Subject: Secrets FFTs revealed!!
Posted by Peter Brooker on Thu, 04 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I have just been through a learning curve on FFTs. Much thanks to Alan Barnett for putting me on the right track. I think I have them figured out and now want to write a reference that captures my present level of understanding. Realize that I have learned only as much of the FFT theory as needed. My motivation is that I am going to be applying the FFT functions to modeling the effect of a finite lens size on the image. (The finite lens size will chop off the higher order frequencies).
Perhaps somebody else will want to expand/improve this reference. These are only my best guesses to how everything works. Perhaps this is something for the IDL FAQ.

This reference is organized as follows:
PART#1:Relate complex expansion to real Fourier series
PART#2:IDL form of complex expansion
PART#3:Specific example.

PART#1: Relate complex expansion to real Fourier series.
Assume you have a function f(t) that is periodic in t with a period T.
Then there exists coefficients a_n & b_n such that

f(t) = a_o + sum_n(a_n*cos(2*pi*n*t/T)+b_n*sin(2*pi*n*t/T)) , n=1,2,...

This is just the Fourier series of function with period X. Nothing new here. See eq #4, section 10.3, Advanced Engineering Mathematics, Kreyszig. This expansion though assumes -T/2 < t < T/2

Now consider an alternate form of the Fourier series expansion.

f(t)=sum_n(A_n*exp(j*2*pi*n*t/T)), n=0,1,2,...

In order for me to be comfortable with this expansion I need to see how this expansion relates to the expansion above. In particular, how do the complex An relate to the real a_n and b_n?

Consider the following:

II= sum_n(A_n*exp(j*2*pi*n*t/T)), n=0,1,2,... j*j = -1

=A_o+sum_n(A_n*(cos(2*pi*n*t/T)+j*sin(2*pi*n*t/T)), n=1,2,...

Let A_n=(aa_n+j*bb_n)

II= A_o+sum_n((aa_n+j*bb_n)*(cos(2*pi*n*t/T)+j*sin(2*pi*n*t/T))

= A_o + sum_n(aa_n*cos() - bb_n*sin() + j*(bb_n*cos()+aa_n*sin())

Real(II) = Real(A_o) + sum_n(aa_n*cos(2*pi*n*t/T)-bb_n*sin(2*pi*n*t/T))

Comparing to the first expansion we see that

Real(A_o)=a_o,   aa_n=a_n,   -bb_n=b_n

To me, this proves existence of the complex expansion. Knowing one, you can figure out the other. Part #1 is complete.

Part #2: IDL form of complex expansion

Let f(t) be a periodic function with period T defined on an interval [0,T].

Then there exist complex A_n such that

f(t)= sum_n(A_n*exp(j*2*pi*n*t/T)), n=0,1,2,...   j*j = -1

Divide the interval into N sections. t~t_i = i*T/N
Then,
f(t_i) = sum_n(A_n*exp(j*2*pi*n*t_i/T))
        = sum_n(A_n*exp(j*2*pi*n*i*(T/N)/T))
        = sum_n(A_n*exp(j*2*pi*n*i/N)) , n=0,1,...

This is exactly what is found in the IDL manual under the section for FFT. The only difference is that t has been replaced by u and A_n has been replaced by F(u). Note that the period T has dropped out. Also note that t has been replaced by t_i = i*T/N. In order for this to happen, the interval over which t is defined must be from [0,T]. This is different from the definition of t being defined over the interval [-T/2,T/2]. Perhaps this is why b_n = -bb_n.

********UNFORTUNATELY IT IS WRONG***************

What is wrong is the values of n in the sum. IDL does not use the values of n=0,1,2,... IDL actually uses n= -N/2+1, -N/2+2, ...-1,0,1,...,N/2
The reason for doing this must have to do with FFT theory. Note also that the number of values of n is N.

It gets more complicated. From the manual we have

F(u) = 1/N*sum_x(f(x)*exp(-j*2pi*ux/N)) , x=0,1,...N-1

First thing to realize is that F(u) is really F_n. Where n is an integer. This comes from the fact that f(x) is periodic in x.

The manual also mentions that the "frequencies" are
Fo, 1/(NT),2/(NT),...,1/2T,-(N-2)/(2NT),...,-1/NT

After trial and error I have determined that the value of the ns range
for -N/2 to N/2. Futhermore, the F_n are stored in the order associated
with the following values of n

0,1,2,...,N/2,-(N/2-1),-(N/2-1),...,-1 <== this is bizarre!!

Let N=8. Then N/2=4

The F_n would be stored in an array. The array of n values associated
with this array would be:

[0,1,2,3,4,-3,-2,-1]

Part #3: Specific Example

Consider the interval t = [0,1]. This choice of interval implies T=1.
Let f(t) = sin (4*pi*t)

f(t_i)=sin(2pi*2*i/N), i=0,1,...N

f(t_i)=sum_n(A_n*exp(-j*2pi*n*i/N)) , n=-N/2,...-1,0,1,...N/2
    = A_nN/2... + A_n2*(cos(2pi*(-2)*i/N)+j*sin(2pi*(-2)*i/N))+
        + A_o+A_n1*exp()+A_1*exp()+
        A_2*(cos(2pi*(2)*i/N)+j*sin(2pi*(2)*i/N))
+ A_3*exp()+...
    =... + A_n2*cos(2pi*2*i/N)+A_2*cos(2pi*2*i/N) +
      +A_n2*j*(-1)*sin(2pi*2*i/N)+A_2*j*sin(2pi*2*i/N)) + ...
    = ... + (A_n2+A_2)*cos(2pi*2*i/N)+j*(-A_n2 + A_2)*sin(2pi*2*i/N)
+ ...

where A_n2 stands for A_n where n= -2

Equating the series to sin(2pi*2*i/n) we conclude

A_n = 0 for all n except n = -2 or n = 2.

A_n2+A_2=0
j*(-A_n2 + A_2) = 1

Let A_n2=(a_n2+j*b_n2) and A_2=(a_2 + j*b_2)

The above equations imply

(a_n2 + a_2)   + j*(b_n2+b_2) = 0 &
j*[(-a_n2 + a_2) + j*(-b_n2 + b_2)] = 1

==> a_n2 + a_2=0, b_n2+b_2 =0 ==> a_n2= - a_n2, b_n2 = -b_n2

==> 2*a_n2=0 ==> a_n2=a_2 = 0
==> j*j*(2*b_2)=1 ==> 2*b_2 = -1/2, b_2 = 1/2

A_n2 = 0 + j*(1/2)

A_2 = 0 + j*(-1/2)

We now have calculated the solutions.

The following code calculates this and displays the correct answers. It shows how to plot A_n vs n correctly.

```
;idl_program fft_sine.pro
TT=1
Npts=100
t=findgen(Npts)/(Npts-1)*TT
f_t=sin(4.*!pi*t/TT)
!p.multi=[0,2,3]
plot,t,f_t, title='f(t) vs t'
A_n=fft(f_t,-1) ; complex fourier coefficients

plot,float(A_n),yrange=[-.5,.5],title='float(A_n)'
plot,imaginary(A_n),yrange=[-.5,.5],title='imaginary(A_n)'

a=findgen(Npts/2+1)
b=-reverse(findgen(Npts/2-1)+1)
c=[a,b] ; c=[-N/2+1,-N/2+2, ...,-1,0,1,...,N/2]
print,c
sub=sort(c)
 plot,c(sub),float(A_n(sub)),yrange=[-.5,.5],title='float(A_n ) vs n'
plot,c(sub),imaginary(A_n(sub)),yrange=[-.5,.5], $
       title='imaginary(A_n) vs n'
plot,c(sub),imaginary(A_n(sub)),xrange=[-5,5],$
       title='imaginary(An) vs n' ; finer x scale
end
```