Subject: Re: polar interpolation
Posted by Thomas Gutzler on Mon, 20 Jan 2003 03:33:54 GMT
View Forum Message <> Reply to Message

Hi,

Rick Towler wrote:
>
> Interpolate between points a and b:
>
> iFactor = (Ti - Ta) / (Tb - Ta)
> Ri = Ra + (iFactor * (Rb - Ra))
> Where Ti is the Theta value where you are interpolating your radius Ri.

I stole this from interpol.pro. The function is the same, just other
names for variables:

```
;    ,------------------------------------------------------------ -----
;   | Purpose:
;   | linear, polar interpolation for irregular grids
;   | T: The theta values (in degrees) for R (radius). This vector
;   | must have same # of elements as R. The values MUST be
;   | "monotonically ascending" or "descending" (see problem below).
;   | U: The theta values for the result. The result will have
;   | the same number of elements as U. U does not need to be
;   | monotonic. If U is outside the range of T, then the
;   | closest two endpoints of (T,R) are linearly extrapolated.
;    `------------------------------------------------------------ ------
FUNCTION polar_interpol, R, T, U
  m = N_elements(R)                  ; # of input pnts
  s = VALUE_LOCATE(T, U) > 0L < (m-2)  ; Subscript intervals.
  p = ( U - T[s] ) * ( R[s+1] - R[s] ) / ( T[s+1] - T[s] ) + R[s]
  RETURN, p
END
```

But what to do, if an array like this wants to be interpolated:
R: [1,2,3]
T: [340,350,20]
U: [355,5,15,25]

I can see 2 solutions at the moment:

1) - find the first index (i) after crossing the magic 0-line.
     In this case: i_T = 2 and i_U = 1
   - T[i_T:*] = T[i_T:*] + 360
   - U[i_U:*] = U[i_U:*] + 360
   - interpolate T: [340,350,380], U: [355,365,375,385]
   - RESULT = [2.16667, 2.50000, 2.83333, 3.16667]

2) - split the vectors
R1: [1,2,2.33333] (last value must be interpolated)
T1: [340,350,360]
U1: [355]
R2: [2.33333,3]
T2: [0,20]
U2: [5,15,25]
- interpolate separately
RESULT1: [2.16667]
RESULT2: [2.50000, 2.83333, 3.16667]
- return [RESULT1, RESULT2]

Is there a trick to do this easier, using some strange functions
(VALUE_LOCATE was strange for me) ?
Keep in mind that the next step would be a quadratic interpolation.

thanks, Tom
