Ast 4101/Phys 4041 Homework #10
Due Tuesday, November 24, 2015

For all problems you should include a well-commented copy of the source code as well as any requested solutions and plots obtained. Codes may be written in FORTRAN, C or C++. “Apply” means provide appropriate results from the calculations.

1) For this problem write and apply a program to test the Fast Fourier Transform routine provided for you on the course homepage. There you will find both C++ and FORTRAN versions of the FFT routine, ‘CFFT’. The ‘C’ indicates that complex FFTs are returned. Arguments for CFFT are (x,y,N,m,itype). X and Y are the sets of 4 byte real and imaginary values of the function, f, to be transformed. These variables will be vectors; each should have N = 2^m elements. If itype = -1 the routine computes the inverse transform. Otherwise it returns the forward transform. Note that in either case, x and y are replaced by their transformed values. So, save copies of the original x and y vectors. Note also that if x and y are not 4 byte reals, the transform results will be incorrect. If your input function is real, simply set y(i) = 0, for each index i. Always remember that both the input function and its transform are assumed to be periodic with period N. Remember also that FFT routines shift the transformed function so that it runs between 0 ≤ k ≤ N-1, whereas the ‘normal’ order runs between –N/2 ≤ k ≤ N/2. The extra value contained in the latter range is available using periodicity.

a) Write your program so that it will transform an arbitrary ‘user defined’ function with N = 2^m uniformly sampled values over the “time” interval 0 ≤ T ≤ 1. Several test functions will be applied. Your program should then also carry out the inverse FFT in order to recover the original function.

b) Test your program using N = 128 with the following functions: i) f(t) = 1, ii) f(t) = cos(2πt), iii) f(t) = sin(2πt). The functions would be discretely sampled at t = i/N, where 0 ≤ i ≤ N-1. In each case compare the transform found (both real and imaginary parts) with theoretical expectations and comment on the degree to which the symmetries of the computed discrete Fourier Transform are consistent with the theoretical relations presented in class. Provide plots of each function, its FFT and also the inverse FFT of the FFT you compute.

2) Many detected signals, s(t), are convolved through a response function, r(t). FFTs are a powerful way to model that process (and to do the deconvolution). In this problem you need to write a routine using FFTs to compute the convolved function h(t) = r*s = ∫-∞∞ r(τ)s(t-τ)dτ. To do this, given s(t) and r(t) you need to find the FFTs of each, then take their (complex) product. Finally, you need to compute the inverse FFT of the product to obtain h(t). Provide plots to demonstrate your results.

Write your program so that it can find the convolution of arbitrary, user defined functions s and r.

a) Test your program using the identity response function r(t) = 1 if t = 0 and otherwise r(t) = 0. Assume the ‘top hat’ signal function, s(t) = 1 if 1/3 ≤ t ≤ 2/3 and s(t) = 0 otherwise.
b) Apply your routine now using the Gaussian response function

\[ r_i = \frac{a}{N \sqrt{\pi}} \exp\left\{ -\frac{(ai)^2}{N^2} \right\}, \]

where \(-N/2 \leq i \leq N/2\). You will need to shift the indices into the range \(0 \leq i \leq N-1\), before you carry out the FFT of \(r\). Since \(r\) is periodic you can do this simply by adding \(N\) to the indices whenever \(i < 0\). Try several values of \(a/N\) to see the influence of “softer” (broader) or “harder” (narrower) response functions. Present plots and describe the effect of varying \(a/N\).