Subject: Re: IDL FFT (spec -> interferogram)
Posted by Randall Skelton on Fri, 05 Apr 2002 09:17:04 GMT
View Forum Message <> Reply to Message

On Thu, 4 Apr 2002, Paul van Delst wrote:

> Randall Skelton wrote:
>>
>> Hi all,
>>
>> Having read through all of the FFT posts that google groups keeps, I am no
>> closer to understanding why I am unable to transform a spectrum into an
>> interferogram using IDL. All of the data files, procedures, and pictures
>> of this are at http://tulip.atm.ox.ac.uk/~rhskelto/fft-help/
>
> all your spec data is zero.

Not quite. In the real domain there is a single sharp peak while in the imaginary part there is a sharp dispersion curve-- both are centered about the index 256 in the 512 point case (or 1024 in the 2048 point case). Experimentally, this can be thought of as a the result of inputting a mode stabilized laser into an interferometer.

>
>> The result 'idl_igm' contains twice the number of points (minus 2)
>> because of the required reflection about the Nyquist frequency.
>> Moreover, the result appears to be modulated (almost like a frequency
>> chirp)?
>
> This looks like the correct result to me - i.e. an interferogram of a
> spectrum. Do you want the envelope of this?

I'm not entirely sure what you mean by the "envelope of this?" I am trying to examine the center-burst of the resulting interferogram as you did with the AIRS data in your FFT comparison. I would expect a single line spectrum to give rise to a center-burst interferogram (am I off base here?).

The procedure shown for AIRS (and in fft_to_interferogram.pro) does not seem to work for me:

```
IDL> n = 2048                  ; could have used the 512 case
IDL> spec = dcomplexarr(n)
IDL> read_cmplx, 'spec2048.dat', spec

IDL> real_data = double(spec)
IDL> imag_data = imaginary(spec)
IDL> real_part = [ real_data, REVERSE( real_data[ 1 : n - 2 ] ) ]
```

```
IDL> imag_part = [ imag_data, -1.0 * REVERSE( imag_data[ 1 : n - 2 ] ) ]
IDL> cxs = COMPLEX( real_part, imag_part )
IDL> ifg = FFT( cxs, /INVERSE )
IDL> ifg = shift( ifg, -1 * ( n - 1 ) )
IDL> plot, ifg
```

In this case, 'ifg' does not look correct. I have been out of the lab and sitting in front of a computer for over a year now but this isn't what I remember seeing on the scope when I did these sorts of things.

The following example, however, does give what I would expect. Note that instead folding about the Nyquist frequency with the imaginary part rotated, I simply change the sign of all the even indexed points (i.e. reflecting even points about the x-axis).

```
IDL> for i=0, n-1, 2 do spec[i] = -spec[i]
IDL> ifg2 = fft(spec)
IDL> for i=0, n-1, 2 do ifg2[i] = -ifg2[i]
IDL> plot, ifg2
```

To be perfectly honest, I'm not exactly sure why this works (yet). If anyone has any insight, I'd love to hear it! Otherwise, I'm off to the engineering library...

Cheers,
Randall

---