



Techniques of Water-Resources Investigations of the United States Geological Survey

Chapter B7

ANALYTICAL SOLUTIONS FOR ONE-, TWO-, AND THREE-DIMENSIONAL SOLUTE TRANSPORT IN GROUND-WATER SYSTEMS WITH UNIFORM FLOW

By Eliezer J. Wexler

Book 3

APPLICATIONS OF HYDRAULICS

Attachment 3.—Subroutine Listing and Data File GLQ.PTS

Subroutine EXERFC
Subroutine GLQPTS
Subroutine OFILE
Subroutine TITLE
Subroutine PLOT1D
Subroutine PLOT2D
Subroutine PLOT3D
Subroutine CNTOUR
DATA FILE GLQ.PTS

```

C
C *****
C *
C *
C *
C *
C *
C *
C *****
C
C SUBROUTINE EXERFC (X,YY,Z)
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C DIMENSION P1(5),Q1(5),P2(9),Q2(9),P3(6),Q3(6)
C
C THIS ROUTINE USES RATIONAL CHEBYSHEV APPROXIMATIONS
C FOR EVALUATING THE ERROR FUNCTION AND COMPLEMENTARY
C ERROR FUNCTION IN ORDER TO EVALUATE THE PRODUCT OF
C EXP(X) AND ERFC(Y)
C
C DATA P1/3.209377589138469472562D03,3.774852376853020208137D02,
1 1.138641541510501556495D02,3.161123743870565596947D0,
2 1.857777061846031526730D-01/
C DATA Q1/2.844236833439170622273D03,1.282616526077372275645D03,
1 2.440246379344441733056D02,2.360129095234412093499D01,
2 1.0D0 /
C DATA P2/1.23033935479799725272D03,2.05107837782607146532D03,
1 1.71204761263407058314D03,8.81952221241769090411D02,
2 2.98635138197400131132D02,6.61191906371416294775D01,
3 8.88314979438837594118D00,5.64188496988670089180D-01,
4 2.15311535474403846343D-08/
C DATA Q2/1.23033935480374942043D03,3.43936767414372163696D03,
1 4.36261909014324715820D03,3.29079923573345962678D03,
2 1.62138957456669018874D03,5.37181101862009857509D02,
3 1.17693950891312499305D02,1.57449261107098347253D01,
4 1.0D0 /
C DATA P3/-6.58749161529837803157D-04,-1.60837851487422766278D-02,
1 -1.25781726111229246204D-01,-3.60344899949804439429D-01,
2 -3.05326634961232344035D-01,-1.63153871373020978498D-02/
C DATA Q3/2.33520497626869185443D-03,6.05183413124413191178D-02,
1 5.27905102951428412248D-01,1.87295284992346047209D00,
2 2.56852019228982242072D00,1.0D0/
C
C IF(YY.EQ.0.0D0) Z=DEXP(X)
C IF(YY.EQ.0.0D0) RETURN
C Y=DABS(YY)
C
C FOR 0.0 < Y < .46875
C IF (Y.GT.0.46875D0) GO TO 20
C SUMP=0.0D0
C SUMQ=0.0D0
C DO 10 I=1,5
C Y2I=Y**(2*(I-1))
C SUMP=SUMP+P1(I)*Y2I
C SUMQ=SUMQ+Q1(I)*Y2I

```

10	CONTINUE	54
	ERF=Y*SUMP/SUMQ	55
	IF(YY.LT.0.0) ERF=-ERF	56
	ERFCY=1.000-ERF	57
	Z=DEXP(X)*ERFCY	58
	RETURN	59
C		60
C	FOR 0.0 < Y < .46875	61
20	IF (Y.GT.4.000) GO TO 40	62
	SUMP=0.000	63
	SUMQ=0.000	64
	DO 30 I=1,9	65
	YI=Y**(I-1)	66
	SUMP=SUMP+P2(I)*YI	67
	SUMQ=SUMQ+Q2(I)*YI	68
30	CONTINUE	69
	Z=DEXP(X-Y*Y)*SUMP/SUMQ	70
	IF(YY.LT.0.000) Z=2.000*DEXP(X)-Z	71
	RETURN	72
40	SUMP=0.000	73
	SUMQ=0.000	74
	DO 50 I=1,6	75
	Y2I=Y**(-2*(I-1))	76
	SUMP=SUMP+P3(I)*Y2I	77
	SUMQ=SUMQ+Q3(I)*Y2I	78
50	CONTINUE	79
	SQRTPI=0.564189583547756286948100	80
	Z=SQRTPI+SUMP/(Y*Y*SUMQ)	81
	Z=DEXP(X-Y*Y)*Z/Y	82
	IF(YY.LT.0.000) Z=2.000*DEXP(X)-Z	83
	RETURN	84
	END	85

```

C
C *****
C *
C *
C *
C *
C *
C *
C *****
C
C SUBROUTINE GLQPTS (N)
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C CHARACTER*1 SKIP
C COMMON /GLPTS/ WN(256),ZN(256)
C COMMON /IOUNIT/ IN,IO
C
C THIS ROUTINE READS THE NORMALIZED ROOTS ZN(I) AND WEIGHTS WN(I)
C OF THE LEGENDRE POLYNOMIALS FROM THE DATA FILE 'GLQ.PTS'
C
C N IS THE NUMBER OF INTEGRATION POINTS AND CAN ONLY HAVE A
C VALUE OF EITHER 4,20,60,104,OR 256
C
C IN2=77
C OPEN(IN2,FILE='GLQ.PTS',STATUS='OLD')
C
C SKIP LINES IN FILE UNTIL CORRECT COEFFICIENTS ARE REACHED
C ISKIP=-1
C IF(N.EQ.4) ISKIP=7
C IF(N.EQ.20) ISKIP=9
C IF(N.EQ.60) ISKIP=15
C IF(N.EQ.104) ISKIP=31
C IF(N.EQ.256) ISKIP=57
C IF (ISKIP.EQ.-1) WRITE(IO,201)
C IF (ISKIP.EQ.-1) STOP
C DO 60 I=1,ISKIP
60 READ(IN2,101) SKIP
C
C READ IN ZN(I) AND WN(I), FOUR VALUES PER LINE
C NC=N/8
C IF (MOD(N,8).NE.0) NC=NC+1
C DO 80 I=1,NC
C K=(I-1)*8-1
80 READ(IN2,102) (ZN(K+J*2),J=1,4)
C CONTINUE
C DO 100 I=1,NC
C K=(I-1)*8-1
C READ(IN2,102) (WN(K+J*2),J=1,4)
100 CONTINUE
C
C FILL IN THE SYMMETRIC TERMS
C DO 120 J=2,N,2
C J1=J-1
C ZN(J)=-ZN(J1)
120 WN(J)=WN(J1)

```

	CLOSE(IN2)	54
	RETURN	55
C		56
C	FORMAT STATEMENTS	57
101	FORMAT(A1)	58
102	FORMAT(4D20.0)	59
201	FORMAT(1H0,20X,'***** ERROR IN ROUTINE GLQPTS *****'/	60
	1 1H ,20X,'NO. OF ROOTS SPECIFIED MUST EQUAL 4,20,60,104 OR 256')	61
	END	62


```

C
C *****
C *
C *          SUBROUTINE TITLE
C *
C *          VERSION CURRENT AS OF 10/01/87
C *
C *****
C
SUBROUTINE TITLE
CHARACTER*1 LINE1(60),EQUAL,BLANK
CHARACTER DATE*16,TIME*8
CHARACTER*60 LINE
CHARACTER*61 TITLE,T1
COMMON /IOUNIT/ IN,IO
COMMON /TITLES/ TITLE(4)
DATA EQUAL/'='/,BLANK/' '/
DATA TITLE/4*'
1      '/'
C
C      THIS ROUTINE CREATES A TITLE BOX ON THE FIRST PAGE OF
C      PROGRAM OUTPUT. THE ROUTINE READS AND PRINTS ALL DATA
C      CARDS UNTIL IT ENCOUNTERS AN '=' IN COLUMN 1. THE FIRST 4
C      LINES READ IN ARE ALSO USED AS TITLES ON PLOTS.
C
CALL TIME$(TIME)
CALL DATE$(DATE)
WRITE(IO,201)
DO 10 L=1,60
READ(IN,101,END=20) LINE
IF (LINE(1:1).EQ.EQUAL) GOTO 60
T1=LINE
C      STRIP OFF TRAILING BLANKS AND CENTER LINE
DO 15 N=1,60
NN=61-N
15  IF(LINE(NN:NN).NE.BLANK) GOTO 20
20  NN1=NN+1
T1(NN1:NN1)='$'
IF(L.LT.5) TITLE(L)=T1
NS=(60-NN)/2
IF(NS.EQ.0) GO TO 35
DO 30 I=1,60
30  LINE1(I)=BLANK
35  NS1=NS+1
DO 40 I=1,NN
40  LINE1(NS+I)=LINE(I:I)
10  WRITE(IO,202) (LINE1(I),I=1,60)
60  WRITE(IO,203) DATE,TIME
RETURN
C
C      FORMAT STATEMENTS
101  FORMAT (A60)
201  FORMAT(1H1//////////1H ,16X,68(1H*))

```


202	FORMAT(1H ,16X,1H*,66X,1H*/1H ,16X,1H*,3X,60A1,3X,1H*)	54
203	FORMAT(1H ,16X,1H*,66X,1H*/1H ,16X,1H*,12X,'PROGRAM RUN ON ',	55
	1 A16,' AT ',A8,11X,1H*/1H ,16X,1H*,66X,1H*/1H ,16X,68(1H*)	56
	2 /1H1)	57
	END	58

```

C
C *****
C *
C *          SUBROUTINE PLOT1D
C *
C *          VERSION CURRENT AS OF 10/01/87
C *
C *****
C
SUBROUTINE PLOT1D (X,C,NX,T,IT,NT,TUNITS,LUNITS,XSCLP)
COMMON /XP/ XPOS,YPOS,XPOS1,YPOS1,XAXIS,YAXIS
COMMON /TITLES/ TITLE(4)
DIMENSION X(NX),C(NX)
COMMON ID
CHARACTER*10 LUNITS,TUNITS
CHARACTER*17 LAB
CHARACTER*27 LAB1
CHARACTER*26 LABX
CHARACTER*36 LABX1
CHARACTER*24 LABY
CHARACTER*61 TITLE

C
C THIS ROUTINE PLOTS CONCENTRATION VS. DISTANCE AT EACH OF THE
C TIMES SPECIFIED IN THE INPUT DATA. THE ROUTINE USES DISSPLA
C SOFTWARE PLOT CALLS.
C
C INITIALIZE PLOT - SCALE BASED ON MAXIMUM X-DISTANCE
HITE=0.1
IF(IT.EQ.1) THEN
CALL COMPRS
CALL SETCLR ('BLUE')
X1=X(NX)-X(1)
XAXIS=INT(X1/XSCLP)
X11=X1/XSCLP
IF((X11-XAXIS).GT.0.0) XAXIS=XAXIS+1.0
YAXIS=10.0
XPM=XAXIS+1.5
YPM=12.2
CALL PAGE(XPM,YPM)
CALL AREA2D (XAXIS,YAXIS)
CALL HEADIN (TITLE(1),100,1.,4)
CALL HEADIN (TITLE(2),100,1.,4)
CALL HEADIN (TITLE(3),100,1.,4)
CALL HEADIN (TITLE(4),100,1.,4)
C LABEL AXES
LABX='DISTANCE ALONG X-AXIS, IN '
LABX1=LABX//LUNITS
LABY='NORMALIZED CONCENTRATION'
CALL XNAME (LABX1,36)
CALL YNAME (LABY,24)
C DRAW AND NUMBER AXES
CALL INTAXS
CALL YAXANG (0.)

```

	CALL XREVTK	54
	CALL YREVTK	55
	XMIN=X(1)	56
	XMAX=XSCLP*XAXIS	57
	YMAX=1.0	58
	CALL GRAF(XMIN,XSCLP,XMAX,0.0,0.1,YMAX)	59
	CALL RESET('XREVTK')	60
	CALL RESET('YREVTK')	61
C	DRAW EXTRA AXIS TO CLOSE BOX	62
	CALL XNONUM	63
	CALL XGRAXS(XMIN,XSCLP,XMAX,XAXIS,' ',1,0.0,YAXIS)	64
	CALL YNONUM	65
	CALL YGRAXS(0.0,0.1,YMAX,YAXIS,' ',1,XAXIS,0.0)	66
	CALL RESET('XNONUM')	67
	CALL RESET('YNONUM')	68
C	BEGIN LEGEND	69
	XPOS=XAXIS-.85*HITE*(27+4)-.1	70
	YPOS=YAXIS-.1-2.0*HITE	71
	CALL HEIGHT (HITE)	72
	LAB='ELAPSED TIME, IN	73
	LAB1=LAB//TUNITS	74
	CALL MESSAG (LAB1,27,XPOS,YPOS)	75
	YPOS=YPOS-.5*HITE	76
C	BLANK OUT AREA FOR MESSAGE	77
	WIDE=HITE*0.85*35.	78
	HIGH=HITE*1.5*(NT+3)	79
	XPOS=XAXIS-WIDE-0.1	80
	YPOS1=YAXIS-HIGH-0.1	81
	CALL BLREC(XPOS,YPOS1,WIDE,HIGH,1.0)	82
	CALL BLKEY(ID)	83
	XPOS=XAXIS-2.75	84
	END IF	85
C		86
C	DRAW PLOT OF C VS X	87
	CALL MARKER(IT)	88
	CALL CURVE (X,C,NX,1)	89
	CALL MARKER(IT)	90
C	PLACE LABEL IN BOX	91
	CALL BLOFF(ID)	92
	YPOS=YPOS-1.5*HITE	93
	XPOS1=XPOS+3*.85*HITE	94
	XP=XPOS1*XSCLP	95
	YP=(YPOS+0.05)/10.	96
	CALL CURVE (XP,YP,1,-1)	97
	XPOS1=XPOS+6*.85*HITE	98
	CALL MESSAG ('TIME =',6,XPOS1,YPOS)	99
	IPL=104	100
	CALL REALNO (T,IPL,'ABUT','ABUT')	101
	CALL BLON(ID)	102
C		103
C	CLOSE PLOT FILE	104
	IF(IT.EQ.NT) THEN	105
	CALL ENDPL (0)	106

```
CALL DONEPL 107  
END IF 108  
RETURN 109  
END 110
```

```

C
C *****
C *
C *          SUBROUTINE PLOT2D          *
C *
C *          VERSION CURRENT AS OF 11/30/88      *
C *
C *****
C
C SUBROUTINE PLOT2D (XP,YP,CP,TP,DELTA,NX,NY,NXY,NXY2,IT,NT,IPLT,
1 TUNITS,LUNITS,XSCLP,YSCLP,XPC,YPC,IFLAG)
C CHARACTER*10 TUNITS,LUNITS
C CHARACTER*26 LABX,LABY
C CHARACTER*36 LABX1,LABY1
C CHARACTER*61 TITLE
C DIMENSION XP(NX),YP(NY),CP(NXY),XPC(50),YPC(50),IFLAG(NXY2)
C COMMON /IUNIT/ IN,IO
C COMMON /TITLES/ TITLE(4)
C
C THIS ROUTINE INITIALIZES A CONTOUR PLOT ON THE RECTANGULAR GRID
C DEFINED IN THE X-Y PLANE BY THE X AND Y VALUES READ IN. ONE
C SUBPLOT IS GENERATED FOR EACH TIME VALUE. THE ROUTINE USES
C DISSPLA (ISCO) SOFTWARE SUBROUTINE CALLS.
C
C CALCULATE PLOT SIZE AND DRAW BORDER
XSPC=1.5
YSPC=2.0
X1=XP(NX)-XP(1)
XAXIS=INT(X1/XSCLP)
IF(AMOD(X1,XSCLP).GT.0.0) XAXIS=XAXIS+1.0
Y1=YP(NY)-YP(1)
YAXIS=INT(Y1/YSCLP)+1.0
IF(AMOD(Y1,YSCLP).GT.0.0) YAXIS=YAXIS+1.0
IF(IT.EQ.1) THEN
  CALL COMPRS
  XPM=(XAXIS+XSPC)*NT+XSPC
  YPM=YAXIS+YSPC
  CALL PAGE(XPM,YPM)
END IF
C CHOOSE PLOT SIZE BASED ON MAXIMUM COORDINATE VALUES
XORIG=(IT-1)*(XAXIS+XSPC)+XSPC
YORIG=0.75
CALL SETCLR ('BLUE')
CALL PHYSOR(XORIG,YORIG)
CALL AREA2D (XAXIS,YAXIS)
IF(IT.EQ.1) THEN
  CALL HEADIN (TITLE(1),100,1.,4)
  CALL HEADIN (TITLE(2),100,1.,4)
  CALL HEADIN (TITLE(3),100,1.,4)
  CALL HEADIN (TITLE(4),100,1.,4)
END IF
C ROTATE Y VALUES, PUT TICK MARKS ON INSIDE, AND DEFINE AXES LABEL
CALL INTAXS

```

```

CALL YAXANG (0.) 54
CALL XREVTK 55
CALL YREVTK 56
LABX='DISTANCE ALONG X-AXIS, IN ' 57
LABY='DISTANCE ALONG Y-AXIS, IN ' 58
LABX1=LABX//LUNITS 59
LABY1=LABY//LUNITS 60
C DRAW AND LABEL AXES 61
CALL XNAME(LABX1,36) 62
CALL YNAME(LABY1,36) 63
XMIN=XP(1) 64
YMIN=YP(1) 65
XMAX=XSCLP*XAXIS + XMIN 66
YMAX=YSCLP*YAXIS + YMIN 67
CALL GRAF(XMIN,XSCLP,XMAX,YMIN,YSCLP,YMAX) 68
C DRAW EXTRA AXIS TO CLOSE BOX 69
CALL RESET('XREVTK') 70
CALL RESET('YREVTK') 71
CALL XNONUM 72
CALL YNONUM 73
CALL XGRAXS(XMIN,XSCLP,XMAX,XAXIS,' ',1,0.0,YAXIS) 74
CALL YGRAXS(YMIN,YSCLP,YMAX,YAXIS,' ',1,XAXIS,0.0) 75
CALL RESET('XNONUM') 76
CALL RESET('YNONUM') 77
C PRINT TITLE 78
HITE=(XAXIS-1.0)/(55.*.86) 79
IF(HITE.GT.0.14) HITE=0.14 80
CALL HEIGHT (HITE) 81
YP3=YAXIS-0.07-1.5*HITE 82
CALL MESSAG('NORMALIZED CONCENTRATION AT TIME =$',100,0.5,YP3) 83
IPL=3 84
IF(AMOD(TP,0.01).EQ.0.0) IPL=2 85
IF(AMOD(TP,0.1).EQ.0.0) IPL=1 86
IF((TP-INT(TP)).EQ.0.0) IPL=0 87
CALL REALNO(TP,IPL,'ABUT','ABUT') 88
CALL MESSAG(TUNITS,10,'ABUT','ABUT') 89
C COUNT NUMBER OF DIGITS IN CONTOUR LABEL 90
YP3=YP3-1.5*HITE 91
CALL MESSAG('CONTOUR INTERVAL =$',100,0.5,YP3) 92
IPL=3 93
IF(AMOD(Delta,0.01).EQ.0.0) IPL=2 94
IF(AMOD(Delta,0.1).EQ.0.0) IPL=1 95
CALL REALNO(Delta,IPL,'ABUT','ABUT') 96
CALL MESSAG('C/Co$',100,'ABUT','ABUT') 97
C 98
C CALL ROUTINE THAT ACTUALLY DOES THE CONTOURING 99
CALL CNTOUR(XP,YP,CP,DELTA,NX,NY,NXY,NXY2,XSCLP,YSCLP,XPC,YPC, 100
1 IFLAG,IPL) 101
C 102
C SUBPLOT IS FINISHED 103
CALL ENDGR(0) 104
IF (IT.EQ.NT) THEN 105
CALL ENDPL (0) 106

```

```
CALL DONEPL  
END IF  
RETURN  
END
```

```
107  
108  
109  
110
```

```

C          1
C          2
C          3
C          4
C          5
C          6
C          7
C          8
C          9
SUBROUTINE PLOT3D (XP,YP,ZP,CP,TP,DELTA,NX,NY,NXY,NXY2,IZ,NZ,IPLT, 10
1 TUNITS,LUNITS,XSCLP,YSCLP,XPC,YPC,IFLAG) 11
CHARACTER*10 TUNITS,LUNITS 12
CHARACTER*26 LABX,LABY 13
CHARACTER*36 LABX1,LABY1 14
CHARACTER*61 TITLE 15
DIMENSION XP(NX),YP(NY),CP(NXY),XPC(50),YPC(50),IFLAG(NXY2) 16
COMMON /IOUNIT/ IN,IO 17
COMMON /TITLES/ TITLE(4) 18
C          19
C          20
C          21
C          22
C          23
C          24
C          25
C          26
C          27
C          28
C          29
C          30
C          31
C          32
C          33
C          34
C          35
C          36
C          37
C          38
C          39
C          40
C          41
C          42
C          43
C          44
C          45
C          46
C          47
C          48
C          49
C          50
C          51
C          52
C          53

```

 * SUBROUTINE PLOT3D *
 * VERSION CURRENT AS OF 11/30/88 *

THIS ROUTINE INITIALIZES A CONTOUR PLOT ON THE RECTANGULAR GRID
 DEFINED IN THE X-Y PLANE BY THE X AND Y VALUES READ IN. ONE
 SUBPLOT IS GENERATED FOR EACH Z VALUE AND A NEW PLOT IS
 GENERATED FOR EACH TIME VALUE. THE ROUTINE USES DISSPLA (ISCO)
 SOFTWARE SUBROUTINE CALLS.
 PLOT SCALING FACTORS (XSCLP,YSCLP) AND CONTOUR INTERVAL (DELTA)
 ARE SPECIFIED IN THE MAIN PROGRAM.

CALCULATE PLOT SIZE AND DRAW BORDER
 XSPC=1.5
 YSPC=2.0
 X1=XP(NX)-XP(1)
 XAXIS=INT(X1/XSCLP)
 IF(AMOD(X1,XSCLP).GT.0.0) XAXIS=XAXIS+1.0
 Y1=YP(NY)-YP(1)
 YAXIS=INT(Y1/YSCLP)+1.0
 IF(AMOD(Y1,YSCLP).GT.0.0) YAXIS=YAXIS+1.0
 IF(IZ.EQ.1) THEN
 CALL COMPRS
 XPM=(XAXIS+XSPC)*NZ+XSPC
 YPM=YAXIS+YSPC
 CALL PAGE(XPM,YPM)
 END IF

CHOOSE PLOT SIZE BASED ON MAXIMUM COORDINATE VALUES
 XORIG=(IZ-1)*(XAXIS+XSPC)+XSPC
 YORIG=0.75
 CALL SETCLR ('BLUE')
 CALL PHYSOR(XORIG,YORIG)
 CALL AREA2D (XAXIS,YAXIS)
 IF(IZ.EQ.1) THEN
 CALL HEADIN (TITLE(1),100,1.,4)
 CALL HEADIN (TITLE(2),100,1.,4)
 CALL HEADIN (TITLE(3),100,1.,4)
 CALL HEADIN (TITLE(4),100,1.,4)


```

END IF 54
C ROTATE Y VALUES, PUT TICK MARKS ON INSIDE, AND DEFINE AXES LABEL 55
CALL INTAXS 56
CALL YAXANG (0.) 57
CALL XREVTK 58
CALL YREVTK 59
LABX='DISTANCE ALONG X-AXIS, IN ' 60
LABY='DISTANCE ALONG Y-AXIS, IN ' 61
LABX1=LABX//LUNITS 62
LABY1=LABY//LUNITS 63
C DRAW AND LABEL AXES 64
CALL XNAME(LABX1,36) 65
CALL YNAME(LABY1,36) 66
XMIN=XP(1) 67
YMIN=YP(1) 68
XMAX=XSCLP*XAXIS + XMIN 69
YMAX=YSCLP*YAXIS + YMIN 70
CALL GRAF(XMIN,XSCLP,XMAX,YMIN,YSCLP,YMAX) 71
C DRAW EXTRA AXIS TO CLOSE BOX 72
CALL RESET('XREVTK') 73
CALL RESET('YREVTK') 74
CALL XNONUM 75
CALL YNONUM 76
CALL XGRAXS(XMIN,XSCLP,XMAX,XAXIS,' ',1,0.0,YAXIS) 77
CALL YGRAXS(YMIN,YSCLP,YMAX,YAXIS,' ',1,XAXIS,0.0) 78
CALL RESET('XNONUM') 79
CALL RESET('YNONUM') 80
C PRINT TITLE 81
HITE=(XAXIS-1.0)/(55*.86) 82
IF(HITE.GT.0.14) HITE=0.14 83
CALL HEIGHT (HITE) 84
YP3=YAXIS-0.07-1.5*HITE 85
CALL MESSAG('NORMALIZED CONCENTRATION AT TIME =$',100,0.5,YP3) 86
IPL=3 87
IF(AMOD(TP,0.01).EQ.0.0) IPL=2 88
IF(AMOD(TP,0.1).EQ.0.0) IPL=1 89
IF((TP-INT(TP)).EQ.0.0) IPL=0 90
CALL REALNO(TP,IPL,'ABUT','ABUT') 91
CALL MESSAG(TUNITS,10,'ABUT','ABUT') 92
YP3=YP3-1.5*HITE 93
CALL MESSAG(' AND AT Z =$',100,0.5,YP3) 94
IPL=3 95
IF(AMOD(ZP,0.01).EQ.0.0) IPL=2 96
IF(AMOD(ZP,0.1).EQ.0.0) IPL=1 97
IF((ZP-INT(ZP)).EQ.0.0) IPL=0 98
CALL REALNO(ZP,IPL,'ABUT','ABUT') 99
CALL MESSAG(LUNITS,10,'ABUT','ABUT') 100
C COUNT NUMBER OF DIGITS IN CONTOUR LABEL 101
YP3=YP3-1.5*HITE 102
CALL MESSAG('CONTOUR INTERVAL =$',100,0.5,YP3) 103
IPL=3 104
IF(AMOD(Delta,0.01).EQ.0.0) IPL=2 105
IF(AMOD(Delta,0.1).EQ.0.0) IPL=1 106

```

```
      CALL REALNO(DELTA,IPL,'ABUT','ABUT')          107
      CALL MESSAG('C/Co$',100,'ABUT','ABUT')       108
C
      CALL ROUTINE THAT ACTUALLY DOES THE CONTOURING 109
C      CALL CNTOUR(XP,YP,CP,DELTA,NX,NY,NXY,NXY2,XSCLP,YSCLP,XPC,YPC,
1 IFLAG,IPL)                                     111
C
      SUBPLOT IS FINISHED                          112
C
      CALL ENDGR(0)                                 113
      CALL ENDGR(0)                                 114
      IF (IZ.EQ.NZ) THEN                            115
        CALL ENDPL (0)                              116
        CALL DONEPL                                 117
      END IF                                         118
      RETURN                                         119
      END                                            120
      END                                            121
```

```

C
C *****
C *
C *
C *
C *
C *
C *
C *****
C
SUBROUTINE CNTOUR (XP, YP, CP, DELTA, NX, NY, NXY, NXY2, XSCLP, YSCLP,
1 XPC, YPC, IFLAG, IPL)
DIMENSION XP(NX), YP(NY), CP(NXY), XPC(50), YPC(50), IFLAG(NXY2)
C
C THIS ROUTINE IS CALLED BY PLOT2D AND PLOT3D TO CONTOUR VALUES
C OF NORMALIZED CONCENTRATION VALUES ON THE RECTANGULAR GRID.
C NUMBER OF SEGMENTS DRAWN BEFORE THE CONTOUR LINE IS LABELED
C (NUM), AND CHARACTER HEIGHT ARE SET HERE, BUT CAN BE
C EASILY MODIFIED.
C XPC, YPC, AND IFLAG ARE WORK ARRAYS USED BY THIS ROUTINE.
C IFLAG MUST BE DIMENSIONED TO TWICE THE NUMBER OF RECTANGULAR
C BLOCKS SINCE EACH BLOCK IS DIVIDED INTO TWO TRIANGLES.
C
NUM=40
HITE=0.10
RAD=57.2957795
C COMPUTE SPACE NEEDED FOR CONTOUR LABEL
CALL HEIGHT (HITE)
CALL NUMODE('NOLEADSPACE')
SPC1=(IPL+2)*HITE
CALL SETCLR ('RED')
C
C FIND MIN AND MAX VALUES AND NUMBER OF CONTOURS
VMIN=1.0E36
VMAX=-1.0E36
DO 10 N=1, NXY
VAL=CP(N)
IF (VAL.GT.VMAX) VMAX=VAL
IF (VAL.LT.VMIN) VMIN=VAL
10 CONTINUE
GDEL=VMAX-VMIN
MAXCNT=GDEL/DELTA
MAXCNT=MAXCNT+1
C
C FIND FIRST CONTOUR VALUE
INC=VMIN/DELTA
VALINC=INC*DELTA
C
C SET UP MASTER LOOP FOR ALL CONTOURS
C EACH RECTANGULAR BLOCK IS DIVIDED INTO TWO TRIANGLES.
C CONTOURS ARE DRAWN BY LINEARLY INTERPOLATING ACROSS EACH
C TRIANGLE.
NTR=(NX-1)*(NY-1)*2
NY2=(NY-1)*2

```

```

DO 20 M=1,MAXCNT                                54
VALINC=VALINC+DELTA                              55
IF(VALINC.GT.VMAX) GOTO 20                       56
C                                                  57
C      INITIALIZE FLAGS ON TRIANGLES WITH CONTOURS PASSING THROUGH 58
IFIRST=0                                          59
DO 30 N=1,NTR                                    60
  N1=(N-1)/NY2                                    61
  N2=(N-(N1*NY2)+1)/2                            62
  NG1=N1*NY+N2                                    63
  NG2=NG1+NY                                      64
  NG3=NG1+1                                       65
  IF(MOD(N,2).EQ.0) THEN                          66
    NG1=NG1+1                                     67
    NG2=NG1+NY-1                                 68
    NG3=NG1+NY                                   69
  END IF                                          70
  IFLAG(N)=0                                      71
  CP1=CP(NG1)                                     72
  CP2=CP(NG2)                                     73
  CP3=CP(NG3)                                     74
  CPMAX=AMAX1(CP1,CP2,CP3)                       75
  CPMIN=AMIN1(CP1,CP2,CP3)                       76
  IF(CPMAX.LT.VALINC .OR. CPMIN.GT.VALINC) GOTO 30 77
  IFLAG(N)=1                                      78
  IF(IFIRST.EQ.0) IFIRST=N                       79
  ILAST=N                                         80
CONTINUE                                         81
C                                                  82
C      LOOP THROUGH ALL FLAGGED TRIANGLES          83
DO 40 N=IFIRST,ILAST                             84
  IF(IFLAG(N).EQ.0) GO TO 40                     85
C                                                  86
C      START UP A NEW CONTOUR SEGMENT             87
ISTART=0                                          88
ICLK=0                                           89
IPT=1                                            90
NEXT=N                                           91
C                                                  92
C      CONTROL LOOP FOR FOLLOWING CONTOUR SEGMENT THROUGH ELEMENTS 93
50 N1=(NEXT-1)/NY2                                94
  N2=(NEXT-(N1*NY2)+1)/2                        95
  IEVEN=0                                         96
  IF(MOD(NEXT,2).EQ.0) IEVEN=1                  97
  NG1=N1*NY+N2                                   98
  NG2=NG1+NY                                     99
  NG3=NG1+1                                     100
  IF(IEVEN.EQ.1) THEN                            101
    NG1=NG1+1                                    102
    NG2=NG1+NY-1                                103
    NG3=NG1+NY                                  104
  END IF                                          105
  CP1=CP(NG1)                                    106

```

	CP2=CP(NG2)	107
	CP3=CP(NG3)	108
	DELV21=CP2-CP1	109
	DELV31=CP3-CP1	110
	DELV32=CP3-CP2	111
	X1=XP(N1)	112
	X2=XP(N1+1)	113
	X3=XP(N1)	114
	Y1=YP(N2)	115
	Y2=YP(N2)	116
	Y3=YP(N2+1)	117
	IF(IEVEN.EQ.1) THEN	118
	X3=XP(N1+1)	119
	Y1=YP(N2+1)	120
	END IF	121
	DELX21=X2-X1	122
	DELX31=X3-X1	123
	DELX32=X3-X2	124
	DELY21=Y2-Y1	125
	DELY31=Y3-Y1	126
	DELY32=Y3-Y2	127
C	RESET FLAG, INCREMENT COUNTER, AND FIND NEIGHBORING ELEMENTS	128
	IFLAG(NEXT)=0	129
	IPT=IPT+1	130
	IUP=NEXT+1	131
	IDN=NEXT-1	132
	ISIDE=NEXT-NY2+1	133
	IF(IEVEN.EQ.1) ISIDE=NEXT+NY2-1	134
C		135
C	SPECIAL CASE 1. CONTOURS ALONG ELEMENT SIDES	136
	IF(CP1.EQ.CP2 .AND. CP1.EQ.VALINC) THEN	137
	NEXT=-1	138
	XPC(1)=X1	139
	YPC(1)=Y1	140
	XPC(2)=X2	141
	YPC(2)=Y2	142
	IF(CP3.NE.CP1) GO TO 60	143
	IPT=4	144
	XPC(3)=X3	145
	YPC(3)=Y3	146
	XPC(4)=X1	147
	YPC(4)=Y1	148
	ELSE IF(CP1.EQ.CP3 .AND. CP1.EQ.VALINC) THEN	149
	NEXT=-1	150
	XPC(1)=X3	151
	YPC(1)=Y3	152
	XPC(2)=X1	153
	YPC(2)=Y1	154
	ELSE IF(CP2.EQ.CP3 .AND. CP2.EQ.VALINC) THEN	155
	NEXT=-1	156
	XPC(1)=X2	157
	YPC(1)=Y2	158
	XPC(2)=X3	159

```

        YPC(2)=Y3                                160
    END IF                                        161
    IF(NEXT.EQ.-1) GO TO 60                       162
C
C        SPECIAL CASE 2. SINGLE POINTS EQUAL TO CONTOUR VALUE 164
C        CHECK NODE 1 FIRST                       165
    JUMP=0                                        166
C
C        CHECK IF SEGMENT DEAD-ENDS AT NODE 1     168
    IF(CP1.EQ.VALINC .AND. ISTART.EQ.2) THEN    169
        NEXT=-1                                  170
        XPC(IPT)=X1                              171
        YPC(IPT)=Y1                              172
C
C
C        OTHERWISE, START NEW SEGMENT AT NODE 1   174
    ELSE IF(CP1.EQ.VALINC .AND. ISTART.EQ.0) THEN 175
        IF((CP2.GT.VALINC .AND. CP3.GT.VALINC) .OR. (CP2.LT.VALINC
1 .AND. CP3.LT.VALINC)) GO TO 40                177
        JUMP=1                                    178
        XPC(1)=X1                                179
        YPC(1)=Y1                                180
        NEXT=IUP                                  181
        ISTART=1                                  182
        IF(IEVEN.EQ.1) THEN                       183
            NEXT=ISIDE                            184
            ISTART=3                              185
        END IF                                    186
        RATIO=(VALINC-CP2)/DELX32                 187
        XPC(2)=X2+RATIO*DELX32                   188
        YPC(2)=Y2+RATIO*DELY32                   189
C
C
C        NEXT CHECK NODE 2                       191
    ELSE IF(CP2.EQ.VALINC .AND. ISTART.EQ.3) THEN 192
        NEXT=-1                                  193
        XPC(IPT)=X2                              194
        YPC(IPT)=Y2                              195
    ELSE IF(CP2.EQ.VALINC .AND. ISTART.EQ.0) THEN 196
        IF((CP1.GT.VALINC .AND. CP3.GT.VALINC) .OR. (CP1.LT.VALINC
1 .AND. CP3.LT.VALINC)) GO TO 40                198
        JUMP=1                                    199
        XPC(1)=X2                                200
        YPC(1)=Y2                                201
        NEXT=ISIDE                                202
        ISTART=2                                  203
        IF(IEVEN.EQ.1) THEN                       204
            NEXT=IUP                              205
            ISTART=2                              206
        END IF                                    207
        RATIO=(VALINC-CP1)/DELX31                 208
        XPC(2)=X1+RATIO*DELX31                   209
        YPC(2)=Y1+RATIO*DELY31                   210
C
C
C        NEXT CHECK NODE 3                       212

```

```

ELSE IF(CP3.EQ.VALINC .AND. ISTART.EQ.1) THEN                213
  NEXT=-1                                                    214
  XPC(IPT)=X3                                                215
  YPC(IPT)=Y3                                                216
ELSE IF(CP3.EQ.VALINC .AND. ISTART.EQ.0) THEN                217
  IF((CP1.GT.VALINC .AND. CP2.GT.VALINC) .OR. (CP1.LT.VALINC
1 .AND. CP2.LT.VALINC)) GO TO 40                              218
  JUMP=1                                                      219
  XPC(1)=X3                                                  220
  YPC(1)=Y3                                                  221
  NEXT=IDN                                                    222
  ISTART=3                                                    223
  IF(IEVEN.EQ.1) ISTART=2                                    224
  RATIO=(VALINC-CP1)/DELV21                                  225
  XPC(2)=X1+RATIO*DELX21                                     226
  YPC(2)=Y1+RATIO*DELY21                                     227
END IF                                                        228
IF(JUMP.EQ.1 .OR. NEXT.EQ.-1) GO TO 60                       229
C                                                            230
C    ROUTINE FOR DRAWING CONTOUR SEGMENT THROUGH TYPICAL ELEMENTS 231
C    START SEGMENT, IF NECESSARY                               232
C    IF(ISTART.EQ.0) THEN                                      233
C    CHECK FOR CONTOUR ENTERING ON BOTTOM OF TRIANGLE (SIDE 1-2) 234
C    IF((CP1.GT.VALINC .AND. CP2.LT.VALINC) .OR. (CP1.LT.VALINC
1 .AND. CP2.GT.VALINC)) THEN                                  235
  ISTART=1                                                    236
  RATIO=(VALINC-CP1)/DELV21                                  237
  XPC(1)=X1+RATIO*DELX21                                     238
  YPC(1)=Y1+RATIO*DELY21                                     239
C                                                            240
C    CONTOUR MUST START ON SIDE 2 OR 3. PICK STARTING POINT  241
C    BASED ON WHETHER ELEMENT IS ODD OR EVEN                  242
C    ELSE                                                      243
C    FOR ODD ELEMENT, START ON SIDE 1-3                        244
C    IF(MOD(NEXT,2).NE.0) THEN                                  245
  ISTART=3                                                    246
  RATIO=(VALINC-CP1)/DELV31                                  247
  XPC(1)=X1+RATIO*DELX31                                     248
  YPC(1)=Y1+RATIO*DELY31                                     249
C    IF EVEN, START CONTOUR ON SIDE 2-3                        250
C    ELSE                                                      251
  ISTART=2                                                    252
  RATIO=(VALINC-CP2)/DELV32                                  253
  XPC(1)=X2+RATIO*DELX32                                     254
  YPC(1)=Y2+RATIO*DELY32                                     255
  END IF                                                      256
  END IF                                                      257
  END IF                                                      258
  END IF                                                      259
END IF                                                        260
C                                                            261
C    CHECK FOR CONTOUR ENTERING ON BOTTOM OF TRIANGLE (SIDE 1-2) 262
C    IF(ISTART.EQ.1) THEN                                       263
C                                                            264
C    CHECK WHETHER CONTOUR EXITS SIDE OR TOP                  265

```

```

      IF((CP3.GT.VALINC .AND. CP1.LT.VALINC) .OR. (CP3.LT.VALINC
1  .AND. CP1.GT.VALINC)) THEN
      266
C
      267
C
      268
      CONTOUR MUST EXIT BETWEEN NODES 1 AND 3
      269
      NEXT=ISIDE
      270
      ISTART=2
      271
      IF(IEVEN.EQ.1) THEN
      272
      NEXT=IUP
      273
      ISTART=1
      274
      END IF
      275
      RATIO=(VALINC-CP1)/DELV31
      276
      XPC(IPT)=X1+RATIO*DELX31
      277
      YPC(IPT)=Y1+RATIO*DELY31
      278
C
      279
C
      CONTOUR MUST EXIT BETWEEN NODES 2 AND 3
      280
      ELSE
      281
      NEXT=IUP
      282
      ISTART=1
      283
      IF(IEVEN.EQ.1) THEN
      284
      NEXT=ISIDE
      285
      ISTART=3
      286
      END IF
      287
      RATIO=(VALINC-CP2)/DELV32
      288
      XPC(IPT)=X2+RATIO*DELX32
      289
      YPC(IPT)=Y2+RATIO*DELY32
      290
      END IF
      291
C
      292
C
      CHECK FOR CONTOUR ENTERING ON SIDE 2-3
      293
      ELSE IF(ISTART.EQ.2) THEN
      294
C
      295
C
      CHECK WHETHER CONTOUR EXITS BOTTOM OR SIDE 1-3
      296
      IF((CP3.GT.VALINC .AND. CP1.LT.VALINC) .OR. (CP3.LT.VALINC
1  .AND. CP1.GT.VALINC)) THEN
      297
C
      298
C
      CONTOUR MUST EXIT BETWEEN NODES 1 AND 3
      299
      300
      NEXT=ISIDE
      301
      ISTART=2
      302
      IF(IEVEN.EQ.1) THEN
      303
      NEXT=IUP
      304
      ISTART=1
      305
      END IF
      306
      RATIO=(VALINC-CP1)/DELV31
      307
      XPC(IPT)=X1+RATIO*DELX31
      308
      YPC(IPT)=Y1+RATIO*DELY31
      309
C
      310
C
      CONTOUR MUST EXIT BETWEEN NODES 1 AND 2
      311
      ELSE
      312
      NEXT=IDN
      313
      ISTART=3
      314
      IF(IEVEN.EQ.1) ISTART=2
      315
      RATIO=(VALINC-CP1)/DELV21
      316
      XPC(IPT)=X1+RATIO*DELX21
      317
      YPC(IPT)=Y1+RATIO*DELY21
      318

```


	END IF	319
C		320
C	CHECK FOR START OF CONTOUR SEGMENT ALONG SIDE 3-1	321
	ELSE IF(ISTART.EQ.3) THEN	322
C		323
C	CHECK WHETHER CONTOUR EXITS BOTTOM OR SIDE 2-3	324
	IF((CP2.GT.VALINC .AND. CP1.LT.VALINC) .OR. (CP2.LT.VALINC	325
1	.AND. CP1.GT.VALINC)) THEN	326
C		327
C	CONTOUR MUST EXIT BETWEEN NODES 1 AND 2	328
	NEXT=IDN	329
	ISTART=3	330
	IF(IEVEN.EQ.1) ISTART=2	331
	RATIO=(VALINC-CP1)/DELV21	332
	XPC(IPT)=X1+RATIO*DELX21	333
	YPC(IPT)=Y1+RATIO*DELY21	334
C		335
C	CONTOUR MUST EXIT BETWEEN NODES 2 AND 3	336
	ELSE	337
	NEXT=IUP	338
	ISTART=1	339
	IF(IEVEN.EQ.1) THEN	340
	NEXT=ISIDE	341
	ISTART=3	342
	END IF	343
	RATIO=(VALINC-CP2)/DELV32	344
	XPC(IPT)=X2+RATIO*DELX32	345
	YPC(IPT)=Y2+RATIO*DELY32	346
	END IF	347
	END IF	348
C	CHECK IF CONTOUR LINE SEGMENT HAS ENDED	349
60	IF(NEXT.EQ.-1) GO TO 70	350
C	CHECK IF CONTOUR LINE SEGMENT HAS LEFT BOUNDARY	351
	IF(NEXT.LT.1 .OR. NEXT.GT.NTR) GO TO 80	352
	IF(MOD(NEXT,NY2).EQ.0 .AND. ISTART.EQ.3) GO TO 80	353
	IF(MOD((NEXT-1),NY2).EQ.0 .AND. ISTART.EQ.1) GO TO 80	354
C	CHECK FOR END OF CLOSED CONTOUR LOOP	355
	IF(IFLAG(NEXT).EQ.0) GO TO 70	356
C		357
C	OTERWISE, CONTINUE SEGMENT, OR BREAK AFTER 'NUM' SEGMENTS	358
	IF(IPT.NE.NUM) GO TO 50	359
	ICLK=1	360
C		361
C	BLANK OUT SPACE AT END OF SEGMENT TO WRITE LABEL	362
80	IF(IPT.LT.NUM) GOTO 70	363
	XPT=XPC(IPT)	364
	YPT=YPC(IPT)	365
	IP1=IPT	366
C	CHECK IF ENOUGH SPACE IS CREATED BY BLANKING OUT ONE POINT	367
90	IP1=IP1-1	368
	IF(IP1.LE.1) GO TO 100	369
	XP1=XPC(IP1)	370
	YP1=YPC(IP1)	371

```

      DELX=(XPT-XP1)/XSCLP                                372
      DELY=(YPT-YP1)/YSCLP                                373
      XLEN=SQRT(DELX*DELX+DELY*DELY)                      374
C      IF NOT, DROP ANOTHER POINT ON CURVE                375
      IF(XLEN.LT.SPC1) GO TO 90                            376
C      MAKE SURE LABELS ARE RIGHT-SIDE UP                 377
100     OFSET=(XLEN-SPC1)/2.0                              378
      ANG=90.                                              379
      IF(DELY.LT.0.0) ANG=270.                             380
      IF(DELX.NE.0.0) ANG=ATAN2(DELY,DELX)*RAD            381
      IF (ABS(ANG).LE.90) THEN                             382
        CALL ANGLE(ANG)                                    383
        XP1=XP1+(OFSET*COS(ANG/RAD)+HITE*SIN(ANG/RAD)/2.0)*XSCLP 384
        YP1=YP1+(OFSET*SIN(ANG/RAD)-HITE*COS(ANG/RAD)/2.0)*YSCLP 385
        CALL RLREAL(VALINC,IPL,XP1,YP1)                   386
      ELSE                                                 387
        ANG=ANG-180.                                      388
        CALL ANGLE(ANG)                                    389
        XPT=XPT+(OFSET*COS(ANG/RAD)+HITE*SIN(ANG/RAD)/2.0)*XSCLP 390
        YPT=YPT+(OFSET*SIN(ANG/RAD)-HITE*COS(ANG/RAD)/2.0)*YSCLP 391
        CALL RLREAL(VALINC,IPL,XPT,YPT)                   392
      END IF                                              393
      CALL RESET ('ANGLE')                                 394
      IPT=IPL                                             395
C                                                       396
C      DRAW CONTOUR SEGMENT                               397
70     CALL CURVE(XPC,YPC,IPT,0)                           398
C      EITHER CONTINUE CONTOUR SEGMENT WHERE IT LEFT OFF 399
      IF (ICLK.EQ.1) THEN                                  400
        ICHK=0                                            401
        ISTART=0                                          402
        IPT=1                                             403
        GO TO 50                                          404
      END IF                                              405
C      OR START SEARCH FOR NEXT SEGMENT                   406
40     CONTINUE                                           407
20     CONTINUE                                           408
      CALL RESET('HEIGHT')                                 409
      CALL RESET('NUMODE')                                 410
      RETURN                                              411
      END                                                 412

```

```

*****
*
*                               DATA FILE GLQ.PTS
*
*
*****

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88

```

0.33998104358485D+000.86113631159405D+00
0.65214515486254D+000.34785484513745D+00
0.76526521133497D-010.22778585114164D+000.37370608871542D+000.51086700195082D+00
0.63605368072651D+000.74633190646015D+000.83911697182221D+000.91223442825132D+00
0.96397192727791D+000.99312859918509D+00
0.15275338713072D+000.14917298647260D+000.14209610931838D+000.13168863844917D+00
0.11819453196152D+000.10193011981724D+000.83276741576705D-010.62672048334108D-01
0.40601429800387D-010.17614007139152D-01
0.25959772301248D-010.77809333949536D-010.12944913539694D+000.18073996487342D+00
0.23154355137603D+000.28172293742326D+000.33114284826845D+000.37967005657680D+00
0.42717374158308D+000.47352584176171D+000.51860140005856D+000.56227890075394D+00
0.60444059704851D+000.64497282848947D+000.68376632738135D+000.72071651333573D+00
0.75572377530658D+000.78869373993226D+000.81953752616214D+000.84817198478593D+00
0.87451992264690D+000.89851031081004D+000.92007847617762D+000.93916627611642D+00
0.95572225583999D+000.96970178876505D+000.98106720175259D+000.98978789522222D+00
0.99584052511884D+000.99921012322744D+00
0.51907877631221D-010.51767943174910D-010.51488451500981D-010.51070156069855D-01
0.50514184532509D-010.49822035690550D-010.48995575455757D-010.48037031819971D-01
0.46948988848912D-010.45734379716114D-010.44396478795787D-010.42938892835935D-01
0.41365551235585D-010.39680695452381D-010.37888867569243D-010.35994898051084D-01
0.34003892724946D-010.31921219019296D-010.29752491500789D-010.27503556749925D-01
0.25180477621521D-010.22789516943998D-010.20337120729457D-010.17829901014208D-01
0.15274618596785D-010.12678166476816D-010.10047557182288D-010.73899311633454D-02
0.47127299269535D-020.20268119688737D-02
0.15030805704206D-010.45078833455378D-010.75086122510670D-010.10502555464787D+00
0.13487007296848D+000.16459270634967D+000.19416659381859D+000.22356500882721D+00
0.25276138340572D+000.28172933217251D+000.31044267617922D+000.33887546656923D+00
0.36700200802816D+000.39479688200531D+000.42223496968490D+000.44929147468653D+00
0.47594194547414D+000.50216229745345D+000.52792883473767D+000.55321827156203D+00
0.57800775332758D+000.60227487725540D+000.62599771263251D+000.64915482063112D+00
0.67172527368334D+000.69368867439393D+000.71502517397392D+000.73571549017836D+00
0.75574092473217D+000.77508338022830D+000.79372537648276D+000.81165006633204D+00
0.82884125085838D+000.84528339402936D+000.86096163673850D+000.87586181023427D+00
0.88997044892541D+000.90327480255122D+000.91576284770591D+000.927423298707080+00
0.93824561779910D+000.94822002468348D+000.95733750537007D+000.96558982034615D+00
0.97296951206736D+000.97946991179059D+000.98508514581446D+000.98981014133672D+00
0.99364063268970D+000.99657317140603D+000.99860516265198D+000.99973522187609D+00
0.30059347260915D-010.30032181992593D-010.29977876005781D-010.29896478377947D-01
0.29788062669856D-010.29652726859082D-010.29490593251467D-010.29301808370591D-01
0.29086542825356D-010.28844991155801D-010.28577371657294D-010.28283926183256D-01
0.27964919926590D-010.27620641180020D-010.27251401075562D-010.2685753303340D-01
0.26439393810028D-010.25997360477175D-010.25531832779705D-010.25043231424904D-01
0.24531997972222D-010.23998594434228D-010.23443502859080D-010.22867224894895D-01
0.22270281336402D-010.21653211654298D-010.21016573507721D-010.20360942240297D-01
0.19686910360204D-010.18995087004733D-010.18286097389822D-010.17560592245083D-01
0.16819197234815D-010.16062612365553D-010.15291511380693D-010.14506591142768D-01
0.13708561003966D-010.12898142165527D-010.12076067026716D-010.11243078524165D-01
0.10399929462595D-010.95473818383288D-020.86862061579237D-020.78171807584252D-02
0.69410911251809D-020.60587292747506D-020.51708932073277D-020.42783867524722D-02
0.33820208186870D-020.24826218021768D-020.15810952911948D-020.67947618248455D-03
0.61239123751895D-020.18370818478814D-010.30614968779979D-010.42854526536379D-01
0.55087655694634D-010.67312521165716D-010.79527289100232D-010.91730127163519D-01
0.10391920481051D+000.11609269356033D+000.12824876727061D+000.14038560241137D+00
0.15250137833866D+000.16459427756755D+000.17666248604490D+000.18870419342139D+00
0.20071759332312D+000.21270088362263D+000.22465226670913D+000.23656994975828D+00
0.24845214500106D+000.26029706999194D+000.27210294787633D+000.2838680065708D+00
0.29559048446013D+000.30726861979932D+000.31890066184010D+000.3304848656242D+00
0.34201949352237D+000.35350281511297D+000.36493310782365D+000.37630865699871D+00
0.38762775619452D+000.39888870743546D+000.41008982146872D+000.42122941801762D+00
0.43230582603374D+000.44331738394753D+000.45426243991759D+000.46513935207848D+00
0.47594648878698D+000.48668222886689D+000.49734496185218D+000.507933088228610+00
0.51844501967367D+000.52887917929482D+000.53923400186606D+000.54950793406272D+00
0.55969943469448D+000.56980697493657D+000.57982903855908D+000.58976412215445D+00
0.59961073536297D+000.60936740109633D+000.61903265575926D+000.62860504946901D+00
0.63808314627290D+000.64746552436372D+000.65675077629297D+000.66593750918204D+00
0.67502434493116D+000.68400992042607D+000.69289288774257D+000.70167191434868D+00
0.71034568330454D+000.71891289345997D+000.72737225964965D+000.73572251288591D+00
0.74396240054911D+000.75209068657549D+000.76010615164265D+000.76800759335244D+00
0.77579382641132D+000.78346368280818D+000.79101601198954D+000.79844968103217D+00
0.80576357481299D+000.81295659617643D+000.82002766609892D+000.82697572385081D+00
0.83379972715550D+000.84049865234576D+000.84707149451729D+000.85351726767950D+00
0.85983500490337D+000.86602375846655D+000.87208259999549D+000.87801062060471D+00
0.88380693103316D+000.88947066177761D+000.89500096322308D+000.90039700577030D+00
0.90565797996014D+000.91078309659506D+000.91577158685749D+000.92062270242514D+00
0.92533571558332D+000.92990991933400D+000.93434462750200D+000.9386391748371D+00
0.942792971711746D+000.94680523123913D+000.95067551531663D+000.95440318876971D+00
0.95798769241117D+000.96142848853073D+000.96472506097570D+000.96787691522849D+00
0.97088357848074D+000.97374459970437D+000.97645954971923D+000.97902802125762D+00
0.98144962902546D+000.98372400976031D+000.98585082228612D+000.98782974756486D+00
0.98966048874506D+000.99134277120758D+000.99287634260882D+000.99426097292240D+00
0.99549645448109D+000.99658260202338D+000.99751925275672D+000.99830626647300D+00

```

0.99894352584341D+000.99943093746626D+000.99976843740926D+000.99995605001899D+00	89
0.12247671640290D-010.12245834369748D-010.12242160104273D-010.12236649395040D-01	90
0.12229303068710D-010.12220122227304D-010.12209108248037D-010.12196262783115D-01	91
0.12181587759482D-010.12165085378535D-010.12146758115794D-010.12126608720527D-01	92
0.12104640215340D-010.12080855895724D-010.12055259329560D-010.12027854356582D-01	93
0.11998645087806D-010.11967635904906D-010.11934831459564D-010.11900236672766D-01	94
0.11863856734071D-010.11825697100824D-010.11785763497343D-010.11744061914061D-01	95
0.11700598606621D-010.11655380094945D-010.11608413162253D-010.11559704854044D-01	96
0.11509262477039D-010.11457093598091D-010.11403206043039D-010.11347607895545D-01	97
0.11290307495875D-010.11231313439650D-010.11170634576553D-010.11108280009010D-01	98
0.11044259090814D-010.10978581425729D-010.10911256866049D-010.10842295511115D-01	99
0.10771707705805D-010.10699504038980D-010.10625695341897D-010.10550292686581D-01	100
0.10473307384170D-010.10394750983212D-010.10314635267934D-010.10232972256478D-01	101
0.10149774199095D-010.10065053576306D-010.99788230970349D-020.98910956966958D-02	102
0.98018845352573D-020.97112029952662D-020.96190646798406D-020.95254834106292D-02	103
0.94304732257376D-020.93340483776232D-020.92362233309562D-020.91370127604508D-02	104
0.90364315486628D-020.89344947837582D-020.88312177572487D-020.87266159616988D-02	105
0.86207050884010D-020.85135010250225D-020.84050198532215D-020.82952778462352D-02	106
0.81842914664382D-020.80720773628734D-020.79586523687543D-020.78440334989397D-02	107
0.77282379473815D-020.76112830845456D-020.74931864548058D-020.73739657738123D-02	108
0.72536389258339D-020.71322239610754D-020.70097390929698D-020.68862026954463D-02	109
0.67616333001738D-020.66360495937811D-020.65094704150536D-020.63819147521079D-02	110
0.62534017395424D-020.61239506555679D-020.59935809191153D-020.58623120869226D-02	111
0.57301638506014D-020.55971560336829D-020.54633085886443D-020.53286415939159D-02	112
0.51931752508693D-020.50569298807868D-020.49199259218138D-020.47821839258926D-02	113
0.46437245556800D-020.45045685814479D-020.43647368779680D-020.42242504213815D-02	114
0.40831302860526D-020.39413976414088D-020.37990737487662D-020.36561799581425D-02	115
0.35127377050563D-020.33687685073155D-020.32242939617942D-020.30793357411993D-02	116
0.29339155908297D-020.27880553253277D-020.26417768254275D-020.24951020347307D-02	117
0.23480529563273D-020.22006516498399D-020.20529202279661D-020.19048808534997D-02	118
0.1756557363307D-020.16079671307493D-020.14591373333107D-020.13100886819025D-02	119
0.11608435575677D-020.10114243932084D-020.86185370142008D-030.71215416347332D-03	120
0.56234895403141D-030.41246325442617D-030.26253494429644D-030.11278901782227D-03	121