Yigong Zhang HW3

BioEC165/EEC145B Fall
Prof. Steve Conolly

Homework #3

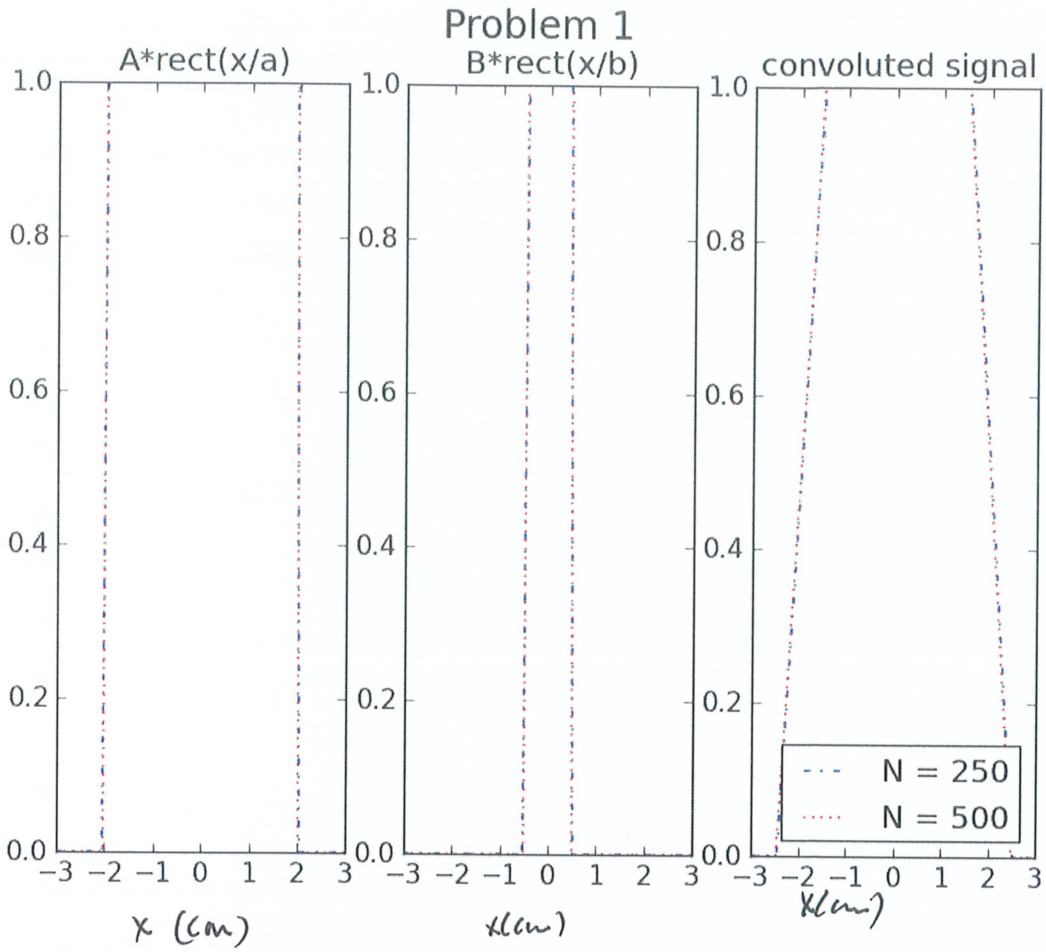
Problem 1: In an earlier HW, you computed A rect(x/a)$\ast$ B rect(x/b) **analytically**. You found that the peak amplitude was AB min(a,b). Here, let's let A=B=1 and a=4, b=1.
(a) Write a **Matlab** routine that computes the discrete version of this convolution using Matlab's conv(). Write your TestConvolveRect(N) routine where N is the number of sample points over the interval x=(-10 to 10). (Of course, you may have more inputs if you like.) **Hint**: don't neglect the scale factor to convert the discrete convolution into an approximation to the continuous result.
(b) Check the width of your convolution is correct and nearly identical for N=250 and for N=500.
(c) Check the amplitude is correct and nearly identical for N=250 and for N=500.

Problem 2: Discrete Calculus
If you convolve a vector with [1 -1] you obtain a function proportional to the derivative of the original function. Why? Read below:
(a) Consider a continuous function, f(x) sampled at spacing $\Delta x=L/N$ where L is the total FOV and N is the total number of points. Use the limit-formula for a derivative to show that you can write the sampled version of the derivative of f(x) as f[]$\ast$ [1 -1]*d. **Hint**: all you need to do is find the scale factor, d, which may depend on (N, Δx, L).
(b) Check you answer in (a). Using **Matlab**, plot the function $\cos(2\pi x)$ over the range x=(0, 4) as well as it's *analytic* derivative, $-2\pi \sin(2\pi x)$. Compare your discrete derivative and theory with both N=250 and N=500 points (4 total plots displayed on 2 axes). Does your approximate result match $-2\pi \sin(2\pi x)$? **Hint**: all 4 plots should have amplitude 2π once you choose d properly.

Problem 3: Modeling a More Realistic Sampler
In class, we modeled the sampling operation by multiplying f(x) by comb(x/ Δx); this essentially models the loss of all information between sample points. In reality, most sampling functions integrate the function over a short interval before they produce a sample point. That is, the nth sample point is the integral of f(x) from $x= n\Delta x- \Delta x/2$ to $n\Delta x+ \Delta x/2$. Assume that F(k) is bandlimited to $\pm k0$.
(a) Model this **integrating-sampling** operation (using only multiplication and convolution operations) between f(x), comb(x) and one other familiar function.
(b) Re-derive the sampling theorem for this type of sampling.
(c) Does this type of sampling still obey the (same) Nyquist limit?
(d) Does this type of sampling lose any information?

Problem 4: FT of comb(x/Δx)
We stated (almost without proof) that the continuous Fourier transform of the comb(x/Δx) function is Δxcomb(kΔx). Here you directly plot the continuous FT of comb(x) to show it looks like a comb!
(a) show analytically that comb(x/Δx) has FT $\Delta x\sum_m \exp(-i\, 2\pi k\, m\, \Delta x)$; $m=(-\infty,\infty)$.
(b) Write a **Matlab** routine that plots this sum above as a function of k for a *finite* number of terms m from (-M,M). (Hint: you can simply add terms or notice there is a closed-form solution for the finite sum.) Plot the resulting estimate for M=10 and for M=50. Does this look like Δxcomb(kΔx)? Plot this over the range of $k=\mp 3/\Delta x$. Here let $\Delta x=0.1$ cm.

Problem 5: Sampling in X-Space
(a) Estimate the Nyquist sampling rate for $\exp(-\pi x^2/w^2)$?
(b) Roughly how many *visible* samples would that give you in x-space?



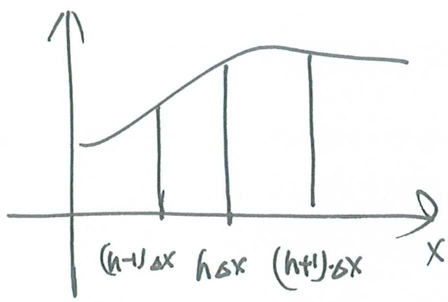
Published from p1.py using Pweave 0.22rc on 27-09-2013.

We can see that the line from N=250 overlaps with that from N=500. So ~~for~~ for N=250 and 500, they have the same width and amplitude.

```
import numpy as np
import matplotlib.pyplot as plt
ctable= ['b','r']
lstyle = ['-.',':']
f0, [ax0,ax1,ax2] = plt.subplots(1,3)
f0.suptitle('Problem 1',fontsize = 'x-large')
ax0.set_title('A*rect(x/a)')
ax1.set_title('B*rect(x/b)')
ax2.set_title('convoluted signal')
for i, N in enumerate([250,500]):
    A = 1.
    B = 1.
    a = 4.
    b = 1.
    x1 = np.linspace(-10,10,N)
    x2 = np.linspace(-10,10,N)
    y1 = np.where((x1>-a/2) & (x1<=a/2), A*1., 0.)
    y2 = np.where((x2>-b/2) & (x2<=b/2), B*1., 0.)
    #y1 = y1/y1.sum()
    y2_scaled = y2/y2.sum()
    trapz = np.convolve(y1,y2_scaled,mode='same')
    ax0.hold(True)
    ax0.plot(x1,y1, color = ctable[i], linestyle = lstyle[i])
    ax0.set_xlim(-3,3)
    ax1.plot(x2,y2, color = ctable[i], linestyle = lstyle[i])
    ax1.set_xlim(-3,3)
    ax2.plot(x1,trapz, color = ctable[i], linestyle = lstyle[i])
    ax2.set_xlim(-3,3)
    ax2.legend(('N = 250','N = 500'),loc = 'lower right')
plt.show()
```

2.

a)

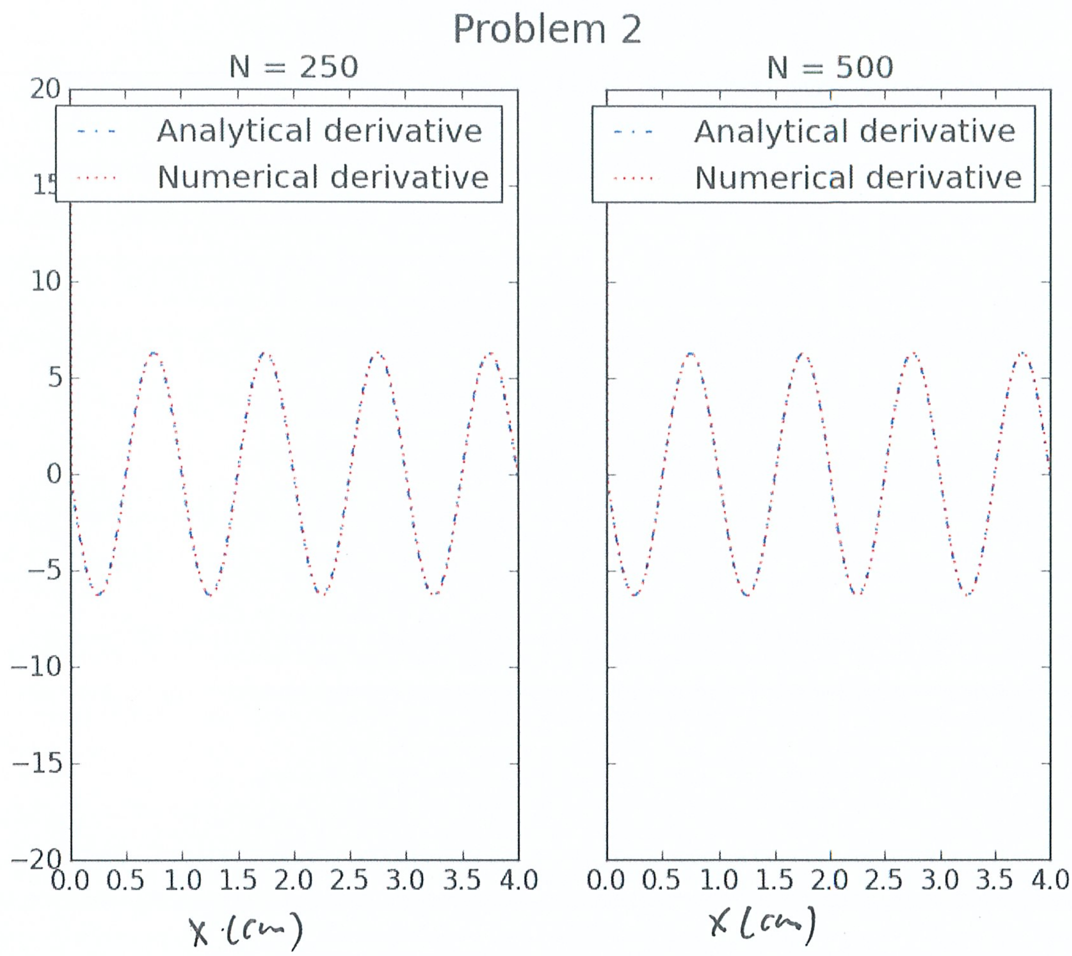


$$f'(n\cdot\Delta x) = \frac{f[n\cdot\Delta x] - f[(n-1)\cdot\Delta x]}{\Delta x}$$

$$f(n\cdot\Delta x) * [1, -1]\cdot d = \left(f[n\cdot\Delta x] - f[(n-1)\cdot\Delta x]\right)\cdot d$$

So $\quad d = \frac{1}{\Delta x} = \frac{N}{L}$

b). [Next page]
In those two plots, the line given by analytic way and by numerical way overlap with each other. So they have the same shape, same amplitude. proof!

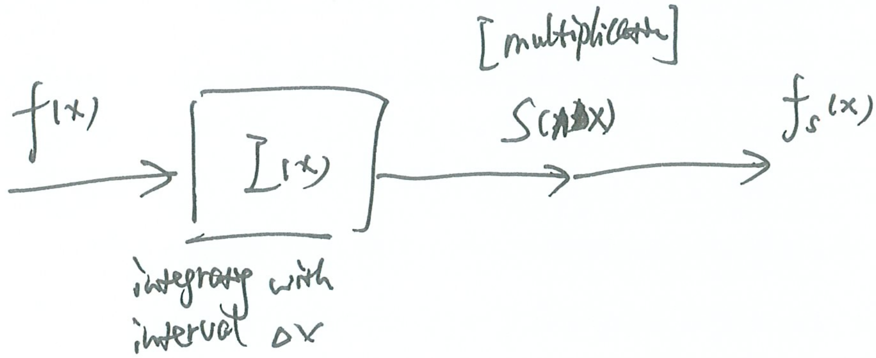


Published from p2.py using Pweave 0.22rc on 27-09-2013.

```
##############
# Problem 2 #
##############
import numpy as np
import matplotlib.pyplot as plt
from math import pi
f0, ax = plt.subplots(1,2,sharex=True, sharey=True)
for i, N in enumerate([250,500]):
    x = np.linspace(0,4,N)
    y = np.cos(2*pi*x)
    y_analytic = -2*pi*np.sin(2*pi*x)
    der = [1,-1]
    y_num = np.convolve(y,der,mode='same')
    ax[i].plot(x,y_analytic, linestyle = '-.')
    ax[i].hold(True)
    ax[i].plot(x,y_num*N/4, c = 'r', linestyle = ':')
    ax[i].legend(['Analytical derivative','Numerical derivative'])
    ax[i].set_title('N = %s' %str(N))
    ax[i].set_ylim(-20,20)
f0.suptitle('Problem 2', fontsize = 'x-large')
plt.show()
```

3.

a)



$$f_s(x) = f(x) * \Pi\left(\frac{x}{\Delta x}\right) \cdot \text{comb}\left(\frac{x}{\Delta x}\right)$$

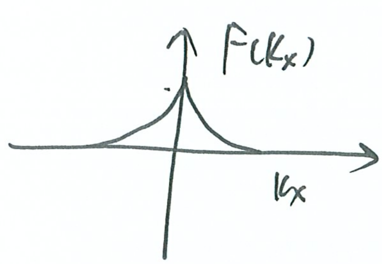
Integrating - sampling operation

b)

In frequency domain:

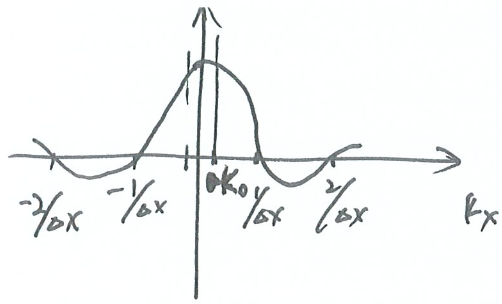
$$F_s(k_x) = F(k_x) \cdot \Delta x \,\text{sinc}(\Delta x \cdot k_x) * \Delta x \cdot \text{comb}(\Delta x \cdot k_x)$$

so let's say we have $F(k_x)$ as



So what we are doing in frequency domain while we are doing the integrating sampling in spatial domain is:

First, multiply the frequency domain by $\Delta x \cdot \mathrm{sinc}(\Delta x \cdot k_x)$.



Second, replicate the result at distance $= n \cdot \frac{1}{\Delta x}$. $n = 1, 2, \cdots$

Interpolation:

Since it has limited bandwidth $= 2K_0$,

$$\frac{\Pi\left(\frac{k}{K_0}\right) \cdot F_s(k)}{\Pi\left(\frac{k}{K_0}\right) \cdot \mathrm{sinc}(\Delta x \cdot k_x)} = F(k).$$

c) To avoid aliasing, we need

$\frac{1}{\Delta x} > 2 \cdot K_0$, which is the Nyquist criteria for this integrating sampling.

D) No. Based on the Interpolation theorem derived above, everything is robust!

4. $\mathcal{F}\{\text{comb}(x/\Delta x)\} = \int_{-\infty}^{\infty} \text{comb}(x/\Delta x) \cdot e^{-i2\pi kx}\, dx$

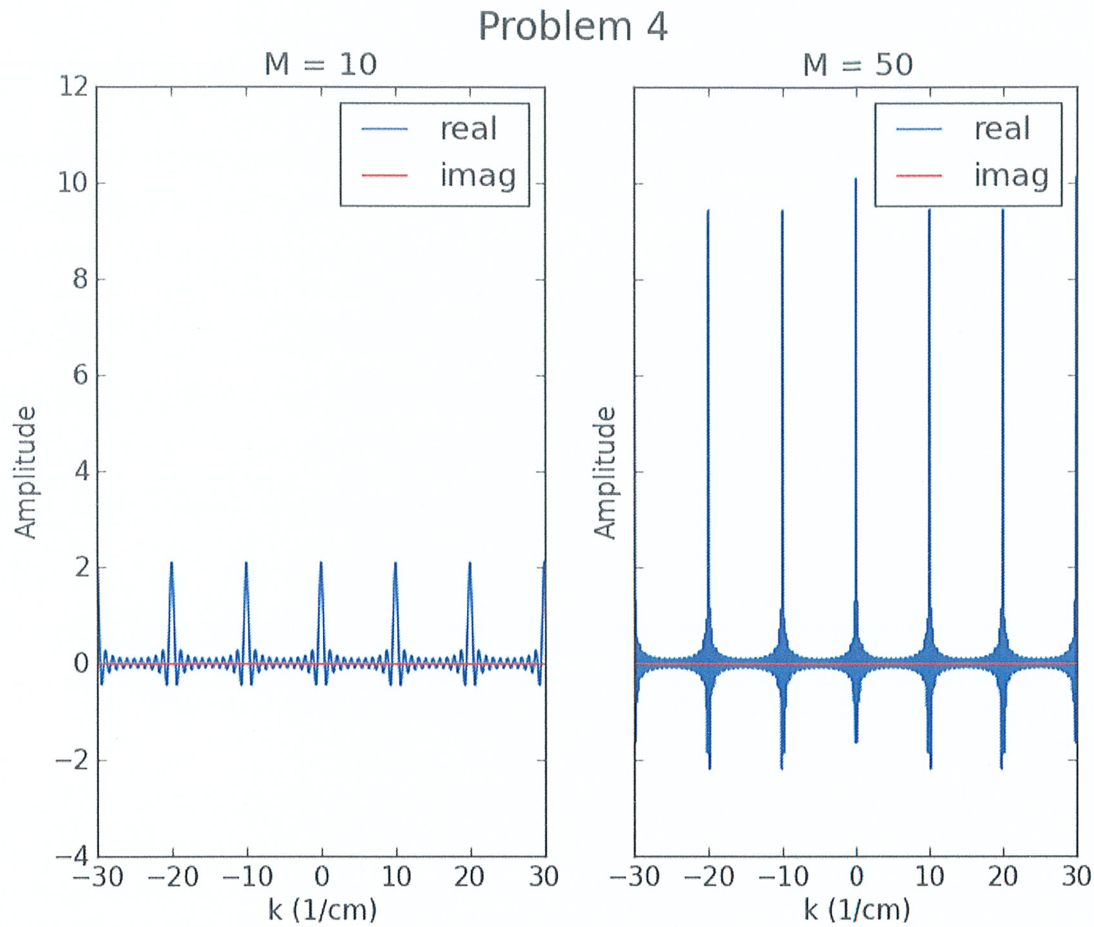
$$= \int_{-\infty}^{\infty} \Delta x \cdot \sum_m \delta(x - m\Delta x) \cdot e^{-i2\pi kx}\, dx$$

$$= \Delta x \sum_m \exp(-i \cdot 2\pi k m \Delta x) \qquad \text{for } m = (-\infty, +\infty)$$

The plot is on next page.

b). $\Delta x \cdot \text{comb}(k\Delta x)$ is a ~~function~~ bunch of delta function seperated by distance $1/\Delta x$ and have amplitude $1/\Delta x$. Here $\Delta x = 0.1$ cm.

So we can see that as M increases, the plot looks more like $\Delta x \cdot \text{comb}(k\Delta x)$ The reason is because we add more terms in fourier series. so it's more similier to what it is.



Published from p4.py using Pweave 0.22rc on 27-09-2013.

```
##############
# Problem 4 #
##############
import numpy as np
import matplotlib.pyplot as plt
from math import pi
f0, ax = plt.subplots(1,2,sharex=True, sharey=True)
f0.suptitle('Problem 4', fontsize = 'x-large')
N_k = 1001
for i, M in enumerate([10,50]):
    m = np.array(range((2*M+1)))-M
    k = np.linspace(-30,30,N_k)
    dx = 0.1
    m = m[:][np.newaxis]
    k = k[:][np.newaxis]
    matx = np.dot(m.T,k)*np.complex(0,1)
    ft_matx = dx*np.exp(2*pi*matx*dx)
    k = np.linspace(-30,30,N_k)
    sigma = np.sum(ft_matx,axis = 0)
    ax[i].plot(k,sigma.real, c= 'b')
    ax[i].hold(True)
    ax[i].plot(k,sigma.imag, c = 'r')
    ax[i].legend(['real','imag'])
    ax[i].set_title('M = %s' %M)
    ax[i].set_xlabel('k (1/cm)')
    ax[i].set_ylabel('Amplitude')
plt.show
```

5.

a) $$\mathcal{F}\{e^{-\pi x^2/w^2}\} = e^{-\pi k_x^2 \cdot w^2}$$

If we sample in spatial domain with separation Δx,
we are gonna replicates its frequency domain distribution by distance $1/\Delta x$

Assume: the amplitude lower than 0.01 has not effect on aliasing in freq space.

$$e^{-\pi k_0^2 \cdot w^2} = 0.01$$

$$k_0 \cdot w = 1.21$$

$$k_0 = 1.21/w$$

So

$$\frac{1}{\Delta x} > 2k_0$$

$$\frac{1}{\Delta x} > 2.42/w$$

$$\Delta x < w/2.42$$

b).

Assume: we can't see the amplitude lower than 0.01 in spatial domain, either.

$\cancel{N = \Delta x}$

$$e^{-\pi x_0^2/w^2} = 0.01$$

$$\left(x_0/w\right) = 1.21$$

$$x_0 = 1.21 \cdot w.$$

$$\cancel{N = 2 \cdot x_0/\Delta x \quad \text{if} \sim \text{Nyquist freq} \quad 2x_0.}$$

$$N = 2x_0/\Delta x > \cancel{2x} \; 2 \cdot 1.21 \cdot w \cdot 2.42/w = 5.13$$

$$N \geq 6$$