DIRTRAN-I USER'S MANUAL

Prepared by
DAVID DVORE
Aerodyne Research, Inc.
Crosby Drive
Bedford, MA 01730

Under Contract DAAK20-79-Q-0043
CONTRACT MONITOR: LOUIS DUNCAN

Approved for public release; distribution unlimited

US Army Electronics Research and Development Command
ATMOSPHERIC SCIENCES LABORATORY
White Sands Missile Range, NM 88002
NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government endorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.
### DIRTRAN-I User's Manual

Approved for public release; distribution unlimited

Contract Monitor: Louis Duncan

The DIRTRAN-I Code is a computer-implemented model for predicting the optical effects of an explosion-produced dust cloud as it disperses in the lower atmosphere. This model is based on first principles of fluid dynamics, atmospherics, and optics. The model has been validated using cloud dimension and line-of-sight optical transmission data from the DIRT I and Graf II-Winter Army dust obscuration field trials. The DIRTRAN-I Code exploits information available about crater sizes produced by explosions in conjunction with...
with distinct models for coupling of energy to the ground for artillery projectiles versus bare charges. The model recognizes that dust ejecta are partitioned into a buoyantly rising fireball and a non-buoyant "dust skirt" which is subject to diffusion in the vertically sheared wind field. The DIRTRAN-I Code solves separately for these two clouds. The solutions are based on atmospheric diffusion theory and take into account the effects of wind and temperature profiles in the constant shear stress layer of the lower atmosphere for different atmospheric stability categories. Separate treatment is given to particles of different sizes, the larger ones being allowed to settle out. Outputs of the code include dust cloud displacement and dimensions for both the non-buoyant wind-dominated skirt and the initial buoyant fireball as it is wind blown and eventually also becomes subject to wind diffusion. Line-of-sight transmittances at several wavelength bands (visible: 0.4 - 0.7 \( \mu \)m; infrared: 0.8 - 1.1, 3.5 - 4.0, and 8.5 - 12 \( \mu \)m; mm wave: 94 - 140 GHz) are also output options. Because of the analytic solutions derived from first-principle physics, the DIRTRAN-I Code has been designed to keep both storage and computation time to a minimum. In order to be machine transportable, the code has been written in ANSI FORTRAN IV with no system specific enhancements.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>FUNCTIONS AND SUBROUTINES</td>
<td>6</td>
</tr>
<tr>
<td>3.</td>
<td>USAGE</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Common Blocks Used to Pass Information to DIRTRAN-I</td>
<td>11</td>
</tr>
<tr>
<td>3.1.1</td>
<td>COMMON /IOUNIT/ IOIN, IOUT, ISPTPF, LOUNIT, DIRTU, NBSCAT</td>
<td>11</td>
</tr>
<tr>
<td>3.1.2</td>
<td>COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT, FOGPRB, IPASCT, WNDVEL, WNDDIR</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Input from FORTRAN Logical Unit IOIN</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Example</td>
<td>14</td>
</tr>
<tr>
<td>4.</td>
<td>DIRTRAN-I GEOMETRY</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>User Coordinates</td>
<td>20</td>
</tr>
<tr>
<td>4.2</td>
<td>Wind Geometry</td>
<td>20</td>
</tr>
<tr>
<td>4.3</td>
<td>Transmission Geometry</td>
<td>20</td>
</tr>
<tr>
<td>4.4</td>
<td>Observer Geometry</td>
<td>20</td>
</tr>
<tr>
<td>5.</td>
<td>LISTING OF DIRTRAN-I CODE</td>
<td>23</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

The DIRTRAN-I Code is a computer-implemented model for predicting the optical effects of an explosion-produced dust cloud as it disperses in the lower atmosphere. This model is based on first principles of fluid dynamics, atmospherics, and optics. The model has been validated using cloud dimension and line-of-sight optical transmission data from the DIRT I and Graf II-Winter Army dust obscuration field trials. The DIRTRAN-I Code exploits information available about crater sizes produced by explosions in conjunction with distinct models for coupling of energy to the ground for artillery projectiles versus bare charges. The model recognizes that dust ejecta are partitioned into a buoyantly rising fireball and a non-buoyant "dust skirt" which is subject to diffusion in the vertically sheared wind field. The DIRTRAN-I Code solves separately for these two clouds. The solutions are based on atmospheric diffusion theory and take into account the effects of wind and temperature profiles in the constant shear stress layer of the lower atmosphere for different atmospheric stability categories. Separate treatment is given to particles of different sizes, the larger ones being allowed to settle out. Outputs of the code include dust cloud displacement and dimensions for both the non-buoyant wind-dominated skirt and the initial buoyant fireball as it is wind blown and eventually also becomes subject to wind diffusion. Line-of-sight transmittances at several wavelength bands (visible: 0.4 - 0.7 \( \mu \text{m} \); infrared: 0.8 - 1.1, 3.5 - 4.0, and 8.5 - 12 \( \mu \text{m} \); mm wave: 94 - 140 GHz) are also output options.

Because of the analytic solutions derived from first-principle physics, the DIRTRAN-I Code has been designed to keep both storage and computation time to a minimum. In order to be machine transportable, the code has been written in ANSI FORTRAN IV with no system specific enhancements.
2. FUNCTIONS AND SUBROUTINES

There is a great deal of internal documentation in the code listed in Section 5. In Figs. 2.1 through 2.5 are box diagrams illustrating the structure of the code.

![Diagram of controlling routines]

Figure 2.1 Controlling Routines.
Figure 2.2 Routines for Interpreting Meteorological Data.

Figure 2.3 Routines for Buoyant Rise Model
Figure 2.4 Routines for Wind Dispersion Model.
Figure 2.5 Routines for Determining Cloud Dimensions.

- COMPUTES VISIBLE CLOUD DIMENSIONS FROM BUOYANT AND WIND DISPERSION
- FINDS EDGE OF THE CLOUD AT A SPECIFIED MELOE
- CLIMB TRACES CONTOUR OF TWO-DIMENSIONAL FUNCTION
- SCOMP LOCATES FIRST POINT OF CONTOUR
- PTISPIC SEARCHES FOR A STRING OF CONTOURS FROM A STARTING POINT
- IMTRP INTERPOLATES COORDINATES BETWEEN POINTS
- SECANT METHOD FOR CONVERGING ON THE ZERO OF A FUNCTION GIVEN AN INITIAL GUES
Miscellaneous Routines

**FUNCT** Two-dimensional function which provides optically weighted column densities for visible contour tracing.

**GRAD2** Computes the two-dimensional gradient vector of a function.

**GFUN** The restriction of a function of two variables to a line specified by a base point and direction vector.

**PERP** Computes a vector rotated $90^\circ$ counterclockwise from a given vector.

**UNIT** Computes the length of a vector and a unit vector in the same direction as a given vector.

**VSUN** Adds a given scaler multiple of one given vector to another.
3. USAGE

The user accesses the DIRTRAN-I Code via Subroutine DRTRAN.

\[
\text{CALL DRTRAN (WAVEL, ICLMAT, TRNLOS, IERR)}
\]

where:

\[
\begin{align*}
\text{WAVEL} & \quad \text{is the wavelength in micrometers. Valid ranges are:} \\
0.4 & \quad \text{to} \quad 0.7 \text{ micrometers} \\
0.8 & \quad \text{to} \quad 1.1 \text{ micrometers} \\
3.5 & \quad \text{to} \quad 4.0 \text{ micrometers} \\
8.5 & \quad \text{to} \quad 12.0 \text{ micrometers} \\
2100.0 & \quad \text{to} \quad 3200.0 \text{ micrometers}
\end{align*}
\]

\[
\text{ICLMAT} \quad \text{is an integer index which is 0 if meteorological data is read from an external unit with other inputs and is 1 if passed in COMMON /CLYMAT/}.
\]

\[
\text{TRNLOS} \quad \text{is the returned value of the transmittance along the line-of-sight between the transmitter and the receiver.}
\]

\[
\text{IERR} \quad \text{is an integer error code which is returned one if a fatal error occurs and zero otherwise.}
\]

3.1 Common Blocks Used to Pass Information to DIRTRAN-I

There are two common blocks which communicate information to DIRTRAN-I:

\[
/\text{IOUNIT}/ \quad \text{and} \quad /\text{CLYMAT}/.
\]

3.1.1 COMMON /IOUNIT/ IOIN, IOOUT, ISPTPF, IOUNIT, NDIRTU, NRSCAT

Of the variables in /IOUNIT/ only IOIN, IOOUT, and NDIRTU are used by DIRTRAN-I.

\[
\text{IOIN} \quad \text{The Fortran logical unit from which DIRTRAN-I reads input data to be described in Subsection 3.2.}
\]
IOOUT The Fortran logical unit onto which DIRTRAN-I writes output and error messages.

NDIRU The Fortran logical unit from which DIRTRAN-I reads the data file containing tabulated values for the moments of the atmospheric diffusion equation.

3.1.2 COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDMT, CLDHYT, FOGRBB, IPASCT, WNDVEL, WNDDIR

DIRTRAN-I uses only TEMP, IPASCT, WNDVEL, and WNDDIR.

TEMP Temperature in degrees C taken at approximately two meters above ground.

IPASCT Integer 1 - 6 corresponding to Pasquill Categories A - F.

WNDVEL The wind speed in meters per second measured at approximately two meters above ground.

WNDDIR The wind direction (in degrees clockwise from true north) from which the wind is blowing. With this option the user's coordinate system must have the positive x-axis pointing east and positive y-axis pointing north. (Thus 0° corresponds to a wind blowing from the north to the south; 90° is a wind blowing from the east to the west; etc.)

3.2 Input from FORTRAN Logical Unit IOIN

The input read from logical unit, IOIN, provides the user with a variety of ways to use DIRTRAN-I. There are seven types of records to be distinguished on the input file. They are listed here with the names of variables contained on each record and the format type.
The description of the variables is in the comments in subroutine DRTRAN and appears on Pages 5-1 - 5-6 in this manual. Additional descriptions for some of the variables appears in Section 4.

The input file contains one or several sequences of these records each of which must begin with record 1, end with record 7, and contain a subset of records 2 - 6 corresponding to the entries in record 1. In each sequence, each of records 2 through 6 must appear if and only if the corresponding control variable entered in record 1 is .TRUE..

If NEWATM is .TRUE. then record 2 must appear.
If NEWSRC is .TRUE. then record 3 must appear.
If LOSTRN is .TRUE. then record 4 must appear.
If EDGE is .TRUE. then record 5 must appear.
If NEWTIM is .TRUE. then record 6 must appear.

On the first record of the input file NEWATM, NEWSRC, and NEWTIM must all be .TRUE.. This initializes the meteorological conditions, the charge and soil characteristics, and time after blast for the first observation. After that, DIRTRAN-I assumes that:
Meteorological data are unchanged until NEWATM is .TRUE. again.

Charge and Soil Characteristics are unchanged until NEWSRC is .TRUE. again.

Time after blast is unchanged until NEWTIM is .TRUE. again.

The first time that NEWATM is .TRUE., if ICLMAT = 1 (see arguments of DRTRAN in Section 2), then record 2 should not appear following record 1. This is the only time that a control variable in record 1 may be .TRUE. without the corresponding record 2 - 6 following.

Record 7 contains the control variable, CNTNU, which is .TRUE. if another sequence of records 1 - 7 is to be read on this call of DRTRAN and .FALSE. is DRTRAN is to return control to the calling routine. An example follows.

3.3 Example
The results from the above input file follow.
DIRTRAN-I DUST CLOUD INFRARED TRANSMISSION CALCULATION

ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED.

PASQUILL CATEGORY  
HT 1.00 TEMP 271.30  
WIND DIRECTION 90.00

SOIL INDEX 1
CHARGE INDEX 1
WEIGHT OF CHARGE 15.00
DETONATION DEPTH 0.00
DEPTH OF SOD 0.15
SOURCE COORDINATES 0.00 0.00

TIME AFTER BLAST 1.00

WAVELENGTH 0.55 MICROMETERS
TRANSMITTER COORDINATES 175.00 -1148.00 1.80
RECEIVER COORDINATES -173.00 1135.00 1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT 0.149E-02

OBSERVER COORDINATES 175.00 -1148.00
THE HEIGHT OF THE CLOUD IS 8.39 METERS
THE CENTROID COORDINATES ARE 0.41 4.79
THE WIDTH AT THE CENTROID IS 7.19 METERS
THE WIDTH AT 1.80 METERS IS 14.06 METERS
5 CONTOUR POINTS HAVE BEEN DETERMINED
-6.654 1.800
-3.164 4.794
0.410 6.388
4.003 4.794
7.406 1.800

TIME AFTER BLAST 2.00

WAVELENGTH 0.55 MICROMETERS
TRANSMITTER COORDINATES 175.00 -1148.90 1.80
RECEIVER COORDINATES -173.00 1135.00 1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT 0.296E-02

OBSERVER COORDINATES 175.00 -1148.00
THE HEIGHT OF THE CLOUD IS 12.83 METERS
THE CENTROID COORDINATES ARE 0.83 8.34
THE WIDTH AT THE CENTROID IS 8.97 METERS
THE WIDTH AT 1.80 METERS IS 14.69 METERS
5 CONTOUR POINTS HAVE BEEN DETERMINED
-6.590  1.800
-3.650  8.345
 0.832  12.827
 5.315  8.345
 8.098  1.800

TIME AFTER BLAST  5.00
WAVELENGTH  0.55 MICROMETERS
TRANSMITTER COORDINATES  175.00  -1148.90  1.80
RECEIVER COORDINATES  -173.00  1135.00  1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT  0.156E-01

OBSERVER COORDINATES  175.00  -1148.00
THE HEIGHT OF THE CLOUD IS  23.46 METERS
THE CENTROID COORDINATES ARE  2.21  16.85
THE WIDTH AT THE CENTROID IS 13.23 METERS
THE WIDTH AT 1.80 METERS IS 16.56 METERS
5 CONTOUR POINTS HAVE BEEN DETERMINED
-6.396  1.800
-4.409  16.845
 2.206  23.462
 8.825  16.845
10.166  1.800

TIME AFTER BLAST  10.00
WAVELENGTH  0.55 MICROMETERS
TRANSMITTER COORDINATES  175.00  -1148.90  1.80
RECEIVER COORDINATES  -173.00  1135.00  1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT  0.107E-00

OBSERVER COORDINATES  175.00  -1148.00
THE HEIGHT OF THE CLOUD IS  36.10 METERS
THE CENTROID COORDINATES ARE  4.71  26.94
THE WIDTH AT THE CENTROID IS 18.32 METERS
THE WIDTH AT 1.80 METERS IS 19.37 METERS
5 CONTOUR POINTS HAVE BEEN DETERMINED
-5.762  1.800
-4.453  26.943
 4.709  36.105
13.671  26.943
13.613  1.800
**TIME AFTER BLAST**

**20.00**

**WAVELENGTH**

0.55 MICROMETERS

**TRANSMITTER COORDINATES**

175.00 -1148.90 1.80

**RECEIVER COORDINATES**

-173.00 1135.00 1.80

**TRANSMITTANCE ALONG THE LINE-OF-SIGHT**

0.508E 00

**OBSERVER COORDINATES**

175.00 -1148.00

-4.179 1.800
-11.030 40.057
10.032 59.232
30.983 40.057
19.883 1.800

**THE HEIGHT OF THE CLOUD IS**

59.23 METERS

9.98 40.06

**THE WIDTH AT THE CENTROID IS**

42.01 METERS

24.06 METERS

5 CONTOUR POINTS HAVE BEEN DETERMINED

**TIME AFTER BLAST**

**30.00**

**WAVELENGTH**

0.55 MICROMETERS

**TRANSMITTER COORDINATES**

175.00 -1148.90 1.80

**RECEIVER COORDINATES**

-173.00 1135.00 1.80

**TRANSMITTANCE ALONG THE LINE-OF-SIGHT**

0.789E 00

**OBSERVER COORDINATES**

175.00 -1148.00

-4.265 1.800
-18.638 40.178
17.039 67.779
50.413 40.178
25.528 1.800

-2.285 1.800

-18.638 40.178
17.039 67.779
50.413 40.178
25.528 1.800

**THE HEIGHT OF THE CLOUD IS**

67.78 METERS

15.89 40.18

**THE WIDTH AT THE CENTROID IS**

69.05 METERS

27.81 METERS

5 CONTOUR POINTS HAVE BEEN DETERMINED

**TIME AFTER BLAST**

**40.00**

**WAVELENGTH**

0.55 MICROMETERS

**TRANSMITTER COORDINATES**

175.00 -1148.90 1.80

**RECEIVER COORDINATES**

-173.00 1135.00 1.80

**TRANSMITTANCE ALONG THE LINE-OF-SIGHT**

0.909E 00

5 CONTOUR POINTS HAVE BEEN DETERMINED
**OBSERVER COORDINATES**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>175.00</td>
<td>-1148.00</td>
</tr>
</tbody>
</table>

The height of the cloud is **72.35 Meters**.
The centroid coordinates are **21.67 40.70**.
The width at the centroid is **83.14 Meters**.
The width at 1.80 Meters is **30.63 Meters**.

5 contour points have been determined:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.235</td>
<td>1.800</td>
</tr>
<tr>
<td>-19.906</td>
<td>40.703</td>
</tr>
<tr>
<td>20.352</td>
<td>72.355</td>
</tr>
<tr>
<td>63.238</td>
<td>40.703</td>
</tr>
<tr>
<td>30.860</td>
<td>1.800</td>
</tr>
</tbody>
</table>

**TIME AFTER BLAST**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>175.00</td>
<td>-1148.90</td>
</tr>
</tbody>
</table>

Wavelength: **0.55 MICROMETERS**

Transmitter coordinates: **175.00 -1148.90 1.80**

Receiver coordinates: **-173.00 1135.0 1.80**

Transmittance along the line-of-sight: **0.977E00**

**OBSERVER COORDINATES**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>175.00</td>
<td>-1148.00</td>
</tr>
</tbody>
</table>

The height of the cloud is **77.53 Meters**.
The centroid coordinates are **33.99 48.28**.
The width at the centroid is **99.13 Meters**.
The width at 1.80 Meters is **36.56 Meters**.

5 contour points have been determined:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.962</td>
<td>1.800</td>
</tr>
<tr>
<td>-15.571</td>
<td>48.284</td>
</tr>
<tr>
<td>34.569</td>
<td>77.535</td>
</tr>
<tr>
<td>83.556</td>
<td>48.284</td>
</tr>
<tr>
<td>41.524</td>
<td>1.800</td>
</tr>
</tbody>
</table>
4. DIRTRAN-I GEOMETRY

4.1 User Coordinates

A top view of the (x,y) user coordinate plane for the DIRTRAN-I geometry is shown in Fig. 4.1. All inputs are in this user coordinate plane. Negative coordinates are allowed.

4.2 Wind Geometry

THWND is the angle in degrees that the wind velocity vector makes with the user's positive x-axis and is measured counter-clockwise. If the x-axis points east, then the following relationship holds between THWND and WINDIR (which is the standard meteorological wind direction as discussed in Sub-Section 3.1.2):

\[ \text{THWND} = 270 - \text{WINDIR} \]

4.3 Transmission Geometry

The height above ground of the transmitter and receiver, TRNCOR(3) and RECCOR(3), must be equal and must be between 0 and 5 meters. The transmitter height will be used as a default for the receiver height.

4.4 Observer Geometry

A local coordinate system, x', y', z' is used for viewing geometry. The x'-axis is aligned from the detonation point with coordinates, SRCCOR(I), to the observer position with coordinates, OBSCOR(I). The y'-axis is rotated 90° counterclockwise from the x'-axis, and the z'-axis is vertical so as to make a right-handed coordinate system. The visible cloud is projected along the observer's line-of-sight, the x'-axis. The visible cloud contour thus appears on the y'-z' coordinate plane, shown in Fig. 4.2. Five contour points are reported by DIRTRAN-I:
Figure 4.1 Top View of DIRTRAN-I Geometry.
The left and right edges of the cloud at a specified height, SPCHT,
The left and right edges of the cloud at the centroid height, and
The top of the cloud.

The centroid is determined by first locating the leading edge of the cloud which is (CPTS(1,4), CPTS(2,4)) in Fig. 4.2. A horizontal line is drawn across the cloud to determine (CPTS(1,2), CPTS(2,2)) in Fig. 4.2. The centroid is defined to be the midpoint of these two.
5. LISTING OF DIRTRAN-I CODE

DIRTRAN-I CODE

TEST CALLING ROUTINE

COMMON /IOUNIT/ IOIN, IOOUT, ISPTPF, LOUNIT, NDIRTU, NBSCAT
DATA IOIN, IOOUT, NDIRTU/5, 1, 7/
WAVE= .55
ICLMAT= 0
CALL DRTRAN(WAVE, ICLMAT, TRNLOS, IERR)
STOP
END

UTILITY ROUTINE FOR INTERFACING DIRTRAN-I WITH EC-SAEL

************************************** SUBFILE 1 **************************************

SUBROUTINE DRTRAN(WAVE, ICLMAT, TRNLOS, IERR)
IMPLICIT INTEGER*4 (I-N)
LOGICAL NEWATM, NEWSRC, LOSTRN, EDGE, NEWTIM, CNTNU, CLMRED
DIMENSION ZTMP(2), TMPMES(2), ZWND(2), WNDMES(2), SRCCOR(2), TRNCOR(3)
1, RECCOR(3), CPTS(2, 200), CNTRD(2), OBSCOR(2)
COMMON /IOUNIT/ IOIN, IOOUT, ISPTPF, LOUNIT, NDIRTU, NBSCAT
COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT,
1 FOGPRB, IPASCT, WNDVEL, WINDIR

************************************** SUBFILE 1 **************************************

DAVID DVORE
AERODYNE RESEARCH, INC.
(617)275-9400 X127

PURPOSE

DRTRAN CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES
THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,
BOMB CHARACTERISTICS, AND WAVELENGTH.

INPUT

NEWATM A LOGICAL VARIABLE WHICH IS .TRUE. IF NEW ATMOSPHERIC
CONDITIONS ARE TO BE SPECIFIED AND .FALSE. IF PREVIOUS
VALUES ARE TO BE ASSUMED. IF .TRUE. THEN NATMOS, ZTMP,
TMPMES, ZWND, WNDMES, HTHINV, AND THWND MUST BE
SPECIFIED.
NATMOS INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL CATEGORIES A TO F. IF WIND AND TEMPERATURE MEASUREMENTS ARE AVAILABLE AT TWO HEIGHTS THEN NATMOS SHOULD BE SPECIFIED 0.

ZTMP A SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS, MEASURED IN METERS, AT WHICH TEMPERATURE MEASUREMENTS ARE AVAILABLE. THE HEIGHTS SHOULD BE IN INCREASING ORDER. IF ONLY ONE IS AVAILABLE, THE PASQUILL CATEGORY MUST BE SPECIFIED IN NATMOS. VALID RANGE: 0.5-100. M.

TMPMES A SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO TEMPERATURE MEASUREMENTS IN DEGREES KELVIN TAKEN AT HEIGHTS ZTMP. VALID RANGE: 273.-315. K.

ZWND A SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS, MEASURED IN METERS, AT WHICH WIND MEASUREMENTS ARE AVAILABLE. THE HEIGHTS SHOULD BE IN INCREASING ORDER. IF ONLY ONE IS AVAILABLE, THE PASQUILL CATEGORY MUST BE SPECIFIED IN NATMOS. VALID RANGE: 0.5-100.M.

WNDMES A SINGLY DIMENSIONED ARRAY CONTAINING WIND SPEEDS MEASURED IN METERS/SEC, AT HEIGHTS ZWND. VALID RANGE: 0.1 - 20. M/S.

HTINV THE INVERSION HEIGHT MEASURED IN METERS. DEFAULTS TO 200 METERS IF A NUMBER LESS THAN 10.0 IS GIVEN. VALID RANGE: 10.0 - 200.0 M.

THWND THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES WITH THE POSITIVE X AXIS MEASURED IN DEGREES COUNTERCLOCKWISE. VALID RANGE: -360.0 - 360.0 DEGREES.

NEWSRC A LOGICAL VARIABLE WHICH IS .TRUE. IF A NEW SOURCE IS TO BE SPECIFIED AND .FALSE. IF FURTHER RESULTS ARE DESIRED FROM THE PRESENT SOURCE. IF .TRUE. THEN CHWT, NCHRG, DETDEF, NSOIL, DSOD, NWL, AND SRCCOR MUST BE SPECIFIED.

CHWT THE WEIGHT OF THE CHARGE IN LBS-TNT. VALID RANGE: 1.0 - 150.0 LBS-TNT.

NCHRG CHARGE TYPE INDEX WITH FOLLOWING VALUES

1 SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC, TIP ON GROUND
2 BARE CHARGE ON SURFACE

24
DEGREE TILTED TIP AT 0.3 METER DEPTH
5 DEGREE TILTED TIP AT 0.6 METER DEPTH
HORIZONTAL PROJECTILE ON SURFACE
DEFAULT VALUE IS 1 IF MCHRG IS NOT BETWEEN 1 AND 5.

DETDEP THE DEPTH OF DETONATION IN METERS.
VALID RANGE: 0.0 - 2.0 M.

NSOIL INTEGER INDEX OF SOIL TYPE. NSOIL IS
1 FOR EUROPEAN SOIL
2 FOR DESERT SOIL

DSOD DEPTH OF SOD IN METERS
VALID RANGE: 0.0 - 1.0 M.

WAVEI WAVELENGTH IN MICROMETERS. USED TO DETERMINE NWL.
VALID RANGES:

NWL INTEGER INDEX FOR WAVELENGTH
1 FOR 0.4 - 0.7 MICROMETER (VISIBLE)
2 FOR 0.8 - 1.1 MICROMETER
3 FOR 3.5 - 4.0 MICROMETER
4 FOR 8.5 - 12.0 MICROMETER
5 FOR 2100 - 3200 MICROMETER

SRCCOR A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND
Y COORDINATES OF THE DETONATION POSITION.
VALID RANGE: -9999.9 - 9999.99 M.

LOSTRN A LOGICAL VARIABLE WHICH IS TRUE. IF THE
TRANSMITTANCE ALONG A LINE-OF-SIGHT IS DESIRED
AND FALSE. IF NOT. IF TRUE. THEN TRNCOR AND RECCOR
MUST BE SPECIFIED.

TRNCOR A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE
COORDINATES OF THE TRANSMITTER. THE COORDINATE SYSTEM
MUST BE IN METERS. THE THIRD COORDINATE IS RESTRICTED
TO BE BETWEEN 1 AND 5 METERS (HEIGHT).
VALID RANGE: -9999.9 - 9999.99 M.

RECCOR A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE
COORDINATES OF THE RECEIVER. (METERS)
THE THIRD COORDINATE MUST BE THE SAME AS THE THIRD
COORDINATE OF TRNCOR.
VALID RANGE: -9999.9 - 9999.99 M.

EDGE A LOGICAL VARIABLE WHICH .TRUE. IF CLOUD DIMENSIONS
ARE TO BE CALCULATED AND .FALSE. IF NOT. IF .TRUE. THEN OBSCOR AND SPCHT MUST BE SPECIFIED.

OBSCOR  A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND Y COORDINATES, RESP., OF THE OBSERVER. (METERS)
VALID RANGE:  -9999.9 - 9999.99

SPCHT  A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED. MUST BE BETWEEN 1 AND 5 METERS.

NEWTIM  A LOGICAL VARIABLE WHICH IS .TRUE. IF ONE WISHES TO ADVANCE THE TIME AND FALSE IF MORE RESULTS ARE REQUIRED AT THE CURRENT TIME. IF .TRUE. THEN TIME MUST BE SPECIFIED.

TIME  TIME MEASURED IN SECONDS AFTER DETONATION. SUCCESSIVE CALLS OF DRTRAN MAY KEEP THE TIME UNCHANGED FROM PREVIOUS CALL (NEWTIM = .FALSE.) OR SPECIFY A NEW TIME GREATER THAN IN THE PREVIOUS CALL. WHEN A NEW SOURCE IS SPECIFIED (NEWSRC = .TRUE.) A NEW TIME MUST ALSO BE SPECIFIED (NEWTIM = .TRUE.) AND ONLY IN THIS CASE MAY A TIME LESS THAN THE PREVIOUS CALL BE SPECIFIED. VALID RANGE: 0.5 - 1000.0 SECONDS.

OUTPUT

TRNLOS  THE TRANSMITTANCE ALONG THE LINE-OF-SIGHT BETWEEN THE TRANSMITTER AND THE RECEIVER.

IERR  INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR OCCURS AND 0 OTHERWISE


HEIGHT  THE HEIGHT OF THE CLOUD IN METERS.

CENWTH  THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID HEIGHT

SPCWH  THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED HEIGHT
NCPTS  THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE CLOUD.


NERR  AN INTEGER ERROR CODE WITH VALUES
-1 CPTS NOT LARGE ENOUGH FOR THE NUMBER OF POINTS CALCULATED
 0 NO ERRORS
 1 NO CONTOUR FOUND BY THE CONTOUR TRACING ROUTINE. THIS MEANS THAT THE CLOUD HAS DISSIPATED.
 2 CONVERGENCE WAS NOT ACHIEVED IN SEARCHING FOR A POINT ON THE CONTOUR. MAY REFER TO A DISCONTINUITY IN THE CLOUD EDGE.
 3 CONTOUR TOO SMALL TO TRACE I.E. LESS THAN A METER IN DIAMETER.
 4 LOSTRN AND EDGE WERE BOTH SPECIFIED FALSE SO NO RESULTS WERE CALCULATED
 5 NEWSRC WAS FALSE WHEN NEWATM WAS TRUE. A NEW ATMOSPHERE CANNOT BE SPECIFIED DURING THE EVOLUTION OF ONE DUST CLOUD.
 6 THE SPECIFIED HEIGHT FOR THE WIDTH OF THE CLOUD WAS ABOVE THE CLOUD
 7 THE CALCULATION OF THE ATMOSPHERIC PARAMETERS DID NOT CONVERGE. THE MEASURED DATA MAY BE INCONSISTENT.

SUBROUTINES CALLED

ATMCAL  ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES NECESSARY PARAMETERS IN COMMON /WNDPRM/ AND /IMPPRM/.

SOURCE  ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL VALUES IN COMMON /PRTINF/ AND /BUOYCL/.

RISE  GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER TIME.
CONBYN converts cloud dimensions as specified by variables used in RISE to output variables describing cloud dimensions.

WIND computes wind velocity as a function of height.

SCONF determines a set of points defining the observable contour of the cloud as viewed from a distance at a given angle to the wind velocity.

PTSPEC uses the set of coordinate points determined by SCONF to calculate output variables describing cloud dimensions.

Functions called:

CWIND computes the optically weighted concentration for times after the transition from buoyant rise to wind blown.

*IERR=0 CLMRED=.FALSE.*
WRITE(IOOUT,900)

900 FORMAT(49H DIRTRAN-I DUST CLOUD INFRARED TRANSMISSION , 1 12H CALCULATION //, 2 46H ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED.//)

100 READ(IOIN,701)NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM
IF(.NOT.NEWATM) GO TO 10
IF(ICLMAT.EQ.1.AND..NOT.CLMRED) GO TO 5
READ(IOIN,702)NATMOS,(ZTMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,2),ZINV,THWND
GO TO 6

IPASCT PASQUILL CATEGORY

WNDVEL wind velocity in m/s measured at 2 meters above ground

WINDIR wind direction in degrees clockwise from true north

TEMP temperature in degrees C measured at 2 meters height

NATMOS=IPASCT
ZWND(1)=2.
ZTMP(1)=2.
WNDMES(1)=WNDVEL
TMPMES(1)=TEMP+273.0

NOTE THAT POSITIVE X AXIS IS ASSUMED TO BE EAST.

NOTE THAT WINDIR IS THE DIRECTION OF ORIGIN OF THE WIND.
C

THWND=270. -WINDDIR
ZINV=150.0
CLMRED=. TRUE.
6 CONTINUE
950 FORMAT(1X)
WRITE(IOUT, 950)
WRITE(IOOUT, 951) NATMOS
N=1
IF(NATMOS.EQ.0)N=2
WRITE(IOOUT, 952) (ZTMP(I), TMPMES(I), ZWND(I), WNDMES(I), I=1, N)
WRITE(IOOUT, 953) THWND
951 FORMAT(30H PASQUILL CATEGORY , 12)
952 FORMAT(8H HT , F7.2, 7H TEMP , F7.2, 7H WIND , 1 F7.2)
953 FORMAT(22H WIND DIRECTION , F7.2)
10 CONTINUE
IF(. NOT. NEWSRC) GO TO 20
READ(IOIN, 703) NSOIL, NCHRG, CHWT, DETDEP, DSOD, SRCCOR(1), 1 SRCCOR(2)
WRITE(IOOUT, 950)
WRITE(IOOUT, 955) NSOIL
WRITE(IOOUT, 956) NCHRG
WRITE(IOOUT, 957) CHWT
WRITE(IOOUT, 958) DETDEP
WRITE(IOOUT, 959) DSOD
WRITE(IOOUT, 959) (SRCCOR(I), I=1, 2)
WRITE(IOOUT, 950)
954 FORMAT(15H CHARGE INDEX , 12)
955 FORMAT(15H SOIL INDEX , 12)
964 FORMAT(30H WAVELENGTH , F7.2, 13H MICROMETERS )
956 FORMAT(30H WEIGHT OF CHARGE , F7.2)
957 FORMAT(30H DETONATION DEPTH , F7.2)
958 FORMAT(30H DEPTH OF SOD , F7.2)
959 FORMAT(30H SOURCE COORDINATES , 2F10.2)
20 CONTINUE
IF(. NOT. LOSTRN) GO TO 30
READ(IOIN, 704) (TRNCOR(I), I=1, 3), (RECCOR(J), J=1, 3)
IF(WAVE1.LT.0.4) GO TO 29
IF(WAVE1.GT.0.7) GO TO 21
NWL=1
GO TO 30
21 IF(WAVE1.LT.0.8) GO TO 29
IF(WAVE1.GT.1.1) GO TO 22
NWL=2
GO TO 30
22 IF(WAVE1.LT.3.5) GO TO 29
IF(WAVE1.GT.4.0) GO TO 23
NWL=3
GO TO 30
23 IF(WAVE1.LT.8.5)GO TO 29
 IF(WAVE1.GT.12.0)GO TO 24
NWL=4
GO TO 30
24 IF(WAVE1.LT.2100)GO TO 29
 IF(WAVE1.GT.3200)GO TO 29
NWL=5
GO TO 30
29 WRITE(IOOUT,25)
25 FORMAT(42H *** DIRTRAN-I ERROR - WAVE1 OUT OF RANGE )
 IERR=1
GO TO 999
30 IF (EDGE) READ (IOIN,704) (OBSCOR(IO),IO-1,2),SPCHT
 TIMTST=TIME
 IF (NEWTIM) READ (IOIN,704) TIME
 IF (.NOT. NEWSRC.AND. TIME.LT.TIMTST)GO TO 997
WRITE(IOOUT,960)TIME
960 FORMAT(//30H TIME AFTER BLAST ,F7.2)
NERR=0
 CALL DUSTCL (NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,ZINV,THWkd,
 1 NEWSRC,CHWT,NCHRG,DETDEF,NSOIL,DSOD,NWL,SRCCOR,
 2 LOSTRN,TRNCR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,TIME,
 3 TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWH,NCPTS,CPTS,NERR)
C IF (NERR.EQ.0)GO TO 600
C WRITE(IOOUT,969)NERR
C 969 FORMAT(20H ***** ERROR NUMBER ,12)
C GO TO 800
600 IF(.NOT.LOSTRN)GO TO 700
 WRITE(IOOUT,950)
 WRITE(IOOUT,964)WAVE1
 WRITE(IOOUT,961) (TRNCR(I),I=1,3)
 WRITE(IOOUT,962) (RECCOR(I),I=1,3)
961 FORMAT(30H TRANSMITTER COORDINATES ,3FI0.2)
962 FORMAT(30H RECEIVER COORDINATES ,3FI0.2)
 WRITE(IOOUT,970)TRNLOS
970 FORMAT(40H TRANSMITTANCE ALONG THE LINE-OF-SIGHT ,E10.3)
700 IF(.NOT.EDGE)GO TO 800
 WRITE(IOOUT,950)
 WRITE(IOOUT,963) (OBSCOR(I),I=1,2)
963 FORMAT(30H OBSERVER COORDINATES ,2F10.2)
 WRITE(IOOUT,971)HEIGHT
971 FORMAT(20H THE HEIGHT OF THE CLOUD IS ,10X,F10.2,7H METERS)
C 990 FORMAT((1X,2F10.2))
C WRITE(IOOUT,972) (CNTRD(IO),IO=1,2)
FORMAT(30H THE CENTROID COORDINATES ARE ,8X,2F10.2)
WRITE(IOOUT,973)CENWTH
FORMAT(30H THE WIDTH AT THE CENTROID IS ,8XF10.2, 7H METERS)
WRITE(IOOUT,974)SPCWT,SPCWTH
FORMAT(14H THE WIDTH AT ,F8.2,11H METERS IS ,5X,F10.2,7H METERS)
WRITE(IOOUT,975)NCPTS
FORMAT(1X,I3,37H CONTOUR POINTS HAVE BEEN DETERMINED)
WRITE(IOOUT,708)((CPTS(I0,IPT),10.1,2),IPT-1,NCPTS)
READ(IOIN,701)CNTNU
IF(CNTNU)GO TO 100
FORMAT (5L1)
FORMAT (11,1X,10F7.2)
FORMAT (2(11,1X),5F7.2)
FORMAT (6F7.2)
FORMAT (12)
FORMAT ((1X,2(F10.3,2X)))
FORMAT (2(I2, IX),2F7.2)
GO TO 999
WRITE(IOOUT,998)
FORMAT(54H *** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING ORDER)
IERR=1
RETURN
END

CONTROLLING ROUTINE FOR DIRTRAN-I CODE

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

SUBROUTINE DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,HTEMV,
1 THWND,NEWSTR,CHWT,NCHRG,DETDY,NSOIL,NSD,NSL,SRCCOR,
2 LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCWH,NCPTS,CPTS,NERR)
IMPLICIT INTEGER*4 (I-N)
LOGICAL NEWATM,NEWSTR,LOSTRN,EDGE,NEWTIM,HORIZ,ERR
DIMENSION ZTMP(2) ,TMPMES(2 ),ZWND(2),WNDMES(2),SRCCOR(2),TRNCOR(3)
1 ,RECCOR(3),CPTS(2,200),ORIG(2),TRNFRM(2,2),TRN(3),REC(3)
2 ,PNT(3),DELS(3),CNTRD(2),OBSCOR(2),DIR(2),SCRN(2)
DIMENSION OWF(5,2)
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
COMMON /MODE/ HORIZ
COMMON /WNDPRM/ DXZO,DYXO,DZO,U0,UM,DN,ZINV
COMMON /CLOCK/ FTIME,TWIND
COMMON /SEPRTN/ SEPI(2),SEP2(2),PRSEP1,PRSEP2,NUM1,NUM2
COMMON /IUNIT/ IOIN,IOOUT,ISPTPF,LUNIT,NDIRTU,NBSCT
EXTERNAL FUNCT
DATA PI/3.14159267/,VISEXT,ZMIN,RES,TANT/.1,0.,1.,1./
DATA ONEM/-1./,RTPI/1.77245/
DATA OWF/1.,93.,52.,44.,7.E-4,1.,93.,66.,52,7.,E-4/
*******************************************************************************
DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE, BOMB CHARACTERISTICS, AND WAVELENGTH.

SEE COMMENTS IN DRTRAN FOR DETAILS.

IF (LOSTRN. OR. EDGE) GO TO 100
  NERR = 4
  GO TO 999
100 IF (.NOT. NEWATH) GO TO 200
    IF (NEWSRC) GO TO 150
    NERR = 5
    GO TO 999
150 CONTINUE
  ZINV = HINV
  CALL ATMCAL (NATHOS, ZTMP, TMPMES, ZWND, WNDMES, ERR)
  IF (.NOT. ERR) GO TO 155
  NERR = 7
  GO TO 999
155 CONTINUE

COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN THE WIND DIRECTION.

THETAX = THWND * PI / 180.
TRNFRM(1, 1) = COS (THETAX)
TRNFRM(2, 2) = TRNFRM(1, 1)
TRNFRM(1, 2) = SIN (THETAX)
TRNFRM(2, 1) = -TRNFRM(1, 2)

200 CONTINUE
  IF (.NOT. NEWSRC) GO TO 300
  TWIND = 1E5
  TPRES = 0.
  DEL = .001
  CALL SOURCE (CHWT, NCHRNG, DETDEP, NSOIL, DSOD)
  DO 250 I = 1, 2
    ORIG(I) = SRCCOR(I)
  250 CONTINUE
  OVERLAP = SPACNG / 2.
  SEP1(1) = SPACNG * COS (THARRY - THETAX)
SEP1(2)=SPACNG*SIN(THARRY-THETAX)
CALL PERP(SEP1,SEP2)
300 CONTINUE
IF(.NOT.LOSTRN) GO TO 400
C
C CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT
C DETONATION POINT AND X AXIS IN WIND DIRECTION.
C
TRN(3)=TRNCOR(3)
REC(3)=RECCOR(3)
DO 320 I=1,2
TRN(I)=0.
REC(I)=0.
DO 310 J=1,2
TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIC(J))
REC(I)=REC(I)+TRNFRII(I,J)*(RECCOR(J)-ORIC(J))
310 CONTINUE
320 CONTINUE
400 CONTINUE
IF(.NOT.EDGE) GO TO 500
C
C COMPUTE SIN AND COS OF THE ANGLE BETWEEN THE OBSERVERS VIEWING
C ANGLE AND THE LOCAL POSITIVE X AXIS WHICH IS IN THE WIND
C DIRECTION.
C
CALL VSUM(ORIG,OBSCOR,ONEM,DIG)
CALL UNIT(DIR,DIR,RANGE)
COSTH=0.
SINTH=0.
DO 410 J=1,2
COSTH=COSTH+TRNFRM(1,J)*DIR(J)
SINTH=SINTH+TRNFRM(2,J)*DIR(J)
410 CONTINUE
SINTH2=SINTH*SINTH
COSTH2=COSTH*COSTH2
SCRN(1)=SINTH
SCRN(2)=-COSTH
500 CONTINUE
IF(NEWTIM) CALL RISE(TPRES,TIME,DEL)
600 IF(.NOT.EDGE) GO TO 650
FTIME=TIME
CALL CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS,
1 NERR)
IF(.NOT.ERR) GO TO 650
ERR=6
GO TO 999
650 CONTINUE
IF(.NOT.LOSTRN) GO TO 999
DETERMINE THE OPTICALLY WEIGHTED CL VALUE
ALONG THE LINE CONNECTING THE TRANSMITTER AND RECEIVER

CALL VSUM(REC,TRN,ONEM,DIR)
CALL UNIT(DIR,DIM,RANGE)
COSTH=DIR(1)
SINTH=DIR(2)
SINTH2=SINTH*SINTH
COSTH2=COSTH**2
SCRN(1)=SINTH
SCRN(2)=-COSTH
X=DOTPRD(SCRN,TRN)
HORIZ=.TRUE.
TRNLOS=EXP(-CWIND(X,Y,TRN(3),TIME)*OWF(NWL,NSOIL))
999 RETURN
END

CALCULATION OF ATMOSPHERIC PARAMETERS FOR DIRTRAN-I CODE

SUBROUTINE ATMCAL(NATM,ZTTMES,ZU,UMES, ERR)
IMPLICIT INTEGER*4 (I-N)
REAL M,N
LOGICAL ERR
DIMENSION ZT(2),TMES(2),ZU(2),UMES(2),ZLO(6)
COMMON /WNDPRM/ DXZO,DYXO,DZO,UO,M,N,ZINV
COMMON /TMPPRM/ TO, TI, TM
COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF, LOUNIT,NDIRTU,NBSCAT
DATA ZLO/-2.5,-4.5,-13.5,1000.,55.,20./

PURPOSE

TO FIT THE BEST POWER-LAW PROFILES OF WINDSPEED,
DIFFUSIVITY, AND TEMPERATURE CONSISTENT WITH KNOWN RELATIONS
GOVERNING THE CONSTANT SHEAR STRESS LAYER TO GIVEN
MEASUREMENTS AT TWO HEIGHTS.

INPUTS

NATM INTEGER WHICH IS 0 IF WINDSPEED AND TEMPERATURE
ARE AVAILABLE AT TWO HEIGHTS AND EQUAL TO THE
PASQUILL CATEGORY OTHERWISE.
ZT SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS (IN METERS) AT WHICH TEMPERATURES WILL BE GIVEN. MUST BE IN ASCENDING ORDER.

TMES SINGLY DIMENSIONED ARRAY CONTAINING THE TWO TEMPERATURE MEASUREMENTS IN DEGREES KELVIN AT HEIGHTS ZT.

ZU SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO HEIGHTS (METERS) AT WHICH WIND SPEEDS WILL BE GIVEN. MUST BE IN ASCENDING ORDER.

UMES SINGLY DIMENSIONED ARRAY CONTAINING THE ONE OR TWO WIND SPEED MEASUREMENTS (M/S) AT HEIGHTS UMES.

ZINV INVERSION HEIGHT IN METERS.

OUTPUTS

ERR A LOGICAL WHICH IS TRUE IF AN ERROR IS INCURRED DURING THE CALCULATION.

DXZO THE RATIO OF THE DIFFUSIVITY IN THE X DIRECTION TO THE DIFFUSIVITY IN THE Z DIRECTION. RETURNED IN COMMON /WNDPRM/.

DYXO THE RATIO OF THE DIFFUSIVITY IN THE Y DIRECTION TO THE DIFFUSIVITY IN THE X DIRECTION. RETURNED IN COMMON /WNDPRM/.

DZO THE COEFFICIENT OF Z**N IN THE VERTICAL PROFILE OF VERTICAL DIFFUSIVITY. RETURNED IN COMMON /WNDPRM/.

UO THE COEFFICIENT OF Z**M IN THE VERTICAL PROFILE OF HORIZONTAL WIND SPEED. RETURNED IN COMMON /WNDPRM/.

M THE EXPONENT OF Z IN THE HORIZONTAL WIND SPEED PROFILE. RETURNED IN COMMON /WNDPRM/.

N THE EXPONENT OF Z IN THE VERTICAL DIFFUSIVITY PROFILE. RETURNED IN COMMON /WNDPRM/.

TO, TI, TM CONSTANTS FOR THE TEMPERATURE PROFILE SUCH THAT
CALLED FROM DUSTCL

NEEDED FUNCTIONS AND SUBROUTINES

TMPCAL    CALCULATES SCALED TEMPERATURE PROFILES
WNDCAL    CALCULATES SCALED WIND SPEED PROFILES

ERR= .FALSE.

DELTH IS THE DIFFERENCE IN POTENTIAL TEMPERATURE BETWEEN THE
TWO HEIGHTS WHERE TEMPERATURE IS GIVEN.

ZO=0.02
IF(NATM.EQ.0)GO TO 100

ASSIGN ATMOSPHERIC PROFILE ACCORDING TO SPECIFIED PASQUILL
CATEGORY

ZO  FRICTION HEIGHT
ZL  MOMIN-OBUKOV LENGTH
USTAR THE FRICTION VELOCITY
TSTAR THE SCALING TEMPERATURE

ZL=ZLO(NATM)
NP=IFIX(SIGN(1.,ZL))
USTAR=UMES(1)/WNDCAL(ZO,ZL,ZU(I))
TSTAR=TMES(1)*USTAR**2/1.568/ZL
IF(NATM-4)200,300,210
100 CONTINUE

USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE
TO MATCH DATA AT TWO HEIGHTS

DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))
NP=SIGN(1.,DELTH)
DELU=UMES(2)-UMES(1)
ZULOG=ALOG(ZU(2)/ZU(1))
ZTLOG=ALOG(ZT(2)/ZT(1))
USTAR=(UMES(2)-UMES(1))/ZULOG
TSTAR=DELTH/ZTLOG
ZL=.638*TMES(1)*USTAR**2/TSTAR
IF(ABS(ZL).GE.1000.)GO TO 300
DO 110 IER=1,100
USTAR=DELU/(WNDCAL(ZO,ZL,ZU(2))-WNDCAL(ZO,ZL,ZU(1)))
TSTAR=DELTH/(TMPCAL(ZO,ZL,ZT(2))-TMPCAL(ZO,ZL,ZT(1)))
ZLP=ZL
ZL=.638*TMES(I)*USTAR*2/TSTAR
IF(ABS((ZL-ZLP)/ZLP).LT..01)GO TO 120
110 CONTINUE
ERR=.TRUE.
GO TO 999
120 CONTINUE
ZO=EXP (.4*(WNDCAL(ZO,ZL,ZU(I))-UMES(1)/USTAR))*ZO
IF(NF).200,300,210
200 CONTINUE
C C UNSTABLE ATMOSPHERE
C DZ0=3.15*.4*USTAR/ABS(ZL)**(1./3.)
DZ0=2.
M=1./7.
N=4./3.
GO TO 430
210 CONTINUE
C C STABLE ATMOSPHERE
C DXZ0=6.
M=1./7.
N=1.
DZ0=.4*USTAR/(1.+47./ZL)
GO TO 430
300 CONTINUE
C C NEUTRAL ATMOSPHERE
C DZ0=.4*USTAR
DXZ0=5.
NP=0
N=1.
M=1./7.
430 CONTINUE
C C COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES
C IF(ZINV.LE.10.)ZINV=200.
UO=UMES(1)/ZU(1)**M
DXYO=1.
IF(NATM.EQ.0)UO=(UO+UMES(2)/ZU(2)**M)/2.
IF(NATM.EQ.0)GO TO 450
ZT(2)=2.*ZT(1)
DELT=STAR*(TMPCAL(ZO,ZL,ZT(2))-TMPCAL(ZO,ZL,ZT(1)))
T1-DELT/(ZT(2)**M-ZT(1)**M)
TM=H
TO-TMES(I)-T1*ZT(I)**M
DZ0=.4*USTAR
N-1.
999 RETURN
END
C
C ******************* SUBFILE 4 *******************
C
FUNCTION WNDCAL(ZO,ZL,Z)
IMPLICIT INTEGER*4 (I-N)
C
PURPOSE
C
TO CALCULATE THE WIND SPEED, U/U*, SCALED BY THE FRICTION
C VELOCITY FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV LENGTH AT A
C SPECIFIED HEIGHT.
C
INPUTS
C ZO THE FRICTION HEIGHT IN METERS.
C ZL THE MONIN-OBUKHOV LENGTH IN METERS.
C Z THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C IN METERS
C
RETURNS VELOCITY SCALED BY FRICTION VELOCITY
C
CALLED BY ATMICAL
C
LOGICAL LOW
PSIM(S)=(ALOG((S+1.)*2*(S*S+1)/8.)+2.*(ATAN(1.)-ATAN(S)))
PSIMS(Z)=7.*Z
PHIM(Z)=(1.-16.*Z)**(-.25)
C
PHIM THE SHEAR OF MOMENTUM
C PSIM THE UNIVERSAL FUNCTION FOR THE DEVIATION FROM
LOGARITHMIC WIND VELOCITY BOUNDARY LAYER PROFILE IN AN
UNSTABLE ATMOSPHERE
THE SAME AS PSIM FOR A STABLE ATMOSPHERE

IF(ABS(ZL).LE.1.E3)GO TO 100
WNDCAL=ALOG(1.+Z/Z0)
GO TO 999
100 CONTINUE
P=SIGN(1.,ZL)
LOW=.TRUE.
S=Z/ZL
IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
S=AMIN(S,1.5)
S=AMAX(S,-2.)
LOW=.FALSE.
10 CONTINUE
IF(P)120,130,130
120 S=1./PHIM(S)
PSI=PSIM(S)
GO TO 52
130 CONTINUE
PSI=PSIMS(S)
52 CONTINUE
WNDCAL=ALOG(1.+Z/Z0)-PSI
IF(LOW)GO TO 999
IF(P)53,53,54
53 WNDCAL=WNDCAL+.75-.95*(-ZL/Z)**(1./3.)
GO TO 999
54 WNDCAL=WNDCAL-15.+10.*Z/ZL
999 WNDCAL=WNDCAL/.4
RETURN
END

FUNCTION TMPCAL(Z0,ZL,Z)
IMPLICIT INTEGER*4 (I-N)

PURPOSE
TO CALCULATE THE POTENTIAL TEMPERATURE SCALED BY THE SCALE
TEMPERATURE, T*, FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV
LENGTH AT A SPECIFIED HEIGHT.
INPUTS

Z0  THE FRICTION HEIGHT IN METERS.
ZL  THE MONIN-OBUKHOV LENGTH IN METERS.
Z   THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED IN METERS

RETURNS SCALED TEMPERATURE

CALLED BY ATMCAL

**************************************************************************
LOGICAL LOW

PHIM(Z) = (1.16.*Z)**(-.25)
PSIH(S) = -2.*ALOG((S*S+1.)/2.)
PSIHS(Z) = -11.*Z

PHIM  THE SHEAR OF MOMENTUM
PSIH  THE UNIVERSAL FUNCTION FOR DEVIATION FROM LOGARITHMIC
      POTENTIAL TEMPERATURE PROFILE IN THE BOUNDARY LAYER
PHIHS THE SAME AS PHIH EXCEPT FOR STABLE ATMOSPHERE

IF(ABS(ZL).LE.1.E3)GO TO 100
TMPCAL=ALOG(1.+Z/Z0)
GO TO 999

100 CONTINUE
P=SIGN(1.,ZL)
LOW=.TRUE.
S=Z/ZL
IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
S=AMIN1(S,1.5)
S=AMAX1(S,-2.)
LOW=.FALSE.

10 CONTINUE
IF(P)120,130,130

120 S=1./PHIM(S)
PSI=PSIH(S)
GO TO 52

130 CONTINUE
PSI=PSIHS(S)

52 CONTINUE
TMPCAL=ALOG(1.+Z/Z0)-PSI
IF(LOW)GO TO 999
IF(P)53,53,54
SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD)
IMPLICIT INTEGER*4 (I-N)

PURPOSE
TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE
DIRTRAN-I CODE

INPUT
W    THE WEIGHT OF THE CHARGE IN LBS. TNT
DD   DETONATION DEPTH IN METERS
NSOIL INTEGER SOIL INDEX
DSOD DEPTH OF SOD IN METERS

OUTPUT (RETURNED IN COMMON /PRTINF/ AND /BUOYCL/)
RO   INITIAL CLOUD RADIUS IN METERS
VGRAV SINGLY DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED
AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN
THE PARTICLE DISTRIBUTION (METERS/SEC)
NPRTS THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE
PARTICLE SIZE SPECTRUM
RSRH THE INITIAL RADIUS OF THE CLOUD IN METERS
DELT THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD
AND SURROUNDINGS (DEGREES KELVIN)
VXSHP THE INITIAL HORIZONTAL VELOCITY OF THE CLOUD (M/S)
VZSHP THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)
XCMSPH INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)
ZCMSPH INITIAL HEIGHT OF THE CLOUD (METERS)
RISTIM TIME LAPSED SINCE DETONATION IN SECONDS

CALLED BY DUSTCL
LOGICAL HORIZ
DIMENSION CR(5,4),CD(5,4),OWML(3,4),OWSV(3,4),PRTTN(4)
DIMENSION S(3),BURHTR(5),WTRAT(5)
COMMON/PRTINF/ RO, VGRAV(3), NPRTS
COMMON/BUOYCL/ RSPH, DELT, VXSPH, VZSPH, XCMSPH, ZCMSPH,
* SPHNS(3), RISTIM
COMMON/TMPRMR/ TO, T1, TM
COMMON/WNDPRM/ DXZ0, DYX0, DZ0, U0, UM, DN, ZINV
COMMON/BURST/ ACCEL, TBURST
COMMON/GEOM/COSTH2, SINTH, SINTH2, VISEX, RTRI
COMMON/MODE/ HORIZ
COMMON/DISCS/NDSCS, TDSC(10), XDSC(10), ZDSC(10), R2DSC(10),
1 QDSC(10,3)
C
C CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE
DATA CR/.483, -.328, .0319, .0536, .0 .386,-.291, .0543, .057, 1.1*0./
C CD IS THE CRATER DIAMETER INDEXED BY COEFFICIENT AND SOIL TYPE
DATA CD/.261, -.453, .0681, .224, .055, .198, -.355, .0562, .215, .0586,
1 10*0./
C OWML IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY
C BIN SIZE AND SOIL TYPE
DATA OWML/1.3E4, 2*0., 2.61E4, 8*0./
C OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC)
C INDEXED BY BIN SIZE AND SOIL TYPE
DATA OWSV/12*0./
C PRTTN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE
DATA PRTTN/4*.9/
C BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT
C IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD
DATA BURHTR/0., .4, .2, .4, .3., /, WTRAT/.6, .1, .8, 1., .6/
C
C
C RISTIM = 0.
XCMSPH = 0.
NPRTS = 1
W3 = (W* WTRAT (NCHR)) **.333333
RO = 1.535*W3
TAMB = TO + T1*RO**TM
DELT = .57*TAMB
RSPH = RO
ZCMSPH = RO
VXSPH = U0*ZCMSPH*UM
BURHT = BURHTR (NCHR) * RO
BURVZ = 1.3*SQRT (RO)
TBURST = .15*RO
VZSPH = 2.*BURHT/TBURST-BURVZ
ACCEL = (BURVZ-VZSPH)/TBURST

42
CLAM=DD/W3

C CALCULATE CRATER RADIUS AND DEPTH

RC=CR(1,NSOIL)
DC=CD(1,NSOIL)
IF (CLAM.LT.1.E-30) GO TO 98
TERM=1.
DO 100 I=2,5
TERM=TERM*CLAM
RC=RC + CR(I,NSOIL)*TERM
DC=DC + CD(I,NSOIL)*TERM
100 CONTINUE
98 CONTINUE
RC=RC*W3
DC=DC* (W*WTRAT(NCHRC))**.3

C GET CRATER VOLUME

VC=(3.1415926/3.) *(RC/DC)**2 *(DC - DSOD)**3

C CALCULATE OPTICALLY WEIGHTED PARAMETERS

NDSCS=MAX0(5,IFIX(5.*W3/2.4))
DO 101 L=1,NPRTS
S(L)=OWML(L,NSOIL) * VC
VGRAV(L)=WSV(L,NSOIL)
SPHNS(L)=PRTTN(NSOIL) * S(L)
QDSC(I,L)=(1.-PRTTN(NSOIL)) * S(L)/FLOAT(NDSCS)
101 CONTINUE
DELH=2.*RO/FLOAT(NDSCS)
Z=-DELH/2.
DO 200 I=1,NDSCS
Z=Z+DELH
ZDSC(I)=Z
DO 201 J=1,NPRTS
QDSC(I,J)=QDSC(1,J)
201 CONTINUE
CON=ALOG(QDSC(1,1))/VISEXT/DELH/(2.*RO)/3.14159)
IF(CON.GT.1.)GO TO 210
D=1.
GO TO 230
210 D=CON
DO 220 IT=1,5
D=(CON-1.+ALOG(D))*D/(D-1)
220 CONTINUE
230 R2DSC(I)=4.*RO*RO/D
TDSC(I)=-DELH*DELH/D/(DZO*Z*DN)/4.
SIGZ=DELH*DELD/H
C WRITE(I,777)SIGZ
XDSC(I)=UOZ**UM*TDSC(I)
C WRITE(I,777)TDSC(I),XDSC(I),ZDSC(I),R2DSC(I),QDSC(I,1)
777 FORMAT(1X,5(1PE10.3))
200 CONTINUE
RETURN
END

C WIND DISPERSION OF DUST CLOUD FOR DIRTRAN-I CODE

C *************************************************
C SUBFILE 7 *****************************************
C
FUNCTION FUNCT(X,Z)
IMPLICIT INTEGER*4 (I-N)
COMMON /CLOCK/ TIME,TWIND
C *************************************************

PURPOSE

TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING
ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE.

INPUT

X THE HORIZONTAL COORDINATE IN METERS
Z THE VERTICAL COORDINATE IN METERS

OUTPUT

RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE FOR THE
LINE-OF-SIGHT SPECIFIED BY X,Z

FUNCTIONS CALLED

CWIND

CALLED BY GFUN, CLIMB, GRAD2

Y=0.
C WRITE(1,1)X,Z,TIME
1 FORMAT(1X,3(1PE10.2))
IF(Z.LE.0.)GO TO 100
EXT=CWIND(X,Y,Z,TIME)
IF(EXT.LE.1.E-30)GO TO 100
FUNCT=ALOG(EXT)
GO TO 999
C
C SUBFILE 8
C
C FUNCTION CWIND(X,Y,Z,T)
C IMPLICIT INTEGER*4 (I-N)
C REAL M,N
C LOGICAL HORIZ
C DIMENSION REF(2),REFO(2)
C COMMON/PRTINF/R0,VGRAV(3),NPRTS
C COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
C COMMON /MODE/HORIZ
C COMMON /WNDPRM/DXZO,DYXO,DZO,UO,M,N,ZINV
C COMMON/DISCS/NDSCS,TDSC(IO),XDSC(IO),ZDSC(IO),R2DSC(IO),
C QDSC(IO,3)
C COMMON /SEPRTN/SEPI(2),SEP2(2),PRSEPI,PRSEP2,NUM1,NUM2
C
C PURPOSE
C TO COMPUTE THE CONCENTRATION AT A POINT OR INTEGRATED ALONG
C A HORIZONTAL LINE
C
C INPUT
C X,Y,Z COORDINATES IN METERS. IF LINE INTEGRAL IS DESIRED,
C Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.
C T THE TIME IN SECONDS AFTER DETONATION
C
C OUTPUT
C RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
C THE LINE INTEGRAL OF CONCENTRATION IF HORIZ IS .TRUE.
C
C SUBROUTINES CALLED
C MOMENT COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM
C TABLE OF HIGHER ORDER MOMENTS.
C CALLED BY FUNCT,DUSTCL
**COMMON /PRTINF/**

RO    INITIAL RADIUS OF THE CLOUD IN METERS
VGRAV SINGLY DIMENSIONED ARRAY. VGRAV(I) IS THE OPTICALLY
       WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE
       I SIZE RANGE
NPRTS THE NUMBER OF PARTICLE SIZE RANGES

**COMMON /DISCS/**

NDSCS THE NUMBER OF DISC SOURCES
TDSC SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE
       OF THE DISC SOURCES
XDSC SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE
       OF THE CENTER OF THE DISC SOURCES
ZDSC SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE
       OF THE CENTER OF THE DISC SOURCES
R2DSC SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE
       RADII OF THE DISC SOURCES
QDSC DOUBLY DIMENSIONED ARRAY. QDSC(I,J) IS THE NUMBER OF
       PARTICLES OF THE J SIZE RANGE IN THE I DISC.

SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE
OPTICALLY WEIGHTED CONCENTRATION AT X,Y,Z,T

CWIND=0.
DO 211 I=1,NDSCS
TOF=T-TDSC(I)
REFO(1)=XDSC(I)
REFO(2)=0.
H=R2DSC(I)
QDSC=H2
IF(HORIZ) REFO(1)=REFO(I)*SINTH
DO 210 J=1,NPRTS

**DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z**

CALL MOMENT(VGRAV(J),Z,TOF,Q,XBAR,SIGW2,SIGP2)
IF(Q.CT.1.E-20) GO TO 113

IF Q IS TOO SMALL, ITS CONTRIBUTION IS IGNORED

CWIND=0.
GO TO 210
** *** COMPUTE CONCENTRATION AT (X,Y,Z,T) ***

ARG = -(X-REF0(1)-XBAR)**2/RX2
IF (ABS' RG).GT.30.) GO TO 150
CWNDSC (Q/RTPI/SQRT(RX2)) * EXP(ARG)
ARG = -(Y-REF0(2))**2/RY2
IF (ABS(ARG).GT.30.) GO TO 150
CY = EXP(ARG) / RTPI / SQRT(RY2)
CWNDSC = QDSC(1, J) * CWNDSC * CY
GO TO 150

** *** COMPUTE CONCENTRATION ALONG LINE-OF-SIGHT SPECIFIED BY X,Z ***

120 CONTINUE
REF(1) = REF0(1)
reff2 = RX2*SINTH2+RY2*COSTH2
ARG = -(X-REF(1)-XBAR*SINTH)**2/REFF2
IF (ABS(ARG).GT.30.) GO TO 150
CWNDSC = EXP(ARG) / SQRT(REFF2) / RTP1
CWNDSC = CWNDSC * Q * QDSC(1, J)
150 CONTINUE
CWND = CWIND + CWNDSC
210 CONTINUE
211 CONTINUE
RETURN
END

** SUBFILE 9 ***********************************************

SUBROUTINE CONVRT(T)
IMPLICIT INTEGER*4 (I-N)
REAL M,N
LOGICAL HORIZ
COMMON/BUOYCL/RSPH,DELT,VXS(1),VZSPH,XCMSPH,ZCMSPH,
1 SPHNS(3),RISTIM
COMMON/PRTINF/ RO,VGRAV(1),NPRTS
COMMON /WNDPRM/DXZO,DYXO,DZO,UC,M,N,ZINV
COMMON /GEOH/COSTH2,SINTH,SINTH2,VISEXT,RTPI
COMMON /MODE/ HORIZ
COMMON/DISC/NDSCE,TDSC(10),XDS(10),ZDSC(10),R2DSC(10),
1 QDSC(10, 3)
COMMON /CLOCK/ TIME,TWIND
C ************************************************************
PURPOSE

TO CONVERT THE CURRENT BUOYANT CLOUD INTO DISC SOURCES FOR USE BY THE WIND DISPERSION MODEL

INPUT

T THE TIME IN SECONDS AFTER DETONATION

OUTPUT

SUBROUTINES CALLED

MOMENT COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM TABLE OF HIGHER ORDER MOMENTS.

CALLED BY FUNCT,DUSTCL

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

COMMON /BUOYCL/

RSPH RADIUS OF THE SPHERICAL BUOYANT CLOUD IN METERS
DELT TEMPERATURE EXCESS OF CLOUD OVER AMBIENT ATMOSPHERE AT SAME HEIGHT IN DEGREES KELVIN
VXSHP X COMPONENT OF CLOUD VELOCITY IN METERS/SEC
VZSHP Z COMPONENT OF CLOUD VELOCITY IN METERS/SEC
XCMSPH X COORDINATE OF CENTER OF SPHERE IN METERS
ZCMSPH Z COORDINATE OF CENTER OF SPHERE IN METERS
SPHNS SINGLY DIMENSIONED ARRAY. SPHNS(I) IS THE NUMBER OF PARTICLES OF THE I SIZE RANGE IN THE SPHERE.
RISTIM THE TIME IN SECONDS AFTER DETONATION THAT THE CLOUD HAS RISEN BUOYANTLY

COMMON /PRTINF/

RO INITIAL RADIUS OF THE CLOUD IN METERS
VGRAV SINGLY DIMENSIONED ARRAY. VGRAV(I) IS THE OPTICALLY WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE I SIZE RANGE
NPRTS THE NUMBER OF PARTICLE SIZE RANGES

COMMON /DISCS/

48
NDSCS THE NUMBER OF DISC SOURCES
TDSC SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE
OF THE DISC SOURCES
XDSC SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE
OF THE CENTER OF THE DISC SOURCES
ZDSC SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE
OF THE CENTER OF THE DISC SOURCES
R2DSC SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE
RADIi OF THE DISC SOURCES
QDSC DOUBLY DIMENSIONED ARRAY. QDSC(I,J) IS THE NUMBER OF
PARTICLES OF THE J SIZE RANGE IN THE I DISC.

NSPH=5
DEL=2./FLOAT(NSPH)
A2=3./FLOAT(NSPH)**2
A3=2./FLOAT(NSPH)**3
DELZ=DEL*RSPH
HREF=ZCMSPH-RSPH-.5*DELZ
R2=RSPH*RSPH
DO 50 I=1,NSPH
IDS C-N
DSC S+I
ZDSC(IDSC)-HREF+FLOAT(I)*DELZ
HM-ZDSC(IDSC)-ZCMSPH
RD2=R2-HMZ*HMZ
VFRAC=A2*(FLOAT(2*I-1))-A3*FLOAT(3*I*(I-1)+1)
DO 5 IPRT=1,NPRTS
QDSC(IDSC,IPRT)=VFRAC*SPhNS(IPRT)
5 CONTINUE
A=ALOG(QDSC(IDSC,1))/VISEXT/DELZ/SQRT(RD2)/RTPI**2
IF(A.GT.1.)GO TO 210
D=1.
GO TO 230
210 D=A
DO 220 IT=1,5
D=(A-1.+ALOG(D))*D/(D-1)
220 CONTINUE
230 R2DSC(IDSC)=RD2/D
GAPTIM=-DELZ*DELZ/D/4./(DZO*ZDSC(IDSC)**N)
XDSC(IDSC)=UO*ZDSC(IDSC)**M * GAPTIM +XCMSPH
TDSC(IDSC)=T+GAPTIM
50 CONTINUE
NDSCS=NDSCS+NSPH
TWIND=T
RETURN
END

BUOYANTLY RISING DUST CLOUD FOR DIRTRAN-I CODE

* * * * * * * * * SUBFILE 10 * * * * * * * * * * * * * * *
SUBROUTINE RISE(TPRES,TNEXT,DEL)
IMPLICIT INTEGER*4 (I-N)
REAL M,NDIF
DIMENSION WK(12,6)
COMMON/BUOYCL/ Y(6),SPHNS(3),RISTIM
COMMON /WNDPRM/ DXZO,DYXO,DZO,UO,M,NDIF,ZINV
COMMON /CLOCK/ TIME,TWIND
COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,IOUNIT,NDIRTU,,BSCAT
DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0.,12,12/

PURPOSE

THIS ROUTINE CALLS A RUNGE KUTTA ROUTINE TO INTEGRATE IN TIME
THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES
ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND
DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED.
SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).

ARGUMENTS

TPRES AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF
INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST
SUCCESSFUL INTEGRATION STEP.
TNEXT THE ENDPOINT OF THE TIME INTERVAL WHICH IS INPUT.

REQUIRED SUBROUTINES

RKM A RUNGE-KUTTA-MERSON INTEGRATION ROUTINE
CONVRT A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT
DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE
WIND DISPERSION MODEL. A GAP TIME DURING WHICH THE
BUOYANT MODEL IS CONTINUED IS COMPUTED.

CALLED BY DUSTCL

IF(TNEXT.GT.TWIND)GO TO 999
T2=TPRES

PERFORM INTEGRATION IN SEGMENTS OF TIME

10 DO 20 NT=1,40
   T1=T2
   T2=1.2*T1
   20 CONTINUE
IF(T2.LE.0.)T2=.5
IF(T2.GT.TNEXT)T2=TNEXT
IF(Y(1)+Y(6).GE.ZINV)GO TO 200
IF(DEL.LT.HMIN)DEL=HMIN
CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)
C
C CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION:
C OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO
C THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB.
C
5 DIFB=ABS(.25*Y(1)*Y(4))
DIFW=DZO*(Y(6)+Y(1))**NDIF
IF(DIFB.GT.DIFW)GO TO 15
CALL CONVRT(T2)
GO TO 200
15 CONTINUE
IF(T2.GE.TNEXT)GO TO 200
IF(T2.GT.300.)GO TO 99
20 CONTINUE
99 WRITE(IOOUT,98)
98 FORMAT(55H *** DIRTRAN-I ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE)
STOP
200 TPRES=T2
RISTIM=TPRES
999 RETURN
END
C
C **************************************** SUBFILE 11 *****************************************
C
SUBROUTINE DIFEQ(N,T,Y,YP)
IMPLICIT INTEGER*4 (I-N)
DIMENSION Y(N),YP(N)
COMMON/PRTINF/ROCL,VGRAV(3),NPRTS
COMMON /TMPPRM/TO,T1,TM
COMMON/WNDPRM/DXZO,DYXO,DZO,UC,UM,DN,ZINV
COMMON/BURST/ACCEL,TBURST
DATA ALPHAK/.25/
C
C **************************************** SUBFILE 11 *****************************************
C
SUBROUTINE DIFEQ(N,T,Y,YP)
IMPLICIT INTEGER*4 (I-N)
DIMENSION Y(N),YP(N)
COMMON/PRTINF/ROCL,VGRAV(3),NPRTS
COMMON /TMPPRM/TO,T1,TM
COMMON/WNDPRM/DXZO,DYXO,DZO,UC,UM,DN,ZINV
COMMON/BURST/ACCEL,TBURST
DATA ALPHAK/.25/
C
C PURPOSE
C
DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE
C RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.
C
C INPUT
THE NUMBER OF DEPENDENT VARIABLES

THE INDEPENDENT VARIABLE, I.E. TIME

RADIUS OF CLOUD

CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE

CLOUD VELOCITY IN WIND DIRECTION

VERTICAL VELOCITY OF CLOUD

X-COORDINATE OF CENTER OF MASS FOR THE CLOUD

THE HEIGHT OF THE CLOUD C.O.M.

OUTPUT

YP AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.

REQUIRED FUNCTIONS

NONE

CALLED BY RKM

IF(T.LT.TBURST)GO TO 200
DELT=T*I*Y(6)**TM
TA=TO+DELT
DTADZ=TM*DELT/Y(6)
VV=U0*Y(6)**UM
VR=SQR((Y(3)-VW)*(Y(3)-VW)+Y(4)*Y(4))

TA THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT

DTADZ THE TEMPERATURE GRADIENT AT CLOUD HEIGHT

VV THE WIND VELOCITY AT CLOUD HEIGHT

VR THE RELATIVE VELOCITY OF THE CLOUD WITH RESPECT TO WIND

TR=Y(2)/TA

CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO WHICH DEPENDS ON THE NUMBER AND PLACEMENT OF CHARGES

ARVOL=3./Y(1)

CONTINUE

YP(1)=ALPHA*K*VR
YP(2)=-(1.+TR)*ARVOL*Y(2)*YP(1)-Y(4)*(DTADZ)
YP(3)=ARVOL*(VW-Y(3))*YP(1)
YP(4)=9.8*TR-ARVOL*Y(4)*YP(1)
YP(5)=Y(3)
YP(6)=Y(4)
GO TO 999

200 N=6
DO 210 I=1,N
YP(I)=0.
210 CONTINUE
YP(4)=ACCEL
YP(5)=Y(3)
YP(6)=Y(4)

999 RETURN
END

C ********************************************************************
C SUBROUTINE CLDIM(CNTRD, HEIGHT, CENWTH, SPCHT, SPCWTH, NCPTS, CPTS5,
C NERR)
IMPLICIT INTEGER*4 (I-N)
C ********************************************************************

PURPOSE

CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS
SEEN FROM THE SPECIFIED OBSERVER POSITION. CLDIM REQUIRES CLOUD
PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH
ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS
VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/. SPCHT IS
REQUIRED INPUT IN THE ARGUMENTS. ALL OUTPUTS ARE ARGUMENTS.

INPUT

SPCHT THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD
IS DESIRED. (METERS)

OUTPUT

CNTRD A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS
THE HORIZONTAL AND VERTICAL COORDINATES, RESPECTIVELY, OF THE
CLOUD CENTROID. (METERS)
HEIGHT THE HEIGHT OF THE CLOUD IN METERS
CENWTH THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS
SPCWTH THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS)
NCPTS  THE NUMBER OF CONTOUR POINTS (\*5)

CPTS  A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N), N.GE.5, WHICH
CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF
THE FIVE CONTOUR POINTS. (METERS)

REQUIRED SUBROUTINES

SCONF  MAIN ROUTINE FOR CONTOUR TRACING ALGORITHM. USED
WITH WIND MODEL.

CALLED BY DUSTCL

**************************************************************************************
DIMENSION CNTRD(2),CPTS(2,200),CPTS5(2,5)
LOGICAL HORIZ,NOCONT
COMMON /BUOYCL/RSPH,DELT,VX,VZ,XCM,ZCM,SPHNS(3),TI
COMMON /PRINF/R0,VGRAV(3),NPRTS
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
COMMON /MODE/
COMMON /CLOCK/T,TWIND
COMMON/WNDPR./DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
COMMON/DISCS/NDSCS,TDSC(10),XDSC(10),ZDSC(10),R2DSC(10),
1 QDSC(10,3)
COMMON /SPECS/ RES,STEP,TANT,CON
EXTERNAL FUNCT,GFUN
DATA VISEXT,ZMIN,RES,TANT /1,0.,.4,.1/
HORIZ=.TRUE.
CON=ALOG(VISEXT)
CPTS5(2,1)=SPCHT
CPTS5(2,5)=SPCHT
U=U0*CPTS5(2,1)**UM
CPTS5(1,1)=T*U*SINTH
CPTS5(1,5)=CPTS5(1,1)
NSERCH=-1
STEP=20.
CALL CLIMB(FUNCT,GFUN,CPTS5,FP1,NSERCH,NOCONT)
NSERCH=1
STEP=20.
CALL CLIMB(FUNCT,GFUN,CPTS5(1,5),FP1,NSERCH,NOCONT)
SPCUTH=CPTS5(1,5)-CPTS5(1,1)
NCPTS=5
IF(T.LE.TWIND)GO TO 50
IND=NDSCS-2
CPTS(2,1)=ZDSC(IND)
CPTS(1,1)=(XDSC(IND)+(T-TDSC(IND))*U0*ZDSC(IND)**UM)*SINTH
STEP=20.
CALL SCONF(FUNCT,ZMIN,NCPTS,CPTS,NERR)
C IF(NERR.NE.0) GO TO 999
   CALL PTSPEC(N:CPTS,CPTS,XHIGH,HEIGHT,XC1,CNTRD,XC2,CENWTH)
   GO TO 100
50 CNTRD(1)=XCM*SINTH
   CNTRD(2)=ZCM
   XHIGH=CNTRD(1)
   HEIGHT=ZCM+RSPH
   CENWTH=2.*RSPH
100 CPTS5(1,2)=CNTRD(1)-CENWTH/2.
   CPTS5(1,4)=CNTRD(1)+CENWTH/2.
   CPTS5(1,3)=XHIGH
   CPTS5(2,2)=CNTRD(2)
   CPTS5(2,3)=HEIGHT
   CPTS5(2,4)=CNTRD(2)
   NPTS=5
999 RETURN
END

C ROUTINE FOR TRACING A CONTOUR OF A FUNCTION OF TWO VARIABLES

C ********************************** SUBFILE 13 **********************************

C SCONF IS THE CONTROLLING ROUTINE FOR THE CONTOUR TRACING
C ALGORITHM. THE FUNCTION WHOSE CONTOUR IS DESIRED MUST BE A
C CONTINUOUS, REAL VALUED FUNCTION OF TWO VARIABLES.
C THE CONTOUR FOUND IS ONE CONTINUOUS CLOSED CONTOUR OR
C ONE CONTINUOUS CURVE BEGINNING AND ENDING ON THE SPECIFIED
C BOUNDARY. THERE MAY BE OTHER PIECES TO THE CONTOUR.
C
C INPUT
C
C FUNCTTHE FUNCTION WHOSE CONTOUR IS TO BE FOUND
C CONTHE CONTOUR LEVEL.
C YMNLOWER BOUND FOR THE SECOND COORDINATE
C RESTHE RESOLUTION LENGTH.
C DELTA THE INITIAL STEP SIZE WHEN LOOKING ALONG THE
C GRADIENT TO FIND A POINT ON THE CONTOUR.
C THETAN THE TANGENT OF THE MAXIMUM ALLOWABLE ROTATION ANGLE
C BETWEEN SUCCESSIVE LINE SEGMENTS OF THE POLYGONAL
C CONTOUR.
C MAXDIM THE NUMBER OF POINTS FOR WHICH STORAGE HAS BEEN
C ALLOCATED IN THE ARRAY CP.
C CP A DOUBLY DIMENSIONED ARRAY TO BE FILLED WITH THE
C COORDINATES OF THE CONTOUR POINTS. CP(I,J) IS THE
C I-TH COORDINATE, I=1,2, OF THE J-TH POINT OF THE
C CONTOUR. UPON CALLING SCONF, CP(I,1),I=1,2, SHOULD
C BE THE COORDINATES OF THE BEST GUESS AS TO WHERE THE
CONTOUR IS LOCATED.

OUTPUT

CP A DOUBLY DIMENSIONED ARRAY FILLED WITH THE
COORDINATES OF THE CONTOUR POINTS. CP(I,J) IS THE
1-TH COORDINATE, I=1,2, OF THE J-TH POINT OF THE
CONTOUR.
MAXDIM THE NUMBER OF POINTS REPORTED IN CP.
ERROR A NUMERICAL FLAG TELLING WHAT TYPE OF ERROR
OCCURRED WHILE RUNNING, IF ANY.
MEANING OF ERROR CODE-
-1=>ARRAY FILLED COMPLETELY
0=>CONTOUR CLOSED OR MET BOUNDARIES, NO PROBLEM.
1=>NO CONTOUR FOUND
2=>PTFIND UNABLE TO FIND NEXT POINT. CHECK FOR
KINK OR DISCONTINUITY OF THE FUNCTION.

CALLED SUBROUTINES.

CLIMB LOCATES FIRST POINT OF CONTOUR OR THAT CONTOUR
DOESN'T EXIST.
PTFIND FINDS ADDITIONAL POINTS ON THE CONTOUR GIVEN AT LEAST
ONE.

LOCAL VARIABLES

LEFT A LOGICAL WHICH IS .TRUE. FOR COUNTERCLOCKWISE CONTOUR
SEARCH AND .FALSE. FOR CLOCKWISE SEARCH.
TMP TEMPORARY STORAGE USED FOR SWAPPING POSITIONS IN
THE ARRAY OF CONTOUR POINTS.
NOCONT A LOGICAL ERROR FLAG MEANING EITHER THE CONTOUR
DOES NOT EXIST OR IS TOO SMALL TO BE TRACED.
ENDCON A NUMERICAL FLAG INDICATING THE STATUS OF THE
CONTOUR POINTS TRACED WITH VALUES:
-1 THE ARRAY CP IS FILLED
0 THE CONTOUR IS CLOSED
1 THE CONTOUR HAS MET A BOUNDARY
2 THE REQUIRED STEP SIZE HAS BECOME LESS
   THAN THE RESOLUTION.

***********************************************************************
SUBROUTINE SCONF(FUNCT, YMN, MAXDIM, CP, ERROR)
IMPLICIT INTEGER*4 (I-N)
EXTERNAL FUNCT, GFUN

56
INTEGER ENDCON, ERROR
LOGICAL NOCONT, ERR, LEFT
DIMENSION TMP(2)
DIMENSION CP(2,200)
COMMON/LINE/BASE(2), DIR(2), DFDS/SPECS/RES, DELTA, THETAN, CON
COMMON/LIMIT/YMIN, FMIN
COMMON /IOUNIT/ IOIN, IOOUT, ISPTPF, LUNIT, NDIRTU, NBSCAT
DATA FMIN/-29./
MAXDIM=200
ERROR=0
YMIN=YMN

C FINDING THE POINT ON THE CONTOUR, OR IF A CONTOUR EVEN EXISTS.
C
NSERCH=0
CALL CLIMIB(FUNCT, GFUN, CP, FCP, NSERCH, NOCONT)
C *** APPROPRIATE ACTION IF THE CONTOUR DOES NOT EXIST,
IF(NOCNT)GO TO 99
L=1
LEFT=.TRUE.
CALL PTFIND(LEFT, GFUN, MAXDIM, CP, L, ENDCON)
C
C ** THE NEXT TWO DO LOOPS ARE FOR SWITCHING THE POSITIONS
C OF THE POINTS IN THE ARRAY AROUND FOR EASE IN
C WORKING WITH LATER ON.
C
MID=L/2
DO 17 I=1, MID
   J=L-I+1
   DO 19 K=1, 2
      TMP(K)=CP(K, I)
      CP(K, I)=CP(K, J)
      CP(K, J)=TMP(K)
   19 CONTINUE
   17 CONTINUE
C ** IS THE CONTOUR CLOSED OR IS THE ARRAY FILLED
IF(ENDCON)97, 76, 27
27 IF(ENDCON.EQ.2)ERROR=2
   IF(ENDCON.EQ.2)WRITE(IOOUT, 796)
   LEFT=.FALSE.
   CALL PTFIND(LEFT, GFUN, MAXDIM, CP, L, ENDCON)
   MAXDIM=L
   IF(ENDCON)97, 76, 95
76 CONTINUE
   MAXDIM=L+1
   CP(1, MAXDIM)=CP(1, 1)
   CP(2, MAXDIM)=CP(2, 1)
   GO TO 98
IF(ENDCON.EQ.1)GO TO 98
ERROR=2
MAXDIM=L
WRITE(100OUT, 796)
796 FORMAT(53H *** DIRTRAN-I ERROR - CONTOUR TRACING ROUTINE STUCK /
1 56H CHECK FOR DISCONTINUITY OF FUNCTION )
GO TO 999
ERROR=-1
WRITE(100OUT, 797)
797 FORMAT(51H *** DIRTRAN-I ERROR - ARRAY CPTS NOT LARGE ENOUGH )
GO TO 999
ERROR=0
GO TO 999
ERROR=1
WRITE(100OUT, 799)
799 FORMAT(51H *** DIRTRAN-I WARNING - CLOUD CONTOUR NOT FOUND /
1 49H MAY HAVE DISSIPATED )
GO TO 999
99 RETURN
END

FUNCTION GFUN(S)
IMPLICIT INTEGER*4 (I-N)
GFUN IS THE RESTRICTION OF THE TWO DIMENSIONAL FUNCTION, F, TO
A LINE. I.E. FORM G(S)=F(X,Y), WHERE (X,Y)=BASE+S*DIR.
EXTERNAL FUNCT
DIMENSION P(2)
COMMON/LINE/BASE(2), DIR(2), DFDS/SPECS/RES, DELTA, THETAN, CON
CALL VSUM(BASE, DIR, S, P)
GFUN=FUNCT(P(1), P(2))
RETURN
END

THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR
BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG
IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL.
AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE
UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION.
IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS.
ARGUMENTS PASSED.
INPUT

FUNCT - THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.
P1 - THE STARTING POINT.

OUTPUT

P1 - THE POINT ON THE CONTOUR OR THE POINT AT WHICH
THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR
LEVEL
FP1 - THE VALUE OF THE FUNCTION AT P
NOCONT - THE ERROR FLAG.
  F - NO PROBLEM
  T - NO CONTOUR FOUND.
ERR - ERROR FLAG RETURNED BY 'NTRSCT'
  F - NO ERROR
  T - ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEEDED

IN ADDITION, IN COMMON ARE...,

YMIN - THE LOWER LIMIT ON Y.
DELTA - THE STEP SIZE, MODIFIED IN THIS SUBROUTINE.
CON - THE CONTOUR LEVEL.
RES - THE RESOLUTION LENGTH

OTHER VARIABLES INCLUDE

GRAD - THE GRADIENT VECTOR
PO - THE CURRENT POINT ON THE GRADIENT.
P1 - THE POINT ON THE GRADIENT BEING TESTED
  TO SEE ABOUT CONTOUR EXISTENCE.
FP0, FP1 - THE FUNCTION VALUES OF PO AND P1.

CALLED SUBROUTINES

GRAD2 - FINDS THE GRADIENT VECTOR OF A FUNCTION AT
A POINT AND THE SLOPE THERE.
UNIT - CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.
VSUM - VECTOR SUM OF THE FORM C*A+B WHERE S IS SCALAR
  MULTIPLIER OF B.

SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSERCH,NOCONT)
IMPLICIT INTEGER*4 (I-N)
EXTERNAL FUNCT,GFUN
LOGICAL NOCONT
DIMENSION GRAD(2),PO(2),P1(2),P(2)
COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES1,LTA,THETAN,CON
COMMON/LIMIT/YMIN,FMIN
NOCONT=.FALSE.

59
ONEM = 1.0
IF (NSERCH .EQ. 0) GO TO 7
DELTA = SIGN (DELTA, FLOAT (NSERCH))
FP1 = FUNCT (P1(1), P1(2))
IF (FP1 .LT. CON) GO TO 25
GO TO 22
3 CONTINUE
P0(1) = P1(1)
P0(2) = P1(2)
FP0 = FP1
C ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.
4 CALL GRAD2 (PO, FUNCT, RES, GRAD, DFDS)
5 CALL VSUM (PO, GRAD, DELTA, P1)
C ** IS THE POINT HEADING BELOW YMIN ??
6 IF (P1(2) .GE. YMIN) GO TO 7
   P1(2) = YMIN
   CALL VSUM (P1, PO, ONEM, GRAD)
   CALL UNIT (GRAD, GRAD, DELTA)
   IF (ABS (DELTA) .LT. RES) GO TO 25
   FP1 = FUNCT (P1(1), P1(2))
C ** HAS THE CONTOUR BEEN CROSSED ??
8 IF (FP1 .GT. FP0) GO TO 22
   IF (FP1 .GT. FP0) GO TO 3
   DELTA = DELTA / 2.
   IF (ABS (DELTA) .LT. RES) GO TO 25
   GO TO 5
25 NOCONT = .TRUE.
   GO TO 99
22 CONTINUE
C BEGIN HORIZONTAL SEARCH
P0(2) = P1(2)
31 P0(1) = P1(1)
FP0 = FP1
40 P1(1) = P0(1) + DELTA
   FP1 = FUNCT (P1(1), P1(2))
   IF (ABS (DELTA) .LT. RES / 2) GO TO 99
   IF (FP1 .GE. CON) GO TO 31
   DELTA = DELTA / 2.
   GO TO 40
99 CONTINUE
RETURN
END

C ************************************************************ SUBFILE 16 ************************************************************ C
C
C
C
C THIS SUBROUTINE CALCULATES THE UNIT GRADIENT VECTOR
OF A FUNCTION AT THE GIVEN POINT USING THE FORMULA
PARTIAL DF/DX = (F(X+R,Y)-F(X,Y))/R WHERE R IS
SMALL, SIMILARLY FOR PARTIAL DF/DY. IT THEN
NORMALIZES THE RESULTANT VECTOR FOR THE UNIT GRADIENT
AND FINDS THE SLOPE, WHICH IS THE MAGNITUDE OF
OF THE REGULAR GRADIENT.

ARGUMENTS PASSED
PT - THE POINT AT WHICH THE UNIT GRADIENT IS FOUND.
FUNCT - THE FUNCTION(X,Y)
RES - R
GRAD - THE UNIT GRADIENT
SLOPE - SLOPE AT PT

OTHER VARIABLES
COO, C10, C01 - THE FUNCTION AT THE POINTS F(X,Y), F(X+R,Y),
AND F(X,Y+R) RESPECTIVELY.

SUBROUTINES CALLED
UNIT - NORMALIZES A VECTOR AND FINDS ITS MAGNITUDE.

SUBROUTINE GRAD2(PT, FUNCT, RES, GRAD, SLOPE)
IMPLICIT INTEGER*4 (I-N)
DIMENSION PT(2), GRAD(2)
COO = FUNCT(PT(1), PT(2))
C10 = FUNCT(PT(1)+RES, PT(2))
C01 = FUNCT(PT(1), PT(2)+RES)
GRAD(1) = (C10-COO)/RES
GRAD(2) = (C01-COO)/RES
CALL UNIT(GRAD, GRAD, SLOPE)
RETURN
END

*******************************************************************************
SUBFILE 17
*******************************************************************************

THIS SUBROUTINE FINDS THE INTERSECTION OF A LINE SPECIFIED BY
A BASE POINT, BASE, AND A DIRECTION VECTOR, DIR. THE ROUTINE
DETERMINES A REAL NUMBER, S, SUCH THAT BASE+S*DIR IS THE
POINT OF INTERSECTION WITH A CONTOUR THROUGH NEWTON-RAPHSON
ITERATION.

INPUT TERMS.

SLIM - THE LIMITING VALUE OF S, TO PREVENT THE INTERPOLATION
FROM BREAKING DOWN IN UNUSUAL CASES AND GOING INTO AN
INFINITE LOOP
GFUN - FUNCTION OF S EQUIVALENT TO FUNCTION(X,Y).
OUTPUT TERMS.

S - A SCALAR PARAMETER SPECIFYING POINTS ON THE LINE.
P - THE POINT OF INTERSECTION.
ERR - AN ERROR FLAG.
F - NO PROBLEM
T - NO INTERSECTION FOUND WITHIN THE SET LIMITATIONS.

TERMS IN COMMON.
BASE - A 2-VECTOR CONTAINING THE COORDINATES OF THE BASE
POINT OF THE LINE. BASE MUST BE INITIALIZED BEFORE
THE CALL TO NTRSCT.
DIR - A UNIT DIRECTION 2-VECTOR FOR THE LINE. DIR ALSO MUST
BE INITIALIZED PREVIOUS TO THE CALL.
CON - THE CONTOUR LEVEL TO BE FOUND.
DFDS - DF/DS WITH F = FUNCTION(X,Y), USED AS AN INITIALIZER
FOR THE INTERPOLATION. FIRST ONE MUST BE INITIALIZED
BEFORE THE CALL.
RES - THE RESOLUTION

CALLED SUBROUTINES

VSUM - VECTOR ADDITION SUBROUTINE OF FORM VSUM(A,B,S,C) =
C = A + S * B WHERE A, B, C ARE 2-VECTORS AND S SCALAR.

SUBROUTINE NTRSCT(S,P,GFUN,SLIM,ERR)
IMPLICIT INTEGER*4 (I-N)
EXTERNAL GFUN,FUNCT
LOGICAL ERR
DIMENSION F(2)
COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
COMMON/LIMIT/YMIN,FMIN
SOLD=0.0
ERR=.FALSE.

C *** ITER IS THE NUMBER OF ITERATIONS.***
C
ITER=0
FSOLD=GFUN(SOLD)
IF(FSOLD.LE.FMIN)GO TO 999
DS=(CON-FSOLD)/DFDS
S=SOLD+DS
IF(ABS(S).LT.SLIM)GO TO 110
DS=SIGN(SLIM/2.,DS)
S=SOLD+DS
110 ITER=ITER+1

62
FS=GFUN(S)
IF(FS.LE.FMIN)GO TO 999
5 DFDS=(FS-FSOLD)/(S-SOLD)
   DS=(CON-FS)/DFDS
   SOLD=S
   FSOLD=FS
   S=S+DS
IF(ABS(DS).LT.RES)GO TO 998
IF(ABS(S).GT.SLIM)GO TO 999
IF(ITER.GT.9)GO TO 999
   GO TO 110
998 CALL VSUM(BASE,DIR,S,P)
   GO TO 1000
999 ERR=.TRUE.
1000 RETURN
END

C
C ****************************************** SUBFILE 18 ******************************************
C
C PTFIND DETERMINES A STRING OF POINTS ON THE CONTOUR MARCHING
C EITHER CLOCKWISE (.NOT.LEFT) OR COUNTERCLOCKWISE (LEFT) UNTIL
C THE CONTOUR CLOSES, A BOUNDARY IS MET, THE ARRAY CPTS IS FILLED,
C OR AN ERROR OCCURS. A STARTING POINT MUST BE PROVIDED. PTFIND
C BEGINS BY CALCULATING THE TANGENT TO THE CONTOUR WHICH IS
C PERPENDICULAR TO THE GRADIENT. SUBSEQUENT POINTS ARE FOUND BY
C EXTENDING THE TANGENT THROUGH THE TWO MOST RECENTLY DETERMINED
C POINTS AND SEARCHING IN THE PERPENDICULAR DIRECTION. IN CASE
C OF ERROR, ONE RESTART IS ALLOWED.
C THE STEP SIZE, DELTA, IS HALVED WHEN THE EXTRAPOLATED POINT
C EXCEEDS BOUNDARY OR THE SEARCH IN THE PERPENDICULAR DIRECTION
C DID NOT YIELD A POINT WITHIN LIMITS DETERMINED BY CURVATURE.
C DELTA IS LENGTHENED INVERSELY AS THE LOCAL CURVATURE OF THE CURVE.
C
C INPUT TERMS
C GFUN-THE ONE VARIABLE EQUIVALENT TO THE FUNCTION
C RESTRICTED TO A LINE THROUGH POINT, BASE, IN DIRECTION,DIR
C MAXDIM-THE MAXIMUM NUMBER OF POINTS IN THE ARRAY.
C CP-THE ARRAY OF CONTOUR POINTS AS IT INITIALLY IS.
C L-THE INDEX OF HOW MANY POINTS HAD BEEN FOUND BEFORE.
C
C OUTPUT TERMS
C CP-CURRENT ARRAY OF CONTOUR POINTS
C L-CURRENT INDEX OF POINTS. ON RETURN, L IS
C THE NUMBER OF POINTS FOUND.
C ENDCON-A NUMERICAL FLAG THAT TELLS
C WHY THE SUBROUTINE IS ENDING THE SEARCH.
C ENDCON EQUALS.
-1=>THE ARRAY CP IS FILLED.
0=>THE CONTOUR IS CLOSED.
1=>THE CONTOUR HAS MET A BOUNDARY.
2=>REQUIRED STEP SIZE BECAME LESS THAN RESOLUTION.

OTHER PERTINENT VARIABLES IN COMMON...

CON-THE CONTOUR LEVEL
BASE-THE POINT ALONG THE TANGENT.
DFDS-THE SLOPE OF THE CURVE. MODIFIED IN 'NTRSCT'
RES-THE RESOLUTION LENGTH
DELTA-THE STEP SIZE ALONG THE TANGENT, MODIFIED HERE.
YMIN-LOWER BOUND FOR SECOND COORDINATE

OTHER VARIABLES

PI-CURRENT POINT ON CONTOUR
PNEXT-NEXT POINT ON CONTOUR, NOT YET APPROVED BY 'ENDTST'.
FULTAN-THE APPROXIMATION OF THE TANGENT AT PNEXT BY THE VECTOR FROM PI TO PNEXT.
TAN-THE UNIT VECTOR OF FULTAN
DIST-THE MAGNITUDE OF FULTAN, USED IN 'ENDTST'.
S-THE DISTANCE ALONG THE GRADIENT FROM BASE TO A POINT ON THE CONTOUR.
SMAX-THE LARGEST ALLOWABLE VALUE OF S.

SUBROUTINES CALLED.

VSLM(A,B,S,C)-VECTOR ADDITION OF THE FORM C=A+S*B WHERE A,B,C ARE 2-VECTORS AND S IS SCALAR.
PERP(A,B)-A,B ARE 2-VECTORS AND B IS A VECTOR ROTATED 90 DEGREES COUNTERCLOCKWISE.
NTRSCT(S,P,GFUN,SLIM,IC,ERR)-FINDS THE INTERSECTION OF A LINE DRAWN FROM BASE(IN COMMON) ALONG DIR, THE DIRECTION VECTOR(IN COMMON), WITH THE CONTOUR LEVEL (IN COMMON). S IS THE DISTANCE FROM BASE TO THE CONTOUR, P IS THE POINT OF INTERSECTION, GFUN IS GFUN(S) AND IS A ONE VARIABLE EQUIVALENT OF THE FUNCTION WHOSE CONTOUR IS BEING SOUGHT, SLIM IS THE MAXIMUM ALLOWABLE VALUE OF S, AND ERR IS THE ERROR FLAG (LOGICAL). IC IS A LOGICAL VARIABLE WHERE T MEANS FOR NTRSCT TO STOP AFTER 5 ITERATIONS.
ENDTST-F(PO,P,TAN,DIST,SMAX,1,M,ENDCON)-FINDS IF THE CONTOUR IS CLOSED, HAS RUN INTO THE Y BOUNDARY OR IF THE ARRAY OF CONTOUR POINTS IS FILLED.
UNIT(A, B, S) = A, B ARE 2-VECTORS AND B IS THE UNIT NORM OF A
AND S IS THE MAGNITUDE OF A.

SUBROUTINE PTDFIND(LEFT, GFUN, MAXDIM, CP, L, ENDCON)
IMPLICIT INTEGER*4 (I-N)
EXTERNAL GFUN, FUNCT
INTEGER ENDCON
LOGICAL ERR, ONCE, LEFT
DIMENSION CHORD(2), TAN(2)
DIMENSION CP(2, MAXDIM)
COMMON/LINE/BASE(2), DIR(2), DFDS/SPECS/RES, DELTA, THETAI, CO
COMMON/LIMIT/YMIN, FMIN
ERR=.FALSE.
RES2=RES/2.

CALL GRAD2(CP(1, L), FUNCT, 5.*RES, DIR, DFDS)
CALL PERP(DIR, TAN)
IF(.NOT.LEFT)GO TO 31
TAN(1) = TAN(1)
TAN(2) = TAN(2)
DFDS = -DFDS
31 DELTA = 10.
TANTH = 1.

C *** STEPPING DELTA ALONG THE TANGENT ***
10 CALL VSUM(CP(1, L), TAN, DELTA, BASE)
C *** IS BASE BELOW THE MINIMUM PERMISSIBLE Y ***
IF(BASE(2).GT.YMIN)GO TO 29
DELTA = DELTA/2.
IF(ABS(DELTA).LT.RES2)GO TO 91
GO TO 10

C DETERMINE THE RANGE OF SEARCH PERPENDICULAR TO THE EXTENDED
TANGENT, SMAX, AND CALL NTRSCAT TO LOCATE THE CONTOUR POINT
C
29 SMAX = AMIN1(ABS(DELTA*TANTH+RES), ABS((BASE(2)-YMIN)/DIR(2)))
DFDS0 = DFDS
30 CALL NTRSCAT(S, CP(1, L+1), GFUN, SMAX, ERR)
C *** HAS THE CONTOUR BEEN FOUND WITHIN LIMITING CONDITIONS ***
IF(.NOT.ERR)GO TO 32
DFDS = DFDS0
DELTA = DELTA/2.
C
C   CHECK TO SEE IF STEP SIZE, DELTA, IS BELOW MINIMUM, RES2.
C   IF SO, ALLOW ONE RESTART THEN ABORT
C
IF(ABS(Delta).GE.RES2)GO TO 10
IF(ONCE)GO TO 92
GO TO 9
C   DETERMINE TANGENT FOR NEWLY FOUND POINT
32  CALL VSUM(CP(I,L+1),CP(I,L),-1.,TAN)
    CALL UNIT(TAN,TAN,DIST)
    CALL PERP(TAN,DIR)
C ** IS THE CONTOUR CLOSED **
C WHAT THIS NEXT CHECK DOES IS TO LOOK WITHIN A BOX
C DELTA BY 2*SMAX, WHERE P AND THE PREVIOUS POINT ARE AT EACH
C END OF THE LINE SEGMENT BETWEEN THE MIDPOINTS OF THE TWO
C OPPOSITE SIDES 2*SMAX LONG, TO SEE IF THE STARTING POINT
C IS WITHIN.
    CALL VSUM(CP,CP(1,L),-1.,CHORD)
    VAR1=DOTPRD(DIR,CHORD)
    VAR2=DOTPRD(TAN,CHORD)
    IF(ABS(VAR1).GE.RES*2.)GO TO 35
    IF((VAR2.LT.DIST).AND.(VAR2.GT.0.))GO TO 90
C   BEGIN SEARCH FOR NEXT POINT
35  L=L+1
    IF(CP(2,L).LT.RES2+YMIN)GO TO 91
    IF(L.GE.MAXDIM)GO TO 89
    DELTA=AMINI(10.,DELTA*(SMAX+2.*RES)/((ABS(S)+RES)*2.),DELTA*2.)
    ONCE=.FALSE.
    TANTH=THEIAN
    GO TO 16
89  ENDCON=-1
    GO TO 99
90  ENDCON=0
    L=L+1
    GO TO 99
91  ENDCON=1
    GO TO 99
92  ENDCON=2
99  RETURN
END
C
C **************************************** SUBFILE 19 ****************************************
C
SUBROUTINE VSUM(A,B,S,C)
IMPLICIT INTEGER*4 (I-N)
DIMENSION A(2),B(2),C(2)
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR
DO 14 J=1,2
14 C(J)*A(J)+S*B(J)
RETURN
END

C ***** SUBFILE 20 ****************************
C
C SUBROUTINE PERP(A,B)
IMPLICIT INTEGER*4 (I-N)
DIMENSION A(2),B(2)
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A
B(1)=-A(2)
B(2)=A(1)
RETURN
END

C ***** SUBFILE 21 ****************************
C
C SUBROUTINE UNIT(A,B,XNORM)
IMPLICIT INTEGER*4 (I-N)
DIMENSION A(2),B(2)
C *** B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE
XNORM=SQRT(A(1)**2+A(2)**2)
B(1)=A(1)/XNORM!
B(2)=A(2)/XNORM!
RETURN
END

C ***** SUBFILE 22 ****************************
C
C FUNCTION DOTPRD(A,B)
IMPLICIT INTEGER*4 (I-N)
DIMENSION A(2),B(2)
C
C DOTPRD IS THE SCALER PRODUCT OF A AND B
C
C DOTPRD=A(1)*B(1)+A(2)*B(2)
RETURN:
END

C ***** SUBFILE 23 ****************************
C
C THIS SUBROUTINE TAKES AN ARRAY OF POINTS OF A CONTOUR
C AND FINDS THE HIGHEST POINT, THE TWO POINTS ALONG A
C LINE OF SIGHT (THESE TWO MAY NOT BE IN THE ARRAY AND WILL
C BE FOUND BY INTERPOLATION FROM RATIOS) AND THEIR CENTROID,
C THE CENTROID AND THE TWO POINTS WITH THE SAME Y VALUE, THE LENGTH
C OF THE LINE OF SIGHT, AND THE SHEAR DISTANCE. NOTE- THE POINT ON
C THE CENTROID LINE WITH THE GREATER X VALUE IS THE LEADING
EDGE POINT.

INPUT TERMS

SPCHT - A SPECIFIED HEIGHT AT WHICH THE WIDTH IS DESIRED
CP - THE ARRAY OF CONTOUR POINTS
L - THE NUMBER OF POINTS IN CP.

OUTPUT TERMS

ZHI - Z COORDINATE OF HIGHEST POINT ON CONTOUR.
XHI - X COORDINATE OF HIGHEST POINT ON CONTOUR.
XCL - X COORDINATE OF LEFTMOST POINT ON CONTOUR.
XCR - X COORDINATE OF RIGHTMOST POINT ON CONTOUR.
CNTRD - THE CENTROID
CENWTH - WIDTH AT THE CENTROID HEIGHT
XSPCL - X COORDINATE OF LEFTMOST POINT OF CONTOUR AT THE
SPECIFIED HEIGHT, SPCHT
XSPCR - X COORDINATE OF RIGHTMOST POINT OF CONTOUR AT THE
SPECIFIED HEIGHT
SPCWTH - THE WIDTH OF THE CONTOUR AT THE SPECIFIED HEIGHT

SUBROUTINES CALLED

INTRP - PERFORMS A SIMPLE INTERPOLATION BY RATIO TO FIND
A POINT WITH A GIVEN Y COORDINATE, GIVEN A POINT
ON EITHER SIDE. THE CALL IS
CALL INTRP(P1,P2,Y,P) WHERE P1,P2 ARE THE GIVEN
POINTS, Y THE GIVEN Y COORD., AND P THE POINT RETURNED.

IN INTRP...
VSUM(A,B,S,C) WHERE A,B,S ARE GIVEN AND
C=A+S*B. A,B,C ARE 2-VECTORS AND S, A SCALAR.

SUBROUTINE PTSPEC(L,CP,XHI,ZHI,XCL,CNTRD,XCR,CENWTH)
IMPLICIT INTEGER*4 (I-N)
DIMENSION CNTRD(2),CP(2,L)
LOGICAL ERR,BTWN
BTWN(A,B,C)=(A.LE.B.AND.B.LT.C).OR.(A.GE.B.AND.B.GT.C)
ERR=.FALSE.
NHI=1
NLO=1
NL=1
NR=1
DO 10 J=1,L
IF(CP(2,J).GT.CP(2,NHI))NHI=J
IF(CP(2,J).LT.CP(2,NLO))NLO=J

68
IF(CP(1,J).LT.CP(1,NL))NL=J
IF(CP(1,J).GT.CP(1,NR))NR=J

10 CONTINUE

XHI=CP(1,NHI)
ZHI=CP(2,NHI)
CNTRD(2)=MAX1(CP(2,NL),CP(2,NR))
NCR=NL
NCL=NR
DO 20 J=1,L
IF(.NOT.BTWN(CP(2,J),CNTRD(2),CP(2,J+1)))GO TO 20
IF(CP(1,J).LT.CP(1,NCL))NCL=J
IF(CP(1,J).GT.CP(1,NCR))NCR=J

20 CONTINUE

CALL INTRP(CP(1,NCL),CP(1,NCL+1),CNTRD(2),XCL)
CALL INTRP(CP(1,NCR),CP(1,NCR+1),CNTRD(2),XCR)
CNTRD(1)=(XCL+XCR)/2.
CENWTH=XCR-XCL

C IF(.NOT.BTWN(ZLO,SPCHT,ZHI))GO TO 99
C NCR=NCL
C NCL=NE
C DO 30 J=1,L
C IF(.NOT.BTWN(CP(2,J),SPCHT,CP(2,J+1)))GO TO 30
C IF(CP(1,J).LT.CP(1,NCL))NCL=J
C IF(CP(1,J).GT.CP(1,NCR))NCR=J

30 CONTINUE

CALL INTRP(CP(1,NCL),CP(1,NCL+1),SPCHT,XSPCL)
CALL INTRP(CP(1,NCR),CP(1,NCR+1),SPCHT,XSPCR)
SPCWT=XSPCR-XSPCL
GO TO 99
C ERR=.TRUE.
C XSPCL=0.0
C XSPCR=0.0
C SPCWT=0.0
C GO TO 99
C RETURN
END

**************************************************************************
** SUPFILE 24 **************************************************************************

SUBROUTINE INTRP(P1,P2,Y,X)
IMPLICIT INTEGER*4 (1-N)

GIVEN POINTS P1 AND P2 AND SECOND COORDINATE, Y,
INTRP DETERMINES THE FIRST COORDINATE, X, OF A POINT, (X,Y),
WHICH IS ON A LINE DRAWN THROUGH P1 AND P2

DIMENSION P1(2),P2(2),P(2),DIF(2)
IF(ABS(P2(2)-P1(2)).LT.1.E-30)GO TO 89
RATIO=(Y-P1(2))/(P2(2)-P1(2))
ONEM=1.
CALL VSUM(P2,P1,ONEM,DIF)
CALL VSUM(P1,DIF,RATIO,P)
X=P(1)
GO TO 99
89 X=P1(1)
99 RETURN
END

CALCULATION OF 0-ORDER AND INTERPOLATION OF HIGHER ORDER MOMENTS

C ************************** SUBFILE 25 **************************

SUBROUTINE MOMENT(VGRAV, ZIN, H, TIN, Q, XBAR, SIGW2, SIGP2)
IMPLICIT INTEGER*4 (I-N)
REAL M,N,NM
DIMENSION AL(9),Z(9),T(9),XB(81,4,4),SW(81,4,4),SP(81,4,4),NM(9)
DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),ITC(4),II(4),Y(9,4)
LOGICAL FIRST
C
C PURPOSE
C TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THE; COMPUTE:
C THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF
C THE HIGHER ORDER MOMENTS
C
C INPUT
C
C VGRAV THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE
C IN METERS / SEC
C ZIN THE HEIGHT (METERS) AT WHICH THE MOMENTS ARE DESIRED
C H THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS
C TIN THE TIME IN SECONDS AFTER RELEASE
C
C OUTPUT
C
C Q THE VERTICAL CONCENTRATION OF PARTICLES IN PARTS/METER
C AT HEIGHT Z
C XBAR THE DISPLACEMENT (METERS) IN THE X (IE WIND) DIREC
C OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z

70
SIGW2 THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD
DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
SIGP2 THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND
DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2

SUBROUTINES CALLED

DTERPS PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS
INTO A ONE DIMENSIONAL ARRAY
DTERPI A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE
FOR GIVEN ARGUMENTS AND ARRAYS
GREEN CALCULATES THE GREENS FUNCTION WHICH IS THE
0-ORDER MOMENT

CALLED BY CWIND

*******************************************************************************
IF(.NOT.FIRST)GO TO 5

READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT

Z LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULATED
T LOG OF NON-DIMENSIONAL TIMES AT WHICH MOMENTS ARE TABULATED
AL NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE TABULATED
NM DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE TABULATED
XB TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED TO MEAN HORIZONTAL DISPLACEMENT)
SW TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECOND ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION)
SP TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO WIND AND CROSS-WIND VARIANCES

READ(NDIRTU,1) NZ,NT,NA,NN
1 FORMAT(413)
  NT(1)=NZ-1
  NT(2)=NT-1
  NT(3)=NA-1
  NT(4)=NN-1
READ(NDIRTU,2) (Z(I),I=1,NZ)
2 FORMAT(6EL3.5)
READ(NDIRTU,2) (T(I),I=1,NT)
READ(NDIRTU,2) (AL(I),I=1,NA)
READ(NDIRTU,2) (NM(I),I=1,NN)
NZT=NZ*NT
DO 3 L=1,NN
READ(NDIRTU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)
READ(NDIRTU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)
READ(NDIRTU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)
3 CONTINUE
FIRST=.FALSE.
REWIND NDIRTU
5 CONTINUE

C CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM:
C
SCLU=DZO*H**(N-1.)
XI(1)=ZIN/H
XI(2)=SCLU*TIN/H
XI(3)=VGRAV/SCLU
XI(4)=N
CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)
Q=Q/H
IF(Q .LE. 1.E-20) GO TO 999

C TAKE LOGS FOR LOGARITHMIC INTERPOLATION
C
XI(1)=ALOG(XI(1))
XI(2)=ALOG(XI(2))
C
DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO
C BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS
C OF THE CUBE HAVE TABULATED VALUES
C
DO 100 I=1,4
11(I)=IB(I)
100 CONTINUE
DO 101 I11=1,4
1=5-I11
C IA=I11
IA=I11(I)
IF(XI(1) .GE. X(IA,1) .AND. X(IA,1) .LE. X(IA+1,1)) GO TO 101
IF(XI(1) .LT. X(IA,1) .AND. IA .EQ. 1) GO TO 101
IF(XI(1) .GT. X(IA,1) .AND. IA .EQ. NTC(I)) GO TO 101
ISAV=I11(I)
11(I1)=IA + 1F1X(SIGN(1.,XI(I1))-X(IA,1)))
1T=0
DO 102 J1=1,2
J1X=J1 + 11(I11) - 1
DO 102 I1J1=1,2
1JX=J1X + (I1J1 + 11(2) - 2)*NZ
102 K=1,2
KK=K-1 + 11(3)
102 L=1,2
LX=L-1+II(4)
IF(XB(IJX,KX,LX).GT. -100.) IT=IT+1
CONTINUE
IF(IT.GT. ITC) GO TO 6
II(I)=ISAV
CONTINUE

PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS

DO 103 I=1,4
  I2=I*2
  I1=I2-1
  IA=II(I)
  XVAL(I1)=X(IA,I)
  XVAL(I2)=X(IA+1,I)
CONTINUE
CALL DTERPS(II,XB,VAL,NZ)
XBAR=DTERPI(4,X1,XVAL,VAL,-100.,W)
CALL DTERPS(II,SW,VAL,NZ)
SIGW2=DTERPI(-4,X1,XVAL,VAL,-100.,W)
CALL DTERPS(II,SP,VAL,NZ)
SIGP2=DTERPI(-4,X1,XVAL,VAL,-100.,W)

CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED TO THE USUAL DIMENSIONAL FORM

SCL=U0*H**(M+1.)/SCLU
XBAR=SCL*EXP(XBAR)
SIGW2=SCL*SIGW2*EXP(SIGW2)
SIGP2=2.*DXZ0*H**EXP(SIGP2)
SIGW2=SIGW2+SIGP2
SIGP2=DYXZ0*SIGP2
999 RETURN
END

*********************************************************************************** SUBFILE 26 ***********************************************************************************

SUBROUTINE DTERPS(II,X,VAL,NZ)
IMPLICIT INTEGER*4 (I-N)
DIMENSION X(81,4,4),VAL(1),II(1)

PURPOSE

TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH LOWEST CORNER INDICES GIVEN
INPUT

II  SIMPLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE
LOWEST CORNER OF THE CUBE

X  A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED
V^204 VALUES TO BE SET UP. THE FIRST INDEX IS THE COLLAPSED
INDEX FOR THE FIRST TWO INDICES OF A FOUR DIMENSIONAL
ARRAY

NZ  THE RANGE OF THE FIRST INDEX OF THE FOUR DIMENSIONAL
ARRAY

OUTPUT

VAL  SIMPLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X
FOR THE 16 CORNER POINTS OF THE CUBE

CALLED BY MOMENT

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

M=0
DO 104 L=1,2
LX=L + II(4) - 1
DO 103 K=1,2
KX=K + II(3) - 1
DO 102 J1=1,2
J1X=(J1 + II(2) - 2)*NZ
DO 101 IJ=1,2
IJX=J1X + IJ + II(1) -1
M=M+1
VAL(M) = X(IJX,KX,LX)
101 CONTINUE
102 CONTINUE
103 CONTINUE
104 CONTINUE
RETURN
END

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

FUNCTION DTERPI(NDIM,X1,XVAL,VAL,VMIN,WORK)
IMPLICIT INTEGER*4 (I-N)

PURPOSE

74
PERFORMS AN N-DIMENSIONAL LINEAR INTERPOLATION

INPUT

NDIM - THE NUMBER OF DIMENSIONS. (- DONT RECALCULATE WEIGHTS)
XI  - THE POINT IN THE HYPERSPACE AT WHICH THE INTERPOLATED
      VALUE IS DESIRED. XI MUST BE A VECTOR OF ATLEAST NDIM
      IN LENGTH.
XVAL - THE COORDINATE VALUES AT THE CORNERS OF THE HYPERCUBE.
      THE VECTOR MUST BE SET UP LIKE A TWO-DIMENSIONAL ARRAY
      (2 x NDIM), WHERE THE FIRST SUBSCRIPT REFERS TO THE
      HYPERCUBE COORDINATES IN THE SECOND SUBSCRIPTS
      DIRECTION.
VAL  - THE FUNCTIONAL VALUES AT THE CORNERS OF THE HYPERCUBE
      SURROUNDING XI. THIS VECTOR MUST BE FILLED EQUIVALENT
      TO AN NDIM ARRAY WITH EACH DIMENSION AS 2. THE SIZE
      OF VAL SHOULD BE ATLEAST 2**NDIM.
DMIN - A MINIMUM VALUE OF VAL FOR WHICH THE INTERPOLATION
      WILL USE A CORNER VALUE.
WORK - A WORK VECTOR OF ATLEAST NDIM*2. USE TO STORE COORD-
      INATE WEIGHTS.

OUTPUT

RETURNS INTERPOLATED VALUE OF VAL AT XI
CALLED BY MOMENT

***************************************************************
DIMENSION XI(1),XVAL(1),VAL(1),WORK(1)
***************************************************************

SET UP THE COORDINATE WEIGHTS

NDI=1ABS(NDIM)
IF(NDIM .LT. 0) GO TO 1
DO 100 I=1,NDI
   12=1*2
   II=I2-1
   WORK(I2)=(XI(I)-XVAL(II))/(XVAL(I2)-XVAL(II))
   WORK(I)=1. - WORK(I2)
100 CONTINUE

INTERPOLATE - USE BINARY COUNTER FOR COORDINATE LOCATION

DI=SIGMA(0.,SUM)
SUM=0.
ND=2*NDI
DO 201 I=1,ND
IF(VAL(I).LT.WMIN) GO TO 201
L=1-1
WEIGHT=1.
DO 200 J=1,NDI
N=MOD(L,2) + J*2 - 1
WEIGHT=WEIGHT*WORK(N)
L=L/2
200 CONTINUE
SUM=SUM+WEIGHT
DTERPI=DTERPI+WEIGHT*VAL(I)
201 CONTINUE
IF(SUM.EQ.0.) GO TO 202
DTERPI=DTERPI/SUM
RETURN
C
202 STOP
END
C
SUBFILE 28
SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)
IMPLICIT INTEGER*4(I-N)
PURPOSE
TO COMPUTE THE GENERALIZED GREENS FUNCTION
USES GREENI
SEE GREENI FOR ARGUMENT LIST
REAL X,N
CолнNDPRM/NDXO,DYXO,DZC,UC,M,N,ZINV
C
IF(I1.EQ.1.) GO TO 2
X2=2.*X
AT=ALPHA*X
T0=0.
IF(AT.GE.Z1) RETURN:
CALL GREENI((((Z+AT)**X2,Z1**X2,X2*X2*T0,(N-1.)/X2,T1,IER)
T1=T1*X2*Z1**(1.-N)
U=1.
T=0.
IF(ABS(ALPHA).LT.1.E-4) GO TO 1
ZM2=2-Z1*AT
X2=0.
AN1=ALPHA*X2
ZM2=Z1**X2 = (Z1*AT)**X2
ARG=(-AN1*ZM2/2.)/(4.*ZM2)
IF(ARG.LT.-70.) GO TO 3
T2=SORT(AN1/(4.*3.1415926*ZM2)) EXP(ARG)
C
76
3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN

C CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY
C
CALL GREENI(Z+AT,Z1,T,0.,TIU,IER)
X2=2.
ANI=ALPHA*X2
ZN2=Z1**X2 - (Z1-AT)**X2
T2=0.
ARG=(-ANI*ZN2*ZN2)/(4.*ZN)
IF(ARG .LT. -70.) GO TO 4
T2=SQRT(ANI/(4.*3.1415926*ZN2))**EXP(ARG)
4 IF(TIU.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1
CALL GREENI(Z,Z1,T,ALPHA,T0,IER)
U=(G-T2)/(TIU-T2)
IF(U .LT. 0.) U=0.
IF(U .GT. 1.) U=1.

C COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO
C
TO=U*T1 + (1.-U)*T2
RETURN
2 CALL GREENI(Z,Z1,T,ALPHA,T0,IER)
RETURN
END

C ********************************** SUBFILE 29 **********************************

SUBROUTINE GREENI

PURPOSE
COMPUTE THE I BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT
AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREEN'S
FUNCTION FOR THE WIND DIFFUSION EQUATION

USAGE
CALL GREENI(Z,Z1,T,NU,B1,IER)

DESCRIPTION OF PARAMETERS
Z,Z1,T -THE ARGUMENTS OF THE FUNCTION DESIRED
NU -THE ORDER OF THE I BESSEL FUNCTION
B1 -THE RESULTANT BESSEL FUNCTION
IER -RESULTANT ERROR CODE WHERE
IER=1 EXPONENTIAL UNDERFLOW (NON-FATAL), B1 SET TO 0.0
IER=0 NO ERROR
IER=1 NU NEAR NEGATIVE INTEGER
IER=2 OVERFLOW IN GAMMA
IER=3 UNDERFLOW, B1 .LT. 1.E-32, B1 SET TO 0.0
IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N
IER=5 X IS NEGATIVE

REMARKS
NU IS A REAL NUMBER
N AND X MUST BE .GE. ZERO
THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE
I BESSEL FUNCTION FOR INTEGER ORDERS. THE CHANGE REQUIRES THE
USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE
SERIES. THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME
RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO
UNCHANGED. BEST IS IN THE IBM SYSTEM/360 SCIENTIFIC SUBROUTINE
PACKAGE. MODIFICATIONS MADE BY D. DvOE, AERODYNE RESEARCH,

SUBROUTINES AND FUNCTIONS REQUIRED
GAMMA WHICH COMPUTES THE GAMMA FUNCTION

METHOD
COMPUTES I BESSEL FUNCTION USING SERIES OR ASYMPTOTIC
APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT.

CALLED BY MOMENT

SUBROUTINE GREEN1(Z, Z1, T, NU, BI, IER)
IMPLICIT INTEGER*4 (I-N)
REAL NU
XX=2.*SQRT(Z*Z1)/T

CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT
IER=0
BI=1.0
10 IF(NU)O, 15, 10
15 IF(X)160, 20, 20
17 IF(X)160, 17, 20
16 IF(ARG .LT. -80.) GO TO 170
17 ARG=-(Z+Z1)/T
BI=EXP(ARG)/T
RETURN

DEFINE TOLERANCE
20 TOL=1.E-3

IF ARGUMENT GT 12 AND CT NU, USE ASYMPTOTIC FORM
30 IF(X-12.)40, 40C

40 XX=X/2.
N=INT(NU)
FN=N
R=NU-FN
CALL GAMMA(1.+NU, GR, IER)
IF(IER .EQ. 0) GO TO 60
50 BI=0.0
RETURN
60 TERM=1./GR
70 BI=BI+TERM
XX=XX*XX

COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERANCE
DO 90 K=1,1000
IF(ABS(TERM)-ABS(BI*TOL))95, 95, 80
80 FX=K
FK=FX*(NU+FK)
TERM=TERM*(XX/FK)
90 BI=BI+TERM
95 ARG=-(Z+Z1)/T

78
IF(ARG LT -50.) GO TO 170
BI = BI * (21/N)**NU * EXP(ARG)/T

RETURN BI AS ANSWER

100 RETURN

C X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION

110 FK = 4. * NU * NU
115 XX = 1. / (8. * X)
TERM = 1.
BI = 1.
DO 130 K = 1, 30
IF(ABS(TERM)-ABS(BI*TOL)) 140, 140, 120
FK = (2*K-1)**2
TERM = TERM***(FK-FK)/FLOAT(K)
130 BI = BI + TERM

SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES

GO TO 40
140 PI = 3.141592653
ARG = X - (Z + Z 1)/T
IF(ARG LT -80.) GO TO 170
BI = BI * (21/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T
GO TO 100
160 IER = 5
GO TO 100
170 BI = 0. C
GO TO 50
END

************************************************************************************** SUBFILE 30 **************************************************************************************

SUBROUTINE GAMMA

PURPOSE
COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT

USAGE
CALL GAMMA(XX, GX, IER)

DESCRIPTION OF PARAMETERS
XX - THE ARGUMENT FOR THE GAMMA FUNCTION
GX - THE RESULTANT GAMMA FUNCTION
IER - THE RESULTANT ERROR CODE WHERE
IER = 0 NO ERROR
IER = 1 XX IS WITHIN .00001 OF BEING A NEGATIVE INTEGER
IER = 2 XX GT 57, OVERFLOW, GX SET TO 1.E72

COMMENTS
NONE

SUBROUTINES AND FUNCTIONS
NONE

METHOD
THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION
BY C. HASTINGS, JR., "APPROXIMATIONS FOR DIGITAL COMPUTERS",
PRINCETON UNIVERSITY PRESS, 1955
SUBROUTINE GMNA(XX, GX, IER)
IMPLICIT INTEGER*4 (I-N)
IF(XX-5.7) 6, 6, 4
IER=2
GX=1.E32
RETURN
6 IF(X-2.0) 50, 50, 15
10 IF(X-2.0) 110, 110, 15
15 X=X-1.0
G X=GX*
GO TO 10
50 IF(X-1.0) 60, 120, 110
SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO
60 IF(X-ERR) 62, 62, 80
62 Y=FLOAT(INT(X))-X
IF(ABS(Y)-ERR) 130, 130, 70
X NOT NEAR A NEGATIVE INTEGER OR ZERO
70 IF(X-1.0) 60, 60, 110
11 GX=G *X
X=X+1.0
GO TO 70
110 Y=Y(-0.57715174*0.95866406+Y*(-0.5768218+Y*(0.032212+
L*Y*(-0.5664729+Y*(0.2548206+Y*(-0.0514993C)))))
GX=G*Y
120 RETURN
120 IER=1
RETURN
END
SUBROUTINE RLS(N, XL, XU, YIMIN, DEL, ACCURC, WK, ND)
IMPLICIT INTEGER*4 (I-N)
NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S
USING THE RUNGE-KUTTA-MERSON TECHNIQUE
INPUT PARAMETERS:
N - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS
XL - INITIAL ABCISSA OF THE INTERVAL
XU - THE FINAL ABCISSA OF THE INTEGRATION INTERVAL
Y - A SINGLELY DIMENSIONED ARRAY OF LENGTH N. WHEN
RHS IS CALLED IT MUST CONTAIN THE VALUES OF
THE DEPENDENT VARIABLES AT XL. UPON RETURN
TO THE CALLING PROGRAM Y CONTAINS THE VALUES
OF THE DEPENDENT VARIABLES AT X.
YMIN - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE
INTEGRATION
DEL - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON
RETURN TO THE CALLING PROGRAM DEL CONTAINS THE
FINAL STEP SIZE USED. THIS VALUE SHOULD BE USED
IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION.
DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS BEEN HALVED BELOW HMIN.

ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING THE STEP SIZE.

WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS USED FOR A WORK ARRAY.

ND - THE DIMENSION OF ARRAYS Y AND WK.

IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE DEFINING THE DIFFERENTIAL EQUATIONS. THE SUBROUTINE STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP).

WHERE

N - THE NUMBER OF EQUATIONS
X - THE INDEPENDENT VARIABLE
Y - SIMPLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES
YP - SIMPLY DIMENSIONED ARRAY OF THE RATE OF Y AT X
YP(1) = dY(1)/dX

DIMENSION Y(ND),WK(ND,6)
LOGICAL FIRST,QUIT

SET UP NEEDED VARIABLES UPON ENTRY

XN=XL
H=DEL
FIRST=.TRUE.
QUIT=.FALSE.

CHECK IF XN IS CLOSE TO XU

20 IF(XN+H .LT. XU) GO TO 30
DEL=H
H=XC-XU
QUIT=.TRUE.
IF(FIRST) DEL=H

MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL

30 CALL DIFEQ(N,XN,Y,WK(1,1))

PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM

40 H3=H/3.
DO 50 I=1,N
WK(I,3)=H3*WK(I,1)
50 WK(I,6)=Y(I)+WK(I,3)
CALL DIFEQ(N,XN+H3,WK(I,3),WK(I,2))
DO 60 I=1,N
WK(I,6)=Y(I)+WK(I,3)+H3*WK(I,2))/2.
CALL DIFEQ(N,XN+H6,WK(I,6),WK(I,2))
DO 70 I=1,N
WK(I,5)=H3*WK(I,2)
70 WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/6.
CALL DIFEQ(N,XN+H2,WK(I,6),WK(I,2))
DO 80 I=1,N
WK(I,5)=H3*WK(I,2)
80 WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2.
CALL DIFEQ(N,XN+H2,WK(I,6),WK(I,2))

FIND THE LARGEST RELATIVE ERROR
TEST=0.
DO 90 I=1,N
YX=Y(I)
IF(YX,EQ,0.) YX=ACCURC
E=(WK(I,3)-9.*WK(I,4))/2.+4.*WK(I,5)+3*WK(I,2)/2.)/YX
90 TEST=MAX1(TEST,ABS(E))
FIRST=.FALSE.
IF(TEST.LT.ACCURC) GO TO 100
C IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP
C SIZE AND TRY AGAIN.
C
H=H/2.
IF(H.LT.HMIN) GO TO 10
QUIT=.FALSE.
GO TO 40
C TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO
C SET UP FOR THE NEXT INTERVAL
C
100 XN=XN-H
DO 110 I=1,N
110 Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+3*WK(I,2))/2.
C CHECK FOR STEP SIZE DOUBLING. DOUBLE IF LARGEST RELATIVE
C ERROR IS 32 TIMES LESS THAN ACCURC.
C
IF(.NOT.(TEST.GE.ACCURC/32. .OR. QUIT)) H=H*4!
IF(.NOT. QUIT) GO TO 20
RETURN
C THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM.
C REPORT THIS AND ERROR OUT.
C
20 CONTINUE
DEL=.0.
RETURN
END