Subject: Re: FFT example. Help!
Posted by Paul van Delst on Mon, 01 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Peter Brooker wrote:
>
> I am trying to understand the FFT routine IDL uses. Part of my problem
> is that though I am familiar with Fourier transforms, I am somewhat
> unfamiliar with the fast Fourier transform.
>
> Has anybody written a program that works through a known transform using
> the FFT procedure? In particular, I want to be able to plot out F(u) vs
> u and have it "make sense".

Umm, I'm not quite sure what exactly it is you are asking but I have code I use to transform spectral data (i.e. down- or up-welling measured atmospheric radiance) into interferograms and vice versa. Check out:

http://airs2.ssec.wisc.edu/~paulv/#IDL Spectral

and look at the functions fft_to_interferogram.pro and fft_to_spectrum.pro. The actual FFT'ing is performed in one line (of course) with all the rest of the code for input checking and "x"-value (abscissae?) calculation (e.g. from frequency in cm^-1 to optical delay in cm and back).

If, on the other hand, you want FFT references, the online help gives a good general description and, I believe, a reference. For a comparison I did of the IDL FFT with the (Fortran) NR FFT, check out

http://airs2.ssec.wisc.edu/~paulv/fft/fft_comparison.html

It details some of the intricacies of organising the input and output data (mostly for the Fortran FFT) so you get what you expect on the ass-end of the problem. :o) The test code and data files are at the bottom of the page.

Hope this is helpful.

paul
--
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

Subject: Re: FFT example. Help!
Posted by Peter Brooker on Tue, 02 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

thanks for taking the time to reply. This helps allot!!

peter brooker

Alan Barnett wrote:

> I think your confusion stems from a misunderstanding of what the FFT
> does. There are at least three related, but not identical, operations
> that are referred to as "Fourier transforms".
>
> 1) Fourier integral.
> A function f(x) that is "well behaved", _continuous_ and _not periodic_
> has a Fourier transform F(k) that is "well behaved", continuous and not
> periodic. F and f are related by the Fourier transform and Fourier
> inversion formulas:
> F(k)=int(f(x) exp(-i kx) dx)
> and
> f(x)=int(F(k) exp(i kx) dx)
> The limits of the integral are + and - infinity.
>
> 2) Fourier series.
> If the function f(x) is a periodic function of x with period L, its
> Fourier transform is zero for all k that are not integral multiples of 2
> Pi /L, and the integral diverges if k is an integral multiple of 2 Pi /
> L. F(k) can be expressed as a sum of dirac delta functions, but it is
> more convenient to simplify the notation and write f(x) as a Fourier
> series:
> f(x) = sum(An sin(n Pi x / L) + Bn cos(n Pi x / L))
> where the Fourier coefficients An and Bn are
> An=(2/L) int(f(x) sin(n pi x / L) dx)
> Bn=(2/L) int(f(x) cos(n pi x / L) dx)
>
> 3) Discrete Fourier transform (DFT)
> If the function F(x) is a periodic function of x with period L and is only
> defined at N equally spaced discrete points xn = n L / N, n = 0, N-1, then
> its Fourier transform is a periodic function of k = 2 Pi / L and is
> nonzero only at the discrete values km = 2 m Pi / L, m = 0, N-1. Instead
> of considering the functions F(x) and f(k), which are sums of delta
> function, it is easier to consider Fm and fn, the coefficients of the
> delta functions. These coefficients are related by the discrete fourier
> tranform:
>
> Fm = sum(fn exp(-i 2 Pi m n / N))
>
> and its inverse

>
> fn = (1/N) sum(Fm exp(i 2 Pi m n / N))
>
> The FFT is a fast algorithm for computing the discrete fourier transform.
>
> If you want an example using the FFT that "makes sense" (I asume that you
> mean one that you can compute analytically), you have to remember that the
> FFT computes the discrete Fourier transform, not the Fourier integral.
> The simplest example is
> fn = 1 for all n
> Fm = N for m = 0; 0 otherwise
>
> The FFT is frequently applied to the problem of estimating the Fourier
> integral of a non-periodic function whose value is only known at a
> discrete set of sampled points. One then must cope with errors introduced
> by sampling (aliasing) and truncation (Gibbs ringing).
>
> For a good overview, look in Numerical Recipes by Press et al.
>
> I hope this helps.

Subject: Re: FFT example. Help!
Posted by asb on Tue, 02 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I think your confusion stems from a misunderstanding of what the FFT
does. There are at least three related, but not identical, operations
that are referred to as "Fourier transforms".

1) Fourier integral.
A function f(x) that is "well behaved", _continuous_ and _not periodic_
has a Fourier transform F(k) that is "well behaved", continuous and not
periodic. F and f are related by the Fourier transform and Fourier
inversion formulas:
F(k)=int(f(x) exp(-i kx) dx)
and
f(x)=int(F(k) exp(i kx) dx)
The limits of the integral are + and - infinity.

2) Fourier series.
If the function f(x) is a periodic function of x with period L, its
Fourier transform is zero for all k that are not integral multiples of 2
Pi /L, and the integral diverges if k is an integral multiple of 2 Pi /
L. F(k) can be expressed as a sum of dirac delta functions, but it is
more convenient to simplify the notation and write f(x) as a Fourier
series:
f(x) = sum(An sin(n Pi x / L) + Bn cos(n Pi x / L))

where the Fourier coefficients An and Bn are
An=(2/L) int(f(x) sin(n pi x / L) dx)
Bn=(2/L) int(f(x) cos(n pi x / L) dx)

3) Discrete Fourier transform (DFT)
If the function F(x) is a periodic function of x with period L and is only defined at N equally spaced discrete points xn = n L / N, n = 0, N-1, then its Fourier transform is a periodic function of k = 2 Pi / L and is nonzero only at the discrete values km = 2 m Pi / L, m = 0, N-1. Instead of considering the functions F(x) and f(k), which are sums of delta function, it is easier to consider Fm and fn, the coefficients of the delta functions. These coefficients are related by the discrete fourier tranform:

Fm = sum(fn exp(-i 2 Pi m n / N))

and its inverse

fn = (1/N) sum(Fm exp(i 2 Pi m n / N))

The FFT is a fast algorithm for computing the discrete fourier transform.

If you want an example using the FFT that "makes sense" (I asume that you mean one that you can compute analytically), you have to remember that the FFT computes the discrete Fourier transform, not the Fourier integral.
The simplest example is
fn = 1 for all n
Fm = N for m = 0; 0 otherwise

The FFT is frequently applied to the problem of estimating the Fourier integral of a non-periodic function whose value is only known at a discrete set of sampled points. One then must cope with errors introduced by sampling (aliasing) and truncation (Gibbs ringing).

For a good overview, look in Numerical Recipes by Press et al.

I hope this helps.

Subject: Re: FFT example. Help!
Posted by Julio Maranhao on Tue, 02 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

If you want to use the IDL FFT functions in continuous mathematical

straight. You must understand how real signals are discretized and processed. A classical book is "Discrete-Time Signal Processing" of Oppenheim&Schafer. I suggest you to talk to a professor or a person that understand Discrete Fourier Transform for a fast study. All I can advance is that the "key" of DFT is : if the signal is periodic and discrete (a vector in IDL could be a "window" of this infinite signal, for instance) than the Transform is periodic and discrete.

For example, try these:

```
IDL> a=[1.0, 1, 1, 1, 0, 0, 0, 0]
IDL> window,0  &  plot,abs(fft(a,-1))
IDL> b=fltarr(500)  &  b[0:7]=a
IDL> window,1  &  plot,abs(fft(b,-1))
```

In the second plot it seems more like a sinc, because I added more zero points in the vector. The other fft is undersampled. This is one of the innumerous properties of discrete periodic signals. And this is only part of the iceberg. :-). Good luck.

mensagem:390DA60B.B8F5CEBA@email.sps.mot.com...

```
>  I am trying to understand the FFT routine IDL uses. Part of my problem
>  is that though I am familiar with Fourier transforms, I am somewhat
>  unfamiliar with the fast Fourier transform.
>
>  Has anybody written a program that works through a known transform using
>  the FFT procedure? In particular, I want to be able to plot out F(u) vs
>  u and have it "make sense".
>
>  An example of a known transform is
>
>  For
>  f(x) = 1 for -1/2<x<1/2
>       = 0  else
>
>  the Fourier transform F(u) is given by
>
>  F(u)=int(f(x)*exp(-j*2*pi*ux)*dx
>       =[sin(pi*u)]/pi*u
>
>
>  thanks-Peter Brooker
>
```