Subject: Re: PV-WAVE Filled Contours
Posted by sterner on Fri, 22 Jan 1993 23:00:09 GMT
View Forum Message <> Reply to Message

mayer@teal.csn.org (Mike Mayer) writes:

> Hello,

> Does anyone have a procedure written in PV-WAVE that will allow
> one to create filled contours correctly?

> Have seen a few here in comp.lang.idl-pvwave, but were either written
> for IDL (WAVE didn't have the routines it was calling) and one to
> replace POLYCONTOUR (looked the same when plotted).

> Trying to get around having to write out a polygon/path file or
> otherwise fiddle with the original array. In an ideal world, we'd
> have something like CONTOUR, array, /Fill   ...and bingo there it is.

> Thanks.

> Mike

> --
> ___---^^^---___---^^^---___---^^^---___---^^^--- Catch The WAVE ---___
> Michael Mayer, Senior Technical Support Engineer, Visual Numerics, Inc. (was
> Precision Visuals, Inc.) 32915 Aurora Rd. Suite # 160, Solon, OH, 44139 USA
> Email: mayer@pvi.com   Human: 216/248-4900   Fax: 216-248-2733

Here is a short function that does filled contours (one screen long
after dropping the help text and blank lines). It won't do all the
fancy stuff, but if you just want simple filled contours it might
work. It uses only the standard routines that come with IDL (and
probably PV-WAVE). To display the help text do something like:
  x=fill_cont(/help)

If z is a 2-d array try something like:
  tv,fill_cont(z),0
  tvscl,z,1
The first should have 8 colors, the second should be continuous.
tv,fill_cont(z,n_levels=16) will give 16 colors and so on.
You may also specify the actual levels to contour and the colors
to use. It has not been tested on many cases sinces its only a few
minutes old.

Ray Sterner   sterner%str.decnet@warper.jhuapl.edu
Johns Hopkins University   North latitude 39.16 degrees.
Applied Physics Laboratory   West longitude 76.90 degrees.

Laurel, MD 20723-6099

```
;---------  fill_cont.pro  = fill contours with colors  -----------
; R. Sterner, 22 Jan, 1993

 function fill_cont, z, levels=lv, colors=clr, n_levels=n_lv, help=hlp

 if (n_params(0) lt 1) or keyword_set(hlp) then begin
   print,' Returns a byte array with filled contours.'
   print,' fill_cont, z'
   print,'   z = 2-d array to contour.    in'
   print,' Keywords:'
   print,'   LEVELS=lv  Array of contour levels.  Default'
   print,'     is 8 contours from array min to max.  First value'
   print,'     in LEVELS should be the minimum of the first contour'
   print,'     range, and the last should be the maximum of the'
   print,'     last contour range.  For example, for 3 contour'
   print,'     ranges from 23 to 130 LEVEL should be:'
   print,'     23.0000      58.6667      94.3333      130.00'
   print,'   N_LEVELS=n Number of evenly spaced levels to contour'
   print,'     from array min to max.  Only if LEVELS not given.'
   print,'   COLORS=clr Array of contour colors.  Default'
   print,'     is enough colors to handle LEVELS'
   print,'     spaced from 0 to !d.ncolors-1.  Number of colors'
   print,'     is 1 less than the number in LEVELS.'
   return, -1
 endif

 ;--------  Set defaults  ---------
 mn = min(z,max=mx)   ; Find data min and max.
 d = mx-mn    ; Data range.
 if n_elements(n_lv) eq 0 then n_lv=8 ; Default number of levels.
 n_lv = n_lv>1    ; Must be at least 1.
 ;---  If LEVELS array not given make it.  -----
 if n_elements(lv) eq 0 then lv = findgen(n_lv+1)*d/n_lv+mn
 nlv = n_elements(lv)   ; How many levels?
 ;---  If COLORS array not given make it.  -----
 if n_elements(clr) eq 0 then $
   clr = (findgen(nlv)*(!d.n_colors-1)/(nlv-1))(1:*)
 lstc = n_elements(clr)-1  ; Last color index.

 ;-------  Set up output array  ----------
 out = byte(z*0)

 ;---  Loop through levels filling contours.  -------
 for i = 0, nlv-2 do begin
   w = where((z ge lv(i)) and (z le lv(i+1)), cnt)
   if cnt gt 0 then out(w) = clr(i<lstc)
```

```
endfor

return, out
end
```