Subject: Re: Non-uniform FFT?
Posted by Eric Hudson on Wed, 06 Apr 2011 15:32:11 GMT
View Forum Message <> Reply to Message

On Apr 5, 1:54 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < 9cb6bd9c-a2b4-487a-b039-a9e636ba5...@c26g2000vbq.googlegroup s.com >,
> Eric Hudson <ehud...@mit.edu> wrote:
>
>> Hi,
>
>> I was wondering if anyone has implemented a non-uniform FFT algorithm
>> in IDL. We have non-regularly spaced real space data that we need to
>> Fourier transform, and it is painfully slow to do the discrete
>> transform. I have found several c algorithms online (e.g.
>> http://www-user.tu-chemnitz.de/~potts/nfft/download.php) but before
>> launching into either converting them or figuring out how to run C
>> code from within IDL thought maybe someone else had already gone to
>> the trouble.
>
>> Thanks,
>> Eric
>
> The approach could depend on just how non-uniform your data are.
>
> Do you need the whole spectrum, or do you know in advance
> which wavenumbers are of interest?
>
> You can do the DFT using least squares (regression), but that will
> be slow if you need the full spectrum.
>
> If you only need low wavenumbers, you could interpolate to
> a regular grid and then use least squares or the FFT.
>
> Ken Bowman

Hi Ken,

Thanks for the response. Unfortunately I need the whole spectrum (I have 2D data, slightly irregularly gridded, and want the equivalent of what you'd see if you did a 2D FFT on regularly gridded data). I had thought of doing interpolation and then the standard FFT, which I guess is to an extent what they are doing in these NFFT algorithms, but it seems they are a little more clever than that, which is why I was hoping someone had coded the NFFT routine in IDL. For now I am just directly integrating A(r) exp(i*q*r) over the whole image for each q, which is painfully slow because I have to loop on q (I don't have enough memory to make the whole q*r array in one go).

Eric