)
C       VECTOR PERPENDICULAR TO THE PLANE OF INCIDENCE - COVARIANT COMP.
```

```
          Z12=(Z53*Z61-Z63*Z51)*GSQRD
          Z22=(Z63*Z41-Z43*Z61)*GSQRD
          Z32=(Z43*Z51-Z53*Z41)*GSQRD
C         VECTOR PERPENDICULAR TO THE PLANE OF INCIDENCE - CONTRAVARIANT
          Z42=(Z23*Z31-Z33*Z21)/GSQRD
          Z52=(Z33*Z11-Z13*Z31)/GSQRD
          Z62=(Z13*Z21-Z23*Z11)/GSQRD
        END IF
C       REVOLUTION ANGLE OF THE RAY CENTRED COORDINATE SYSTEM
        C=(H11*Z41+H21*Z51+H31*Z61)/C1
        S=-H11*Z42-H21*Z52-H31*Z62
C       UNIT VECTOR PERPENDICULAR TO THE R/T RAY IN THE PLANE OF INCIDENCE
        E11=Z11*C2-Z13*S2
        E21=Z21*C2-Z23*S2
        E31=Z31*C2-Z33*S2
C       UNIT VECTOR TANGENT TO THE R/T RAY
        E13=Z11*S2+Z13*C2
        E23=Z21*S2+Z23*C2
        E33=Z31*S2+Z33*C2
C       SLOWNESS VECTOR
        Y(6)=E13/V2
        Y(7)=E23/V2
        Y(8)=E33/V2
C       FIRST BASIS VECTOR OF THE RAY CENTRED COORDINATE SYSTEM
        Y(9)=E11*C-Z12*S
        Y(10)=E21*C-Z22*S
        Y(11)=E31*C-Z32*S
C
C       (5.9.4) CURVATURE OF THE INTERFACE AND VELOCITY GRADIENTS IN THE
C       LOCAL CARTESIAN COORDINATE SYSTEM:
        V11=VD1(2)*Z41+VD1(3)*Z51+VD1(4)*Z61
        V21=VD1(2)*Z42+VD1(3)*Z52+VD1(4)*Z62
        V31=VD1(2)*Z43+VD1(3)*Z53+VD1(4)*Z63
        V12=VD2(2)*Z41+VD2(3)*Z51+VD2(4)*Z61
        V22=VD2(2)*Z42+VD2(3)*Z52+VD2(4)*Z62
        V32=VD2(2)*Z43+VD2(3)*Z53+VD2(4)*Z63
        CALL SMVPRD(FAUX(5),Z41,Z51,Z61,AUX1,AUX2,AUX3)
        D11=(AUX1*Z41+AUX2*Z51+AUX3*Z61)/ZZZ
        D12=(AUX1*Z42+AUX2*Z52+AUX3*Z62)/ZZZ
        CALL SMVPRD(FAUX(5),Z42,Z52,Z62,AUX1,AUX2,AUX3)
        D22=(AUX1*Z42+AUX2*Z52+AUX3*Z62)/ZZZ
C
C       (5.9.5) DYNAMIC RAY TRACING ACROSS A CURVED INTERFACE
        P=S1/V1
        AUX=C1/V1-C2/V2
        E11=(S1*C1*V31-(1.+C1*C1)*V11)/V1-(S2*C2*V32-(1.+C2*C2)*V12)/V2
        E12=V22/V2-V21/V1
        CFC11=(AUX*D11+P*E11)/C2/C2
        CFC12=(AUX*D12+P*E12)/C2
        CFC22= AUX*D22
        AUX=C1/C2
        DO 51 I=1,4
          J=4*I
          Q1= C*Y(8+J)+S*Y(9+J)
          Q2=-S*Y(8+J)+C*Y(9+J)
          P1= C*Y(10+J)+S*Y(11+J)
          P2=-S*Y(10+J)+C*Y(11+J)
          Q1=Q1/AUX
          P1=P1*AUX+CFC11*Q1+CFC12*Q2
          P2=P2    +CFC12*Q1+CFC22*Q2
```

```
      Y(8+J)=C*Q1-S*Q2
      Y(9+J)=S*Q1+C*Q2
      Y(10+J)=C*P1-S*P2
      Y(11+J)=S*P1+C*P2
   51 CONTINUE
C
C     (5.9.6) TRANSFORMATION OF REDUCED AMPLITUDES
      IF(IABS(ICBNEW).EQ.IABS(ICB1)) THEN
C       REFLECTION:
        NCODE=1
        COEF=1.
      ELSE
C       TRANSMISSION:
        NCODE=3
        COEF=RO2/RO1
      END IF
      COEF=SQRT(COEF*ABS(C2/C1)*V2/V1)
      IF(C1.GE.0) THEN
C       EPSILON=+1. (SEE C.R.T.5.9.7)
        ND=0
      ELSE
C       EPSILON=-1. (SEE C.R.T.5.9.7)
        ND=1
      END IF
      RMODSH=0.
      R2A2=0.
      Y2A2=0.
      R2B2=0.
      Y2B2=0.
      IF(ICB1.GE.0) THEN
C       INCIDENT P WAVE:
        IF(ICBNEW.LT.0) THEN
C         OUTGOING S WAVE:
          NCODE=NCODE+1
        END IF
        R1A1=Y(28)
        Y1A1=Y(29)
        IF(NAMPL1.GT.2) THEN
          R1B1= Y(30)
          Y1B1= Y(31)
        END IF
      ELSE
C       INCIDENT S WAVE:
        IF(ICB2.EQ.0) THEN
          NCODE=NCODE+2
        ELSE
          NCODE=NCODE+4
        END IF
        R1A1= C*Y(28)+S*Y(30)
        Y1A1= C*Y(29)+S*Y(31)
        IF(NAMPL1.GT.4) THEN
          R1B1= C*Y(32)+S*Y(34)
          Y1B1= C*Y(33)+S*Y(35)
        END IF
        IF(ICBNEW.LT.0) THEN
C         OUTGOING S WAVE:
          NCODE=NCODE+1
          IF(ICBNEW.EQ.ICB1) THEN
            CALL COEFSH(P,VS1,RO1,VS2,RO2,1,RMODSH,RPHSH)
          ELSE
```

```
              CALL COEFSH(P,VS1,RO1,VS2,RO2,2,RMODSH,RPHSH)
            END IF
            RMODSH=COEF*RMODSH
            RR=RMODSH*COS(RPHSH)
            YR=RMODSH*SIN(RPHSH)
            R2A1=-S*Y(28)+C*Y(30)
            Y2A1=-S*Y(29)+C*Y(31)
            R2A2=RR*R2A1-YR*Y2A1
            Y2A2=YR*R2A1+RR*Y2A1
            IF(NAMPL1.GT.4) THEN
              R2B1=-S*Y(32)+C*Y(34)
              Y2B1=-S*Y(33)+C*Y(35)
              R2B2=RR*R2B1-YR*Y2B1
              Y2B2=YR*R2B1+RR*Y2B1
            END IF
          END IF
        END IF
        CALL COEF50(P,VP1,VS1,RO1,VP2,VS2,RO2,NCODE,ND,RMOD,RPH)
        RMOD=COEF*RMOD
        IF(RMOD.EQ.0..AND.RMODSH.EQ.0.) THEN
C         ZERO REFLECTION/TRANSMISSION COEFFICIENT
          IEND=33
          RETURN
        END IF
        RR=RMOD*COS(RPH)
        YR=RMOD*SIN(RPH)
        R1A2=RR*R1A1-YR*Y1A1
        Y1A2=YR*R1A1+RR*Y1A1
        IF(NAMPL1.GT.4) THEN
          R1B2=RR*R1B1-YR*Y1B1
          Y1B2=YR*R1B1+RR*Y1B1
        END IF
        IF(ICBNEW.GE.0) THEN
C         OUTGOING P WAVE:
          Y(28)=R1A2
          Y(29)=Y1A2
          IF(NAMPL2.GT.2) THEN
            Y(30)=R1B2
            Y(31)=Y1B2
          END IF
        ELSE
C         OUTGOING S WAVE:
          Y(28)= C*R1A2-S*R2A2
          Y(29)= C*Y1A2-S*Y2A2
          Y(30)= S*R1A2+C*R2A2
          Y(31)= S*Y1A2+C*Y2A2
          IF(NAMPL1.GT.4) THEN
            Y(32)= C*R1B2-S*R2B2
            Y(33)= C*Y1B2-S*Y2B2
            Y(34)= S*R1B2+C*R2B2
            Y(35)= S*Y1B2+C*Y2B2
          END IF
        END IF
C
        IEND=0
        RETURN
        END
C
C======================================================================
C
```

```
      SUBROUTINE CONV(KSTORE,YL,Y,IY,NAMPL,YC)
      INTEGER KSTORE,IY(12),NAMPL
      REAL YL(6),Y(35),YC(12)
C
C THIS SUBROUTINE REPLACES THE AMPLITUDES Y(28) TO Y(IY(1)) EXPRESSED IN
C THE RAY-CENTRED COORDINATE SYSTEM BY THE AMPLITUDES INVOLVING
C APPROPRIATE CONVERSION COEFFICIENTS, SEE C.R.T.5.5.4.
C
C INPUT:
C     KSTORE..SPECIFIES WHETHER THE CONVERSION COEFFICIENTS ARE TO BE
C             CONSIDERED (SEE C.R.T.5.5.4 AND 5.6E):
C             KSTORE.LE.1... NO AMPLITUDE CONVERSION COEFFICIENTS ARE
C               APPLIED AT THE POINT OF INTERSECTION WITH AN INTERFACE.
C             KSTORE.GE.2... AMPLITUDE CONVERSION COEFFICIENTS ARE
C               APPLIED AT THE POINT OF INTERSECTION WITH AN INTERFACE.
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF
C             INCIDENCE.
C     Y...    ARRAY OF THE DIMENSION AT LEAST 39.  THE LOCATIONS Y(1) TO
C             Y(IY(1)) CONTAIN BASIC QUANTITIES COMPUTED ALONG THE RAY,
C             CORRESPONDING TO THE INCIDENT WAVE.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY, CORRESPONDING TO THE INCIDENT WAVE.
C     YC...   ARRAY OF THE DIMENSION AT LEAST 12.
C
C OUTPUT:
C     NAMPL...NUMBER OF REAL QUANTITIES REPRESENTING COMPLEX-VALUED
C             VECTORIAL REDUCED AMPLITUDES.  IF NO CONVERSION
C             COEFFICIENTS ARE APPLIED NAMPL=IY(1)-27,  OTHERWISE
C             NAMPL=6 OR 12 (SEE C.R.T.5.5.4).
C     YC...   ARRAY CONTAINING REAL QUANTITIES REPRESENTING COMPLEX-
C             -VALUED VECTORIAL REDUCED AMPLITUDES.  IF NO CONVERSION
C             COEFFICIENTS ARE APPLIED, YC IS A COPY OF Y(28) TO
C             Y(IY(1)).  OTHERWISE,  YC REPRESENTS THE VECTORIAL REDUCED
C             AMPLITUDES INVOLVING APPROPRIATE CONVERSION COEFFICIENTS,
C             EXPRESSED IN RAY-CENTRED COORDINATE SYSTEM (SEE
C             C.R.T.5.5.4):
C             P WAVE AT THE INITIAL POINT OF THE RAY:
C               NAMPL=6,
C               YC(1)=REAL(A13),  YC(2)=AIMAG(A13),
C               YC(3)=REAL(A23),  YC(4)=AIMAG(A23),
C               YC(5)=REAL(A33),  YC(6)=AIMAG(A33).
C             S WAVE AT THE INITIAL POINT OF THE RAY:
C               NAMPL=12,
C               YC(1)=REAL(A11),  YC(2)=AIMAG(A11),
C               YC(3)=REAL(A21),  YC(4)=AIMAG(A21),
C               YC(5)=REAL(A31),  YC(6)=AIMAG(A31),
C               YC(7)=REAL(A12),  YC(8)=AIMAG(A12),
C               YC(9)=REAL(A22),  YC(10)=AIMAG(A22),
C               YC(11)=REAL(A32), YC(12)=AIMAG(A32).
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL KOOR,METRIC,SRFC2,PARM2,VELOC,SMVPRD,COEF50
      INTEGER KOOR
C     FUNCTION KOOR()... SUBROUTINE PACKAGE 'METRIC'.
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)... SUBROUTINE PACKAGE
C           'METRIC'.
C     SUBROUTINE SRFC2(ISRF,COOR,F) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'SRFC'.
C     SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C             ... SUBROUTINE PACKAGE 'PARM'.
```

```
C       SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)
C       SUBROUTINE SMVPRD(G,A1,A2,A3,B1,B2,B3)... SUBROUTINE PACKAGE
C               'RAYCB'.
C       SUBROUTINE COEF50(P,VP1,VS1,RO1,VP2,VS2,RO2,NCODE,ND,RMOD,RPH)...
C               SUBROUTINE PACKAGE 'COEF'.
C
C DATE: 1990, FEBRUARY 9
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C       AUXILIARY STORAGE LOCATIONS FOR LOCAL MODEL PARAMETERS:
        REAL G(12),GAMMA(18),GSQRD,FAUX(10)
        REAL UP(10),US(10),RO,QP,QS,VP,VS,VD(10),QL
C       THESE AUXILIARY VARIABLES AND ARRAYS NEED NOT BE LOCATED IN A
C       COMMON BLOCK.  THERE IS NO REASON TO LOCATE THEM IN THE AUXILIARY
C       COMMON BLOCK /AUXMOD/ BUT TO SHARE THE MEMORY.
        COMMON/AUXMOD/ G,GAMMA,GSQRD,FAUX,UP,US,RO,QP,QS,VP,VS,VD,QL
C
C       G,GAMMA,GSQRD... SEE SUBROUTINE METRIC OF THE PACKAGE 'METRIC'.
C       FAUX... AUXILIARY ARRAY TO STORE A FUNCTIONAL VALUE AND ITS
C               DERIVATIVES.
C       UP,US,RO,QP,QS... SEE SUBROUTINE PARM2 OF THE PACKAGE 'PARM'.
C       UP,US,QP,QS,VP,VS,VD,QL... SEE SUBROUTINE VELOC OF THE PACKAGE
C               'MODEL'.
C
C.......................................................................
C
C       AUXILIARY STORAGE LOCATIONS:
C
        INTEGER ICB2,NCODE,ND,NW,IW,I
        REAL VP1,VS1,RO1,V1,VP2,VS2,RO2,V2
        REAL H11,H13,H43,HHH,Z13,Z42,Z43,ZZZ
        REAL H21,H23,H53,    Z23,Z52,Z53
        REAL H31,H33,H63,    Z33,Z62,Z63
        REAL P,C1,S1,C2,S2,C,S,CR,CI,SR,SI
        REAL RMOD,RPH,RR,YR
        REAL R1A1,Y1A1,R2A1,Y2A1,R1B1,Y1B1,R2B1,Y2B1
        REAL R1A2,Y1A2,R2A2,Y2A2,R1B2,Y1B2,R2B2,Y2B2,RA2,YA2,RB2,YB2

C       ICB2... INDEX OF A COMPLEX BLOCK AT THE OTHER SIDE OF THE
C               INTERFACE.  AUXILIARY VARIABLE.
C       NCODE,ND... TYPE OF THE R/T COEFFICIENT.
C       NW...   TOTAL NUMBER OF R/T WAVES.
C       IW...   LOOP VARIABLE. INDEX OF AN R/T WAVE.
C       I...    AUXILIARY LOOP VARIABLE.
C       VP1,VS1,RO1... PARAMETERS OF THE COMPLEX BLOCK IABS(IY(5)).
C       V1...   INCIDENT WAVE VELOCITY.
C       VP2,VS2,RO2... PARAMETERS OF THE COMPLEX BLOCK ICB2.
C       V2...   R/T WAVE VELOCITY.
C       H11,H21,H31,H13,H23,H33... COVARIANT COMPONENTS OF THE BASIS
C               VECTORS OF RAY-CENTRED COORDINATE SYSTEM OF THE INCIDENT
C               WAVE.
C       H43,H53,H63... CONTRAVARIANT COMPONENTS OF THE SLOWNESS VECTOR OF
C               THE INCIDENT WAVE.  NOT USED IN CARTESIAN COORDINATES.
C       HHH...  NORM OF THE SLOWNESS VECTOR.
C       Z13,Z23,Z33... COVARIANT COMPONENTS OF THE LOCAL CARTESIAN
C               COORDINATE SYSTEM.
C       Z42,Z52,Z62,Z43,Z53,Z63... CONTRAVARIANT COMPONENTS OF THE LOCAL
C               CARTESIAN COORDINATE SYSTEM.
```

```
C       ZZZ...   NORM OF THE GRADIENT OF THE FUNCTION DESCRIBING THE
C                INTERFACE.
C       C1,S1...COSINE AND SINE OF THE ANGLE OF INCIDENCE.
C       C2,S2...COSINE AND SINE OF THE R/T ANGLE.  AUXILIARY VARIABLES.
C       C,S...   COSINE AND SINE OF THE REVOLUTION ANGLE.
C       P...     SNELL RAY PARAMETER.
C       CR,CI,SR,SI...REAL AND IMAGINARY PARTS OF THE COSINE AND SINE OF
C                THE ROTATION ANGLE IN THE PLANE OF INCIDENCE.
C       RMOD,RPH... ANY R/T COEFFICIENT.
C       RR,YR...REAL AND IMAGINARY PART OF A COMPLEX-VALUED R/T
C                COEFFICIENT.
C       ( R1A1,Y1A1,R2A1,Y2A1 )... AMPLIDUDE COEFFICIENTS IN RAY-CENTRED
C       ( R1B1,Y1B1,R2B1,Y2B1 )    COORDINATE SYSTEM OF THE INCIDENT WAVE
C       ( R1A2,Y1A2,R2A2,Y2A2 )
C       ( R1B2,Y1B2,R2B2,Y2B2 )                 INITIAL
C                PART:          COMPONENT:      POLARISATION:  WAVE:
C                R...REAL       1...P-SV, SV    A...FIRST      1...INCIDENT
C                Y...IMAGINARY  2...SH I  I     B...SECOND     2...R/T
C                                      I  I...R/T WAVE
C                                      I...INCIDENT WAVE
C       ( RA2,YA2 )... P-SV AMPLIDUDE COEFFICIENTS IN RAY-CENTRED
C       ( RB2,YB2 )    COORDINATE SYSTEM OF THE R/T WAVE
C                PART:          POLARISATION:  WAVE:
C                R...REAL       A...FIRST      2...R/T
C                Y...IMAGINARY  B...SECOND
C
C.......................................................................
C
C       CHECK WHETHER THE AMPLITUDE CONVERSION COEFFICIENTS ARE APPLIED
        IF(KSTORE.LE.1) THEN
          NAMPL=IY(1)-27
          DO 10 I=1,NAMPL
            YC(I)=Y(27+I)
   10     CONTINUE
          RETURN
        END IF
C
C       MATERIAL PARAMETERS CORRESPONDING TO THE INCIDENT WAVE:
        VP1=YL(1)
        VS1=YL(2)
        RO1=YL(3)
        IF(IY(5).GE.0) THEN
          V1=VP1
        ELSE
          V1=VS1
        END IF
C       MATERIAL PARAMETERS ON THE OTHER SIDE OF THE INTERFACE:
        ICB2=IABS(IY(8))
        IF(ICB2.NE.0) THEN
          CALL PARM2(ICB2,Y(3),UP,US,RO2,QP2,QS2)
          CALL VELOC(ISIGN(ICB2,ICBNEW),UP,US,QP2,QS2,VP2,VS2,VD,QL)
          NW=4
        ELSE
C         FREE SPACE:
          RO2=0.
          NW=2
        END IF
C       VP1, VS1, RO1, V1, VP2, VS2, RO2 AND NW ARE DEFINED.
C
C       BASIS OF THE RAY-CENTRED COORDINATE SYSTEM AND THE LOCAL CARTESIAN
```

```
C     COORDINATE SYSTEM CONNECTED WITH THE INTERFACE:
      CALL SRFC2(IABS(IY(6)),Y(3),FAUX)
C     NON-UNIT VECTORS - COVARIANT COMPONENTS
      H13=Y(6)
      H23=Y(7)
      H33=Y(8)
      H11=Y(9)
      H21=Y(10)
      H31=Y(11)
      Z13=FAUX(2)
      Z23=FAUX(3)
      Z33=FAUX(4)
C     (C) NON-UNIT VECTORS - CONTRAVARIANT COMPONENTS
      IF(KOOR().EQ.0) THEN
C       CARTESIAN COORDINATES:
        GSQRD=1.
C       NORM OF THE SLOWNESS VECTOR
        HHH=SQRT(H13*H13+H23*H23+H33*H33)
C       NORM OF THE GRADIENT OF THE FUNCTION DESCRIBING THE INTERFACE
        ZZZ=SQRT(Z13*Z13+Z23*Z23+Z33*Z33)
C       UNIT NORMAL TO THE INTERFACE:
C       COVARIANT COMPONENTS
        Z13=Z13/ZZZ
        Z23=Z23/ZZZ
        Z33=Z33/ZZZ
C       CONTRAVARIANT COMPONENTS
        Z43=Z13
        Z53=Z23
        Z63=Z33
      ELSE
C       CURVILINEAR COORDINATES:
        CALL METRIC(Y(3),GSQRD,G,GAMMA)
C       SLOWNESS VECTOR - CONTRAVARIANT COMPONENTS
        CALL SMVPRD(G(7),H13,H23,H33,H43,H53,H63)
C       NORM OF THE SLOWNESS VECTOR
        HHH=SQRT(H13*H43+H23*H53+H33*H63)
C       CONTRAVARIANT GRADIENT OF THE FUNCTION DESCRIBING THE INTERFACE
        CALL SMVPRD(G(7),Z13,Z23,Z33,Z43,Z53,Z63)
C       NORM OF THE GRADIENT OF THE FUNCTION DESCRIBING THE INTERFACE
        ZZZ=SQRT(Z13*Z43+Z23*Z53+Z33*Z63)
C       UNIT NORMAL TO THE INTERFACE:
C       COVARIANT COMPONENTS
        Z13=Z13/ZZZ
        Z23=Z23/ZZZ
        Z33=Z33/ZZZ
C       CONTRAVARIANT COMPONENTS
        Z43=Z43/ZZZ
        Z53=Z53/ZZZ
        Z63=Z63/ZZZ
      END IF
C     UNIT VECTOR TANGENT TO THE INCIDENT RAY
      H13=H13/HHH
      H23=H23/HHH
      H33=H33/HHH
C     COSINE OF THE ANGLE OF INCIDENCE
      C1=H13*Z43+H23*Z53+H33*Z63
C     SINE OF THE ANGLE OF INCIDENCE
      S1=SQRT(1.-C1*C1)
C     C1 AND S1 ARE THE COSINE AND SINE OF THE ANGLE OF INCIDENCE.
C
```

```
C       REVOLUTION ANGLE OF THE BASIS OF THE RAY-CENTRED COORDINATE SYSTEM
C       AROUND RAY:
        IF(S1.LT.0.0001) THEN
C         NEARLY NORMAL INCIDENCE
          C=1.
          S=0.
        ELSE
C         VECTOR PERPENDICULAR TO THE PLANE OF INCIDENCE - CONTRAVARIANT
          Z42=(Z23*H33-Z33*H23)
          Z52=(Z33*H13-Z13*H33)
          Z62=(Z13*H23-Z23*H13)
C         REVOLUTION ANGLE OF THE RAY CENTRED COORDINATE SYSTEM
          C=-(H11*Z43+H23*Z53+H31*Z63)/S1
          S=-(H11*Z42+H21*Z52+H31*Z62)/S1/GSQRD
        END IF
C       C AND S ARE THE COSINE AND SINE OF THE REVOLUTION ANGLE.
C
C       AMPLITUDES OF THE INCIDENT WAVE IN THE ROTATED RAY-CENTRED
C       COORDINATE SYSTEM OF THE INCIDENT WAVE:
        NAMPL=6
        IF(IY(5).GE.0) THEN
C         INCIDENT P WAVE:
          NCODE=0
          R1A1=Y(28)
          Y1A1=Y(29)
          R1A2=0.
          Y1A2=0.
          R2A2=0.
          Y2A2=0.
          YC(5)=R1A1
          YC(6)=Y1A1
          IF(IY(1).GT.29) THEN
            NAMPL=12
            R1B1= Y(30)
            Y1B1= Y(31)
            R1B2=0.
            Y1B2=0.
            R2B2=0.
            Y2B2=0.
            YC(11)=R1B1
            YC(12)=Y1B1
          END IF
        ELSE
C         INCIDENT S WAVE:
          NCODE=NW
          R1A1= C*Y(28)+S*Y(30)
          Y1A1= C*Y(29)+S*Y(31)
          R2A1=-S*Y(28)+C*Y(30)
          Y2A1=-S*Y(29)+C*Y(31)
          R1A2=R1A1
          Y1A2=Y1A1
          R2A2=R2A1
          Y2A2=Y2A1
          YC(5)=0.
          YC(6)=0.
          IF(IY(1).GT.31) THEN
            NAMPL=12
            R1B1= C*Y(32)+S*Y(34)
            Y1B1= C*Y(33)+S*Y(35)
            R2B1=-S*Y(32)+C*Y(34)
```

```
            Y2B1=-S*Y(33)+C*Y(35)
            R1B2=R1B1
            Y1B2=Y1B1
            R2B2=R2B1
            Y2B2=Y2B1
            YC(11)=0.
            YC(12)=0.
          END IF
        END IF
C
C       SNELL PARAMETER
        P=S1/V1
C
        IF(C1.GE.0) THEN
C         EPSILON=+1. (SEE C.R.T.5.9.7)
          ND=0
        ELSE
C         EPSILON=-1. (SEE C.R.T.5.9.7)
          ND=1
        END IF
C
C       LOOP OVER THE GENERATED WAVES:
        DO 90 IW=1,NW
          NCODE=NCODE+1
C
C         PROPAGATION VELOCITY OF THE GENERATED WAVE
          IF(IW.EQ.1) THEN
            V2=VP1
          ELSE IF(IW.EQ.2) THEN
            V2=VS1
          ELSE IF(IW.EQ.3) THEN
            V2=VP2
          ELSE
            V2=VS2
          END IF
C
C         ROTATION ANGLE OF THE BASIS OF THE RAY-CENTRED COORDINATE SYSTEM
C         IN THE PLANE OF INCIDENCE:
          S2=P*V2
          C2=1.-S2*S2
          CR=S1*S2
          SR=C1*S2
          IF(C2.GE.0.) THEN
            C2=SIGN(SQRT(C2),C1)
            IF(IW.LE.2) C2=-C2
            CR=CR+C1*C2
            SR=SR-S1*C2
            CI=0.
            SI=0.
          ELSE
            C2=SIGN(SQRT(-C2),C1)
            IF(IW.LE.2) C2=-C2
            CI= C1*C2
            SI=-S1*C2
          END IF
C         COSINE OF THE ROTATION ANGLE IS CR+I*CI, WHERE I=SQRT(-1),
C         SINE OF THE ROTATION ANGLE IS   SR+I*SI.
C
C         TRANSFORMATION OF REDUCED AMPLITUDES (5.9.6)
          CALL COEF50(P,VP1,VS1,RO1,VP2,VS2,RO2,NCODE,ND,RMOD,RPH)
```

```
        RR=RMOD*COS(RPH)
        YR=RMOD*SIN(RPH)
        RA2=RR*R1A1-YR*Y1A1
        YA2=YR*R1A1+RR*Y1A1
        IF(NAMPL.GT.6) THEN
          RB2=RR*R1B1-YR*Y1B1
          YB2=YR*R1B1+RR*Y1B1
        END IF
        IF(IW.EQ.1.OR.IW.EQ.3) THEN
C         OUTGOING P WAVE:
          R1A2=R1A2+SR*RA2-SI*YA2
          Y1A2=Y1A2+SI*RA2+SR*YA2
          YC(5)=YC(5)+CR*RA2-CI*YA2
          YC(6)=YC(6)+CI*RA2+CR*YA2
          IF(NAMPL.GT.6) THEN
            R1B2=R1B2+SR*RA2-SI*YA2
            Y1B2=Y1B2+SI*RA2+SR*YA2
            YC(11)=YC(11)+CR*RA2-CI*YA2
            YC(12)=YC(12)+CI*RA2+CR*YA2
          END IF
        ELSE
C         OUTGOING S WAVE:
          R1A2=R1A2+CR*RA2-CI*YA2
          Y1A2=Y1A2+CI*RA2+CR*YA2
          YC(5)=YC(5)-SR*RA2+SI*YA2
          YC(6)=YC(6)-SI*RA2-SR*YA2
          IF(NAMPL.GT.6) THEN
            R1B2=R1B2+CR*RA2-CI*YA2
            Y1B2=Y1B2+CI*RA2+CR*YA2
            YC(11)=YC(11)-SR*RA2+SI*YA2
            YC(12)=YC(12)-SI*RA2-SR*YA2
          END IF
          IF(IY(5).LT.0) THEN
C           SH WAVE COEFFICIENTS:
            IF(IW.EQ.2) THEN
              CALL COEFSH(P,VS1,RO1,VS2,RO2,1,RMOD,RPH)
            ELSE
              CALL COEFSH(P,VS1,RO1,VS2,RO2,2,RMOD,RPH)
            END IF
            RR=RMOD*COS(RPH)
            YR=RMOD*SIN(RPH)
            R2A2=R2A2+RR*R2A1-YR*Y2A1
            Y2A2=Y2A2+YR*R2A1+RR*Y2A1
            IF(NAMPL.GT.6) THEN
              R2B2=R2B2+RR*R2B1-YR*Y2B1
              Y2B2=Y2B2+YR*R2B1+RR*Y2B1
            END IF
          END IF
        END IF
   90 CONTINUE
C
C     ROTATION OF THE BASIS OF THE RAY-CENTRED COORDINATE SYSTEM AROUND
C     RAY:
      YC(1)= C*R1A2-S*R2A2
      YC(2)= C*Y1A2-S*Y2A2
      YC(3)= S*R1A2+C*R2A2
      YC(4)= S*Y1A2+C*Y2A2
      IF(NAMPL.GT.6) THEN
        YC(7)= C*R1B2-S*R2B2
        YC(8)= C*Y1B2-S*Y2B2
```

```
          YC(9)= S*R1B2+C*R2B2
          YC(10)=S*Y1B2+C*Y2B2
        END IF
C
        RETURN
        END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'COEF' - REFLECTION/TRANSMISSION COEFFICIENTS
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF:
C     COEFSH..SUBROUTINE DESIGNED TO EVALUATE SH R/T COEFFICIENTS
C             (SEE C.R.T.5.9.7).
C     COEF50..SUBROUTINE DESIGNED TO EVALUATE P-SV R/T COEFFICIENTS
C             (SEE C.R.T.5.9.7).
C
C DATE: 1989, DECEMBER 8
C
C=======================================================================
C
      SUBROUTINE COEFSH(P,VS1,RO1,VS2,RO2,NCODE,RMOD,RPH)
C
C     *****************************************************************
C
C     THE ROUTINE COEFSH IS DESIGNED FOR THE COMPUTATION OF REFLECTION
C     AND TRANSMISSION COEFFICIENTS AT A PLANE INTERFACE BETWEEN TWO
C     HOMOGENEOUS SOLID HALFSPACES OR AT A FREE SURFACE OF A HOMOGENEOUS
C     SOLID HALFSPACE.
C
C     I N P U T   P A R A M E T E R S
C          P...RAY PARAMETER
C          VS1,RO1...PARAMETERS OF THE FIRST HALFSPACE
C          VS2,RO2...PARAMETERS OF SECOND HALFSPACE. FOR THE FREE
C                    SURFACE TAKE RO2=0. AND ARBITRARY
C                    VALUE OF VS2
C          NCODE...CODE OF THE COMPUTED COEFFICIENT
C                    S1S1...NCODE=1
C                    S1S2...NCODE=2
C
C     O U T P U T   P A R A M E T E R S
C          RMOD,RPH...MODUL AND ARGUMENT OF THE COEFFICIENT
C
C     N O T E S
C     1/ TIME FACTOR OF INCIDENT WAVE...EXP(-I*OMEGA*T)
C     2/ FORMULAE ARE TAKEN FROM CERVENY,MOLOTKOV AND PSENCIK,
C        RAY METHOD IN SEISMOLOGY, PAGE 34
C
C     *****************************************************************
C
      COMPLEX A,B
      X= 1.-P*P*VS1*VS1
      Y= 1.-P*P*VS2*VS2
      C= RO1*VS1*SQRT(ABS(X))
      D= RO2*VS2*SQRT(ABS(Y))
      A= CMPLX(C,0.)
      IF(X.LT.0.) A= CMPLX(0.,C)
      B= CMPLX(D,0.)
      IF(Y.LT.0.) B= CMPLX(0.,D)
      GO TO (1,2), NCODE
    1 A= (A-B)/(A+B)
      GO TO 3
    2 A= (A+A)/(A+B)
    3 RMOD= SQRT(REAL(A)*REAL(A)+AIMAG(A)*AIMAG(A))
      RPH= ATAN2(AIMAG(A),REAL(A))
      RETURN
      END
```

```
C
C=====================================================================
C
      SUBROUTINE COEF50(P,VP1,VS1,RO1,VP2,VS2,RO2,NCODE,ND,RMOD,RPH)
C
C     ******************************************************************
C
C     THE ROUTINE COEF50 IS DESIGNED TO EVALUATE THE REFLECTION AND
C     TRANSMISSION COEFFICIENTS AT A PLANE INTERFACE BETWEEN TWO
C     HOMOGENEOUS SOLID HALFSPACES OR AT A FREE SURFACE OF A HOMOGENEOUS
C     SOLID HALFSPACE.
C
C     THE KINDS OF INDIVIDUAL COEFFICIENTS ARE SPECIFIED BY THE
C     FOLLOWING NUMBERS
C     A/ INTERFACE BETWEEN TWO SOLID HALFSPACES
C     P1P1...1      P1S1...2      P1P2...3      P1S2...4
C     S1P1...5      S1S1...6      S1P2...7      S1S2...8
C     B/ FREE SURFACE (FOR RO2.LT.0.00001)
C     PP.....1      PX.....5      PX,PZ...X- AND Z- COMPONENTS OF THE
C     PS.....2      PZ.....6      COEF.OF CONVERSION,INCIDENT P WAVE
C     SP.....3      SX.....7      SX,SZ...X- AND Z- COMPONENTS OF THE
C     SS.....4      SZ.....8      COEF.OF CONVERSION,INCIDENT S WAVE
C
C     I N P U T   P A R A M E T E R S
C           P...RAY PARAMETER
C           VP1,VS1,RO1...PARAMETERS OF THE FIRST HALFSPACE
C           VP2,VS2,RO2...PARAMETERS OF SECOND HALFSPACE. FOR THE FREE
C                    SURFACE TAKE RO2.LT.0.000001,EG.RO2=0., AND
C                    ARBITRARY VALUES OF VP2 AND VS2
C           NCODE...CODE OF THE COMPUTED COEFFICIENT
C           ND...=0  WHEN THE INTERFACE IS SITUATED AT THE RIGHT-HAND
C                    SIDE OF THE INCIDENT RAY (X AGAINST P)
C               =1   WHEN THE INTERFACE IS SITUATED AT THE LEFT-HAND
C                    SIDE OF THE INCIDENT RAY (X ALONG P)
C
C     O U T P U T   P A R A M E T E R S
C           RMOD,RPH...MODUL AND ARGUMENT OF THE COEFFICIENT
C
C     N O T E S
C     1/ POSITIVE P...IN THE DIRECTION OF PROPAGATION
C     2/ POSITIVE S...TO THE LEFT FROM P
C     3/ POSITIVE X...TO THE RIGHT FROM Z (TO THE RIGHT FROM P)
C     4/ POSITIVE Z...IN THE DIRECTION OF P
C     5/ TIME FACTOR OF INCIDENT WAVE...EXP(-I*OMEGA*T)
C     6/ FORMULAE ARE TAKEN FROM CERVENY AND RAVINDRA, THEORY OF SEISMIC
C        HEAD WAVES,PAGES 63-67. DUE TO THE NOTE 2, THE SIGNS AT CERTAIN
C        COEFFICIENTS ARE OPPOSITE (AND TIME FACTOR IS CHANGED, L.K.)
C        (SIGNS OF CONVERSION COEFFICIENTS ARE OPPOSITE TO CERVENY,
C        MOLOTKOV AND PSENCIK, RAY METHOD IN SEISMOLOGY, PAGE 35)
C
C     ******************************************************************
C
      COMPLEX B(4),RR,C1,C2,C3,C4,H1,H2,H3,H4,H5,H6,H,HB,HC
      DIMENSION PRMT(4),D(4),DD(4)
      PRMT(1)=VP1
      PRMT(2)=VS1
      PRMT(3)=VP2
      PRMT(4)=VS2
      IF(RO2.LT.0.000001)GO TO 150
      A1=VP1*VS1
```

```
      A2=VP2*VS2
      A3=VP1*RO1
      A4=VP2*RO2
      A5=VS1*RO1
      A6=VS2*RO2
      Q=2.*(A6*VS2-A5*VS1)
      PP=P*P
      QP=Q*PP
      X=RO2-QP
      Y=RO1+QP
      Z=RO2-RO1-QP
      G1=A1*A2*PP*Z*Z
      G2=A2*X*X
      G3=A1*Y*Y
      G4=A4*A5
      G5=A3*A6
      G6=Q*Q*PP
      DO 21 I=1,4
      DD(I)=P*PRMT(I)
   21 D(I)=SQRT(ABS(1.-DD(I)*DD(I)))
      IF(DD(1).LE.1..AND.DD(2).LE.1..AND.DD(3).LE.1..AND.DD(4).LE.1.)
     1GO TO 100
C
C     COMPLEX COEFFICIENTS
      DO 22 I=1,4
      IF(DD(I).GT.1.)GO TO 23
      B(I)=CMPLX(D(I),0.)
      GO TO 22
   23 B(I)= CMPLX(0.,D(I))
   22 CONTINUE
      C1=B(1)*B(2)
      C2=B(3)*B(4)
      C3=B(1)*B(4)
      C4=B(2)*B(3)
      H1=CMPLX(G1,0.)
      H2=G2*C1
      H3=G3*C2
      H4=G4*C3
      H5=G5*C4
      H6=G6*C1*C2
      H=1./(H1+H2+H3+H4+H5+H6)
      HB=2.*H
      HC=HB*P
      GO TO (1,2,3,4,5,6,7,8),NCODE
    1 RR=H*(H2+H4+H6-H1-H3-H5)
      GO TO 26
    2 RR=VP1*B(1)*HC*(Q*Y*C2+A2*X*Z)
      IF(ND.NE.0)RR=-RR
      GO TO 26
    3 RR=A3*B(1)*HB*(VS2*B(2)*X+VS1*B(4)*Y)
      GO TO 26
    4 RR=-A3*B(1)*HC*(Q*C4-VS1*VP2*Z)
      IF(ND.NE.0)RR=-RR
      GO TO 26
    5 RR=-VS1*B(2)*HC*(Q*Y*C2+A2*X*Z)
      IF(ND.NE.0)RR=-RR
      GO TO 26
    6 RR=H*(H2+H5+H6-H1-H3-H4)
      GO TO 26
    7 RR=A5*B(2)*HC*(Q*C3-VP1*VS2*Z)
```

```
      IF(ND.NE.0)RR=-RR
      GO TO 26
    8 RR=A5*B(2)*HB*(VP1*B(3)*Y+VP2*B(1)*X)
      GO TO 26
C     REAL COEFFICIENTS
  100 E1=D(1)*D(2)
      E2=D(3)*D(4)
      E3=D(1)*D(4)
      E4=D(2)*D(3)
      S1=G1
      S2=G2*E1
      S3=G3*E2
      S4=G4*E3
      S5=G5*E4
      S6=G6*E1*E2
      S=1./(S1+S2+S3+S4+S5+S6)
      SB=2.*S
      SC=SB*P
      GO TO (101,102,103,104,105,106,107,108),NCODE
  101 R=S*(S2+S4+S6-S1-S3-S5)
      GO TO 250
  102 R=VP1*D(1)*SC*(Q*Y*E2+A2*X*Z)
      IF(ND.NE.0)R=-R
      GO TO 250
  103 R=A3*D(1)*SB*(VS2*D(2)*X+VS1*D(4)*Y)
      GO TO 250
  104 R=-A3*D(1)*SC*(Q*E4-VS1*VP2*Z)
      IF(ND.NE.0)R=-R
      GO TO 250
  105 R=-VS1*D(2)*SC*(Q*Y*E2+A2*X*Z)
      IF(ND.NE.0)R=-R
      GO TO 250
  106 R=S*(S2+S5+S6-S1-S3-S4)
      GO TO 250
  107 R=A5*D(2)*SC*(Q*E3-VP1*VS2*Z)
      IF(ND.NE.0)R=-R
      GO TO 250
  108 R=A5*D(2)*SB*(VP1*D(3)*Y+VP2*D(1)*X)
      GO TO 250
C
C     EARTHS SURFACE,COMPLEX COEFFICIENTS AND COEFFICIENTS OF CONVERSION
  150 A1=VS1*P
      A2=A1*A1
      A3=2.*A2
      A4=2.*A1
      A5=A4+A4
      A6=1.-A3
      A7=2.*A6
      A8=2.*A3*VS1/VP1
      A9=A6*A6
      DD(1)=P*VP1
      DD(2)=P*VS1
      DO 151 I=1,2
  151 D(I)=SQRT(ABS(1.-DD(I)*DD(I)))
      IF(DD(1).LE.1..AND.DD(2).LE.1.)GO TO 200
      DO 154 I=1,2
      IF(DD(I).GT.1.)GO TO 155
      B(I)=CMPLX(D(I),0.)
      GO TO 154
  155 B(I)= CMPLX(0.,D(I))
```

```
  154 CONTINUE
      H1=B(1)*B(2)
      H2=H1*A8
      H=1./(A9+H2)
      GO TO (161,162,163,164,165,166,167,168),NCODE
  161 RR=(-A9+H2)*H
      GO TO 26
  162 RR=-A5*B(1)*H*A6
      IF(ND.NE.0)RR=-RR
      GO TO 26
  163 RR=A5*B(2)*H*A6*VS1/VP1
      IF(ND.NE.0)RR=-RR
      GO TO 26
  164 RR=-(A9-H2)*H
      GO TO 26
  165 RR=-A5*H1*H
      IF(ND.NE.0)RR=-RR
      GO TO 26
  166 RR=A7*B(1)*H
      GO TO 26
  167 RR=-A7*B(2)*H
      GO TO 26
  168 RR=-A5*VS1*H1*H/VP1
      IF(ND.NE.0)RR=-RR
   26 Z2=REAL(RR)
      Z3=AIMAG(RR)
      IF(Z2.EQ.0..AND.Z3.EQ.0.)GO TO 157
      RMOD=SQRT(Z2*Z2+Z3*Z3)
      RPH=ATAN2(Z3,Z2)
      RETURN
  157 RMOD=0.
      RPH=0.
      RETURN
C
C     EARTHS SURFACE,REAL COEFFICIENTS AND COEFFICIENTS OF CONVERSION
  200 S1=D(1)*D(2)
      S2=A8*S1
      S=1./(A9+S2)
      GO TO (201,202,203,204,205,206,207,208),NCODE
  201 R=(-A9+S2)*S
      GO TO 250
  202 R=-A5*D(1)*S*A6
      IF(ND.NE.0)R=-R
      GO TO 250
  203 R=A5*D(2)*S*A6*VS1/VP1
      IF(ND.NE.0)R=-R
      GO TO 250
  204 R=(S2-A9)*S
      GO TO 250
  205 R=-A5*S1*S
      IF(ND.NE.0)R=-R
      GO TO 250
  206 R=A7*D(1)*S
      GO TO 250
  207 R=-A7*D(2)*S
      GO TO 250
  208 R=-A5*VS1*S1*S/VP1
      IF(ND.NE.0)R=-R
  250 IF(R.LT.0.)GO TO 251
      RMOD=R
```

```
      RPH=0.
      RETURN
  251 RMOD=-R
      RPH=3.14159
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'INIT' TO READ THE INPUT DATA FOR THE INITIAL
C SURFACE, AND TO DEFINE INITIAL VALUES FOR COMPLETE RAY TRACING.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C      INITB...BLOCK DATA SUBROUTINE DEFINING COMMON BLOCK /INITC/ TO
C              STORE THE IMPORTANT QUANTITIES AT THE INITIAL POINT OF THE
C              RAY, AND COMMON BLOCK /INISC/ TO STORE SOME OF THE INPUT
C              DATA.
C      INIT1...INTERFACE SUBROUTINE TO INIS1 READING THE INPUT DATA FOR
C              THE FOUR FUNCTIONS SPECIFYING THE INITIAL CONDITIONS FOR
C              THE COMPUTED RAYS.
C      INIT2...SUBROUTINE EVALUATING, FOR GIVEN RAY TAKE-OFF PARAMETERS,
C              THE VALUES OF THE COMPUTED QUANTITIES AT THE INITIAL POINT
C              OF THE RAY, AND STORING THE IMPORTANT QUANTITIES AT THE
C              INITIAL POINT OF THE RAY IN THE COMMON BLOCK /INITC/.
C      INIS1...SAMPLE SUBROUTINE DESIGNED TO READ THE INPUT DATA FOR THE
C              INITIAL POINTS OF RAYS.  A TWO-PARAMETRIC SYSTEM OF RAYS
C              OF EACH ELEMENTARY WAVE IS ASSUMED.  A RAY OF THE
C              ELEMENTARY WAVE IS SPECIFIED BY ITS TWO TAKE-OFF
C              PARAMETERS.  THE COMPUTED RAYS MAY START FROM A SINGLE
C              INITIAL POINT COMMON TO ALL RAYS, FROM A CURVE ALONG WHICH
C              AN INITIAL TRAVEL TIME IS DEFINED, FROM AN INITIAL SURFACE
C              ALONG WHICH AN INITIAL TRAVEL TIME IS DEFINED, ETC.
C      INIS2...SAMPLE SUBROUTINE RETURNING THE FUNCTIONAL VALUES AND
C              THEIR FIRST AND SECOND DERIVATIVES, OF THE FUNCTIONS
C              DESCRIBING THE INITIAL SURFACE.
C      SQRT3...SUBROUTINE EVALUATING THE SQUARE ROOT OF THE GIVEN
C              REAL-VALUED POSITIVE-SEMIDEFINITE SYMMETRIC 3*3 MATRIX.
C      SPHERE..SUBROUTINE TRANSFORMING SPHERICAL COORDINATES PAR1, PAR2
C              INTO THE CARTESIAN COORDINATES OF THE CORRESPONDING POINT
C              ON THE UNIT SPHERE.  IT ALSO EVALUATES THE FIRST AND
C              SECOND DERIVATIVES OF THE CARTESIAN COORDINATES WITH
C              RESPECT TO PAR1 AND PAR2.
C SUBROUTINES INIS1 AND INIS2, DEFINING THE COMMON INITIAL POINT,
C INITIAL CURVE OR INITIAL SURFACE, CALL SUBROUTINES VAL1 AND VAL2 WHICH
C MUST BE APPENDED.  IN ADDITION, SUBROUTINES CURVN1 (OR ITS ALTERNATIVE
C CURVB1), CURV2D (OR ITS ALTERNATIVE CURVBD), SURFB1, SURFBD, VAL3B1,
C VAL3BD, VGEN, TERMS, SNHCSH, TRIDEC, TRISOL, DSPLNZ, INTRVL FROM THE
C SUBROUTINE PACKAGE 'FITPACK' BY ALAN KAYLOR CLINE, DEPARTMENT OF
C COMPUTER SCIENCES, UNIVERSITY OF TEXAS AT AUSTIN, ARE USED.  IN THE
C COMPLETE RAY TRACING, SUBROUTINES INIS1 AND INIS2 MAY BE REPLACED BY
C ANY USER-DEFINED PACKAGE CONTAINING SUBROUTINES INIS1 AND INIS2 WITH
C THE SAME NUMBER, TYPE AND MEANING OF THEIR PARAMETERS AS IN THIS
C PACKAGE.
C
C INPUT DATA FOR THE INITIAL POINTS OF RAYS:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C     THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C     I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C     INTEGER.  OTHERWISE (EXCEPT TEXTI), THE INPUT PARAMETER IS OF THE
C     TYPE REAL.
C (1) TEXTI
C     STRING DESCRIBING THE DATA.  ONLY THE FIRST 80 CHARACTERS OF THE
C     STRING ARE SIGNIFICANT.
C (2) INIDIM,INIPAR
C     QUANTITIES DEFINING THE KIND OF INITIAL CONDITIONS (THE KIND OF
```

```
C      THE SOURCE).
C      INIDIM..DETERMINES THE DIMENSIONALITY OF THE SOURCE:
C              0...SINGLE INITIAL POINT (POINT SOURCE),
C              1...INITIAL CURVE (LINE SOURCE),
C              2...INITIAL SURFACE.
C      INIPAR..DETERMINES THE PARAMETRIZATION OF RAYS:
C              FOR INIDIM=0:
C                INIPAR.LE.1: RAY PARAMETERS ARE POLAR-LIKE SPHERICAL
C                  COORDINATES (COLATITUDE,LONGITUDE) CONNECTED WITH THE
C                  LOCAL CARTESIAN COORDINATE SYSTEM WHICH BASIS VECTORS
C                  ARE GIVEN BY THE SQUARE ROOT OF THE METRIC TENSOR AT
C                  THE INITIAL POINT.
C                INIPAR.GE.2: RAY PARAMETERS ARE GEOGRAPHIC-LIKE
C                  SPHERICAL COORDINATES (LONGITUDE,LATITUDE) CONNECTED
C                  WITH THE LOCAL CARTESIAN COORDINATE SYSTEM WHICH BASIS
C                  VECTORS ARE GIVEN BY THE SQUARE ROOT OF THE METRIC
C                  TENSOR AT THE INITIAL POINT.
C              FOR INIDIM=1:
C                INIPAR MUST BE 1 OR 2.  THE INIPAR-TH RAY PARAMETER IS
C                IDENTICAL WITH THE PARAMETER PARAMETRIZING THE INITIAL
C                CURVE.  THE OTHER RAY PARAMETER IS THE ANGLE BETWEEN THE
C                RAY TAKE-OFF PLANE AND THE NORMAL TO THE INTERPOLATED
C                SURFACE.  THE RAY TAKE-OFF PLANE IS GIVEN BY THE TANGENT
C                TO THE INITIAL LINE AND BY THE SLOWNESS VECTOR.
C                FOR INIPAR=1, THE INITIAL LINE IS THE LINE PAR2=0 AT THE
C                  INTERPOLATED SURFACE AND IS PARAMETRIZED BY PAR1.
C                FOR INIPAR=2, THE INITIAL LINE IS THE LINE PAR1=0 AT THE
C                  INTERPOLATED SURFACE AND IS PARAMETRIZED BY PAR2.
C              FOR INIDIM=2:
C                RAY PARAMETERS ARE IDENTICAL WITH TWO PARAMETERS
C                PARAMETRIZING THE INITIAL SURFACE.
C                INIPAR.LE.0: INITIAL SURFACE IS DESCRIBED IN TERMS OF
C                  FUNCTIONS SPECIFYING THE DEPENDENCE OF GENERAL
C                  COORDINATES (X1,X2,X3) ON TWO PARAMETERS OF THE
C                  INITIAL SURFACE.
C                INIPAR.EQ.1: INITIAL SURFACE IS SPECIFIED IN THE
C                  POLAR-LIKE SPHERICAL COORDINATES (COLATITUDE,
C                  LONGITUDE, RADIUS) CONNECTED WITH THE LOCAL CARTESIAN
C                  COORDINATE SYSTEM WHICH BASIS VECTORS ARE GIVEN BY THE
C                  SQUARE ROOT OF THE METRIC TENSOR AT THE GIVEN POINT.
C                  COLATITUDE AND LONGITUDE ARE THE PARAMETERS, AND THE
C                  INITIAL SURFACE IS DETERMINED BY A FUNCTION SPECIFYING
C                  THE DEPENDENCE OF THE RADIUS ON THESE PARAMETERS
C                  (COLATITUDE AND LONGITUDE).
C                INIPAR.GE.2: INITIAL SURFACE IS SPECIFIED IN THE
C                  GEOGRAPHIC-LIKE SPHERICAL COORDINATES (LONGITUDE,
C                  LATITUDE, RADIUS) CONNECTED WITH THE LOCAL CARTESIAN
C                  COORDINATE SYSTEM WHICH BASIS VECTORS ARE GIVEN BY THE
C                  SQUARE ROOT OF THE METRIC TENSOR AT THE GIVEN POINT.
C                  LONGITUDE AND LATITUDE ARE THE PARAMETERS, AND THE
C                  INITIAL SURFACE IS DETERMINED BY A FUNCTION SPECIFYING
C                  THE DEPENDENCE OF THE RADIUS ON THESE PARAMETERS
C                  (LONGITUDE AND LATITUDE).
C (3) DATA DESCRIBING THE INITIAL POINT, CURVE OR SURFACE.
C     FOR INIDIM=0:
C             (3A) X1INI,X2INI,X3INI,TTINI
C               X1INI,X2INI,X3INI... COORDINATES OF A SINGLE INITIAL
C                 POINT.
C               TTINI... INITIAL VALUE OF THE TRAVEL TIME.
C     FOR INIDIM=1:
```

```
C               (3B) INPUT DATA FOR NFUNC=4 FUNCTIONS X1(.,.),X2(.,.),
C                 X3(.,.), TT(.,.) OF TWO VARIABLES.  THE INIPAR-TH ONE OF
C                 THE 2 VARIABLES BEING SIMULTANEOUSLY THE RAY PARAMETER).
C     FOR INIDIM=2, INIPAR.LE.0:
C               (3B) INPUT DATA FOR NFUNC=4 FUNCTIONS X1(.,.),X2(.,.),
C                 X3(.,.), TT(.,.) OF TWO VARIABLES (TWO INITIAL SURFACE
C                 PARAMETERS, BEING SIMULTANEOUSLY THE RAY PARAMETERS).
C     FOR INIDIM=2, INIPAR.GE.1: THE FOLLOWING TWO DATA SETS (3A), (3B):
C               (3A) X1INI,X2INI,X3INI... COORDINATES OF A GIVEN POINT,
C                 SEE THE DESCRIPTION OF THE INPUT DATA (1).
C               (3B) INPUT DATA FOR NFUNC=2 FUNCTIONS R(.,.),TT(.,.) OF
C                 TWO VARIABLES (TWO INITIAL SURFACE PARAMETERS, BEING
C                 SIMULTANEOUSLY THE RAY PARAMETERS).
C                 R(.,.) DESCRIBES THE RADIUS, SEE INPUT DATA (1),
C                 TT(.,.) IS THE INITIAL TRAVEL TIME.
C THE STRUCTURE OF THE INPUT DATA (3B) IS GIVEN BY THE SUBROUTINE VAL1
C AND IS DESCRIBED LATER ON.
C
C STORAGE IN THE MEMORY:
C     THE INPUT DATA (2) AND (3A) ARE STORED IN THE COMMON BLOCK
C     /INISC/.  THE IMPORTANT QUANTITIES AT THE INITIAL POINT OF THE RAY
C     (SEE C.R.T.6.1) ARE STORED IN THE COMMON BLOCK /INITC/.  THESE
C     COMMON BLOCKS ARE DEFINED IN THE FOLLOWING SUBROUTINE:
C     -----------------------------------------------------------------
      BLOCK DATA INITB
      INTEGER INIDIM,INIPAR
      REAL X1INI,X2INI,X3INI,TTINI
      COMMON/INISC/INIDIM,INIPAR,X1INI,X2INI,X3INI,TTINI
      SAVE /INISC/
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
      SAVE /INITC/
      END
C     -----------------------------------------------------------------
C     INIDIM,INIPAR... INPUT DATA (2).
C     X1INI,X2INI,X3INI,TTINI... INPUT DATA (3A) IF THEY ARE DEFINED.
C     ISB1I,ICB1I... INDICES OF A SIMPLE AND A COMPLEX BLOCKS IN WHICH
C             THE INITIAL POINT OF THE RAY IS SITUATED (SEE C.R.T.6.1).
C     YLI...  ARRAY CONTAINING THE VALUES OF THE QUANTITIES YL(1)-YL(6)
C             (SEE C.R.T.5.5.4) DESCRIBING THE LOCAL PROPERTIES OF THE
C             MODEL AT THE INITIAL POINT OF THE RAY, SEE C.R.T.6.1.
C             THEY MUST NOT BE CHANGED OUTSIDE THE SUBROUTINE INIT2.
C     YI...   ARRAY CONTAINING THE FOLLOWING QUANTITIES DESCRIBING THE
C             PROPERTIES OF THE RAYS AND OF THE TRAVEL-TIME FIELD, SEE
C             C.R.T.6.1:
C     YI(1)...INITIAL TRAVEL TIME.
C     YI(2)...INITIAL IMAGINARY PART OF THE COMPLEX TRAVEL TIME.
C     YI(3)-YI(5)... COORDINATES OF THE INITIAL POINT OF THE RAY.
C     YI(6)-YI(8)... COVARIANT COMPONENTS OF THE INITIAL SLOWNESS
C             VECTOR.
C     YI(9)-YI(11)... COVARIANT COMPONENTS OF THE FIRST BASIS VECTOR OF
C             THE RAY-CENTRED COORDINATE SYSTEM AT THE INITIAL POINT OF
C             THE RAY (PERPENDICULAR TO THE SLOWNESS VECTOR
C             YI(6)-YI(8)).
C     YI(12),YI(16)     QR11,QR12
C     YI(13),YI(17)     QR21,QR22
C     YI(14),YI(18)     PR11,PR12
```

```
C       YI(15),YI(19)...  PR21,PR22
C                 ELEMENTS OF THE RAY GEOMETRICAL SPREADING MATRIX QR, AND
C                 OF THE MATRIX PR (SEE C.R.T.,EQ.(5.13)) AT THE INITIAL
C                 POINT OF THE RAY.
C       YI(20),YI(21)... TAKE-OFF PARAMETERS OF THE RAY.
C       IN ADDITION TO THE ABOVE QUANTITIES DESCRIBING THE PROPERTIES
C       DEFINED FOR A SINGLE RAY, THERE ARE ALSO QUANTITIES DESCRIBING
C       THE PROPERTIES OF THE DISCRETE SYSTEM OF COMPUTED RAYS IN THE
C       VICINITY OF THE COMPUTED RAY.  THESE QUANTITIES Y(22)-Y(25) ARE
C       NOT DEFINED IN THE SUBROUTINE INIT2:
C       YI(22)..AREA OF THE ELEMENT OF THE RAY-PARAMETER SURFACE,
C                 CORRESPONDING TO THE RAY, SEE C.R.T.,EQ.(6.1).
C       YI(23),YI(24),YI(25)... COMPONENTS 11, 12, 22 OF THE SYMMETRIC
C                 MATRIX INVERSE TO THE SPECIFIC MOMENT OF THE ELEMENT OF
C                 THE RAY-PARAMETER SURFACE CORRESPONDING TO THE RAY, SEE
C                 C.R.T.,EQ.(6.2).
C       COMMON BLOCK /INISC/ IS INCLUDED IN SUBROUTINES INIS1 AND INIS2
C       IN ORDER TO COMMUNICATE THE INPUT DATA TO THE SUBROUTINE INIS2.
C       COMMON BLOCK /INITC/ IS INCLUDED IN EXTERNAL PROCEDURES INIT2,
C       OUTP (SEE RAYCB), AND MAY BE INCLUDED IN ANY OTHER SUBROUTINE.
C       THE INDEX OF THE LAST ALLOCATED NUMERIC UNIT OF ARRAY FSRFCA IS
C       NAMED MSRFCA.  IF MSRFCA IS CHANGED, IT MUST BE ADJUSTED IN ALL
C       SUBROUTINES WHICH INCLUDE COMMON BLOCK /INITC/.
C
C ABOVE MENTIONED INPUT DATA (2B) FOR THE INITIAL CURVE OR FOR THE
C INITIAL SURFACE ARE READ IN BY THE SUBROUTINE VAL1 AND HAVE THE
C FOLLOWING STRUCTURE:
C       THESE INPUT DATA DEFINE AT LEAST NFUNC INDIVIDUAL FUNCTIONS
C       DESCRIBING THE INITIAL CONDITIONS.  THEY ARE READ IN BY SUBROUTINE
C       VAL1 CALLED BY INIS1.  THE NUMBER MFUNC OF ALL FUNCTIONS SPECIFIED
C       IN THE INPUT DATA MAY BE GREATER OR EQUAL TO NFUNC.  THE DATA ARE
C       READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).
C (1) MFUNC
C       THE NUMBER OF ALL INPUT FUNCTIONS.  IT MUST BE GREATER OR EQUAL TO
C       THE NUMBER NFUNC OF THE FUNCTIONS REQUIRED TO DESCRIBE THE
C       COORDINATES AND TRAVEL TIME ALONG THE INITIAL CURVE OR SURFACE.
C       THE FUNCTIONS INDEXED 1 TO NFUNC MUST BE THE FUNCTIONS DESCRIBING
C       THE COORDINATES AND TRAVEL TIME ALONG THE INITIAL CURVE OR
C       SURFACE.
C (2) NFUNC-TIMES (I.E. ONCE FOR EACH FUNCTION) INPUT DATA (2A)+(2B):
C (2A) TEXTF,IFUNC
C       IDENTIFICATION OF THE FUNCTION.
C       TEXTF...ANY STRING. ITS FIRST 3 CHARACTERS MUST DIFFER FROM 'END'.
C       IFUNC...INDEX OF THE FUNCTION:
C                 1 TO 3... COORDINATES AND 4... TRAVEL TIME, OR
C                 1... RADIUS AND 2... TRAVEL TIME.  AMPLITUDE AND/OR OTHER
C                 QUANTITIES MAY FOLLOW.
C (2B) 'INPUT DATA FOR ONE FUNCTION', SEE BELOW.
C (3) TEXTE,AUX
C       END OF DATA.
C       TEXTE...STRING, THE FIRST 3 CHARACTERS OF WHICH MUST BE UPPER-CASE
C                 'END'.
C       AUX...  ANY NUMBER OR A SLASH.
C
C INPUT DATA FOR ONE FUNCTION:
C       THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C       THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C       THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C       THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C       I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
```

```
C       INTEGER.  OTHERWISE, THE INPUT PARAMETER IS OF THE TYPE REAL.
C (1)  IVAR1,IVAR2,0,SIGMA
C       THE FORM OF THE FUNCTION.
C       IVAR1,IVAR2... DENOTE THE FORM OF THE FUNCTION. THE FUNCTION MUST
C               BE OF THE FORM
C                 F(G1,G2) = W(A1,A2)-B1-B2   .
C               G1, G2 ARE THE RAY PARAMETERS. EACH OF A1, A2, B1, B2 MUST
C               BE EITHER:  (A) ONE OF RAY PARAMETERS G1, G2,  OR (B) MUST
C               BE LEFT OUT.  AT MOST 2 OF PARAMETERS A1-B2 MAY BE OF KIND
C               (A).  NOTE THAT IVAR1 CONTROLS THE TYPE OF A1 AND B1,
C               IVAR2 CONTROLS THE TYPE OF A2 AND B2.
C               FOR IVAR1.EQ.0: A1, B1 ARE EMPTY (LEFT OUT),
C               FOR IVAR1.EQ.1: A1=G1, B1 IS EMPTY,
C               FOR IVAR1.EQ.2: A1=G2, B1 IS EMPTY,
C               FOR IVAR1.EQ.-1: B1=G1, A1 IS EMPTY,
C               FOR IVAR1.EQ.-2: B1=G2, A1 IS EMPTY,
C               THE MEANING OF THE PARAMETER IVAR2 IS SIMILAR.
C               EXAMPLES:
C               IVAR1: IVAR2: THE FORM OF THE FUNCTION:
C                 1      2     F(G1,G2)=W(G1,G2)
C                 2      1     F(G1,G2)=W(G2,G1)
C                 1      0     F(G1,G2)=W(G1)
C                 1     -2     F(G1,G2)=W(G1)-G2
C               FUNCTION W IS INTERPOLATED BY MEANS OF SPLINES UNDER
C               TENSION.
C       SIGMA...IS THE TENSION FACTOR (ITS SIGN IS IGNORED).  THIS VALUE
C               INDICATES THE CURVINESS DESIRED.  IF ABS(SIGMA) IS NEARLY
C               ZERO (E.G. 0.001), THE RESULTING SURFACE IS APPROXIMATELY
C               THE TENSOR PRODUCT OF CUBIC SPLINES.  IF ABS(SIGMA) IS
C               LARGE (E.G. 50.), THE RESULTING SURFACE IS APPROXIMATELY
C               TRI-LINEAR.  IF SIGMA EQUALS ZERO, TENSOR PRODUCTS OF
C               CUBIC SPLINES RESULT.  A RECOMMENDED VALUE FOR SIGMA IS
C               APPROXIMATELY 1. IN ABSOLUTE VALUE.
C (2)  NX(1),...,NX(NVAR)
C       THE NUMBERS OF GRID COORDINATES FOR THE INTERPOLATION.
C       THIS INPUT IS PERFORMED IF AT LEAST ONE OF IVAR1, IVAR2 IS
C       POSITIVE.
C       EACH OF NX(1),...,NX(NVAR) CORRESPONDS TO ONE POSITIVE VALUE OF
C       IVAR1, IVAR2 AND SPECIFIES THE NUMBER OF GRID COORDINATES
C       CORRESPONDING TO THAT INDEPENDENT VARIABLE OF FUNCTION W, SEE (1).
C       THE SIGN OF NX(1),...,NX(NVAR) IS IGNORED. NVAR (.LE.2) IS THE
C       NUMBER OF POSITIVE VALUES OF THE ABOVE QUANTITIES IVAR1, IVAR2,
C       I.E. THE NUMBER OF INDEPENDENT VARIABLES OF FUNCTION W, SEE (1).
C (3)  X1(1),...,X1(NX(1))
C       THE GRID COORDINATES CORRESPONDING TO THE FIRST INDEPENDENT
C       VARIABLE OF FUNCTION W, SEE (1).
C       THIS INPUT IS PERFORMED IF NX(1) IS SPECIFIED, SEE (2), AND IS NOT
C       ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (4)  X2(1),...,X2(NX(2))
C       THE GRID COORDINATES CORRESPONDING TO THE SECOND INDEPENDENT
C       VARIABLE OF FUNCTION W, SEE (1).
C       THIS INPUT IS PERFORMED IF NX(2) IS SPECIFIED, SEE (2), AND IS NOT
C       ZERO. THE GRID COORDINATES MAY BE SPECIFIED IN ANY ORDER.
C (5)  (((W(I,J),I=1,MAX(NX(1),1)),J=1,MAX(NX(2),1))
C       THE VALUES OF FUNCTION W AT GRID POINTS. FUNCTION VALUE W(I,J)
C       CORRESPONDS TO POINT (X1(I),X2(J)).
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
```

```
C=======================================================================
C
      SUBROUTINE INIT1(LUN)
      INTEGER LUN
C
C SUBROUTINE INIT1 CALLS THE SUBROUTINE INIS1 TO READ THE INPUT DATA FOR
C THE INITIAL POINTS OF RAYS.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C THE INPUT PARAMETER IS NOT ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK /INITC/:
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK, EXCEPT ICB1I, ARE
C ALTERED.  ICB1I IS SET TO ZERO.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL INIS1
C     SUBROUTINE INIS1(LUN,NFUNC) AND SUBSEQUENT ROUTINES... THIS
C             SUBROUTINE PACKAGE.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER NFUNC
      PARAMETER (NFUNC=4)
C
      CALL INIS1(LUN,NFUNC)
      ICB1I=0
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE INIT2(PAR1,PAR2,YL,Y,YY,IY,IEND,IWAVE0,IKODE)
      REAL PAR1,PAR2,YL(6),Y(35),YY(5)
      INTEGER IY(12),IEND,IWAVE0,IKODE
C
C SUBROUTINE INIT2 EVALUATES, FOR GIVEN RAY TAKE-OFF PARAMETERS, THE
C VALUES OF THE COMPUTED QUANTITIES AT THE INITIAL POINT OF THE RAY, AND
C STORES THE IMPORTANT QUANTITIES AT THE INITIAL POINT OF THE RAY IN THE
C COMMON BLOCK /INITC/.
C
C INPUT:
C     PAR1,PAR2... RAY TAKE-OFF PARAMETERS.
C     YL,Y,YY,IY... ARRAYS OF DIMENSIONS AT LEAST 6, 35, 5, 12,
C             RESPECTIVELY.
C     IWAVE0..INDEX OF THE ALREADY COMPUTED ELEMENTARY WAVE HAVING THE
C             MOST NUMEROUS COMMON ELEMENTS WITH THE CURRENT ELEMENTARY
```

```
C               WAVE.  NEED NOT BE DEFINED IF IKODE=0.
C       IKODE...THE LENGTH OF THE COMMON PART OF THE CODES OF THE IWAVE-TH
C               AND IWAVE0-TH ELEMENTARY WAVES.
C THE INPUT PARAMETERS PAR1, PAR2 ARE NOT ALTERED.
C
C OUTPUT.
C       YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE INITIAL POINT OF
C               THE RAY (SEE C.R.T.5.5.4).  THE QUANTITIES ARE LISTED IN
C               THE SUBROUTINE PACKAGE 'RAY'.
C       Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY
C               AT THE INITIAL POINT OF THE RAY (SEE C.R.T.5.2.1).  THE
C               QUANTITIES ARE LISTED IN THE SUBROUTINE PACKAGE 'RAY'.
C       YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C               THE RAY AT THE INITIAL POINT OF THE RAY (SEE C.R.T.5.2.2).
C               THE QUANTITIES ARE LISTED IN THE SUBROUTINE PACKAGE 'RAY'.
C       IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C               ALONG THE RAY AT THE INITIAL POINT OF THE RAY (SEE
C               C.R.T.5.2.2).  THE QUANTITIES ARE LISTED IN THE SUBROUTINE
C               PACKAGE 'RAY'.
C       IEND... INFORMATION ON THE INITIAL POINT OF THE RAY:
C               0...   COMPUTATION OF THE RAY MAY FOLLOW.
C               71... THERE IS NO RAY CORRESPONDING TO THE GIVEN RAY
C                 PARAMETERS.  E.G., THE GIVEN PARAMETERS DO NOT BELONG TO
C                 THE DOMAIN OF THE INITIAL SURFACE.
C               72... INITIAL POINT OF THE RAY IS NOT SITUATED IN THE
C                 REQUIRED COMPLEX BLOCK.
C               74... RAY OF THE GENERATED WAVE IS NOT REAL-VALUED.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /INITC/:
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE.  ICB1I MUST BE ZERO BEFORE THE FIRST INVOCATION OF THIS
C SUBROUTINE, OTHER STORAGE LOCATIONS MAY BE UNDEFINED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL KOOR,METRIC,CODE,PARM2,VELOC,BLOCK,SMVPRD,INIS2
      INTEGER KOOR
C             FUNCTION KOOR()... SUBROUTINE PACKAGE 'METRIC'.
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)... SUBROUTINE PACKAGE
C           'METRIC'.
C     SUBROUTINE CODE(IY,KODIND,ICBNEW,IEND) AND SUBSEQUENT ROUTINES...
C             SUBROUTINE PACKAGE 'CODE'.
C     SUBROUTINE PARM2(ICB,COOR,UP,US,RO,QP,QS) AND SUBSEQUENT ROUTINES
C             ... SUBROUTINE PACKAGE 'PARM'.
C     SUBROUTINE VELOC(IWAVE,UP,US,QP,QS,VP,VS,VD,QL)
```

```
C     SUBROUTINE BLOCK(COOR,ISRF1,ISB1,ISRF2,ISB2,ICB2)... SUBROUTINE
C             PACKAGE 'MODEL'.
C     SUBROUTINE SMVPRD(G,A1,A2,A3,B1,B2,B3)... SUBROUTINE PACKAGE
C             'RAYCB'.
C     SUBROUTINE INIS2(NFUNC,PAR1,PAR2,E,F) AND SUBSEQUENT ROUTINES...
C             THIS SUBROUTINE PACKAGE.
C
C ERROR MESSAGES:
C     611...  WRONG FUNCTION OF SUBROUTINE CODE:
C             OTHER REASON OF THE TERMINATION OF THE RAY COMPUTATION
C             THAN 22 SHOULD NOT BE REPORTED BY THE SUBROUTINE CODE WHEN
C             REFERENCED BY THE SUBROUTINE INIT2.  CONTACT THE AUTHORS.
C     612...  WRONG FUNCTION OF SUBROUTINE CODE:
C             SUBROUTINE CODE REQUIRES THE FIRST RAY ELEMENT TO BE
C             SITUATED IN ANOTHER COMPLEX BLOCK THAN THE INITIAL POINT.
C             THIS ERROR SHOULD NOT APPEAR.  CONTACT THE AUTHORS.
C     613...  THIS PROGRAM BRANCH IS NOT CODED:
C             INTERPOLATION MODE OF THE COMPLETE RAY TRACING PROGRAM HAS
C             NOT BEEN ENABLED YET.
C
C DATE: 1990, JANUARY 25
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS FOR LOCAL MODEL PARAMETERS:
      REAL G(12),GAMMA(18),GSQRD,FAUX(10)
      REAL UP(10),US(10),RO,QP,QS,VP,VS,VD(10),QL
C     THESE AUXILIARY VARIABLES AND ARRAYS NEED NOT BE LOCATED IN A
C     COMMON BLOCK.  THERE IS NO REASON TO LOCATE THEM IN THE AUXILIARY
C     COMMON BLOCK /AUXMOD/ BUT TO SHARE THE MEMORY.
      COMMON/AUXMOD/ G,GAMMA,GSQRD,FAUX,UP,US,RO,QP,QS,VP,VS,VD,QL
C
C     G,GAMMA,GSQRD... SEE SUBROUTINE METRIC OF THE PACKAGE 'METRIC'.
C     FAUX... AUXILIARY ARRAY TO STORE A FUNCTIONAL VALUE AND ITS
C             DERIVATIVES.
C     UP,US,RO,QP,QS... SEE SUBROUTINE PARM2 OF THE PACKAGE 'PARM'.
C     UP,US,QP,QS,VP,VS,VD,QL... SEE SUBROUTINE VELOC OF THE PACKAGE
C             'MODEL'.
C
C.......................................................................
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER KODIND,ICBNEW,ISB2,ICB2,I,NY,INIDIM,NFUNC
      PARAMETER (NFUNC=4)
      REAL E11,F21,F31,FF11,FFE11,C11,B11,CB11,ECB11,U1,T1,BT1
      REAL E12,F22,F32,FF12,FFE12,C12,B12,CB12,ECB12,U2,T2,BT2
      REAL E22,F23,F33,FF22,FFE21,C22,B22,CB21,ECB21
      REAL E(3),F(6,NFUNC), FFE22,        CB22,ECB22
      EQUIVALENCE (E(1),E11),(E(2),E12),(E(3),E22)
      REAL V0,DETC,TRECE,Z1,Z2,Z3,AUX1,AUX2,AUX3
      SAVE V0
C
C     KODIND..POSITION IN THE CODE OF ELEMENTARY WAVE.
C     ICBNEW..THE INDEX OF THE COMPLEX BLOCK IN WHICH THE INITIAL POINT
C             OF THE RAY SHOULD BE SITUATED, SUPPLEMENTED BY THE SIGN
C             '+' FOR P WAVE OR '-' FOR S WAVE.  OUTPUT FROM SUBROUTINE
C             CODE.
C     ISB2... INDEX OF THE SIMPLE BLOCK IN WHICH THE INITIAL POINT IS
C             SITUATED.
```

```
C      ICB2... INDEX OF THE COMPLEX BLOCK IN WHICH THE INITIAL POINT IS
C              SITUATED.
C      I...    AUXILIARY AND LOOP VARIABLE.
C      NY...   NUMBER OF DEFINED LOCATIONS OF ARRAY Y.
C      INIDIM..DISTINGUISHES BETWEEN INITIAL POINT, LINE, OR SURFACE.
C      E=(E11,E12,E22)... PROJECTION MATRIX, SEE THE SUBROUTINE INIS2.
C      F...FUNCTIONS DESCRIBING THE INITIAL SURFACE AND THEIR
C              DERIVATIVES, SEE THE SUBROUTINE INIS2.
C      F21,F22,F23, F31,F32,F33... COVARIANT COMPONENTS OF THE VECTORS
C              F(2,1),F(2,2),F(2,3), F(3,1),F(3,2),F(3,3) TANGENT TO THE
C              INITIAL SURFACE.
C      FF11,FF12,FF22... COMPONENTS OF THE MATRIX AUXILIARY FF.
C      FFE11,FFE21,FFE12,FFE22... COMPONENTS OF THE MATRIX FF*E.
C      C11,C12,C22... COMPONENTS OF MATRIX C OF SPACE OR SPACE-TIME
C              SCALAR PRODUCTS, EVENTUALLY OF ITS SQUARE ROOT.
C      B11,B12,B22... COMPONENTS OF INVERSE SQUARE ROOT OF MATRIX C.
C      CB11,CB21,CB12,CB22... COMPONENTS OF THE MATRIX 1-C*B.
C      ECB11,ECB21,ECB12,ECB22... COMPONENTS OF THE MATRIX E*CB/TR(E*C).
C      U1,U2... SLOWNESS DERIVATIVES WITH RESPECT TO PAR1,PAR2.
C      T1,T2... VELOCITY*DERIVATIVES OF THE TRAVEL TIME ALONG THE INITIAL
C               SURFACE WITH RESPECT TO PAR1,PAR2.
C      BT1,BT2... COMPONENTS OF THE VECTOR (B+ECB)*T.
C      V0...   PROPAGATION VELOCITY AT THE INITIAL POINT.
C      DETC... DETERMINANT OF MATRIX C, EVENTUALLY OF ITS SQUARE ROOT.
C      TRECE... TRACE OF THE MATRIX E*C*E.
C      Z1,Z2,Z3... UNIT NORMAL TO THE INITIAL SURFACE - COVARIANT COMP.
C      AUX1,AUX2,AUX3... AUXILIARY STORAGE LOCATIONS.
C
C.......................................................................
C
      IF(MODCRT.GE.3) THEN
        STOP 'ERROR 613 IN INIT2: THIS PROGRAM BRANCH IS NOT CODED'
      END IF
C
      CALL INIS2(NFUNC,PAR1,PAR2,E,F)
      IF(E11.EQ.0..AND.E12.EQ.0..AND.E22.EQ.0.) THEN
C       INITIAL POINT:
        INIDIM=0
      ELSE IF(E11*E22-E12*E12.LE.0.001) THEN
C       INITIAL LINE:
        INIDIM=1
      ELSE
C       INITIAL SURFACE:
        INIDIM=2
      END IF
C
C     INITIAL TRAVEL TIME
      YI(1)=F(1,4)
C     INITIAL IMAGINARY TRAVEL TIME
      YI(2)=0.
C     COORDINATES OF THE INITIAL POINT OF THE RAY
      YI(3)=F(1,1)
      YI(4)=F(1,2)
      YI(5)=F(1,3)
C
C     MATERIAL PARAMETERS (DEFINING ISB1I, ICB1I, YL(1) TO YL(6)):
      IF(INIDIM.NE.0.OR.ICB1I.EQ.0) THEN
        CALL BLOCK(YI(3),0,ISB1I,I,ISB2,ICB2,FAUX)
        ISB1I=ISB2
      ELSE
```

```
            ICB2=IABS(ICB1I)
          ENDIF
          IY(2)=0
          IY(6)=0
          IY(8)=ICB2
          CALL CODE(IY,KODIND,ICBNEW,IEND)
          IF(IEND.EQ.22) THEN
            IEND=72
            RETURN
          ELSE IF(IEND.NE.0) THEN
            STOP 'ERROR 611 IN INIT2: WRONG FUNCTION OF SUBROUTINE CODE'
          END IF
          IF(ICB2.NE.IABS(ICBNEW)) THEN
            STOP 'ERROR 612 IN INIT2: WRONG FUNCTION OF SUBROUTINE CODE'
          END IF
          IF(INIDIM.GT.0.OR.ICB1I.EQ.0.OR.ICB1I.NE.ICBNEW) THEN
            ICB1I=ICBNEW
            CALL PARM2(IABS(ICBNEW),YI(3),UP,US,YLI(3),QP,QS)
            CALL VELOC(ICBNEW,UP,US,QP,QS,YLI(1),YLI(2),VD,QL)
            V0=VD(1)
            YLI(4)=VD(2)
            YLI(5)=VD(3)
            YLI(6)=VD(4)
          ENDIF
C
C         IMPORTANT QUANTITIES AT THE INITIAL POINT OF THE RAY (C.R.T.6.1):
C         SLOWNESS DERIVATIVES WITH RESPECT TO RAY PARAMETERS
          AUX1=-(YLI(4)*F(2,1)+YLI(5)*F(2,2)+YLI(6)*F(2,3))/(V0*V0)
          AUX2=-(YLI(4)*F(3,1)+YLI(5)*F(3,2)+YLI(6)*F(3,3))/(V0*V0)
          U1=E11*AUX1+E12*AUX2
          U2=E12*AUX1+E22*AUX2
C         COVARIANT COMPONENTS OF VECTORS TANGENT TO THE INITIAL SURFACE
          IF(KOOR().NE.0) THEN
            CALL METRIC(YI(3),GSQRD,G,GAMMA)
            CALL SMVPRD(G,F(2,1),F(2,2),F(2,3),F21,F22,F23)
            CALL SMVPRD(G,F(3,1),F(3,2),F(3,3),F31,F32,F33)
          ELSE
            GSQRD=1.
            F21=F(2,1)
            F22=F(2,2)
            F23=F(2,3)
            F31=F(3,1)
            F32=F(3,2)
            F33=F(3,3)
          END IF
C         SCALAR PRODUCTS OF VECTORS TANGENT TO THE INITIAL SURFACE
          C11=F(2,1)*F21+F(2,2)*F22+F(2,3)*F23
          C12=F(2,1)*F31+F(2,2)*F32+F(2,3)*F33
          C22=F(3,1)*F31+F(3,2)*F32+F(3,3)*F33
          DETC=C11*C22-C12*C12
C         UNIT NORMAL TO THE INITIAL SURFACE - COVARIANT COMPONENTS
          AUX1=GSQRD/SQRT(DETC)
          Z1=(F(2,2)*F(3,3)-F(2,3)*F(3,2))*AUX1
          Z2=(F(2,3)*F(3,1)-F(2,1)*F(3,3))*AUX1
          Z3=(F(2,1)*F(3,2)-F(2,2)*F(3,1))*AUX1
C         SLOWNESS VECTOR
          AUX1=( C22*F(2,4)-C12*F(3,4))/DETC
          AUX2=(-C12*F(2,4)+C11*F(3,4))/DETC
          AUX3=V0**(-2)-AUX1*F(2,4)-AUX2*F(3,4)
          IF(AUX3.LE.0.) THEN
```

```
C         EVANESCENT WAVE
          IEND=74
          RETURN
        END IF
        AUX3=SQRT(AUX3)
        YI(6)=F21*AUX1+F31*AUX2+Z1*AUX3
        YI(7)=F22*AUX1+F33*AUX2+Z2*AUX3
        YI(8)=F23*AUX1+F33*AUX2+Z3*AUX3
C       SPACE-TIME SCALAR PRODUCTS OF VECTORS TANGENT TO THE SURFACE
        T1=F(2,4)*V0
        T2=F(3,4)*V0
        C11=C11-T1*T1
        C12=C12-T1*T2
        C22=C22-T2*T2
        DETC=SQRT(C11*C22-C12*C12)
        IF(INIDIM.NE.1) THEN
C         INITIAL SURFACE OR INITIAL POINT:
C         SQUARE ROOT OF THE MATRIX C
          AUX1=SQRT(C11+C22+DETC+DETC)
          C11=(C11+DETC)/AUX1
          C12=C12/AUX1
          C22=(C22+DETC)/AUX1
C         INVERSE SQUARE ROOT OF THE MATRIX C
          B11= C22/DETC
          B12=-C12/DETC
          B22= C11/DETC
C         FIRST BASIS VECTOR OF RAY-CENTRED COORDINATE SYSTEM
          AUX3=V0*(B11*T1+B12*T2)
          YI(9) =F21*B11+F31*B12-YI(6)*AUX3
          YI(10)=F22*B11+F32*B12-YI(7)*AUX3
          YI(11)=F23*B11+F33*B12-YI(8)*AUX3
C         GEOMETRICAL SPREADING MATRIX Q
          YI(12)=C11*E11+C12*E12
          YI(13)=C12*E11+C22*E12
          YI(16)=C11*E12+C12*E22
          YI(17)=C12*E12+C22*E22
C         MATRIX P
          FF11=F(4,4)-YI(6)*F(4,1)-YI(7)*F(4,2)-YI(8)*F(4,3)-T1*U1
          FF12=F(5,4)-YI(6)*F(5,1)-YI(7)*F(5,2)-YI(8)*F(5,3)
          FF21=FF12-T2*U1
          FF12=FF12-T1*U2
          FF22=F(6,4)-YI(6)*F(6,1)-YI(7)*F(6,2)-YI(8)*F(6,3)-T2*U2
          YI(14)=B11*FF11+B12*FF21
          YI(15)=B12*FF11+B22*FF21
          YI(18)=B11*FF12+B12*FF22
          YI(19)=B12*FF12+B22*FF22
        ELSE
C         INITIAL LINE:
C         INFINITE PART OF THE INVERSE SQUARE ROOT OF THE MATRIX C
          B11=(1.-E11)/DETC
          B12=   -E12 /DETC
          B22=(1.-E22)/DETC
C         MATRIX CB=1-C*B
          CB11=1.-C11*B11+C12*B12
          CB21=  -C12*B11+C22*B12
          CB12=  -C11*B12+C12*B22
          CB22=1.-C12*B12+C22*B22
C         E-PROJECTION OF THE FINITE PART OF THE INVERSE SQUARE ROOT OF C
          TRECE=SQRT(E11*C11+2.*E12*C12+E22*C22)
          ECB11=(E11*CB11+E12*CB21)/TRECE
```

```
          ECB21=(E12*CB11+E22*CB21)/TRECE
          ECB12=(E11*CB12+E12*CB22)/TRECE
          ECB22=(E12*CB12+E22*CB22)/TRECE
C         FIRST BASIS VECTOR OF RAY-CENTRED COORDINATE SYSTEM
          AUX1=B11+ECB11
          AUX2=B12+ECB12
          BT1=AUX1*T1+AUX2*T2
          BT2=(B12+ECB21)*T1+(B22+ECB22)*T2
          AUX3=V0*BT1
          YI(9) =F21*AUX1+F31*AUX2-YI(6)*AUX3
          YI(10)=F22*AUX1+F32*AUX2-YI(7)*AUX3
          YI(11)=F23*AUX1+F33*AUX2-YI(8)*AUX3
C         GEOMETRICAL SPREADING MATRIX Q
          YI(12)=E11*TRECE
          YI(13)=E12*TRECE
          YI(16)=E12*TRECE
          YI(17)=E22*TRECE
C         MATRIX P
          FF11=F(4,4)-YI(6)*F(4,1)-YI(7)*F(4,2)-YI(8)*F(4,3)
          FF12=F(5,4)-YI(6)*F(5,1)-YI(7)*F(5,2)-YI(8)*F(5,3)
          FF22=F(6,4)-YI(6)*F(6,1)-YI(7)*F(6,2)-YI(8)*F(6,3)
          FFE11=FF11*E11+FF12*E12
          FFE21=FF12*E11+FF22*E12
          FFE12=FF11*E12+FF12*E22
          FFE22=FF12*E12+FF22*E22
          YI(14)=B11*FF11+B12*FF12+ECB11*FFE11+ECB12*FFE21-BT1*U1
          YI(15)=B12*FF11+B22*FF12+ECB12*FFE11+ECB22*FFE21-BT2*U1
          YI(18)=B11*FF12+B12*FF22+ECB11*FFE12+ECB12*FFE22-BT1*U2
          YI(19)=B12*FF12+B22*FF22+ECB12*FFE12+ECB22*FFE22-BT2*U2
        END IF
C       TAKE-OFF PARAMETERS
        YI(20)=PAR1
        YI(21)=PAR2
C
C       INITIAL VALUES FOR THE COMPLETE RAY TRACING (C.R.T.6.2):
        DO 10 I=1,6
          YL(I)=YLI(I)
   10   CONTINUE
        DO 20 I=1,11
          Y(I)=YI(I)
   20   CONTINUE
        IF(ICB1I.GE.0) THEN
          NY=27+2
        ELSE
          NY=27+8
        ENDIF
        DO 30 I=12,NY
          Y(I)=0.0
   30   CONTINUE
        Y(12)=1.0
        Y(17)=1.0
        Y(22)=1.0
        Y(27)=1.0
        Y(28)=1.0
        IF(NY.GE.34) Y(34)=1.0
        YY(1)=0.0
        YY(2)=UEB
        YY(3)=0.0
        YY(4)=0.0
        YY(5)=0.0
```

```
      IY(1)=NY
      IY(2)=KODIND
      IY(3)=0
      IY(4)=ISB1I
      IY(5)=ICB1I
      IY(6)=0
C     NOTE: IY(7),IY(8) MAY BE UNDEFINED
      IY(7)=0
      IY(8)=0
      IY(9)=0
      IY(10)=0
      IY(11)=0
      IY(12)=0
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE INIS1(LUN,NFUNC)
      INTEGER LUN,NFUNC
C
C SUBROUTINE INIS1 READS THE INPUT DATA FOR THE INITIAL POINTS OF RAYS
C AND STORES THEM IN COMMON BLOCK /INISC/, AND IF REQUIRED, CALLS THE
C SUBROUTINE VAL1 TO READ THE INPUT DATA FOR THE INTERPOLATED FUNCTIONS
C OF TWO VARIABLES (RAY PARAMETERS), TO DETERMINE THE COEFFICIENTS
C NECESSARY TO COMPUTE AN INTERPOLATORY FUNCTION ON A TWO DIMENSIONAL
C RECTANGULAR GRID, AND TO STORE THEM IN THE MEMORY.  THE FUNCTIONS
C DETERMINED CAN BE REPRESENTED AS A TENSOR PRODUCT OF SPLINES UNDER
C TENSION.  FOR ACTUAL MAPPING OF POINTS IT IS NECESSARY TO CALL THE
C SUBROUTINE INIS2, WHICH ALSO RETURNS THE FIRST AND SECOND PARTIAL
C DERIVATIVES.
C
C INPUT:
C     LUN...   LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C              CONTAINING THE INPUT DATA.
C     NFUNC...NUMBER OF THE FUNCTIONS REQUIRED TO BE DEFINED DURING THE
C              CURRENT INVOCATION OF INIS1.  SINCE THE FUNCTIONS
C              SPECIFIED IN THE INPUT DATA DO NOT COINCIDE WITH THE
C              REQUIRED FUNCTIONS BUT ARE TRANSFORMED TO THEM, NFUNC NEED
C              NOT EQUAL THE NUMBER OF FUNCTIONS SPECIFIED IN THE INPUT
C              DATA.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK /INISC/:
      INTEGER INIDIM,INIPAR
      REAL X1INI,X2INI,X3INI,TTINI
      COMMON/INISC/INIDIM,INIPAR,X1INI,X2INI,X3INI,TTINI
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL VAL1
C     VAL1, SORTV, READV... SUBROUTINE PACKAGE 'VAL'.
C     CURVN1 OR CURVB1 (ALTERNATIVES), SURFB1, VAL3B1, SNHCSH, VGEN,
C               TERMS, TRIDEC, TRISOL... SUBROUTINE PACKAGE 'FITPACK'.
C
C ERROR MESSAGES:
C     601...   SMALL NUMBER OF INPUT FUNCTIONS:
```

```
C              THE NUMBER OF INPUT FUNCTIONS IS LESS THAN THE NUMBER OF
C              FUNCTIONS NECESSARY TO DESCRIBE COORDINATES AND TRAVEL
C              TIME ALONG THE INITIAL SURFACE.
C      602...  WRONG VALUE OF INIPAR:
C              FOR INIDIM=1, THERE MUST BE INIPAR=1 OR INIPAR=2.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER LFUNC,MFUNC
      CHARACTER*80 TEXTI
      CHARACTER*3 TFUNCT
      DATA TFUNCT/'   '/
C
C     TEXTI...THE NAME OF THE DATA. STRING OF 80 CHARACTERS.
C     LFUNC...DIFFERENCE BETWEEN THE NUMBER NFUNC OF THE REQUIRED
C             FUNCTIONS AND THE NUMBER OF INPUT FUNCTIONS SPECIFYING
C             THEM.
C     MFUNC...NUMBER OF FUNCTIONS SPECIFIED IN THE INPUT DATA.
C
C     (1) READING THE NAME OF THE INPUT DATA
      READ(LUN,*) TEXTI
C
C     (2) QUANTITIES DEFINING THE KIND OF INITIAL CONDITIONS
      READ(LUN,*) INIDIM,INIPAR
C
C     (3) DATA DESCRIBING THE INITIAL POINT, CURVE OR SURFACE.
      IF(INIDIM.LE.0) THEN
        READ(LUN,*) X1INI,X2INI,X3INI,TTINI
      ELSE
        IF(INIDIM.EQ.2.AND.INIPAR.GT.0) THEN
          READ(LUN,*) X1INI,X2INI,X3INI
          LFUNC=2
        ELSE
          LFUNC=0
          IF(INIDIM.EQ.1.AND.(INIPAR.LT.1.OR.INIPAR.GT.2)) THEN
            STOP 'ERROR 602 IN INIS1: WRONG VALUE OF INIPAR'
          END IF
        END IF
        READ(LUN,*) MFUNC
        IF(NFUNC-LFUNC.LE.MFUNC) THEN
          CALL VAL1(LUN,3,MFUNC,1,TFUNCT)
        ELSE
          STOP 'ERROR 601 IN INIS1: SMALL NUMBER OF INPUT FUNCTIONS'
        END IF
      END IF
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE INIS2(NFUNC,PAR1,PAR2,E,F)
      INTEGER NFUNC
      REAL PAR1,PAR2,E(3),F(6,NFUNC)
C
C SUBROUTINE INIS2 EVALUATES THE FUNCTIONAL VALUES AND THE DERIVATIVES
```

```
C OF THE FUNCTIONS DESCRIBING THE INITIAL SURFACE.  THE FIRST THREE
C FUNCTIONS OF GIVEN RAY PARAMETERS ARE COORDINATES OF THE POINT
C CORRESPONDING TO THE GIVEN RAY PARAMETERS, THE FOURTH FUNCTION IS THE
C INITIAL VALUE OF THE TRAVEL TIME.  THE SINGLE INITIAL POINT COMMON TO
C ALL RAYS OR THE INITIAL LINE ARE TREATED AS SINGULAR LIMITING CASES OF
C THE INITIAL SURFACE.  THE INPUT DATA SPECIFYING THE FUNCTIONS MUST
C HAVE BEEN READ BY THE SUBROUTINE INIS1.
C
C INPUT:
C     NFUNC...NUMBER OF FUNCTIONS REQUIRED.  IT IS ASSUMED TO BE 4 IN
C             THIS VERSION (THREE COORDINATES AND THE TRAVEL TIME).
C     PAR1,PAR2... RAY PARAMETERS.
C     E...    ARRAY OF THE DIMENSION AT LEAST 3.
C     F...    ARRAY OF THE DIMENSION AT LEAST 6*NFUNC.
C NONE OF THE INPUT PARAMETERS EXCEPT E AND F ARE ALTERED.
C
C OUTPUT:
C     E...    ARRAY CONTAINING THE COMPONENTS E11, E12, E22 OF THE 2*2
C             SYMMETRIC PROJECTION MATRIX ONTO THE TANGENT SPACE TO THE
C             RAY PARAMETER'S MANIFOLD.  FOR A NON-DEGENERATE INITIAL
C             SURFACE, E IS THE IDENTITY MATRIX.  FOR A SINGLE INITIAL
C             POINT, E IS THE ZERO MATRIX.  FOR THE INITIAL LINE, E IS
C             THE PROJECTION MATRIX OF THE RANK 1.  NOTE THAT A
C             PROJECTION MATRIX E SATISFIES THE RELATION E*E=E.
C     F(1:6,I)... FOR A NON-DEGENERATE INITIAL SURFACE, THE VALUE AND
C             THE FIRST AND SECOND PARTIAL DERIVATIVES F, F1, F2, F11,
C             F12, F22 OF THE I-TH FUNCTION F(PAR1,PAR2).  NOTE THAT
C             F1 = E11,E12 * F1
C             F2   E12,E22   F2 ,
C             AND
C             F11,F12 = E11,E12 * F11,F12 * E11,E12
C             F12,F22   E12,E22   F12,F22   E12,E22 .
C             THUS, IN A DEGENERATE CASE (I.E. IF E IS NOT THE IDENTITY
C             MATRIX), THE FIRST DERIVATIVES ARE MODIFIED IN THE
C             FOLLOWING WAY,
C             F1 = F1 + F31 - E11,E12 * F31
C             F2   F2   F32   E12,E22   F32 ,
C             AND SECOND DERIVATIVES ARE MODIFIED AS FOLLOWS:
C             F11,F12 = F11,F12 + F311,F312 - E11,E12*F311,F312*E11,E12
C             F12,F22   F12,F22   F312,F322   E12,E22 F312,F322 E12,E22.
C             HERE F31, F32, F311, F312 AND F322 ARE THE DERIVATIVES OF
C             F1, F2, F11, F12 AND F22 WITH RESPECT TO THE SMALL
C             PARAMETER (E.G. A RADIUS) WHICH SHRINKS TO ZERO UPON AN
C             INITIAL LINE OR AT A SINGLE INITIAL POINT.
C
C COMMON BLOCK /INISC/:
      INTEGER INIDIM,INIPAR
      REAL X1INI,X2INI,X3INI,TTINI
      COMMON/INISC/INIDIM,INIPAR,X1INI,X2INI,X3INI,TTINI
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL METRIC,VAL2,SPHERE,SQRT3
C     SUBROUTINE METRIC(COOR,GSQRD,G,GAMMA)... SUBROUTINE PACKAGE
C            'METRIC'.
C     VAL2... SUBROUTINE PACKAGE 'VAL'.
C     CURV2D OR CURVBD (ALTERNATIVES), SURFBD, VAL3BD, SNHCSH, DSPLNZ,
C             INTRVL... SUBROUTINE PACKAGE 'FITPACK'.
C     SUBROUTINE SPHERE(INIPAR,PAR1,PAR2,FF)... THIS SUBROUTINE PACKAGE.
C     SUBROUTINE SQRT3(B,A)... THIS SUBROUTINE PACKAGE.
```

```
C
C DATE: 1990, FEBRUARY 5
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,I1,I2,I11,I22
      REAL DUMMY,R(3),G(12),FF(6,3)
C
C     I...    AUXILIARY LOOP VARIABLE.
C     I1,I11..ARRAY SUBSCRIPTS OF THE FIRST AND SECOND DERIVATIVES WITH
C             RESPECT TO INIPAR-TH RAY PARAMETER.
C     I2,I22..ARRAY SUBSCRIPTS OF THE FIRST AND SECOND DERIVATIVES WITH
C             RESPECT TO THE OTHER THAN INIPAR-TH RAY PARAMETER.
C     AUX1,AUX2... AUXILIARY STORAGE LOCATIONS.
C     DUMMY...DUMMY STORAGE LOCATION.
C     R...    ARRAY USED FOR GENERAL COORDINATES OR RAY PARAMETERS.
C     G...    G(1)-G(6)... COVARIANT COMPONENTS OF THE SYMMETRIC 3*3
C               METRIC TENSOR, OR CONTRAVARIANT COMPONENTS OF THE
C               SYMMETRIC 3*3 MATRIX OF THE BASIS VECTORS OF THE LOCAL
C               CARTESIAN COORDINATE SYSTEM (I.E. THE SQUARE ROOT OF THE
C               CONRAVARIANT METRIC TENSOR).
C             G(7)-G(12)... CONTRAVARIANT COMPONENTS OF THE SYMMETRIC
C               3*3 METRIC TENSOR.
C     FF...   GENERAL-PURPOSE AUXILIARY ARRAY.  USED TO STORE LOCAL
C             CARTESIAN COORDINATES AND THEIR DERIVATIVES WITH RESPECT
C             TO THE RAY PARAMETERS.  TEMPORARILY USED ALSO AS DUMMY
C             STORAGE LOCATION FOR CHRISTOFFEL SYMBOLS, FOR THE LAST
C             INTERPOLATED FUNCTION, E.T.C.
C
C.......................................................................
C
      IF(INIDIM.LE.1.OR.INIPAR.GE.1) THEN
C       MATRIX E OF LOCAL CARTESIAN COORDINATE SYSTEM BASIS VECTORS
        R(1)=X1INI
        R(2)=X2INI
        R(3)=X3INI
        CALL METRIC(R,DUMMY,G,FF)
      END IF
      IF(INIDIM.LE.0) THEN
C       SINGLE INITIAL POINT:
C       PROJECTION MATRIX
        E(1)=0.
        E(2)=0.
        E(3)=0.
C       CONTRAVARIANT COMPONENTS OF THE SYMMETRIC 3*3 MATRIX OF THE
C       BASIS VECTORS OF THE LOCAL CARTESIAN COORDINATE SYSTEM
        CALL SQRT3(G(7),G)
C       MAPPING OF THE RAY PARAMETERS ONTO A UNIT SPHERE
        CALL SPHERE(INIPAR,PAR1,PAR2,FF)
C       REQUIRED FUNCTIONS (3 GENERAL COORDINATES AND A TRAVEL TIME)
        F(1,1)=X1INI
        F(1,2)=X2INI
        F(1,3)=X3INI
        F(1,4)=TTINI
        DO 14 I=2,6
          F(I,1)=G(1)*FF(I,1)+G(2)*FF(I,2)+G(4)*FF(I,3)
          F(I,2)=G(2)*FF(I,1)+G(3)*FF(I,2)+G(5)*FF(I,3)
          F(I,3)=G(4)*FF(I,1)+G(5)*FF(I,2)+G(6)*FF(I,3)
```

```
            F(I,4)=0.
   14     CONTINUE
        ELSE
C         INITIAL LINE OR INITIAL SURFACE:
C         INTERPOLATED FUNCTIONS
          R(1)=PAR1
          R(2)=PAR2
          R(3)=0.
          IF(INIDIM.EQ.2.AND.INIPAR.GT.0) THEN
            LFUNC=2
          ELSE
            LFUNC=0
          END IF
          DO 21 I=LFUNC+1,NFUNC-1
            CALL VAL2(3,I-LFUNC,1,R,F(1,I),DUMMY)
   21     CONTINUE
          CALL VAL2(3,NFUNC-LFUNC,1,R,FF,DUMMY)
          DO 22 I=1,6
            F(I,NFUNC)=FF(I,1)
   22     CONTINUE
          IF(INIDIM.EQ.1) THEN
C           INITIAL LINE:
C           COVARIANT COMPONENTS OF THE VECTOR TANGENT TO THE INITIAL LINE
            I1=1+INIPAR
            FF(6,1)=G(1)*F(I1,1)+G(2)*F(I1,2)+G(4)*F(I1,3)
            FF(6,2)=G(2)*F(I1,1)+G(3)*F(I1,2)+G(5)*F(I1,3)
            FF(6,3)=G(4)*F(I1,1)+G(5)*F(I1,2)+G(6)*F(I1,3)
C           CONTRAVARIANT UNIT VECTOR TANGENT TO THE INITIAL LINE
            AUX2=F(I1,1)*FF(6,1)+F(I1,2)*FF(6,2)+F(I1,3)*FF(6,3)
            AUX1=SQRT(AUX2)
            FF(1,1)=F(I1,1)/AUX1
            FF(1,2)=F(I1,2)/AUX1
            FF(1,3)=F(I1,3)/AUX1
C           DERIVATIVE OF THE UNIT VECTOR TANGENT TO THE INITIAL LINE
            I11=2*I1
            AUX2=(F(I11,1)*FF(6,1)+F(I11,2)*FF(6,2)+F(I11,3)*FF(6,3))/AUX2
            FF(2,1)=(F(I11,1)-FF(1,1)*AUX2)/AUX1
            FF(2,2)=(F(I11,2)-FF(1,2)*AUX2)/AUX1
            FF(2,3)=(F(I11,3)-FF(1,3)*AUX2)/AUX1
C           COVARIANT VECTOR NORMAL TO THE INTERPOLATED SURFACE
            I2=5-I1
            FF(3,1)=FF(1,2)*F(I2,3)-FF(1,3)*F(I2,2)
            FF(3,2)=FF(1,3)*F(I2,1)-FF(1,1)*F(I2,3)
            FF(3,3)=FF(1,1)*F(I2,2)-FF(1,2)*F(I2,1)
C           DERIVATIVE OF THE VECTOR NORMAL TO THE INTERPOLATED SURFACE
            FF(4,1)=FF(2,2)*F(I2,3)-FF(2,3)*F(I2,2)+
     *              FF(1,2)*F(5,3) -FF(1,3)*F(5,2)
            FF(4,2)=FF(2,3)*F(I2,1)-FF(2,1)*F(I2,3)+
     *              FF(1,3)*F(5,1) -FF(1,1)*F(5,3)
            FF(4,3)=FF(2,1)*F(I2,2)-FF(2,2)*F(I2,1)+
     *              FF(1,1)*F(5,2) -FF(1,2)*F(5,1)
C           CONTRAVARIANT COMPONENTS
            FF(5,1)=G(7) *FF(3,1)+G(8) *FF(3,2)+G(10)*FF(3,3)
            FF(5,2)=G(8) *FF(3,1)+G(9) *FF(3,2)+G(11)*FF(3,3)
            FF(5,3)=G(10)*FF(3,1)+G(11)*FF(3,2)+G(12)*FF(3,3)
            FF(6,1)=G(7) *FF(4,1)+G(8) *FF(4,2)+G(10)*FF(4,3)
            FF(6,2)=G(8) *FF(4,1)+G(9) *FF(4,2)+G(11)*FF(4,3)
            FF(6,3)=G(10)*FF(4,1)+G(11)*FF(4,2)+G(12)*FF(4,3)
C           REQUIRED FUNCTIONS (3 GENERAL COORDINATES AND A TRAVEL TIME)
            E(2)=0.
```

```
            IF(INIPAR.LE.1) THEN
              E(1)=1.
              E(3)=0.
              AUX1=COS(PAR2)
              AUX2=SIN(PAR2)
            ELSE
              E(1)=0.
              E(3)=1.
              AUX1=COS(PAR1)
              AUX2=SIN(PAR1)
            END IF
            I22=10-I11
            DO 24 I=1,3
              F(I22,I)=-AUX2*F(I2,I)+AUX1*FF(5,I)
              F(I2,I) = AUX1*F(I2,I)+AUX2*FF(5,I)
              F(5,I)  = AUX1*F(5,I) +AUX2*FF(6,I)
   24       CONTINUE
            F(I22,4)=0.
            F(I2,4)=0.
            F(5,4)=0.
          ELSE
C           INITIAL SURFACE:
C           PROJECTION MATRIX
            E(1)=1.
            E(2)=0.
            E(3)=1.
C           REQUIRED FUNCTIONS (3 GENERAL COORDINATES AND A TRAVEL TIME)
            IF(INIPAR.GE.1) THEN
C             CONTRAVARIANT COMPONENTS OF THE SYMMETRIC 3*3 MATRIX OF THE
C             BASIS VECTORS OF THE LOCAL CARTESIAN COORDINATE SYSTEM
              CALL SQRT3(G(7),G)
C             MAPPING OF THE RAY PARAMETERS ONTO A UNIT SPHERE
              CALL SPHERE(INIPAR,PAR1,PAR2,FF)
C             LOCAL CARTESIAN COORDINATES
              DO 33 I=1,3
                FF(6,I)=F(6,3)*FF(1,I)+2.*F(3,3)*FF(3,I)+F(1,3)*FF(6,I)
                FF(5,I)=F(5,3)*FF(1,I)+F(3,3)*FF(2,I)+F(3,3)*FF(1,I)
     *                                                  +F(1,3)*FF(5,I)
                FF(4,I)=F(4,3)*FF(1,I)+2.*F(2,3)*FF(2,I)+F(1,3)*FF(4,I)
                FF(3,I)=F(3,3)*FF(1,I)+F(1,1)*FF(3,1)
                FF(2,I)=F(2,3)*FF(1,I)+F(1,1)*FF(2,1)
                FF(1,I)=F(1,3)*FF(1,I)
   33         CONTINUE
C             GENERAL COORDINATES
              DO 34 I=1,6
                F(I,1)=G(1)*FF(I,1)+G(2)*FF(I,2)+G(4)*FF(I,3)
                F(I,2)=G(2)*FF(I,1)+G(3)*FF(I,2)+G(5)*FF(I,3)
                F(I,3)=G(4)*FF(I,1)+G(5)*FF(I,2)+G(6)*FF(I,3)
   34         CONTINUE
              F(1,1)=F(1,1)+X1INI
              F(1,2)=F(1,2)+X2INI
              F(1,3)=F(1,3)+X3INI
            END IF
          END IF
        END IF
        RETURN
        END
C
C=======================================================================
C
```

```
      SUBROUTINE SQRT3(B,A)
      REAL B(6),A(6)
C
C SUBROUTINE SQRT3 EVALUATES THE SQUARE ROOT A OF THE GIVEN REAL-VALUED
C POSITIVE-SEMIDEFINITE SYMMETRIC 3*3 MATRIX B.  THE SQUARE ROOT IS THE
C REAL-VALUED POSITIVE-SEMIDEFINITE SYMMETRIC 3*3 MATRIX A SATISFYING
C THE EQUATION A*A=B.
C
C INPUT:
C     B...    ARRAY OF DIMENSION AT LEAST 6, CONTAINING THE COMPONENTS
C             A11, A12, A22, A13, A23, A33 OF THE GIVEN SYMMETRIC 3*3
C             MATRIX B.
C     A...    ARRAY OF DIMENSION AT LEAST 6.
C THE INPUT PARAMETER B IS NOT ALTERED.
C
C OUTPUT.
C     A...    ARRAY CONTAINING THE COMPONENTS A11, A12, A22, A13, A23,
C             A33 OF THE SYMMETRIC 3*3 MATRIX A WHICH IS THE SQUARE ROOT
C             OF THE GIVEN MATRIX B.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C ERROR MESSAGES:
C     614...  MATRIX IS NOT POSITIVE-SEMIDEFINITE:
C             INPUT MATRIX B IS NOT POSITIVE-SEMIDEFINITE.
C     615...  THIS PROGRAM BRANCH IS NOT CODED:
C             THE SQUARE ROOT OF GENERAL SYMMETRIC MATRIX B HAS NOT BEEN
C             CODED.  AT PRESENT, ONLY THE SQUARE ROOT OF DIAGONAL
C             MATRIX CAN BE EVALUATED.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     NO AUXILIARY STORAGE LOCATIONS
C
      IF(B(2).EQ.0..AND.B(4).EQ.0..AND.B(5).EQ.0.) THEN
C       DIAGONAL MATRIX
        IF(B(1).LT.0..OR.B(3).LT.0..OR.B(6).LT.0.) THEN
C         B IS NOT POSITIVE-SEMIDEFINITE MATRIX
          STOP 'ERROR 614 IN SQRT3: MATRIX IS NOT POSITIVE-SEMIDEFINITE'
        ELSE
          A(1)=SQRT(B(1))
          A(2)=0.
          A(3)=SQRT(B(3))
          A(4)=0.
          A(5)=0.
          A(6)=SQRT(B(6))
        END IF
      ELSE
C       GENERAL SYMMETRIC MATRIX
        STOP 'ERROR 615 IN SQRT3: THIS PROGRAM BRANCH IS NOT CODED'
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SPHERE(INIPAR,PAR1,PAR2,FF)
```

```
      INTEGER INIPAR
      REAL PAR1,PAR2,FF(6,3)
C
C SUBROUTINE SPHERE TRANSFORMS SPHERICAL COORDINATES PAR1, PAR2 INTO THE
C CARTESIAN COORDINATES OF THE CORRESPONDING POINT ON THE UNIT SPHERE.
C IT ALSO EVALUATES THE FIRST AND SECOND DERIVATIVES OF THE CARTESIAN
C COORDINATES WITH RESPECT TO PAR1 AND PAR2.
C
C INPUT:
C     INIPAR..DETERMINES THE TYPE OF SPHERICAL COORDINATES:
C             INIPAR.LE.1: POLAR-LIKE SPHERICAL COORDINATES (COLATITUDE,
C               LONGITUDE).
C             INIPAR.GE.2: GEOGRAPHIC-LIKE SPHERICAL COORDINATES
C               (LONGITUDE, LATITUDE).
C     PAR1,PAR2... RAY PARAMETERS.
C     FF...   ARRAY OF THE DIMENSION AT LEAST 6*3.
C NONE OF THE INPUT PARAMETERS EXCEPT FF ARE ALTERED.
C
C OUTPUT:
C     FF(1:6,I)...I-TH CARTESIAN COORDINATE OF THE POINT ON THE UNIT
C             SPHERE GIVEN BY PAR1, PAR2, AND ITS FIRST AND SECOND
C             PARTIAL DERIVATIVES WITH RESPECT TO PAR1 AND PAR2 IN THE
C             ORDER FF, FF1, FF2, FF11, FF12, FF22.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      REAL C1,C2,S1,S2
C
C     C1,C2...COSINES OF THE TAKE-OFF ANGLES AT A SINGLE INITIAL POINT.
C     S1,S2...SINES OF THE TAKE-OFF ANGLES AT A SINGLE INITIAL POINT.
C
      C1=COS(PAR1)
      S1=SIN(PAR1)
      C2=COS(PAR2)
      S2=SIN(PAR2)
      IF(INIPAR.LE.1) THEN
C       POLAR-LIKE SPHERICAL COORDINATES
        FF(1,1)=S1*C2
        FF(1,2)=S1*S2
        FF(1,3)=C1
        FF(2,1)= C1*C2
        FF(2,2)= C1*S2
        FF(2,3)=-S1
        FF(3,1)=-S1*S2
        FF(3,2)= S1*C2
        FF(3,3)=0.
        FF(4,1)=-S1*C2
        FF(4,2)=-S1*S2
        FF(4,3)=-C1
        FF(5,1)=-C1*S2
        FF(5,2)= C1*C2
        FF(5,3)=0.
        FF(6,1)=-S1*C2
        FF(6,2)=-S1*S2
```

```
          FF(6,3)=0.
        ELSE
C         GEOGRAPHIC-LIKE SPHERICAL COORDINATES
          FF(1,1)=C2*C1
          FF(1,2)=C2*S1
          FF(1,3)=S2
          FF(2,1)=-C2*S1
          FF(2,2)= C2*C1
          FF(2,3)=0.
          FF(3,1)=-S2*C1
          FF(3,2)=-S2*S1
          FF(3,3)= C2
          FF(4,1)=-C2*C1
          FF(4,2)=-C2*S1
          FF(4,3)=0.
          FF(5,1)= S2*S1
          FF(5,2)=-S2*C1
          FF(5,3)=0.
          FF(6,1)=-C2*C1
          FF(6,2)=-C2*S1
          FF(6,3)=-S2
        END IF
        RETURN
        END
C
C======================================================================
C
```

```
C SUBROUTINE PACKAGE 'RPAR' TO CONTROL THE TAKE-OFF PARAMETERS OF RAYS.
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING SUBROUTINES:
C       RPARB...BLOCK DATA SUBROUTINE DEFINING COMMON BLOCK /RPARC/ TO
C               TO STORE THE INPUT DATA FOR THE TAKE-OFF PARAMETERS OF
C               RAYS.
C       RPAR1...SUBROUTINE DESIGNED TO READ THE INPUT DATA FOR THE
C               TAKE-OFF PARAMETERS , TO PREPARE THE PARAMETERS FOR THE
C               SUBROUTINE RPAR2 AND TO STORE THEM IN THE COMMON BLOCK
C               /RPARC/.  IT IS CALLED WHEN STARTING THE COMPUTATION OF A
C               NEW ELEMENTARY WAVE.
C       RPAR2...SUBROUTINE DESIGNED TO SPECIFY THE TAKE-OFF PARAMETERS OF
C               THE INDIVIDUAL RAYS.
C       RPAR31..SUBROUTINE CALLED WITH CONSTANT STEP STORE OF THE
C               INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE POINTS OF
C               INTERSECTION WITH INTERFACES EITHER BEFORE AND AFTER THE
C               TRANSFORMATION, I.E. CALLED SIMULTANEOUSLY WITH THE
C               SUBROUTINE WRIT31 (SEE ALSO C.R.T.5.5.1).  IT MAY BE USED
C               TO KEEP THE QUANTITIES USEFUL FOR THE DETERMINATION OF THE
C               TAKE-OFF PARAMETERS IN THE MEMORY.
C       RPAR32..SUBROUTINE CALLED AT THE POINTS OF INTERSECTION OF THE RAY
C               WITH SPECIFIED INTERFACES, I.E. CALLED SIMULTANEOUSLY WITH
C               THE SUBROUTINE WRIT32 (SEE ALSO C.R.T.5.5.2).  IT MAY BE
C               USED TO KEEP THE QUANTITIES USEFUL FOR THE DETERMINATION
C               OF THE TAKE-OFF PARAMETERS IN THE MEMORY.
C       RPAR33..SUBROUTINE CALLED AT THE ENDPOINTS OF THE ELEMENTS OF
C               RAYS, SITUATED AT STRUCTURAL INTERFACES, I.E. CALLED
C               SIMULTANEOUSLY WITH THE SUBROUTINE WRIT33 (SEE ALSO
C               C.R.T.5.5.3).  IT MAY BE USED TO KEEP THE QUANTITIES
C               USEFUL FOR THE DETERMINATION OF THE TAKE-OFF PARAMETERS IN
C               THE MEMORY.
C       RPAR4...SUBROUTINE CALLED AFTER FINISHING THE COMPUTATION OF EACH
C               RAY.  IT DEFINES FOUR STORAGE LOCATIONS OF THE COMMON
C               BLOCK /INITC/ INTRODUCED IN THE PACKAGE 'INIT'.
C
C INPUT DATA FOR THE TAKE-OFF PARAMETERS OF RAYS:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  IF
C     THE FIRST LETTER OF THE SYMBOLIC NAME OF THE INPUT VARIABLE IS
C     I-N, THE CORRESPONDING VALUE IN INPUT DATA MUST BE OF THE TYPE
C     INTEGER.  OTHERWISE (EXCEPT TEXTP), THE INPUT PARAMETER IS OF THE
C     TYPE REAL.
C (1) TEXTP
C     STRING DESCRIBING THE DATA.  ONLY THE FIRST 80 CHARACTERS OF THE
C     STRING ARE SIGNIFICANT.
C FOR EACH ELEMENTARY WAVE IWAVE THE FOLLOWING DATA (2) AND (3):
C (2) ANY TIMES:
C     PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
C     THE FIRST RAY TAKE-OFF PARAMETER PAR1 TAKES THE VALUES FROM PAR1L
C     TO PAR1M WITH A FIXED STEP OF THE SIZE ABS(PAR1D).  FOR PAR1D=0, A
C     SINGLE VALUE PAR1=PAR1L OF THE FIRST RAY PARAMETER IS CONSIDERED.
C     THE SECOND RAY TAKE-OFF PARAMETER PAR2 TAKES THE VALUES FROM PAR2L
C     TO PAR2M WITH A FIXED STEP OF THE SIZE ABS(PAR2D).  FOR PAR2D=0,
C     A SINGLE VALUE PAR1=PAR1L OF THE SECOND RAY PARAMETER IS
C     CONSIDERED.
C     DEFAULT VALUES: ZEROES WHEN STARTING THE COMPLETE RAY TRACING
C     PROGRAM, OTHERWISE THE PREVIOUS VALUES.
```

```
C (3) /(A SLASH)
C
C STORAGE IN THE MEMORY:
C      THE INPUT DATA (2) ARE STORED IN COMMON BLOCK /RPARC/ DEFINED IN
C      THE FOLLOWING SUBROUTINE:
C      -----------------------------------------------------------------
       BLOCK DATA RPARB
       INTEGER LURPAR
       REAL PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
       COMMON/RPARC/ LURPAR,PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
       SAVE /RPARC/
       END
C      -----------------------------------------------------------------
C      LURPAR..LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C              CONTAINING THE INPUT DATA.
C      PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D... LAST VALUES OF THE INPUT
C              DATA (2).
C
C DATE: 1990, JANUARY 24
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
      SUBROUTINE RPAR1(LUN,IWAVE)
      INTEGER LUN,IWAVE
C
C THIS SUBROUTINE IS CALLED WHEN STARTING THE COMPUTATION OF A NEW
C ELEMENTARY WAVE.  IT PREPARES THE PARAMETERS FOR THE SUBROTINE RPAR2
C TO SPECIFY THE TAKE-OFF PARAMETERS OF THE INDIVIDUAL RAYS.
C
C INPUT:
C     LUN...   LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C              CONTAINING THE INPUT DATA.
C     IWAVE...INDEX OF THE ELEMENTARY WAVE THAT IS GOING TO BE COMPUTED,
C              I.E. THE OUTPUT FROM THE LAST INVOCATION OF THE SUBROUTINE
C              CODE1.
C
C NO OUTPUT.
C
C COMMON BLOCK /RPARC/:
      INTEGER LURPAR
      REAL PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
      COMMON/RPARC/ LURPAR,PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1990, JANUARY 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      CHARACTER*80 TEXTP
C
C     TEXTP...THE NAME OF THE DATA. STRING OF 80 CHARACTERS.
C
      IF(IWAVE.EQ.1) THEN
C       READING THE NAME OF THE INPUT DATA
```

```
          READ(LUN,*) TEXTP
C
C         DEFAULT VALUES
          PAR1L=0.
          PAR1M=0.
          PAR1D=0.
          PAR2L=0.
          PAR2M=0.
          PAR2D=0.
        END IF
C
        LURPAR=LUN
        RETURN
        END
C
C=======================================================================
C
        SUBROUTINE RPAR2(IRAY,PAR1,PAR2)
        REAL PAR1,PAR2
        INTEGER IRAY
C
C THIS SUBROUTINE DETERMINES THE TAKE-OFF PARAMETERS OF THE RAY.
C
C INPUT:
C       IRAY... NUMBER OF THE ALREADY COMPUTED RAYS.  IRAY=0 WHEN
C               BEGINNING THE COMPUTATION OF A NEW ELEMENTARY WAVE.
C               OTHERWISE, THE OUTPUT FROM THE PREVIOUS INVOCATION OF
C               RPAR2.
C
C OUTPUT:
C       IRAY... IRAY=0 IF ALL RAYS ARE COMPUTED AND THE COMPUTATION OF THE
C               ELEMENTARY WAVE WILL BE TERMINATED.
C               OTHERWISE, INPUT VALUE INCREASED BY 1.
C       PAR1,PAR2... TAKE-OFF PARAMETERS OF THE RAY.
C NONE OF THE INPUT PARAMETERS EXCEPT IRAY ARE ALTERED.
C
C COMMON BLOCK /RPARC/:
        INTEGER LURPAR
        REAL PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
        COMMON/RPARC/ LURPAR,PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
C ALL THE STORAGE LOCATIONS, EXCEPT LURPAR, OF THE COMMON BLOCK ARE
C REDEFINED IN THIS SUBROUTINE.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1989, DECEMBER 21
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C       AUXILIARY STORAGE LOCATIONS:
C       REAL AUX1,AUX2,AUX3,AUX4,AUX5,AUX6
C
        IRAY=IRAY+1
        IF(IRAY.EQ.1) THEN
          GO TO 10
        END IF
        PAR1=PAR1+PAR1D
        IF(PAR1D*(PAR1-PAR1M-PAR1D/1000.).GE.0.) THEN
          PAR1=PAR1L
```

```
          PAR2=PAR2+PAR2D
          IF(PAR2D*(PAR2-PAR2M-PAR2D/1000.).GE.0.) THEN
            GO TO 10
          END IF
        END IF
        RETURN
C
C       READING INPUT DATA:
   10 CONTINUE
      AUX1=PAR1L
      AUX2=PAR1M
      AUX3=PAR1D
      AUX4=PAR2L
      AUX5=PAR2M
      AUX6=PAR2D
      READ(LURPAR,*) PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
      IF(IRAY.NE.1) THEN
        IF(PAR1L.EQ.AUX1.AND.PAR1M.EQ.AUX2.AND.PAR1D.EQ.AUX3.AND.
     *     PAR2L.EQ.AUX4.AND.PAR2M.EQ.AUX5.AND.PAR2D.EQ.AUX6) THEN
          IRAY=0
          RETURN
        END IF
      END IF
      IF((PAR1M-PAR1L)*PAR1D.LT.0.) THEN
        PAR1D=-PAR1D
      END IF
      IF((PAR2M-PAR2L)*PAR2D.LT.0.) THEN
        PAR2D=-PAR2D
      END IF
      PAR1=PAR1L
      PAR2=PAR2L
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE RPAR31(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SUBROUTINE MAY BE USED TO STORE THE QUANTITIES USEFUL FOR THE
C DETERMINATION OF THE TAKE-OFF PARAMETERS IN THE MEMORY.  IT IS CALLED
C WITH CONSTANT STEP STORE (SEE THE INPUT DATA IN THE SUBROUTINE PACKAGE
C 'RAY') OF THE INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE POINTS OF
C INTERSECTION WITH INTERFACES EITHER BEFORE AND AFTER THE
C TRANSFORMATION.
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
```

```
C DATE: 1990, JANUARY 15
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE RPAR32(ISTOR,YL,Y,YY,IY,NAMPL,YC)
      INTEGER ISTOR,IY(12),NAMPL
      REAL YL(6),Y(27),YY(5),YC(NAMPL)
C
C THIS SUBROUTINE MAY BE USED TO STORE THE QUANTITIES USEFUL FOR THE
C DETERMINATION OF THE TAKE-OFF PARAMETERS IN THE MEMORY.  IT IS CALLED
C AT THE POINTS OF INTERSECTION OF THE RAY WITH INTERFACES SPECIFIED IN
C THE INPUT DATA IN THE SUBROUTINE PACKAGE 'RAY' FOR STORING THE
C COMPUTED QUANTITIES (SEE ALSO C.R.T.5.5.2).
C
C INPUT:
C     ISTOR...THE SEQUENTIAL NUMBER IN THE IPUT DATA OF THE SPECIFIED
C             SURFACE.
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C             QUANTITIES Y(28) TO Y(IY(1)) REPRESENTING REDUCED
C             AMPLITUDES ARE IGNORED.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C     NAMPL...NUMBER OF REAL QUANTITIES REPRESENTING COMPLEX-VALUED
C             VECTORIAL REDUCED AMPLITUDES.  IF NO CONVERSION
C             COEFFICIENTS ARE APPLIED NAMPL=IY(1)-27,  OTHERWISE
C             NAMPL=6 OR 12 (SEE C.R.T.5.5.4).
C     YC...   ARRAY CONTAINING REAL QUANTITIES REPRESENTING COMPLEX-
C             -VALUED VECTORIAL REDUCED AMPLITUDES.  IF NO CONVERSION
C             COEFFICIENTS ARE APPLIED, YC IS A COPY OF Y(28) TO
C             Y(IY(1)).  OTHERWISE,  YC REPRESENTS THE VECTORIAL REDUCED
C             AMPLITUDES INVOLVING APPROPRIATE CONVERSION COEFFICIENTS,
C             EXPRESSED IN RAY-CENTRED COORDINATE SYSTEM (SEE
C             C.R.T.5.5.4):
C             P WAVE AT THE INITIAL POINT OF THE RAY:
C               NAMPL=6,
C               YC(1)=REAL(A13),  YC(2)=AIMAG(A13),
C               YC(3)=REAL(A23),  YC(4)=AIMAG(A23),
C               YC(5)=REAL(A33),  YC(6)=AIMAG(A33).
C             S WAVE AT THE INITIAL POINT OF THE RAY:
C               NAMPL=12,
C               YC(1)=REAL(A11),  YC(2)=AIMAG(A11),
C               YC(3)=REAL(A21),  YC(4)=AIMAG(A21),
C               YC(5)=REAL(A31),  YC(6)=AIMAG(A31),
C               YC(7)=REAL(A12),  YC(8)=AIMAG(A12),
C               YC(9)=REAL(A22),  YC(10)=AIMAG(A22),
C               YC(11)=REAL(A32), YC(12)=AIMAG(A32).
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
```

```
C
C DATE: 1990, JANUARY 15
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE RPAR33(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SUBROUTINE MAY BE USED TO STORE THE QUANTITIES USEFUL FOR THE
C DETERMINATION OF THE TAKE-OFF PARAMETERS IN THE MEMORY.  IT IS CALLED
C AT THE ENDPOINTS OF THE ELEMENTS OF RAYS, SITUATED AT STRUCTURAL
C INTERFACES  (SEE ALSO C.R.T.5.5.3).
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.  FOR THE DEFINITION OF ARRAYS YL, Y, YY AND
C             IY SEE THE PACKAGE 'RAY'.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE RPAR4(IRAY,PAR1,PAR2,YL,Y,YY,IY,IEND)
C
      REAL PAR1,PAR2,YL(6),Y(35),YY(5)
      INTEGER IY(12),IEND,IRAY
C
C THIS SUBROUTINE IS CALLED AFTER FINISHING THE COMPUTATION OF A RAY.
C IT DEFINES THE STORAGE LOCATIONS YI(22) TO YI(25) OF THE COMMON BLOCK
C /INITC/.
C
C INPUT:
C     IRAY... INDEX OF THE RAY, I.E. THE NUMBER OF THE ALREADY COMPUTED
C             RAYS, I.E. THE OUTPUT FROM THE LAST INVOCATION OF RPAR2.
C     PAR1,PAR2... TAKE-OFF PARAMETERS OF THE RAY.
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
```

```
C                THE RAY.
C       IY...    ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C                ALONG THE RAY.  FOR THE DEFINITION OF ARRAYS YL, Y, YY AND
C                IY SEE THE PACKAGE 'RAY'.
C       IEND...  REASON OF THE TERMINATION OF THE COMPUTATION OF THE RAY
C                (SEE C.R.T.5.4).  FOR A DETAILED DESCRIPTION SEE
C                SUBROUTINE RAY (SUBROUTINE PACKAGE 'RAY').
C
C NO OUTPUT.
C
C COMMON BLOCK /INITC/ (SEE SUBROUTINE PACKAGE 'INIT'):
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
C     ..............................................................
C     YI(22)..AREA OF THE ELEMENT OF THE RAY-PARAMETER SURFACE,
C             CORRESPONDING TO THE RAY,
C     YI(23),YI(24),YI(25)... COMPONENTS 11, 12, 22 OF THE SYMMETRIC
C             MATRIX INVERSE TO THE SPECIFIC MOMENT OF THE ELEMENT OF
C             THE RAY-PARAMETER SURFACE CORRESPONDING TO THE RAY, SEE
C             C.R.T.6.1.
C LOCATIONS YI(22) TO YI(25) OF THE COMMON BLOCK ARE DEFINED IN THIS
C SUBROUTINE, OTHER STORAGE LOCATIONS REMAIN UNCHANGED.
C
C COMMON BLOCK /RPARC/:
      INTEGER LURPAR
      REAL PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
      COMMON/RPARC/ LURPAR,PAR1L,PAR1M,PAR1D,PAR2L,PAR2M,PAR2D
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     NO AUXILIARY STORAGE LOCATIONS.
C
      IF(PAR1D.EQ.0.) THEN
        YI(22)=1.
        YI(23)=0.
      ELSE
        YI(22)=ABS(PAR1D)
        YI(23)=12./(PAR1D*PAR1D)
      END IF
      IF(PAR2D.EQ.0.) THEN
        YI(25)=0.
      ELSE
        YI(22)=YI(22)*ABS(PAR2D)
        YI(25)=12./(PAR2D*PAR2D)
      END IF
      YI(24)=0.
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'WRIT' TO CREATE THE OUTPUT OF COMPLETE RAY TRACING
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF:
C     WRITB...BLOCK DATA SUBROUTINE DEFINING COMMON BLOCKS /WRITT/,
C             /WRITC/ AND /WRITN/ TO STORE THE INPUT DATA ON THE NAMES
C             OF OUTPUT FILES.
C     WRIT1...SUBROUTINE CALLED WHEN STARTING THE COMPLETE RAY TRACING
C             PROGRAM, AND WHEN STARTING THE COMPUTATION OF A NEW
C             ELEMENTARY WAVE.
C     WRIT2...SUBROUTINE CALLED WHEN STARTING THE COMPLETE TRACING OF A
C             NEW RAY.
C     WRIT31..SUBROUTINE STORING THE COMPUTED QUANTITIES ALONG A RAY
C             (C.R.T.5.5.1).  IT IS CALLED WITH CONSTANT STEP STORE (SEE
C             INPUT DATA IN THE PACKAGE 'RAY') OF THE INDEPENDENT
C             VARIABLE ALONG THE RAY, AND AT THE POINTS OF INTERSECTION
C             WITH INTERFACES EITHER BEFORE AND AFTER THE
C             TRANSFORMATION.
C     WRIT32..SUBROUTINE STORING THE COMPUTED QUANTITIES AT SPECIFIED
C             SURFACES (C.R.T.5.5.2).  IT IS CALLED AT THE POINTS OF
C             INTERSECTION OF THE RAY WITH THE SPECIFIED SURFACES.
C     WRIT33..SUBROUTINE STORING THE COMPUTED QUANTITIES AT THE
C             ENDPOINTS OF THE INDIVIDUAL ELEMENTARY WAVES
C             (C.R.T.5.5.3).  IT IS CALLED AT THE ENDPOINTS OF THE
C             ELEMENTS OF RAYS, SITUATED AT STRUCTURAL INTERFACES.
C     WRIT4...SUBROUTINE STORING THE QUANTITIES AT THE INITIAL POINT OF
C             THE RAY (C.R.T.6.1).  IT IS CALLED AFTER TERMINATION OF
C             TRACING THE RAY.
C     CHECK...AUXILIARY SUBROUTINE TO WRIT4.
C     WRIT5...SUBROUTINE CALLED AFTER TERMINATION OF THE COMPUTATION OF
C             AN ELEMENTARY WAVE, AND WHEN TERMINATING THE COMPLETE RAY
C             TRACING PROGRAM.
C     FNAME...SUBROUTINE ANALYSING THE SPECIFIED STRINGS AND COMPOSING
C             THE FILENAME OF THEM.  AUXILIARY SUBROUTINE TO WRIT1.
C
C INPUT DATA TO SPECIFY THE NAMES OF THE OUTPUT FILES WITH THE COMPUTED
C QUANTITIES:
C     THE DATA ARE READ IN BY THE LIST DIRECTED INPUT (FREE FORMAT).  IN
C     THE LIST OF INPUT DATA BELOW, EACH NUMBERED PARAGRAPH INDICATES
C     THE BEGINNING OF A NEW INPUT OPERATION (NEW READ STATEMENT).  ALL
C     INPUT VARIABLES ARE OF THE TYPE CHARACTER.  THE DATA ARE READ BY
C     THE SUBROUTINE WRIT1.
C     THE FILENAMES ARE GENERATED USING SEVERAL COMPONENTS (STRINGS) IN
C     THE FOLLOWING WAY:  EACH COMPONENT (STRING) IS DIVIDED INTO WORDS,
C     EACH WORD BEING FOLLOWED BY JUST ONE SPACE.  HERE A WORD IS A
C     SUBSTRING CONTAINING NO SPACE, PRECEDED AND FOLLOWED BY SPACES OR
C     BY THE BEGINNING OR END OF THE STRING.  AN EMPTY WORD LIES BETWEEN
C     TWO CONSECUTIVE SPACES.  THE FILENAME IS COMPOSED OF
C             THE 1-ST WORD OF THE 1-ST COMPONENT,
C             THE 1-ST WORD OF THE 2-ND COMPONENT,
C             ...
C             THE 1-ST WORD OF THE LAST COMPONENT,
C             THE 2-ND WORD OF THE 1-ST COMPONENT,
C             THE 2-ND WORD OF THE 2-ND COMPONENT,
C             ...
C             THE LAST WORD OF THE LAST COMPONENT.
C     EXAMPLE:
C             1-ST COMPONENT = 'D/  .OUT',
C             2-ND COMPONENT = 'A B',
```

```
C              3-RD COMPONENT = '1 2',
C              RESULTING FILENAME = 'D/A1B2.OUT'.
C      NOTE THAT ONLY FIRST 16 CHARACTERS OF EACH FILENAME COMPONENT ARE
C      SIGNIFICANT.
C (1)  TEXTW
C      STRING DESCRIBING THE DATA.  ONLY THE FIRST 80 CHARACTERS OF THE
C      STRING ARE SIGNIFICANT.
C (2)  FN1A,FN1I
C      FN1A... STRING CONTAINING THE FIRST COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED ALONG THE
C              RAYS (C.R.T.5.5.1).
C      FN1I... STRING CONTAINING THE FIRST COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE QUANTITIES AT THE INITIAL POINTS OF
C              RAYS (C.R.T.6.1), RELATED TO THE OUTPUT FILE WITH THE
C              COMPUTED QUANTITIES STORED ALONG THE RAYS (C.R.T.5.5.1).
C              THE OTHER COMPONENTS OF THE FILENAME ARE COMMON WITH THE
C              RELATED FILE CONTAINING THE COMPUTED QUANTITIES.
C (3)  FN2A,FN2I,(FN2B(I),I=1,NENDF),/(A SLASH)
C      FN2A... STRING CONTAINING THE FIRST COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED AT THE
C              SPECIFIED SURFACES (C.R.T.5.5.2).
C      FN2I... STRING CONTAINING THE FIRST COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE QUANTITIES AT THE INITIAL POINTS OF
C              RAYS (C.R.T.6.1), RELATED TO THE OUTPUT FILE WITH THE
C              COMPUTED QUANTITIES STORED AT THE SPECIFIED SURFACES
C              (C.R.T.5.5.2).  THE OTHER COMPONENTS OF THE FILENAME ARE
C              COMMON WITH THE RELATED FILE CONTAINING THE COMPUTED
C              QUANTITIES.
C      FN2B(I)... STRING CONTAINING THE SECOND COMPONENT OF THE NAME OF
C              THE OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED AT THE
C              I-TH SPECIFIED SURFACE (C.R.T.5.5.2).  THE ABOVE COUNT
C              NENDF MUST BE GREATER OR EQUAL TO THE NUMBER OF SPECIFIED
C              SURFACES.
C (4)  FN3A,FN3I,(FN3B(I),I=1,NELEM),/(A SLASH)
C      FN3A... STRING CONTAINING THE FIRST COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED AT THE
C              ENDPOINTS OF THE ELEMENTS OF RAYS (C.R.T.5.5.3).
C      FN3I... STRING CONTAINING THE FIRST COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE QUANTITIES AT THE INITIAL POINTS OF
C              RAYS (C.R.T.6.1), RELATED TO THE OUTPUT FILE WITH THE
C              COMPUTED QUANTITIES STORED AT THE ENDPOINTS OF THE
C              ELEMENTS OF RAYS (C.R.T.5.5.3).  THE OTHER COMPONENTS OF
C              THE FILENAME ARE COMMON WITH THE RELATED FILE CONTAINING
C              THE COMPUTED QUANTITIES.
C      FN3B(I)... STRING CONTAINING THE SECOND COMPONENT OF THE NAME OF
C              THE OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED AT THE
C              ENDPOINTS OF THE I-TH ELEMENT OF RAYS (C.R.T.5.5.3).
C FOR EACH ELEMENTARY WAVE IWAVE THE FOLLOWING DATA (5):
C (5)  FN1C,FN2C,FN3C
C      FN1C... STRING CONTAINING THE SECOND COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED ALONG THE
C              RAYS (C.R.T.5.5.1).
C      FN2C... STRING CONTAINING THE THIRD COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED AT THE
C              SPECIFIED SURFACES (C.R.T.5.5.2).
C      FN3C... STRING CONTAINING THE THIRD COMPONENT OF THE NAME OF THE
C              OUTPUT FILE WITH THE COMPUTED QUANTITIES STORED AT THE
C              ENDPOINTS OF THE ELEMENTS OF RAYS (C.R.T.5.5.3).
C
C STORAGE IN THE MEMORY:
```

```
C     THE INPUT DATA (1) TO (5) DESCRIBING THE NAMES OF THE OUTPUT FILES
C     WITH THE COMPUTED QUANTITIES ARE STORED IN THE COMMON BLOCK
C     /WRITT/.  OTHER IMPORTANT VARIABLES SHARED BY THE SUBROUTINES OF
C     THIS PACKAGE ARE STORED IN THE COMMON BLOCKS /WRITC/ AND /WRITN/.
C     THE COMMON BLOCKS ARE DEFINED IN THE FOLLOWING SUBROUTINE:
C     -----------------------------------------------------------------
      BLOCK DATA WRITB
      INTEGER MF
      PARAMETER (MF=64)
      CHARACTER*80 TEXTW
      CHARACTER*16 FN1A,FN2A,FN3A,FN1I,FN2I,FN3I,FNB(MF),FN1C,FN2C,FN3C
      COMMON/WRITT/TEXTW,
     *             FN1A,FN2A,FN3A,FN1I,FN2I,FN3I,FNB,FN1C,FN2C,FN3C
      SAVE /WRITT/
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      SAVE /WRITC/
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
      SAVE /WRITN/
      DATA LUW/10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25/
      DATA JENDR/10,21,22,23,24,25,26,30,32,33,40,50,60,61,71,72,74/
      END
C     -----------------------------------------------------------------
C     TEXTW...THE NAME OF THE DATA SET. STRING OF 80 CHARACTERS.
C     FN1A,FN2A,FN3A,FN1I,FN2I,FN3I,FNB(MF),FN1C,FN2C,FN3C... STORAGE
C             LOCATIONS FOR THE INPUT DATA (2) TO (5).
C     JWAVE...INDEX OF THE CURRENT ELEMENTARY WAVE.
C     JRAY... INDEX OF THE CURRENT RAY.
C     JUEB... COUNT OF RAYS EXCEEDING THE SPECIFIED UPPER BOUNDS OF THE
C             CHECKED QUANTITIES.
C     JFCT... TOTAL NUMBER OF INVOCATIONS OF THE SUBROUTINE FCT.
C     JOUTP...TOTAL NUMBER OF STEPS OF THE NUMERICAL INTEGRATION.
C     JTRANS..TOTAL NUMBER OF INVOCATIONS OF THE SUBROUTINE TRANS.
C     KWRIT1,KWRIT2,KWRIT3... NUMBERS OF INITIAL RAY ELEMENTS ALONG
C             WHICH NOTHING IS STORED IN THE SUBROUTINES WRIT31, WRIT32
C             AND WRIT33, RESPECTIVELY.
C     NF1...  NUMBER OF OPEN FILES, SEE THE INPUT DATA (2) AND THE
C             SUBROUTINE WRIT31.
C     NF2...  NUMBER OF OPEN FILES, SEE THE INPUT DATA (3) AND THE
C             SUBROUTINE WRIT32.
C     NF3...  NUMBER OF OPEN FILES, SEE THE INPUT DATA (4) AND THE
C             SUBROUTINE WRIT33.
C     MF3...  NUMBER OF FILES CORRESPONDING TO THE CURRENT ELEMENTARY
C             WAVE, SEE THE INPUT DATA (4) AND THE SUBROUTINE WRIT33.
C     NUW...  THE MAXIMUM COUNT OF AVAILABLE LOGICAL UNIT NUMBERS.
C     LUW...  LIST OF THE LOGICAL UNIT NUMBERS CORRESPONDING TO THE
C             OUTPUT FILES.
C     JPOINT..ARRAY CONTAINING NUMBERS OF POINTS STORED INTO THE FILES.
C     MENDR...TOTAL NUMBER OF DISTINCT REASONS OF THE TERMINATION OF THE
C             COMPUTATION OF THE RAY, SEE SUBROUTINE RAY2 OF THE PACKAGE
C             'RAY' AND SUBROUTINE INIT2 OF THE PACKAGE 'INIT'.
C     JENDR...LIST OF THE INDICES CORRESPONDING TO DISTINCT REASONS OF
C             THE TERMINATION OF THE COMPUTATION OF THE RAY, SEE
C             SUBROUTINE RAY2 OF THE PACKAGE 'RAY' AND SUBROUTINE INIT2
```

```
C              OF THE PACKAGE 'INIT'.
C       NENDR...NENDR(I),I=1,MENDR... NUMBER OF RAYS TERMINATED FOR THE
C                 REASON JENDR(I).
C               NENDR(MENDR+1)... NUMBER OF RAYS TERMINATED FOR THE REASON
C                 NOT LISTED IN THE ARRAY JENDR.
C               NENDR(MENDR+2)... TOTAL NUMBER OF COMPUTED RAYS OF THE
C                 CURRENT ELEMENTARY WAVE.
C       NAME2...LIST OF THE NAMES OF THE OPEN FILES.
C
C OUTPUT FILES WITH THE COMPUTED QUANTITIES STORED ALONG THE RAYS
C (C.R.T.5.5.1):
C       UNFORMATTED OUTPUT.  THE COMPUTED QUANTITIES ARE STORED WITH THE
C       STEP STORE (SEE THE INPUT DATA IN THE SUBROUTINE PACKAGE 'RAY')
C       ALONG ALL RAYS, AND AT ALL POINTS OF INTERSECTION OF RAYS WITH
C       STRUCTURAL INTERFACES EITHER BEFORE AND AFTER THE TRANSFORMATION.
C       FOR EACH POINT - ONE RECORD CONTAINING THE FOLLOWING QUANTITIES:
C (1)   IWAVE,IRAY,NY,ICB1,ISRF,(YL(I),I=1,6),(Y(I),I=1,NY)
C       IWAVE...INDEX OF THE ELEMENTARY WAVE (OUTPUT OF THE SUBROUTINE
C               CODE1).  ELEMENTARY WAVES ARE INDEXED BY 1,2,3,... .
C       IRAY... INDEX OF THE RAY (OUTPUT OF THE SUBROUTINE RPAR2).  RAYS
C               WITHIN EACH ELEMENTARY WAVE ARE INDEXED BY 1,2,3,... .
C               NOTE THAT SOME OF THE INDEXED RAYS NEED NOT EXIST.
C       NY=IY(1)=27+NAMPL... NUMBER OF THE BASIC QUANTITIES DESCRIBING THE
C               POINT OF THE RAY, SEE THE PACKAGE 'RAY'.
C       ICB1=IY(5)... INDEX OF THE COMPLEX BLOCK IN WHICH THE STORED POINT
C               OF THE RAY IS SITUATED, SUPPLEMENTED BY A SIGN '+' FOR P
C               WAVE AND SIGN '-' FOR S WAVE.
C       ISRF=IY(6)... INDEX OF THE SURFACE AT WHICH THE ENDPOINT OF THE
C               COMPUTED ELEMENT OF THE RAY IS SITUATED, SUPPLEMENTED BY
C               A SIGN '+' OR '-' FOR THE ENDPOINT SITUATED AT THE
C               POSITIVE OR NEGATIVE SIDE OF THE SURFACE, RESPECTIVELY.
C               ZERO INSIDE THE COMPLEX BLOCK.  NOTE: THE SIGN OF THIS
C               QUANTITY IS THE EXEPTION TO THE DEFINITION IN THE ORIGINAL
C               PAPER ON C.R.T.
C       YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY
C               (C.R.T.5.5.4).
C       Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY
C               (C.R.T.5.2.1).
C       THE DETAILED DESCRIPTION OF THE QUANTITIES YL, YY AND IY MAY BE
C       FOUND IN THE PACKAGE 'RAY'.
C
C OUTPUT FILES WITH THE COMPUTED QUANTITIES STORED AT THE SPECIFIED
C SURFACES (C.R.T.5.5.2):
C       UNFORMATTED OUTPUT.  THE COMPUTED QUANTITIES ARE STORED AT THE
C       POINTS OF INTERSECTION OF RAYS WITH THE SPECIFIED SURFACES.
C       FOR EACH POINT - ONE RECORD CONTAINING THE FOLLOWING QUANTITIES:
C (1)   IWAVE,IRAY,NY,ICB1,ISRF,(YL(I),I=1,6),(Y(I),I=1,NY)
C       THE SAME QUANTITIES AS IN THE DATA SET DESCRIBED ABOVE, EXCEPT
C       THAT THE VECTORIAL REDUCED AMPLITUDES Y(28)-Y(IY(1)) MAY (BUT NEED
C       NOT) BE REPLACED BY THE REDUCED AMPLITUDES YC(1) TO YC(NY-27)
C       INVOLVING APPROPRIATE CONVERSION COEFFICIENTS, SEE C.R.T.5.5.4.
C       YC(1) TO YC(NY-27) ARE DESCRIBED IN THE SUBROUTINE CONV (PACKAGE
C       'TRANS') AND IN THE SUBROUTINE WRIT32 (THIS PACKAGE).  NOTE THAT
C       NY=33 FOR P WAVE AND NY=39 FOR S WAVE AT THE INITIAL POINT OF THE
C       RAY.
C
C OUTPUT FILES WITH THE COMPUTED QUANTITIES STORED AT THE ENDPOINTS OF
C THE ELEMENTS OF RAYS (C.R.T.5.5.3):
C       UNFORMATTED OUTPUT.  THE FILES, GENERATED ONLY IF KSTORE.GE.2 (SEE
C       INPUT DATA IN THE SUBROUTINE PACKAGE 'RAY'),  ARE AUXILIARY DISK
```

```
C     STORAGE LOCATIONS TO THE PROGRAM CRT AND ARE NOT INTENDED TO BE
C     THE OUTPUT OF THE COMPLETE RAY TRACING USED BY THE USER
C     APPLICATION PROGRAMS.  THE COMPUTED QUANTITIES ARE STORED AT THE
C     ENDPOINTS OF THE ELEMENTS OF ALL RAYS BEFORE THE TRANSFORMATION.
C     FOR EACH POINT - ONE RECORD CONTAINING THE FOLLOWING QUANTITIES:
C (1) IRAY,(IY(I),I=1,12),(YL(I),I=1,6),(Y(I),I=1,IY(1)),(YY(I),I=1,5)
C     IRAY... INDEX OF THE RAY (OUTPUT OF THE SUBROUTINE RPAR2).  RAYS
C             WITHIN EACH ELEMENTARY WAVE ARE INDEXED BY 1,2,3,... .
C             NOTE THAT SOME OF THE INDEXED RAYS MAY NOT EXIST.
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C             (C.R.T.5.5.4).
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C             (C.R.T.5.2.1).
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY (C.R.T.5.2.2).
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY (C.R.T.5.2.2).
C     THE DETAILED DESCRIPTION OF THE QUANTITIES YL, Y, YY AND IY MAY BE
C     FOUND IN THE PACKAGE 'RAY'.
C
C OUTPUT FILES WITH THE QUANTITIES AT THE INITIAL POINTS OF RAYS
C (C.R.T.6.1):
C     UNFORMATTED OUTPUT.  THE QUANTITIES ARE STORED AT THE INITIAL
C     POINTS OF ALL RAYS.
C     FOR EACH RAY - ONE RECORD CONTAINING FOLLOWING 34 QUANTITIES:
C (1) (-IWAVE),IRAY,ICB1I,(YLI(I),I=1,6),(YI(I),I=1,25)
C     IWAVE...INDEX OF THE ELEMENTARY WAVE (OUTPUT OF THE SUBROUTINE
C             CODE1).  ELEMENTARY WAVES ARE INDEXED BY 1,2,3,... .
C     IRAY... INDEX OF THE RAY (OUTPUT OF THE SUBROUTINE RPAR2).  RAYS
C             WITHIN EACH ELEMENTARY WAVE ARE INDEXED BY 1,2,3,... .
C             NOTE THAT SOME OF THE INDEXED RAYS MAY NOT EXIST.
C     ICB1I...INDEX OF THE COMPLEX BLOCK IN WHICH THE INITIAL POINT OF
C             THE RAY IS SITUATED, SUPPLEMENTED BY A SIGN '+' FOR P WAVE
C             AND SIGN '-' FOR S WAVE (SEE C.R.T.6.1).
C     YLI...  ARRAY CONTAINING THE VALUES OF THE QUANTITIES YL(1)-YL(6),
C             SEE C.R.T.5.5.4, DESCRIBING THE LOCAL PROPERTIES OF THE
C             MODEL AT THE INITIAL POINT OF THE RAY.  SEE THE LIST OF
C             YL(1) TO YL(6) IN THE PACKAGE 'RAY'.
C     YI...   ARRAY CONTAINING THE QUANTITIES DESCRIBING THE PROPERTIES
C             OF THE RAYS AND OF THE TRAVEL-TIME FIELD AT THE INITIAL
C             POINT OF THE RAY, SEE C.R.T.6.1.  THESE QUANTITIES ARE
C             LISTED IN THE PACKAGE 'INIT'.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
      SUBROUTINE WRIT1(LUN,LULOG,IWAVE,IWAVE0,IKODE)
      INTEGER LUN,LULOG,IWAVE,IWAVE0,IKODE
C
C THIS SUBROUTINE IS CALLED WHEN STARTING THE COMPLETE RAY TRACING
C PROGRAM, AND WHEN STARTING THE COMPUTATION OF A NEW ELEMENTARY WAVE.
C
C INPUT:
C     LUN...  LOGICAL UNIT NUMBER OF THE EXTERNAL INPUT DEVICE
C             CONTAINING THE INPUT DATA.
C     LULOG...LOGICAL UNIT NUMBER OF THE LOG OUTPUT DEVICE.
C     IWAVE...ZERO WHEN STARTING THE COMPLETE RAY TRACING PROGRAM,
C             OTHERWISE THE INDEX OF THE ELEMENTARY WAVE WHICH WILL BE
```

```
C               COMPUTED (I.E. THE OUTPUT OF THE SUBROUTINE CODE1 FROM THE
C               PACKAGE 'CODE').
C       IWAVE0..INDEX OF THE ALREADY COMPUTED ELEMENTARY WAVE HAVING THE
C               MOST NUMEROUS COMMON ELEMENTS WITH THE CURRENT ELEMENTARY
C               WAVE.  NEED NOT BE DEFINED IF IWAVE=0.
C       IKODE...THE LENGTH OF THE COMMON PART OF THE CODES OF THE IWAVE-TH
C               AND IWAVE0-TH ELEMENTARY WAVES.  NEED NOT BE DEFINED IF
C               IWAVE=0.
C
C NO OUTPUT.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /WRITT/:
      INTEGER MF
      PARAMETER (MF=64)
      CHARACTER*80 TEXTW
      CHARACTER*16 FN1A,FN2A,FN3A,FN1I,FN2I,FN3I,FNB(MF),FN1C,FN2C,FN3C
      COMMON/WRITT/TEXTW,
     *             FN1A,FN2A,FN3A,FN1I,FN2I,FN3I,FNB,FN1C,FN2C,FN3C
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C ALL THE STORAGE LOCATIONS OF THE COMMON BLOCKS ARE DEFINED IN THIS
C SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL LCODE,NELEM,SCRO1,FNAME
      INTEGER LCODE,NELEM
C     FUNCTION LCODE()... SUBROUTINE PACKAGE 'CODE'.
C     FUNCTION NELEM(IKODE)... SUBROUTINE PACKAGE 'CODE'.
C     SUBROUTINE SCRO1(IWAVE)... SUBROUTINE PACKAGE 'SCRO'.
C     SUBROUTINE FNAME(NUM1,NAME1,NAME2)... THIS PACKAGE.
C
C ERROR MESSAGES:
C     551...  OPEN FILE ERROR:
C             OPEN FILE ERROR WHEN OPENING THE FILE TO STORE:
C             1 COMPUTED QUANTITIES ALONG THE RAYS.
C             2 COMPUTED QUANTITIES AT THE SPECIFIED SURFACES.
C             3 COMPUTED QUANTITIES AT THE ENDPOINTS OF THE ELEMENTS OF
C               THE RAYS OF THE ELEMENTARY WAVE.
C             4 QUANTITIES AT THE INITIAL POINTS OF RAYS, CORRESPONDING
C               TO THE ABOVE FILE 1.
```

```
C                5 QUANTITIES AT THE INITIAL POINTS OF RAYS, CORRESPONDING
C                  TO THE ABOVE FILE 2.
C                6 QUANTITIES AT THE INITIAL POINTS OF RAYS, CORRESPONDING
C                  TO THE ABOVE FILE 3.
C                THE TYPE 1 TO 6 OF THE FILE IS GIVEN BY THE FIRST DECIMAL
C                PLACE OF THE STATUS.
C       554...   FEW LOGICAL UNITS AVAILABLE:
C                THERE ARE NUW LOGICAL UNITS AVAILABLE, SEE THE COMMON
C                BLOCKS /WRITC/ AND /WRITN/ DEFINED IN THE BLOCK DATA
C                WRITB.  THIS NUMBER SHOULD BE INCREASED.
C       555...   FEW FILENAMES SPECIFIED:
C                THERE ARE MORE SURFACES TO STORE THE COMPUDED QUANTITIES
C                THAN THE CORRESPONDING SPECIFIED FILENAMES.
C       556...   FEW FILENAMES SPECIFIED:
C                THERE ARE MORE ELEMENTS OF THE RAYS OF AN ELEMENTARY WAVE
C                THAN THE CORRESPONDING SPECIFIED FILENAMES.
C       557...   FEW FILENAMES SPECIFIED:
C                THERE ARE MORE REQUIRED ELEMENTARY WAVES THAN THE
C                CORRESPONDING SPECIFIED FILENAMES.
C       558...   END OF INPUT FILE:
C                END OF INPUT DATA FILE WRIT SPECIFYING THE NAMES OF THE
C                OUTPUT FILES ENCOUNTERED BEFORE ALL REQUIRED ELEMENTARY
C                WAVES ARE COMPUTED.  THE NUMBER OF ELEMENTARY WAVES
C                EXCEEDS THE NUMBER OF SPECIFIED OUTPUT FILENAMES.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER NKODE,NE1,NE2,I,J
      CHARACTER*16 NAME1(3)
C
      IF(IWAVE.LE.0) THEN
C
C       READING THE NAME OF THE INPUT DATA
        READ(LUN,*) TEXTW
C
C       READING STRINGS COMPOSING THE NAMES OF OUTPUT FILES (CRT.5.5.1)
        FN1A=' '
        FN1I=' '
        READ(LUN,*) FN1A,FN1I
C       NUMBER OF OUTPUT FILES TO BE OPEN
        IF(STORE.EQ.0.) THEN
          NF1=0
        ELSE
          NF1=1
        END IF
C
C       READING STRINGS COMPOSING THE NAMES OF OUTPUT FILES (CRT.5.5.2)
        FN2A=' '
        FN2I=' '
        DO 21 I=1,MF
          FNB(I)=' '
   21   CONTINUE
        READ(LUN,*) FN2A,FN2I,(FNB(I),I=1,MF)
C       NUMBER OF OUTPUT FILES TO BE OPEN
        NF2=NSTOR
        DO 22 I=1,NSTOR
```

```
            IF(FNB(I).EQ.' ') THEN
              STOP 'ERROR 555 IN WRIT1: FEW FILENAMES SPECIFIED'
            END IF
   22     CONTINUE
          IF(2*(NF1+NF2).GT.NUW) THEN
            STOP 'ERROR 554 IN WRIT1: FEW LOGICAL UNITS AVAILABLE'
          END IF
C
C         READING STRINGS COMPOSING THE NAMES OF OUTPUT FILES (CRT.5.5.3)
          FN3A=' '
          FN3I=' '
          DO 31 I=NF2+1,MF
            FNB(I)=' '
   31     CONTINUE
          READ(LUN,*) FN3A,FN3I,(FNB(I),I=NF2+1,MF)
C         NUMBER OF OUTPUT FILES TO BE OPEN
          IF(MODCRT.LE.1) THEN
            MF3=0
          ELSE
            DO 32 I=NF2+1,MF
              IF(FNB(I).EQ.' ') THEN
                MF3=I-1-NF2
                GO TO 33
              END IF
   32       CONTINUE
            MF3=MF-NF2
   33       CONTINUE
          END IF
C
        ELSE
C
C         NUMBERS OF INITIAL RAY ELEMENTS ALONG WHICH NOTHING IS STORED
          IF(MODCRT.EQ.0) THEN
            KWRIT1=0
          ELSE
            KWRIT1=NELEM(IKODE)
          END IF
          KWRIT2=KWRIT1
          IF(MODCRT.LE.1) THEN
            KWRIT3=MKODE
          ELSE
            KWRIT3=NELEM(IKODE)
          END IF
C
C         READING STRINGS COMPOSING THE NAMES OF OUTPUT FILES
          FN1C=' '
          FN2C=' '
          FN3C=' '
          DO 41 I=JWAVE+1,IWAVE
            READ(LUN,*,END=98) FN1C,FN2C,FN3C
   41     CONTINUE
C
C         OPENING THE OUTPUT FILES (CRT.5.5.1)
          IF(IWAVE.EQ.1.OR.FN1C.NE.' ') THEN
            IF(NF1.NE.0.) THEN
              NAME1(1)=FN1A
              NAME1(2)=FN1C
              CALL FNAME(2,NAME1,NAME2(1))
              OPEN(LUW(1),FILE=NAME2(1),FORM='UNFORMATTED',IOSTAT=IE1)
              IF(IE1.NE.0) THEN
```

```
C             OPEN FILE ERROR
              IE2=1
              J=1
              GO TO 91
            END IF
            NAME1(1)=FN1I
            CALL FNAME(2,NAME1,NAME2(2))
            OPEN(LUW(2),FILE=NAME2(2),FORM='UNFORMATTED',IOSTAT=IE1)
            IF(IE1.NE.0) THEN
C             OPEN FILE ERROR
              IE2=4
              J=2
              GO TO 91
            END IF
          END IF
        END IF
C
C       OPENING THE OUTPUT FILES (CRT.5.5.2)
        IF(IWAVE.EQ.1.OR.FN2C.NE.' ') THEN
          DO 42 I=1,NF2
            NAME1(1)=FN2A
            NAME1(2)=FNB(I)
            NAME1(3)=FN2C
            J=2*(NF1+I)-1
            CALL FNAME(3,NAME1,NAME2(J))
            OPEN(LUW(J),FILE=NAME2(J),FORM='UNFORMATTED',IOSTAT=IE1)
            IF(IE1.NE.0) THEN
C             OPEN FILE ERROR
              IE2=2
              GO TO 91
            END IF
            NAME1(1)=FN2I
            J=2*(NF1+I)
            CALL FNAME(3,NAME1,NAME2(J))
            OPEN(LUW(J),FILE=NAME2(J),FORM='UNFORMATTED',IOSTAT=IE1)
            IF(IE1.NE.0) THEN
C             OPEN FILE ERROR
              IE2=5
              GO TO 91
            END IF
   42     CONTINUE
        END IF
C
C       OPENING THE OUTPUT FILES (CRT.5.5.3)
        IF(MODCRT.GE.1) THEN
          IF(FN3C.EQ.' ') THEN
            STOP 'ERROR 557 IN WRIT1: FEW FILENAMES SPECIFIED'
          ELSE
            NKODE=NELEM(LCODE())
C           NKODE IS THE NUMBER OF RAY ELEMENTS
            IF(MODCRT.GE.3) NKODE=MIN0(KWRIT3+1,NKODE)
C           NKODE IS THE INDEX OF THE LAST RAY ELEMENT TO BE STORED
            NF3=NKODE-KWRIT3
            IF(NF3.GT.MF3) THEN
              STOP 'ERROR 556 IN WRIT1: FEW FILENAMES SPECIFIED'
            ELSE IF(2*(NF1+NF2+NF3).GT.NUW) THEN
              STOP 'ERROR 554 IN WRIT1: FEW LOGICAL UNITS AVAILABLE'
            END IF
            DO 45 I=1,NF3
              NAME1(1)=FN3A
```

```
                  NAME1(2)=FNB(KWRIT3+I)
                  NAME1(3)=FN3C
                  J=2*(NF1+NF2+I)-1
                  CALL FNAME(3,NAME1,NAME2(J))
                  OPEN(LUW(J),FILE=NAME2(J),FORM='UNFORMATTED',IOSTAT=IE1)
                  IF(IE1.NE.0) THEN
C                   OPEN FILE ERROR
                    IE2=5
                    GO TO 91
                  END IF
                  NAME1(1)=FN3I
                  J=2*(NF1+NF2+I)
                  CALL FNAME(3,NAME1,NAME2(J))
                  OPEN(LUW(J),FILE=NAME2(J),FORM='UNFORMATTED',IOSTAT=IE1)
                  IF(IE1.NE.0) THEN
C                   OPEN FILE ERROR
                    IE2=6
                    GO TO 91
                  END IF
   45           CONTINUE
              END IF
            END IF
          END IF
C
C         INITIAL VALUES FOR THE ELEMENTARY WAVE
          JWAVE=IWAVE
          JRAY=0
          JUEB=0
          JFCT=0
          JOUTP=0
          JTRANS=0
          DO 51 I=1,MENDR+2
            NENDR(I)=0
   51     CONTINUE
          DO 52 I=1,2*(NF1+NF2+NF3)
            JPOINT(I)=0
   52     CONTINUE
C
C         WRITING TO THE OUTPUT LOG FILE
          IF(IWAVE.NE.0) THEN
            WRITE(LULOG,81) IWAVE
   81       FORMAT(' WAVE',I5,':')
          END IF
C
C         SCREEN OUTPUT
          CALL SCRO1(IWAVE)
          RETURN
C
   91     CONTINUE
          WRITE(*,'(''STATUS'',I4,''.'',I1,'': '',A)') IE1,IE2,NAME2(J)
          STOP 'ERROR 551 IN WRIT1: OPEN FILE ERROR'
   98     CONTINUE
          STOP 'ERROR 558 IN WRIT1: END OF INPUT FILE'
          END
C
C=======================================================================
C
          SUBROUTINE WRIT2(LULOG,IRAY)
          INTEGER LULOG,IRAY
C
```

```
C THIS SUBROUTINE IS CALLED WHEN STARTING THE COMPLETE TRACING OF A NEW
C RAY.
C
C INPUT:
C     LULOG...LOGICAL UNIT NUMBER OF THE LOG OUTPUT DEVICE.
C     IRAY... THE INDEX OF THE RAY WHICH WILL BE COMPUTED (I.E. THE
C             OUTPUT OF THE SUBROUTINE RPAR2 FROM THE PACKAGE 'RPAR').
C
C NO OUTPUT.
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C JRAY IS REDEFINED IN THIS SUBROUTINE.  NO OTHER OF THE STORAGE
C LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SCRO2
C     SUBROUTINE SCRO2(IRAY)... SUBROUTINE PACKAGE 'SCRO'.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     NO AUXILIARY STORAGE LOCATIONS.
C
      JRAY=IRAY
C
C     SCREEN OUTPUT
      CALL SCRO2(IRAY)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE WRIT31(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SUBROUTINE STORES THE COMPUTED QUANTITIES ALONG A RAY (SEE
C C.R.T.5.5.1).  IT IS CALLED WITH CONSTANT STEP STORE OF THE
C INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE POINTS OF INTERSECTION
C WITH INTERFACES EITHER BEFORE AND AFTER THE TRANSFORMATION.
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
```

```
C
C NO OUTPUT.
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C ARRAY JPOINT IS MODIFIED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL NELEM,SCRO3
      INTEGER NELEM
C     FUNCTION NELEM(KODIND)... PACKAGE 'CODE'.
C     SUBROUTINE SCRO3(YL,Y,YY,IY)... SUBROUTINE PACKAGE 'SCRO'.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I
C
      IF(NELEM(IY(2)).GT.KWRIT1) THEN
        WRITE(LUW(1)) JWAVE,JRAY,IY(1),IY(5),IY(6),YL,(Y(I),I=1,IY(1))
        JPOINT(1)=JPOINT(1)+1
      END IF
C
C     SCREEN OUTPUT
      CALL SCRO3(YL,Y,YY,IY)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE WRIT32(ISTOR,YL,Y,YY,IY,NAMPL,YC)
      INTEGER ISTOR,IY(12),NAMPL
      REAL YL(6),Y(27),YY(5),YC(NAMPL)
C
C THIS SUBROUTINE STORES THE COMPUTED QUANTITIES AT SPECIFIED SURFACES
C (SEE C.R.T.5.5.2).  IT IS CALLED AT THE POINTS OF INTERSECTION OF THE
C RAY WITH SPECIFIED SURFACES.
C
C INPUT:
C     ISTOR...THE SEQUENTIAL NUMBER IN THE INPUT DATA OF THE SPECIFIED
C             SURFACE.
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C             QUANTITIES Y(28) TO Y(IY(1)) REPRESENTING REDUCED
C             AMPLITUDES ARE IGNORED.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
```

```
C     NAMPL...NUMBER OF REAL QUANTITIES REPRESENTING COMPLEX-VALUED
C             VECTORIAL REDUCED AMPLITUDES.  IF NO CONVERSION
C             COEFFICIENTS ARE APPLIED NAMPL=IY(1)-27,  OTHERWISE
C             NAMPL=6 OR 12 (SEE C.R.T.5.5.4).
C     YC...   ARRAY CONTAINING REAL QUANTITIES REPRESENTING COMPLEX-
C             -VALUED VECTORIAL REDUCED AMPLITUDES.  IF NO CONVERSION
C             COEFFICIENTS ARE APPLIED, YC IS A COPY OF Y(28) TO
C             Y(IY(1)).  OTHERWISE,  YC REPRESENTS THE VECTORIAL REDUCED
C             AMPLITUDES INVOLVING APPROPRIATE CONVERSION COEFFICIENTS,
C             EXPRESSED IN RAY-CENTRED COORDINATE SYSTEM (SEE
C             C.R.T.5.5.4):
C             P WAVE AT THE INITIAL POINT OF THE RAY:
C               NAMPL=6,
C               YC(1)=REAL(A13),  YC(2)=AIMAG(A13),
C               YC(3)=REAL(A23),  YC(4)=AIMAG(A23),
C               YC(5)=REAL(A33),  YC(6)=AIMAG(A33).
C             S WAVE AT THE INITIAL POINT OF THE RAY:
C               NAMPL=12,
C               YC(1)=REAL(A11),  YC(2)=AIMAG(A11),
C               YC(3)=REAL(A21),  YC(4)=AIMAG(A21),
C               YC(5)=REAL(A31),  YC(6)=AIMAG(A31),
C               YC(7)=REAL(A12),  YC(8)=AIMAG(A12),
C               YC(9)=REAL(A22),  YC(10)=AIMAG(A22),
C               YC(11)=REAL(A32), YC(12)=AIMAG(A32).
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C ARRAY JPOINT IS MODIFIED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL NELEM
      INTEGER NELEM
C     FUNCTION NELEM(KODIND)... PACKAGE 'CODE'.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER NY
C
      IF(NELEM(IY(2)).GT.KWRIT2) THEN
        NY=27+NAMPL
        WRITE(LUW(2*(NF1+ISTOR)-1)) JWAVE,JRAY,NY,IY(5),IY(6),YL,Y,YC
        JPOINT(2*(NF1+ISTOR)-1)=JPOINT(2*(NF1+ISTOR)-1)+1
      END IF
      RETURN
      END
```

```
C
C=======================================================================
C
      SUBROUTINE WRIT33(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SUBROUTINE STORES THE COMPUTED QUANTITIES AT THE ENDPOINTS OF THE
C INDIVIDUAL ELEMENTARY WAVES (SEE C.R.T.5.5.3).  IT IS CALLED AT THE
C ENDPOINTS OF THE ELEMENTS OF RAYS, SITUATED AT STRUCTURAL INTERFACES.
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL NELEM
      INTEGER NELEM
C     FUNCTION NELEM(KODIND)... PACKAGE 'CODE'.
C
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATION:
      INTEGER I
C
      I=NELEM(IY(2))-KWRIT3
      IF(1.LE.I.AND.I.LE.NF3) THEN
        WRITE(LUW(2*(NF1+NF2+I)-1)) JRAY,IY,YL,(Y(I),I=1,IY(1)),YY
        JPOINT(2*(NF1+NF2+I)-1)=JPOINT(2*(NF1+NF2+I)-1)+1
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE WRIT4(LULOG,IRAY,YL,Y,YY,IY,IEND)
C
      INTEGER LULOG,IRAY,IY(12),IEND
```

```
      REAL YL(6),Y(35),YY(5)
C
C THIS SUBROUTINE STORES THE QUANTITIES AT THE INITIAL POINT OF THE RAY
C (SEE C.R.T.6.1).  IT IS CALLED AFTER TERMINATION OF TRACING THE RAY.
C
C INPUT:
C     LULOG...LOGICAL UNIT NUMBER OF THE LOG OUTPUT DEVICE.
C     IRAY... THE INDEX OF THE RAY WHICH HAS BEEN COMPUTED (I.E. THE
C             OUTPUT OF THE SUBROUTINE RPAR2 FROM THE PACKAGE 'RPAR').
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C     IEND... REASON OF THE TERMINATION OF THE COMPUTATION OF A RAY (SEE
C             C.R.T.5.4).  FOR A DETAILED DESCRIPTION SEE SUBROUTINES
C             RAY2 (PACKAGE 'RAY') AND INIT2 (PACKAGE 'INIT').
C
C NO OUTPUT.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /INITC/ (SEE SUBROUTINE PACKAGE 'INIT'):
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
C NONE OF THE STORAGE LOCATIONS ARE ALTERED.
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C JUEB, JFCT, JOUTP, JTRANS, ARRAYS JPOINT AND NENDR ARE MODIFIED IN
C THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SCRO4,CHECK
C     SUBROUTINE SCRO4(IRAY,YL,Y,YY,IY,IEND)... SUBROUTINE PACKAGE
C             'SCRO'.
C     SUBROUTINE CHECK(Q,QUEB,IRATE,I)... THIS PACKAGE.
C
C DATE: 1990, JANUARY 30
```

```
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
      REAL S12,S13,S23,S14,S24,S34
C
C     WRITING THE QUANTITIES AT THE INITIAL POINT OF THE RAY (C.R.T.6.1)
      I1=-JWAVE
      DO 10 I=1,NF1+NF2+NF3
        WRITE(LUW(2*I)) I1,IRAY,ICB1I,YLI,YI
        JPOINT(2*I)=JPOINT(2*I)+1
   10 CONTINUE
C
C     STATISTICS
      JFCT=JFCT+IY(9)
      JOUTP=JOUTP+IY(10)
      JTRANS=JTRANS+IY(11)
      DO 21 I=1,MENDR
        IF(IEND.EQ.JENDR(I)) THEN
          NENDR(I)=NENDR(I)+1
          GO TO 22
        END IF
   21 CONTINUE
      NENDR(MENDR+1)=NENDR(MENDR+1)+1
   22 CONTINUE
      NENDR(MENDR+2)=NENDR(MENDR+2)+1
C
C     WRITING TO THE OUTPUT LOG FILE
C     SIMPLECTICITY OF THE PROPAGATOR MATRIX (SEE C.R.T.5.6L)
      S12=Y(12)*Y(18)+Y(13)*Y(19)-Y(14)*Y(16)-Y(15)*Y(17)
      S13=Y(12)*Y(22)+Y(13)*Y(23)-Y(14)*Y(20)-Y(15)*Y(21)-1.
      S23=Y(16)*Y(22)+Y(17)*Y(23)-Y(18)*Y(20)-Y(19)*Y(21)
      S14=Y(12)*Y(26)+Y(13)*Y(27)-Y(14)*Y(24)-Y(15)*Y(25)
      S24=Y(16)*Y(26)+Y(17)*Y(27)-Y(18)*Y(24)-Y(19)*Y(25)-1.
      S34=Y(20)*Y(26)+Y(21)*Y(27)-Y(22)*Y(24)-Y(23)*Y(25)
C     CHECKING EXCEEDING THE SPECIFIED LIMITS
      I=0
      CALL CHECK(YY(2),UEB,I0,I)
      CALL CHECK(YY(3),UEBPP,I1,I)
      CALL CHECK(YY(4),UEBPH,I2,I)
      CALL CHECK(YY(5),UEBHH,I3,I)
      CALL CHECK(S12,UEBDRT,I4,I)
      CALL CHECK(S13,UEBDRT,I5,I)
      CALL CHECK(S23,UEBDRT,I6,I)
      CALL CHECK(S14,UEBDRT,I7,I)
      CALL CHECK(S24,UEBDRT,I8,I)
      CALL CHECK(S34,UEBDRT,I9,I)
C     I IS THE NUMBER OF CHECKED QUANTITIES EXCEEDING THEIR SPECIFIED
C     UPPER LIMITS
      IF(I.GT.0) THEN
C       WRITING REPORT ON THIS RAY
        IF(JUEB.EQ.0) THEN
          WRITE(LULOG,31)
   31     FORMAT(7X,'RAY:',5X,
     *      'CHECKED QUANTITIES IN PERCENTS OF THEIR SPECIFIED LIMITS')
        END IF
        JUEB=JUEB+1
        WRITE(LULOG,32) IRAY,I0,I1,I2,I3,I4,I5,I6,I7,I8,I9
```

```
   32   FORMAT(  I10,': ',10I6)
      END IF
C
C     SCREEN OUTPUT
      CALL SCRO4(IRAY,YL,Y,YY,IY,IEND)
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CHECK(Q,QUEB,IRATE,I)
      REAL Q,QUEB
C
C     AUXILIARY SUBROUTINE TO WRIT4
C
      IF(QUEB.GT.0.) THEN
        IRATE=INT(100.*Q/QUEB+0.5)
      ELSE
        IRATE=0
      END IF
      IF(ABS(Q).GT.QUEB) THEN
        I=I+1
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE WRIT5(LULOG,IWAVE)
      INTEGER LULOG,IWAVE
C
C THIS SUBROUTINE IS CALLED AFTER TERMINATION OF THE COMPUTATION OF AN
C ELEMENTARY WAVE, AND WHEN TERMINATING THE COMPLETE RAY TRACING
C PROGRAM.
C
C INPUT:
C     LULOG...LOGICAL UNIT NUMBER OF THE LOG OUTPUT DEVICE.
C     IWAVE...ZERO WHEN TERMINATING THE COMPLETE RAY TRACING PROGRAM,
C             OTHERWISE THE INDEX OF THE ELEMENTARY WAVE WHICH HAS BEEN
C             COMPUTED (I.E. THE OUTPUT OF THE SUBROUTINE CODE1 FROM THE
C             PACKAGE 'CODE').
C
C NO OUTPUT.
C
C COMMON BLOCKS /WRITC/ AND /WRITN/:
      INTEGER JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS
      INTEGER KWRIT1,KWRIT2,KWRIT3,NF1,NF2,NF3,MF3,NUW,MENDR
      PARAMETER (NUW=16)
      PARAMETER (MENDR=17)
      INTEGER LUW(NUW),JPOINT(NUW),JENDR(MENDR),NENDR(MENDR+2)
      COMMON/WRITC/JWAVE,JRAY,JUEB,JFCT,JOUTP,JTRANS,KWRIT1,KWRIT2,
     *             KWRIT3,NF1,NF2,NF3,MF3,LUW,JPOINT,JENDR,NENDR
      CHARACTER*16 NAME2(NUW)
      COMMON/WRITN/NAME2
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SCRO5
C     SUBROUTINE SCRO5(IWAVE)... SUBROUTINE PACKAGE 'SCRO'.
C
```

```
C DATE: 1990, JANUARY 30
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,J,N,K(4)
C
C     WRITING TO THE OUTPUT LOG FILE
      IF(IWAVE.NE.0) THEN
        IF(JUEB.GT.0) THEN
          WRITE(LULOG,10)
   10     FORMAT(11X,'ABOVE RAYS ARE COMPUTED WITH DECREASED ACCURACY')
        END IF
        WRITE(LULOG,21) NENDR(MENDR+2),JFCT,JOUTP,JTRANS
        N=0
        DO 12 J=1,MENDR
          IF(NENDR(J).GT.0) THEN
            N=N+1
            K(N)=J
          END IF
          IF(N.EQ.4.OR.(J.EQ.MENDR.AND.N.GT.0)) THEN
            WRITE(LULOG,22) (NENDR(K(I)),JENDR(K(I)),I=1,N)
            N=0
          END IF
   12   CONTINUE
        IF(NENDR(MENDR+1).NE.0) THEN
          WRITE(LULOG,23) NENDR(MENDR+1)
        END IF
        DO 14 I=1,2*(NF1+NF2+NF3)
          WRITE(LULOG,24) JPOINT(I),NAME2(I)
   14   CONTINUE
   21   FORMAT(  I10,' RAYS ',I12,' FCT  ',I12,' STEPS',I12,' TRANS')
   22   FORMAT(4(I10,' IEND=',I2))
   23   FORMAT(  I10,' IEND=OTHER')
   24   FORMAT(  I10,' POINTS IN FILE: ',A)
      END IF
C
C     SCREEN OUTPUT
      CALL SCRO5(IWAVE)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE FNAME(NUM1,NAME1,NAME2)
      INTEGER NUM1
      CHARACTER*(*) NAME1(NUM1)
      CHARACTER*(*) NAME2
C
C THIS SUBROUTINE IS DESIGNED TO ANALYSE THE SPECIFIED STRINGS AND TO
C COMPOSE THE FILENAME OF THEM.
C
C INPUT:
C     NAME1...CHARACTER ARRAY OF CHARACTER STRINGS TO BE ANALYSED.
C     NUM1... NUMBER OF FILENAMES TO BE ANALYSED.
C
C OUTPUT:
C     NAME2...FILENAME COMPOSED OF THE ANALYSED INPUT COMPONENTS
C             (STRINGS).
```

```
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C ERROR MESSAGES:
C     559...  TOO MANY STRINGS:
C             THE NUMBER NUM1 OF INPUT STRINGS EXCEEDS THE DIMENSION M1
C             OF ARRAYS J1 AND L1.  PARAMETER M1 MUST BE INCREASED.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER M1
      PARAMETER (M1=3)
      INTEGER J1(M1),L1(M1),I1,J2,L2,I2,N,I
C
C     M1...   MAXIMUM NUMBER OF ANALYSED STRINGS.
C     J1(N)...NUMBER OF THE CURRENTLY ANALYSED CHARACTERS OF NAME1(N),
C             I.E. THE POSITION OF THE SPACE AFTER THE LAST ANALYSED
C             WORD OF NAME1(N).  A WORD IS A SUBSTRING BETWEEN TWO
C             CONSECUTIVE SPACES.
C     L1(N)...LENGTH OF NAME1(N).
C     I1...   BEGINNING OF THE LAST ANALYSED WORD OF NAME1(N), I.E. THE
C             PREVIOUS VALUE OF J1(N) INCREASED BY 1.
C     J2...   NUMBER OF THE CURRENTLY DEFINED CHARACTERS OF NAME2.
C     L2...   LENGTH OF NAME2.
C     I2...   AUXILIARY VARIABLE - PREVIOUS VALUE OF J2 INCREASED BY 1.
C     N...    LOOP VARIABLE - INDEX OF INPUT FILENAME.
C     I...    LOOP VARIABLE - POSITION IN A STRING.
C
C.......................................................................
C
      IF(M1.LT.NUM1) THEN
        STOP 'ERROR 559 IN FNAME: TOO MANY STRINGS'
      END IF
      DO 11 N=1,NUM1
        J1(N)=0
        L1(N)=LEN(NAME1(N))
   11 CONTINUE
      J2=0
      L2=LEN(NAME2)
C
   20 CONTINUE
      DO 29 N=1,NUM1
C       (A) ANALYSING NAME1(N):
        I1=J1(N)+1
        DO 21 I=I1,L1(N)
          IF(NAME1(N)(I:I).EQ.' ') THEN
            J1(N)=I
            GO TO 22
          END IF
   21   CONTINUE
        J1(N)=L1(N)+1
   22   CONTINUE
C       J1(N)-TH CHARACTER OF NAME1(N) IS FOUND TO BE BLANK.
C       (B) COPYING THE WORD FROM NAME1(N) TO NAME2:
        IF(J1(N).GT.I1) THEN
          I2=J2+1
```

```
          J2=MIN0(J2+J1(N)-I1,L2)
          NAME2(I2:J2)=NAME1(N)(I1:J1(N)-1)
        END IF
   29 CONTINUE
C
      DO 31 N=1,NUM1
        IF(J1(N).LT.L1(N)) GO TO 20
   31 CONTINUE
      DO 32 I=J2+1,L2
        NAME2(I:I)=' '
   32 CONTINUE
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'SCRONUL' - DUMMY SCREEN OUTPUT SUBROUTINES
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF THE FOLLOWING EXTERNAL PROCEDURES:
C     SCRO1...SCREEN OUTPUT SUBROUTINE CALLED WHEN STARTING THE COMPLETE
C             RAY TRACING PROGRAM, AND WHEN STARTING THE COMPUTATION OF
C             A NEW ELEMENTARY WAVE.
C     SCRO2...SCREEN OUTPUT SUBROUTINE CALLED WHEN STARTING THE COMPLETE
C             TRACING OF A NEW RAY.
C     SCRO3...SCREEN OUTPUT SUBROUTINE CALLED WITH CONSTANT STEP STORE
C             OF THE INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE
C             POINTS OF INTERSECTION WITH INTERFACES EITHER BEFORE AND
C             AFTER THE TRANSFORMATION.
C     SCRO4...SCREEN OUTPUT SUBROUTINE CALLED AFTER TERMINATION OF
C             TRACING THE RAY.
C     SCRO5...SCREEN OUTPUT SUBROUTINE CALLED AFTER TERMINATION OF THE
C             COMPUTATION OF AN ELEMENTARY WAVE, AND WHEN TERMINATING
C             THE COMPLETE RAY TRACING PROGRAM.
C
C=======================================================================
C
      SUBROUTINE SCRO1(IWAVE)
      INTEGER IWAVE
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED WHEN STARTING THE COMPLETE RAY
C TRACING PROGRAM, AND WHEN STARTING THE COMPUTATION OF A NEW ELEMENTARY
C WAVE.
C
C INPUT:
C     IWAVE...ZERO WHEN STARTING THE COMPLETE RAY TRACING PROGRAM,
C             OTHERWISE THE INDEX OF THE ELEMENTARY WAVE WHICH WILL BE
C             COMPUTED (I.E. THE OUTPUT OF THE SUBROUTINE CODE1 FROM THE
C             PACKAGE 'CODE').
C
C NO OUTPUT.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO2(IRAY)
      INTEGER IRAY
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED WHEN STARTING THE COMPLETE
C TRACING OF A NEW RAY.
C
C INPUT:
C     IRAY... THE INDEX OF THE RAY WHICH WILL BE COMPUTED (I.E. THE
C             OUTPUT OF THE SUBROUTINE RPAR2 FROM THE PACKAGE 'RPAR').
C
C NO OUTPUT.
C
C DATE: 1989, DECEMBER 4
```

```
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO3(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED WITH CONSTANT STEP STORE OF
C THE INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE POINTS OF
C INTERSECTION WITH INTERFACES EITHER BEFORE AND AFTER THE
C TRANSFORMATION.  IT PLOTS THE PART OF THE RAY COMPUTED IN THE LAST
C STEP OF THE NUMERICAL INTEGRATION.  IT IS CALLED BY THE SUBROUTINE
C WRIT31.
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C DATE: 1989, DECEMBER 4
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO4(IRAY,YL,Y,YY,IY,IEND)
C
      INTEGER IRAY,IY(12),IEND
      REAL YL(6),Y(35),YY(5)
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED AFTER TERMINATION OF TRACING
C THE RAY.
C
C INPUT:
C     IRAY... THE INDEX OF THE RAY WHICH HAS BEEN COMPUTED (I.E. THE
C             OUTPUT OF THE SUBROUTINE RPAR2 FROM THE PACKAGE 'RPAR').
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C     IEND... REASON OF THE TERMINATION OF THE COMPUTATION OF A RAY (SEE
C             C.R.T.5.4).  FOR A DETAILED DESCRIPTION SEE SUBROUTINE RAY
```

```
C               (SUBROUTINE PACKAGE 'RAY').
C
C NO OUTPUT.
C
C DATE: 1989, DECEMBER 4
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCR05(IWAVE)
      INTEGER IWAVE
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED AFTER TERMINATION OF THE
C COMPUTATION OF AN ELEMENTARY WAVE, AND WHEN TERMINATING THE COMPLETE
C RAY TRACING PROGRAM.
C
C INPUT:
C     IWAVE...ZERO WHEN TERMINATING THE COMPLETE RAY TRACING PROGRAM,
C             OTHERWISE THE INDEX OF THE ELEMENTARY WAVE WHICH HAS BEEN
C             COMPUTED (I.E. THE OUTPUT OF THE SUBROUTINE CODE1 FROM THE
C             PACKAGE 'CODE').
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
C     SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)... CALCOMP GRAPHICS SUBROUTINE.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'SCROPC' - SCREEN OUTPUT SUBROUTINES FOR PC'S
C
C BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK
C
C THIS PACKAGE CONSISTS OF:
C     SCROB...BLOCK DATA SUBROUTINE DEFINING AUXILIARY COMMON BLOCK
C             /SCROC/ TO CONFIGURE THE SCREEN OUTPUT.
C     SCRO1...SCREEN OUTPUT SUBROUTINE CALLED WHEN STARTING THE COMPLETE
C             RAY TRACING PROGRAM, AND WHEN STARTING THE COMPUTATION OF
C             A NEW ELEMENTARY WAVE.
C     SCRO2...SCREEN OUTPUT SUBROUTINE CALLED WHEN STARTING THE COMPLETE
C             TRACING OF A NEW RAY.
C     SCRO3...SCREEN OUTPUT SUBROUTINE CALLED WITH CONSTANT STEP STORE
C             OF THE INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE
C             POINTS OF INTERSECTION WITH INTERFACES EITHER BEFORE AND
C             AFTER THE TRANSFORMATION.
C     SCRO4...SCREEN OUTPUT SUBROUTINE CALLED AFTER TERMINATION OF
C             TRACING THE RAY.
C     SCRO5...SCREEN OUTPUT SUBROUTINE CALLED AFTER TERMINATION OF THE
C             COMPUTATION OF AN ELEMENTARY WAVE, AND WHEN TERMINATING
C             THE COMPLETE RAY TRACING PROGRAM.
C     CURSOR..CHARACTER FUNCTION THAT RETURNS THE ANSI ESCAPE SEQUENCE
C             POSITIONING THE CURSOR AT THE BEGINNING OF THE GIVEN LINE.
C     SPECIFICATION OF THE USED CALCOMP GRAPHICS SUBROUTINES.
C
C CONFIGURATION OF THE SCREEN OUTPUT:
C     THE QUANTITIES SPECIFYING THE SCALE AND OTHER PROPERTIES OF THE
C     SIMPLE GRAPHIC OUTPUT ARE STORED IN THE COMMON BLOCK /SCROC/
C     DEFINED IN THE FOLLOWING SUBROUTINE:
C     -----------------------------------------------------------------
      BLOCK DATA SCROB
      INTEGER MKOLOR
      PARAMETER (MKOLOR=16)
      INTEGER NKOLOR,KOLOR(MKOLOR)
      REAL HMIN,HMAX,VMIN,VMAX,WIDTH
      REAL H1A,H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
      COMMON/SCROC/NKOLOR,KOLOR,HMIN,HMAX,VMIN,VMAX,WIDTH,H1A,
     *             H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
      SAVE /SCROC/
C     DIMENSIONS OF THE PLOT AREA:
      DATA HMIN,HMAX,VMIN,VMAX,WIDTH/7.45,29.66,0.00,21.00,0.045/
C     COLOURS:
      DATA NKOLOR/6/,KOLOR/2,3,5,6,4,7,14,1,1,1,1,1,1,1,1,1/
      END
C     -----------------------------------------------------------------
C     MKOLOR..DIMENSION OF THE ARRAY KOLOR.
C     NKOLOR..NUMBER OF DIFFERENT COLOURS USED TO PLOT DIFFERENT
C             ELEMENTS OF A RAY.
C     KOLOR...ARRAY CONTAINING THE INDICES OF COLOURS USED TO PLOT RAYS.
C             KOLOR(I)... THE COLOR OF EACH I-TH ELEMENT OF A RAY.  IF
C               THE NUMBER OF RAY ELEMENTS IS GREATER THAN NKOLOR, THE
C               COLORS KOLOR(1) TO KOLOR(NKOLOR) ARE CYCLICLY REPEATED.
C             KOLOR(NKOLOR+1)... INDEX OF THE COLOUR TO PLOT THE FRAMES
C               AROUND THE THREE PROJECTIONS OF THE MODEL BOX.
C     HMIN, HMAX... HORIZONTAL COORDINATES OF THE VERTICAL BOUNDARIES OF
C             THE PLOTTING AREA IN THE PLOT COORDINATES.
C     VMIN, VMAX... VERTICAL COORDINATES OF THE HORIZONTAL BOUNDARIES OF
C             THE PLOTTING AREA IN THE PLOT COORDINATES.
C     WIDTH...THICKNESS OF THE PLOTTED LINE.
C     H1A,H1B,H2A,H2B,V2A,V2B,V3A,V3B... AUXILIARY VARIABLES:
```

```
C               COEFFICIENTS OF THE LINEAR FUNCTIONS PROJECTING GENERAL
C               COORDINATES OF A POINT ON A RAY TO THE PLOT COORDINATES.
C       H1OLD,H2OLD,V2OLD,V3OLD... AUXILIARY VARIABLES: POSITIONS OF THE
C               PROJECTIONS OF THE LAST POINT OF A RAY TO THE SCREEN IN
C               PLOT COORDINATES.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C=======================================================================
C
      SUBROUTINE SCRO1(IWAVE)
      INTEGER IWAVE
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED WHEN STARTING THE COMPLETE RAY
C TRACING PROGRAM, AND WHEN STARTING THE COMPUTATION OF A NEW ELEMENTARY
C WAVE.
C
C INPUT:
C     IWAVE...ZERO WHEN STARTING THE COMPLETE RAY TRACING PROGRAM,
C             OTHERWISE THE INDEX OF THE ELEMENTARY WAVE WHICH WILL BE
C             COMPUTED (I.E. THE OUTPUT OF THE SUBROUTINE CODE1 FROM THE
C             PACKAGE 'CODE').
C
C NO OUTPUT.
C
C COMMON BLOCK /DCRT/ (SEE SUBROUTINE PACKAGE 'RAY'):
      INTEGER MEND,MSTOR
      PARAMETER (MEND=128)
      PARAMETER (MSTOR=128)
      INTEGER KSTORE,NEXPS,NHLF,MODCRT
      REAL STORE,STEP,UEB,UEBPP,UEBPH,UEBHH,UEBDRT,BOUNDR(7)
      INTEGER NSRFCA,NEND,KEND(MEND),NSTOR,KSTOR(MSTOR)
      COMMON/DCRT/ KSTORE,NEXPS,NHLF,MODCRT,STORE,STEP,UEB,UEBPP,UEBPH,
     *             UEBHH,UEBDRT,BOUNDR,NSRFCA,NEND,KEND,NSTOR,KSTOR
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /SCROC/:
      INTEGER MKOLOR
      PARAMETER (MKOLOR=16)
      INTEGER NKOLOR,KOLOR(MKOLOR)
      REAL HMIN,HMAX,VMIN,VMAX,WIDTH
      REAL H1A,H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
      COMMON/SCROC/NKOLOR,KOLOR,HMIN,HMAX,VMIN,VMAX,WIDTH,H1A,
     *             H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
C STORAGE LOCATIONS H1B,H2A,H2B,V2A,V2B,V3A,V3B OF THE COMMON BLOCK ARE
C DEFINED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL CURSOR,PLOTS,PLOT,NEWPEN
      CHARACTER*8 CURSOR
C     FUNCTION CURSOR(LINE)... THIS PACKAGE.
C     SUBROUTINE PLOTS(0,0,0)... CALCOMP GRAPHICS SUBROUTINE.
C     SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)... CALCOMP GRAPHICS SUBROUTINE.
C     SUBROUTINE NEWPEN(INP)... CALCOMP GRAPHICS SUBROUTINE.
C
C ERROR MESSAGES:
C     560...  TOO MANY COLOURS:
C             THE DIMENSION MKOLOR OF THE ARRAY KOLOR MUST BE GREATER
C             THAN THE NUMBER NKOLOR OF COLOURS USED TO PLOT THE RAY
```

```
C               ELEMENTS, SEE THE DATA BLOCK SUBROUTINE SCROB.
C
C DATE: 1990, FEBRUARY 5
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      REAL STREET,H1,H2,V1,V2
C
      IF(IWAVE.GT.0) THEN
C
C       ERASING SCREEN:
        WRITE(*,'(A)') CURSOR(0)
C
C       DIMENSIONS OF THE PLOT AREA:
        STREET=2.*WIDTH
C
C       COEFFICIENTS OF THE LINEAR PROJECTIONS ONTO THE SCREEN:
        H1A=((HMAX-HMIN-STREET)/2.-3.*WIDTH)/(BOUNDR(2)-BOUNDR(1))
        H1B=HMIN+1.5*WIDTH-BOUNDR(1)*H1A
        H2A=((HMAX-HMIN-STREET)/2.-3.*WIDTH)/(BOUNDR(4)-BOUNDR(3))
        H2B=HMAX-1.5*WIDTH-BOUNDR(4)*H2A
        V2A=((VMAX-VMIN-STREET)/2.-3.*WIDTH)/(BOUNDR(4)-BOUNDR(3))
        V2B=VMAX-1.5*WIDTH-BOUNDR(4)*V2A
        V3A=((VMAX-VMIN-STREET)/2.-3.*WIDTH)/(BOUNDR(6)-BOUNDR(5))
        V3B=VMIN+1.5*WIDTH-BOUNDR(5)*V3A
C
C       PLOT INITIALIZATION:
        CALL PLOTS(0,0,0)
        IF(NKOLOR.GE.MKOLOR) THEN
          STOP 'ERROR 560 IN SCRO1: TOO MANY COLOURS'
        END IF
        CALL NEWPEN(KOLOR(NKOLOR+1))
C
C       PLOTTING FRAMES:
        H1=HMIN+WIDTH/2.
        H2=(HMIN+HMAX-STREET-WIDTH)/2.
        V1=(VMIN+VMAX+STREET+WIDTH)/2.
        V2=VMAX-WIDTH/2.
        CALL PLOT(H1,V1,3)
        CALL PLOT(H2,V1,2)
        CALL PLOT(H2,V2,2)
        CALL PLOT(H1,V2,2)
        CALL PLOT(H1,V1,2)
        V1=VMIN+WIDTH/2.
        V2=(VMIN+VMAX-STREET-WIDTH)/2.
        CALL PLOT(H1,V1,3)
        CALL PLOT(H2,V1,2)
        CALL PLOT(H2,V2,2)
        CALL PLOT(H1,V2,2)
        CALL PLOT(H1,V1,2)
        H1=(HMIN+HMAX+STREET+WIDTH)/2.
        H2=HMAX-WIDTH/2.
        CALL PLOT(H1,V1,3)
        CALL PLOT(H2,V1,2)
        CALL PLOT(H2,V2,2)
        CALL PLOT(H1,V2,2)
        CALL PLOT(H1,V1,2)
C
```

```
C        WRITING TO THE SCREEN:
         WRITE(*,'(A,A,I8)') CURSOR(3),'WAVE:',IWAVE
         WRITE(*,'(A,A)')    CURSOR(2),'                              '
         WRITE(*,'(A,A)')    CURSOR(1),'NEW ELEMENTARY WAVE '
C
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO2(IRAY)
      INTEGER IRAY
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED WHEN STARTING THE COMPLETE
C TRACING OF A NEW RAY.
C
C INPUT:
C     IRAY... THE INDEX OF THE RAY WHICH WILL BE COMPUTED (I.E. THE
C             OUTPUT OF THE SUBROUTINE RPAR2 FROM THE PACKAGE 'RPAR').
C
C NO OUTPUT.
C
C COMMON BLOCK /INITC/ (SEE SUBROUTINE PACKAGE 'INIT'):
      INTEGER MSRFCA
      PARAMETER (MSRFCA=128)
      INTEGER ISB1I,ICB1I
      REAL YLI(6),YI(25),FSRFCA(MSRFCA)
      COMMON/INITC/ISB1I,ICB1I,YLI,YI,FSRFCA
C NONE OF THE STORAGE LOCATIONS OF THE COMMON BLOCK ARE ALTERED.
C
C COMMON BLOCK /SCROC/:
      INTEGER MKOLOR
      PARAMETER (MKOLOR=16)
      INTEGER NKOLOR,KOLOR(MKOLOR)
      REAL HMIN,HMAX,VMIN,VMAX,WIDTH
      REAL H1A,H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
      COMMON/SCROC/NKOLOR,KOLOR,HMIN,HMAX,VMIN,VMAX,WIDTH,H1A,
     *             H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
C STORAGE LOCATIONS H1OLD,H2OLD,V2OLD,V3OLD OF THE COMMON BLOCK ARE
C DEFINED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL CURSOR
      CHARACTER*8 CURSOR
C     FUNCTION CURSOR(LINE)... THIS PACKAGE.
C
C DATE: 1990, FEBRUARY 5
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     NO AUXILIARY STORAGE LOCATIONS.
C
C     WRITING TO THE SCREEN:
      WRITE(*,'(A,A)')       CURSOR(1),'COMPUTING                     '
      WRITE(*,'(A,A,I8)')    CURSOR(4),'RAY: ',IRAY
      WRITE(*,'(A,A,F15.6)') CURSOR(5),'PAR1:',YI(20)
      WRITE(*,'(A,A,F15.6)') CURSOR(6),'PAR2:',YI(21)
      WRITE(*,'(A,A)')       CURSOR(7),'                              '
```

```
      WRITE(*,'(A,A)')        CURSOR(8),'        ISB ICB ISRF'
C
C     INITIAL POSITION FOR PLOTTING THE RAY:
      H1OLD=YI(3)*H1A+H1B
      H2OLD=YI(4)*H2A+H2B
      V2OLD=YI(4)*V2A+V2B
      V3OLD=YI(5)*V3A+V3B
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO3(YL,Y,YY,IY)
      REAL YL(6),Y(35),YY(5)
      INTEGER IY(12)
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED WITH CONSTANT STEP STORE OF
C THE INDEPENDENT VARIABLE ALONG THE RAY, AND AT THE POINTS OF
C INTERSECTION WITH INTERFACES EITHER BEFORE AND AFTER THE
C TRANSFORMATION.  IT PLOTS THE PART OF THE RAY COMPUTED IN THE LAST
C STEP OF THE NUMERICAL INTEGRATION.  IT IS CALLED BY THE SUBROUTINE
C WRIT31.
C
C INPUT:
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C NO OUTPUT.
C
C COMMON BLOCK /SCROC/:
      INTEGER MKOLOR
      PARAMETER (MKOLOR=16)
      INTEGER NKOLOR,KOLOR(MKOLOR)
      REAL HMIN,HMAX,VMIN,VMAX,WIDTH
      REAL H1A,H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
      COMMON/SCROC/NKOLOR,KOLOR,HMIN,HMAX,VMIN,VMAX,WIDTH,H1A,
     *             H1B,H2A,H2B,V2A,V2B,V3A,V3B,H1OLD,H2OLD,V2OLD,V3OLD
C STORAGE LOCATIONS H1OLD,H2OLD,V2OLD,V3OLD OF THE COMMON BLOCK ARE
C REDEFINED IN THIS SUBROUTINE.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL NELEM,PLOT,NEWPEN
      INTEGER NELEM
C     FUNCTION NELEM(KODIND)... SUBROUTINE PACKAGE 'CODE'.
C     SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)... CALCOMP GRAPHICS SUBROUTINE.
C     SUBROUTINE NEWPEN(INP)... CALCOMP GRAPHICS SUBROUTINE.
C
C DATE: 1990, JANUARY 18
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      REAL H1NEW,H2NEW,V2NEW,V3NEW
C
```

```
C     WRITING TO THE SCREEN:
      IF(IY(6).NE.0) THEN
        IF(IY(8).NE.0) THEN
          WRITE(*,'('' .......'',3I4)') IY(4),IY(5),IY(6)
        ELSE
          WRITE(*,'(''        '',3I4)') IY(4),IY(5),IY(6)
        END IF
      END IF
C
C     PLOTTING THE RAY AT THE SCREEN:
      H1NEW=Y(3)*H1A+H1B
      H2NEW=Y(4)*H2A+H2B
      V2NEW=Y(4)*V2A+V2B
      V3NEW=Y(5)*V3A+V3B
      CALL NEWPEN(KOLOR(MOD(NELEM(IY(2))-1,NKOLOR)+1))
      CALL PLOT(H1OLD,V2OLD,3)
      CALL PLOT(H1NEW,V2NEW,2)
      CALL PLOT(H1OLD,V3OLD,3)
      CALL PLOT(H1NEW,V3NEW,2)
      CALL PLOT(H2OLD,V3OLD,3)
      CALL PLOT(H2NEW,V3NEW,2)
      H1OLD=H1NEW
      H2OLD=H2NEW
      V2OLD=V2NEW
      V3OLD=V3NEW
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO4(IRAY,YL,Y,YY,IY,IEND)
C
      INTEGER IRAY,IY(12),IEND
      REAL YL(6),Y(35),YY(5)
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED AFTER TERMINATION OF TRACING
C THE RAY.
C
C INPUT:
C     IRAY... THE INDEX OF THE RAY WHICH HAS BEEN COMPUTED (I.E. THE
C             OUTPUT OF THE SUBROUTINE RPAR2 FROM THE PACKAGE 'RPAR').
C     YL...   ARRAY CONTAINING LOCAL QUANTITIES AT THE POINT OF THE RAY.
C     Y...    ARRAY CONTAINING BASIC QUANTITIES COMPUTED ALONG THE RAY.
C     YY...   ARRAY CONTAINING REAL AUXILIARY QUANTITIES COMPUTED ALONG
C             THE RAY.
C     IY...   ARRAY CONTAINING INTEGER AUXILIARY QUANTITIES COMPUTED
C             ALONG THE RAY.
C     IEND... REASON OF THE TERMINATION OF THE COMPUTATION OF A RAY (SEE
C             C.R.T.5.4).  FOR A DETAILED DESCRIPTION SEE SUBROUTINE RAY
C             (SUBROUTINE PACKAGE 'RAY').
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL CURSOR,NELEM
      CHARACTER*8 CURSOR
      INTEGER NELEM
C     FUNCTION CURSOR(LINE)... THIS PACKAGE.
C     NELEM(KODIND)... SUBROUTINE PACKAGE 'CODE'.
C
```

```
C DATE: 1990, FEBRUARY 5
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
      INTEGER I,NLINES
C
C     NUMBER OF LINES AVAILABLE ON THE SCREEN: NLINES:
C     NLINES MAY BE LOWER THAN THE ACTUAL NUMBER OF LINES BUT SHOULD NOT
C     EXCEED THAT.  NLINES=25 IS A GOOD CHOICE FOR IBM-PC'S, BUT
C     NLINES=24 FITS ALSO VAX COMPUTERS.
      PARAMETER (NLINES=24)
C
C     WRITING TO THE SCREEN:
      WRITE(*,'('' END:'',I3,3I4)') IEND,IY(4),IY(5),IY(6)
      DO 10 I=1,NLINES-9-2*NELEM(IY(2))
        WRITE(*,'(''                                          '')')
   10 CONTINUE
      WRITE(*,'(A,A)') CURSOR(1),'                                    '
C
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE SCRO5(IWAVE)
      INTEGER IWAVE
C
C THIS SCREEN OUTPUT SUBROUTINE IS CALLED AFTER TERMINATION OF THE
C COMPUTATION OF AN ELEMENTARY WAVE, AND WHEN TERMINATING THE COMPLETE
C RAY TRACING PROGRAM.
C
C INPUT:
C     IWAVE...ZERO WHEN TERMINATING THE COMPLETE RAY TRACING PROGRAM,
C             OTHERWISE THE INDEX OF THE ELEMENTARY WAVE WHICH HAS BEEN
C             COMPUTED (I.E. THE OUTPUT OF THE SUBROUTINE CODE1 FROM THE
C             PACKAGE 'CODE').
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL CURSOR,PLOT
      CHARACTER*8 CURSOR
C     FUNCTION CURSOR(LINE)... THIS PACKAGE.
C     SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)... CALCOMP GRAPHICS SUBROUTINE.
C
C DATE: 1990, JANUARY 22
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     NO AUXILIARY STORAGE LOCATIONS.
C
      IF(IWAVE.EQ.0) THEN
C       ERASING TEXT SCREEN:
        WRITE(*,'(A)') CURSOR(0)
      ELSE
C       CLOSING DOWN PLOTTING:
        CALL PLOT(0.0,0.0,999)
```

```
      END IF
      RETURN
      END
C
C=======================================================================
C
      CHARACTER*8 FUNCTION CURSOR(LINE)
      INTEGER LINE
C
C THIS FUNCTION RETURNS THE ANSI ESCAPE SEQUENCE POSITIONING THE CURSOR
C AT THE BEGINNING OF THE GIVEN LINE.
C
C INPUT:
C     LINE... INDEX OF THE GIVEN LINE, OR ZERO.  IF LINE=0, THE ENTIRE
C             SCREEN IS TO BE ERASED AND THE CURSOR RETURNED TO THE HOME
C             POSITION.
C
C OUTPUT:
C     CERSOR..ESCAPE SEQUENCE POSITIONING THE CURSOR AT THE BEGINNING OF
C             THE GIVEN LINE, OR ERASING THE SCREEN.
C
C NO SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED.
C
C DATE: 1990, JANUARY 18
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     NO AUXILIARY STORAGE LOCATIONS.
C
      IF(LINE.EQ.0) THEN
        CURSOR='    '//CHAR(27)//CHAR(91)//'2J'
      ELSE
        CURSOR=' '//CHAR(27)//CHAR(91)//
     *         CHAR(48+LINE/10)//CHAR(48+MOD(LINE,10))//CHAR(59)//'1H'
      END IF
      RETURN
      END
C
C=======================================================================
C
C SPECIFICATION OF THE USED CALCOMP GRAPHICS SUBROUTINES:
C     SUBROUTINE PLOTS(I1,I2,I3)... INITIALIZES PLOTTING.  IT IS CALLED
C             WHEN STARTING THE COMPUTATION OF THE NEW ELEMENTARY WAVE.
C             I1,I2,I3... SET TO ZEROS.
C     SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)... MOVES PEN OR PLOTS A LINE.
C             XPAGE, YPAGE... COORDINATES, OFTEN IN CENTIMETRES.
C             IPEN... CONTROLS THE PLOTTING:
C               IPEN=2... THE PEN IS DOWN DURING THE MOVEMENT, THUS
C                 DRAWING A LINE.
C               IPEN=3... THE PEN IS UP DURING THE MOVEMENT.
C               IPEN=999... TERMINATES PLOTTING INITIALIZED BY THE
C                 SUBROUTINE PLOTS.
C     SUBROUTINE NEWPEN(INP)... CHANGES THE COLOR.
C             INP... SPECIFIES THE NUMBER OF COLOR TO BE SELECTED.
C     ALL PARAMETERS ARE INPUT PARAMETERS AND SHOULD NOT BE MODIFIED.
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'PLOTNUL' - DUMMY PLOT SUBROUTINES
C
C=========================================================================
C
      SUBROUTINE PLOTS(I1,I2,I3)
      INTEGER I1,I2,I3
C
      RETURN
      END
C
C=========================================================================
C
      SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)
      REAL    XPAGE,YPAGE
      INTEGER IPEN
C
      RETURN
      END
C
C=========================================================================
C
      SUBROUTINE NEWPEN(INP)
      INTEGER INP
C
      RETURN
      END
C
C=========================================================================
C
```

```
C SUBROUTINE PACKAGE 'PLOTVDI' - PLOT INTERFACE SUBROUTINES TO GSS-VDI
C
C=====================================================================
C
      SUBROUTINE PLOTS(I1,I2,I3)
      INTEGER I1,I2,I3
C
C     ----------------------------------------------------------------
C     SPECIFICATIONS COMMON TO ALL THE ROUTINES OF THIS FILE:
      INTEGER*2 LUVDI
      REAL*4    STARTX,STARTY,SCALEX,SCALEY,OLDX,OLDY
      COMMON /VDICOM/ LUVDI,STARTX,STARTY,SCALEX,SCALEY,OLDX,OLDY
      INTEGER*2 VDIERR
C     ----------------------------------------------------------------
C
C     CALLED VDI FUNCTIONS
      EXTERNAL VOPNWK,VQERR
      INTEGER*2 VOPNWK,VQERR
C
C     AUXILIARY STORAGE LOCATIONS
      INTEGER*2 WORKIN(19),WORKOT(66)
      CHARACTER*8 DEVICE
C
      DEVICE= 'DISPLAY '
      HSIZE=29.7
      VSIZE=21.0
      SCALEX= 32768./HSIZE
      SCALEY= 32768./VSIZE
      STARTX= 0.0
      STARTY= 0.0
      WORKIN(1) = 0
      WORKIN(2) = 1
      WORKIN(3) = 1
      WORKIN(4) = 1
      WORKIN(5) = 1
      WORKIN(6) = 1
      WORKIN(7) = 1
      WORKIN(8) = 1
      WORKIN(9) = 1
      WORKIN(10)= 1
      WORKIN(11)= 1
      WORKIN(12)= ICHAR(DEVICE(1:1))
      WORKIN(13)= ICHAR(DEVICE(2:2))
      WORKIN(14)= ICHAR(DEVICE(3:3))
      WORKIN(15)= ICHAR(DEVICE(4:4))
      WORKIN(16)= ICHAR(DEVICE(5:5))
      WORKIN(17)= ICHAR(DEVICE(6:6))
      WORKIN(18)= ICHAR(DEVICE(7:7))
      WORKIN(19)= ICHAR(DEVICE(8:8))
      VDIERR= VOPNWK(WORKIN,LUVDI,WORKOT)
      IF(VDIERR.NE.0) THEN
        VDIERR= VQERR()
        WRITE(*,'('' VDI ERROR NO.'',I6)') VDIERR
        STOP
      END IF
      STARTX= SCALEX*STARTX
      STARTY= SCALEY*STARTY
      OLDX  = 0.
      OLDY  = 0.
      RETURN
```

```
      END
C
C==========================================================================
C
      SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)
      REAL    XPAGE,YPAGE
      INTEGER IPEN
C
C     ---------------------------------------------------------------------
C     SPECIFICATIONS COMMON TO ALL THE ROUTINES OF THIS FILE:
      INTEGER*2 LUVDI
      REAL*4    STARTX,STARTY,SCALEX,SCALEY,OLDX,OLDY
      COMMON /VDICOM/ LUVDI,STARTX,STARTY,SCALEX,SCALEY,OLDX,OLDY
      INTEGER*2 VDIERR
C     ---------------------------------------------------------------------
C
C     CALLED VDI FUNCTIONS
      EXTERNAL VPLINE,VCLRWK,VCLSWK
      INTEGER*2 VPLINE,VCLRWK,VCLSWK
C
C     AUXILIARY STORAGE LOCATIONS
      INTEGER*2 IXY(4)
      REAL      X,Y
      INTEGER*4 IX,IY
C
      X = XPAGE
      Y = YPAGE
C
      IF(IABS(IPEN).EQ.2) THEN
        IX     = INT(STARTX+SCALEX*OLDX+0.5)
        IY     = INT(STARTY+SCALEY*OLDY+0.5)
        IF(IX.GE.0.AND.IX.LE.32767.AND.
     *       IY.GE.0.AND.IY.LE.32767) THEN
          IXY(1)= IX
          IXY(2)= IY
          IX     = INT(STARTX+SCALEX*X+0.5)
          IY     = INT(STARTY+SCALEY*Y+0.5)
          IF(IX.GE.0.AND.IX.LE.32767.AND.
     *       IY.GE.0.AND.IY.LE.32767) THEN
            IXY(3)= IX
            IXY(4)= IY
            VDIERR= VPLINE(LUVDI,2,IXY)
          END IF
        END IF
      END IF
      IF(IPEN.GE.0) THEN
        OLDX   = X
        OLDY   = Y
      ELSE
        STARTX= STARTX+SCALEX*X
        STARTY= STARTY+SCALEY*Y
        OLDX   = 0.
        OLDY   = 0.
      END IF
      IF(IPEN.GE.999) THEN
        VDIERR= VCLRWK(LUVDI)
        VDIERR= VCLSWK(LUVDI)
        WRITE(*,'('' =3h'')')
      END IF
      RETURN
```

```
      END
C
C=====================================================================
C
      SUBROUTINE NEWPEN(INP)
      INTEGER INP
C
C     ---------------------------------------------------------------
C     SPECIFICATIONS COMMON TO ALL THE ROUTINES OF THIS FILE:
      INTEGER*2 LUVDI
      REAL*4     STARTX,STARTY,SCALEX,SCALEY,OLDX,OLDY
      COMMON /VDICOM/ LUVDI,STARTX,STARTY,SCALEX,SCALEY,OLDX,OLDY
      INTEGER*2 VDIERR
C     ---------------------------------------------------------------
C
C     CALLED VDI FUNCTIONS
      EXTERNAL VSLCOL
      INTEGER*2 VSLCOL
C
      VDIERR= VSLCOL(LUVDI,INP)
      RETURN
      END
C
C=====================================================================
C
```

```
C SUBROUTINE PACKAGE 'PLOTH' - PLOT INTERFACE SUBROUTINES TO HALO
C
C=======================================================================
C
      SUBROUTINE PLOTS(I1,I2,I3)
      INTEGER I1,I2,I3
C
C INPUT:
C     I1,I2,I3...ANY INTEGERS.  IGNORED.
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SETDEV,INITGR,SETIEE,SETWOR,SETCOL
C     SUBROUTINE SETDEV(NAME)
C     SUBROUTINE INITGR(MODE)
C     SUBROUTINE SETIEE(IEEE)
C     SUBROUTINE SETWOR(0.,0.,HSIZE,VSIZE)
C     SUBROUTINE SETCOL(ICOL)
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
C
      INTEGER MODE,IEEE
      INTEGER ICOL,NCOL,MCOL
      CHARACTER*22 NAME
C
      NAME(1:22)='/C:\FOR4\HALOIBMV.DEV/'
      MODE=7
      IEEE=1
      ICOL=15
      HSIZE=29.7
      VSIZE=21.0
      CALL SETDEV(NAME)
      CALL INITGR(MODE)
      CALL SETIEE(IEEE)
C     CALL INQCRA(MCOL)
      CALL SETWOR(0.,0.,HSIZE,VSIZE)
      CALL SETCOL(ICOL)
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE PLOT(XPAGE,YPAGE,IPEN)
C
C INPUT:
C     XPAGE,YPAGE... COORDINATES, OFTEN IN CENTIMETRES.
C     IPEN... CONTROLS THE PLOTTING:
C             IPEN=2... THE PEN IS DOWN DURING THE MOVEMENT, THUS
C               DRAWING A LINE.
C             IPEN=3... THE PEN IS UP DURING THE MOVEMENT.
C             IPEN=999... TERMINATES PLOTTING INITIALIZED BY THE
C               SUBROUTINE PLOTS.
C
C NO OUTPUT.
```

```
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL MOVABS,LNABS,CLOSEG
C     SUBROUTINE MOVABS(XPAGE,YPAGE)
C     SUBROUTINE CALL LNABS(XPAGE,YPAGE)
C     SUBROUTINE CLOSEG()
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATION:
      CHARACTER*1 DUMMY
C
      IF(IPEN.EQ.3) THEN
        CALL MOVABS(XPAGE,YPAGE)
      ELSE IF(IPEN.EQ.2) THEN
        CALL LNABS(XPAGE,YPAGE)
      ELSE IF(IPEN.GE.999) THEN
        CALL CLOSEG()
        WRITE(*,'('' =3h'')')
      END IF
      RETURN
      END
C
C=======================================================================
C
      SUBROUTINE NEWPEN(INP)
      INTEGER INP
C
C INPUT:
C     INP... SPECIFIES COLOUR.
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL SETCOL
C     SUBROUTINE SETCOL(ICOL)
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
C
      CALL SETCOL(INP)
      RETURN
      END
C
C=======================================================================
C
```

```
C SUBROUTINE PACKAGE 'PLOTVAX' - PLOT INTERFACE SUBROUTINES TO BUPLOT
C ROUTINES ON VAX COMPUTERS
C
C=======================================================================
C
      SUBROUTINE PLOTS(I1,I2,I3)
      INTEGER I1,I2,I3
C
C INPUT:
C     I1,I2,I3...ANY INTEGERS.  IGNORED.
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL OPNPLT,ADVPLT
C     SUBROUTINE OPNPLT(MODE,WIDTH,NTERM)
C     SUBROUTINE ADVPLT()
C
C DATE: 1990, JANUARY 17
C CODED BY LUDEK KLIMES
C
C-----------------------------------------------------------------------
C
C     FLAG IF THE PLOT IS INITIALIZED
      LOGICAL OPEN
      SAVE OPEN
      DATA OPEN/.FALSE./
C
      IF(OPEN) THEN
        CALL ADVPLT()
      END IF
      CALL OPNPLT(1,21.4,4010)
      OPEN=.TRUE.
      RETURN
      END
C
C=======================================================================
C
C     SUBROUTINE PLOT(XPAGE,YPAGE,IPEN) IS CONTAINED IN BUPLOT SYSTEM.
C
C     NOTE: THE FEATURE IPEN=999 IS IGNORED.  THUS, PLOTTING WILL NOT
C     BE CLOSED.
C
C=======================================================================
C
      SUBROUTINE NEWPEN(INP)
      INTEGER INP
C
C INPUT:
C     INP... SPECIFIES COLOUR.
C
C NO OUTPUT.
C
C SUBROUTINES AND EXTERNAL FUNCTIONS REQUIRED:
      EXTERNAL PCOLOR
C     SUBROUTINE PCOLOR(INP)
C
C DATE: 1989, DECEMBER 19
C CODED BY LUDEK KLIMES
C
```

```
C-----------------------------------------------------------------------
C
C     AUXILIARY STORAGE LOCATIONS:
C
      CALL PCOLOR(INP)
      RETURN
      END
C
C=======================================================================
C
```

```
FILES *.DAT:  SAMPLE INPUT DATA FOR THE COMPLETE RAY TRACING PROGRAM

BY VLASTISLAV CERVENY, LUDEK KLIMES, IVAN PSENCIK

THE SAMPLE 3-D MODEL CONSISTS OF TWO LAYERS AND OF THE LENTICULAR
INCLUSION WITH EDGES, SITUATED IN THE UPPER LAYER, SEE THE SCHEMATIC
FIGURE 5 IN V.CERVENY, L.KLIMES, I.PSENCIK: COMPLETE SEISMIC-RAY
TRACING IN 3-D STRUCTURES (IN: D.DOORNBOS, ED.: SEISMOLOGICAL
ALGORITHMS, ACADEMIC PRESS 1988).

THE FILES *.DAT LISTED BELOW CONTAIN THE SAMPLE INPUT DATA FOR THE
COMPLETE RAY TRACING PROGRAM.  THE INPUT DATA CONSIST OF THE 7 DATA
SETS (1) TO (7) LISTED IN THE PACKAGE 'GUIDE.FOR'.  NOTE THAT SEVERAL
DATA SETS MAY BE LOCATED IN A SINGLE FILE.
    CRT.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (1) CRT
    MODEL.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (2) MODEL
    DCRT.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (3) DCRT
    INIT.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (4) INIT
    CODE.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (5) CODE
    RPAR.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (6) RPAR
    WRIT.DAT... FILE CONTAINING THE SAMPLE DATA SET:
            (7) WRIT

REMARKS WITHIN THE INPUT DATA FILES:
    THE COMMENTS IN BRACKETS SITUATED AT THE ENDS OF LINES, AFTER THE
            INPUT ITEMS, ARE NOT THE PART OF THE INPUT DATA.  THEY ARE
            JUST REMARKS PLACED SO AS TO BE SKIPPED WHEN READING THE
            INPUT DATA.
    THE LAST INPUT LINE OF A DATA FILE IS USUALLY FOLLOWED BY
            (A) EMPTY LINE, LINE '====' AND EMPTY LINE, OR BY
            (B) EMPTY LINE, LINE '----', EMPTY LINE, TEXT DESCRIBING
                THE INPUT DATA, EMPTY LINE, LINE '====' AND EMPTY
                LINE.

DATE: 1990, FEBRUARY 5

=========================================================================
```

```
'SAMPLE MAIN INPUT DATA CRT.DAT: DATA FILENAMES.  1990, JANUARY 18.'

'MODEL.DAT'
'DCRT.DAT'
'INIT.DAT'
'CODE.DAT'
'RPAR.DAT'
'WRIT.DAT'
'LOG.OUT'
------------------------------------------------------------------------

  MODEL.DAT, DCRT.DAT, INIT.DAT, CODE.DAT, RPAR.DAT AND WRIT.DAT ARE THE
      FILES WITH THE INPUT DATA.
  LOG.OUT IS THE OUTPUT LOG FILE CONTAINING VERY BRIEF NOTES ON THE
      PROGRAM RUNNING.

========================================================================
```

```
'SAMPLE INPUT DATA MODEL.DAT: DATA FOR THE MODEL.  1990, JANUARY 18.'
0  1  1                  (CARTESIAN COORDINATES, VELOCITIES, LOSS FACTORS)
0 60 -20 20 -25 10       (BOUNDARIES OF THE MODEL)
4 SURFACES
4 SIMPLE BLOCKS:
   -1   2  -3  /
   -1   3      /
    1   2  -4  /
    4          /
3 COMPLEX BLOCKS:
    1   3      /
    2          /
    4          /

'SURFACE' 1
    1   2  -3    0       (I.E. W(X1,X2)-X3=0, TENSION=0)
    3   3                (NUMBERS OF GRID POINTS)
    0  20  40            (X1 GRID COORDINATES)
  -20   0  20            (X2 GRID COORDINATES)

    0   0   0
    0 -12   0
    0   0   0            (X3 COORDINATES AT GRID POINTS)

'SURFACE' 2
    1  -3   0    0       (I.E. W(X1)-X3=0, TENSION=0)
    4                    (NUMBERS OF GRID POINTS)
    0  20  40   60       (X1 GRID COORDINATES)

    3   3   0  3.5       (X3 COORDINATES AT GRID POINTS)

'SURFACE' 3
    1   2  -3    0       (I.E. W(X1,X2)-X3=0, TENSION=0)
    3   3                (NUMBERS OF GRID POINTS)
    0  20  30            (X1 GRID COORDINATES)
  -20   0  20            (X2 GRID COORDINATES)

  -12 -12 -12
  -12  -5 -12
  -12 -12 -12            (X3 COORDINATES AT GRID POINTS)

'SURFACE' 4
    1  -3   0    0       (I.E. W(X1)-X3=0, TENSION=0)
    5                    (NUMBERS OF GRID POINTS)
    0  10  25   40   60  (X1 GRID COORDINATES)

  -16 -17 -20  -17  -15  (X3 COORDINATES AT GRID POINTS)

'END OF SURFACES' /

'COMPLEX BLOCK' 1
'VP ' 1                  (P WAVE VELOCITY)
    3   0   0    0       (I.E. VP=W(X3), TENSION=0)
    2                    (NUMBERS OF GRID POINTS)
    0 -20                (X3 GRID COORDINATES)

    4   6                (VELOCITIES AT GRID POINTS)

'DEN' 1                  (DENSITY)
    4   0   0    0       (I.E. DENSITY=W(VP), TENSION=0)
```

```
   2                    (NUMBERS OF GRID POINTS)
   0   1                (VP GRID COORDINATES)

  1.7 1.9               (DENSITIES AT GRID POINTS, DENSITY=1.7+0.2*VP)

'COMPLEX BLOCK' 2
'VP ' 1                 (P WAVE VELOCITY)
   0   0   0   0        (I.E. VP=CONSTANT, TENSION=0)
   6                    (THE VALUE OF VELOCITY)

'DEN' 1                 (DENSITY)
   0   0   0   0        (I.E. DENSITY=CONSTANT, TENSION=0)
  2.9                   (THE VALUE OF DENSITY)

'COMPLEX BLOCK' 3
'VP ' 1                 (P WAVE VELOCITY)
   0   0   0   0        (I.E. VP=CONSTANT, TENSION=0)
   7                    (THE VALUE OF VELOCITY)

'DEN' 1                 (DENSITY)
   0   0   0   0        (I.E. DENSITY=CONSTANT, TENSION=0)
  3.1                   (THE VALUE OF DENSITY)

'END OF COMPLEX BLOCKS, END OF THE INPUT DATA FOR THE MODEL' /
```

THIS 3-D MODEL CONSISTS OF TWO LAYERS AND OF THE LENTICULAR INCLUSION WITH EDGES, SITUATED IN THE UPPER LAYER, SEE THE SCHEMATIC FIGURE 5 IN V.CERVENY, L.KLIMES, I.PSENCIK: COMPLETE SEISMIC-RAY TRACING IN 3-D STRUCTURES (IN: D.DOORNBOS, ED.: SEISMOLOGICAL ALGORITHMS, ACADEMIC PRESS 1988). FOR THE GENERAL DESCRIPTION OF THE INPUT DATA FORMAT SEE THE SUBROUTINE PACKAGES 'MODEL', 'SRFC' AND 'PARM'. P WAVE VELOCITY IN THE UPPER LAYER (COMPLEX BLOCK 1) INCREASES LINEARLY WITH DEPTH AND HAS NO LATERAL VARIATIONS. THE LENTICULAR INCLUSION (COMPLEX BLOCK 2) AND THE BOTTOM LAYER (COMPLEX BLOCK 3) ARE HOMOGENEOUS. S WAVE VELOCITIES AND LOSS FACTORS ARE NOT SPECIFIED BY THESE INPUT DATA, THE DEFAULT VALUES (VS=0.57735*VP, ZERO LOSS FACTORS) ARE ASSUMED. THE COMMENTS IN BRACKETS (STARTING FROM THE COLUMN 23) IN THESE INPUT DATA HAVE NO INFLUENCE ON READING THE DATA.

THE INPUT DATA ARE STORED IN THE COMMON BLOCKS /MODELT/, /MODELC/ AND /VALC/. FOR THE DESCRIPTION OF THE COMMON BLOCKS /MODELT/ AND /MODELC/ REFER TO THE SUBROUTINE PACKAGE 'MODEL', FOR THE COMMON BLOCK /VALC/ REFER TO THE SUBROUTINE PACKAGE 'VAL'. THE ARRAYS KSB AND KCB OF THE COMMON BLOCK /MODELC/ AND THE ARRAYS IPAR AND RPAR OF THE COMMON BLOCK /VALC/ ARE FILLED BY THE PARAMETERS OF THE MODEL IN THE FOLLOWING WAY:

ARRAYS KSB AND KCB OF THE COMMON BLOCK /MODELC/:

```
I-----------I-----------I   I-----------I-----------I
I           I  K KSB(K) I   I           I  K KCB(K) I
I-----------I-----------I   I-----------I-----------I
I NSB       I  0     4  I   I NCB       I  0     3  I
I-----------I-----------I   I-----------I-----------I
I S.BLOCK 1 I  1     7  I   I C.BLOCK 1 I  1     5  I
I S.BLOCK 2 I  2     9  I   I C.BLOCK 2 I  2     6  I
```

```
I S.BLOCK 3 I  3     12  I
I S.BLOCK 4 I  4     13  I
I-----------I------------I
I           I  5     -1  I
I S.BLOCK 1 I  6      2  I
I           I  7     -3  I
I-----------I------------I
I S.BLOCK 2 I  8     -1  I
I           I  9      3  I
I-----------I------------I
I           I 10      1  I
I S.BLOCK 3 I 11      2  I
I           I 12     -4  I
I-----------I------------I
I S.BLOCK 4 I 13      4  I
I-----------I------------I
```

```
I C.BLOCK 3 I  3      7  I
I-----------I------------I
I C.BLOCK 1 I  4      1  I
I           I  5      3  I
I-----------I------------I
I C.BLOCK 2 I  6      2  I
I-----------I------------I
I C.BLOCK 3 I  7      4  I
I-----------I------------I
```

ARRAYS IPAR AND RPAR OF THE COMMON BLOCK /VACL/:

```
I-----------------------------------I------------I
I            DESCRIPTION            I  K IPAR(K) I
I-----------------------------------I------------I
I (A) NUMBER OF THE CLASSES         I  0      2  I
I-----------------------I-----------I------------I
I (B) CLASSES           I  CLASS 1  I  1      6  I
I                       I  CLASS 2  I  2      9  I
I-----------I-----------I-----------I------------I
I           I           I SURFACE 1 I  3     10  I
I           I CLASS 1   I SURFACE 2 I  4     11  I
I           I           I SURFACE 3 I  5     12  I
I (C) GROUPSI           I SURFACE 4 I  6     13  I
I           I-----------I-----------I------------I
I           I           I C.BLOCK 1 I  7     18  I
I           I CLASS 2   I C.BLOCK 2 I  8     23  I
I           I           I C.BLOCK 3 I  9     28  I
I-----------I-----------I-----------I------------I
I           I SURFACE 1 I           I 10     81  I
I           I-----------I-----------I------------I
I           I SURFACE 2 I           I 11    116  I
I           I-----------I-----------I------------I
I           I SURFACE 3 I           I 12    169  I
I           I-----------I-----------I------------I
I           I SURFACE 4 I           I 13    211  I
I           I-----------I-----------I------------I
I           I           I P VELOCITYI 14    232  I
I           I           I DENSITY   I 15    253  I
I           I C.BLOCK 1 I NO DATA   I 16    253  I
I (D)       I           I NO DATA   I 17    253  I
I FUNCTIONS I           I NO DATA   I 18    253  I
I           I-----------I-----------I------------I
I           I           I P VELOCITYI 19    260  I
I           I           I DENSITY   I 20    267  I
I           I C.BLOCK 2 I NO DATA   I 21    267  I
I           I           I NO DATA   I 22    267  I
I           I           I NO DATA   I 23    267  I
I           I-----------I-----------I------------I
I           I           I P VELOCITYI 24    274  I
I           I           I DENSITY   I 25    281  I
I           I C.BLOCK 3 I NO DATA   I 26    281  I
I           I           I NO DATA   I 27    281  I
```

		NO DATA	28 281

SURFACE 1:

K	IPAR(K) RPAR(K)
29	1
30	1.000
31	1
32	2
33	-3
34	0.000
35	3
36	3
37	0.000
38	20.000
39	40.000
40	-20.000
41	0.000
42	20.000
43	0.000
44	0.000
45	0.000
46	0.000
47	-12.000
48	0.000
49	0.000
50	0.000
51	0.000
52	0.002
53	0.000
54	0.002
55	0.167
56	0.167
57	0.000
58	-0.007
59	0.000
60	0.000
61	0.000
62	0.002
63	0.000
64	0.002
65	0.167
66	0.167
67	0.002
68	0.000
69	0.002
70	0.167
71	0.167
72	0.000
73	-0.007
74	0.000
75	0.000
76	0.000
77	0.002
78	0.000
79	0.002
80	0.167

SURFACE 2:

K	IPAR(K) RPAR(K)
82	1
83	1.000
84	1
85	-3
86	0
87	0.000
88	4
89	0.000
90	20.000
91	40.000
92	60.000
93	3.000
94	3.000
95	0.000
96	3.500
97	0.000
98	-0.019
99	0.029
100	0.000
101	0.000
102	0.000
103	0.000
104	0.000
105	0.000
106	0.000
107	0.000
108	0.000
109	0.000
110	0.000
111	0.000
112	0.000
113	0.000
114	0.000
115	0.000
116	0.000

SURFACE 3:

K	IPAR(K) RPAR(K)
117	1
118	1.000
119	1
120	2
121	-3
122	0.000
123	3
124	3
125	0.000
126	20.000
127	30.000
128	-20.000
129	0.000
130	20.000
131	-12.000
132	-7.200
133	-12.000
134	-8.000
135	2.200
136	-8.000
137	-12.000
138	-7.200
139	-12.000
140	0.004
141	0.000
142	0.004
143	0.067
144	0.067
145	0.000
146	-0.015
147	0.000
148	0.000
149	0.000
150	0.005
151	0.000
152	0.005
153	0.333
154	0.333
155	0.002
156	0.000
157	0.002
158	0.167
159	0.167
160	0.000
161	-0.007
162	0.000
163	0.000
164	0.000
165	0.002
166	0.000
167	0.002
168	0.167

SURFACE 4:

K	IPAR(K) RPAR(K)
170	1
171	1.000
172	1
173	-3
174	0
175	0.000
176	5
177	0.000
178	10.000
179	25.000
180	40.000
181	60.000
182	-16.000
183	-17.000
184	-20.000
185	-17.000
186	-15.000
187	0.000
188	-0.028
189	0.052
190	-0.020
191	0.000
192	0.000
193	0.000
194	0.000
195	0.000
196	0.000
197	0.000
198	0.000
199	0.000
200	0.000
201	0.000
202	0.000
203	0.000
204	0.000
205	0.000
206	0.000
207	0.000
208	0.000
209	0.000
210	0.000
211	0.000

```
       I  81    0.167 I                         I 169    0.167 I
       I--------------I                         I--------------I

       C.BLOCK 1: VP:     C.B.1: DENSITY:    C.BLOCK 2: VP:     C.BLOCK 3: VP:
       I--------------I   I--------------I   I--------------I   I--------------I
       I  K   IPAR(K) I   I  K   IPAR(K) I   I  K   IPAR(K) I   I  K   IPAR(K) I
       I      RPAR(K) I   I      RPAR(K) I   I      RPAR(K) I   I      RPAR(K) I
       I--------------I   I--------------I   I--------------I   I--------------I
       I 212   1      I   I 233   3      I   I 254   1      I   I 268   1      I
       I 213   1.000  I   I 234   1.000  I   I 255   1.000  I   I 269   1.000  I
       I 214   3      I   I 235   4      I   I 256   0      I   I 270   0      I
       I 215   0      I   I 236   0      I   I 257   0      I   I 271   0      I
       I 216   0      I   I 237   0      I   I 258   0      I   I 272   0      I
       I 217   0.000  I   I 238   0.000  I   I 259   0.000  I   I 273   0.000  I
       I 218   2      I   I 239   2      I   I 260   6.000  I   I 274   7.000  I
       I 219 -20.000  I   I 240   0.000  I   I--------------I   I--------------I
       I 220   0.000  I   I 241   1.000  I
       I 221   6.000  I   I 242   1.700  I   C.B.2: DENSITY:    C.B.3: DENSITY:
       I 222   4.000  I   I 243   1.900  I   I--------------I   I--------------I
       I 223   0.000  I   I 244   0.000  I   I  K   IPAR(K) I   I  K   IPAR(K) I
       I 224   0.000  I   I 245   0.000  I   I      RPAR(K) I   I      RPAR(K) I
       I 225   0.000  I   I 246   0.000  I   I--------------I   I--------------I
       I 226   0.000  I   I 247   0.000  I   I 261   3      I   I 275   3      I
       I 227   0.000  I   I 248   0.000  I   I 262   1.000  I   I 276   1.000  I
       I 228   0.000  I   I 249   0.000  I   I 263   0      I   I 277   0      I
       I 229   0.000  I   I 250   0.000  I   I 264   0      I   I 278   0      I
       I 230   0.000  I   I 251   0.000  I   I 265   0      I   I 279   0      I
       I 231   0.000  I   I 252   0.000  I   I 266   0.000  I   I 280   0.000  I
       I 232   0.000  I   I 253   0.000  I   I 267   2.900  I   I 281   3.100  I
       I--------------I   I--------------I   I--------------I   I--------------I
C
C===========================================================================
C
```

```
'SAMPLE INPUT DATA DCRT.DAT: NUMERICAL PARAMETERS.  1990, FEBRUARY 5.'

0   0   5   0            (KSTORE, NEXPS, NHLF, MODCRT)
2.0 0.5 0.010 0.005 0.005 0.005 0.001
/                        (COMPUTATIONAL VOLUME IS THE WHOLE MODEL)
/                        (NO AUXILIARY SURFACES)
/                        (NO END SURFACES)
2 /                      (STORING SURFACES)
------------------------------------------------------------------------

  THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING THE NUMERICAL
  PARAMETERS OF THE RAY TRACING MAY BE FOUND IN THE SUBROUTINE PACKAGE
  'RAY'.

========================================================================
```

```
'SAMPLE INPUT DATA INIT.DAT: INITIAL POINTS OF RAYS.  1990, FEBRUARY 5.'

0    2                     (SINGLE INITIAL POINT, GEOGRAPHIC-LIKE SPH.COOR.)
25.  5.  -15.   0.         (3 COORDINATES OF THE INITIAL POINT, TRAVEL TIME)
-----------------------------------------------------------------------------

  THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING THE INITIAL
  CONDITIONS FOR THE COMPUTED RAYS MAY BE FOUND IN THE SUBROUTINE
  PACKAGE 'INIT'.

=============================================================================
```

```
'SAMPLE INPUT DATA CODE.DAT: ELEMENTARY WAVE CODES.  1990, FEBRUARY 5.'
/
5   1 1 1 1 1 1 1 1 / (REFLECTION-TRANSMISSION CODE, REFRACTED WAVE)
5   1 2 1 1 1 1 1 1 / (REFLECTION-TRANSMISSION CODE, REFLECTED WAVE)
/
```

ABOVE SAMPLE INPUT DATA SET CODE.DAT SPECIFIES TWO ELEMENTARY WAVES TO BE COMPUTED.

THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING ELEMENTARY WAVES MAY BE FOUND IN THE SUBROUTINE PACKAGE 'CODE'.

===

```
'SAMPLE INPUT DATA RPAR.DAT: RAY PARAMETERS.  1990, FEBRUARY 5.'
 0.3927  6.0000  1.5708 -1.3090  1.5000  0.2618
/
/
/
```

ABOVE SAMPLE INPUT DATA SET RPAR.DAT ALLOWS FOR MAXIMUM OF 2 COMPUTED ELEMENTARY WAVES WITH THE SAME TAKE-OFF RAY PARAMETERS.

THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING THE TAKE-OFF PARAMETERS OF THE COMPUTED RAYS MAY BE FOUND IN THE SUBROUTINE PACKAGE 'RPAR'.

===

```
'SAMPLE DATA SET WRIT.DAT: NAMES OF OUTPUT FILES.  1990, FEBRUARY 5.'
'R .OUT'    'R I.OUT'
'S .OUT'    'S I.OUT'                    '1' '2' '3' '4' '5' '6' '7' '8' '9'/
'C E .OUT' 'C E I.OUT'
               ' 01' ' 02' ' 03' ' 04' ' 05' ' 06' ' 07' ' 08' ' 09' ' 10'
               ' 11' ' 12' ' 13' ' 14' ' 15' ' 16' ' 17' ' 18' ' 19' ' 20'
               ' 21' ' 22' ' 23' ' 24' ' 25' ' 26' ' 27' ' 28' ' 29' ' 30'/
'01' ' ' '01'
'02' ' ' '02'
'03' ' ' '03'
'04' ' ' '04'
'05' ' ' '05'
'06' ' ' '06'
'07' ' ' '07'
'08' ' ' '08'
'09' ' ' '09'
'10' ' ' '10'
```

SPECIFICATION OF THE NAMES OF OUTPUT FILES WITH COMPUTED QUANTITIES:

ABOVE SAMPLE INPUT DATA SET WRIT.DAT ALLOWS FOR MAXIMUM OF 9 SURFACES FOR STORING THE COMPUTED QUANTITIES, 10 ELEMENTARY WAVES WHICH RAYS MAY NOT EXCEED THE LENGTH OF 30 RAY ELEMENTS.

EXAMPLE: IF THE COMPUTED QUANTITIES ARE STORED ALONG RAYS AND AT 1 GIVEN SURFACE OF ANY INDEX, IF THERE ARE 2 ELEMENTARY WAVES AND NO STORING OF THE QUANTITIES AT THE ENDPOINTS OF RAY ELEMENTS, THEN (BESIDES THE LOG OUTPUT FILE) FOLLOWING 6 FILES WILL BE GENERATED:

```
    S1.OUT... COMPUTED QUANTITIES AT THE GIVEN SURFACE (BOTH WAVES).
    S1I.OUT... QUANTITIES AT THE INITIAL POINTS OF RAYS CORRESPONDING
            TO THE ABOVE FILE.
    R01.OUT... COMPUTED QUANTITIES ALONG RAYS OF THE FIRST WAVE.
    R01I.OUT... QUANTITIES AT THE INITIAL POINTS OF RAYS CORRESPONDING
            TO THE ABOVE FILE.
    R02.OUT... COMPUTED QUANTITIES ALONG RAYS OF THE SECOND WAVE.
    R02I.OUT... QUANTITIES AT THE INITIAL POINTS OF RAYS CORRESPONDING
            TO THE ABOVE FILE.
```

THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING THE NAMES OF OUTPUT FILES WITH COMPUTED QUANTITIES MAY BE FOUND IN THE SUBROUTINE PACKAGE 'WRIT'.

===

FILES *.SPH: SPHERICALLY SYMMETRIC EARTH.
GUIDE TO THE INPUT DATA FOR THE COMPLETE RAY TRACING PROGRAM.

BY RAY HADDON, LUDEK KLIMES

THE SPHERICALLY SYMMETRIC EARTH MODEL CONSISTS OF TWO COMPLEX BLOCKS (LAYERS). THE MANTLE (COMPLEX BLOCK 1) IS LIMITED BY EARTH'S SURFACE (SURFACE 1) AND THE CORE-MANTLE BOUNDARY (SURFACE 2). THE OUTER CORE (COMPLEX BLOCK 2) IS LIMITED BY THE CORE-MANTLE BOUNDARY (SURFACE 2) AND BY THE BOTTOM OF THE MODEL. THE MODEL IS SPECIFIED IN GEOGRAPHIC SPHERICAL COORDINATES: X1=LONGITUDE (IN RADIANS), X2=LATITUDE (IN RADIANS), X3=RADIUS.

THE FILES *.SPH LISTED BELOW CONTAIN THE SAMPLE INPUT DATA FOR THE COMPLETE RAY TRACING PROGRAM. THE INPUT DATA CONSIST OF THE 7 DATA SETS (1) TO (7) LISTED IN THE PACKAGE 'GUIDE.FOR'. NOTE THAT SEVERAL DATA SETS MAY BE LOCATED IN A SINGLE FILE.

```
    CRT.SPH... FILE CONTAINING THE DATA SETS:
            (1) CRT
            (3) DCRT
            (4) INIT
    MODEL.SPH... FILE CONTAINING THE DATA SET:
            (2) MODEL
    CODE.SPH... FILE CONTAINING THE DATA SET:
            (5) CODE
    RPAR.SPH... FILE CONTAINING THE DATA SET:
            (6) RPAR
THE LAST INPUT FILE IS COMMON WITH THE SAMPLE INPUT DATA FILES *.DAT:
    WRIT.DAT... FILE CONTAINING THE DATA SET:
            (7) WRIT
```

REMARKS WITHIN THE INPUT DATA FILES:

THE COMMENTS IN BRACKETS SITUATED AT THE ENDS OF LINES, AFTER THE INPUT ITEMS, ARE NOT THE PART OF THE INPUT DATA. THEY ARE JUST REMARKS PLACED SO AS TO BE SKIPPED WHEN READING THE INPUT DATA.

THE LAST INPUT LINE OF A DATA FILE IS USUALLY FOLLOWED BY
- (A) EMPTY LINE, LINE '====' AND EMPTY LINE, OR BY
- (B) EMPTY LINE, LINE '----', EMPTY LINE, TEXT DESCRIBING THE INPUT DATA, EMPTY LINE, LINE '====' AND EMPTY LINE.

DATE: 1990, FEBRUARY 5

===

```
'DATA FILE CRT.SPH: SPHERICALLY SYMMETRIC EARTH.  1990, FEBRUARY 05.'
'MODEL.SPH'
' '   (DATA SET DCRT.SPH IS INCLUDED IN THIS FILE)
' '   (DATA SET INIT.SPH IS INCLUDED IN THIS FILE)
'CODE.SPH'
'RPAR.SPH'
'WRIT.DAT'
'LOG.OUT'

'DATA SET DCRT.SPH'
0   0   5   0           (KSTORE, NEXPS, NHLF, MODCRT)
10.0  5.0 0.010 0.100 0.100 0.100 0.010
/                       (COMPUTATIONAL VOLUME IS THE WHOLE MODEL)
/                       (NO AUXILIARY SURFACES)
/                       (NO END SURFACES)
1 /                     (STORING SURFACES)

'DATA SET INIT.SPH'
0    2                  (SINGLE INITIAL POINT, GEOGRAPHIC-LIKE SPH.COOR.)
.0000 .0000 6371.  0. (3 COORDINATES OF THE INITIAL POINT, TRAVEL TIME)
------------------------------------------------------------------------

  THE GENERAL DESCRIPTION OF THE MAIN INPUT DATA SET CRT MAY BE FOUND IN
      THE PACKAGE 'CRT'.
  THE GENERAL DESCRIPTION OF THE INPUT DATA SET DCRT CONTAINING THE
      PARAMETERS OF THE RAY TRACING MAY BE FOUND IN THE SUBROUTINE
      PACKAGE 'RAY'.
  THE GENERAL DESCRIPTION OF THE INPUT DATA SET INIT SPECIFYING THE
      INITIAL CONDITIONS FOR THE COMPUTED RAYS MAY BE FOUND IN THE
      SUBROUTINE PACKAGE 'INIT'.

========================================================================
```

```
'DATA FILE MODEL.SPH: SPHERICALLY SYMMETRIC EARTH.  1990, FEBRUARY 06.'
2 1 1                      (SPHERICAL COORDINATES, VELOCITIES, LOSS FACTORS)
-3.14 3.14 -1.57 1.57 1221 6400
2 SURFACES
2 SIMPLE BLOCKS
1 -2 /
2    /
2 COMPLEX BLOCKS
1 /
2 /
'SURFACE' 1
-3 0 0 0                   (I.E. CONSTANT-X3=0, WHERE X3 IS THE RADIUS)
6371                       (THE ABOVE CONSTANT)
'SURFACE' 2
-3 0 0 0                   (I.E. CONSTANT-X3=0, WHERE X3 IS THE RADIUS)
3483                       (THE ABOVE CONSTANT)
'END OF SURFACES' /
'COMPLEX BLOCK' 1
'VP ' 1
3 0 0 0                    (I.E. VP=W(X3), TENSION=0)
30                         (NUMBER OF GRID POINTS)
  6371    6271    6171    6071    5971    5871    5821    5701
  5600    5500    5400    5300    5200    5100    5000    4900
  4800    4700    4600    4500    4400    4300    4200    4100
  4000    3900    3800    3700    3600    3483
   7.7000  7.9000  8.0000  8.6920  9.0000  9.6459  9.9018 10.5000
  11.0656 11.2450 11.4156 11.5783 11.7336 11.8821 12.0244 12.1613
  12.2932 12.4208 12.5447 12.6655 12.7839 12.9004 13.0158 13.1106
  13.23   13.35   13.47   13.59   13.71   13.86
'VS ' 1
3 0 0 0                    (I.E. VS=W(X3), TENSION=0)
30                         (NUMBER OF GRID POINTS)
  6371    6271    6171    6071    5971    5871    5791    5671
  5571    5471    5371    5271    5171    5071    4971    4871
  4771    4671    4571    4471    4371    4271    4171    4071
  3971    3871    3771    3671    3571    3483
   4.5000  4.5000  4.5000  4.5000  4.7200  5.5000  5.6500  5.9300
   6.1500  6.2700  6.3550  6.4250  6.5000  6.5700  6.6200  6.6680
   6.7160  6.7640  6.8120  6.8600  6.9080  6.9560  7.0040  7.0520
   7.1000  7.1480  7.1960  7.2440  7.2920  7.3400
'COMPLEX BLOCK' 2
'VP ' 1
3 0 0 0                    (I.E. VP=W(X3), TENSION=0)
24                         (NUMBER OF GRID POINTS)
  3483    3400    3300    3200    3100    3000    2900    2800
  2700    2600    2500    2400    2300    2200    2100    2000
  1900    1800    1700    1600    1500    1400    1300    1221
   8.0648  8.1994  8.3602  8.5130  8.6580  8.7957  8.9263  9.0501
   9.1675  9.2788  9.3842  9.4841  9.5788  9.6687  9.7539  9.8350
   9.9121  9.9855 10.0557 10.1229 10.1874 10.2496 10.3097 10.3557
'VS ' 1
0 0 0 0                    (I.E. VS=CONSTANT)
0                          (THE ABOVE CONSTANT)
'END OF COMPLEX BLOCKS, END OF THE INPUT DATA FOR THE MODEL' /
```

THIS SPHERICALLY SYMMETRIC EARTH MODEL CONSISTS OF TWO COMPLEX BLOCKS (LAYERS). THE MANTLE (COMPLEX BLOCK 1) IS LIMITED BY EARTH'S SURFACE

(SURFACE 1) AND THE CORE-MANTLE BOUNDARY (SURFACE 2). THE OUTER CORE (COMPLEX BLOCK 2) IS LIMITED BY THE CORE-MANTLE BOUNDARY (SURFACE 2) AND BY THE BOTTOM OF THE MODEL. THE MODEL IS SPECIFIED IN GEOGRAPHIC SPHERICAL COORDINATES: X1=LONGITUDE (IN RADIANS), X2=LATITUDE (IN RADIANS), X3=RADIUS.

FOR THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING THE MODEL SEE THE SUBROUTINE PACKAGES 'MODEL', 'SRFC' AND 'PARM'.

==

```
'DATA FILE CODE.SPH: SPHERICALLY SYMMETRIC EARTH.  1990, FEBRUARY 05.'
/
5   1  1  1  1  1  1  1  1 /             (R/T CODE, REFRACTED PKP WAVE)
5   1  1 -1 -1 -1 -1 -1 -1 /             (R/T CODE, REFRACTED PKS WAVE)
5  -1  1  1  1  1  1  1  1 /             (R/T CODE, REFRACTED SKP WAVE)
5  -1  1 -1 -1 -1 -1 -1 -1 /             (R/T CODE, REFRACTED SKS WAVE)
5   1  2  1  1  1  1  1  1 /             (R/T CODE, REFLECTED PP WAVE)
5   1 -2 -1 -1 -1 -1 -1 -1 /             (R/T CODE, REFLECTED PS WAVE)
5  -1  2  1  1  1  1  1  1 /             (R/T CODE, REFLECTED SP WAVE)
5  -1 -2 -1 -1 -1 -1 -1 -1 /             (R/T CODE, REFLECTED SS WAVE)
/
------------------------------------------------------------------------

  THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING ELEMENTARY WAVES
  MAY BE FOUND IN THE SUBROUTINE PACKAGE 'CODE'.

========================================================================
```

```
'DATA FILE RPAR.SPH: SPHERICALLY SYMMETRIC EARTH.  1990, FEBRUARY 05.'
 0 0 0  -1.149001  0 0
 0 0 0  -1.166767  0 0
 0 0 0  -1.183206  0 0
 0 0 0  -1.200071  0 0
 0 0 0  -1.216433  0 0
 0 0 0  -1.232367  0 0
 0 0 0  -1.247855  0 0
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
/
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
/
 0 0 0  -1.149001  0 0
 0 0 0  -1.166767  0 0
 0 0 0  -1.183206  0 0
 0 0 0  -1.200071  0 0
 0 0 0  -1.216433  0 0
 0 0 0  -1.232367  0 0
 0 0 0  -1.247855  0 0
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
 0 0 0  -1.435000 -1.500000 -0.012000
/
 0 0 0  -1.232367  0 0
 0 0 0  -1.247855  0 0
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
```

```
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
 0 0 0  -1.435000 -1.500000 -0.012000
/
 0 0 0  -1.149001  0 0
 0 0 0  -1.166767  0 0
 0 0 0  -1.183206  0 0
 0 0 0  -1.200071  0 0
 0 0 0  -1.216433  0 0
 0 0 0  -1.232367  0 0
 0 0 0  -1.247855  0 0
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
/
 0 0 0  -1.149001  0 0
 0 0 0  -1.166767  0 0
 0 0 0  -1.183206  0 0
 0 0 0  -1.200071  0 0
 0 0 0  -1.216433  0 0
 0 0 0  -1.232367  0 0
 0 0 0  -1.247855  0 0
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
/
 0 0 0  -1.149001  0 0
 0 0 0  -1.166767  0 0
 0 0 0  -1.183206  0 0
 0 0 0  -1.200071  0 0
 0 0 0  -1.216433  0 0
 0 0 0  -1.232367  0 0
 0 0 0  -1.247855  0 0
 0 0 0  -1.264124  0 0
 0 0 0  -1.284346  0 0
 0 0 0  -1.303761  0 0
 0 0 0  -1.322477  0 0
 0 0 0  -1.340576  0 0
 0 0 0  -1.358135  0 0
 0 0 0  -1.375221  0 0
 0 0 0  -1.391888  0 0
 0 0 0  -1.408185  0 0
 0 0 0  -1.422820  0 0
/
 0 0 0  -1.149001  0 0
 0 0 0  -1.166767  0 0
```

```
0 0 0  -1.183206  0 0
0 0 0  -1.200071  0 0
0 0 0  -1.216433  0 0
0 0 0  -1.232367  0 0
0 0 0  -1.247855  0 0
0 0 0  -1.264124  0 0
0 0 0  -1.284346  0 0
0 0 0  -1.303761  0 0
0 0 0  -1.322477  0 0
0 0 0  -1.340576  0 0
0 0 0  -1.358135  0 0
0 0 0  -1.375221  0 0
0 0 0  -1.391888  0 0
0 0 0  -1.408185  0 0
0 0 0  -1.422820  0 0
/
```

THE GENERAL DESCRIPTION OF THE INPUT DATA SPECIFYING THE TAKE-OFF PARAMETERS OF THE COMPUTED RAYS MAY BE FOUND IN THE SUBROUTINE PACKAGE 'RPAR'.

==

```
$ FORTRAN CRT.FOR
$ FORTRAN METRIC.FOR
$ FORTRAN MODEL.FOR
$ FORTRAN SRFC.FOR
$ FORTRAN PARM.FOR
$ FORTRAN VAL.FOR
$ FORTRAN FITCRT.FOR
$ FORTRAN CODE.FOR
$ FORTRAN RAY.FOR
$ FORTRAN RAYCB.FOR
$ FORTRAN HPCG.FOR
$ FORTRAN TRANS.FOR
$ FORTRAN COEF.FOR
$ FORTRAN RPAR.FOR
$ FORTRAN INIT.FOR
$ FORTRAN WRIT.FOR
$ FORTRAN SCROPC.FOR
$ FORTRAN PLOTNUL.FOR
$ LINK CRT,METRIC,MODEL,SRFC,PARM,VAL,FITCRT,CODE,RAY,RAYCB,HPCG,TRANS,C...
$ WRITE SYS$OUTPUT "COPY THE MAIN INPUT DATA FILE TO FOR005.DAT BEFORE R...
```