
Subject: Re: setting the aspect ratio

Posted by [thompson](#) on Mon, 28 Oct 1991 13:29:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <1991Oct28.035605.19990@murdoch.acc.Virginia.EDU>, gl8f@fermi.clas.Virginia.EDU (Greg Lindahl) writes...

> I have a contour plot where the X axis has 3 times the range of the Y
> axis. I want the contour plot to indicate this by making the plot 3
> times wide as it is tall.

>

> Is there any *easy* way to do this? Most graphics packages let you
> tell them if you want a 1:1 aspect ratio for the X and Y axes. But the
> best I can find in IDL is to use device-specific stuff in the device
> command or for X-windows, to resize the window by hand.

I think the following procedure should do what you want. It sets the IDL scaling parameters so that subsequent plot calls (PLOT, CONTOUR, etc.) are scaled correctly. The actual corners of the plot are not changed, though, so that a contour plot would not go out to the axes. I would think this could be changed too, though, by changing !SC1, !SC2, ... (or more properly these days !P.POSITION).

As an added refinement, I've recently added the system variable !ASPECT for graphics devices without square pixels. This system variable contains the ratio of the pixel height to pixel width, and is normally 1. It can be defined through the command

```
DEFSYSV, '!ASPECT', 1.0
```

or could also easily be removed from the routine below.

Bill Thompson

```
PRO SETSCALE,P1,P2,P3,P4,NOBORDER=NOBORDER
;+
; NAME:
; SETSCALE
; PURPOSE:
; Sets the plot scale such that it is the same in the X and Y directions.
; CATEGORY:
; CALLING SEQUENCE:
; SETSCALE - Resets to default.
; SETSCALE, ARRAY - Calculates scale for CONTOUR.
; SETSCALE, XARRAY, YARRAY - Calculates scale from arrays.
; SETSCALE, XMIN, XMAX, YMIN, YMAX - Calculates scale from limits.
; OPTIONAL INPUT PARAMETERS:
```

```

; ARRAY - Two dimensional array to be used in a simple
; contour plot. The minima are set to zero,
; and the maxima are set to one less than the
; dimensions of the array.
; XARRAY, YARRAY - Arrays from which the minimum and maximum
; values are calculated.
; XMIN, XMAX, YMIN, YMAX - The limits in the X and Y directions from
; which the scale is calculated. The actual
; X and Y ranges must include these values.
; OUTPUTS:
; None.
; OPTIONAL OUTPUT PARAMETERS:
; None.
; OPTIONAL KEYWORD PARAMETERS:
; NOBORDER = If set, then the 5% border is not applied.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; The system variables !X.STYLE, !Y.STYLE, !X.RANGE (!XMIN and !XMAX) and
; !Y.RANGE (!YMIN and !YMAX) are modified. Any previous settings are
; lost.
; RESTRICTIONS:
; If the graphics device or the size of the area used for plotting is
; changed, then this routine must be called again.
; PROCEDURE:
; The data limits in the X and Y directions (plus 5%) are calculated and
; compared against the the physical size of the plotting area in device
; coordinates. Whichever scale is larger is then used for both axes, and
; the plot limits are set to center the data in both directions. The
; parameters !X.STYLE and !Y.STYLE are then set to 1 for exact spacing.
; MODIFICATION HISTORY:
; William Thompson, Feb. 1991.
; William Thompson, Oct. 1991, added !ASPECT system variable.
;-
;
;
IF N_PARAMS() EQ 0 THEN BEGIN
  !X.STYLE = 0 & !X.RANGE = [0,0]
  !Y.STYLE = 0 & !Y.RANGE = [0,0]
  RETURN
END ELSE IF N_PARAMS() EQ 1 THEN BEGIN
  SZ = SIZE(P1)
  IF SZ(0) NE 2 THEN BEGIN
    PRINT,'*** ARRAY must be two dimensional, ' + $
    'routine SETSCALE.'
    RETURN
  ENDIF
  XMIN = 0 & XMAX = SZ(1) - 1
  YMIN = 0 & YMAX = SZ(2) - 1

```

```

END ELSE IF N_PARAMS() EQ 2 THEN BEGIN
  XMIN = MIN(P1, MAX=XMAX)
  YMIN = MIN(P2, MAX=YMAX)
END ELSE IF N_PARAMS() EQ 4 THEN BEGIN
  XMIN = P1 & XMAX = P2
  YMIN = P3 & YMAX = P4
END ELSE BEGIN
  PRINT, '*** SETSCALE must be called with 0-4 parameters:'
  PRINT, '    SETSCALE'
  PRINT, '    SETSCALE, ARRAY'
  PRINT, '    SETSCALE, XARRAY, YARRAY'
  PRINT, '    SETSCALE, XMIN, XMAX, YMIN, YMAX'
  RETURN
ENDELSE
;
; Check to see if the screen coordinates have been initialized.
;
IF !SC1 EQ !SC2 THEN BEGIN
  X = !X.MARGIN * !D.X_CH_SIZE
  !SC1 = X(0)
  !SC2 = !D.X_SIZE - X(1)
ENDIF
IF !SC3 EQ !SC4 THEN BEGIN
  Y = !Y.MARGIN * !D.Y_CH_SIZE
  !SC3 = Y(0)
  !SC4 = !D.Y_SIZE - Y(1)
ENDIF
;
; Calculate the plot scale.
;
XSCALE = ABS(XMAX - XMIN) / (!SC2 - !SC1) * !ASPECT
YSCALE = ABS(YMAX - YMIN) / (!SC4 - !SC3)
IF NOT KEYWORD_SET(NO BORDER) THEN BEGIN
  XSCALE = 1.05 * XSCALE
  YSCALE = 1.05 * YSCALE
ENDIF
SCALE = XSCALE > YSCALE
IF SCALE LE 0 THEN BEGIN
  PRINT, '*** Unable to calculate the plot scale, routine SETSCALE.'
  RETURN
ENDIF
;
; Calculate the edges of the plot.
;
XAVG = (XMIN+XMAX) / 2. & XDELTA = SCALE * (!SC2-!SC1) / (2*!ASPECT)
YAVG = (YMIN+YMAX) / 2. & YDELTA = SCALE * (!SC4-!SC3) / 2.
!X.RANGE = [XAVG - XDELTA, XAVG + XDELTA]
!Y.RANGE = [YAVG - YDELTA, YAVG + YDELTA]

```

```
;
;
; Check to see if the data should be plotted in reverse order.
;
;
IF XMIN GT XMAX THEN !X.RANGE = REVERSE(!X.RANGE)
IF YMIN GT YMAX THEN !Y.RANGE = REVERSE(!Y.RANGE)
;
;
; Set the style parameters to force the data to be plotted with the exact
; edges calculated.
;
;
!X.STYLE = 1
!Y.STYLE = 1
;
;
RETURN
END
```
