Subject: Re: Speaking of curve fitting...
Posted by Paul Van Delst[1] on Thu, 31 Jan 2008 16:17:28 GMT
View Forum Message <> Reply to Message

Lasse Clausen wrote:
> ... run the following code, spot the difference and explain, s'il vous
> plait.
>
> nn = 1000
> xx1 = dindgen(nn)
> xx2 = timegen(nn, start=julday(5,25,1980,11,23))
>
> yy1 = sin(2.*2.*!pi*xx1/(nn-1.))
>
> d = poly_fit(xx1, yy1, 6, yfit=yfit1, /double)
> d = poly_fit(xx2, yy1, 6, yfit=yfit2, /double)

Try
  d = poly_fit(xx2-xx2[0], yy1, 6, yfit=yfit2, /double)

>
> !p.multi = [0,1,2]
> plot, xx1, yy1, /xstyle
> oplot, xx1, yfit1, linestyle=1
> plot, xx2, yy1,/xstyle
> oplot, xx2, yfit2, linestyle=1
>
> end
>
> I had a quick look at POLY_FIT.PRO but I can spot nothing which could
> explain the above behaviour. I run 32bit IDL 6.4 on some Linux.
>
> Cheers
> Lasse Clausen

Subject: Re: Speaking of curve fitting...
Posted by lasse on Thu, 31 Jan 2008 16:33:07 GMT
View Forum Message <> Reply to Message

On 31 Jan, 17:17, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> Lasse Clausen wrote:
>> ... run the following code, spot the difference and explain, s'il vous
>> plait.
>
>> nn = 1000
>> xx1 = dindgen(nn)
>> xx2 = timegen(nn, start=julday(5,25,1980,11,23))

>
>> yy1 = sin(2.*2.*!pi*xx1/(nn-1.))
>
>> d = poly_fit(xx1, yy1, 6, yfit=yfit1, /double)
>> d = poly_fit(xx2, yy1, 6, yfit=yfit2, /double)
>
> Try
> d = poly_fit(xx2-xx2[0], yy1, 6, yfit=yfit2, /double)
>
>
>
>> !p.multi = [0,1,2]
>> plot, xx1, yy1, /xstyle
>> oplot, xx1, yfit1, linestyle=1
>> plot, xx2, yy1,/xstyle
>> oplot, xx2, yfit2, linestyle=1
>
>> end
>
>> I had a quick look at POLY_FIT.PRO but I can spot nothing which could
>> explain the above behaviour. I run 32bit IDL 6.4 on some Linux.
>
>> Cheers
>> Lasse Clausen

Yes, that is indeed a workaround. But isn't that still a bug in
POLY_FIT? Surely the result of the fitting must not depend on an
arbitrary offset of the independent variable.

Cheers
Lasse Clausen

Subject: Re: Speaking of curve fitting...
Posted by pgrigis on Thu, 31 Jan 2008 16:34:17 GMT
View Forum Message <> Reply to Message

I guess this may have something to do with the finite precision
of your computer, which affects differently a polynomial
of 6th order evaluated at x=1000 than a polynomial of 6th
order evaluated at x=2E6 (as the ratio between the term
of different order in x is different in the two cases).

Whatever algorithm poly_fit use is likely to run into a problem
in the latter case, where the lower order terms are so small in
comparison with the high order terms. As the fit critically
depend in balancing out all the coefficients, this may well
lead to failure. It is not a good idea to use high order polynomial

on large numbers anyway...

Ciao,
Paolo

Lasse Clausen wrote:
> ... run the following code, spot the difference and explain, s'il vous
> plait.
>
> nn = 1000
> xx1 = dindgen(nn)
> xx2 = timegen(nn, start=julday(5,25,1980,11,23))
>
> yy1 = sin(2.*2.*!pi*xx1/(nn-1.))
>
> d = poly_fit(xx1, yy1, 6, yfit=yfit1, /double)
> d = poly_fit(xx2, yy1, 6, yfit=yfit2, /double)
>
> !p.multi = [0,1,2]
> plot, xx1, yy1, /xstyle
> oplot, xx1, yfit1, linestyle=1
> plot, xx2, yy1,/xstyle
> oplot, xx2, yfit2, linestyle=1
>
> end
>
> I had a quick look at POLY_FIT.PRO but I can spot nothing which could
> explain the above behaviour. I run 32bit IDL 6.4 on some Linux.
>
> Cheers
> Lasse Clausen

Subject: Re: Speaking of curve fitting...
Posted by Vince Hradil on Thu, 31 Jan 2008 16:42:43 GMT
View Forum Message <> Reply to Message

On Jan 31, 10:17 am, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> Lasse Clausen wrote:
>> ... run the following code, spot the difference and explain, s'il vous
>> plait.
>
>> nn = 1000
>> xx1 = dindgen(nn)
>> xx2 = timegen(nn, start=julday(5,25,1980,11,23))
>
>> yy1 = sin(2.*2.*!pi*xx1/(nn-1.))
>

>> d = poly_fit(xx1, yy1, 6, yfit=yfit1, /double)
>> d = poly_fit(xx2, yy1, 6, yfit=yfit2, /double)
>
> Try
> d = poly_fit(xx2-xx2[0], yy1, 6, yfit=yfit2, /double)
>
>> !p.multi = [0,1,2]
>> plot, xx1, yy1, /xstyle
>> oplot, xx1, yfit1, linestyle=1
>> plot, xx2, yy1,/xstyle
>> oplot, xx2, yfit2, linestyle=1
>
>> end
>
>> I had a quick look at POLY_FIT.PRO but I can spot nothing which could
>> explain the above behaviour. I run 32bit IDL 6.4 on some Linux.
>
>> Cheers
>> Lasse Clausen

Sure that works, but the underlying issue is still there - why should
it matter?
My guess: propagation of (roundoff) errors when poly_fit.pro
calculates the b-matrix.

Subject: Re: Speaking of curve fitting...
Posted by Paul Van Delst[1] on Thu, 31 Jan 2008 17:04:53 GMT
View Forum Message <> Reply to Message

Vince Hradil wrote:
> On Jan 31, 10:17 am, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
>> Lasse Clausen wrote:
>>> ... run the following code, spot the difference and explain, s'il vous
>>> plait.
>>> nn = 1000
>>> xx1 = dindgen(nn)
>>> xx2 = timegen(nn, start=julday(5,25,1980,11,23))
>>> yy1 = sin(2.*2.*!pi*xx1/(nn-1.))
>>> d = poly_fit(xx1, yy1, 6, yfit=yfit1, /double)
>>> d = poly_fit(xx2, yy1, 6, yfit=yfit2, /double)
>> Try
>> d = poly_fit(xx2-xx2[0], yy1, 6, yfit=yfit2, /double)
>>
>>> !p.multi = [0,1,2]
>>> plot, xx1, yy1, /xstyle
>>> oplot, xx1, yfit1, linestyle=1
>>> plot, xx2, yy1,/xstyle

>>> oplot, xx2, yfit2, linestyle=1
>>> end
>>> I had a quick look at POLY_FIT.PRO but I can spot nothing which could
>>> explain the above behaviour. I run 32bit IDL 6.4 on some Linux.
>>> Cheers
>>> Lasse Clausen
>
> Sure that works, but the underlying issue is still there - why should
> it matter?

It shouldn't. But it does because...

> My guess: propagation of (roundoff) errors when poly_fit.pro
> calculates the b-matrix.

I agree that it is likely a precision/roundoff issue... the exact mechanism may be
different, but I reckon you're right. An easy test would be to see what happens using SVDFIT.

cheers,

paulv

Subject: Re: Speaking of curve fitting...
Posted by David Fanning on Thu, 31 Jan 2008 17:13:56 GMT
View Forum Message <> Reply to Message

Folks,

Speaking of curve fitting...

Does anyone have any examples/tutorials/etc that they are
particularly fond of? I'm putting together a lecture. I know,
of course, about the tutorials on Craig's page. Some examples
of how you have approached a problem, with some data would
be *most* appreciated. Astronomy examples are particularly
desirable.

Thanks,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
