Symbolic Math in MATLAB

Declare variables symbolic and enter quadratic function.

```
syms a b c x
f=a*x^2+b*x+c
```

f = $a\,x^2 + b\,x + c$

Now you can integrate and differentiate *f*.

```
diff(f,x)
```

ans = $b + 2\,a\,x$

```
int(f,x)
```

ans =

$$\frac{a\,x^3}{3} + \frac{b\,x^2}{2} + c\,x$$

Solve to get the quadratic formula.

```
solve(f,x)
```

ans =

$$\begin{pmatrix} -\dfrac{b + \sqrt{b^2 - 4\,a\,c}}{2\,a} \\ -\dfrac{b - \sqrt{b^2 - 4\,a\,c}}{2\,a} \end{pmatrix}$$

Fourier and Laplace Transforms

Now we can find Fourier and Laplace transforms. First clear the variables, then declare new symbolic variables.

```
clear
syms t w
```

Define a function and compute its Fourier transform

```
f=exp(-abs(t))
```

```
f = e^{-|t|}
```

$f = e^{-|t|}$

```
F=fourier(f)
```

F =

$$\frac{2}{w^2 + 1}$$

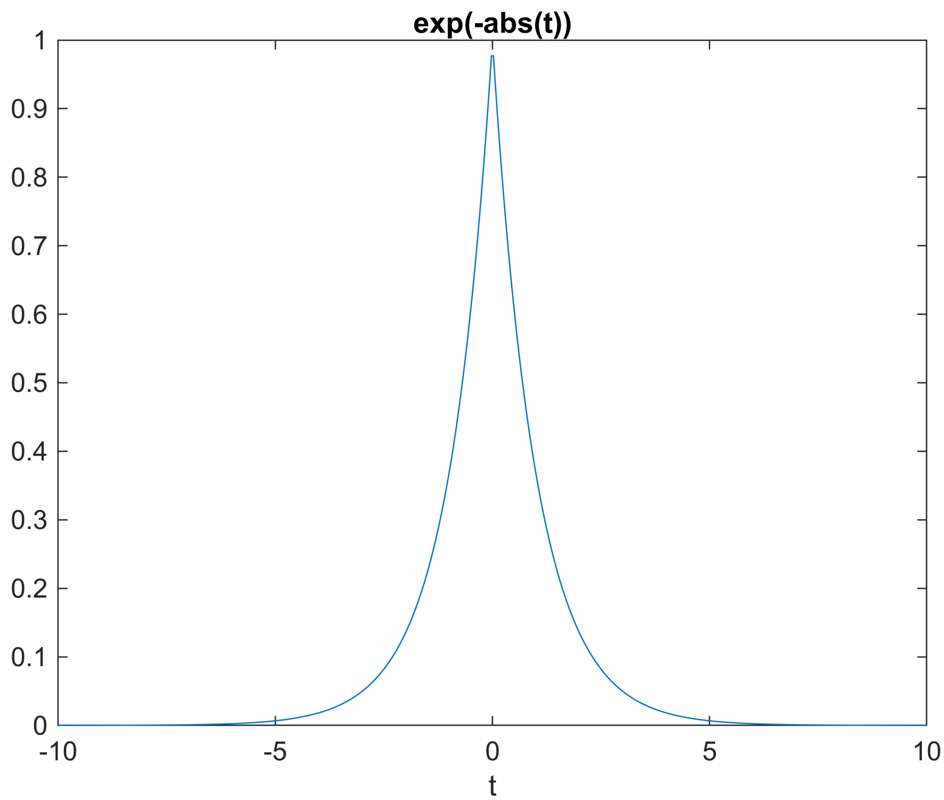
The inverse Fourier transform is just as easy.

```
ifourier(2/(w^2+1))
```

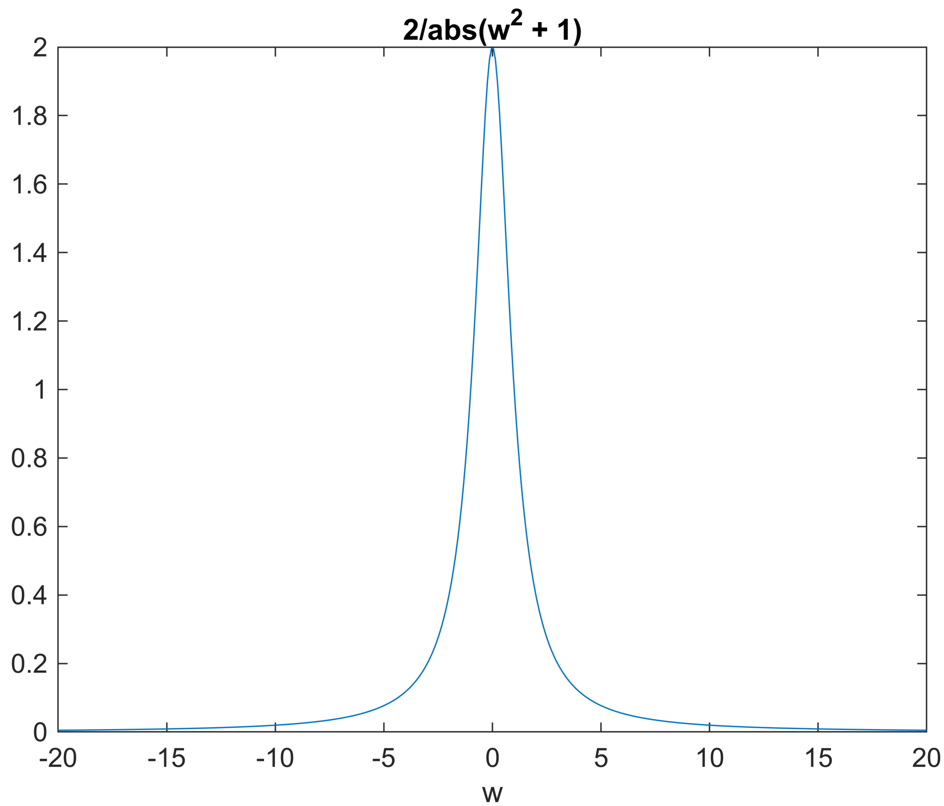
ans = $e^{-|x|}$

You can do quick plots using **ezplot**.

```
ezplot(f,[-10,10,0,1])
```

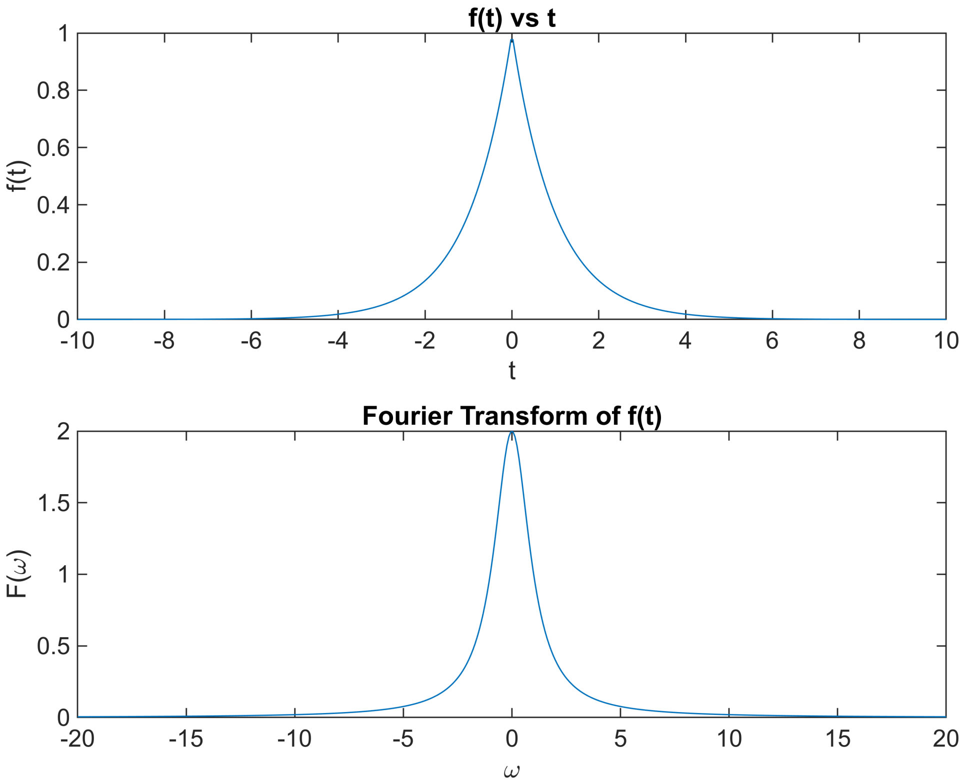



```
ezplot(abs(F),[-20,20,0,2])
```



Build and label plots.

```
clear
syms t w
f=exp(-abs(t));
F=fourier(f);
figure(1)
subplot(2,1,1)
ezplot(f,[-10,10,0,1 ])
xlabel('t')
ylabel('f(t)')
title(['f(t) vs t'] )
subplot(2,1,2)
ezplot(abs(F),[-20,20,0,2])
xlabel('\omega')
ylabel('F(\omega)')
title(['Fourier Transform of f(t)'] )
```



Laplace Transforms

Clear Workspace and try these.

```
clear
syms x y

f=1/sqrt(x);
laplace(f,x,y)
```

ans =

$$\frac{\sqrt{\pi}}{\sqrt{y}}$$

```

F=1/y^2;
ilaplace(F,y,x)
```

ans = x

Convolution

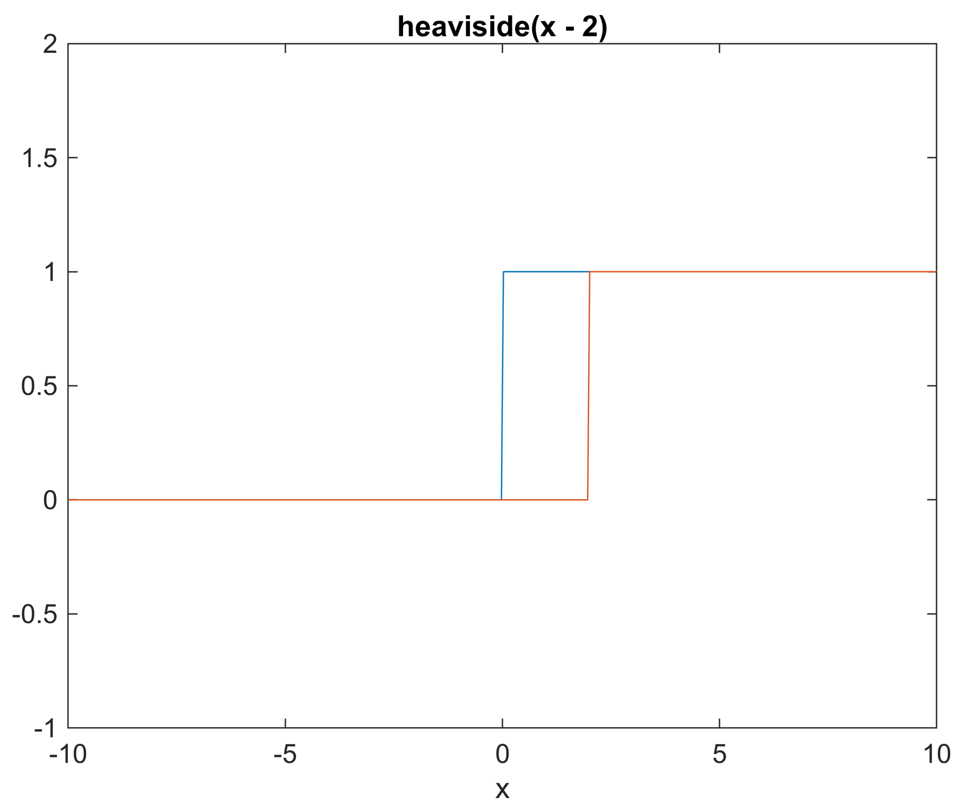
Recal the definition of convolution for Fourier transforms, $(f * g)(x) = \int_{-\infty}^{\infty} f(\xi)g(x-\xi)\,d\xi$.

```
clear
syms x xi
```

We first need the Heaviside function and a shifted version.

```
F=heaviside(x);
figure
ezplot(F,[-10,10,-1,2])

G=heaviside(x-2);
hold on
ezplot(G,[-10,10,-1,2])
hold off
```



We can create a box function by subtracting, $H(x) - H(x-a)$. We can obtain piecewise-defined functions like the trangle function, $x(H(x) - H(x-a))$.

```
f=(heaviside(x+1)-heaviside(x-1));
g=x*(heaviside(x)-heaviside(x-1));
```

Next, we write these in terms of the integration variable and carry out the convolution.

```
g2=subs(g,x,x-xi);
f2=subs(f,x,xi);

h=int(f2*g2,xi,[-10 10]);
```

Now we plot these functions. In order to use the plot command, we first need to replace the symbolic x with a double precision variable t.

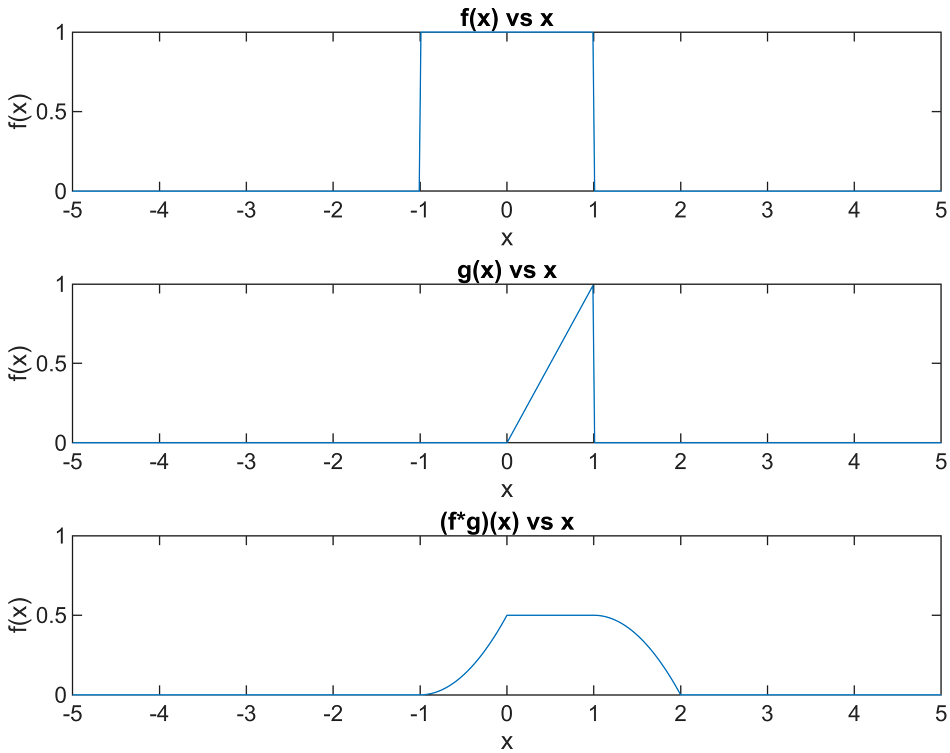
```
dt = 0.01;
t = [ -5:dt:5 ];

f1=subs(f,x,t);
f1=double(f1);
g1=subs(g,x,t);
g1=double(g1);
h1=subs(h,x,t);
h1=double(h1);

figure(1)
subplot(3,1,1)
plot(t,f1)
xlabel('x')
ylabel('f(x)')
title('f(x) vs x')

subplot(3,1,2)
plot(t,g1)
xlabel('x')
ylabel('f(x)')
title('g(x) vs x')

subplot(3,1,3)
plot(t,h1)
xlabel('x')
ylabel('f(x)')
title('(f*g)(x) vs x')
axis([-5 5 0 1])
```



Convolution Theorem

```
syms w
F=fourier(f)
```

F =

$$-\frac{-\sin(w)+\cos(w)\,\mathrm{i}}{w}+\frac{\sin(w)+\cos(w)\,\mathrm{i}}{w}$$

```
G=fourier(g)
```

G =

$$-\frac{1}{w^2}+\frac{(\cos(w)-\sin(w)\,\mathrm{i})\,\mathrm{i}}{w}-\frac{(\sin(w)+\cos(w)\,\mathrm{i})\,\mathrm{i}}{w^2}$$

```
H=F*G
```

H =

$$\left(\frac{-\sin(w)+\cos(w)\,\mathrm{i}}{w}-\frac{\sin(w)+\cos(w)\,\mathrm{i}}{w}\right)\left(\frac{1}{w^2}-\frac{(\cos(w)-\sin(w)\,\mathrm{i})\,\mathrm{i}}{w}+\frac{(\sin(w)+\cos(w)\,\mathrm{i})\,\mathrm{i}}{w^2}\right)$$

```
simplify(H)
```

ans =

$$\frac{2\sin(w)\ (\cos(w) + w\sin(w) - 1 - \sin(w)\,\mathrm{i} + w\cos(w)\,\mathrm{i})}{w^3}$$

First try the symbolic inverse Fourier transform

```
% hh=ifourier(H,w,x);  % Did not work - converted to exp.
hh=ifourier(simplify(rewrite(H, 'exp')))
```

hh =

$$-\frac{\pi\, x\,\mathrm{sign}(x) + \frac{\pi\,\mathrm{sign}(x-1)\ (x-1)^2}{2} - \frac{\pi\,\mathrm{sign}(x+1)\ (x+1)^2}{2} - \frac{\pi\,\mathrm{sign}(x-2)\ (x-2)^2}{2} + \frac{\pi\, x^2\,\mathrm{sign}(x)}{2} - \pi}{2\,\pi}$$

```
h2=subs(hh,x,t);
h2=double(h2);
```

If that does not work, one can try the discrete Fourier transform. We need to account for the differences between the **Continuous Fourier Transform** (symbolic) and the **Discrete Fourier Transform** (numerical). In this case we need to sample the frequency domain corresponding to the time steps, dt. Then, need to rescale the amplitude (proportional to 1/dt).

Also, fft/ifft assumes signal starts at t=0. So, we need to use `fftshift` and `ifftshift` to move the zero-frequency component to the center of the spectrum otherwise the result could be shifted in time.

```
N = length(t);
fs = 1/dt;
% Create w vector from -pi*fs to pi*fs
w_vec = 2*pi * ((-N/2 : N/2-1) * (fs/N));

H_num = double(subs(H, w, w_vec));
% Use ifftshift to align the frequency domain before IFFT
h_from_fft = ifftshift(ifft(fftshift(H_num))) * fs;
```

Now, plot the convolution and the results using the Convolution Theorem.

```
figure
subplot(3,1,1)
plot(t,h1)
xlabel('x')
ylabel('f(x)')
title('(f*g)(x) vs x - Convolution Integral')
axis([-5 5 0 1])

subplot(3,1,2)
plot(t,h2)
```

```
xlabel('x')
ylabel('f(x)')
title('(f*g)(x) vs x - Convolution Theorem')
axis([-5 5 0 1])

subplot(3,1,3)
plot(t,real(h_from_fft))
axis([-5 5 0 1])
xlabel('x')
ylabel('f(x)')
title('(f*g)(x) vs x - Using ifft')
```