

---

Subject: Re: Yet another user with poly\_fit problems  
Posted by [Gus](#) on Mon, 30 Sep 2013 23:15:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Problem has been solved by suicida's reply (THANK YOU VERY MUCH).

Firstly, I forgot to add then "4" when I typed up the original post, but needless to say, I was using that correctly in IDL. I also had the coefficient in the right order (the last one is this case is coefficient for the "X^4"). Thanks David.

Anyhow, as I typed the original message here I had begun to wonder whether my error was indeed caused by the fact I had so few points in my array to execute a 4th order polynomial fit. Before I had come to a solution on how to test for that, I got the answer here. Increasing my data size through linear interpolation, seems to do the trick.

Heinz - Suicida's post solved the problem I have. I had thought about using a different method for fitting, but given that my problem is just a tiny snip of my code, which is used for a variety of different data points, changing the fitting methodology would create some comparative problems in its application. So, I need to stick to a single fitting method. Also, I totally misspoke with the better results for Excel (forgive my blasphemous accusation). I had, inadvertently, used a different dataset for the fitting in Excel.

Thank you everyone for the quick contribution!

On Monday, September 30, 2013 4:59:16 PM UTC-3, 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?

---