## Subject: very fast spline interp function for heavy oversampling?
Posted by R.G. Stockwell on Tue, 20 Jun 2000 07:00:00 GMT

Greetings,

I have a situation where I take a time series, and need to interpolate the
function to many more samples.
i.e. original time index = [1,2,3,4....10]
and I need samples at new time index =
[1.000,1.001,1.002.......9.999,10.000].

The canned IDL routine spline() works great, but is slow. Unfortunately, I
don't have time to rewrite the interpolation to something more efficient.

I don't want to use any linear scheme to interpolate, since I want a smooth
function (i.e. smooth "derivatives") around the data points.  I would guess
that
it would be easy to efficiently calculate this interp with spline, perhaps
some vectorization could be put into the function.  Or perhaps an
 "upsample" function would work, but modifications would be needed as the
time series is not evenly sampled.


Are there any user routines out there that can do this interpolation
efficiently?

Cheers,
bob

## Subject: Re: very fast spline interp function for heavy oversampling?
Posted by Paul van Delst on Thu, 22 Jun 2000 07:00:00 GMT

Peter Suetterlin wrote:
>
> In article <7nS35.70$482.44454@den-news1.rmi.net>,
>       "R.G. Stockwell" <stockwell@co-ra.comremove> writes:
>> Greetings,
>>
>> I have a situation where I take a time series, and need to interpolate the
>> function to many more samples.
>> i.e. original time index = [1,2,3,4....10]
>> and I need samples at new time index =
>> [1.000,1.001,1.002.......9.999,10.000].
>
> You could try fourier interpolation by zero padding, i.e. calculating

> the FFT and insert zeroes (remember the ordering inthe fft!) for the
> high frequency terms, then transform it back.

I think this is a much better suggestion than using spline interpolation
- unless the data you're fitting is very smooth - particularly when
you're dealing with time series/sampling issues. Unfortunately,
depending on how many interpolates you want (i.e. the number of points
can't be factored into primes easily) it can be slow. You also have to
be careful how you perform the interpolation and be cognisant of the
artifacts that can be introduced by fourier interpolation (ringing, or
Gibb's phenomena, for example).

I may be overly cautious, but I *never* use spline interpolation for
interpreting data (but I don't know what you're doing with your
re-interpolated data). If you're at all interested, I have a routine
that performs fourier interpolation (fourier_interpolate.pro) on high
resolution IR spectra - you can find it (and the other routines it
requires) at:

http://airs2.ssec.wisc.edu/~paulv/#idl

Follow the "Spectral" link. Note that a cosine filter is applied to the
edges of the input to minimise ringing.

paulv

p.s. what's the go with all the "remove this text from email address for
reply" that people do nowadays? Are they getting that badly spammed?
--
Paul van Delst        Ph:  (301) 763-8000 x7274
CIMSS @ NOAA/NCEP       Fax: (301) 763-8545
Rm.202, 5200 Auth Rd.   Email: pvandelst@ncep.noaa.gov
Camp Springs MD 20746