Subject: Re: IDL FFT (spec -> interferogram)
Posted by Randall Skelton on Thu, 04 Apr 2002 16:32:10 GMT
View Forum Message <> Reply to Message

It is always the case... I struggled for hours with the IDL FFT routine and the numerical recipes cook book and now that I have written the 'please help' request to the newsgroup, I think I've found it.

Read in the data:

```
IDL> spec = dcomplexarr(2048)
IDL> read_cmplx, 'spec2048.dat', spec
IDL> igm = dcomplexarr(2048)
IDL> read_cmplx, 'igm2048.dat', igm
```

Fourier Transform and unwrap the phase

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

(NB: the for loops can be replaced by a where statement and index that gabs an even number sequence but the above is a little more clear)

There is no need to fold the spectrum about the Nyquist frequency and therefore the above does not require a division by 2 amplitude correction.

Cheers,
Randall

On Thu, 4 Apr 2002, 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/
>
> Given two files:
>
> 1) 'spec.dat' contains 512 points of complex spectral data
>
> 2) 'igm.dat' contains 512 points of complex interferogram data that was
> derived from 'spec.dat' using a prime factor FFT written in C.  This is
> the correct interferogram as far as I am concerned.  The plot
>

> Read in the data:
>
> IDL> spec = dcomplexarr(512)
> IDL> read_cmplx, 'spec.dat', spec
> IDL> igm = dcomplexarr(512)
> IDL> read_cmplx, 'igm.dat', igm
>
> Plot the expected result:
>
> IDL> plot, igm
> IDL> write_jpeg,'igm.jpg',tvrd()
>
> Do the Fourier Transform in IDL (based on Paul van Delst's examples):
>
> IDL> spec2 = temporary( [ spec, reverse( spec[ 1: n_elements(spec) - 2 ] ) ] )
> IDL> idl_igm = fft(temporary(spec2), /double, /inverse)
> IDL> idl_igm = shift(idl_igm, -1 * (n_elements(spec)-1))
>
> Plot the IDL result:
>
> IDL> plot, idl_igm
> IDL> write_jpeg, 'idl_igm.jpg', tvrd()
>
>
> 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)?  I recall having a similar problem with a 2pi phase-wrapping in
> MathCad a number of years ago that gave similar results but I cannot
> remember how to fix it.  I also cannot seem to reproduce the AIRS
> interferograms shown on Paul's site...
>
> My question is, how do I get the desired result (i.e. 'igm.jpg') in IDL?
>
> Cheers,
> Randall
>
> IDL Version 5.3, Linux RH 7.1
>
>

Subject: Re: IDL FFT (spec -> interferogram)
Posted by Paul van Delst on Thu, 04 Apr 2002 17:03:02 GMT
View Forum Message <> Reply to Message

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.

>
> 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?

paulv

--
Paul van Delst              Religious and cultural
CIMSS @ NOAA/NCEP           purity is a fundamentalist
Ph: (301)763-8000 x7274     fantasy
Fax:(301)763-8545           V.S.Naipaul

## 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