NOTES ON THE MATLAB EXERCISE

1) The important points in this first problem are that

```
nx=2048;
kc=64/nx;
x=0:nx-1;
```

The 2048 is not required but note that the FFT works best with powers of 2.
You can make kc wavenumber anything between 2 to nx/2. Try a different kc
The cosine function starts at cos(0)=1 but cos(2*pi*(nx-1)*kc) is one point less than 1. This is important.
Try it a different way and you won’t get perfect peaks.

2) These wavenumbers go from k=-nx/2:nx/2-1; So they are negative through positive. The Nyquist wavenumber is nx/2. The asymmetry is important. Try it a different way and the results won’t work.

3) Redo with a sin function

4) Show that the Fourier transform of a Gaussian function is a Gaussian function.

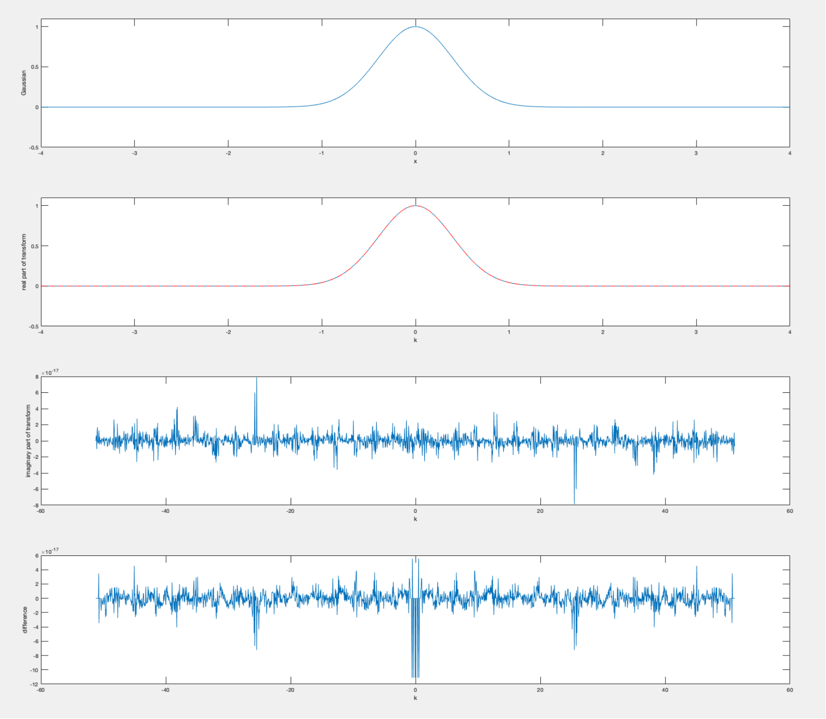
```
%    Plot the difference between the fft result and the exact function.
%    When you do this problem, it is best to make the Gaussian function an even function
%    of x just prior to computing the fft(). If you do this then the transformed
%    Gaussian will be real and even.  Also you will need to scale the transform by
%    the point spacing dx = L/nx;
```

```
nx=2048;
L=20;
dx=L/nx;
a=1.;
x=(-nx/2:nx/2-1)*dx;
g=exp(-pi*x.*x*a*a);
k=(-nx/2:nx/2-1)/L;
```

You will need to scale the transform by dx

```
cg=fftshift(fft(fftshift(g)))*dx;
```

The length L should be about ~10-20 times greater than the width of the Gaussian or you can get Fourier edge effects. Try it!



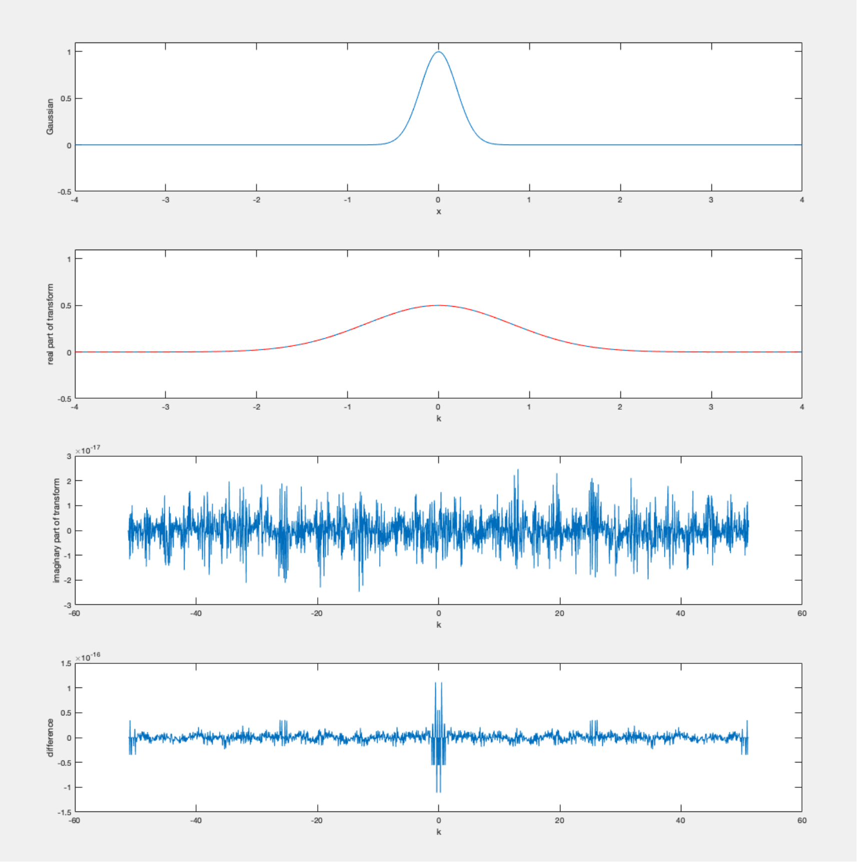
5) use this Gaussian example to demonstrate the stretch property of Fourier Transforms. Increase or decrease a.

```
a=2.;
g=exp(-pi*x.*x*a*a);
```

The analytic result is

```
fg=exp(-pi*k.*k/(a*a))/a;
```

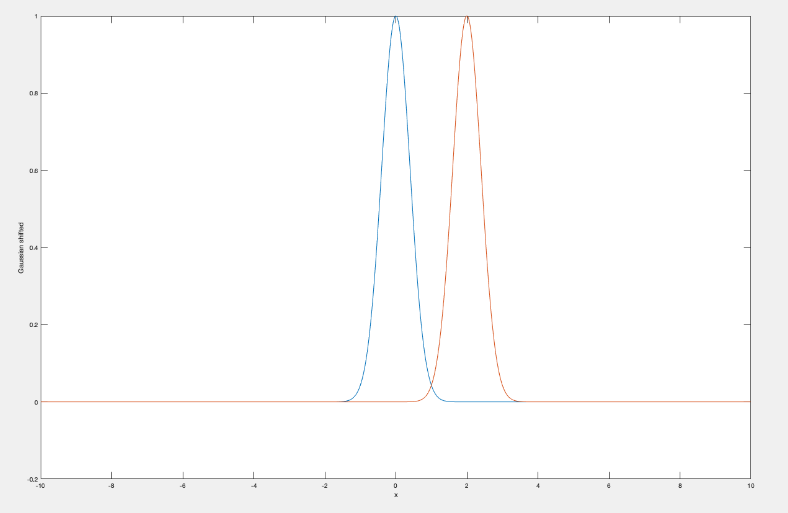
Compare this with the fft result. The Gaussian is narrower in the x-domain so the Gaussian is wider and shorter in the k-domain.



6) use this Gaussian function to illustrate the shift property of the Fourier transform. The blue line is the original unshifted Gaussian and the red is the shifted by 2. I'll give you all the code for this one but try on your own first. Note how L is used.

```
nx=2048;
L=20;
dx=L/nx;
s=1.;
x=s *(-nx/2:nx/2-1)*dx;
g=exp(-pi*x.*x);
%
%  generate the wavenumbers
%
k=(-nx/2:nx/2-1)/L;
%
figure(4)
cg=fftshift(fft(g));
a=2.;
arg=-i*2.*pi*a*k;
shft=exp(arg);
g2=real(ifft(fftshift(shft.*cg)));
plot(x,g,x,g2);
xlabel('x')
```

```
ylabel('Gaussian shifted')
pause
```

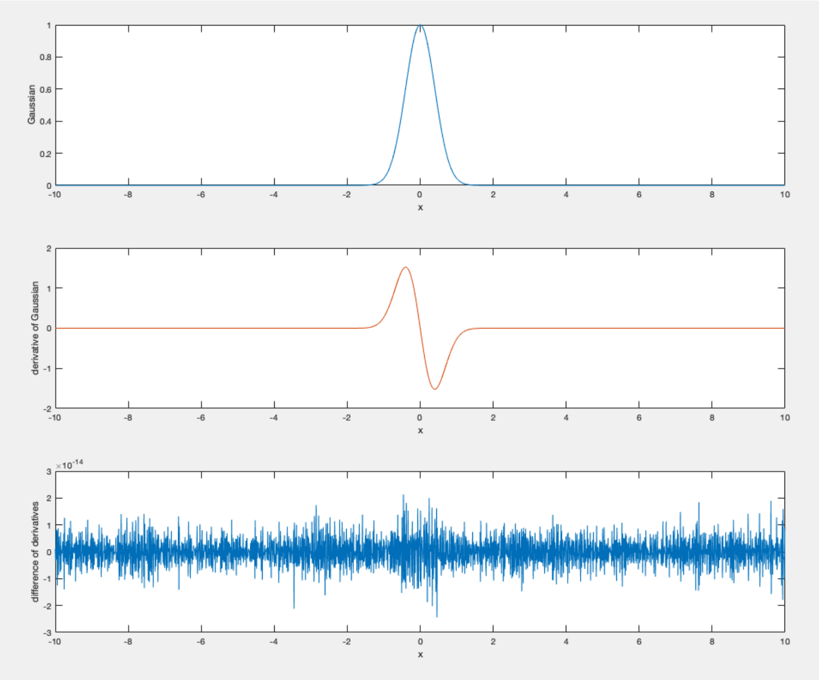


```
%
% 7) use the Gaussian function to demonstrate the derivative property of the Fourier transform
%
```

Same as 6 but replace shft=exp(arg) with i2*pi*k and compare with the analytical derivative of a Gaussian which you will need to calculate on paper.

Note that problems 5, 6, and 7 are all the same. Make any function f(x), take the Fourier transform. Make a transfer function G(k) in the wavenumber domain. This could be shift, scale, upward continuation or anything. Multiple F(k)*G(k). Inverse transform the result and you will have the answer.

KEEP this code because we will use it several more times. The transfer functions will always be a solution to a differential equation such as heat flow, magnetics, gravity, elasticity, . . .


Gaussian
x
derivative of Gaussian
x
$\times 10^{-14}$
difference of derivatives
x