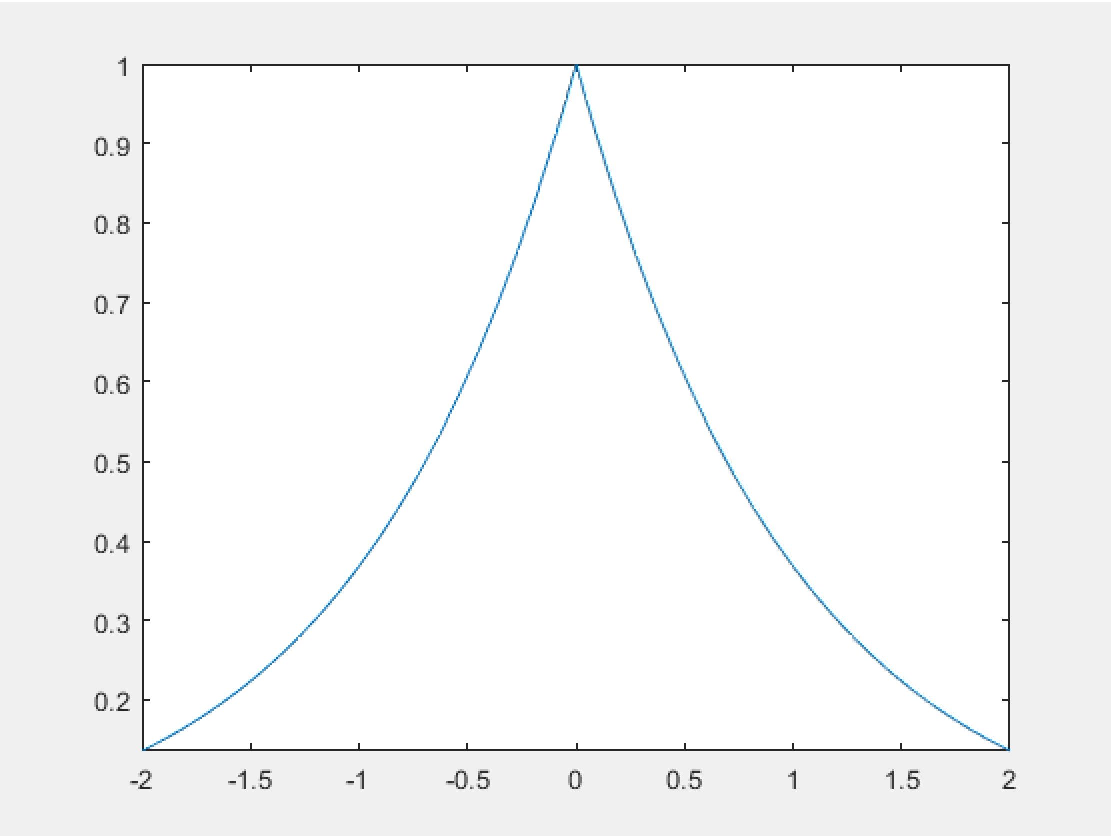
Rafi Rahman – u3191010

Lab Report

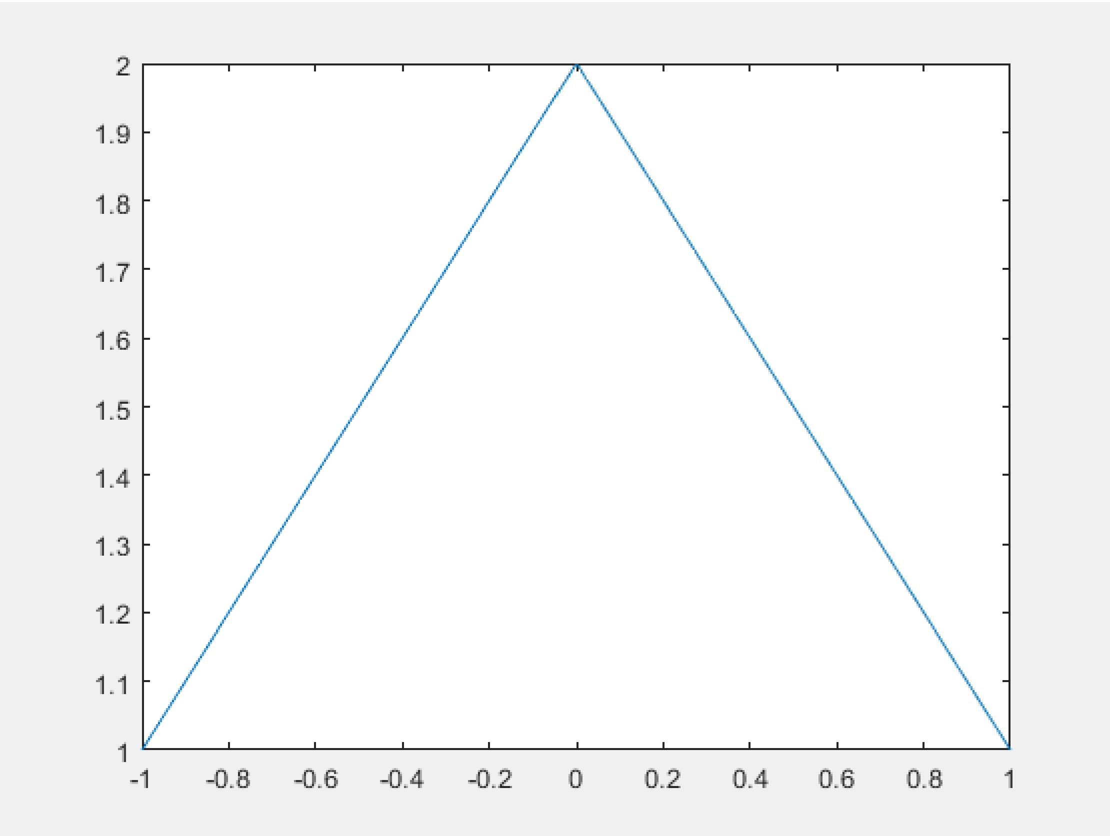
Week 9/10

```
function out = fcos(t)
    out = cos(pi*t);

>> quad('fcos',0,5)

ans =

   2.2204e-16


function out = fexp(t)
    out = exp(-abs(t));

>> quad('fexp',-2,2)

ans =

    1.7293

>> fplot(@fexp,[-2,2])
```

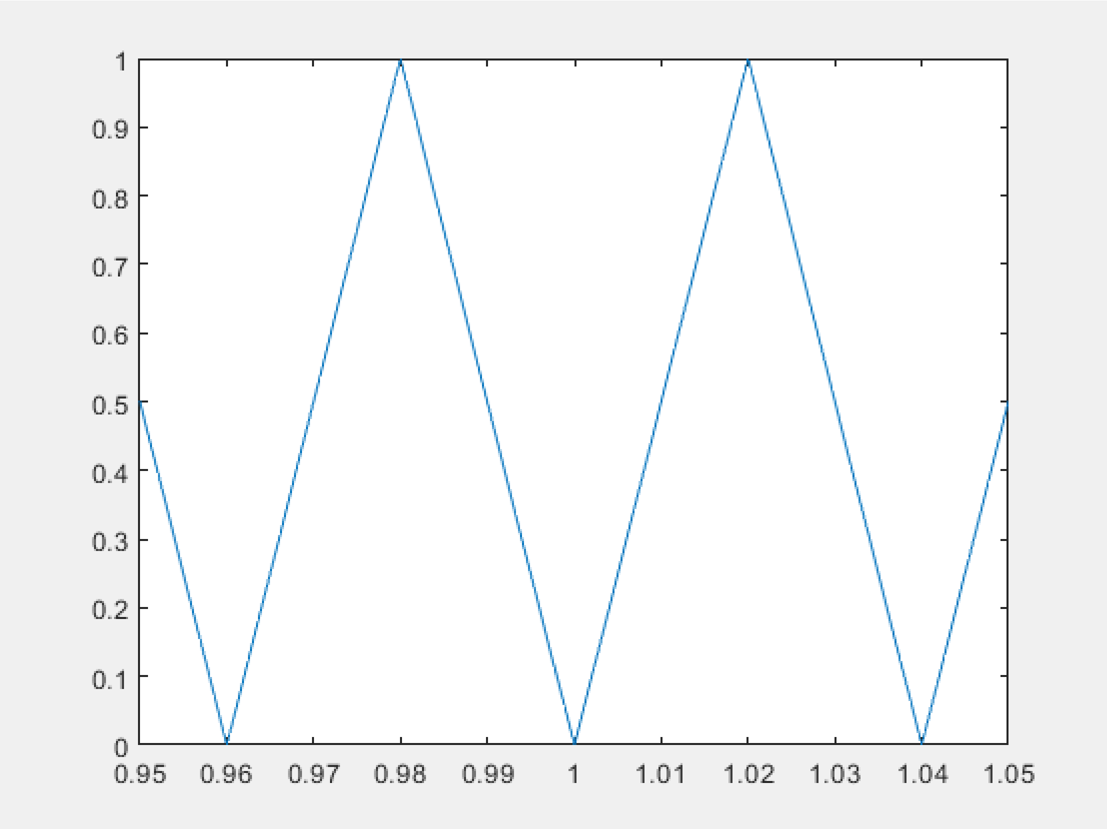



```
>> feval('fcos',0:0.25:2)
```

```
ans =

  Columns 1 through 5

    1.0000    0.7071    0.0000   -0.7071   -1.0000

  Columns 6 through 9

   -0.7071   -0.0000    0.7071    1.0000

>> quad(inline('cos(pi*t)'),0,5)

ans =

   2.2204e-16

>> fplot(inline('2 - abs(t)'),[-1,1])
```




```
>> quad(inline('2 - abs(t)'),-1,1)

ans =

     3

function out = triwave(tt)
    tt = mod(tt,0.04);
    out = (50*tt).*((tt >=0) - (tt >= 0.02)) + (2 - 50*tt).*((tt >= 0.02) -
(tt >= 0.04));

>> fplot(@triwave,[0.96,1.05])
```




```
>>
fplot(inline('A*cos(pi*alfa*t.*t)','t','A','alfa'),[0,1],200,[],[],100,13)
Error using fplot (line 149)
Invalid parameter ''.

>> quad(inline('A*cos(pi*alfa*t.*t)','t','A','alfa'),0,1,[],[],100,13)

ans =

    9.8208
```

Lab Assessable Exercises

```
function out = FSC(t)
out = 240*sin(100*pi*t);
```

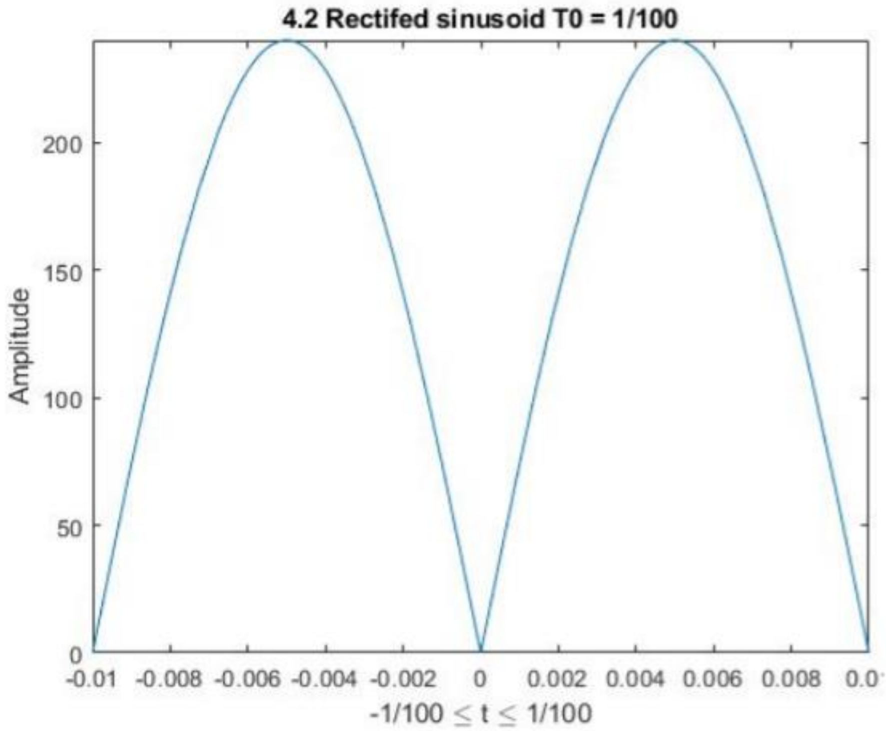
```
>> quad(@FSC, 0, 1/100)

ans =

    1.5279
```

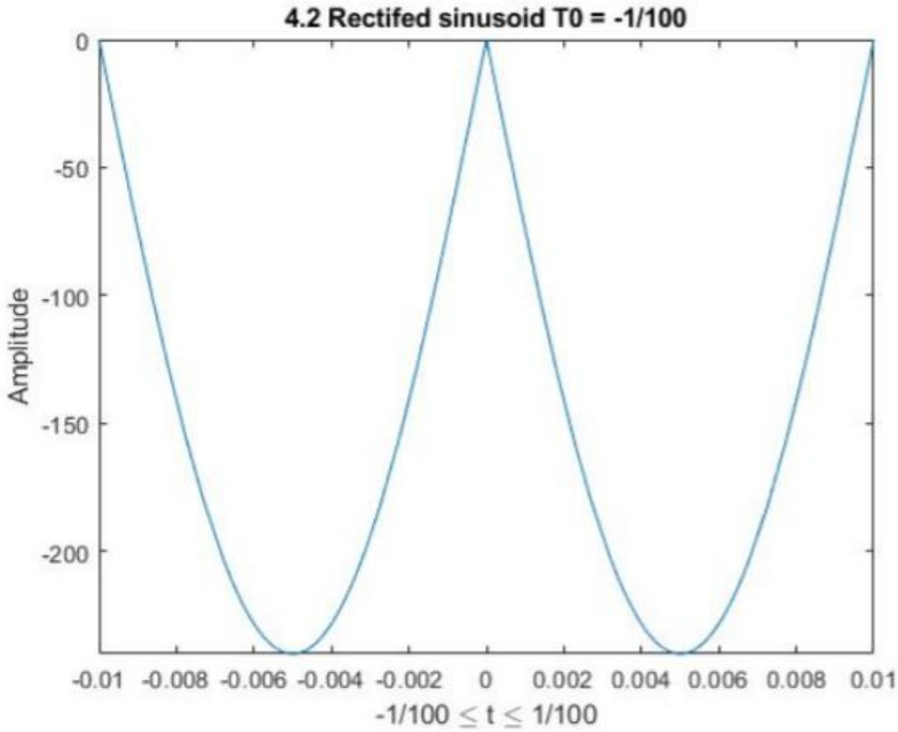
```
function [xx,tt] = synthApprox(fK, xK, fs, duration, ts)
xK=xK(:); fK=fK(:);
tt=ts:(duration/fs):ts+duration;
xx=real( (xK.')*exp((j*(2*pi*fK))*tt) );
```

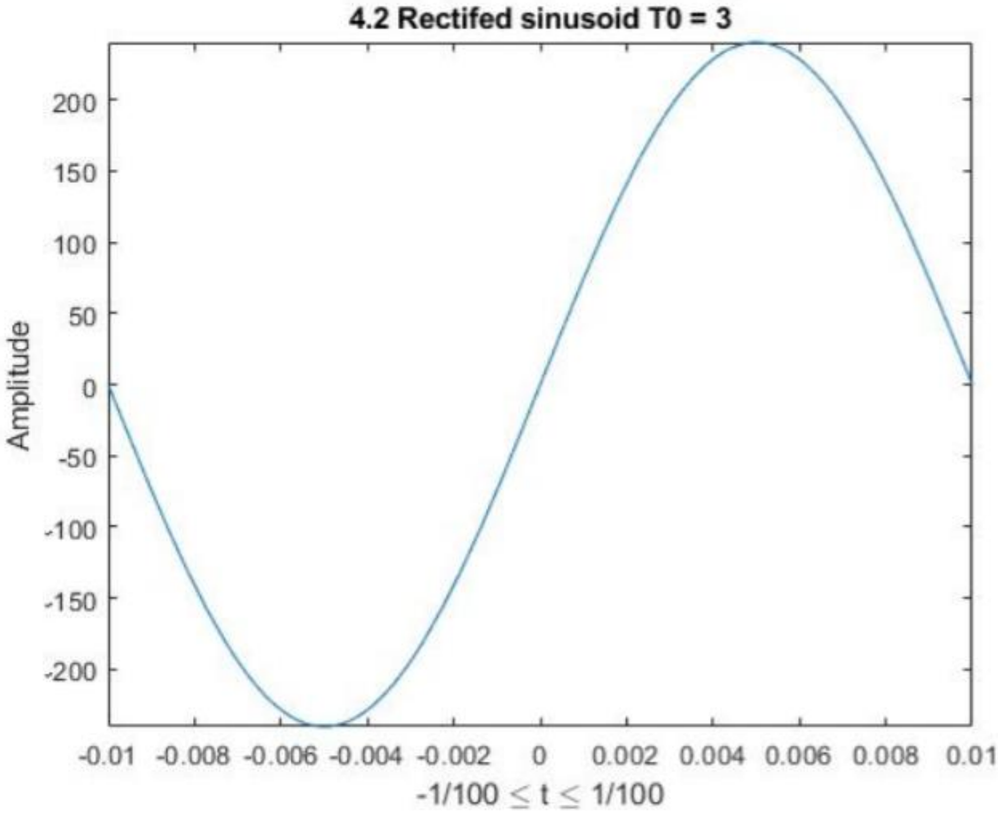
```
function        xx = syn_fourier(tt, aK, fK)
%SYN_FOURIER  Function to synthesize a sum of complex
%             exponentials over the time range given by tt
%  usage:
%    xx = syn_fourier(tt, ak, fk)
%      tt = vector of times, for the time axis
%      ak = vector of complex Fourier coefficients
%      fk = vector of frequencies
%              (usually contains both negative and positive freqs)
%      xx = vector of synthesized waveform values
%
%    Note: fk and ak must be the same length.
%             ak(1) corresponds to frequency fk(1),
%             ak(2) corresponds to frequency fk(2), etc.
%
% Note: the output might have a tiny imaginary part even if it
%       is supposed to be purely real.  If so, take the real part.
%
xx = exp( tt(:)*(2i*pi*fK(:)') )* aK(:);
if( max(abs(imag(xx)))<1e-6 ), xx = real(xx); end
size(tt)
xx = xx';
```

```
function out = FSC(t)
out = 240*sin(100*pi*t);
```

```
>> quad(inline('240*sin(100*pi*t)'), T1, T2)
```








```
fplot(@fscMOD, [-1/100 1/100]); title('4.2 Rectifed sinusoid T0 = 1/100');xlabel('-1/100 \leq t \leq 1/100');ylabel('Amplitude');
fplot(@fscMOD, [-1/100 1/100]); title('4.2 Rectifed sinusoid T0 = 3');xlabel('-1/100 \leq t \leq 1/100');ylabel('Amplitude');
fplot(@fscMOD, [-1/100 1/100]); title('4.2 Rectifed sinusoid T0 = -1/100');xlabel('-1/100 \leq t \leq 1/100');ylabel('Amplitude');
```

```
>> quad(@fscMOD, -1/100, 1/100)

ans =

   -3.0558
```

```
function out = integrandXte(t,k)
out = aK(t).*exp(-1i*2*pi*k*t/4);
```

```
function out = ak(t)
t_t = mod(t,4);
out = 3*((t_t>=0)-(t_t>=1)) - 2*((t_t>=1) - (t_t>=4));
```

```
T = 1/120;
ft = pi/T;
N = 8;
k_k = -N:N;
for k=1:length(k_k)
        wk = 2*pi*k_k(k)/T;
        integrand = inline('sin(ft*t).*exp(-j*wk*t)','t','ft','wk');
        X(k) = (1/T)*integral(integrand,0,T,[],[],ft,wk);
        Kf = k_k(k)+(k_k(k)==0);
        a(k) = T*((j*Kf*pi+1)*exp(j*Kf*pi) - 1)/(4*pi*pi*Kf*Kf);
        if k_k(k)==0, a(k)=2;end
end
tt = linspace(-T,T,1001)';
xx = real( exp(j*2*pi*tt*k_k/T)*(X.') );
```

Week 11/12

```
>> b = [1,3,1];
>> a = [1,5,2,7,3];
>> [r,p,k] = residue(b,a);
>> r

r =

  -0.0856 + 0.0000i
   0.0496 - 0.2369i
   0.0496 + 0.2369i
  -0.0135 + 0.0000i

>> p

p =

  -4.8587 + 0.0000i
   0.1441 + 1.1902i
   0.1441 - 1.1902i
  -0.4295 + 0.0000i

>> k
```

```
k =

     []

>> num = [1,0,4];
>> den = [1,4,7,15,31,75];
>> H1 = tf(num,den);
>> H1

H1 =

                  s^2 + 4
  ----------------------------------------
  s^5 + 4 s^4 + 7 s^3 + 15 s^2 + 31 s + 75

Continuous-time transfer function.

>> z = [-4];
>> p = [-3,-10];
>> k = 20;
>> H2 = zpk(z,p,k);
>> H2

H2 =

    20 (s+4)
  ------------
  (s+3) (s+10)

Continuous-time zero/pole/gain model.

>> s = tf('s')

s =

  s

Continuous-time transfer function.

>> H3 = s*(s + 3)/(s^2 + 2*s + 8)

H3 =

    s^2 + 3 s
  -------------
  s^2 + 2 s + 8

Continuous-time transfer function.

>> tf(H2)

ans =

     20 s + 80
  ---------------
  s^2 + 13 s + 30
```

```
Continuous-time transfer function.

>> zpk(H1)

ans =

                        (s^2 + 4)
  -----------------------------------------------------
  (s+3.081) (s^2 + 2.901s + 5.45) (s^2 - 1.982s + 4.467)

Continuous-time zero/pole/gain model.

>> [num,den] = tfdata(H2,'v');
>> num

num =

     0    20    80

>> den

den =

     1    13    30

>> [z,p,k] = zpkdata(H1,'v');
>> z

z =

   0.0000 + 2.0000i
   0.0000 - 2.0000i

>> p

p =

  -3.0807 + 0.0000i
  -1.4505 + 1.8291i
  -1.4505 - 1.8291i
   0.9909 + 1.8669i
   0.9909 - 1.8669i

>> k

k =

     1

>> num = [2,0,5];
>> den = [1,3,2];
>> [r,p,k] = residue(num,den)

r =

   -13
     7
```

```
p =

    -2
    -1


k =

     2

>> num = [2,7,4];
>> den = [conv(conv([1,1],[1,2]),[1,2])];
>> [r,p,k] = residue(num,den);
>> r

r =

    3.0000
    2.0000
   -1.0000

>> p

p =

   -2.0000
   -2.0000
   -1.0000

>> k

k =

     []

>> num = [8,21,19];
>> den = [conv([0,1,2],[1,1,7])];
>> [r,p,k] = residue(num,den)

r =

   3.5000 - 0.4811i
   3.5000 + 0.4811i
   1.0000 + 0.0000i


p =

  -0.5000 + 2.5981i
  -0.5000 - 2.5981i
  -2.0000 + 0.0000i


k =
```

```
     []

>> [angle,mag] = cart2pol(real(r),imag(r))

angle =

   -0.1366
    0.1366
         0


mag =

    3.5329
    3.5329
    1.0000

>> f = str2sym('sin(a*t) + cos(b*t)');
>> F = laplace(f)

F =

a/(a^2 + s^2) + s/(b^2 + s^2)

>> F = str2sym('(a*s^2)/(s^2 + b^2)');
>> f = ilaplace(F)

f =

a*dirac(t) - a*b*sin(b*t)
```

Lab Assessable Exercises

```
>> a = [1 2 -3]; b = [1 8 12];
>> [r p k] = residue(b,a)

r =

    0.7500
    5.2500


p =

   -3.0000
    1.0000


k =
```

```
>> a = [1]; b = [1 1 1];
>> [r p k] = residue(b,a)

r =

     []


p =

     []


k =

     1     1     1


>> a = [1 10]; b = [1 11 10];
>> [r p k] = residue(b,a)

r =

     0


p =

   -10


k =

     1     1


>> a = [1]; b = [1 3.236 5.235924 5.235924 3.236 1];
>> [r p k] = residue(b,a)

r =

     []


p =

     []


k =

    1.0000    3.2360    5.2359    5.2359    3.2360    1.0000
```

```
>> syms t
>> f = exp(t);
>> laplace (f)

ans =

1/(s - 1)
```

```
>> syms t
>> f = exp(2*t)*cos(200*pi*t)

f =

exp(2*t)*cos(200*pi*t)

>> laplace(f)

ans =

(s - 2)/((s - 2)^2 + 40000*pi^2)
```

```
>> syms t
>> f = ramp(t); laplace(f)

ans =

1/(2*s^2) + laplace(t*sign(t), t, s)/2
```

```
>> syms t
>> f = t*exp(t);laplace(f)

ans =

1/(s - 1)^2
```

```
>> f = (24)/(s*(s+8))

f =

24/(s*(s + 8))

>> ilaplace(f)

ans =

3 - 3*exp(-8*t)


 f =

 20/(s^2 + 4*s + 3)

 >> ilaplace(f)

 ans =

 10*exp(-t) - 10*exp(-3*t)


>> syms s
>> f = (5)/(s^2+6*s+73)

f =

5/(s^2 + 6*s + 73)

>> ilaplace(f)

ans =

(5*sin(8*t)*exp(-3*t))/8


>> syms s
>> f = (10)/(s*(s^2+6*s+73))

f =

10/(s*(s^2 + 6*s + 73))

>> ilaplace(f)

ans =

10/73 - (10*exp(-3*t)*(cos(8*t) + (3*sin(8*t))/8))/73
```