Homework 1 – Fourier's transform using Matlab

1. Write a Matlab script to compute the Fourier transform $\hat{g}_0(k_1)$ of

$$g_0(x_1) = \frac{1}{1 + (x_1/l)^2}$$

   Verify your result with the analytical solution $\hat{g}_0(k_1) = (l/2)\exp(-l|k_1|)$, and plot the two functions. Verify also Parseval's identity.

   Take the inverse Fourier transform to retrieve $g_0(x_1)$.

   Study two cases corresponding to a low and a high frequency resolution, by justifying your choice.

   A tutorial is provided in the next page.

2. Write a short report to document your Matlab script. Note that the script will be used in the next classroom to calculate solutions based on Fourier integrals.

Definition of the Fourier transform

$$\hat{g}_0(k_1) = \mathcal{F}[g_0(x_1)] = \frac{1}{2\pi}\int_{-\infty}^{+\infty} g_0(x_1)e^{-ik_1x_1}dx_1 \qquad g_0(x_1) = \mathcal{F}^{-1}[\hat{g}_0(k_1)] = \int_{-\infty}^{+\infty} \hat{g}_0(k_1)e^{ik_1x_1}dk_1$$

Parseval's identity

$$\int_{-\infty}^{+\infty} g_0^2(x_1)dx_1 = 2\pi\int_{-\infty}^{+\infty} |\hat{g}_0(k_1)|^2\,dk_1$$

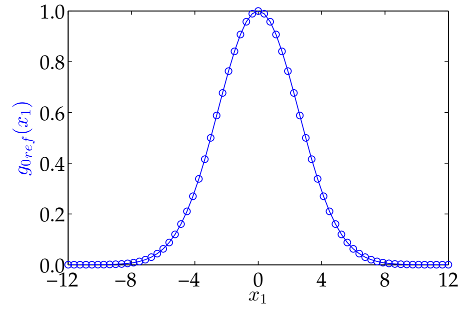
fft	ifft
$[\hat{g}_{0l}] = \sum_{n=1}^{N} g_{0n}\, e^{-i2\pi(n-1)(l-1)/N} \qquad 1 \le l \le N$	$[g_{0n}] = \frac{1}{N}\sum_{l=1}^{N} \hat{g}_{0l}\, e^{i2\pi(n-1)(l-1)/N} \qquad 1 \le n \le N$
$\hat{g}_{0l} = [\hat{g}_{0l}] \times \frac{dx_1}{2\pi}$	$g_{0n} = [g_{0n}] \times \frac{2\pi}{dx_1} \qquad \left(dk_1 = 2\pi \times \frac{1}{N dx_1}\right)$

TABLE 1 – FFT using Matlab. Note that *fftshift* and *ifftshift* are exactly the same for N even : it is recommended to choose N even (and it just doesn't matter !)

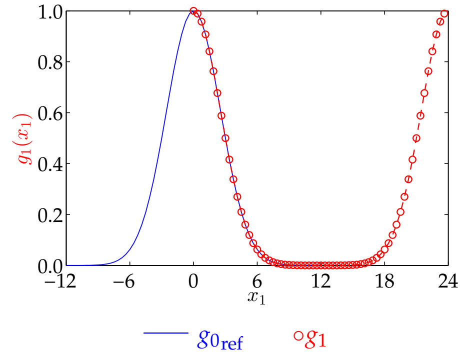
Tutorial for a Gaussian function g_0

$$g_0(x_1) = e^{-\ln 2 \left(\frac{x_1}{b}\right)^2} \qquad \hat{g}_0(k_1) = \frac{b}{2\sqrt{\pi \ln 2}} e^{-\frac{(bk_1)^2}{4\ln 2}}$$

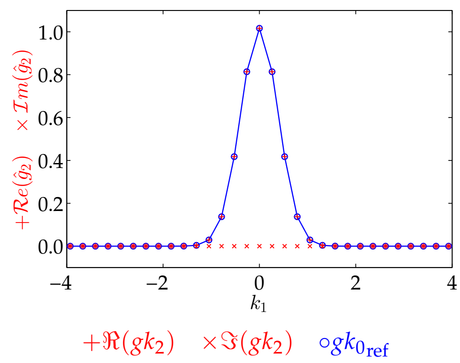
```
%.. Fourier transform of a Gaussian function g0 with nfft even
nfft = 64; nf = nfft/2;

x = linspace(-12,12,nfft+1);
nx = length(x); nx2 = (nx+1)/2;

dx = x(2)-x(1);
dk = 2*pi/(nfft*dx);

%.. analytical solution g0ref and g0kref
b = 3.;
g0ref = exp(-log(2.)*(x/b).^2);

kx = -nf*dk:dk:nf*dk;
g0kref = b/(2.*sqrt(pi*log(2.))) * exp(-(kx*b).^2/(4*log(2.)));
```



```
%.. Fourier transform
x1 = 0:dx:(nfft-1)*dx;
g1 = fftshift(g0ref(1:nfft));
```




```
kx1 = 0:dk:(nfft-1)*dk;
gk1 = fft(g1) * dx/(2*pi);

kx2 = -nf*dk:dk:(nf-1)*dk;
gk2 = fftshift(gk1);
```




```
%.. Inverse Fourier transform
gk2 = ifftshift(gk2);

gdef = ifft(gk2);
gdef = gdef *2*pi/dx;

xdef = -nf*dx:dx:(nf-1)*dx;
gdef = fftshift(gdef);
```