```
!Skew-T Program
!Designed to read in an ascii text file holding an atmospheric sounding,
!and output an ascii file for easy plotting, such as by Excel.
!Copyright © 2007 by Roland Stull
!version 13.  4 Feb 2007


!============ modules =====================

module soundmod                          !holds sounding arrays
    character (len=50) :: filename       !holds name of sounding file
    character (len=60) :: outname        !holds name of output file
    character (len=100) :: title         !title line for sounding
    integer :: nlines                    !number of lines in sounding, initialized to 0
    integer, parameter :: maxlines = 120   !max sounding lines that can be captured
    real, dimension(maxlines) :: P       !Pressure (kPa)
    real, dimension(maxlines) :: z       !Height (m)
    real, dimension(maxlines) :: T       !Temperature (C)
    real, dimension(maxlines) :: Td      !Dew Point (C)
    character (len=1) :: tab=achar(9)    !ascii tab character
end module soundmod

!============ main program ================
program skewtmain                    !Skew-T plotting main program

!declare variables
    implicit none                    !enforce strong typing

!set-up
    call welcome                     !welcome the user
    call getfile                     !read in the sounding file
    call prepareoutput               !prepare disk to receive output

!compute sounding plot data
    call plotsounding                !plot the sounding
    call plotisobar                  !plot the isobars in the background
    call plotisotherm                !plot a few isotherms in the background
    call plotdryadiabat              !plot a few dry adiabats in background
    call plotwetadiabat              !plot a few saturated adiabats in backgr
    call plotisohume                 !plot a few isohumes in the background

!finish
    call cleanup                     !close files and release memory

end program skewtmain



!================================================
subroutine welcome               !Welcome the interactive user
    implicit none                    !enforce strong typing
    write(*,*)
    write(*,*) "================================================="
    write(*,*) "Welcome to Skew-T ... a Sounding Plotting Program"
    write(*,*) "================================================="
    write(*,*)
    write(*,*) "   Based on Stull,R., 2007: Meteorology for Sci & Engr,3rd Ed."
```

```
58      write(*,*) "   Chapters 1 - 6.    Coded by R. Stull, UBC, Feb 2007."
59      write(*,*)
60      write(*,*) "Expects ascii 'Text List' sounding as input, as from U.Wyoming site:"
61      write(*,*) "   http://weather.uwyo.edu/upperair/sounding.html"
62      write(*,*) "Produces .xls tab-delimited output file, which can be read into"
63      write(*,*) "   Excel as one long Series for plotting."
64      write(*,*)
65  end subroutine welcome
66
67
68  !=============================================
69  subroutine getfile                          !read in the sounding file
70      use soundmod                            !use sounding module
71      implicit none                           !enforce strong typing
72      character (len=100) :: line             !holds one line from the sounding
73      integer :: ios                          !input/output error status
74      integer :: i                            !dummy iteration counter
75      real :: PhPa, zm, TC, TdC               !temporary Pressure, height, Temp, Td variabl
76
77      write(*,*) "...starting getfile"        !debugging output
78
79  !First, ask the user for the file name that holds the sounding ascii data
80      do i = 1,3                              !give user 3 tries to enter correct file
81          write(*,*)
82          write(*,"(a)",advance="no") "Type in file name for sounding, then hit Enter: "
83          read(*,*) filename
84
85  !Next, open the file
86          open(1,file=filename,status="old",iostat=ios)   !connect to sounding file
87          if (ios .ne. 0) then                            !Can't open the file
88              write(*,*) "Sorry, can't find file: ",filename
89              write(*,*) "Don't forget to include the suffix .txt in the file name."
90              if (i < 3) write(*,*) "Try again."
91              cycle                                       !allow user to try again
92          else                                            !Can open the file
93              write(*,*) "Good.  Successfully opened: ", filename
94              write(*,*)
95              exit                                        !jump out of do loop
96          endif
97      enddo
98      if (ios .ne. 0) stop "Sorry. Perhaps the file doesn't exist.  Bye."
99
100 !At this point, we have a good file.
101
102 !Read title line and skip other header lines in the data file
103     do i = 1,5                                      !for all 5 header lines
104         read(1,"(a)",iostat=ios) line               !ios is newly set here
105         if (ios .ne. 0) exit                        !Cannot read lines
106         if (i .eq. 1) title = line                  !save title
107     enddo
108
109     write(*,*) title                                !Display title
110     write(*,*)
111     write(*,*) " i      P (kPa)      z (m)       T (C)      Td (C)"
112
113 !Read each data line in the sounding file, and echo to screen
114     i = 0                                           !initialize line counter
```

```
115     do                                                  !for each sounding level
116         read(1,"(F7.1, F7.0, 2F7.1)",iostat=ios) PhPa, zm, TC, TdC  !read obs
117         if (ios .ne. 0) exit                            !finished reading lines
118         if ((PhPa < 100.0) .or. (i >= maxlines)) exit   !don't care about low P.
119
120         !At this point, a good line has been found, so save in data arrays.
121         i = i + 1                                       !count data lines
122         P(i) = 0.1*PhPa                                 !store Pressure, convert kPa
123         z(i) = zm                                       !store heights
124         T(i) = TC                                       !store temperatures
125         Td(i) = TdC                                     !store dew points
126         write(*,"(I3,4F12.2)") i, P(i), z(i), T(i), Td(i)   !echo line to screen
127     enddo
128     nlines = i                                          !save line count
129     write(*,*)
130     write(*,*) "Number of sounding lines captured = ",nlines
131
132 !Finished getting the sounding data.  Close the open file.
133     write(*,*)
134     close(1)                        !close the connection to the input file
135 end subroutine getfile
136
137
138 !================================================
139 subroutine prepareoutput                            !prepare disk to receive output
140     use soundmod                                    !include sounding variables
141     implicit none                                   !enforce strong typing
142     integer :: i                                    !position of "." in filename
143     integer :: ios                                  !error status for file opening
144
145     write(*,*) "...starting prepareoutput"          !debugging output
146     write(*,*)
147
148 !Create an output file to hold the results
149     i = index(filename,".") - 1                     !find the character just before "." in the
150     if (i .le. 0) i = len_trim(filename)            !use whole filename if no "."
151     outname = filename(1:i) // "out.xls"             !create a new output file name
152     write(*,*) "Output will be saved in file: ",outname
153     write(*,*)
154     open(2,file=outname,status="replace",iostat=ios)    !open the output file
155     if (ios .ne. 0) then                            !error opening file
156         write(*,*) "Sorry, can't create file: ", outname
157         stop "Bye."
158     endif
159
160 !Write the title (sounding time and place) from the first header line of the input file
161     write(*,*) trim(title)                          !display title on screen
162     write(*,*)
163     write(*,*) "   Isobars:  P (kPa) = 100, 90, 80, 70, ..., 20, 10"
164     write(*,*) "   Isotherms:  T (C) = -80, -60, -40, -20, 0, 20"
165     write(*,*) "   Dry Adiabats: Theta (C) = -20, 0, 20, 40, 60, 80"
166     write(*,*) "   Saturated adiabat: ThetaL (C) = 0, 10, 20, 30"
167     write(*,*) "   Isohumes: r (/kg) = 0.2, 1.0, 5.0, 20.0"
168     write(*,*)
169     write(*,*) "    X(T)    ",tab,"     Y(P)"   !column headers
170
171     write(2,*) trim(title)                          !display title on output file
```

```
172     write(2,*)
173     write(2,*) "   Isobars:  P (kPa) = 100, 90, 80, 70, ..., 20, 10"
174     write(2,*) "   Isotherms:  T (C) = -80, -60, -40, -20, 0, 20"
175     write(2,*) "   Dry Adiabats: Theta (C) = -20, 0, 20, 40, 60, 80"
176     write(2,*) "   Saturated adiabat: ThetaL (C) = 0, 10, 20, 30"
177     write(2,*) "   Isohumes: r (/kg) = 0.2, 1.0, 5.0, 20.0"
178     write(2,*)
179     write(2,*) "    X(T)    ",tab,"     Y(P)"   !column headers
180
181     write(*,*)
182 end subroutine prepareoutput
183
184
185 !================================================
186 subroutine plotsounding                          !plot the sounding
187     use soundmod                                 !include sounding arrays
188     implicit none                                !enforce strong typing
189     integer :: i                                 !dummy index
190     real :: xst, yst                             !function names
191     real :: y                                    !ordinate value
192     real :: xT, xTd                              !abscissa values for T and Td(C)
193     character (len=40) :: fmt                    !format for output
194
195     write(*,*) "...starting plotsounding"        !debugging output
196     write(*,*)
197     write(2,*)                                   !skip line in output file
198
199     fmt = "(F12.4, a1, F12.4)"                   !format for output
200
201 !   For the temperature profile
202     do i=1,nlines                                !for each sounding level
203         y = yst(P(i))                            !find ordinate value
204         xT = xst(T(i),y)
205         if ((xT .ge. -40.) .and. (xT .le. 40.)) then    !inside skewT domain
206             write(*,fmt) xT,tab,y                !Y & x coord for T, show on screen
207             write(2,fmt) xT,tab,y                !Y & x coord fpr T, write in output file
208         endif
209     enddo
210
211     write(*,*)
212     write(2,*)                                   !skip line in output file
213
214 !   For the dew point profile
215     do i=1,nlines                                !for each sounding level
216         y = yst(P(i))                            !find ordinate value
217         xTd = xst(Td(i),y)
218         if ((xTd .ge. -40.) .and. (xTd .le. 40.)) then  !inside skewT domain
219             write(*,fmt) xTd,tab,y               !Y & x coord for Td, show on screen
220             write(2,fmt) xTd,tab,y               !Y & x coord for Td, write in output file
221         endif
222     enddo
223
224     write(*,*)
225     write(2,*)                                   !skip line in output file
226 end subroutine plotsounding
227
228
```

```
!================================================
subroutine plotisobar                              !plot several isobars in background
!Because isobars are straight lines, we need to find only the left and right ends
    use soundmod
    implicit none                                  !enforce strong typing
    integer :: i                                   !dummy height index
    real :: yst                                    !for the function subroutine
    real, parameter :: xcold = -40.0               !degC cold (left) end of isobar
    real, parameter :: xhot = 40.0                 !degC hot (right) end of isobar
    real :: PkPa                                   !pressure (kPa)
    real :: y                                      !Y coordinate on skewT
    character(len=30) :: fmt                       !string holding output format

    write(*,*) "...starting plotisobar"            !debugging output
    write(*,*)
    write(*,*) "Isobars:"
    write(*,*) "    X(T)    ",tab,"     Y(P)"     !column headers for output
    write(*,*)

    fmt = "(F12.4, a1, F12.4)"                     !format for output
    write(2,*)                                     !skip line in output file
    write(2,*)                                     !skip line in output file

    do i = 10, 100, 10                             !for each isobar to be plotted
        PkPa = real(i)                             !convert pressure to real number
        y = yst(PkPa)                              !get Y coordinate on skewT

        write(*,"(a,F10.1)") "P (kPa) = ",PkPa     !debug output on screen

        write(*,fmt) xcold,tab,y                   !write isobar table to screen
        write(*,fmt) xhot,tab,y
        write(*,*)

        write(2,fmt) xcold,tab,y                   !write isobar table to output
        write(2,fmt) xhot,tab,y
        write(2,*)
    enddo

    write(*,*)
    write(2,*)                                     !skip line in output file
end subroutine plotisobar


!================================================
subroutine plotisotherm                            !plot a few isotherms in the background
    use soundmod                                   !include sounding data
    implicit none                                  !enforce strong typing
    real :: xst, yst                               !type declaration for functions
    real :: x, y                                   !temporary coordinates in skewT
    real :: TC                                     !temperature (C)
    real :: PkPa                                   !pressure (kPa)
    integer :: i                                   !index for pressure height
    integer :: j                                   !index for temperature
    character(len=30) :: fmt                       !string holding output format

    write(*,*) "...starting plotisotherm"          !debugging output
    write(*,*)
```

```
286     write(*,*) "Isotherms:"
287     write(*,*) "    X(T)    ",tab,"     Y(P)"    !column headers for output
288
289     fmt = "(F12.4, a1, F12.4)"                   !format for output
290
291     do j = -80, 20, 20                           !for each isotherm
292         TC = real(j)                             !temperature (C)
293         write(*,*)
294         write(*,"(a,F8.1)") "T (C) = ", TC
295         write(2,*)
296
297         do i = 100, 10, -5                       !for each pressure height
298             PkPa = real(i)                       !pressure (kPa)
299             y = yst(PkPa)                        !y coordiinate of skewT
300             x = xst(TC,y)                        !x coordinate of skewT
301
302             write(*,fmt) x,tab,y                 !x & y coord, show on screen
303             if ((x .ge. -40.) .and. (x .le. 40.)) then  !inside skewT domain
304                 write(2,fmt) x,tab,y             !x & y coord, write to output
305             endif
306         enddo
307     enddo                                        !end of isotherm loop
308
309     write(*,*)
310 end subroutine plotisotherm
311
312
313 !================================================
314 subroutine plotdryadiabat                        !plot a few dry adiabats in background
315     use soundmod                                 !include sounding data
316     implicit none                                !enforce strong typing
317     real :: xst, yst                             !type declaration for functions
318     real :: x, y                                 !temporary coordinates in skewT
319     real :: TC, TK                               !temperature (C) & (K)
320     real :: thetaC, thetaK                       !potential temperature (C) & (K)
321     real :: PkPa                                 !pressure (kPa)
322     real, parameter :: RdCp = 0.28571            !Rd/Cp in eq for pot temperature
323     real, parameter :: Po = 100.                 !reference pressure (kPa)
324     integer :: i                                 !index for pressure height
325     integer :: j                                 !index for potential temperature
326     character(len=30) :: fmt                     !string holding output format
327
328     write(*,*) "...starting plotdryadiabat"      !debugging output
329     write(*,*)
330     write(*,*) "Dry Adiabats:"
331
332     fmt = "(F12.4, a1, F12.4)"                   !format for output
333     write(2,*)
334     write(2,*)
335
336     do j = -20, 80, 20                           !for each adiabat
337         thetaC = real(j)                         !potential temperature (C)
338         thetaK = thetaC + 273.                   !potential temperature (K)
339         write(*,*)
340         write(*,"(a,F8.1)") "Theta (C) = ", thetaC
341         write(*,*) "    P (kPa)   ", tab, "    T (C)"
342         write(2,*)
```

```
343
344         do i = 100, 10, -5                          !for each pressure height
345             PkPa = real(i)                          !pressure (kPa)
346             TK = thetaK*( (PkPa/Po)**RdCp )         !temperature (K)
347             TC = TK - 273.                          !temperature (C)
348             write(*,fmt) PkPa, tab, TC              !debugging output
349
350             y = yst(PkPa)                           !y coordiinate of skewT
351             x = xst(TC,y)                           !x coordinate of skewT
352
353             if ((x .ge. -40.) .and. (x .le. 40.)) then  !inside skewT domain
354                 write(2,fmt) x,tab,y                !x & y coord, write to output
355             endif
356
357         enddo                                       !end of pressure height loop
358     enddo                                           !end of adiabat loop
359
360     write(2,*)
361     write(*,*)
362 end subroutine plotdryadiabat
363
364
365 !=================================================
366 subroutine plotwetadiabat                           !plot a few saturated adiabats in backgr
367     use soundmod                                    !include sounding data
368     implicit none                                   !enforce strong typing
369     integer :: j                                    !thetaL index
370     integer :: i                                    !pressure (height) index
371     real, parameter :: dP = 0.10000                 !pressure increment delta P (kPa)
372     real, parameter :: a = 0.28571                  !dimensionless const for sat adiabat
373     real, parameter :: b = 1.35E7                   !(K^2) constant for sat adiabat
374     real, parameter :: c = 2488.4                   !(K) constant for saturated adiabat
375     real, parameter :: LR = 5423.                   !Lv/Rv (K) in Clausius-Clapeyon eq
376     real, parameter :: eo = 0.611                   !(kPa) reference vapor pressure
377     real, parameter :: Toinv = 1./273.              !inverse of T (K) const in Clausius Clap e
378     real, parameter :: eps = 0.622                  !(g/g) epsilon = ratio of gas constants
379     real :: TC, TK                                  !temperature of rising air parcel (C) and
380     real :: es                                      !(kPa) saturated vapor pressure
381     real :: rs                                      !(g/g) saturated mixing ratio
382     real :: PkPa                                    !pressure (kPa)
383     real :: dTdP                                    !(K/kPa) change of T with P along sat adia
384     real :: num, den                                !temporary numerator & denominator of dT/d
385     real :: xst, yst                                !type declaration for functions
386     real :: x, y                                    !temporary coordinates in skewT
387     character(len=30) :: fmt                        !string holding output format
388
389     write(*,*) "...starting plotwetadiabat"         !debugging output
390     write(*,*)
391     write(2,*)                                      !skip line in output file
392     write(2,*)                                      !skip line in output file
393
394     fmt = "(F12.4, a1, F12.4)"                      !format for output
395
396     do j = 0, 30, 10                                !for each thetaL
397         TC = real(j)                                !initialize thetaL (C)
398         PkPa = 100.                                 !initialize Pressure (kPa)
399         write(*,"(a,F10.1)") "Theta-L (C) = ",TC    !debugging output
```

```
400         write(*,*) "    P(kPa)  ",tab,"     T(C)"   !debugging output
401
402         do i = 1, 1000                              !for each pressure height
403             if (modulo(i,50) .eq. 1) then           !for every 50th data point
404                 write(*,"(F10.1,a1,F10.2)") PkPa, tab, TC   !display data along sat adiaba
405                 y = yst(PkPa)                       !Y coordinate on skewT
406                 x = xst(TC,y)                       !x coordinate on skewT
407                 if ((x .ge. -40.) .and. (x .le. 40.)) then  !inside skewT domain
408                     write(2,fmt) x,tab,y            !x & y coord for Td, write to output file
409                 endif
410             endif
411
412             TK = TC + 273.                          !thetaL (K)
413             es = eo*exp(LR*(Toinv - (1/TK)))        !(kPa) sat vap press from Clausius-Clap
414             rs = (eps*es) / (PkPa - es)             !(g/g) sat mixing ratio
415             num = (a*TK) + (c*rs)                   !numerator of dT/dP eq
416             den = PkPa * (1 + (b*rs/(TK*TK)))       !denominator of dT/dP eq
417             dTdP = num/den                          !temp gradient of sat adiabat
418
419             TC = TC - dTdP*dP                       !new temperature along sat adiabat
420             PkPa = PkPa - dP                        !at new pressure
421             if (PkPa .le. 10.) exit                 !reached top of diagram
422         enddo                                       !end of pressure height loop
423
424         write(2,*)
425         write(*,*)
426     enddo                                           !end of thetaL loop
427
428 end subroutine plotwetadiabat
429
430
431 !================================================
432 subroutine plotisohume                              !plot a few isohumes in the background
433 !Uses Clausius-Clapeyron (eq 3 on crib sheet), for saturation with respect to liquid wate
434     use soundmod                                    !include sounding data
435     implicit none                                   !enforce strong typing
436     integer :: i                                    !index for pressure (height)
437     integer :: j                                    !index for mixing ratio
438     real, dimension(4), parameter :: r = (/ 0.2 ,1. ,5. ,20. /) !mixing ratios (g/kg)
439     real, dimension(5), parameter :: PkPa = (/100., 70., 40., 20., 10. /)   !pressure (kPa
440     real, parameter :: RvLv = 0.0001844             !Rv/Lv, in units of 1/K
441     real, parameter :: Toinv = 1./273.              !inverse of T (K) const in Clausius Clap e
442     real, parameter :: eps = 0.622                  !(g/g) epsilon = ratio of gas constants
443     real, parameter :: eo = 0.611                   !(kPa) reference vapor pressure
444     real :: lnarg                                   !argument of ln in eq (3)
445     real :: sbarg                                   !argument inside square bracket in (3)
446     real :: TdC                                     !dew point (C) temporary variable
447     real :: rg                                      !mixing ratio (g/g) temporary variable
448     real :: x, y                                    !X and Y coordinates on skew T
449     real :: xst , yst                               !type declaration for functions
450     character(len=30) :: fmt                        !string holding output format
451
452
453     write(*,*) "...starting plotisohume"            !debugging output on screen
454     write(*,*)
455     write(*,*) "Isohumes:"
456
```

```
457     fmt = "(F12.4, a1, F12.4)"                  !format for output
458     write(2,*)                                  !skip line in output file
459     write(2,*)                                  !skip line in output file
460
461     do j=1,4                                    !for each humidity
462         write(*,*)                              !debugging output
463         write(*,"(a,F8.2)") "r (g/kg) = ",r(j)          !debugging output
464         write(*,*) "    P(kPa)  ",tab,"     Td(C)"  !debugging output
465         rg = 0.001 * r(j)                       !convert mixing ratio to (g/g)
466
467         do i = 1,5                              !for each pressure
468             lnarg = (rg*PkPa(i)) / (eo*(rg+eps))    !argument of ln in eq (3)
469             sbarg = (Toinv - (RvLv*log(lnarg)))     !argument in square brackets in (3)
470             TdC = (1./sbarg) - 273.             !dew point (C), from eq (3)
471             write(*,fmt) PkPa(i), tab, TdC              !debugging output
472
473             y = yst(PkPa(i))                    !Y coordinate on skew T
474             x = xst(TdC,y)                      !X coordinate on skew T
475
476             if (x .le. 40.) write(2,fmt) x,tab,y    !write coord if inside skewT domain
477         enddo
478
479         write(2,*)                              !blank line between each isohume data
480     enddo
481
482     write(*,*)
483 end subroutine plotisohume
484
485
486 !================================================
487 subroutine cleanup                              !close files and release memory
488     use soundmod                                !including sounding info
489     implicit none                               !enforce strong typing
490     write(*,*) "...starting cleanup"            !debugging output
491     write(*,*)
492
493     write(*,*) title                            !remind user of title
494     write(*,*) "Your sounding output file is: ",outname
495     close(2)                                    !close the output file
496
497     write(*,*)
498     write(*,*) "Bye."
499     write(*,*)
500 end subroutine cleanup
501
502
503 !================================================
504 real function yst (P)                   !finds ordinate (y) value on skewT
505     implicit none                       !enforce strong typing
506     real, intent(in) :: P               !pressure (kPa)
507     real, parameter :: Po = 100.0       !reference pressure (kPa)
508     yst = log(Po/P)                     !ordinate for skewT - logP
509 end function yst
510
511
512 !================================================
513 real function xst (T,y)                 !finds abscissa (x) value on skewT
```

```
514     implicit none                       !enforce strong typing
515     real, intent(in) :: T               !temperature (C)
516     real, intent(in) :: y               !ordinate (dimensionless)
517     real, parameter :: K = 35.0         !reference temperature (C)
518     xst = T + K*y                       !abscissa for skewT - logP
519 end function xst
520
521
522 !======= end ==
523
524
```