Subject: surface fitting...
Posted by rmb on Wed, 28 Jan 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Hi,

I want to fit a polynomial to a surface and have found the function sfit. However, this routine assumes that the coordinate system is defined by the array and I cannot see a way to change this. I have some data that which has its origin at the centre of the array and is most likely radially symmetric.

Is there a similar routine or a work-round in which I can specify the (x,y) coordinates of each data value?

Thanks,
--
R.

-= -=-=-=-=-=-=-
Richard Brockie B.Sc.(Hons), The tall blond one.
Adaptive Optics Chap. Email: R.Brockie@roe.ac.uk
-=-=-=-=-=-=-=-=-=-=-=-= http://www.roe.ac.uk/rmbwww =-=-=-=-=-=-=-=-=-=-=-=-

---

Subject: Re: surface fittin
Posted by mchinand on Fri, 04 Mar 2005 23:12:04 GMT
View Forum Message <> Reply to Message

In article <d09orm$u9k$1@news.iucc.ac.il>, Tal <ftal@post.tau.ac.il> wrote:
> hello,
>
> i worked with SFIT for that and used a 6th order polynomial which does a
> good job (when looking at the output image it creates). however, when trying
> to reconstruct that image that i see, by using the required x, y and the
> coefficients, the resulting surface is far from being similar to what the
> surface really is. i managed to fully and exactly reconstruct a 3x3
> Gaussian. so i know that i read the KX coefficients correctly. i also
> defined my x and y variables correctly (100% sure). but is doesn't work for
> my 72x15 pixels surface. KX is a floating point output. it could be that
> when saved as double precision, results will be better. but there is no
> keyword for that.

It seems to work for the example in the online help:

```
=============================================
X = (FINDGEN(61)/10) # REPLICATE(1,61)
Y = TRANSPOSE(X)
```

```
; Evaluate a function at each point:
F = -SIN(2*X) + COS(Y/2)
; Compute a sixth-degree polynomial fit to the function data:
result = SFIT(F, 6, kx=k)
================================================
```

Now to get the fit from coefficients, first multiply the x and y arrays by 10 since the fit was done with array indices and not the x and y values used to create F.

```
x=x*10.
y=y*10.
```

Create an array for the fit and then loop through the coefficients

```
fit=fltarr(61,61)
for i=0,6 do  for j=0,6 do  fit=fit+k(j,i)*x^i*y^j
```

Hope this helps,

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago