Subject: Re: Skewt plot
Posted by afl on Thu, 20 Jul 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3ukr5lINN5go@meteorology.ho.BoM.GOV.AU>, James Kelly
<jamesk@bom.gov.au> writes:
|> Has anyone used IDL for specialised meteorological displays such as
|> skewt? This plots a temperature profile (on a "skewed" x axis) versus
|> height (log of pressure) through the atmosphere.

James,

Yes. I wrote it, but never really used it.
Please tell me if it works for you.

Andy Loughe

```
;============================================================ ============
;  SKEWT.PRO  (IDL CODE)
;
;  Draw a Skew-T, Log(P) diagram given a temperature range for your data.
;
;  Originator:  Andrew F. Loughe  (afl@cdc.noaa.gov)
;               CIRES/NOAA
;               Boulder, CO  USA
;               This code carries no warranty or claim
;               as to its usefulness or accuracy!
;
;  A Number of the functions found in this file were converted from
;  FORTRAN code that was received from NCAR in Boulder, CO USA.
;  The original source of the equations is thought to be:
;    "Algorithms for Generating a Skew-T, Log P Diagram
;     and Computing Selected Meteorological Quantities"
;     by G.S. Stipanuk, White Sands Missle Range, Report ECOM-5515.
;
;============================================================ ============
;  FUNCTION TO COMPUTE SATURATION VAPOR PRESSURE GIVEN TEMP IN KELVIN.
;  ESAT(MILLIBARS), T(KELVIN)
     FUNCTION  ESAT, T
       TC   = T - 273.16
       ESAT = 6.1078 * EXP( (17.2693882 * TC) / (TC + 237.3) )
     RETURN, ESAT
     END
;============================================================ ============
;  FUNCTION TO COMPUTE MIXING RATIO GIVEN TEMP. AND PRESS.
;  W(GRAMS WATER VAPOR/KILOGRAM DRY AIR), P(MILLIBAR)
```

```
    FUNCTION W, T, P
      IF (T GE 999.) THEN GOTO, JUMP10
      X =  ESAT(T)
      W = 621.97 * X / (P - X)
      RETURN, W

JUMP10:  W = 0.0
    RETURN, W
    END
 ;============================================================ ============
;  FUNCTION TO COMPUTE SATURATION ADIABATIC TEMP AT 1000 MB GIVEN T & P.
;  OS AND T (KELVIN), P (MILLIBARS )
    FUNCTION  OS, T, P
      IF (T LT 100.) THEN T1 = T + 273.16
      IF (T GE 100.) THEN T1 = T
      OS = T1 * ((1000./P)^.286) / (EXP( -2.6518986*W(T1,P) / T1) )
    RETURN, OS
    END
 ;============================================================ ============
;  FUNCTION TO COMPUTE THE TEMPERATURE (KELVIN) OF AIR AT A GIVEN
;  PRESSURE AND WITH A GIVEN MIXING RATIO.
;  TMR(KELVIN), W(GRAMS WATER VAPOR/KILOGRAM DRY AIR), P(MILLIBAR)
    FUNCTION  TMR, W, P
      X   =  ALOG10 ( W * P / (622.+ W) )
      TMR = 10. ^ ( .0498646455 * X + 2.4082965 ) - 7.07475 + $
           38.9114 * ( (10.^( .0915 * X ) - 1.2035 )^2 )
    RETURN, TMR
    END
 ;============================================================ ============
;  FUNCTION TO COMPUTE TEMPERATUE (KELVIN) OF A MOIST ADIABAT GIVEN
;  OS(KELVIN), P(MILLIBARS)
;  SIGN(A,B) REPLACES THE ALGEBRAIC SIGN OF A WITH THE SIGN OF B
    FUNCTION TSA, OS, P
       A  = OS
       TQ = 253.16
       D  = 120
       FOR  I = 1, 12 DO BEGIN
         D = D/2.
;  IF THE TEMPERATURE DIFFERENCE, X, IS SMALL, EXIT THIS LOOP.
         X = A * EXP (-2.6518986*W(TQ,P)/TQ)-TQ*((1000./P)^.286)
         IF ( ABS(X) LT 0.01 ) THEN GOTO, JUMP2
;        TQ = TQ + SIGN(D,X)
            IF (X LT 0) THEN D = -ABS(D)
            IF (X GT 0) THEN D =  ABS(D)
            TQ = TQ + D
       ENDFOR
JUMP2:    TSA=TQ
    RETURN, TSA
```

```
    END
;============================================================ ============
;  Function to determine  position (temp, press) when the isotherms
;  in the diagram are rotated (skewed) 45 degrees to the right.
;  Used for finding the points needed to connect the dots when
;  drawing ALL of the lines (except pressure).
;  Originator: Andrew F. Loughe
    Function Tnew, T, P
    COMMON RANGES, trange, prange

      P0   = prange(0)
      xy1  = convert_coord( [T, P0], /data, /to_device)
      xy2  = convert_coord( [T,  P], /data, /to_device)
      dy   = xy2(1) - xy1(1)
      dx   = dy     ; dx = dy for this 45-45-90 triangle
      xy   = convert_coord( [xy2(0)+dx, xy2(1)], /device, /to_data)
      Tnew = xy(0)
    return, Tnew
    end
;============================================================ ============
;  Function to determine  position (temp, press) in the unskewed
;  coordinate system (Opposite of Tnew).
;  Used only when placing the labels on various lines.
;  Originator: Andrew F. Loughe
    Function Told, T, METHOD
    COMMON RANGES, trange, prange

      P0 = prange(0)
      P1 = prange(1)

      T0 = trange(0)
      T1 = trange(1)

      if (method eq 1) then begin
        xy1 = convert_coord( [T,   P0], /data, /to_device )
        xy2 = convert_coord( [T0,  P0], /data, /to_device )
        dx  = xy2(0) - xy1(0)

        xy  = convert_coord( [xy2(0), xy2(1)+dx], /device, /to_data )

        xy1 = convert_coord( [xy(0),  xy(1)], /data, /to_device )
        xy2 = convert_coord( [xy(0),     P1], /data, /to_device )
        dy  = xy2(1) - xy1(1)

        xy = convert_coord([xy1(0)+(dy/2.), xy1(1)+(dy/2.)],$
            /device, /to_data)
      endif
```

```
        if (method eq 2) then begin
          xy1 = convert_coord( [T,   P0], /data, /to_device )
          xy2 = convert_coord( [T1,  P0], /data, /to_device )
          dx  = xy2(0) - xy1(0)

          xy  = convert_coord( [xy1(0)+dx/2. , xy1(1)+dx/2.], $
              /device, /to_data)
        endif

      return, xy
      end
;============================================================ ============
;  Function to determine the necessary trange for plotting the sounding.
     Function T_RANGE, p_range, t, td, p

     COMMON RANGES, trange, prange

       trange   = [-40, 40]         ; Default which can be changed
       if (p_range eq 0) then $
          prange   = [1050, 100]   ; Default which can be changed

;  Find number of data levels
       szd     = size(t)
       nlevels = szd(1)


; Ensure that temperatures are in Celsius.
       if ( (total(t)/nlevels)  gt 100. ) then t  = t  - 273.16
       if ( (total(td)/nlevels) gt 100. ) then td = td - 273.16


;  Set up dummy plot space.
       if (!d.name eq 'PS') then device, /inch, xsize=7, ysize=7
       daspect = FLOAT(!D.Y_SIZE)/FLOAT(!D.X_SIZE) * $
                 (trange(1)-trange(0))/80.
       margin  = 0.1
       aspect  = 1.0 ; A square
       x0 = 0.50 - (0.5 - margin)*(daspect/aspect)
       y0 = margin
       x1 = 0.50 + (0.5 - margin)*(daspect/aspect)
       y1 = 1.0 - margin

       !P.position=[x0,y0,x1,y1]
       plot_io, trange, prange, yrange=prange, /nodata, /xs, /ys, $
           color=!p.background, /noerase
       !p.multi(0) = !p.multi(0) + 1

;  Determine necessary temperature range for the diagram.
```

```
      xx0 = fltarr(nlevels)  &  yy0=xx0  &  xx1=xx0  &  yy1=yy0
      for i = 0, nlevels-1 do begin
        xx0(i) = tnew( t(i),   p(i)   )
        yy0(i) = p(i)
        xx1(i) = tnew( td(i),  p(i)   )
        yy1(i) = p(i)
      endfor
      xbegin = fix( (min(xx1)-10.) / 10. ) * 10.
      xend   = fix( (max(xx0)+10.) / 10. ) * 10.

    return, [xbegin, xend]
    end

;============================================================ ============
;
;  PROCEDURE TO DRAW A SKEW-T, Log(P) DIAGRAM GIVEN A DESIRED
;  TEMPERATURE RANGE FOR THE DATA.
;
;  Originator:  Andrew F. Loughe
;


PRO SKEWT, t_range, everyT=everyT, everyDA=everyDA, $
         everySA=everySA, everyW=everyW, title=title, notitle=notitle
on_error, 2

COMMON RANGES, trange, prange

if (n_elements(everyT)  le 0) then everyT  = 10   ; T  = Temperature
if (n_elements(everyDA) le 0) then everyDA = 10   ; DA = Dry adiabat
if (n_elements(everySA) le 0) then everySA = 1    ; SA = Saturated adiabat
if (n_elements(everyW)  le 0) then everyW  = 1    ; W  = Mixing ratio

if (not keyword_set(title)) then title='Skew-T, Log(P) Diagram'
if (keyword_set(notitle))   then title=' '
if (n_elements(prange)) eq 0 then prange = [1050., 100]

if (N_params() eq 0) then $
  message,$
  'EXAMPLE:  skewt, [-20, 20], everyT=10, everyDA=10, everySA=2, everyW=2'
if (n_elements(t_range)) eq 1 then t_range=[-40., 40.]

;  Set some defaults
trange   = t_range
charsize = .8           ; Set default character size
!p.multi = [0,1,1]

;  Set default color positions
```

```
RED   = 44
GREEN = 22
BLUE  = 33
BLACK = 0
WHITE = 1


;  Make plot square for arbitrarily chosen trange of 80 degrees.
;  Code from Ken Bowman

if (!d.name eq 'PS') then device, /inch, xsize=7, ysize=7

daspect = FLOAT(!D.Y_SIZE)/FLOAT(!D.X_SIZE) * (trange(1)-trange(0))/80.
margin  = 0.1
aspect  = 1.0 ; A square
x0 = 0.50 - (0.5 - margin)*(daspect/aspect)
y0 = margin
x1 = 0.50 + (0.5 - margin)*(daspect/aspect)
y1 = 1.0 - margin

!P.POSITION = [x0, y0, x1, y1]    ; Set value of sytem variable.


;  Determine character height and width.  Apply charsize.
char_ht = convert_coord([0, !d.y_ch_size], /device, /to_norm)
char_ht = char_ht(1) * 1.0
if (!d.name ne 'X' and charsize gt 1.) then $
   char_ht = char_ht * charsize
char_wd = convert_coord([0, !d.x_ch_size], /device, /to_norm)
char_wd = char_wd(1)


;  Create the plot space.
plot_io, trange, prange, yrange=prange, /nodata, /xs, /ys, $
        xticklen=.01, ytickname=replicate(' ',30), charsize=charsize, $
        title=title

;  Print PRESSURE title along the y-axis.
lnt=alog(prange(1))  &  lnb=alog(prange(0))  &  avg=exp(.5*(lnt+lnb))
xy = convert_coord([trange(0), avg],/data,/to_norm)
xyouts, xy(0)-(5.*char_wd), xy(1), 'PRESSURE  (hPa)', orient=90, $
      /norm, align=.5

;  Print TEMPERATURE title along the x-axis.
xy = convert_coord([.5*(trange(0)+trange(1)), prange(0)], /data, /to_norm)
xyouts, xy(0), xy(1)-(3.*char_ht), 'TEMPERATURE (!uo!nC)', align=.5, /norm

;  Draw Pressure labels next to tick marks along the y-axis.
```

```
pressures = [1050,1000,900,800,700,600,500,400,300,200,100]
for i = 0, 10 do begin
   ytick = pressures(i)
   if (ytick ge prange(1)) then begin
      xy = convert_coord( [trange(0), ytick], /data, /to_norm )
      xyouts, xy(0)-(.2*char_wd), xy(1)-(.25*char_ht), $
         strcompress(string(ytick),/remove_all), align=1, $
         charsize=charsize, /norm

      plots, [trange(0), trange(1)], [ytick, ytick]  ; Horizontal line.
   endif
endfor

clip=[trange(0),prange(0),trange(1),prange(1)]   ; Define clipping space.

 ;============================================================ ============
;  Draw skewed isotherms every "everyT (10C)"  (Lines are straight).
for temp = trange(0)-100, trange(1)+5, everyT do begin
   x0 = temp
   y0 = prange(0)
   x1 = temp
   y1 = prange(1)

;  Draw the line.
   newx0 = tnew(x0, y0)  ; Find rotated temperature position
   newx1 = tnew(x1, y1)  ; Find rotated temperature position
   plots, [newx0, newx1], [y0, y1], color=BLUE, clip=clip, noclip=0

;  Draw line labels
;  Use method #1 in xy function to determine a place for the label.
      drew_label = 'no'
      xy = Told(temp, 1)
      if ( xy(0) gt trange(0) and xy(0) lt trange(1) and $
           xy(1) gt prange(1) and xy(1) lt prange(0) ) then begin
             drew_label = 'yes'
             xyouts, xy(0), xy(1), strcompress(string(fix(temp)), /rem),$
                    color=BLUE, orient=45, align=.5, charsize=charsize
      endif

;  Use method #2 in xy function to determine a place for the label.
      if (drew_label eq 'no') then xy = Told(temp, 2)
      if ( xy(0) gt trange(0) and xy(0) lt trange(1) and $
           xy(1) gt prange(1) and xy(1) lt prange(0) and $
           drew_label eq 'no') then begin
             xyouts, xy(0), xy(1), strcompress(string(fix(temp)), /rem),$
                    color=BLUE, orient=45, align=.5, charsize=charsize
      endif
```

```
endfor

 ;============================================================ ============
;  Draw dry adiabats every "everyDA (10C)"  (Lines are curved).
for temp = trange(0), trange(0)+220, everyDA do begin
   x1  = float(temp)
   y1  = 1050.
   inc = -2.     ; Lines will be curved, so use a small press. increment.
   drew_label='no'
   icount = 0

;  Dry adiabats from 1050mb up to prange(1).
;  For a given temperature and pressure, compute theta and plot a line.
for press = y1, prange(1), inc do begin
   icount = icount + 1
   x0 = float(x1)                              ; Orig Temp
   y0 = float(press + inc)                        ; Orig Press
   y1 = float(y0 + inc)                         ; New  Press
   x1 = (temp+273.16) * ( y1 / 1000. ) ^ (287./1004.)   ; New Temp
   x1 = x1 - 273.16

   newx0 = tnew(x0, y0)  ; Find rotated temperature position
   newx1 = tnew(x1, y1)  ; Find rotated temperature position


;  Draw the labels.
   if (fix(x1) eq fix(trange(0)) and drew_label eq 'no') then begin
      drew_label='yes'
      if ( newx1 gt trange(0) and newx1 lt trange(1) and $
           y1 gt prange(1) and y1 lt prange(0) ) then $
           xyouts,newx1,y1,strcompress(string(fix(temp)),/remove),$
             align=.5, color=RED, charsize=charsize, orientation=-45
   endif

;  Draw the line.
   if (icount gt 1) then $
   plots, [newx0, newx1], [y0, y1], color=RED, clip=clip, noclip=0
   if (newx1 lt trange(0)) then goto, jump2
endfor

jump2: dummy=0
endfor

 ;============================================================ ============
;  Draw saturated adiabats.  Begin at 40C and step backwards by 4C.
;  These lines are curved.
TS = 40.
FOR TS = 40, -64, -everySA*4 DO BEGIN
```

```
   P   = 1060.
   TK  = TS + 273.16
   AOS = OS(TK, 1000.)

   ATSA  = TSA(AOS, P) - 273.16
   FOR J = 0, 85 DO BEGIN
     P0 = P
     T0 = ATSA

     P = P - 10.
     ATSA = TSA(AOS, P) - 273.16
     if (j gt 0) then begin
       newx0=tnew(T0,P0)  ; Find rotated temperature position
       newx1=tnew(ATSA,P) ; Find rotated temperature position

;  Leave a space for the labels and draw them.
       if (P gt 730 or P lt 700) then $
         plots, [newx0, newx1], [P0, P], $
               color=GREEN, clip=clip, noclip=0

       if ( P eq 730 ) then begin
         if (newx1 gt trange(0) and newx1 lt trange(1)) then $
          xyouts,newx1,P,strcompress(string(fix(TS)),/remove),align=.5 ,$
               color=GREEN, charsize=charsize
       endif
     endif
   ENDFOR

ENDFOR

 ;============================================================ ============
;  Draw mixing ratio lines (Lines are straight).
;  Find temperature for a given Ws (g/kg) and Press (mb).

Ws=[ .1,.2,.4,.6,.8,1.,1.5,2.,2.5,4,5,6,7,8,9,10,12, $
     14,16,18,20,24,28,32,36,40,44,48,52,56,60,68,76,84  ]

for i = 0, N_elements(Ws)-1, everyW do begin
   press1 = prange(0)
   tmr1   = tmr(Ws(i), press1) - 273.16

   press2 = 200.
   tmr2   = tmr(Ws(i), press2) - 273.16

   newx0=tnew(tmr1,press1) ; Find rotated temperature position
   newx1=tnew(tmr2,press2) ; Find rotated temperature position

;  Draw the line.
```

```
   plots, [newx0, newx1], [press1, press2], color=22, linestyle=2, $
         clip=clip, noclip=0

;  Draw the line label.
   drew_label='no'
   if (newx0 gt trange(0) and newx0 lt trange(1)) then begin
      drew_label='yes'
      if (Ws(i) ge 1.0) then $
      xyouts, newx0, press1-2, strcompress(string(fix(Ws(i))),/remove),$
            align=.5,color=GREEN, charsize=charsize
      if (Ws(i) lt 1.0) then $
      xyouts, newx0, press1-2, string(Ws(i),format='(f3.1)'), align=.5,$
           color=GREEN, charsize=charsize
   endif
   if (newx1 gt trange(0) and newx1 lt trange(1)) then begin
      if (Ws(i) ge 1.0) then $
      xyouts, newx1, press2-2, strcompress(string(fix(Ws(i))),/remove),$
            align=.5, color=GREEN, charsize=charsize
      if (Ws(i) lt 1.0) then $
      xyouts, newx1, press2-2, string(Ws(i),format='(f3.1)'), align=.5,$
           color=GREEN, charsize=charsize
   endif

endfor

 ;============================================================ ============
; Redraw the plot boundary.
plots, [trange(0),trange(1),trange(1),trange(0),trange(0)], $
       [prange(0),prange(0),prange(1),prange(1),prange(0)], thick=2


!p.position = [.05, .05, .95, .95]      ; Reset position parameter

END
```


--
Andrew F. Loughe   email: afl@cdc.noaa.gov
University of Colorado, CIRES   voice: (303) 492-0707
Campus Box 449   fax: (303) 497-7013
Boulder, CO 80309-0449   USA