```
!Tweak-Vis.F95  Version 1.0
!!2018.10.09.0900cdt JMS- Supports Tweak's .3dv Visualizations.

!  Jeffrey M. Setterholm, 8095 230th St. E., Lakeville, Minnesota 55044, USA
!   I have authored the four Fortran *.f95 source code files listed below.
!                       I hereby place these four files:
!       Tweak-Begin.f95, Tweak-Engine.F95, Tweak-User.F95, & Tweak-Vis.f95
!               and the algorithms which are demonstrated therein,
!                           in the public domain.
!Disclaimer:
!       ************************************************************
!       **********   Individual cognition is always flawed.  **********
!       **********                - So: -                    **********
!       **********      Use this code at your own risk.      **********
!       ************************************************************

!Table of Contents:
!  Subroutine Draw3dJTL(iColor,Thickness,X,Y,Z,iP)
!  Subroutine Morph3dJTL(TLin,iP)
!  Function Log10LF(Value1,iP)
!Module CharDef
!  ~an English character vector font, & more.
!End Module CharDef
!  Subroutine AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iCol,fLineWidth,iP)
!  Subroutine Model7DoF(Xyzh,Rpyh,S,Model7h,iP)


!-------------------------------------------------------------------------------7-9

Subroutine Draw3dJTL(TLin,iP)
!2018.09.12.1100cdt JMS- Records 3D points in JTL ="Jeff`s Thick Line" format,
!                                a list of dots connected by color & thickness.
!                      - Then exports a .3dv file
!                      - Traveler2/Athlon64/WinXPPro-32/APF9.0-32

!This subroutine records "connect-the dots" information, as TL.*'s.
!The first entry is the number of TL.* (points) which follow.

!Each TL.* is: a line color, a line thickenss, and an (X,Y,Z).
!Color=0 represents moving to the point# without drawing a line.
!        That's how you begin a new line, or move to the next colored point.

 !---                                                                globals
use Tweakrec,only: jpU3d &  !Unit# for "recording the dots"
!                           <  6     : return immediately
!                           =  6     : write to screen
!                           >  10-20 : write to Unit# = jpU3d
                  ,TLrec &  !"Thick Line" record format
!type:: TLrec              !"Thick Line" record
! integer*4:: iC             ! DOS color of the line [0,15] to the next dot
!                           =  0: move-to without drawing
!                           = -1: closes the file
! real*4   :: Th             ! Line thickness [1.0 - 20.0] & width     in pixels
! real*16  :: XYZ(3)         ! (X,Y,Z) "next dot"
!end type TLrec;
                  ,TLprev   !for reporting the most recent TL. write
 !---
 implicit none                                                     !arguments
  type(TLrec):: TLin
  integer*4  :: iP
  !---                                                            !internals
  type(TLrec)            :: TL !A local variable; don't use the TL in TweakRec!

  character:: JTLfileName*15 &       !JTL export filename:
                          = "Tweak-3dJTL.txt"c

  character:: Export3DVfileName*16 & !3dv export filename:
                          = "Tweak-3dDraw.3dv"c

  integer*4:: nPtot,nPtot2  !Total number of points in the file
```

```
  71    integer*4:: nP
  72    integer*4:: Init=0      !
  73
  74    character:: cNumber*9     !Used to insert nPtot into JTLfileName
  75    integer*4:: i
  76
  77    type(TLrec),allocatable:: TLr(:)
  78    integer*4:: iAlloc        !Allocation error flag
  79    type(TLrec)             :: TL0 !used to clear TLr()
  80    character:: FormatTL*19 & !Fortran write format of the TL. entries:
  81                            ="(i2,1x,f5.1,3e18.9)"c ! <- (X,Y,Z) 9-place accuracy
  82  !                   & Below: adjust the .3dv export formats to suit your taste.
  83     !-----                                                          !end defs
  84
  85                                                          if(jpU3d < 6) return
  86
  87  !  Internalize the incoming TL arguments: ************************************
  88     TL.iC  = TLin.iC;  if(TL.iC > 15  ) TL.iC = 13  !->Purple
  89                        if(TL.iC < -1  ) TL.iC = -1  !Closes the JTL file
  90     TL.Th  = TLin.Th;  if(TL.Th <  1. ) TL.Th =  1. !Line width & point diameter
  91                        if(TL.Th > 20.0) TL.Th = 20.0
  92     TL.XYZ = TLin.XYZ
  93
  94     if(Init == 0) then   !The recording file hasn't been opened. ***************
  95  !    Try to open it:
  96       select case(jpU3d) !Screening for a valid Unit#:
  97        case(:5,7:9,21:);                                  return !Wrong number(s)
  98        case(    6   ); Init = jpU3d; nPtot = 0                          ;return
  99        case(  10:20 ); Init = jpU3d; nPtot = 0
 100         open(unit=jpU3d,file=JTLfileName, action="write",err=10) !Open it
 101         write(jpU3d,"(i9)") 0
 102         if(iP > 5) write(iP,*) "Draw3dJTL- opened:",JTLfileName
 103       end select!(jpU3d)
 104     endif!Init=0
 105
 106  !  Record the next dot (the beef): ******************************************
 107     if(TL.iC >= 0) then;                            if(Init /= jpU3d) goto 11
 108       nPtot  = nPtot+1
 109       TLprev = TL                                      ;if(Init == 6) return
 110       write(jpU3d,FormatTL,err=12) TL                                 ;return
 111     endif!(iColor>=0)
 112
 113  !  TL.iC < 0 implies that the last record has been recorded.
 114                                     if((TL.iC < 0).and.(Init /= jpU3d)) goto 11
 115                                                     if(Init == 6    )  return
 116
 117  !  Dot collection is done. Finalize the JTL file and the .3dv file. ***********
 118     close(jpU3d,err=13) !  Close JTLfileName
 119
 120  !  Binary-insert nPtot at the beginning of JTLfileName
 121     write(cNumber,"(i9)") nPtot
 122     open(unit=jpU3d, file=JTLfileName   , action="readwrite", status='old'    &
 123                    , form="unformatted" ,recl=1,      err=14, Access='direct'  )
 124       do i=1,9;  write(jpU3d,rec=i,err=15) cNumber(i:i) ;enddo!i
 125     close(jpU3d)
 126
 127  !  Re-import all the TL records in JTLfileName:
 128     allocate(TLr(nPtot),stat=iAlloc)                  ;if(iAlloc /= 0) goto 16
 129     TLr=TL0
 130     open(unit=jpU3d,file=JTLfileName, action="read",err=17)
 131       read(jpU3d,*) nPtot2                            ;if(nPtot2 /= nPtot) goto 18
 132       do Np=1,nPtot; read(jpU3d,FormatTL,err=19) TLr(nP) ;enddo!nP
 133     close(jpU3d)
 134
 135  !  Note: At this point you can write the code to export the JTL data
 136  !          in the 3D CAD File format of your choice! The .3dv export follows.
 137
 138  !  Exporting in .3dv format:
 139     open(unit=jpU3d,file=Export3DVfileName, action="write",err=20)
 140                       write(jpU3d,"(i9)"       ) nPtot2
```

```
 141      do nP=1,nPtot2; write(jpU3d,"(3e18.9)"   ) TLr(nP).XYZ   ;enddo!nP
 142                      write(jpU3d,"(i9)"       ) nPtot2
 143      do nP=1,nPtot2; write(jpU3d,"(i9,1x,i2)" ) nP,TLr(nP).iC ;enddo!nP
 144    close(jpU3d)
 145
 146    deallocate(TLr)
 147    if(iP > 5) write(iP,*) "Draw3dJTL- closed:",Export3DVfileName
 148    Init=0                                                          ;return
 149
 150 !  ------------------------------------
 151 !  Screen error messages -> pause -> stop.
 152 10 write(6,*) "Error opening: "        , JTLfileName                ;goto 30
 153 11 write(6,*) "jpU3d's value changed: ", jpU3d, "/=",  Init         ;goto 30
 154 12 write(6,*) "Error writing TL. TL="  , TL                         ;goto 30
 155 13 write(6,*) "Error closing: "        , JTLfileName                ;goto 30
 156 14 write(6,*) "Error reopening: ", JTLfileName, "for direct access."  ;goto 30
 157 15 write(6,*) "Error overwriting nPtot.  cNumber=",cNumber          ;goto 30
 158 16 write(6,*) "Error allocating TLr. nPtot & iAlloc= ",nPtot,iAlloc  ;goto 30
 159 17 write(6,*) "Error reopening: "      , JTLfileName, "for read."   ;goto 30
 160 18 write(6,*) "nPtot2 /= nPtot:"       , nPtot2,nPtot, "oops."      ;goto 30
 161 19 write(6,*) "Formatted read error. nP=:", nP                      ;goto 30
 162 20 write(6,*) "Error opening: "        , JTLfileName                ;goto 30
 163
 164 30 pause"^^ Error message from Draw3dJTL. Press enter to halt."        ;stop
 165 end Subroutine Draw3dJTL
 166 !-------------------------------------------------------------------------------7-9
 167
 168 Subroutine Morph3dJTL(TLin,iP)
 169 !2018.10.08.0725cdt JMS- Focal point for morphing (X,Y,Z) 3D plot points.
 170 !                       - "Traveler2"/Athlon64/WinXPPro32/APF9.0
 171  !---                                                             globals
 172  use Tweakrec,only: TLrec &   !"Thick Line" record format
 173                    ,jTLmorph  !Morph3dJTL() mode control. (predefined)
 174 !                           = 0: Z-axis- no change
 175 !                                      Z:  (<,-1.),[-1.,1.], ((1.,>)
 176 !                           = 1:         -om(-Z)-1.| linear |  +on(+Z)+1.
 177 !                           = 2:       - -log10(-Z)| linear |+log10(+Z)
 178 !                           = 3:       -  -2.-1./Z | linear | +2.-1./Z
 179 !                           = 4:       - User defined morph X, &/or Y, &/or Z
 180  implicit none                                                    !arguments
 181   type(TLrec):: TLin
 182   integer*4:: iP              !not presently used.
 183   !---                                                           !internals
 184   real*16  :: Log10LF
 185    !-----                                                        !end defs
 186    Select case(jTLmorph)
 187     case(0) !No change:
 188
 189 !        Z:    (<,-1.)  ,[-1.,1.],  ((1.,>)
 190     case(1) ! -Log10LF-1.| linear | Log10LF+1. - Linear within each decade.
 191 !             Empowers linear visual intuition of surfaces whose values
 192 !               vary over orders of mag.
 193      if(TLin.XYZ(3) >  1._16) TLin.XYZ(3)= Log10LF( TLin.XYZ(3),1._16,0) +1._16
 194      if(TLin.XYZ(3) < -1._16) TLin.XYZ(3)=-Log10LF(-TLin.XYZ(3),1._16,0) -1._16
 195
 196 !          Z:  (<,-1.),[-1.,1.],  ((1.,>)
 197     case(2) !-log10(-Z)| linear |+log10(+Z) - Log10 scaling for abs(Z)>1.
 198      if(     TLin.XYZ(3) < -1._16) then
 199              TLin.XYZ(3) = -log10(-TLin.XYZ(3)) -1._16
 200       elseif(TLin.XYZ(3) > +1._16) then
 201              TLin.XYZ(3) = +log10(-TLin.XYZ(3)) +1._16
 202      endif !TLin.XYZ(3)<-1.
 203
 204 !        Z:  (<,-1.),[-1.,1.], ((1.,>)
 205     case(3) ! 2-1/Z | linear | -2+1/Z -Reciporcal scaling for abs(z)>1.
 206
 207      if(TLin.XYZ(3) > 1._16) TLin.XYZ(3) = 2._16 - 1._16/TLin.XYZ(3)
 208      if(TLin.XYZ(3) <-1._16) TLin.XYZ(3) =-2._16 - 1._16/TLin.XYZ(3)
 209
 210     case(4) !User defined morph of X,Y, & Z
```

```
211       pause " jTLmorph=4 has not yet been defined by the user. 'Enter' to halt."
212       stop
213     end select !(jTLmorph)
214                                                                          return
215 End Subroutine Morph3dJTL
216 !-------------------------------------------------------------------------------7-9
217
218 Function Log10LF(Value1,iP)
219 !2018.09.27.1310cdt JMS- Traveler2/Athlon64/WinXPPro-32/APF9.0-32
220
221 !Log10(Value1), with a Linear Fractional value within each decade,
222 !  which aids my linear visual intuition.
223
224 !These decades span [10**n,10**(n+1)] which is an interval width of 9*10**n,
225 !  so the fractional values are not numerically intuitive.
226
227 !Function om() yields intuitive numeric fractions, but the values are
228 !  decade-discontinuous.
229
230 != M.nnn - is an L10LF change scale for positive numbers.
231 !         For Value1 < 1., the reciprocal is used & the result is negative.
232 !         For Value1 <=0., L10LF = 0.
233
234 !  M     - is log10 of the floor decade
235 !   .nnn_- is the linear fractional change between 10**M and 10**(M+1)
236 !The intent is to make wildly-varying scale changes visually intuitive.
237 !
238  !---                                                              No globals
239  implicit none                                                    !arguments
240   real*16  :: Value1  !Value- ending/current                             >0.
241   real*16  :: Log10LF !     - Value1/Value0 measured in omnScale units.
242   integer*4:: iP
243   !---                                                            !internals
244   real*16  :: Vdel1,Vdel2,Vdel3,Vdel4
245    !-----                                                         !end defs
246    if(Value1 <= 0._16) then; Log10LF =   0._16          ;return;endif!Value1<=0
247                              Vdel1   =  Value1
248                              Vdel2   =  Vdel1
249    if(Vdel2   < 1._16)       Vdel2   =  1._16/Value1
250                              Vdel3   =  int(log10(Vdel2))
251                              Vdel4   =  10._16**Vdel3
252                              Log10LF =  Vdel2/(9._16*Vdel4) +Vdel3 -1._16/9._16
253    if(Vdel1  <  1._16)       Log10LF = -Log10LF
254    if(iP > 5) then
255      write(iP,"(f14.9,' = Log10LF(',e16.9,',',i2,')')") Log10LF,Value1,iP
256      write(iP,"(4e14.6,' :Vdel1,Vdel2,Vdel3,Vdel4')") Vdel1,Vdel2,Vdel3,Vdel4
257    endif!iP>5
258                                                                          return
259 End Function Log10LF
260 !-------------------------------------------------------------------------------7-9
261
262 Module CharDef
263 !2018.08.16.2120cdt JMS- Transferred the C implementation to Fortran.
264 !2018.07.17.        JMS- Grid coordinates defining characters are now square.
265 !                        vertical & horizontal lines can have the same width.
266 !2018.04.11.1400cdt JMS- Enlarged the Free symbol char(224)
267 !                      - The identical character data is found in:
268 !                          http://ftp.setterholm.com/3DEnvC in file 3DVecText.h
269 !                      - Traveler2/Athlon64/WinXPPro-32/APF9.0-32
270 !use CharDef                                                        !2018.03.21
271  implicit none
272  !---
273
274  !ASCII character starting locations in CharLoc (if >0) :
275  integer*2:: CharLoc(0:255) = (/                                                &
276 !  0    1    2    3    4    5    6    7    8    9    a    b    c    d    e    f
277 !                                  BEL   BS   HT   LF   VT   FF   CR          0
278    0,   0,   0,   0,   0,   0,   0,   1,  67, 136, 169, 196, 220, 253,   0,   0,&
279 !                                                    ESC                       1
280    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 307,   0,   0,   0,   0,&
```

```
!     !     "     #     $     %     &     '     (     )     *     +     ,     -     .     / 2
 385, 388, 412, 427, 454, 505, 568, 634, 643, 670, 697, 718, 733, 754, 763, 781,&
! 0   1     2     3     4     5     6     7     8     9     :     ;     <     =     >     ? 3
 790, 838, 856, 889, 931, 946, 979,1018,1030,1081,1120,1153,1180,1192,1207,1219,&
! @   A     B     C     D     E     F     G     H     I     J     K     L     M     N     O 4
1276,1336,1354,1393,1426,1450,1471,1489,1522,1543,1564,1585,1609,1621,1639,1654,&
! P   Q     R     S     T     U     V     W     X     Y     Z     [     \     ]     ^     _ 5
1684,1708,1744,1777,1822,1837,1858,1870,1888,1903,1921,1936,1951,1960,1975,1987,&
! `   a     b     c     d     e     f     g     h     i     j     k     l     m     n     o 6
1996,2005,2038,2071,2098,2131,2164,2188,2224,2248,2272,2305,2326,2338,2377,2401,&
! p   q     r     s     t     u     v     w     x     y     z     {     |     }     ~     7
2431,2464,2506,2527,2566,2581,2605,2617,2635,2650,2683,2698,2734,2743,2779,2812,&
!                                                                             8
   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,&
!                          •  box  grid          ™                            9
   0,    0,    0,    0,    0,2860,    0,    0,    0,2884,    0,    0,    0,    0,    0,&
!                                                 ©                 ®         a
   0,    0,    0,    0,    0,    0,    0,    0,    0,2938,    0,    0,    0,    0,2992,&
! °   ±                                                                       b
3049,3091,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,&
!Special use in CEnv3D- Colors (black screen  background):
!mov Wht lRed  Red  Mag  Yel lGrn  Grn  Cyn lBlu  Blu LGry  Gry  Brn  Prp Blk c
3112,3184,3238,3316,3385,3448,3493,3574,3646,3700,3775,3841,3925,4000,4078,4150,&
!Special use in CEnv3D- Shape modifiers:
!toff w1.  w2.  w3.  w4.  w5. ita  und  sub  sup                show          d
4219,4300,4327,4369,4414,4447,4486,4534,4588,4675,   0,   0,   0,4747,   0,   0,&
!Free ©    ®    ™   ©x                                                box  gride
4831,4906,4960,5017,5071,   0,   0,   0,   0,   0,   0,   0,   0,   0,5122,5140,&
!                                                                             f
   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0/)
! 0    1    2    3    4    5    6    7    8    9    a    b    c    d    e    f
!CharLoc(256) defined above.

 integer*2:: PointLoc(5199)           !8 triples-per-line.
        data PointLoc(   1: 600) / &  !~ASCII characters   7- 38 decimal
0,  7, 21,1,-20, 30,2,-10, 30,2, -8, 27,2, -8, 21,2,-12, 18,2,-20, 18,1,-12, 18,&
2, -8, 15,2, -8,  9,2,-10,  6,2,-20,  6,2,-20, 30,1,  6, 12,2, -6, 12,2, -6,-12,&
2,  6,-12,1, -6,  0,2,  2,  0,1,  8, -6,2,  8,-30,2, 20,-30,0,  8, 22,1,-16, 24,&
2, -6, 24,2, -2, 21,2, -2, 15,2, -8, 12,2,-16, 12,1, -8, 12,2, -2,  9,2, -2,  3,&
2, -6,  0,2,-16,  0,2,-16, 24,1, 16, -3,2, 12,  0,2,  6,  0,2,  2, -3,2,  2, -9,&
2, 16,-15,2, 16,-21,2, 12,-24,2,  6,-24,2,  2,-21,0,  9, 10,1,-16, 24,2,-16,  0,&
1,-16, 12,2, -2, 12,1, -2, 24,2, -2,  0,1,  2,  0,2, 18,  0,1, 10,  0,2, 10,-24,&
0, 10,  8,1,-16, 24,2,-16,  0,2, -2,  0,1,  2,-24,2,  2,  0,2, 16,  0,1,  2,-12,&
2, 14,-12,0, 11,  7,1,-16, 24,2, -8,  0,2, -2, 24,1,  2,  0,2, 18,  0,1, 10,  0,&
2, 10,-24,0, 12, 10,1,-16,  0,2,-16, 24,2, -2, 24,1,-16, 12,2, -4, 12,1,  2,-24,&
2,  2,  0,2, 16,  0,1,  2,-12,2, 14,-12,0, 13, 17,1, -2, 18,2, -6, 24,2,-12, 24,&
2,-16, 18,2,-16,  6,2,-12,  0,2, -6,  0,2, -2,  6,1,  2,-24,2,  2,  0,2, 12,  0,&
2, 16, -3,2, 16, -9,2, 12,-12,2,  2,-12,1,  8,-12,2, 16,-24,0, 27, 25,1, -8, 30,&
2,-20, 30,2,-20,  6,2, -8,  6,1,-20, 18,2,-12, 18,1,  6,  9,2,  4, 12,2, -4, 12,&
2, -6,  9,2, -6,  3,2,  6, -3,2,  6, -9,2,  4,-12,2, -4,-12,2, -6, -9,1, 20, -9,&
1, 20,-12,2, 16, -6,2, 12, -6,2,  8,-12,2,  8,-24,2, 12,-30,2, 16,-30,2, 20,-24,&
0, 32,  0,0, 33,  7,1,  0, 30,2,  0,-18,1,  0,-27,2,  2,-30,2,  0,-33,2, -2,-30,&
2,  0,-27,0, 34,  4,1, -8, 30,2, -8, 15,1,  8, 30,2,  8, 15,0, 35,  8,1, -8, 18,&
2,-14,-24,1, 12, 18,2,  6,-24,1,-16,  6,2, 16,  6,1,-18,-12,2, 14,-12,0, 36, 16,&
1, 14, 12,2,  6, 21,2, -6, 21,2,-14, 12,2,-14,  9,2, -6,  0,2,  6,  0,2, 14, -9,&
2, 14,-12,2,  6,-21,2, -6,-21,2,-14,-12,1, -4, 27,2, -4,-27,1,  4, 27,2,  4,-27,&
0, 37, 20,1,-10, 30,2, -6, 27,2, -6, 18,2,-10, 15,2,-16, 15,2,-20, 18,2,-20, 27,&
2,-16, 30,2,-10, 30,1, 20, 30,2,-20,-30,1, 10,-15,2, 16,-15,2, 20,-18,2, 20,-27,&
2, 16,-30,2, 10,-30,2,  6,-27,2,  6,-18,2, 10,-15,0, 38, 21,1, 20,-30,2,-12,  9,&
2,-14, 15,2,-14, 18,2,-10, 27,2, -4, 30,2,  2, 30,2,  8, 27,2, 12, 18,2, 12, 15 &
                                                                              /
        data PointLoc( 601:1200) / &  !~ASCII characters  39- 61
2,  8,  9,2,-12, -6,2,-18, -9,2,-20,-15,2,-20,-18,2,-18,-24,2,-14,-27,2, -6,-30,&
2,  6,-30,2, 10,-27,2, 20,-15,0, 39,  2,1,  0, 30,2,  0, 15,0, 40,  8,1,  6, 30,&
2,  0, 27,2, -4, 18,2, -6,  6,2, -6, -6,2, -4,-18,2,  0,-27,2,  6,-30,0, 41,  8,&
1, -6, 30,2,  0, 27,2,  4, 18,2,  6,  6,2,  6, -6,2,  4,-18,2,  0,-27,2, -6,-30,&
0, 42,  6,1, 10, 24,2,-10,-24,1, 20,  0,2,-20,  0,1, 10,-24,2,-10, 24,0, 43,  4,&
1,  0, 18,2,  0,-18,1, 20,  0,2,-20,  0,0, 44,  6,1,  2,-33,2, -2,-33,2, -2,-30,&
2,  2,-30,2,  2,-36,2, -2,-45,0, 45,  2,1,-20,  0,2, 20,  0,0, 46,  5,1,  0,-27,&
2,  2,-30,2,  0,-33,2, -2,-30,2,  0,-27,0, 47,  2,1,-20,-30,2, 20, 30,0, 48, 15,&
```

```
 1,  0,-30,2,-10,-30,2,-20,-15,2,-20,  0,2,-20, 15,2,-10, 30,2,  0, 30,2, 10, 30,&
 2, 20, 15,2, 20,  0,2, 20,-15,2, 10,-30,2,  0,-30,1,-20,-30,2, 20, 30,0, 49,  5,&
 1,-10,  6,2,  0, 30,2,  0,-30,1,-10,-30,2, 10,-30,0, 50, 10,1,-20, 18,2,-12, 24,&
 2,  0, 30,2, 12, 24,2, 20, 15,2, 12,  6,2,  0,  0,2,-12, -6,2,-20,-30,2, 20,-30,&
 0, 51, 13,1,-20, 18,2,-12, 30,2, 12, 30,2, 20, 18,2, 20,  6,2, 12,  0,2, -4,  0,&
 2, 12,  0,2, 20, -6,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,0, 52,  4,1, 20, -9,&
 2,-20, -9,2, 12, 30,2, 12,-30,0, 53, 10,1, 20, 30,2,-20, 30,2,-20,  0,2,-12,  6,&
 2, 12,  6,2, 20,  0,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,0, 54, 12,1, 20, 18,&
 2, 12, 30,2,-12, 30,2,-20, 18,2,-20,-18,2,-12,-30,2, 12,-30,2, 20,-18,2, 20, -6,&
 2, 12,  0,2,-12,  0,2,-20, -6,0, 55,  3,1,-20, 30,2, 20, 30,2, -6,-30,0, 56, 16,&
 1,-12,  0,2,-20,  6,2,-20, 18,2,-12, 30,2, 12, 30,2, 20, 18,2, 20,  6,2, 12,  0,&
 2,-12,  0,2,-20, -6,2,-20,-18,2,-12,-30,2, 12,-30,2, 20,-18,2, 20, -6,2, 12,  0,&
 0, 57, 12,1, 20,  6,2, 12,  0,2,-12,  0,2,-20,  6,2,-20, 18,2,-12, 30,2, 12, 30,&
 2, 20, 18,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,0, 58, 10,1,  0,  3,2,  2,  0,&
 2,  0, -3,2, -2,  0,2,  0,  3,1,  0,-24,2,  2,-27,2,  0,-30,2, -2,-27,2,  0,-24,&
 0, 59,  8,1,  0,-12,2,  2,-15,2,  0,-18,2, -2,-15,2,  0,-12,1,  0,-27,2,  0,-33,&
 2, -4,-39,0, 60,  3,1, 20, 15,2,-20,  0,2, 20,-15,0, 61,  4,1,-18,  6,2, 18,  6 &
                                                                                 /
         data PointLoc(1201:1800) / &  !~ASCII characters  62- 83
 1,-18, -6,2, 18, -6,0, 62,  3,1,-20, 15,2, 20,  0,2,-20,-15,0, 63, 18,1,-20, 18,&
 2,-18, 24,2,-14, 27,2, -6, 30,2,  6, 30,2, 14, 27,2, 18, 24,2, 20, 18,2, 20, 12,&
 2, 18,  6,2, 14,  3,2,  0, -3,2,  0,-15,1,  0,-27,2,  2,-30,2,  0,-33,2, -2,-30,&
 2,  0,-27,0, 64, 19,1,  8,  6,2,  4,  9,2, -2,  9,2, -8,  3,2, -8, -6,2, -2,-12,&
 2,  4,-12,2,  8, -9,2,  8,  9,1,  8, -9,2, 12,-12,2, 20, -9,2, 20,  9,2, 10, 18,&
 2,-10, 18,2,-20,  9,2,-20, -9,2,-10,-18,2, 10,-18,0, 65,  5,1,-20,-30,2,  0, 30,&
 2, 20,-30,1,-12, -9,2, 12, -9,0, 66, 12,1,-20, 30,2, 12, 30,2, 20, 18,2, 20,  6,&
 2, 12,  0,2,-20,  0,1, 12,  0,2, 20, -6,2, 20,-18,2, 12,-30,2,-20,-30,2,-20, 30,&
 0, 67, 10,1, 20, 18,2, 20, 18,2, 12, 30,2,-12, 30,2,-20, 18,2,-20,-18,2,-12,-30,&
 2, 12,-30,2, 20,-18,2, 20,-18,0, 68,  7,1,-20, 30,2, 12, 30,2, 20, 18,2, 20,-18,&
 2, 12,-30,2,-20,-30,2,-20, 30,0, 69,  6,1,-20,  0,2, 12,  0,1, 20, 30,2,-20, 30,&
 2,-20,-30,2, 20,-30,0, 70,  5,1,-20,  0,2, 12,  0,1, 20, 30,2,-20, 30,2,-20,-30,&
 0, 71, 10,1,  0,  0,2, 20,  0,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,2,-20, 18,&
 2,-12, 30,2, 12, 30,2, 20, 18,0, 72,  6,1,-20, 30,2,-20,-30,1,-20,  0,2, 20,  0,&
 1, 20, 30,2, 20,-30,0, 73,  6,1,-10, 30,2, 10, 30,1,  0, 30,2,  0,-30,1,-10,-30,&
 2, 10,-30,0, 74,  6,1, 20, 30,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,2,-20, -9,&
 0, 75,  7,1,-20, 30,2,-20,-30,1,-20,  0,2,  0,  0,2, 20, 30,1,  0,  0,2, 20,-30,&
 0, 76,  3,1,-20, 30,2,-20,-30,2, 20,-30,0, 77,  5,1,-20,-30,2,-20, 30,2,  0, -9,&
 2, 20, 30,2, 20,-30,0, 78,  4,1,-20,-30,2,-20, 30,2, 20,-30,2, 20, 30,0, 79,  9,&
 1, 12, 30,2, 20, 18,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,2,-20, 18,2,-12, 30,&
 2, 12, 30,0, 80,  7,1,-20,-30,2,-20, 30,2, 12, 30,2, 20, 18,2, 20,  9,2, 12,  0,&
 2,-20,  0,0, 81, 11,1, 12, 30,2, 20, 18,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,&
 2,-20, 18,2,-12, 30,2, 12, 30,1,  0,  0,2, 24,-36,0, 82, 10,1,-20,-30,2,-20, 30,&
 2, 12, 30,2, 20, 18,2, 20,  9,2, 12,  0,2,-20,  0,1, 10,  0,2, 20,-15,2, 20,-30,&
 0, 83, 14,1, 20, 18,2, 20, 18,2, 12, 30,2,-12, 30,2,-20, 18,2,-20,  6,2,-12,  0 &
                                                                                 /
         data PointLoc(1801:2400) / &  !~ASCII characters  84-110
 2, 12,  0,2, 20, -6,2, 20,-18,2, 12,-30,2,-12,-30,2,-20,-18,2,-20,-18,0, 84,  4,&
 1,-20, 30,2, 20, 30,1,  0, 30,2,  0,-30,0, 85,  6,1,-20, 30,2,-20,-18,2,-12,-30,&
 2, 12,-30,2, 20,-18,2, 20, 30,0, 86,  3,1,-20, 30,2,  0,-30,2, 20, 30,0, 87,  5,&
 1,-20, 30,2,-10,-30,2,  0, 15,2, 10,-30,2, 20, 30,0, 88,  4,1,-20, 30,2, 20,-30,&
 1,-20,-30,2, 20, 30,0, 89,  5,1,-20, 30,2,  0,  0,2,  0,-30,1,  0,  0,2, 20, 30,&
 0, 90,  4,1,-20, 30,2, 20, 30,2,-20,-30,2, 20,-30,0, 91,  4,1,  6, 30,2, -6, 30,&
 2, -6,-30,2,  6,-30,0, 92,  2,1,-20, 30,2, 20,-30,0, 93,  4,1, -6, 30,2,  6, 30,&
 2,  6,-30,2, -6,-30,0, 94,  3,1,-16,  0,2,  0, 18,2, 16,  0,0, 95,  2,1,-20,-33,&
 2, 20,-33,0, 96,  2,1, -6, 30,2,  0, 15,0, 97, 10,1, 18,  0,2, 18,-21,2,  8,-30,&
 2, -8,-30,2,-18,-21,2,-18, -9,2, -8,  0,2,  8,  0,2, 18, -9,2, 18,-30,0, 98, 10,&
 1,-18, 24,2,-18,-21,2, -8,-30,2,  8,-30,2, 18,-21,2, 18, -9,2,  8,  0,2, -6,  0,&
 2,-18, -9,2,-18,-30,0, 99,  8,1, 18, -9,2,  8,  0,2, -8,  0,2,-18, -9,2,-18,-21,&
 2, -8,-30,2,  8,-30,2, 18,-21,0,100, 10,1, 18, 24,2, 18,-21,2,  8,-30,2, -8,-30,&
 2,-18,-21,2,-18, -9,2, -8,  0,2,  8,  0,2, 18, -9,2, 18,-30,0,101, 10,1,-18,-15,&
 2, 18,-15,2, 18, -9,2,  8,  0,2, -8,  0,2,-18, -9,2,-18,-21,2, -8,-30,2,  8,-30,&
 2, 18,-27,0,102,  7,1,-10,-30,2,-10, 18,2,  0, 24,2,  8, 24,2, 18, 18,1,-18,  0,&
 2, 18,  0,0,103, 11,1, 18,-21,2,  8,-30,2, -8,-30,2,-18,-21,2,-18, -9,2, -8,  0,&
 2,  8,  0,2, 18, -9,2, 18,-36,2,  8,-45,2, -8,-45,0,104,  7,1,-18, 24,2,-18,-30,&
 2,-18, -9,2, -8,  0,2,  8,  0,2, 18, -9,2, 18,-30,0,105,  7,1,  0,  0,2,  0,-30,&
 1,  0, 21,2,  2, 18,2,  0, 15,2, -2, 18,2,  0, 21,0,106, 10,1, 18,  0,2, 18,-36,&
 2,  8,-45,2, -8,-45,2,-18,-36,1, 18, 21,2, 20, 18,2, 18, 15,2, 16, 18,2, 18, 21,&
 0,107,  6,1,-18, 24,2,-18,-30,1,-18, -6,2, 10, 12,1,-14, -3,2, 18,-30,0,108,  3,&
 1,  0, 24,2,  0,-30,2,  4,-30,0,109, 12,1,-18,-30,2,-18,  0,1,-18, -9,2,-12,  0,&
 2, -6,  0,2,  0, -9,2,  0,-24,1,  0, -9,2,  6,  0,2, 12,  0,2, 18, -9,2, 18,-30,&
```

```
 421 0,110,  7,1,-18,-30,2,-18,  0,1,-18, -9,2, -8,  0,2,  8,  0,2, 18, -9,2, 18,-30 &
 422                                                                                  /
 423         data PointLoc(2401:3000) / &  !~ASCII characters 111-174
 424 0,111,  9,1,  8,  0,2, 18, -9,2, 18,-21,2,  8,-30,2, -8,-30,2,-18,-21,2,-18, -9,&
 425 2, -8,  0,2,  8,  0,0,112, 10,1,-18,-45,2,-18,  0,1,-18, -9,2, -8,  0,2,  8,  0,&
 426 2, 18, -9,2, 18,-21,2,  8,-30,2, -8,-30,2,-18,-21,0,113, 13,1, 18,-21,2,  8,-30,&
 427 2, -8,-30,2,-18,-21,2,-18, -9,2, -8,  0,2,  8,  0,2, 18, -9,2, 18,-21,2, 12,-42,&
 428 2, 14,-45,2, 18,-45,2, 20,-42,0,114,  6,1,-18,-30,2,-18,  0,1,-18, -9,2, -8,  0,&
 429 2,  8,  0,2, 20, -9,0,115, 12,1, 18, -6,2,  8,  0,2, -8,  0,2,-18, -6,2,-18, -9,&
 430 2, -8,-15,2,  8,-15,2, 18,-21,2, 18,-24,2,  8,-30,2, -8,-30,2,-18,-24,0,116,  4,&
 431 1,  0, 24,2,  0,-30,1,-18, 12,2, 18, 12,0,117,  7,1,-18,  0,2,-18,-21,2, -8,-30,&
 432 2,  8,-30,2, 18,-21,2, 18,  0,2, 18,-30,0,118,  3,1,-18,  0,2,  0,-30,2, 18,  0,&
 433 0,119,  5,1,-18,  0,2, -8,-30,2,  0, -6,2,  8,-30,2, 18,  0,0,120,  4,1,-18,  0,&
 434 2, 18,-30,1,-18,-30,2, 18,  0,0,121, 10,1,-18,  0,2,-18,-21,2,-10,-30,2,  8,-30,&
 435 2, 18,-21,1, 18,  0,2, 18,-36,2,  8,-45,2,-10,-45,2,-18,-42,0,122,  4,1,-18,  0,&
 436 2, 18,  0,2,-18,-30,2, 18,-30,0,123, 11,1,  6, 30,2,  0, 27,2, -2, 21,2,  0,  9,&
 437 2, -2,  3,2, -6,  0,2, -2, -3,2,  0, -9,2, -2,-21,2,  0,-27,2,  6,-30,0,124,  2,&
 438 1,  0, 30,2,  0,-30,0,125, 11,1, -6, 30,2,  0, 27,2,  2, 21,2,  0,  9,2,  2,  3,&
 439 2,  6,  0,2,  2, -3,2,  0, -9,2,  2,-21,2,  0,-27,2, -6,-30,0,126, 10,1,-20, -3,&
 440 2,-18,  3,2,-12,  6,2, -8,  6,2, -2,  3,2,  2, -3,2,  8, -6,2, 12, -6,2, 18, -3,&
 441 2, 20,  3,0,127, 15,1,-14,-30,2,-14,-12,2,-16,-15,1, -6,-15,2, -2,-12,2,  2,-12,&
 442 2,  6,-15,2,  6,-21,2,  2,-24,2, -4,-24,2, -6,-30,2,  6,-30,1,  8,-12,2, 20,-12,&
 443 2, 14,-30,0,149,  7,1,  2,  3,2,  4,  0,2,  2, -3,2, -2, -3,2, -4,  0,2, -2,  3,&
 444 2,  2,  3,0,153, 17,1,-18, 24,2,-18, 27,2, -2, 27,2, -2, 24,1,-10, 27,2,-10,  3,&
 445 1,-12,  3,2, -8,  3,1,  0,  3,2,  4,  3,1,  2,  3,2,  2, 27,2, 10, 15,2, 18, 27,&
 446 2, 18,  3,1, 16,  3,2, 20,  3,0,169, 17,1, 10, 21,2, 20, 12,2, 20,-12,2, 10,-21,&
 447 2,-10,-21,2,-20,-12,2,-20, 12,2,-10, 21,2, 10, 21,1, 12,  6,2,  6, 12,2, -6, 12,&
 448 2,-12,  6,2,-12, -6,2, -6,-12,2,  8,-12,2, 12, -6,0,174, 18,1, -8, -3,2, -8, 21 &
 449                                                                                  /
 450         data PointLoc(3001:3600) / &  !~ASCII characters 176-199
 451 2,  4, 21,2,  8, 18,2,  8, 12,2,  4,  9,2, -8,  9,1,  2,  9,2,  8, -3,1,  8, 30,&
 452 2, 20, 21,2, 20, -3,2,  8,-12,2, -8,-12,2,-20, -3,2,-20, 21,2, -8, 30,2,  8, 30,&
 453 0,176, 13,1,  2, 30,2,  6, 27,2,  8, 24,2,  8, 21,2,  6, 18,2,  2, 15,2, -2, 15,&
 454 2, -6, 18,2, -8, 21,2, -8, 24,2, -6, 27,2, -2, 30,2,  2, 30,0,177,  6,1,-20,  0,&
 455 2, 20,  0,1,  0, 15,2,  0,-15,1,-20,-18,2, 20,-18,0,192, 23,1,-20, 15,2,-20, 27,&
 456 2,-16, 21,2,-12, 27,2,-12, 15,1, -4, 27,2, -2, 24,2, -2, 18,2, -4, 15,2, -6, 15,&
 457 2, -8, 18,2, -8, 24,2, -6, 27,2, -4, 27,1,  2, 27,2,  6, 15,2, 10, 27,1, 20, 27,&
 458 2, 14, 27,2, 14, 15,2, 20, 15,1, 14, 21,2, 20, 21,0,193, 17,1,-20, 24,2,-16,  6,&
 459 2,-12, 24,2, -8,  6,2, -4, 24,1,  0, 24,2,  0,  6,1,  0, 15,2, 10, 15,1, 10, 24,&
 460 2, 10,  6,1, 18, 24,2, 18,  6,1, 16, 24,2, 20, 24,1, 16,  6,2, 20,  6,0,194, 25,&
 461 1,-20, 18,2,-20,  6,2,-16,  6,1,-14,  6,2,-14, 24,2, -6, 24,2, -4, 21,2, -4, 18,&
 462 2, -6, 15,2,-14, 15,1,-10, 15,2, -4,  6,1,  8, 24,2,  0, 24,2,  0,  6,2,  8,  6,&
 463 1,  0, 15,2,  6, 15,1, 12, 24,2, 18, 24,2, 20, 18,2, 20, 12,2, 18,  6,2, 12,  6,&
 464 2, 12, 24,0,195, 22,1,-20,  6,2,-20, 24,2,-12, 24,2, -8, 21,2, -8, 18,2,-12, 15,&
 465 2,-20, 15,1,-14, 15,2, -8,  6,1,  6, 24,2, -4, 24,2, -4,  6,2,  6,  6,1, -4, 15,&
 466 2,  4, 15,1, 10, 24,2, 16, 24,2, 20, 18,2, 20, 12,2, 16,  6,2, 10,  6,2, 10, 24,&
 467 0,196, 20,1,-20,  6,2,-20, 24,2,-14, 15,2, -8, 24,2, -8,  6,1, -4,  6,2,  0, 24,&
 468 2,  4,  6,1, -2, 15,2,  2, 15,1, 20, 21,2, 16, 24,2, 12, 24,2,  8, 21,2,  8,  9,&
 469 2, 12,  6,2, 16,  6,2, 20,  9,2, 20, 15,2, 14, 15,0,197, 14,1,-20, 24,2,-14, 15,&
 470 2, -8, 24,1,-14, 15,2,-14,  6,1,  6, 24,2, -4, 24,2, -4,  6,2,  6,  6,1, -4, 15,&
 471 2,  2, 15,1, 10, 24,2, 10,  6,2, 20,  6,0,198, 26,1,-20, 18,2,-20,  6,2,-16,  6,&
 472 1, -4, 21,2, -8, 24,2,-10, 24,2,-14, 21,2,-14,  9,2,-10,  6,2, -8,  6,2, -4,  9,&
 473 2, -4, 15,2, -8, 15,1,  0,  6,2,  0, 24,2,  6, 24,2,  8, 21,2,  8, 18,2,  6, 15,&
 474 2,  0, 15,1,  2, 15,2,  8,  6,1, 12,  6,2, 12, 24,2, 20,  6,2, 20, 24,0,199, 23,&
 475 1,-10, 21,2,-14, 24,2,-16, 24,2,-20, 21,2,-20,  9,2,-16,  6,2,-14,  6,2,-10,  9 &
 476                                                                                  /
 477         data PointLoc(3601:4200) / &  !~ASCII characters 200-207
 478 2,-10, 15,2,-14, 15,1, -6,  6,2, -6, 24,2,  0, 24,2,  4, 21,2,  4, 18,2,  0, 15,&
 479 2, -6, 15,1, -2, 15,2,  4,  6,1,  8,  6,2,  8, 24,2, 20,  6,2, 20, 24,0,200, 17,&
 480 1, -8, 21,2,-12, 24,2,-16, 24,2,-20, 21,2,-20,  9,2,-16,  6,2,-12,  6,2, -8,  9,&
 481 1, -6, 24,2,  0, 15,2,  6, 24,1,  0, 15,2,  0,  6,1, 10,  6,2, 10, 24,2, 20,  6,&
 482 2, 20, 24,0,201, 24,1,-20, 18,2,-20,  6,2,-16,  6,1,-14, 24,2, -6, 24,2, -4, 21,&
 483 2, -4, 18,2, -6, 15,2,-14, 15,1, -6, 15,2, -4, 12,2, -4,  9,2, -6,  6,2,-14,  6,&
 484 2,-14, 24,1,  0, 24,2,  0,  6,2,  8,  6,1, 10, 24,2, 10,  9,2, 12,  6,2, 18,  6,&
 485 2, 20,  9,2, 20, 24,0,202, 21,1,-20, 24,2,-12, 24,2, -8, 21,2, -8, 18,2,-12, 15,&
 486 2,-20, 15,1,-12, 15,2, -8, 12,2, -8,  9,2,-12,  6,2,-20,  6,2,-20, 24,1, -4, 24,&
 487 2, -4,  6,2,  6,  6,1,  8, 24,2,  8,  9,2, 12,  6,2, 16,  6,2, 20,  9,2, 20, 24,&
 488 0,203, 27,1,-20, 18,2,-20,  6,2,-16,  6,1, -4, 21,2, -8, 24,2,-10, 24,2,-14, 21,&
 489 2,-14,  9,2,-10,  6,2, -8,  6,2, -4,  9,2, -4, 15,2, -8, 15,1,  0,  6,2,  0, 24,&
 490 2,  6, 24,2,  8, 21,2,  8, 18,2,  6, 15,2,  0, 15,1,  2, 15,2,  8,  6,1, 12, 24,&
```

```
 491 2, 16, 15,2, 20, 24,1, 16, 15,2, 16,  6,0,204, 24,1,-10, 21,2,-12, 24,2,-18, 24,&
 492 2,-20, 21,2,-20,  9,2,-18,  6,2,-12,  6,2,-10,  9,2,-10, 15,2,-14, 15,1, -6,  6,&
 493 2, -6, 24,2,  2, 24,2,  4, 21,2,  4, 18,2,  2, 15,2, -6, 15,1, -2, 15,2,  4,  6,&
 494 1,  8, 24,2, 14, 15,2, 20, 24,1, 14, 15,2, 14,  6,0,205, 25,1,-20, 24,2,-12, 24,&
 495 2,-10, 21,2,-10, 18,2,-14, 15,2,-20, 15,1,-14, 15,2,-10, 12,2,-10,  9,2,-12,  6,&
 496 2,-20,  6,2,-20, 24,1, -6,  6,2, -6, 24,2,  2, 24,2,  4, 21,2,  4, 18,2,  2, 15,&
 497 2, -6, 15,1, -2, 15,2,  4,  6,1,  8,  6,2,  8, 24,2, 20,  6,2, 20, 24,0,206, 23,&
 498 1,-20,  6,2,-20, 24,2,-12, 24,2,-10, 21,2,-10, 18,2,-12, 15,2,-20, 15,1, -6,  6,&
 499 2, -6, 24,2,  2, 24,2,  4, 21,2,  4, 18,2,  2, 15,2, -6, 15,1, -2, 15,2,  4,  6,&
 500 1,  8,  6,2,  8, 24,2, 16, 24,2, 18, 21,2, 18, 18,2, 16, 15,2,  8, 15,0,207, 22,&
 501 1,-20, 24,2,-12, 24,2,-10, 21,2,-10, 18,2,-14, 15,2,-20, 15,2,-20, 24,1,-14, 15,&
 502 2,-10, 12,2,-10,  9,2,-12,  6,2,-20,  6,2,-20, 15,1, -6, 24,2, -6,  6,2,  4,  6 &
 503                                                                                 /
 504         data PointLoc(4201:4800) / &  !~ASCII characters 208-221
 505 1,  8, 24,2,  8,  6,1,  8, 15,2, 20, 24,1, 12, 18,2, 20,  6,0,208, 26,1,-20, 27,&
 506 2,-16, 27,2,-14, 24,2,-14, 18,2,-16, 15,2,-20, 15,2,-20, 27,1, -6, 27,2, -4, 24,&
 507 2, -4, 18,2, -6, 15,2, -8, 15,2,-10, 18,2,-10, 24,2, -8, 27,2, -6, 27,1,  0, 15,&
 508 2,  0, 27,2,  6, 27,1,  0, 21,2,  4, 21,1, 10, 15,2, 10, 27,2, 16, 27,1, 10, 21,&
 509 2, 14, 21,0,209,  8,1,-18, 27,2,-14, 15,2,-10, 27,2, -6, 15,2, -2, 27,1,  8, 24,&
 510 2, 10, 27,2, 10, 15,0,210, 13,1,-18, 27,2,-14, 15,2,-10, 27,2, -6, 15,2, -2, 27,&
 511 1,  4, 24,2,  8, 27,2, 12, 27,2, 16, 24,2, 16, 21,2,  4, 18,2,  4, 15,2, 16, 15,&
 512 0,211, 14,1,-18, 27,2,-14, 15,2,-10, 27,2, -6, 15,2, -2, 27,1,  4, 24,2,  8, 27,&
 513 2, 12, 27,2, 16, 24,2, 10, 21,2, 16, 18,2, 12, 15,2,  8, 15,2,  4, 18,0,212, 10,&
 514 1,-18, 27,2,-14, 15,2,-10, 27,2, -6, 15,2, -2, 27,1,  4, 27,1, 10, 15,2, 10, 27,&
 515 2,  4, 21,2, 16, 21,0,213, 12,1,-18, 27,2,-14, 15,2,-10, 27,2, -6, 15,2, -2, 27,&
 516 1, 16, 27,2,  6, 27,2,  4, 21,2, 14, 21,2, 16, 18,2, 12, 15,2,  4, 15,0,214, 15,&
 517 1,-20, 24,2,-12, 24,1,-16, 24,2,-16,  6,1,-20,  6,2,-12,  6,1, -8, 24,2,  8, 24,&
 518 1,  0, 24,2,  0,  6,1,  4,  6,2, 12, 24,2, 20,  6,1,  8, 15,2, 16, 15,0,215, 17,&
 519 1,-20, 24,2,-20, 12,2,-16,  6,2,-12,  6,2, -8, 12,2, -8, 24,1, -4,  6,2, -4, 24,&
 520 2,  6,  6,2,  6, 24,1, 10, 24,2, 16, 24,2, 20, 18,2, 20, 12,2, 16,  6,2, 10,  6,&
 521 2, 10, 24,0,216, 28,1, -8, 21,2,-12, 24,2,-16, 24,2,-20, 21,2,-20, 18,2, -8, 12,&
 522 2, -8,  9,2,-12,  6,2,-16,  6,2,-20,  9,1, -4, 24,2, -4,  9,2,  0,  6,2,  2,  6,&
 523 2,  6,  9,2,  6, 24,1, 10,  6,2, 10, 24,2, 16, 24,2, 20, 21,2, 20, 18,2, 16, 15,&
 524 2, 10, 15,1, 16, 15,2, 20, 12,2, 20,  9,2, 16,  6,2, 10,  6,0,217, 23,1, -8, 21,&
 525 2,-12, 24,2,-16, 24,2,-20, 21,2,-20, 18,2, -8, 12,2, -8,  9,2,-12,  6,2,-16,  6,&
 526 2,-20,  9,1, -4, 24,2, -4,  9,2,  0,  6,2,  2,  6,2,  6,  9,2,  6, 24,1, 10,  6,&
 527 2, 10, 24,2, 16, 24,2, 20, 21,2, 20, 18,2, 16, 15,2, 10, 15,0,221, 27,1,-20, 27,&
 528 2,-14, 27,2,-12, 24,2,-12, 18,2,-14, 15,2,-20, 15,2,-20, 27,1, -2, 24,2, -4, 27,&
 529 2, -8, 27,2,-10, 24,2, -2, 18,2, -4, 15,2, -8, 15,2,-10, 18,1, 10, 27,2,  2, 27 &
 530                                                                                 /
 531         data PointLoc(4801:5199) / &  !~ASCII characters 224-239
 532 2,  2, 15,2, 10, 15,1,  2, 21,2,  8, 21,1, 20, 27,2, 12, 27,2, 12, 15,2, 20, 15,&
 533 1, 12, 21,2, 18, 21,0,224, 24,1, -7, -9,2, -7, 21,2, 10, 21,2, 10, 18,2, -4, 18,&
 534 2, -4,  6,2,  8,  6,2,  8,  3,2, -4,  3,2, -4, -9,2, -7, -9,1,  0, 30,2, 12, 27,&
 535 2, 18, 21,2, 20,  6,2, 18, -9,2, 12,-15,2,  0,-18,2,-12,-15,2,-18, -9,2,-20,  6,&
 536 2,-18, 21,2,-12, 27,2,  0, 30,0,225, 17,1, 12,  6,2,  6, 12,2, -6, 12,2,-12,  6,&
 537 2,-12, -6,2, -6,-12,2,  8,-12,2, 12, -6,1, 10, 21,2, 20, 12,2, 20,-12,2, 10,-21,&
 538 2,-10,-21,2,-20,-12,2,-20, 12,2,-10, 21,2, 10, 21,0,226, 18,1, -8, -3,2, -8, 21,&
 539 2,  4, 21,2,  8, 18,2,  8, 12,2,  4,  9,2, -8,  9,1,  2,  9,2,  8, -3,1,  8, 30,&
 540 2, 20, 21,2, 20, -3,2,  8,-12,2, -8,-12,2,-20, -3,2,-20, 21,2, -8, 30,2,  8, 30,&
 541 0,227, 17,1,-18, 24,2,-18, 27,2, -2, 27,2, -2, 24,1,-10, 27,2,-10,  3,1,-12,  3,&
 542 2, -8,  3,1,  0,  3,2,  4,  3,1,  2,  3,2,  2, 27,2, 10, 15,2, 18, 27,2, 18,  3,&
 543 1, 16,  3,2, 20,  3,0,228, 16,1, 12,  6,2,  6, 12,2, -6, 12,2,-12,  6,2,-12, -6,&
 544 2, -6,-12,2,  8,-12,2, 12, -6,1, 20, 21,2, 20,-12,2, 10,-21,2,-20,-21,2,-20, 12,&
 545 2,-10, 21,2, 20, 21,2,-20,-21,0,238,  5,1, 30, 30,2, 30,-30,2,-30,-30,2,-30, 30,&
 546 2, 30, 30,0,239, 19,1, 30, 30,2, 30,-45,2,-30,-45,2,-30, 30,2, 30, 30,1,-30, 15,&
 547 2, 30, 15,1,-30,  0,2, 30,  0,1,-30,-15,2, 30,-15,1,-30,-30,2, 30,-30,1,-14, 30,&
 548 2,-14,-45,1,  0, 30,2,  0,-45,1, 14, 30,2, 14,-45 /
 549 !PointLoc(5199) defined above.                                  JMS 2018.08.16.2120
 550
 551 !-----------------------------------------------------------------------------7 9
 552 !contains
 553 !-----------------------------------------------------------------------------7 9
 554 End Module CharDef
 555 !-----------------------------------------------------------------------------7 9
 556 !-----------------------------------------------------------------------------7 9
 557
 558 !-----------------------------------------------------------------------------7 9
 559 Subroutine AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iCol,fLineWidth,iP)
 560 !2018.09.12.1035cdt JMS- Modified for TweakEngine to output to a .3dv file.
```

```
!                         PosLLCq,RpyDq, & SizeHq are quad precision.
!2018.08.16.2210cdt JMS- Square grid.
!2018.04.06.1620cdt JMS- 7DoF vector-based characters;
!                          the individual character shapes are specified.
!                      - Traveler2/Athlon64/WinXPPro-32/APF9.0-32

!This supports both: font modification and extension  -and-
!                    exporting characters as 3D lines.
!2016.06.24.1600cdt JMS- Worked in C - but - color of first character was
!                                              "kluged-to-correctness".

! Unrotated characters are written parallel-to the +X axis,
!               readable from the +Y direction
! in righthanded "Flight Simulation" (abbrev. 'Fs') coordinates.
!
! Function 'HindSight'- with S.ThreePhase=3: uses Fs-based screen coordinates
!                                 "2D/3D"      for the various projection matrices
!                                              +X:forward,+Y:right,+Z:down.
!                  - with S.ThreePhase=2: uses glOrtho (which is non-FS).
!                                 "2D ortho"   +X:right,+Y:up, +Z:forward,
!                                              a left-handed coordinate frame.
!
! Use 'PrntOrtho' to write stationary color 2D text on the screen.

!Function 'Colors3D' defines the colors

!VecText.h has the following c* & d* characters encoded,
!   ... however, they presently have no useful effect.
!The objective is to enrich the formatting within a text file
!   by repurposeing some relatively unused characters.
!The key question is: How to easily insert hex code characters in text?

!Custom character usages:
!My colors:        [c][0] 'move without draw' - a no-op in this context
!                           (clarifies & simplifies the content of .3dv files.)
!          White   [c][1]
!          Red     [c][2]
!  !Green= Magenta [c][3]   <- ! = "not"
!  !Blue = Yellow  [c][4]
!    light Green   [c][5]
!          Green   [c][6]
!  !Red  = Cyan    [c][7]
!    light Blue    [c][8]
!          Blue    [c][9]
!    light Gray    [c][a]
!          Gray    [c][b]
!          Brown   [c][c]
!          Purple  [c][d]
!  !White  Black   [c][e]
!      user-defined[c][f]

! Shape modifiers  [d][0]    toggles-> off
!    Line width    [ ][1] =1.
!                  [ ][2] =2.
!                  [ ][3] =3.
!                  [ ][4] =4.
!                  [ ][5] =5.
!    Italics       [ ][6]    toggle on/off
!    Underlined    [ ][7]      "
!    Subscript     [ ][8]      "
!    Superscript   [ ][9]      "
!                  [ ][a]
!                  [ ][b]
!                  [ ][c]
!    Show controls [ ][d]      "
!                  [ ][e]
!                  [ ][f]                                                 */

 !-----
 use CharDef                                                      !2018.03.21
```

```
 631  use Tweakrec, only: TL2 !Used in calling Draw3dJTL(TL2,0) to draw every line
 632 !                          segment of each character.
 633 !use opengl_gl
 634  implicit none                                                  !arguments
 635
 636
 637   character:: cLabelL*80 !Label- text
 638   real*16  :: PosLLCq(3) !     - position of the lower left corner (not homog.)
 639   real*16   :: RpyDq(3)   !     - Roll,pitch, & yaw of the label    (not homog.)
 640   real*16   :: SizeHq     !     - size
 641   integer*4:: iCol        !     - color
 642   real*4   ::fLineWidth   !     - line width
 643   integer*4:: iP       !Print flag. Prints when iP>5 to unit# IP.
 644   !--                                                            !Internals
 645   real*8   :: PosLLC(3)  !     - position of the lower left corner (not homog.)
 646   real*8   :: RpyD(3)    !     - Roll,pitch, & yaw of the label    (not homog.)
 647   real*8   :: SizeH      !     - size
 648
 649   integer*4::i,iPass,j,j1,j2,k,m,n,nCh,nChars,nPts,nPu
 650   character:: cA*80, cAi(80)*1      ;equivalence(cA,cAi)
 651   character:: cB*1 ;integer*4:: iB  ;equivalence(cB,iB )
 652   !--
 653   real*8   ::Pin(3)
 654   real*8   ::Pout(3)
 655
 656   real*8   ::pXyz(4)   !Homog.i/o- pos.(X,Y,Z,1.)      - per FSCoords  [meters]
 657   real*8   ::aRpy(4)   !         - att.(Roll,Pitch,Yaw)-              [degrees]
 658   real*8   ::hXR(4,4)  !         - trans & rot matrix- i/o       [i/o:h-meters]
 659  !integer*4::Mode       !=1:p&a->h & hinv
 660   real*8   ::VecBehind(3)
 661   integer*4::iColor    !DOS color
 662    !---                                                          !--EndDefs
 663
 664    !-- Compute the text's translation & rotation offset H-matrix:
 665    PosLLC   = dble(PosLLCq)
 666    RpyD     = dble(RpyDq  )
 667    SizeH    = dble(SizeHq )
 668
 669    pXyz(1:3)=PosLLC; pXyz(4)=1.d0
 670    aRpy(1:3)=RpyD;   aRpy(4)=1.d0
 671
 672    call Model7DoF(pXyz,aRpy,1.d0,hXR,iP)
 673
 674    VecBehind=-hXR(1:3,2)*SizeH/50.d0
 675   !----- Render the text
 676    nChars=len_trim(cLabelL)
 677    cA=Char(0)
 678    cA(1:nChars)=cLabelL(1:nChars)
 679
 680    if(iP.gt.5) write(iP,"('AlphaJS: ',i2,1x,a80)") nChars,cLabelL(1:nChars)
 681
 682    m= 0 !Zero the line counter. 0 is the index of the first line.
 683    n=-1 !Set the character offset to 'before the first character' of the line
 684    !Process the incoming label... character by character... starting at (1):
 685    do i=1,nChars; cB=cAi(i); iB=iand(iB,255); nCh=iB; if(iB.le.0) exit
 686      if(iP.gt.5) write(iP,"('i= ',i2,1x,a1,1x,i3)") i,cB,nCh
 687
 688           ! Recognize & utilise some of the traditonal control characters
 689           !   to easily write paragraphs in 7dof space:
 690      if((nCh.ge.0).and.(nCh.lt.32)) then
 691        select case(nCh)
 692         case( 8); n= n-1                    !BackSpace                  '\b'
 693         case( 9); n=(n+1)/8                 !Horizontal Tab- 8 wide     '\t'
 694                   n=(n+1)*8-1
 695         case(11); m=(m  )/5                 !Vertical   Tab- 5 high     '\v'
 696                   m=(m+1)*5
 697         case(10); m= m+1                    !Line Feed                  '\n'
 698         case(13); n=  -1                    !Carriage Return            '\r'
 699         case(12); m= m+80                   !Form Feed                  '\f'
 700        end select!nCn
```

```
  701        cycle
  702      endif!0 <= nCh <32
  703      n=n+1;                                  !move right by one character.
  704      if(nCh.lt. 32) nCh= 32
  705      if(nCh.gt.255) nCh=255
  706      j1   = CharLoc(nCh);
  707      if(j1.eq.0) cycle   !if the character isn't defined- skip to the next one.
  708      j1   = j1           !This is the beginning of the character's information
  709      nPts = PointLoc(j1+2)
  710      if(nPts.eq.0) cycle !if the charcter has no points- go to the next one.
  711      j2   = j1+(1+nPts)*3!This is the beginning of the following character
  712
  713                          nPu=    0;      k =   0
  714      if(iP.gt.5) write(iP,"('i= ',i2,1x,a1,1x,i3,i3,i2)") i,cB,nCh,nPts,iPass
  715 !    Draw the intended character
  716      iColor=iCol !The incoming colors are DOS colors already.
  717
  718      do j=j1+3,J2-1,3; nPu=nPu+1  !Step through the character's points
  719        if(PointLoc(j).gt.2) stop "Huge problem!"
  720        if(PointLoc(j).eq.1) k = 0
  721                             k = k + 1
  722       !Draw (or move):   Resize first, the 7th dof, and increment:
  723        Pin(1) = (  (PointLoc(j+1)+30.d0) /60.d0+ n      ) *SizeH
  724        Pin(2) =     0.d0
  725        Pin(3) = ( -(PointLoc(j+2)+30.d0) /60.d0 +m*1.5d0) *SizeH !/20.
  726        !Then translate & rotate... the other 6 dof's:
  727        Pout(1)=hXR(1,1)*Pin(1)+hXR(1,3)*Pin(3)+hXR(1,4)
  728        Pout(2)=hXR(2,1)*Pin(1)+hXR(2,3)*Pin(3)+hXR(2,4)
  729        Pout(3)=hXR(3,1)*Pin(1)+hXR(3,3)*Pin(3)+hXR(3,4)
  730                  TL2.iC  = iColor;
  731        if(K == 1) TL2.iC  = 0
  732                  TL2.Th  = fLineWidth
  733                  TL2.XYZ = quad(Pout)                  ;call Draw3dJTL(TL2,0)
  734      enddo!j
  735    enddo!i
  736                                                                        return
  737 End Subroutine AlphaJS
  738 !-------------------------------------------------------------------------------7 9
  739
  740 Subroutine Model7DoF(Xyzh,Rpyh,S,Model7h,iP)              !version: 2018.08.03
  741  !        ="Model 7DoF"          ... "DoF" = "Degrees of Freedom"
  742  !Translation is 3DoF, Rotation is 3Dof, & Scale is 1Dof => 7DoF
  743  !The general transform for inserting rigid 3D objects into 3D views.
  744  !This sub routine concatinates (~mashes) all seven effects into a single 4x4
  745  !   matrix; the pre-concatinated symbolic sub matrices are:
  746                                                  !The inverse sequence is:
  747  !1. S = Scale (3D Magnify) the object:            unTranslate
  748  !   +S , 0 , 0 , 0                                 +1  , 0  , 0  ,-X
  749  !    0 ,+S , 0 , 0                                  0  ,+1  , 0  ,-Y
  750  !    0 , 0 ,+S , 0                                  0  , 0  ,+1  ,-Z
  751  !    0 , 0 , 0 ,+1                                  0  , 0  , 0  ,+1
  752  !    (The reciprocals of a diagonal scaling matrix are the inverse.)
  753  !2. Roll:  sR=sine(Roll), cR=cosine(Roll)         unYaw
  754  !   +1 , 0 , 0 , 0                                 +cY ,+sY , 0  , 0
  755  !    0 ,+cR,-sR, 0                                 -sY ,+cY , 0  , 0
  756  !    0 ,+sR,+cR, 0                                  0  , 0  ,+1  , 0
  757  !    0 , 0 , 0 ,+1                                  0  , 0  , 0  ,+1
  758  !3. Pitch: sP=sine(Pitch),cP=cosine(Ptch)         unPitch
  759  !   +cP, 0 ,+sP, 0                                 +cP , 0  ,-sP , 0
  760  !    0 ,+1 , 0 , 0                                  0  ,+1  , 0  , 0
  761  !   -sP, 0 ,+cP, 0                                 +sP , 0  ,+cP , 0
  762  !    0 , 0 , 0 ,+1                                  0  , 0  , 0  ,+1
  763  !4. Yaw:   sY=sine(Yaw)  ,cY=cosine(Yaw)          unRoll
  764  !   +cY,-sY, 0 , 0                                 +1  , 0  , 0  , 0
  765  !   +sY,+cY, 0 , 0                                  0  ,+cR ,+sR , 0
  766  !    0 , 0 ,+1 , 0                                  0  ,-sR ,+cR , 0
  767  !    0 , 0 , 0 ,+1                                  0  , 0  , 0  ,+1
  768  !  (      The transpose of a pure-rotation matrix is the inverse;
  769  !    this remains true even after [Yaw]<-[Pitch]<-[Roll] concatenation. )
  770  !5. Translate to (X,Y,Z):                         unSscale
```

```
!   +1 , 0 , 0 ,+X                                      +1/S, 0  , 0  , 0
!    0 ,+1 , 0 ,+Y                                       0  ,+1/S, 0  , 0
!    0 , 0 ,+1 ,+Z                                       0  , 0  ,+1/S, 0
!    0 , 0 , 0 ,+1                                       0  , 0  , 0  ,+1
!    (The negatives of a pure-translation matrix are the inverse.)

!Matrix concatenation sequence:
!         [5:translate]<-[4:Scale]<-[3:Yaw]<-[2:Pitch]<-[  1:Roll   ]
! & for the inverse:
!         [   unRoll  ]<-[unPitch]<-[unYaw]<-[unScale]<-[untranslate]

!The resulting (4,4) matrix is implemented below.

!-----
implicit none                                                       !arguments
                        !The seven "Degrees of Freedom"(DoF) are:
 real*8   ::Xyzh(4)     !"translation":  X   ,   Y  ,Z   {,h}   1,2,3
 real*8   ::Rpyh(4)     !" rotation  ": Roll,Pitch,Yaw {,h}   4,5,6
 real*8   ::S           !"  scale    ":      scale              7

 real*8   ::Model7h(4,4) !Resulting homogeneous 7DoF matrix output.

 integer*4::iP          !Write enable>5: write(iP,...)
!---                                                                 !internals
 real*8   :: X   ,  Y   ,   Z     !Translations
 real*8   ::Roll , Pitch,  Yaw    !Rotations
 real*8   ::sr,cr, sp,cp, sy,cy   !Sines & Cosines
 real*8   :: H16(16)
 integer*4::i,j,k
 !-----
 !Extract non-homogeneous values and compute the rotation sines & cosines:
  X    =Xyzh(1)/Xyzh(4)
  Y    =Xyzh(2)/Xyzh(4)
  Z    =Xyzh(3)/Xyzh(4)
  Roll =Rpyh(1)/Rpyh(4); sr=dsind(Roll ); cr=dcosd(Roll )
  Pitch=Rpyh(2)/Rpyh(4); sp=dsind(Pitch); cp=dcosd(Pitch)
  Yaw  =Rpyh(3)/Rpyh(4); sy=dsind(Yaw  ); cy=dcosd(Yaw  )

 !   Translate(X,Y,Z) <- Scale(S) <- Rotate(Roll,Pitch,Yaw)... concatenated.
 !Model 7DoF homogeneous transform:  ------------------------ 2017.05.23.0900
  H16= (/  S*cY*cP , S*(cY*sP*sR-sY*cR) , S*(cY*sP*cR+sY*sR) , X    &
         , S*sY*cP , S*(sY*sP*sR+cY*cR) , S*(sY*sP*cR-cY*sR) , Y    &
         ,-S*sP    , S*cP*sR            , S*cP*cR            , Z    &
         , 0.d0    , 0.d0               , 0.d0               ,+1.d0  /)
  ! ...pack the results in their (4,4) array:
  k=0; do i=1,4; do j=1,4; k=k+1; Model7h(i,j)=H16(k); enddo; enddo!i

!Concatenated Symbolic Inverse:
!    cY*cP       /S,    sY*cP          /S,   -sP   /S , ( -X* cY*cP          /S
!                  -Y* sY*cP          /S  +Z*sP   /S   )
!(cY*sP*sR-sY*cR)/S,   (sY*sP*sR+cY*cR)/S,    cP*sR/S , ( -X*(cY*sP*sR-sY*cR)/S
!                  -Y*(sY*sP*sR+cY*cR)/S  -Z*cP*sR/S   )
!(cY*sP*cR+sY*sR)/S,   (sY*sP*cR-cY*sR)/S,    cP*cR/S , ( -X*(cY*sP*cR+sY*sR)/S
!                  -Y*(sY*sP*cR-cY*sR)/S  -Z*cP*cR/S   )
! 0               ,    0               ,    0       , (+1)

!Fact N  : Numeric matrix inversion sidesteps solving for symbolic inverses.

   if(iP>5) then
     write(iP,"(/'sub:Model7DoF: ',45('-'))")
     write(iP,"(9x,'user units',13x,'degrees',11x,'multiplier')")
     write(iP,"(' X    =',f12.6,'   Roll=',f12.6,'   Scale=',f12.6,)") &
                      X          ,      Roll      ,      S
     write(iP,"(' Y    =',f12.6,'  Pitch=',f12.6)")  Y,Pitch
     write(iP,"(' Z    =',f12.6,'    Yaw=',f12.6)")  Z,Yaw
     write(iP,"( 'Model7h:'          )")
     do i=1,4
       write(iP,"(7x,4f12.6)") (Model7h(i,j),j=1,4)
     enddo!i
   endif !iP>5
```

```
  841
  842    return
  843 End Subroutine Model7DoF
  844 !-------------------------------------------------------------------------------7 9
  845
```