
Subject: Re: POLY_FIT gives wrong answer !
Posted by [davidf](#) on Thu, 11 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

The Powers That Be at Goddard Space Flight Center prefer their employees have no opinions about anything. Hence, they are refusing to publish any news articles sent from the place. (I'm ready to file a Freedom of Information Act request to get Stein Vidar back on-line.) Bill Thompson smuggled this article out in his lunch box and asked if I would post it for him. Guess we will have to re-open discussion. :-(

Newsgroups: comp.lang.idl-pvwave
Subject: Re: POLY_FIT gives wrong answer !
References: <3917F8C2.A6DB4535@oma.be>

Henk Schets <Henk.Schets@oma.be> writes:

> Hi,

> Maybe all of you already know this, but I didn't and I could not find
> any documentation about it.

> The problem : I have two long arrays (x- and y-values) and I made
> subarrays out of them by using an array as subscript, like

> x = fltarray(l)
> y = fltarray(l)
> indexarr = longarray(l)
> (you know, the way to sort different arrays together)
> However, when I do something like out =
> poly_fit(x[indexarr],y[indexarr]), the outcome is simply wrong ! I know
> this because other programs gave another solution (all the same).
> The only way to do it right is by making other arrays like x2 and y2 and
> doing a poly_fit on it.

> Is this a known issue ?

At first I thought that David was right, and that there was a subtle error in the calling program. However, some experimentation leads me to believe that the answer has to do with round-off error.

To test what was going on, I generated a random series of X points between 0 and 1, and a corresponding Y array based on a polynomial, plus some noise.

```
IDL> x = randomu(seed,1000)
IDL> y = 3+5*x-7*x^2+randomn(seed,1000)
```

Applying POLY_FIT to these data gave reasonable results.

```
IDL> param1 = poly_fit(x,y,2,yfit1)
```

I then sorted the input arrays into increasing X, and reapplied POLY_FIT, expecting to get the same answer.

```
IDL> s=sort(x)
IDL> param2 = poly_fit(x(s),y(s),2,yfit2)
```

To my surprise, the parameters from the two runs of POLY_FIT are very close, but not exactly the same, even though the input arrays are exactly the same, just ordered differently. My only conclusion is that the difference must be due to round-off error, which affects the data differently depending on the order.

Perhaps your case is a more extreme example of round-off error affecting the calibration. I've definitely had problems sometimes with completely bogus results from POLY_FIT caused by round-off error. In my case this has always occurred when the range in X from minimum to maximum is small compared to the values of X. In such cases, I've always found it useful to redefine the problem to fit a polynomial to $X - X_0$ instead of X, where X_0 is comparable to the average value of X. For example, if my values of X ranged from 13479.25 to 13480.16, subtracting a typical value of $X_0 = 13480$ gives me a new X' which ranges from -0.75 to +0.16. This reduces the effect of round-off errors dramatically.

William Thompson
