Final Exam - MSSC 6030: Spring 2020

INSTRUCTIONS: This exam is open book, notes, and MATLAB but CLOSED to the internet. Work that is erased or crossed out will not be graded. You must show your work to earn full and partial credit. You must write legibly to earn credit. **You may use your calculator or MATLAB to check your answers but you must show all work unless stated otherwise.** **As with homework, begin each problem on a new page and write the full problem statement.** **The exam is due on D2L dropbox by 12:30am on Wednesday, May 6, 2020.**

PROBLEM 1: Consider the boundary value problem $u''(x) = x - 2$, where $u(0) = 0$, and $u'(1) = 4$ for $0 \leq x \leq 1$.

(a) (5 pts) Solve the IVP by hand to find the true solution $u(x)$.

(b) (25 pts) Solve the IVP using the method of *Finite Differences*.

(i) Using $h = 0.25$, write out all steps of the solution method by hand and justifying all entries in the resulting matrix equation.

(ii) Then, write code in *Matlab* to solve your problem from (i) to determine the solution $u(x)$. Compare your result to that from (a) at the grid nodes.

(iii) Generalize your code to solve the problem with $h = \frac{1}{20}$.

(c) (25 pts) Solve the IVP using the *Finite Element Method*.

(i) Using $h = 0.25$, write out all steps of the solution method by hand and justifying all entries in the resulting matrix equation.

(ii) Then, write code in *Matlab* to solve your problem from (i) to determine the solution $u(x)$. Compare your result to that from (a) at the grid nodes.

(iii) Generalize your code to solve the problem with $h = \frac{1}{20}$.

(d) (10 pts) Compare your results from your FD and FEM to the true solution in (a). Which solution is most accurate? How are you classifying 'most accurate'? Additionally, plot the errors of each solution method compared to the true solution.

PROBLEM 2: This problem will use Fourier Sine series (using terms of the form $sin(n\pi x)$) and Haar wavelets (at most 8 terms).

(a) (20 pts) Approximate the following functions on $[0, 1]$. You may use the computer to verify your computations but you must also show the work by hand.

(i) $f(x) = x^2$

(ii) $g(x) = \begin{cases} 0 & 0 \leq x \leq \frac{1}{2} \\ 1 & \frac{1}{2} \leq x \leq 1 \end{cases}$

(b) (5 pts) Plot the function $f(x)$ and its Fourier sine series and Haar wavelet approximations on the same plot. Repeat this process for $g(x)$ in a new figure.

(c) (5 pts) Which expansion does a better job? How many terms are needed to capture the behavior? How are you classifying 'better'?

(d) (5 pts) Compare and contrast the different expansion methods Fourier sine series and Haar wavelets. Which one is most appropriate when, etc? Give strengths and weaknesses of both.

PROBLEM 3: (20 pts) Explain how the FFT is used in de-noising. Illustrate your point via a specific example (different than code I have provided for you).

PROBLEM 4: (10 pts) Explain how Taylor series can be used to determine the order of the error in numerical methods.

PROBLEM 5: (20 pts) What is *regularization* and why is it needed? Give two examples of regularization methods explaining the basics of how they work and for what type of problems they are appropriate.

PROBLEM 6: (20 pts) We have studied various solution methods for solving differential and partial differential equations. Compare and contrast *Finite Differences*, the *Finite Element Method*, *Separation of Variables*, *Monte Carlo* methods, and *Fourier Transform* methods. What types of problems can be solved by each method?

PROBLEM 7: (10 pts) Discuss how you can determine the 'accuracy' of a numerical method or solution.

PROBLEM 1: Consider the boundary value problem $u''(x) = x - 2$, where $u(0) = 0$, and $u'(1) = 4$ for $0 \le x \le 1$.

(a) (5 pts) Solve the IVP by hand to find the true solution $u(x)$.

1. a) $u''(x) = x - 2 \qquad u(0) = 0 \qquad u'(1) = 4$

$$\int u''(x)dx = \int (x-2)\,dx$$

$$u'(x) = \frac{1}{2}x^2 - 2x + C_1$$

$$\int u'(x)dx = \int \frac{1}{2}x^2 - 2x + C_1 \, dx$$

$$u(x) = \frac{1}{6}x^3 - x^2 + C_1 x + C_2$$

$u(0) = C_2 = 0 \qquad u'(1) = \frac{1}{2} - 2 + C_1 = 4$

$$C_1 = \frac{11}{2}$$

So $u(x) = \frac{1}{6}x^3 - x^2 + \frac{11}{2}x$

(b) (25 pts) Solve the IVP using the method of *Finite Differences.*

(i) Using $h = 0.25$, write out all steps of the solution method by hand and justifying all entries in the resulting matrix equation.

(ii) Then, write code in *Matlab* to solve your problem from (i) to determine the solution $u(x)$. Compare your result to that from (a) at the grid nodes.

(iii) Generalize your code to solve the problem with $h = \frac{1}{20}$.

b.ii) $h = 0.25$

$x_0 = 0$ $\qquad f(x) = x-2$

$x_1 = 0.25$

$x_2 = 0.5$

$x_3 = 0.75$

$x_4 = 1$

$$u''(x) \approx \frac{1}{h}\left[\frac{u_{i+1}-u_i}{h} - \frac{u_i - u_{i-1}}{h}\right] = f(x)$$

$$= \frac{1}{h^2}\left[u_{i+1} - 2u_i + u_{i-1}\right] = x_i \cancel{\oplus} -2$$

$$\left[u_{i+1} - 2u_i + u_{i-1}\right] = h^2(x_i \overset{-2}{\cancel{\oplus}}) \qquad (1)$$

$i=1 \qquad u_2 - 2u_1 + \overset{=0}{u_0} = h^2(x_1 \overset{-2}{\cancel{\oplus}})$ $\qquad -u_3 + u_2 = h^2(x_3+1) - 4h$

$\Uparrow$

$i=2 \qquad u_3 - 2u_2 + u_1 = h^2(x_2 \overset{-2}{\cancel{\oplus}})$ $\qquad 4h + u_3 - 2u_3 + u_2 = h^2(x_3+1)$

$\Uparrow$

$i=3 \qquad u_4 - 2u_3 + u_2 = h^2(x_3 \overset{-2}{\cancel{\oplus}})$

$u_0 = u(0) = 0 \qquad u'(1) = 4 \Rightarrow \frac{u_4 - u_3}{h} = 4 \Rightarrow u_4 = 4h + u_3$

So the matrix will be

$$\begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} h^2(x_1-2) \\ h^2(x_2-2) \\ h^2(x_3-2) - 4h \end{bmatrix}$$

$$\begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -7/64 \\ -3/32 \\ -69/64 \end{bmatrix}$$

Solve the equation

$$\Rightarrow \begin{bmatrix} -2 & 1 & 0 \\ 0 & -3 & 2 \\ 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -7/64 \\ -19/64 \\ -69/64 \end{bmatrix}$$

$\Downarrow$

$$\begin{bmatrix} -2 & 1 & 0 \\ 0 & -3 & 2 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -7/64 \\ -19/64 \\ -113/64 \end{bmatrix} \Rightarrow \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 41/32 \\ 157/64 \\ 113/32 \end{bmatrix}$$

So $u(x_1) = \frac{41}{32} \qquad u(x_2) = \frac{157}{64} \qquad u(x_3) = \frac{113}{32}$

c)

ii)Then, write code in Matlab to solve your problem from (i) to determine the solution u(x). Compare your result to that from (a) at the grid nodes.

h=0.25 and $h = \frac{1}{n+1}$

so$n = \frac{1}{h} - 1 = 3$

From the resulte above $K = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$

From equation (1)

$$[U_{i+1} - 2U_i + U_{i-1}] = h^2 f(x)$$
$$[U_{i+1} - 2U_i + U_{i-1}] = h^2 (x_i - 2)$$

To build K

From the resulte above $K = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$

if we use

```
n=3;
e = ones(n, 1);
K1 = spdiags([-e, 2*e, -e], -1:1, n, n)
```

```
K1 =
   (1,1)        2
   (2,1)       -1
   (1,2)       -1
   (2,2)        2
   (3,2)       -1
   (2,3)       -1
   (3,3)        2
```

We could findout that the sign is opposite with the function we want

So

```
e = ones(n, 1);
K2 = -spdiags([-e, 2*e, -e], -1:1, n, n)
```

```
K2 =
   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
```

And also, for i=n, from the problem

$u'(1) = 4$

$$\frac{U_{n+1} - U_n}{h} = 4$$

$$U_{n+1} - U_n = 4h$$

$$U_{n+1} = 4h + U_n$$

so the last term of the equation (1) is

$$[4h + U_n - 2U_n + U_{n-1}] = h^2(x_i - 2)$$
$$[-U_n + U_{n-1}] = h^2(x_i - 2) - 4h \qquad (2)$$

So the last term of K is -1.

And for vector b, according to equation (1)

$$b_i = h^2(x_i - 2) \qquad (3)$$

Also from (2), the last term of b is:

$$b_n = h^2(x_n - 2) - 4h \qquad (4)$$

```
clear
h = 0.25;%define h
n = 1/h-1; %compute n
u0 = 0;%boundary condition u(0)=0

x = linspace(1/(n + 1), 1 - 1/(n + 1), n);%define xi
```

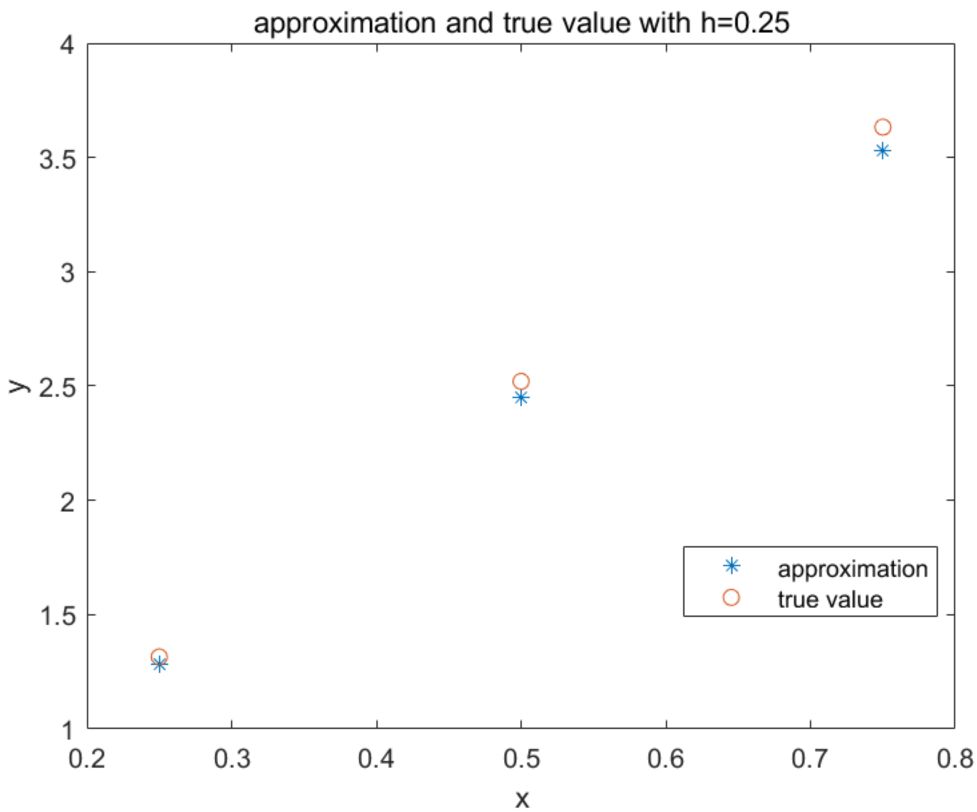
```
x = 1×3
    0.2500    0.5000    0.7500
```

```
r = realf(x); %calculate the ture value
u = finap(n, u0) %calculate the approximation value
```

```
u = 3×1
    1.2812
    2.4531
    3.5312
```

```

%plot the value
plot(x, u, '*')
hold on
plot(x, r, 'o')
xlabel('x')
ylabel('y')
title('approximation and true value with h=0.25')
legend('approximation','true value','Location','best')
hold off
```



The approximate value is a little bit off the true value

```
error_h_0_25=sum(abs(u(:)-r(:)))
```

```
error_h_0_25 = 0.2031
```

The total error of n=3 is 0.2031

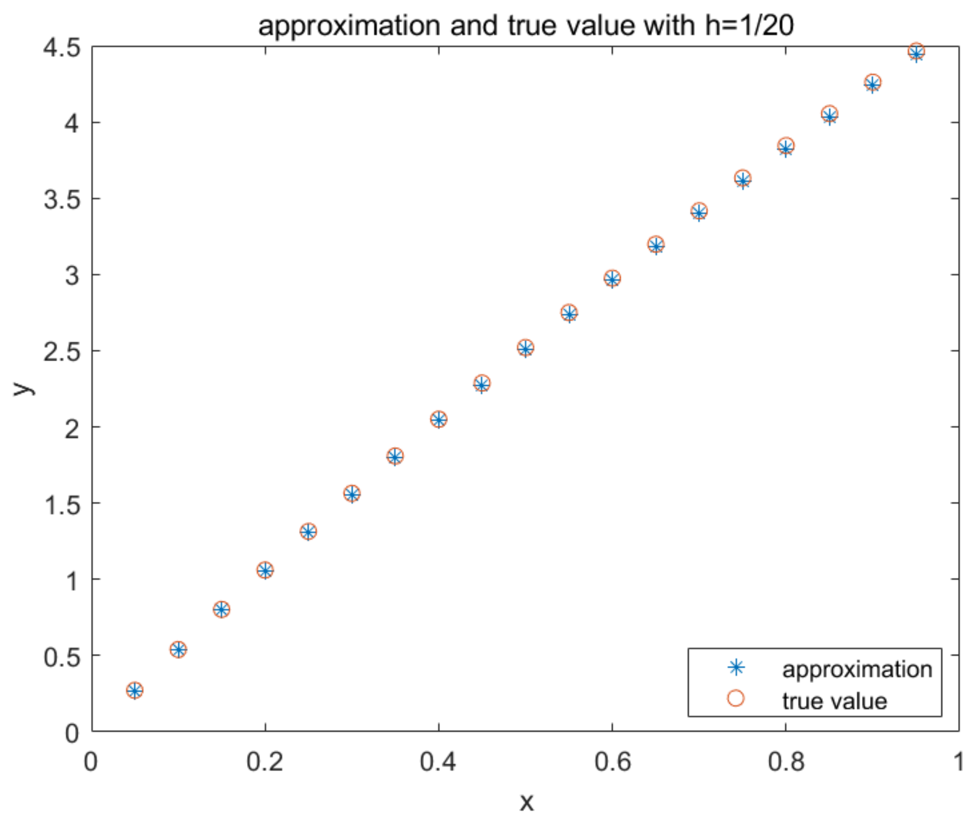
(iii) Generalize your code to solve the problem with $h = \frac{1}{20}$.

```
clear
h2 = 1/20;%define h
n2 = 1/h2-1; %compute n
u0 = 0;%boundary condition u(0)=0

x2 = linspace(1/(n2 + 1), 1 - 1/(n2 + 1), n2);%define xi
r2 = realf(x2); %calculate the ture value
u2 = finap(n2, u0); %calculate the approximation value

%plot the value
plot(x2, u2, '*')
hold on
plot(x2, r2, 'o')
xlabel('x')
ylabel('y')
title('approximation and true value with h=1/20')
legend('approximation','true value','Location','best')
```

```
hold off
```



From the plot we could find out that, at the begining of the plot, the approximate value are almost the same as the true value, but with x increasing the error become larger.

```
error_h_0_005=sum(abs(u2(:)-r2(:)))
```

```
error_h_0_005 = 0.2415
```

The total error of h=1/20 is 0.2415

```
function [y] = f(xi)
%Function of compute f(x) from du^2/d^2x=f(x)
    y = xi-2;
end

function [r]=realf(x)
%Function of compute the true value for comparison. from (a)
    r = 1/6*x.^3 - x.^2+11/2*x;
end

function [u] = finap(n, u0)
%Function of finite difference approximation
    h = 1/(n + 1); %compute h
    xi = linspace(0, 1, n + 2); %define xi
```

```
    xi = xi(2:n + 1); %only need x1 to xn
    b = f(xi)*h*h; %equation (3)
    b(end) = b(end) - 4*h; %equation (4)
    e = ones(n, 1);
    K = -spdiags([-e, 2*e, -e], -1:1, n, n);%form Kn matrix of n by n
    K(n, n) = -1;%modify Kn by changing the (n,n) element to 1
    u = K\b';%calculating

end
```

(c) (25 pts) Solve the IVP using the *Finite Element Method.*

(i) Using $h = 0.25$, write out all steps of the solution method by hand and justifying all entries in the resulting matrix equation.

(ii) Then, write code in *Matlab* to solve your problem from (i) to determine the solution $u(x)$. Compare your result to that from (a) at the grid nodes.

(iii) Generalize your code to solve the problem with $h = \frac{1}{20}$.

c). $\cancel{x}$ $u''(x) = x - 2$ $\quad u(0) = 0 \quad u'(1) = 4$

$1/4 \quad 2/4 \quad 3/4 \qquad Lu = u'' \quad f(x) = x-2$

$\langle LU_2, \varphi_j \rangle = \langle d, \varphi_j \rangle$

$\int_0^1 u'' v \, dx = \int_0^1 f v \, dx$

$\int_0^1 \frac{d}{dx}(u') v \, dx = \int_0^1 f v \, dx$

$\left[u' v \Big|_{x=0}^{x=1} - \int_0^1 u' v' \, dx \right] = \int_0^1 f v \, dx$

$u'(1)V(1) - u'(0)V(0) - \int_0^1 u'v' \, dx = \int_0^1 f v \, dx$

$4v(1) - \int_0^1 u'v' \, dx = \int_0^1 f v \, dx$

$\int_0^1 u'v' \, dx = 4v(1) - \int_0^1 f v \, dx$

From the problem we know that $f(x) = x - 2$ (1)

Since $h = \frac{1}{4}$, $n = \frac{1}{h} - 1 = 3$

So the grid point on x are $x_0 = 0, x_1 = \frac{1}{4}, x_2 = \frac{1}{2}, x_3 = \frac{3}{4}, x_4 = 1$.

The hat function will be:

$$\phi_1(x) = \begin{cases} 4x & 0 \le x \le \frac{1}{4} \\ -4x+2 & \frac{1}{4} \le x \le \frac{1}{2} \\ 0 & \frac{1}{2} \le x \le 1 \end{cases} \phi_2(x) = \begin{cases} 0 & 0 \le x \le \frac{1}{4} \\ 4x-1 & \frac{1}{4} \le x \le \frac{1}{2} \\ -4x+3 & \frac{1}{2} \le x \le \frac{3}{4} \\ 0 & \frac{3}{4} \le x \le 1 \end{cases} \phi_3(x) = \begin{cases} 0 & 0 \le x \le \frac{1}{2} \\ 4x-2 & \frac{1}{2} \le x \le \frac{3}{4} \\ -4x+4 & \frac{3}{4} \le x \le 1 \end{cases} \phi_4(x) = \begin{cases} 0 & 0 \le x \le \frac{3}{4} \\ 4x-3 & \frac{3}{4} \le x \le 1 \end{cases}$$

The graph of these three hat function are:

```
clear
Num_x_hat = 200;
hat_x = linspace(0,1,Num_x_hat);
OneFour = Num_x_hat/4;
TwoFour = Num_x_hat/2;
ThreeFour = 3*Num_x_hat/4;
%define the each stop point.
phi1 = [4*hat_x(1:OneFour) -4*hat_x(OneFour + 1:TwoFour) + 2 ...
    0*hat_x(TwoFour + 1:Num_x_hat)];
%phi1
phi2 = [0*hat_x(1:OneFour) 4*hat_x(OneFour + 1:TwoFour) - 1 ...
    -4*hat_x(TwoFour + 1:ThreeFour) + 3 0*hat_x(ThreeFour + 1:Num_x_hat)];
%phi2
phi3 = [0*hat_x(1:TwoFour) 4*hat_x(TwoFour + 1:ThreeFour) - 2 ...
    -4*hat_x(ThreeFour + 1:Num_x_hat) + 4];
%phi3
phi4 = [0*hat_x(1:ThreeFour) 4*hat_x(ThreeFour + 1:Num_x_hat)-3];
%phi4
plot(hat_x, phi1)
hold on
plot(hat_x, phi2,'-.')
plot(hat_x, phi3,'--')
plot(hat_x, phi4,'--')
legend('\phi_1','\phi_2','\phi_3','\phi_4')
hold off
```

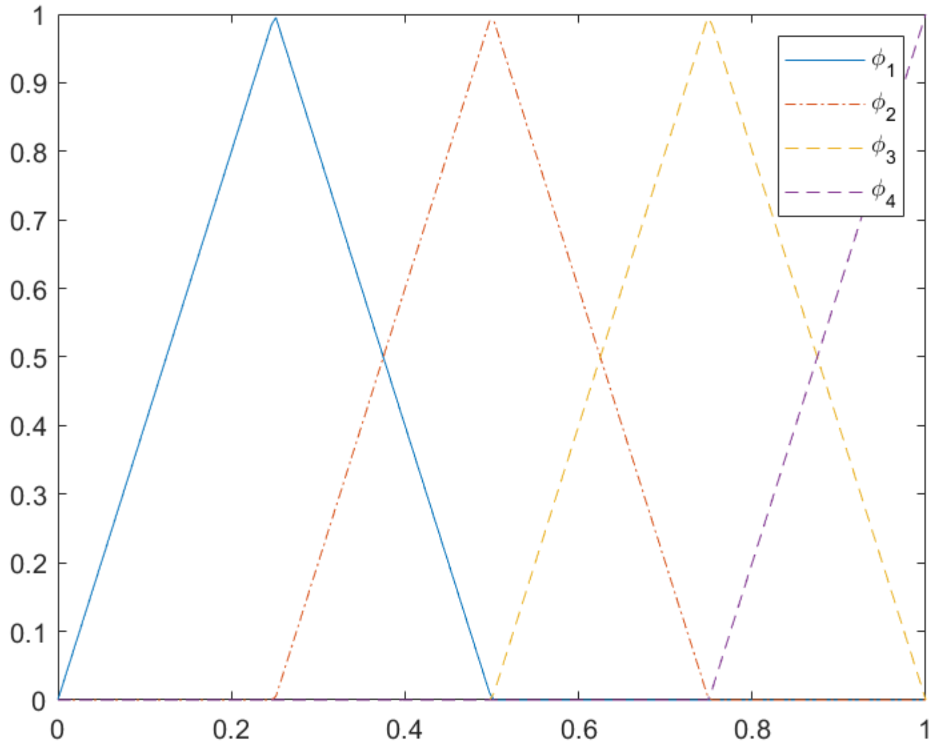


Fig 1.1

The plot shows three hat function by phi1 phi2 and phi3 in the different line style. Which phi1 is in solid line '—', phi2 is in dot line '._.' and phi3 is in '--',phi4 is half hat function with purple.

The Stiffness Martix has the formular:

$\mathrm{Ku} = b$ (2)

which $K = \begin{bmatrix} \langle \phi_1', \phi_1' \rangle & \langle \phi_2', \phi_1' \rangle & \langle \phi_3', \phi_1' \rangle & \langle \phi_4', \phi_1' \rangle \\ \langle \phi_1', \phi_2' \rangle & \langle \phi_2', \phi_2' \rangle & \langle \phi_3', \phi_2' \rangle & \langle \phi_4', \phi_2' \rangle \\ \langle \phi_1', \phi_3' \rangle & \langle \phi_2', \phi_3' \rangle & \langle \phi_3', \phi_3' \rangle & \langle \phi_4', \phi_3' \rangle \\ \langle \phi_1', \phi_4' \rangle & \langle \phi_2', \phi_4' \rangle & \langle \phi_3', \phi_4' \rangle & \langle \phi_4', \phi_4' \rangle \end{bmatrix}, u = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}, b = 4\phi(1) - \begin{bmatrix} \langle f(x), \phi_1 \rangle \\ \langle f(x), \phi_2 \rangle \\ \langle f(x), \phi_3 \rangle \\ \langle f(x), \phi_4 \rangle \end{bmatrix}$

Find the derivative of $\phi_1 \phi_2 \phi_3$:

$$\phi_1'(x) = \begin{cases} 4 & 0 \le x \le \frac{1}{4} \\ -4 & \frac{1}{4} \le x \le \frac{1}{2} \\ 0 & \frac{1}{2} \le x \le 1 \end{cases} \phi_2'(x) = \begin{cases} 0 & 0 \le x \le \frac{1}{4} \\ 4 & \frac{1}{4} \le x \le \frac{1}{2} \\ -4 & \frac{1}{2} \le x \le \frac{3}{4} \\ 0 & \frac{3}{4} \le x \le 1 \end{cases} \phi_3'(x) = \begin{cases} 0 & 0 \le x \le \frac{1}{2} \\ 4 & \frac{1}{2} \le x \le \frac{3}{4} \\ -4 & \frac{3}{4} \le x \le 1 \end{cases} \quad \phi_4'(x) = \begin{cases} 0 & 0 \le x \le \frac{3}{4} \\ 4 & \frac{3}{4} \le x \le 1 \end{cases}$$

(3)

Then compute (2) with (3) :

$$\langle \phi_1', \phi_1' \rangle = \int_0^1 (\phi_1'(x))^2 \mathrm{dx} = \int_0^{\frac{1}{4}} 4^2 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} (-4)^2 \mathrm{dx} + \int_{\frac{1}{2}}^{1} 0 \mathrm{dx} = 8$$

$$\langle \phi_2', \phi_2' \rangle = \int_0^1 (\phi_2'(x))^2 \mathrm{dx} = \int_0^{\frac{1}{4}} 0 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} 4^2 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} (-4)^2 \mathrm{dx} + \int_{\frac{3}{4}}^{1} 0 \mathrm{dx} = 8$$

$$\langle \phi_3', \phi_3' \rangle = \int_0^1 (\phi_3'(x))^2 \mathrm{dx} = \int_0^{\frac{1}{2}} 0 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} (4)^2 \mathrm{dx} + \int_{\frac{3}{4}}^{1} (-4)^2 \mathrm{dx} = 8$$

$$\langle \phi_4', \phi_4' \rangle = \int_0^1 (\phi_4'(x))^2 \mathrm{dx} = \int_0^{\frac{3}{4}} 0 \mathrm{dx} + \int_{\frac{3}{4}}^{1} (4)^2 \mathrm{dx} = 4$$

$$\langle \phi_1', \phi_2' \rangle = \langle \phi_2', \phi_1' \rangle = \int_0^1 \phi_1'(x) \phi_2'(x) \mathrm{dx} = \int_0^{\frac{1}{4}} 0 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} -16 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} 0 \mathrm{dx} + \int_{\frac{3}{4}}^{1} 0 \mathrm{dx} = -4$$

$$\langle \phi_1', \phi_3' \rangle = \langle \phi_3', \phi_1' \rangle = \int_0^1 \phi_1'(x) \phi_3'(x) \mathrm{dx} = \int_0^{\frac{1}{4}} 0 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} 0 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} 0 \mathrm{dx} + \int_{\frac{3}{4}}^{1} 0 \mathrm{dx} = 0$$

$$\langle \phi_1', \phi_4' \rangle = \langle \phi_4', \phi_1' \rangle = \int_0^1 \phi_1'(x) \phi_4'(x) \mathrm{dx} = \int_0^{\frac{1}{4}} 0 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} 0 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} 0 \mathrm{dx} + \int_{\frac{3}{4}}^{1} 0 \mathrm{dx} = 0$$

$$\langle \phi_2', \phi_3' \rangle = \langle \phi_3', \phi_2' \rangle = \int_0^1 \phi_2'(x) \phi_3'(x) \mathrm{dx} = \int_0^{\frac{1}{4}} 0 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} 0 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} -16 \mathrm{dx} + \int_{\frac{3}{4}}^{1} 0 \mathrm{dx} = -4$$

$$\langle \phi_2', \phi_4' \rangle = \langle \phi_4', \phi_2' \rangle = \int_0^1 \phi_2'(x) \phi_4'(x) \mathrm{dx} = \int_0^{\frac{1}{4}} 0 \mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} 0 \mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} 0 \mathrm{dx} + \int_{\frac{3}{4}}^{1} 0 \mathrm{dx} = 0$$

$$\langle \phi_3', \phi_4' \rangle = \langle \phi_4', \phi_3' \rangle = \int_0^1 \phi_3'(x)\phi_4'(x)\mathrm{dx} = \int_0^{\frac{1}{4}} 0\mathrm{dx} + \int_{\frac{1}{4}}^{\frac{1}{2}} 0\mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} 0\mathrm{dx} + \int_{\frac{3}{4}}^{1} -16\mathrm{dx} = -4$$

So $K = \begin{bmatrix} 8 & -4 & 0 & 0 \\ -4 & 8 & -4 & 0 \\ 0 & -4 & 8 & -4 \\ 0 & 0 & -4 & 4 \end{bmatrix}$ (4)

Then compute b with (1):

$$\langle f(x), \phi_1 \rangle = \int_0^1 f(x)\phi_1(x) = \int_0^{\frac{1}{4}} (x-2)4x\mathrm{dx} + \int_{\frac{1}{4}}^{\frac{2}{4}} (x-2)(-4x+2)\mathrm{dx} + \int_{\frac{1}{2}}^{1} 0\mathrm{dx} = -\frac{7}{16}$$

$$\langle f(x), \phi_2 \rangle = \int_0^1 f(x)\phi_2(x) = \int_0^{\frac{1}{4}} 0\mathrm{dx} + \int_{\frac{1}{4}}^{\frac{2}{4}} (x-2)(4x-1)\mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} (x-2)(-4x+3)\mathrm{dx} + \int_{\frac{3}{4}}^{1} 0\mathrm{dx} = -\frac{3}{8}$$

$$\langle f(x), \phi_3 \rangle = \int_0^1 f(x)\phi_3(x) = \int_0^{\frac{1}{2}} 0\mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} (x-2)(4x-2)\mathrm{dx} + \int_{\frac{3}{4}}^{1} (x-2)(-4x+4)\mathrm{dx} = -\frac{5}{16}$$

$$\langle f(x), \phi_4 \rangle = \int_0^1 f(x)\phi_4(x) = \int_0^{\frac{1}{2}} 0\mathrm{dx} + \int_{\frac{1}{2}}^{\frac{3}{4}} 0\mathrm{dx} + \int_{\frac{3}{4}}^{1} (x-2)(4x-3)\mathrm{dx} = -\frac{13}{96}$$

so $b = 4\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} -\frac{7}{16} \\ -\frac{3}{8} \\ -\frac{5}{16} \\ -\frac{13}{96} \end{bmatrix} = \begin{bmatrix} \frac{7}{16} \\ \frac{3}{8} \\ \frac{5}{16} \\ \frac{397}{96} \end{bmatrix}$ (5)

So we could compute u from $\mathrm{Ku} = b$

$$\begin{bmatrix} 8 & -4 & 0 & 0 \\ -4 & 8 & -4 & 0 \\ 0 & -4 & 8 & -4 \\ 0 & 0 & -4 & 4 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} \frac{7}{16} \\ \frac{3}{8} \\ \frac{5}{16} \\ \frac{397}{96} \end{bmatrix}$$

c)ii

(ii) Then, write code in Matlab to solve your problem from (i) to determine the solution u(x). Compare your result to that from (a) at the grid nodes.

```
h = 1/4;%define h
n = 1/h - 1;%compute n
x = linspace(0, 1, 100);%define x
xg = linspace(0,1,n + 2);%true x with 5 points
up = realf(xg);%true value of the grid
u = realf(x);%true solution
b = [7/16; 3/8; 5/16; 397/96];%equation (5)
k = [8 -4 0 0;
    -4 8 -4 0;
     0 -4 8 -4;
     0 0 -4 4;];%equaiton (4)
coe = k\b;%solve equation (2)
coe = [0;coe]
```

```
coe = 5×1
         0
    1.3151
    2.5208
    3.6328
    4.6667
```

The solution with h=0.25 is above.

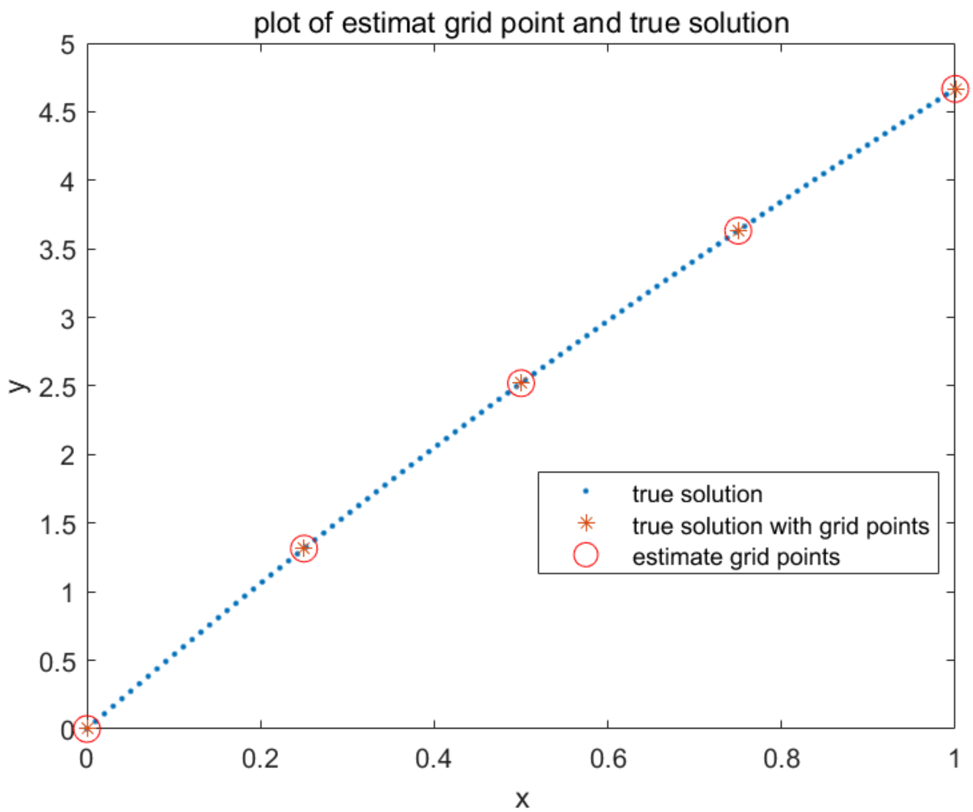
```
error=abs(coe(:) - up(:))%compute the error
```

```
error = 5×1
10^-14 ×
         0
    0.0888
    0.2220
    0.2220
    0.1776
```

Also we could see the error of the result of grid we compute are small enough.

So if we graph the solution and true value on the same page.

```
%plot
plot(x, u, '.')
hold on
plot(xg, up, '*')
plot(xg, coe, 'o', 'Markersize', 10, 'Color','r')
legend('true solution', 'true solution with grid points', 'estimate grid points','Location','be
xlabel('x')
ylabel('y')
title('plot of estimat grid point and true solution')
hold off
```




```
function [r]=realf(x)
%Function of compute the true value for comparison. from (a)
    r = 1/6*x.^3 - x.^2+11/2*x;
end
```

Fig 1.2 This plot shows the result we compute by FEM compare with true solution. On the graph they on the same spot.

c)

(iii) Generalize your code to solve the problem with $h = \frac{1}{20}$

$n = \frac{1}{h} - 1 = 19$

For matrix K, it can be determine by generate (1/h)*Kn matrix

For vector F:

$F = 4v(1) - \int_0^1 \mathrm{fvdx}$ (1)

With $h = \frac{1}{20}$, there will be 19 hat function and 1 half hat function

So $4v(1)$ will be 0 except the last term.

For each hat function, suppose that setting the center of each hat function as nh, n=1,2,...19

Three point are needed:

$((n-1)h, 0)$
$(\mathrm{nh}, 1)$
$((n+1)h, 0)$

For function $y = \mathrm{mx} + b$ at the left side of hat function

$0 = m(n-1)h + b$

$1 = \mathrm{mnh} + b$

$\mathrm{mnh} - m(n-1)h = 1$
$\mathrm{mnh} - \mathrm{mhn} + \mathrm{mh} = 1$
$\mathrm{mh} = 1$
$m = \frac{1}{h}$

$1 = \frac{1}{h}\mathrm{nh} + b$
$1 = n + b$
$b = 1 - n$

$y_1 = \frac{1}{h}x + 1 - n$

For function $y = \mathrm{mx} + b$ at the right side of hat function

$1 = \text{mnh} + b$

$0 = m(n+1)h + b$

$\text{mhn} - m(n+1)h = 1$
$\text{mhn} - \text{mhn} - \text{mh} = 1$
$\text{mh} = -1$
$m = -\frac{1}{h}$

$1 = -\frac{1}{h}\text{hh} + b$
$1 = -n + b$
$b = 1 + n$

$y_2 = -\frac{1}{h}x + 1 + n$

Therefore

$$\int_0^1 \text{fvdx} = \int_{(n-1)h}^{\text{nh}} (x-2)\left(\frac{1}{h}x + 1 - n\right)\text{dx} + \int_{\text{nh}}^{(n+1)h} (x-2)\left(-\frac{1}{h}x + 1 + n\right)\text{dx}$$
$$= \frac{h(3\text{hn} - h - 6)}{6} + \frac{h(3\text{hn} + h - 6)}{6}$$
$$= \frac{h(3\text{hn} - h - 6) + h(3\text{hn} + h - 6)}{6} \tag{2}$$
$$= \frac{h(3\text{hn} - h - 6 + 3\text{hn} + h - 6)}{6}$$
$$= \frac{h(3\text{hn} - 6 + 3\text{hn} - 6)}{6}$$
$$= \frac{h(6\text{hn} - 12)}{6}$$

for n=1,2,...,19

h=1/20

Also for the half hat:

$$b = \int_0^1 \text{fvdx} = \int_{(n-1)h}^{\text{nh}} (x-2)\left(\frac{1}{h}x + 1 - n\right)\text{dx}$$
$$B_{20} = \frac{h(3\text{hn} - h - 6)}{6} \tag{3}$$

So for the hat function

F vector is

$4v(1)-\int_0^1 \mathrm{fvdx}$

$$
=\begin{bmatrix}0\\0\\0\\\vdots\\1\end{bmatrix}-\begin{bmatrix}\frac{h(6h-12)}{6}\\\frac{h(6\mathrm{h}2-12)}{6}\\\frac{h(6\mathrm{h}3-12)}{6}\\\vdots\\\frac{h(6\mathrm{h}19-12)}{6}\\\frac{h(3\mathrm{h}20-h-6)}{6}\end{bmatrix} \qquad (4)
$$

```
clear

h=1/20;%define h
n=1/h-1;%compute n
e = ones(n+1, 1);

%generate Kn
K = spdiags([-e, 2*e, -e], -1:1, n+1, n+1);
K(n+1,n+1)=1;
K=K*(n+1);

%generate v
v=zeros(1,n+1);
v(end)=1;

%generate b vector 1 to 19 by equation (2)
b=zeros(1,n+1);
for i=1:n
    b(i)=h*(6*h*i-12)/6;
    %equation (2)
end
bn=n+1;
b(end)=h*(3*h*bn-h-6)/6;%equation (3)

F=4*v-b;%equation (4)
F=F';
coe = K\F;%solve u
coe = [0;coe]
```

```
coe = 21×1
         0
    0.2725
    0.5402
    0.8031
    1.0613
    1.3151
    1.5645
    1.8096
    2.0507
    2.2877
```

⋮

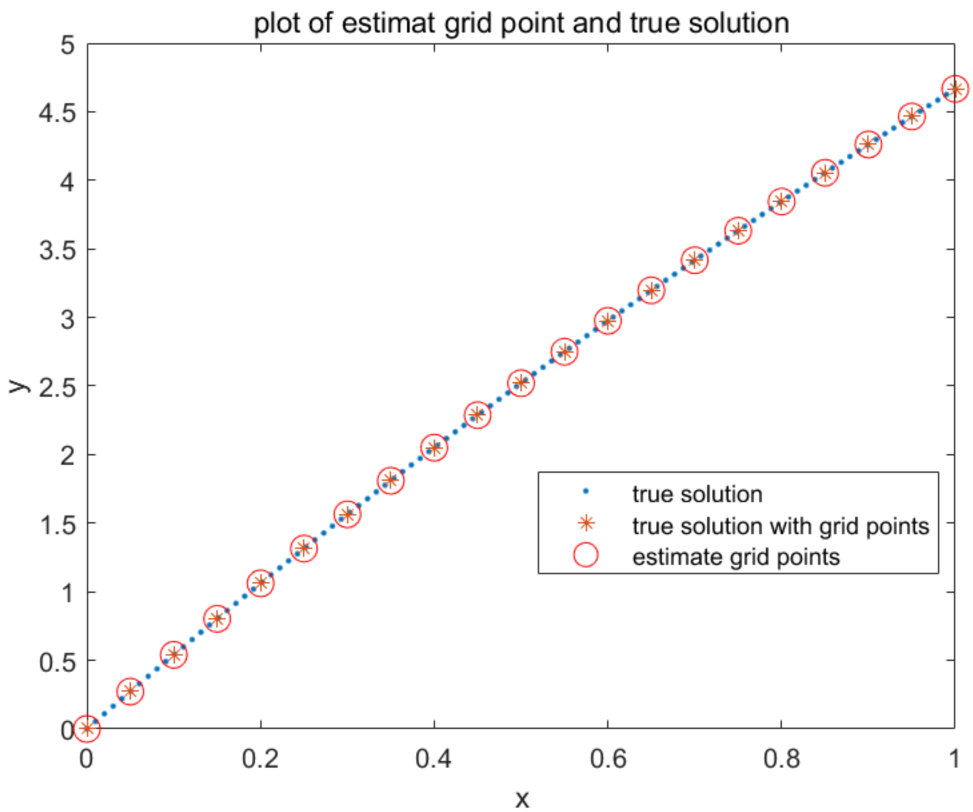
```
x = linspace(0, 1, 100);%define x
xg = linspace(0,1,n + 2);%true x with n+1 points
up = realf(xg);%true value of the grid
u = realf(x);%true solution
```

```
error=abs(coe(:) - up(:))%compute the error
```

```
error = 21×1
10^-14 ×
         0
    0.0278
    0.0555
    0.0666
    0.0888
    0.1110
    0.1332
    0.1554
    0.0888
    0.1332
      ⋮
```

error is still to the order of -14

```
%plot
plot(x, u, '.')
hold on
plot(xg, up, '*')
plot(xg, coe, 'o', 'Markersize', 10, 'Color','r')
legend('true solution', 'true solution with grid points', 'estimate grid points','Location','be
xlabel('x')
ylabel('y')
title('plot of estimat grid point and true solution')
hold off
```




```
function [r]=realf(x)
%Function of compute the true value for comparison. from (a)
    r = 1/6*x.^3 - x.^2+11/2*x;
end
```

d)Compare your results from your FD and FEM to the true solution in (a). Which solution is most accurate? How are you classifying ‘most accurate’? Additionally, plot the errors of each solution method compared to the true solution.

With $h = \frac{1}{4}$

The error of FD is 0.2031 and the error of FEM is to the order of 10^-14

With $h = \frac{1}{20}$

The error of FD is 0.2415 and the error of FEM is still to the order of 10^-14

So with total error, FEM has less total error.

So FEM has higher accuracy on the grid.

PROBLEM 2: This problem will use Fourier Sine series (using terms of the form $sin(n\pi x)$) and Haar wavelets (at most 8 terms).

(a) (20 pts) Approximate the following functions on $[0, 1]$. You may use the computer to verify your computations but you must also show the work by hand.

(i) $f(x) = x^2$

(ii) $g(x) = \begin{cases} 0 & 0 \leq x \leq \frac{1}{2} \\ 1 & \frac{1}{2} \leq x \leq 1 \end{cases}$

2. a) i) $f(x) = x^2 \quad [0,1] \quad L=1$

Fourier. $B_n = 2\int_0^1 x^2 \sin\left(\frac{n\pi x}{1}\right) dx$

$u = x^2 \quad v' = \sin(n\pi x)$

$u' = 2x \quad v = -\frac{\cos(n\pi x)}{n\pi}$

$$I = -\frac{x^2\cos(n\pi x)}{n\pi} - \int -2x\frac{\cos(n\pi x)}{n\pi}\,dx$$

$$= -\frac{x^2\cos(n\pi x)}{n\pi} + \frac{2}{n\pi}\int x\cos(n\pi x)\,dx$$

$u = x \quad v' = \cos(n\pi x)$

$u' = 1 \quad v = \frac{\sin(n\pi x)}{n\pi}$

$$= \frac{x\sin(n\pi x)}{n\pi} - \int \frac{\sin(n\pi x)}{n\pi}\,dx$$

$$-\frac{\cos(n\pi x)}{n^2\pi^2}$$

$$I = 2\left.\frac{2\pi n\, x\sin(n\pi x) + (2 - n^2\pi^2x^2)\cos(n\pi x)}{n^3\pi^3}\right|_0^1$$

$$= 2\frac{2\pi n\sin(n\pi) + (2 - \pi^2 n^2)\cos(n\pi) - 2}{n^3\pi^3}$$

So. Fourier Sine Series is

$$f(x) = \sum_{n=1}^{\infty} 2\frac{2\pi n\sin(n\pi) + (2 - n^2\pi^2)\cos(n\pi) - 2}{n^3\pi^3}\sin(n\pi x)$$

haar $h_0(t) = \begin{cases} 1 & 0<t<1 \\ 0 & \end{cases}$ $\quad h_1(t) = \begin{cases} 1 & 0<t<\frac{1}{2} \\ -1 & \frac{1}{2}<t<1 \\ 0 & \text{else} \end{cases}$

$h_2(t) = \sqrt{2}\, h_1(2t) \quad h_3(t) = \sqrt{2}\, h_1(2t-1)$

$h_4(t) = 2h_1(t) \quad h_5(t) = 2h_1(4t-1)$

$h_6(t) = 2h_1(4t-2) \quad h_7(t) = 2h_1(4t-3)$

$$f(t) = \sum_{n=0}^{7} c_n h_n(t)$$

$$c_n = \langle h_n, f(t) \rangle$$

$$c_0 = \int_0^1 x^2\,dx = \frac{1}{3}x^3\Big|_0^1 = \frac{1}{3}$$

$$c_1 = \int_0^{1/2} x^2\,dx + \int_{1/2}^{1} -x^2\,dx = \frac{1}{3}x^3\Big|_0^{1/2} + -\frac{1}{3}x^3\Big|_{1/2}^{1} = \frac{1}{24}\left(-\frac{1}{3} + \frac{1}{3}\frac{1}{8}\right)$$

$$= \frac{1}{12} - \frac{1}{3} = -\frac{3}{12}$$

$$= -\frac{1}{4}$$

$$c_2 = \int_0^{1/4} \sqrt{2}x^2\,dx + \int_{1/4}^{1/2} -\sqrt{2}x^2\,dx = \frac{1}{3}\sqrt{2}x^3\Big|_0^{1/4}$$

$$- \frac{\sqrt{2}}{3}x^3\Big|_{1/4}^{1/2} = \frac{1}{3}\sqrt{2}(1/4)^3 + \left(-\frac{\sqrt{2}}{3}\frac{1}{8} + \frac{\sqrt{2}}{3}\frac{1}{64}\right)$$

$$= \frac{-\sqrt{2}}{32}$$

$$c_3 = \int_{1/2}^{3/4} \sqrt{2}x^2\,dx + \int_{3/4}^{1} -\sqrt{2}x^2\,dx = \frac{19}{3.2^{11/2}} + \left(-\frac{37}{3.2^{11/2}}\right)$$

$$= -\frac{18}{3.2^{11/2}} = -\frac{3\sqrt{2}}{32}$$

$$c_4 = \int_0^{1/8} 2x^2 + \int_{1/8}^{1/4} -2x^2 = \frac{1}{768} + \left(-\frac{7}{768}\right) = -\frac{6}{768} = -\frac{1}{128}$$

$$c_5 = \int_{1/4}^{3/8} 2x^2 + \int_{3/8}^{1/2} -2x^2 = \frac{19}{768} - \frac{37}{768} = -\frac{18}{768} = -\frac{3}{128}$$

$$c_6 = \int_{1/2}^{5/8} 2x^2 + \int_{5/8}^{3/4} -2x^2 = \frac{61}{768} - \frac{91}{768} = -\frac{30}{768} = -\frac{5}{128}$$

$$c_7 = \int_{3/4}^{7/8} 2x^2 + \int_{7/8}^{1} -2x^2 = \frac{127}{768} - \frac{169}{768} = -\frac{42}{768} = -\frac{7}{128}$$

$$f(t) = \frac{1}{3} + \sum_{n=0}^{7} c_n h_n(t)$$

ii) $g(x) = \begin{cases} 0 & 0 \le x \le 1/2 \\ 1 & 1/2 \le x \le 1 \end{cases}$

Fourier. $B_n = 2\int_0^{1/2} 0 \, dx + 2\int_{1/2}^{1} \sin(n\pi x) dx$

$$= 2\left(\frac{-\cos(n\pi x)}{n\pi}\right)\Bigg|_{1/2}^{1}$$

$$= \frac{2\left(\cos\left(\frac{n\pi}{2}\right) - \cos(n\pi)\right)}{n\pi}$$

$$g(x) = \sum_{n=1}^{\infty} \frac{2\left(\cos\left(\frac{n\pi}{2}\right) - \cos(n\pi)\right)}{n\pi} \sin(n\pi x)$$

Haar. $C_n = \langle h_n, g(t) \rangle$

$C_0 = \int_{1/2}^{1} 1 \, dt = t\Big|_{1/2}^{1} = \frac{1}{2}$

$C_1 = \int_{1/2}^{1} -1 \, dt = -t\Big|_{1/2}^{1} = -1 - (-\frac{1}{2}) = -\frac{1}{2}$

$C_2 = 0 \qquad C_3 = \int_{1/2}^{3/4} \sqrt{2} \, dt + \int_{3/4}^{1} -\sqrt{2} \, dt = \sqrt{2}(\frac{1}{4} - \frac{1}{4}) = 0$

$C_4 = 0 \qquad C_5 = 0$

$C_6 = \int_{1/2}^{5/8} 2 + \int_{5/8}^{3/4} -2 = 2(\frac{1}{8} - \frac{1}{8}) = 0$

$C_7 = \int_{3/4}^{7/8} 2 + \int_{7/8}^{1} -2 = 2(\frac{1}{8} - \frac{1}{8}) = 0$

$g(t) = \sum_{n=0}^{7} C_n h_n(t) = \frac{1}{2} + (-\frac{1}{2}) h_1(t)$

b) Plot the function f(x) and its Fourier sine series and Haar wavelet approximations on the same plot. Repeat this process for g(x) in a new figure.

c) Which expansion does a better job? How many terms are needed to capture the behavior? How are you classifying 'better'?

d) Compare and contrast the different expansion methods Fourier sine series and Haar wavelets. Which one is most appropriate when, etc? Give strengths and weaknesses of both.

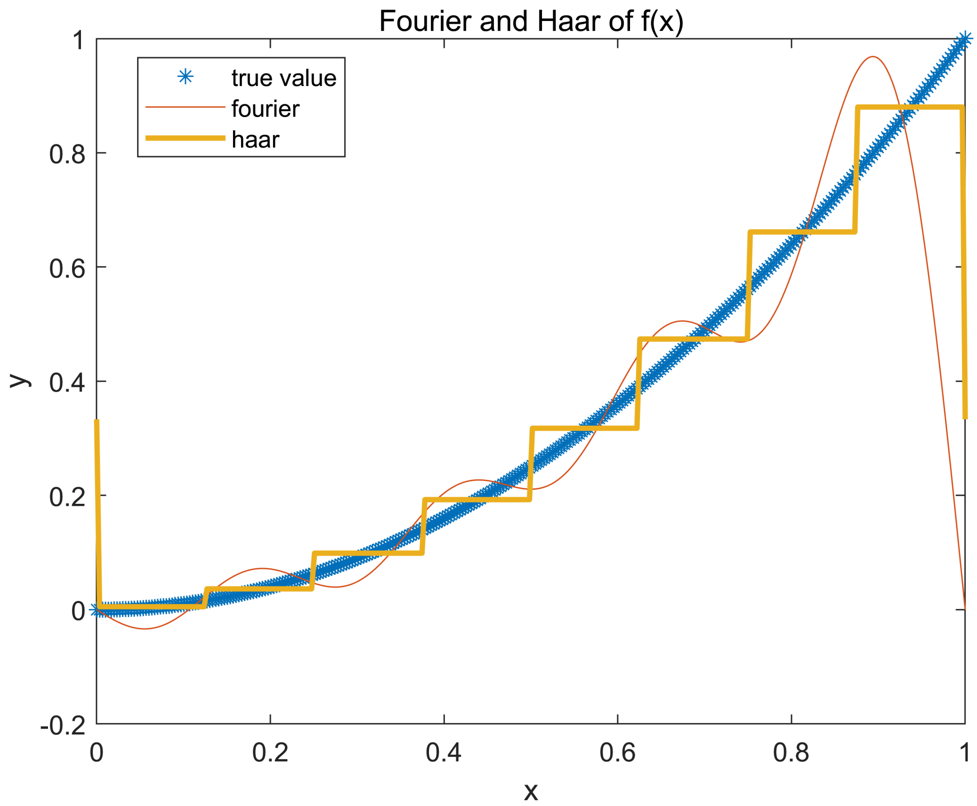
In order compare the behavior of two method I will control the term that these method will have.

Since haar only has 8 term, Fourier will also get 8 term.

i)f(x)

```
clear
n=8;%set n
num=300;
x=linspace(0,1,num);%define x
c1=[1/3 -1/4 -sqrt(2)/32 -3*sqrt(2)/32 -1/128 -3/128 -5/128 -7/128];
%define c0 to c7
[yreal1]=f(x); %f(x)
[yfourier1]=fs1(x,n);%fourier value
%haar value
[yhaar1]=c1(1)*h0(x)+c1(2)*h1(x)+c1(3)*h2(x)+c1(4)*h3(x)+c1(5)*h4(x)+c1(6)*h5(x)+c1(7)*h6(x)+c1

plot(x,yreal1,'*')
hold on
plot(x,yfourier1)
plot(x,yhaar1,'LineWidth',2)
xlabel('x')
ylabel('y')
title('Fourier and Haar of f(x)')
legend('true value','fourier','haar','Location','best')
hold off
```



From the plot: Fourier seems done better job than Haar,

In order to classify them, I will compute the variance of the difference of the expansion value and real value

if the var(difference) is high which mean the method works not good.

```
%variance of the difference of the expansion value and real value
VarHaar=var(yhaar1-yreal1)
```

```
VarHaar = 0.0036
```

```
VarFouier=var(yfourier1-yreal1)
```

```
VarFouier = 0.0247
```

from the result above, haar has less variance than fourier, due to the Gibbs phonomenon, Fouirer will have bad behavior at the edge of the function.

And if we only take some part of the function

```
p=0.97;%take 97% of the data
VarHaar97=var(yhaar1(1:floor(num*p))-yreal1(1:floor(num*p)))
```

```
VarHaar97 = 0.0019
```

```
VarFouier97=var(yfourier1(1:floor(num*p))-yreal1(1:floor(num*p)))
```

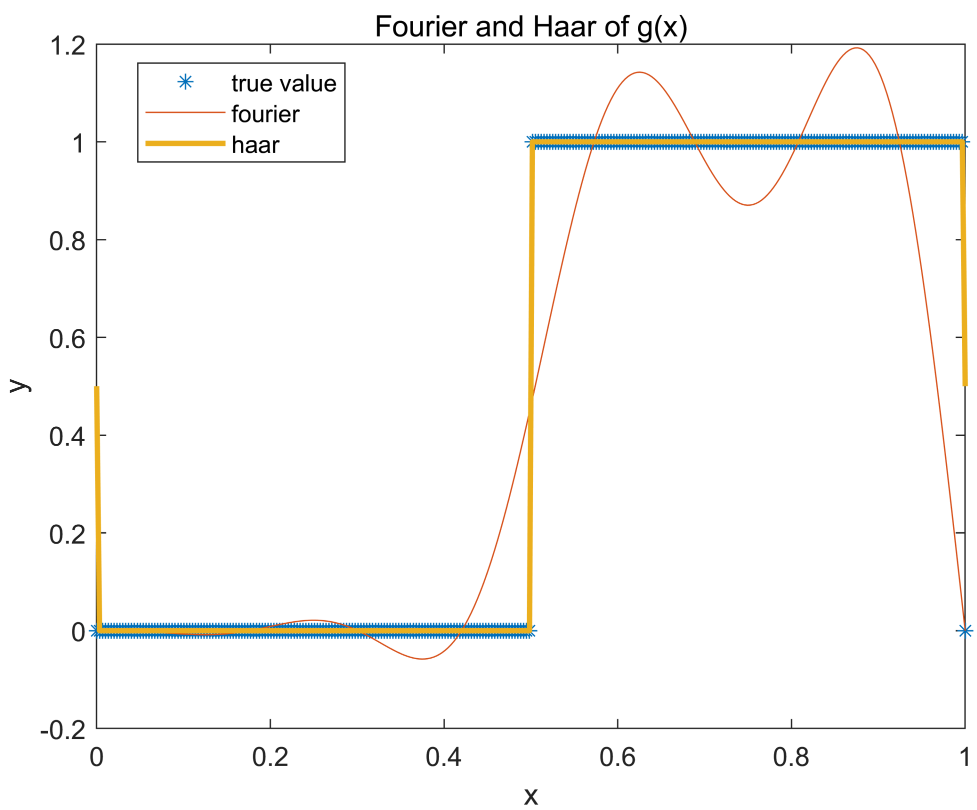
```
VarFouier97 = 0.0069
```

From the result above, with the 97% of the data, fourier will do better than Haar.

ii)g(x)

```
c2=[1/2 -1/2 0 0 0 0 0 0];
[yreal2]=g(x);
[yfourier2]=fs2(x,n);
[yhaar2]=c2(1)*h0(x)+c2(2)*h1(x)+c2(3)*h2(x)+c2(4)*h3(x)+c2(5)*h4(x)+c2(6)*h5(x)+c2(7)*h6(x)+c2

plot(x,yreal2,'*')
hold on
plot(x,yfourier2)
plot(x,yhaar2,'LineWidth',2)
xlabel('x')
ylabel('y')
title('Fourier and Haar of g(x)')
legend('true value','fourier','haar','Location','best')
hold off
```



From the plot above, haar are almost coincide with the real function.

Fourier works badly since the term are less.

```
%variance of the difference of the expansion value and real value
VarHaar2=var(yhaar2-yreal2)
```

```
VarHaar2 = 0.0017
```

```
VarFouier2=var(yfourier2-yreal2)
```

```
VarFouier2 = 0.0352
```

With the result of variance

variance the haar method are more less than the fourier

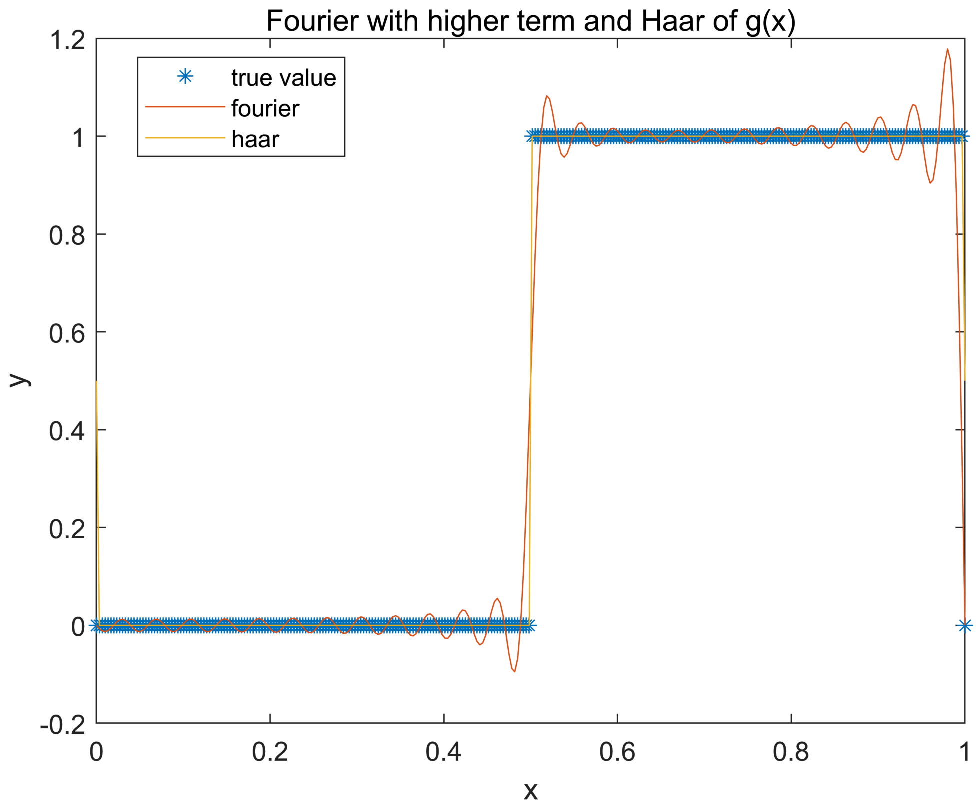
So if we increase the term fourier

```
nt=50
```

```
nt = 50
```

```
[yfouriert]=fs2(x,nt);

plot(x,yreal2,'*')
hold on
plot(x,yfouriert)
plot(x,yhaar2)
xlabel('x')
ylabel('y')
title('Fourier with higher term and Haar of g(x)')
legend('true value','fourier','haar','Location','best')
hold off
```



from the plot the fourier seem works better than before

```
VarHaar2new=var(yhaar2-yreal2)
```

```
VarHaar2new = 0.0017
```

```
VarFouier50term=var(yfouriert-yreal2)
```

```
VarFouier50term = 0.0044
```

Then from the result of the variance, we could see the even fourier has 50 term, it still has less accurate with Haar.

c)

From the case above

Fourier will works better with the curves function, but behave bad when a function has lots of jump point since the Gibbs phonomenon.

Haar will works better with the striate line function especially with the function has lots of jump point, but behave badly when deal with the curves function

```
function [y]=fs1(x,n)
%fourier transfrom of f(x)
    y=0;
```

```
    for i=1:n
        y=y+2*(2*pi*i*sin(i*pi)+(2-i^2*pi^2)*cos(i*pi)-2)/(i^3*pi^3)*sin(i*pi*x);
    end

end

function [y]=fs2(x,n)
%fourier transfrom of g(x)
    y=0;
    for i=1:n
        y=y+2*(cos(i*pi/2)-cos(i*pi))/(i*pi)*sin(i*pi*x);
    end
end

function [y]=f(x)
%f(x)
    y=x.^2;
end

function [y]=g(x)
%g(x)
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 0<x(i) && x(i)<1/2
            y(i)=0;
        elseif 1/2<x(i) && x(i)<1
            y(i)=1;
        end
    end
end

function [y]=h0(x)
%h0 function of haar
    y=0*x+1;
end

function [y]=h1(x)
%h1 function of haar
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 0<x(i) && x(i)<1/2
            y(i)=1;
        elseif 1/2<x(i) && x(i)<1
            y(i)=-1;
        end
    end
end

function [y]=h2(x)
%h2 function of haar
    l=length(x);
    y=zeros(1,l);
```

```
    for i=1:l
        if 0<x(i) && x(i)<1/4
            y(i)=sqrt(2);
        elseif 1/4<x(i) && x(i)<1/2
            y(i)=-sqrt(2);
        end
    end
end

function [y]=h3(x)
%h3 function of haar
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 1/2<x(i) && x(i)<3/4
            y(i)=sqrt(2);
        elseif 3/4<x(i) && x(i)<1
            y(i)=-sqrt(2);
        end
    end
end

function [y]=h4(x)
%h4 function of haar
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 0<x(i) && x(i)<1/8
            y(i)=2;
        elseif 1/8<x(i) && x(i)<1/4
            y(i)=-2;
        end
    end
end

function [y]=h5(x)
%h5 function of haar
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 1/4<x(i) && x(i)<3/8
            y(i)=2;
        elseif 3/8<x(i) && x(i)<1/2
            y(i)=-2;
        end
    end
end

function [y]=h6(x)
%h6 function of haar
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 1/2<x(i) && x(i)<5/8
```

```
            y(i)=2;
        elseif 5/8<x(i) && x(i)<3/4
            y(i)=-2;
        end
    end
end

function [y]=h7(x)
%h7 function of haar
    l=length(x);
    y=zeros(1,l);
    for i=1:l
        if 3/4<x(i) && x(i)<7/8
            y(i)=2;
        elseif 7/8<x(i) && x(i)<1
            y(i)=-2;
        end
    end
end
```

3. Explain how the FFT is used in de-noising. Illustrate your point via a specific example (different than code I have provided for you).

By taking the DFT of a noised signal f ,

$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \\ \vdots \\ \hat{f}_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \omega_N^3 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \omega_N^6 & \cdots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N & \omega_N & \cdots & \omega_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \omega_N^{3(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-1} \end{bmatrix}$$

$\hat{f}$ is a complex vector

In order to get norm we will take $\hat{f}$ multipy its conjugate

$$\text{spectrum} = \frac{\hat{f} \cdot \overline{\hat{f}}}{N}$$

Then we will get spectrum of f

Then we need to set up a threshold and apply to the spectrum we have.

Then the spectrum that less than the threshold will be filtered.

Then we can filter out the noise signal according to the threshold and spectrum we got before.

Then take the iDFT to the filtered back the signal.

For example

```
clear
dx=0.001;
x=0:dx:1;
```

Take the sum of cosine with three frequency

33 66 and 100

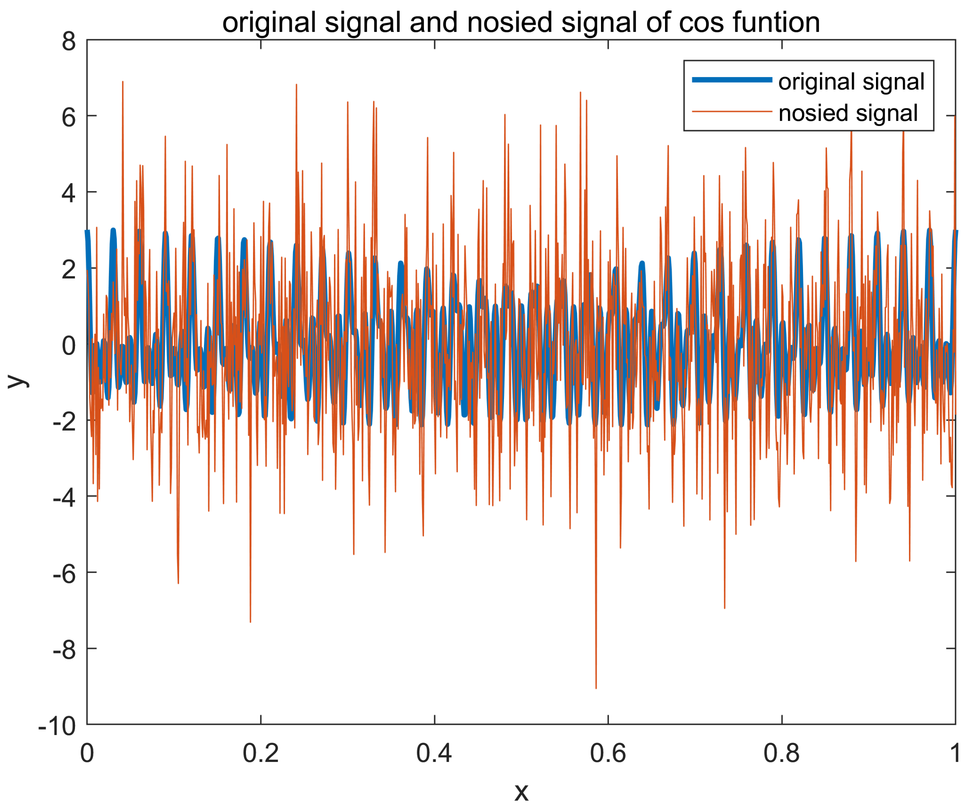
```
f=cos(2*33*x*pi)+cos(2*66*x*pi)+cos(2*100*x*pi);
```

Then add some noise

```
f_noisy=f+2*randn(size(x));

%plot
plot(x,f,'LineWidth',2)
hold on
plot(x,f_noisy)
ylabel('y')
```

```
xlabel('x')
legend('original signal','nosied signal')
title('original signal and nosied signal of cos funtion')
hold off
```



here we take the DFT of both original and noise

```
N=length(x);
F=fft(f,N);
F_noisy=fft(f_noisy);
```

And compute the power spectral density

```
psd=F.*conj(F)/N;
psd_noisy=F_noisy.*conj(F_noisy)/N;
```

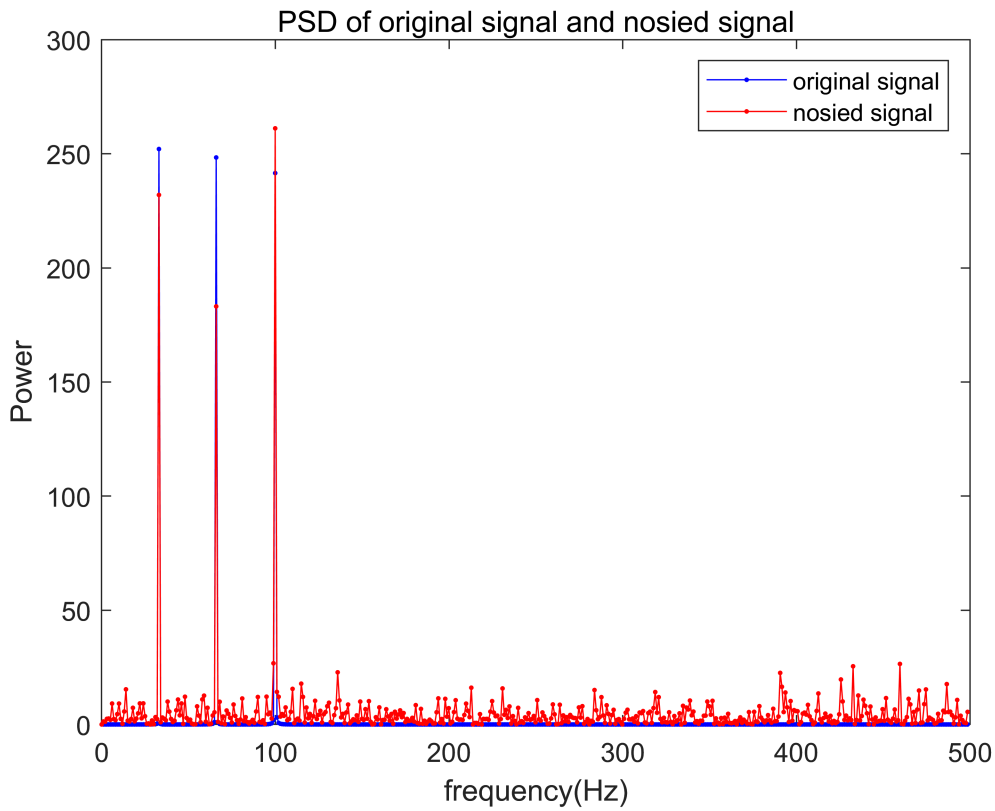
Set the frequecy axis

Since the psd is center symmetric, we only need half of the frequecy axis

```
freq_axis=1/(dx*N)*[0:N-1];
axis_half=1:floor(N/2);

%plot of the spectrum
plot(freq_axis(axis_half),psd(axis_half),'.-b')
hold on
plot(freq_axis(axis_half),psd_noisy(axis_half),'.-r')
xlabel('frequency(Hz)')
```

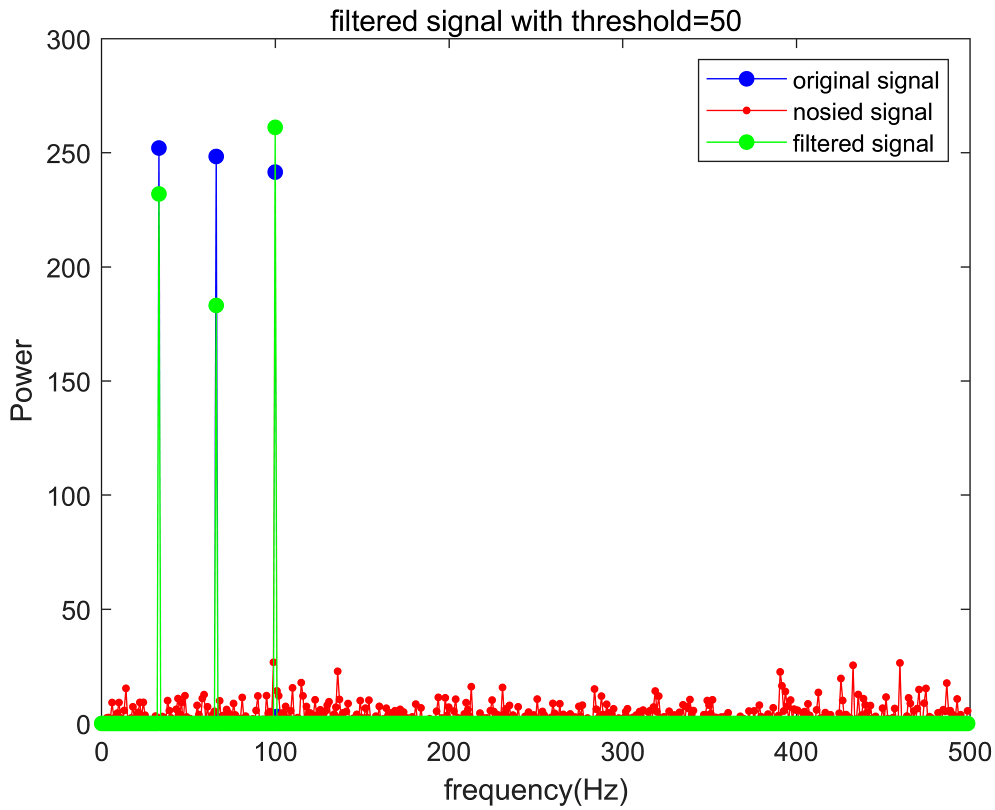
```
ylabel('Power')
legend('original signal','nosied signal')
title('PSD of original signal and nosied signal')
hold off
```



Set threshold=50 and filter out the spectrum less than 50.

```
inds_filter=psd_noisy>50;%return bool
psd_filter=psd_noisy.*inds_filter;

plot(freq_axis(axis_half),psd(axis_half),'.-b','MarkerSize',20)
hold on
plot(freq_axis(axis_half),psd_noisy(axis_half),'.-r','MarkerSize',10)
plot(freq_axis(axis_half),psd_filter(axis_half),'.-g','MarkerSize',20)
xlabel('frequency(Hz)')
ylabel('Power')
legend('original signal','nosied signal','filtered signal')
title('filtered signal with threshold=50')
hold off
```

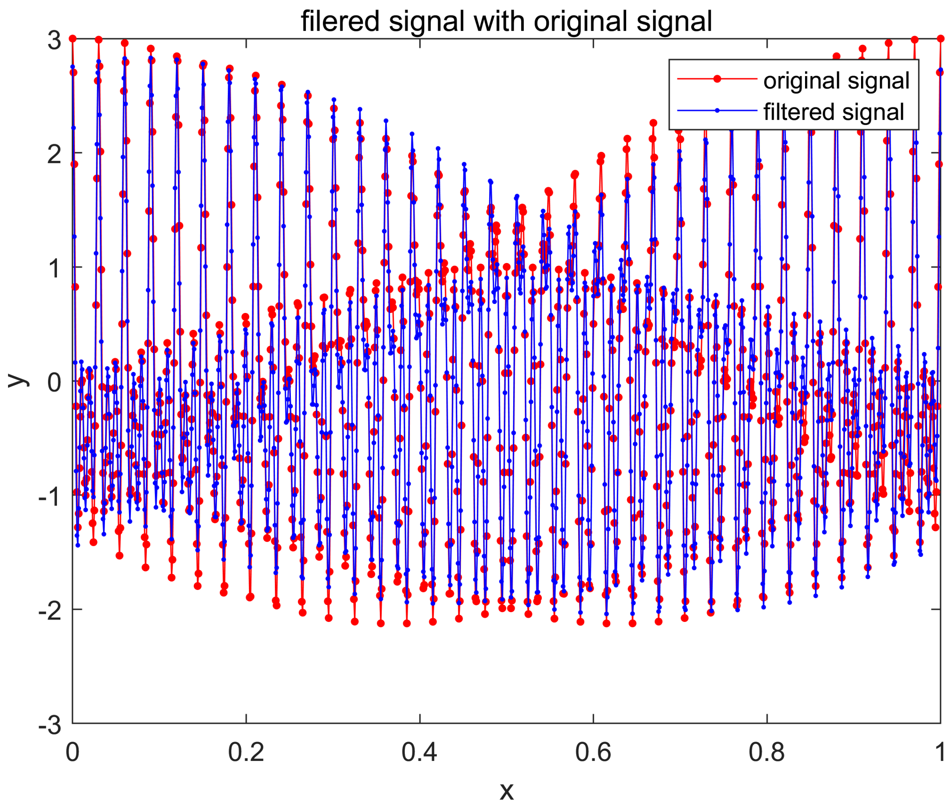


from the plot we can see the filtered signal only has three peek.

Then we need to take iDFT of the filtered spectrum back to the signal

```
F_filter=F_noisy.*inds_filter;%filter out the noied according to the threshold
f_filter=ifft(F_filter);%iDFT

plot(x,f,'.-r','MarkerSize',10)
hold on
plot(x,f_filter,'.-b')
xlabel('x')
ylabel('y')
legend('original signal','filtered signal')
title('filered signal with original signal')
hold off
```



and we can also compute the percentage error

```
Sum_f=sum(abs(f))
```

```
Sum_f = 1.0004e+03
```

```
error=sum(abs((f(:)-f_filter(:))))
```

```
error = 204.4907
```

```
percentage_error=error/Sum_f
```

```
percentage_error = 0.2044
```

PROBLEM 4: (10 pts) Explain how Taylor series can be used to determine the order of the error in numerical methods.

4. with step h.

the taylor expansion of $y(x+h)$ is

$$y(x) + hy'(x) + \frac{h^2}{2!} y''(x) + \frac{h^3}{3!} y'''(x) + \cdots$$

For any numerical method.

the prediction of point x_{n+1} is y_{n+1}

the accuracy is $y(x+h) - y_{n+1}$

Since the prediction of y_{n+1} will need $y(x)$ (which $y(x) = y_0$ is a provided po')

$$y(x+h) - y_{n+1} = hy'(x) + \frac{h^2}{2!} y''(x) + \frac{h^3}{3!} y'''(x) + \cdots$$

$$- h\phi(x, y, h)$$

which $h\phi$ is function that predict the $[y_{n+1} - y(x)]$

~~So~~ the how the order of (h^n) ~~left~~ means the accurate of the method

~~has been~~

Since y', y'' ... can be determine from $f(x)$

So. the term left from the substraction

has order (h^{n+1}) means the method has nth order accurate.

PROBLEM 5: (20 pts) What is *regularization* and why is it needed? Give two examples of regularization methods explaining the basics of how they work and for what type of problems they are appropriate.

Regularization is the method that helping people choosing data from the noise data. With better Regularization method we could de-noise the data and get better result, reduce the over fit.
Method 1: TSVD
Truncked SVD:

for the problem $m = Af + \varepsilon$

F is the things we want
M is the data we have and sigma is the noise

In order to solve m we need $A^{-1}m \approx f$

The singular values of a picture or matrix are separated to wide range. The condition number will be huge and it will be hard to do the inverse computation
Set a limitation and ignore the singular value smaller than the limitation, doing the pseudo-inverse instant of inverse, set the inverse of smaller singular value to 0.

Then using the least square method find vector $\ell(\vec{m})$ that:

$$\|A\ell(\vec{m}) - \vec{m}\| = \min\|A\vec{z} - \vec{m}\|$$

Then vector $\ell(\vec{m})$ will be the solution

Method: Tikhonov

for the problem $m = Af + \varepsilon$

Tikhonov minimize the expression

$\|AT_\alpha(m) - m\|_2^2 + \alpha\|T_\alpha(m)\|_2^2$ with $T_\alpha(m) \in \Re^n$

write $T_\alpha(m) = \arg\min(\|A\vec{z} - \vec{m}\|_2^2 + \alpha\|\vec{z}\|_2^2)$

$$T_\alpha(m) = VS_\alpha^+ U^T \vec{m}$$

$$S_\alpha^+ = diag(\frac{\sigma_1}{\sigma_1^2 + \alpha}, \frac{\sigma_2}{\sigma_2^2 + \alpha}, \ldots, \frac{\sigma_{\min}}{\sigma_{\min}^2 + \alpha})$$

PROBLEM 6: (20 pts) We have studied various solution methods for solving differential and partial differential equations. Compare and contrast *Finite Differences*, the *Finite Element Method*, *Separation of Variables*, *Monte Carlo* methods, and *Fourier Transform* methods. What types of problems can be solved by each method?

6. F D.

① has larger error, $o(h)$

② Can be solve 1st and 2nd order of ODE with a point given, for ~~two~~ 2nd order ODE need two point or one point and one free end

Numerical method.

FEM

① will be used on Solving 2nd order of ODE and PDE

② Has larger Computation ③ numerical method.

Separation of Variable.

① will be used on Solving PDE with IC ~~and~~ or BC.

② True solution will be fund, ③ with some kind of IC, BC will be hard to find the solution.

Monte Carlo :

① Integration : Definite integral

② PDE with IC or BC

③ numerical method ④ take a lot computation by computer.

Fourier Transform.

① Analytical solution will fond

② PDE with 1st or 2nd order.

③ difficult to use if the step of Inverse of Furier Transform cannot be fund

④ Take lots of word to do the convolution.

PROBLEM 7: (10 pts) Discuss how you can determine the 'accuracy' of a numerical method or solution.

7. The accuracy of numerical method can be determined by subtract the approximate value with the taylor expansion of the true value

And for solution, compute the total absolute error ~~betee~~ between the solution and discretized true solution.