Subject: Re: Non-monotonic Abscissa values for IDL function SPLINE_P
Posted by Jeremy Bailin on Wed, 03 Jun 2009 18:31:01 GMT
View Forum Message <> Reply to Message

On Jun 3, 11:50 am, Xavier Ceamanos García <xavier.ceama...@gmail.com>
wrote:
> Thank you so much Jeremy for the answer!
>
> You are very right. I use splines to interpolate spectra. Then, the
> main goal is to re-sample these spectra. Each value of a spectrum
> corresponds to a given wavelength and in my case all points are
> separated by a constant wavelength distance. Hence, it is crucial to
> know which wavelengths corresponds to each point of the over-sampled
> spectra. A good re-sampling is not possible otherwise.
>
> Then, are you telling me that the number of points of the output
> vector depends on the input data? That would mean that it would be
> different for each spectrum interpolation...
>
> So far, I am using the "SPLINE" function which produces a monotonic
> output. In this case, it is easy to know the output wavelengths. It is
> slower though...
>
> I was just wandering if there is any way to get the same results I get
> with SPLINE but using SPLINE_P...
>
> Thanks again!
>
> Xavi

Xavi,

Yes, the number of points in the output vector depends on the input
data. For example:

```
IDL> x = findgen(20)
IDL> y = sin(0.5*x)
IDL> spline_p, x, y, xr, yr, interval=0.25
IDL> help, xr, yr
XR              FLOAT     = Array[96]
YR              FLOAT     = Array[96]
IDL> y = sin(0.5*x)+x
IDL> spline_p, x, y, xr, yr, interval=0.25
IDL> help, xr, yr
XR              FLOAT     = Array[120]
YR              FLOAT     = Array[120]
```

So it will indeed be different for each spectrum. But your code should

be general enough to deal with that using n_elements(xr).

You might want to directly use SPL_INIT and SPL_INTERP... I suspect that may do what you want.

-Jeremy.
