
Subject: interpolation question

Posted by [chen123.dian](#) on Thu, 20 Apr 2006 04:25:15 GMT

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

Hi, I want to use nearest-neighbor method to interpolate 2-D data. After reading some posters here and idl references, I am still confused how to do. I use a 1-D data to explain my problem. For example, there is x-array, [1.0, 2.9, 5.1], the corresponding y-array is [10, 50, 80]. Now we want to get the y-values at x-value [0.5, 2.6, 3.1, 4.6] using nearest-neighbor method. The result of y-values should be [10, 50, 50, 80]. Can the IDL function of interpolate.pro do this work? The input of interpolate.pro, X,Y,Z, arrays of numeric type containing the locations for which interpolated are desired, is very confuse. Is there any function like the function interp1 (interp2 for 2-D data) for matlab? Like, y_values=interp1(x, y, x-values, 'nearest', 'extrap').

Thanks in advance.

Chen

Subject: Re: interpolation question

Posted by [JD Smith](#) on Fri, 21 Apr 2006 01:57:52 GMT

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

On Wed, 19 Apr 2006 21:25:15 -0700, chen123.dian wrote:

> Hi, I want to use nearest-neighbor method to interpolate 2-D data.
> After reading some posters here and idl references, I am still confused
> how to do. I use a 1-D data to explain my problem. For example, there
> is x-array, [1.0, 2.9, 5.1], the corresponding y-array is [10, 50,
> 80]. Now we want to get the y-values at x-value [0.5, 2.6, 3.1, 4.6]
> using nearest-neighbor method. The result of y-values should be [10,
> 50, 50, 80]. Can the IDL function of interpolate.pro do this work? The
> input of interpolate.pro, X,Y,Z, arrays of numeric type containing the
> locations for which interpolated are desired, is very confuse. Is there
> any function like the function interp1 (interp2 for 2-D data) for
> matlab? Like, y_values=interp1(x, y, x-values, 'nearest', 'extrap').

Doesn't it depend on how your 2D array is sorted? Are the values along rows and columns always monotonically increasing, e.g.:

```
0.1 1.2 2.4
0.2 1.3 2.5
.....
```

Otherwise, the problem is undefined.

That said, I think you are not asking a value lookup problem at all, but a simple 2D interpolation on a regular grid, where instead of linear nearest neighbor interpolation, you want "plain" nearest neighbor. `VALUE Locate` can do this for you, assuming "nearest in x" and "nearest in y" are separable.. Let 'yarray' be your data array you'd like to interpolate on, and 'curve_x' be the monotonic, but non-regular grid of x positions, and 'curve_y' similarly, e.g.:

```
IDL> yarray=randomu(sd,100,100)
IDL> curvex=findgen(100)^2
IDL> curvey=alog10(1.+findgen(100))
```

such that in the 2D 'yarray', all rows have the same X vector coordinates 'curvex', and all the columns the same Y vector 'curvey'. Now you have two vectors describing the X,Y positions in this funny space where you would like nearest neighbor interpolates:.

```
IDL> x=(randomu(sd,10)*100)^2
IDL> print,x
    6544.61    2552.47    2239.70    9695.07    144.029    919.768
    933.261    3231.17    151.407    761.378
IDL> y=alog10(1.+randomu(sd,10)*100)
IDL> print,y
    1.63828    1.52555    1.80391    0.983095    1.26320    1.86245
    1.79494    1.67543    1.89285    1.37616
IDL> xpos=value_locate(curvex,x)
IDL> xpos=round(xpos+(x-curvex[xpos])/(curvex[xpos+1]-curvex[xpos ]))
IDL> ypos=value_locate(curvey,y)
IDL> ypos=round(ypos+(y-curvey[ypos])/(curvey[ypos+1]-curvey[ypos ]))
IDL> interp=yarray[xpos,ypos]
```

JD
