Subject: Re: Yet another user with poly_fit problems
Posted by Heinz Stege on Mon, 30 Sep 2013 22:46:58 GMT
View Forum Message <> Reply to Message

On Mon, 30 Sep 2013 12:59:16 -0700 (PDT), Gus wrote:

> Hello everyone,
>
> I've read a few of the older posts on this topic, but their solution didn't really help me solve the problem that I am currently having with the poly_fit function. The set of coefficients generated by the function (a 4th degree polynomial) produces some rather absurd results. Here is a short version of the problem I am having.
>
> X = [0.000000, 11.6667, 822.914, 3458.85, 27703.4, 133928.]
> Y = [15.9000, 16.0000, 17.0000, 18.0000, 19.0000, 20.0000]
>
> C = poly_fit(X, Y, /double, yfit=D)
>
> IDL generates the following coefficients (for C)
>
> 15.940691
> 0.0015355228
> -3.0965110e-007
> 1.1170193e-011
> -6.6767399e-017
>
> Yet, one will clearly see that this fit produces rather undesirable results since, within the same range of X values (0 to roughly 150,000), this fit will produce Y values that can be as high as 1600 and as low as -3000 (rather than between 15.9 and 20). Excel is generating better coefficients than IDL!
>
> Here is what I have already tried to do (and did not solve the problem)
>
> 1) Double precision of X and Y prior to using the poly_fit function (notice that I am using the "/double" keyword function already in that function);
>
> 2) Subtracting the mean of X from that array, before fitting the data - suggested in previous posts;
>
> 3) Subtracting the value of X[0] from that array, before fitting the data;
>
> 4) Subtracting the mean of Y from that array, before fitting the data.
>
> Does anyone know of any other solution to this problem?

As far as I can see, it is not possible to get better results fitting
the given data with a polynomal. The curve you need to fit the data
has to be very steep near x=0 and very flat for x > 20000. I don't see

how to manage this by a polynomal. Please show us the coefficients
from Excel. I'm really interested to see them.

A solution for you may be a function like
  y = (p0 + p1*x + p2*x^2) / (x - x0)
I get the coefficients
  x0=-1543.59
  [p0,p1,p2]=[24607.0, 18.9245, 8.39527e-006]
with a self-written fit routine for non-linear functions. You can try
curvefit (built-in) or mpcurvefit (from Craig Markwardt).

HTH, Heinz
