به نام حضرت دوست

دانشکده مهندسی برق

درس ریاضی مهندسی

گزارش تمرین matlab

سری دوم

تاریخ تحویل تمرین: 96/8/5

نام و نام خانوادگی:

طاها انتصاری 95101117

وحید احمدی 95109083

- **قسمت اول: محاسبه سری فوریه و مجموع های جزئی**

1. برای هریک از توابع زیر سری فوریه را محاسبه می کنیم و تا جمله هفتم سری را چاپ می کنیم.

توابع 2 ضابطه ای را با استفاده از تابع sign می سازیم.

a)

```
clear
syms T f n x
a=@(T ,f ,n, x)(2/T*int(f.*cos(2*pi*n*x/T),x,-T/2,T/2));
b=@(T ,f ,n, x)(2/T*int(f.*sin(2*pi*n*x/T),x,-T/2,T/2));
fouriers=@(f , x ,k ,T)(a(T,f,0,x)/2+symsum(a(T,f,n,x)*cos(2*pi*n*x/T)+b(T,f,n,x)*sin(2*pi*n*x/T),n,1,k));

f=@(x)(x*(sign(x)+1)/2);
ffs=fouriers(f,x,7,2);
disp(f(x))
pretty(f(x));
pretty(ffs)
```

b)

```
g=@(x)(sin(pi*x/2));
gfs=fouriers(g,x,7,2);
pretty(g(x));
pretty(gfs)
```

c)

```
h=@(x)(x^2);
pretty(h(x));
hfs=fouriers(h,x,7,2);
pretty(hfs);
```

d)

```
l=@(x)(sin(x)*(sign(x)+1)/2);
lfs=fouriers(l,x,7,2*pi);
pretty(l(x));
pretty(lfs)
```

با اجرای کد بالا نتایج را در زیر مشاهده می کنیم.

a)

```
x (sign(x) + 1)
---------------
       2

sin(pi x)   cos(3 pi x) 2   cos(5 pi x) 2   cos(7 pi x) 2   2 cos(pi x)   sin(2 pi x)   sin(3 pi x)   sin(4 pi x)
--------- - ------------- - ------------- - ------------- - ----------- - ----------- + ----------- - -----------
   pi             2               2               2               2          2 pi          3 pi          4 pi
              9 pi            25 pi           49 pi             pi

    sin(5 pi x)   sin(6 pi x)   sin(7 pi x)   1
  + ----------- - ----------- + ----------- + -
       5 pi          6 pi          7 pi       4
```

b)

```
   / pi x \
sin| ---- |
   \   2  /

8 sin(pi x)   sin(2 pi x) 16   sin(3 pi x) 24   sin(4 pi x) 32   sin(5 pi x) 40   sin(6 pi x) 48   sin(7 pi x) 56
----------- - -------------- + -------------- - -------------- + -------------- - -------------- + --------------
   3 pi           15 pi            35 pi            63 pi            99 pi            143 pi           195 pi
```

c)

```
 2
x

cos(2 pi x)   4 cos(pi x)   cos(3 pi x) 4   cos(4 pi x)   cos(5 pi x) 4   cos(6 pi x)   cos(7 pi x) 4   1
----------- - ----------- - ------------- + ----------- - ------------- + ----------- - ------------- + -
       2             2               2              2               2             2               2     3
     pi            pi            9 pi           4 pi           25 pi          9 pi           49 pi
```

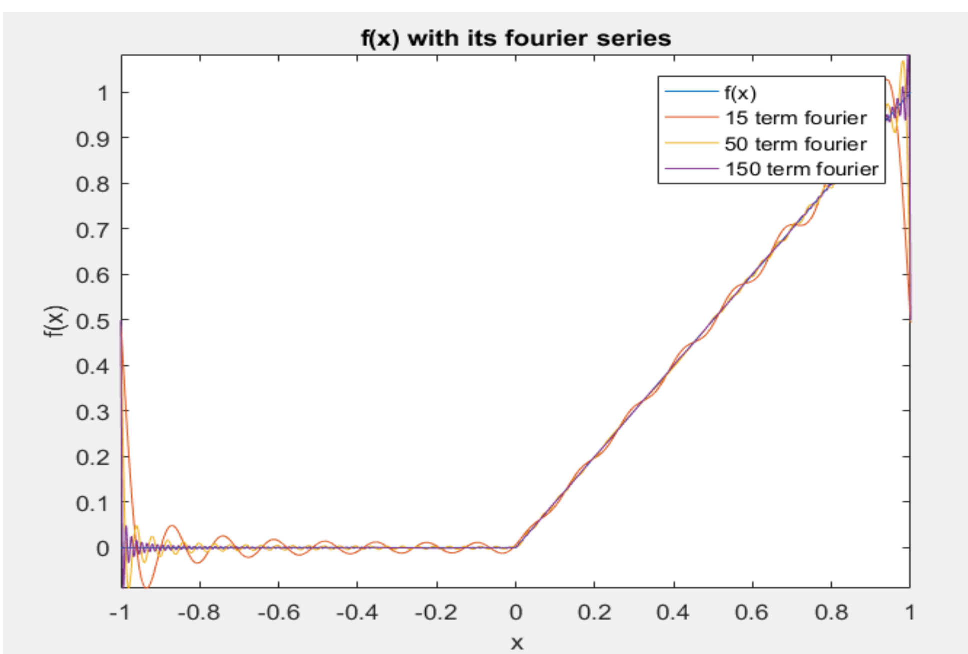
d)

```
sin(x) (sign(x) + 1)
--------------------
         2

5734161139222659 pi sin(x)   cos(4 x) 1911387046407553   cos(6 x) 5734161139222659   cos(2 x) 1911387046407553
-------------------------- - ------------------------- - ------------------------- - -------------------------
    36028797018963968            45035996273704960          315251973915934720           9007199254740992

     5734161139222659
   + -----------------
     18014398509481984
```

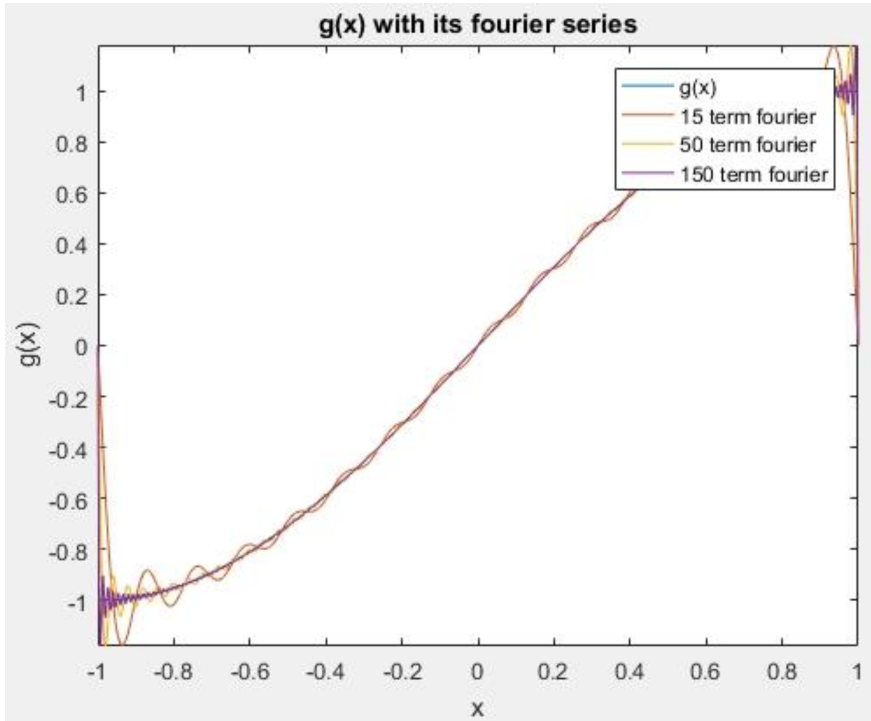
2. به ازای $N = \{15,50,150\}$ مجموع سری را به دست آورده و همراه با منحنی برروی یک نمودار رسم می کنیم.

مشاهده می کنیم با زیاد شدن تعداد جملات، مجموع سری به خود تابع اصلی نزدیک می شود.

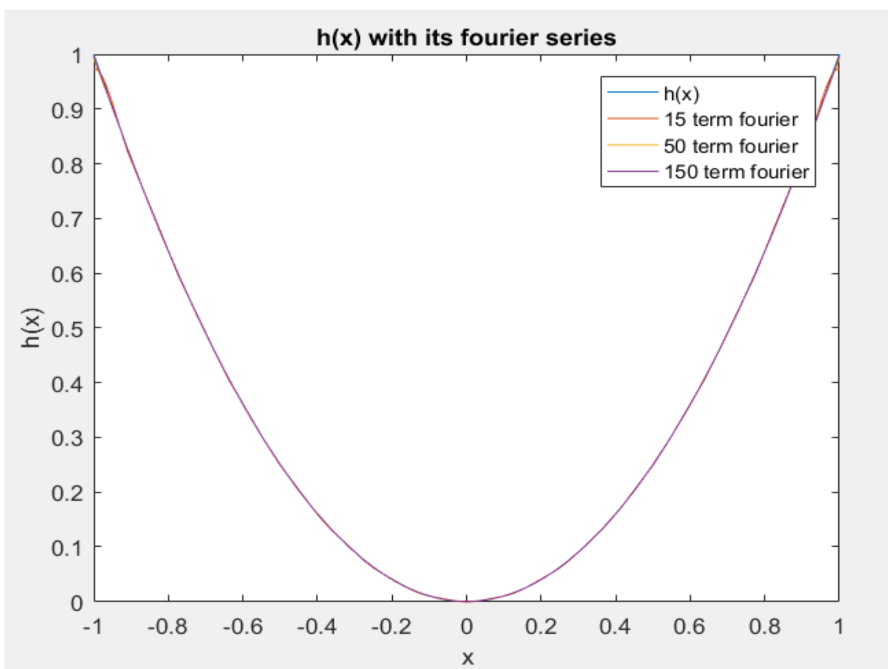
a)



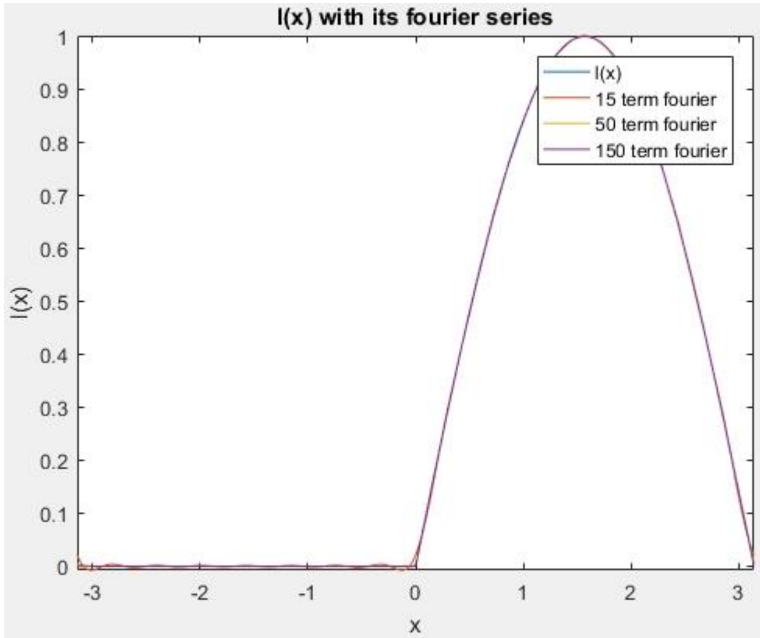
b)


g(x) with its fourier series
g(x)
15 term fourier
50 term fourier
150 term fourier
g(x)
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

c)


h(x) with its fourier series
h(x)
15 term fourier
50 term fourier
150 term fourier
h(x)
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

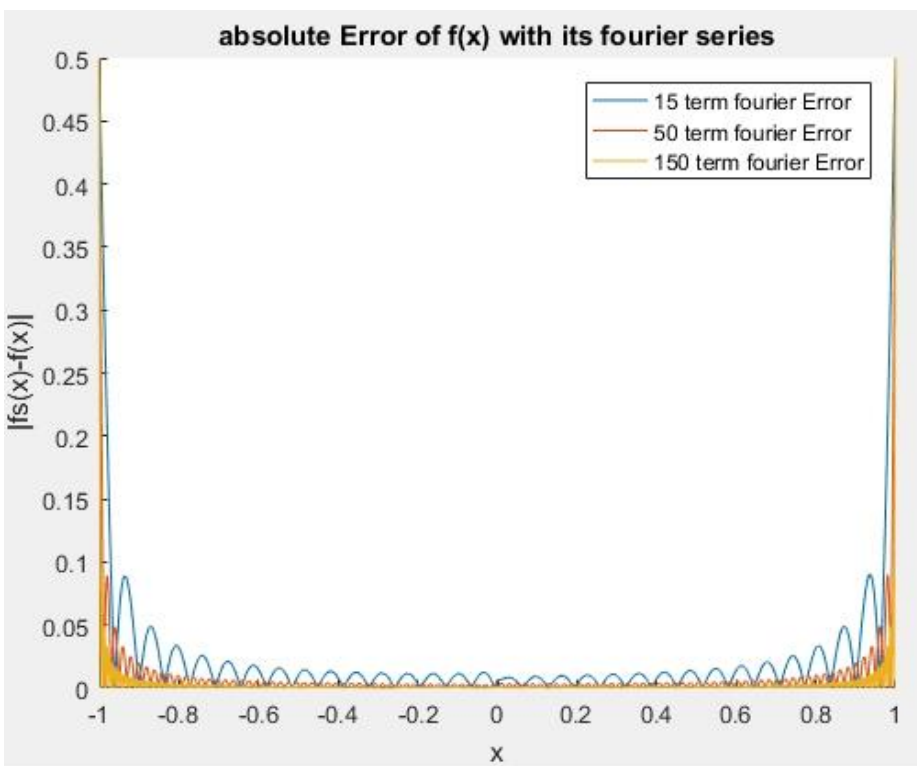
d)



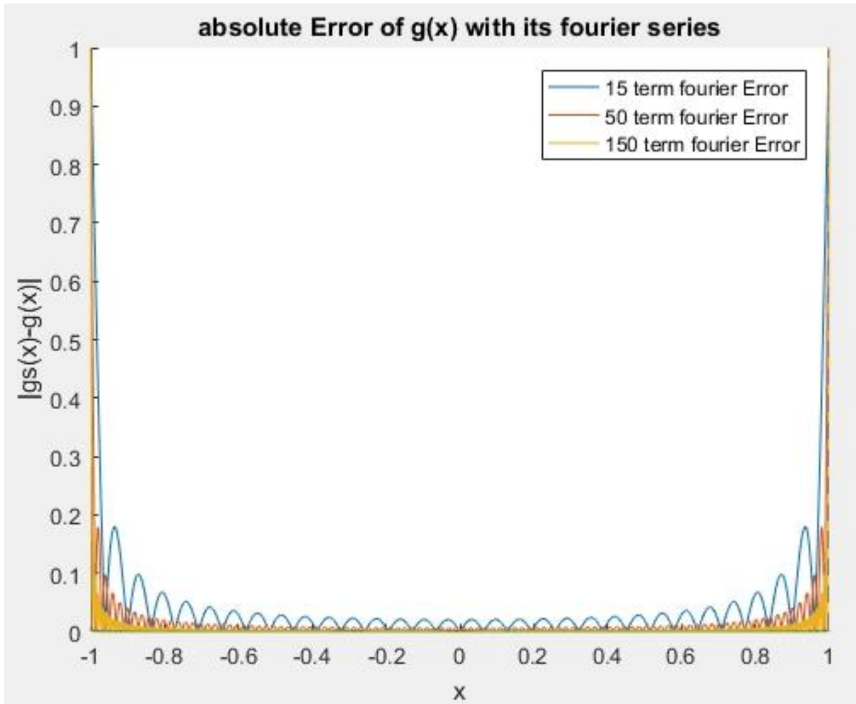
3. برای هر یک از توابع بالا نمودار خطا را به ازای هر کدام از Nها رسم می کنیم.

مشاهده می کنیم که زیادتر شدن جملات سری، میزان انحراف از تابع اصلی کم و کمتر می شود.

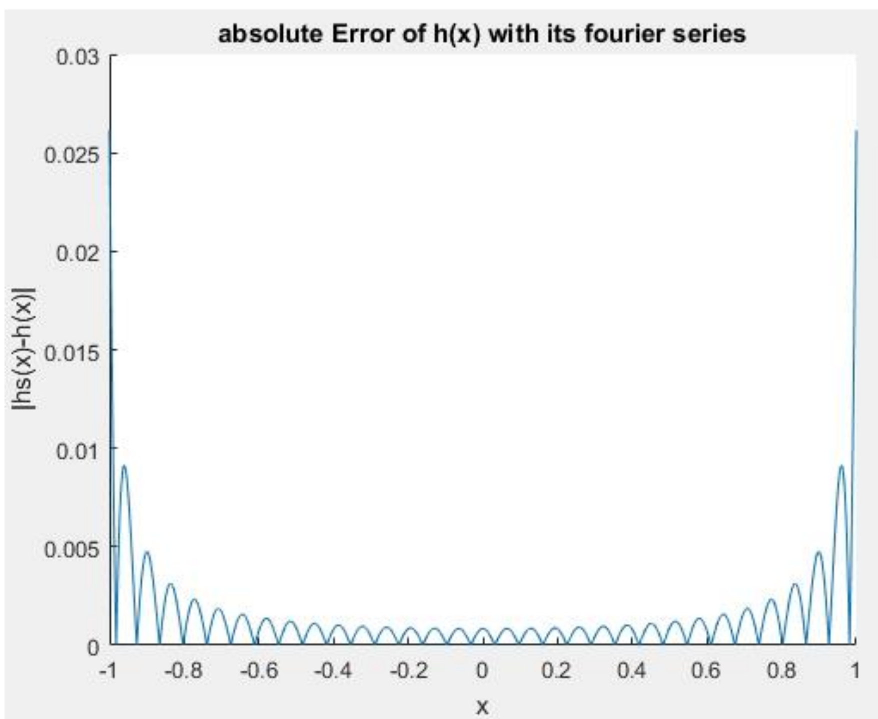
a)



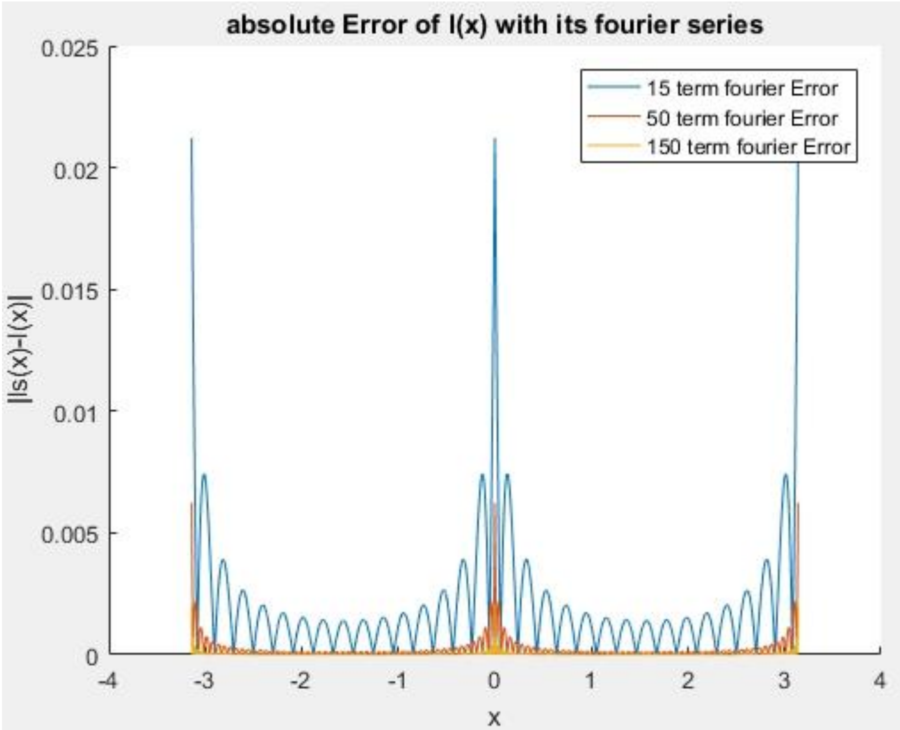
b)


absolute Error of g(x) with its fourier series
15 term fourier Error
50 term fourier Error
150 term fourier Error
|gs(x)-g(x)|
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

c)


absolute Error of h(x) with its fourier series
|hs(x)-h(x)|
x
0.03
0.025
0.02
0.015
0.01
0.005
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

d)



- **قسمت دوم: بررسی انرژی سیگنال (قضیه بسل)**

1. ابتدا با استفاده از متلب انرژی هر یک از تابع های زیر را بر روی دوره تناوب آن به دست می آوریم . (توابع روی یک دوره 1 تناوب داده شده اند این توابع را به همین شکل گسترش متناوب می دهیم)

تعریف انرژی یک تابع را در خط آخر کد زیر مشاهده می کنید.

```
clear
syms T f n x
a=@(T ,f ,n, x)(2/T*int(f.*cos(2*pi*n*x/T),x,-T/2,T/2));
b=@(T ,f ,n, x)(2/T*int(f.*sin(2*pi*n*x/T),x,-T/2,T/2));
fouriers=@(f , x ,k ,T)(a(T,f,0,x)/2+symsum(a(T,f,n,x)*cos(2*pi*n*x/T)+b(T,f,n,x)*sin(2*pi*n*x/T),n,1,k));
E=@(T,x,f)(1/T*int(f(x)^2,x,-T/2,T/2));
```

a)

```
f=@(x)(abs(x));
Ef=E(2,x,f);
pretty(f(x));
pretty(Ef);
```

b)

```
g=@(x)(x);
Eg=E(1,x,g);
pretty(g(x));
pretty(Eg);
```

c)

```
u=@(x)(sinh(x));
Eu=E(4,x,u);
pretty(u(x));
pretty(Eu);
```

با اجرای کد بالا نتایج را در زیر مشاهده می کنیم.

a)

```
|x|

1
-
3
```

b)

```
x

 1
--
12
```

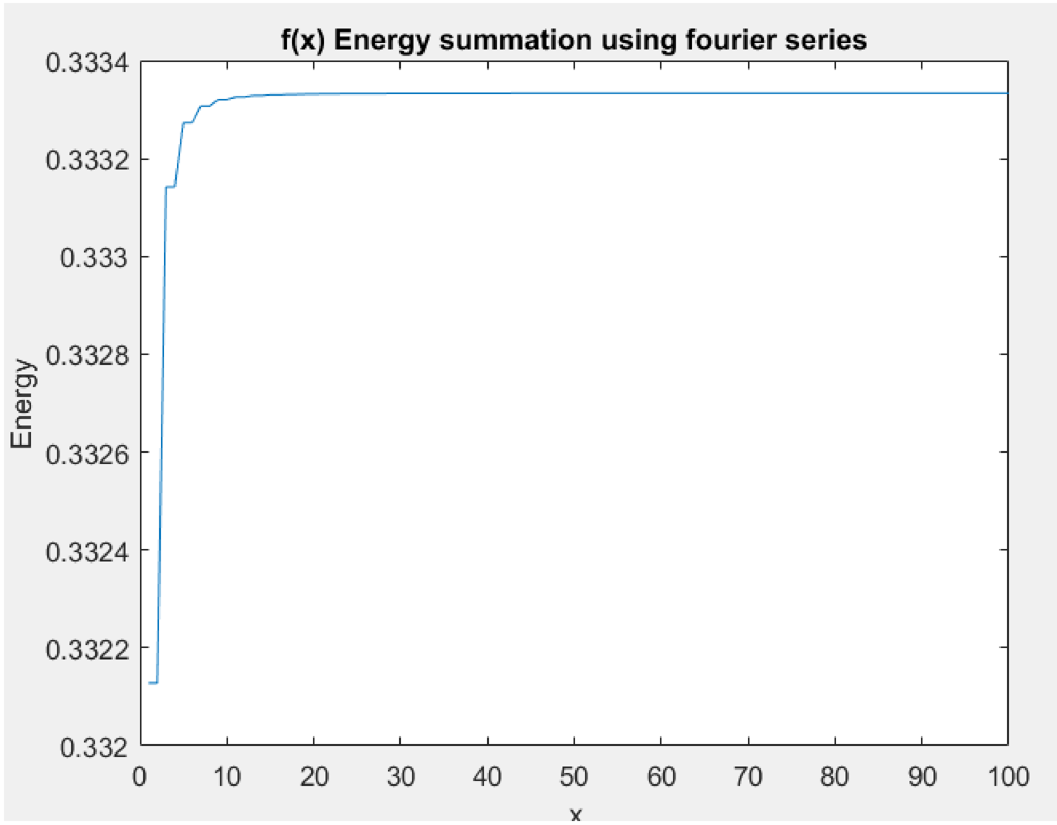
c)

```
sinh(x)

exp(4)   exp(-4)   1
------ - ------- - -
  16        16     2
```

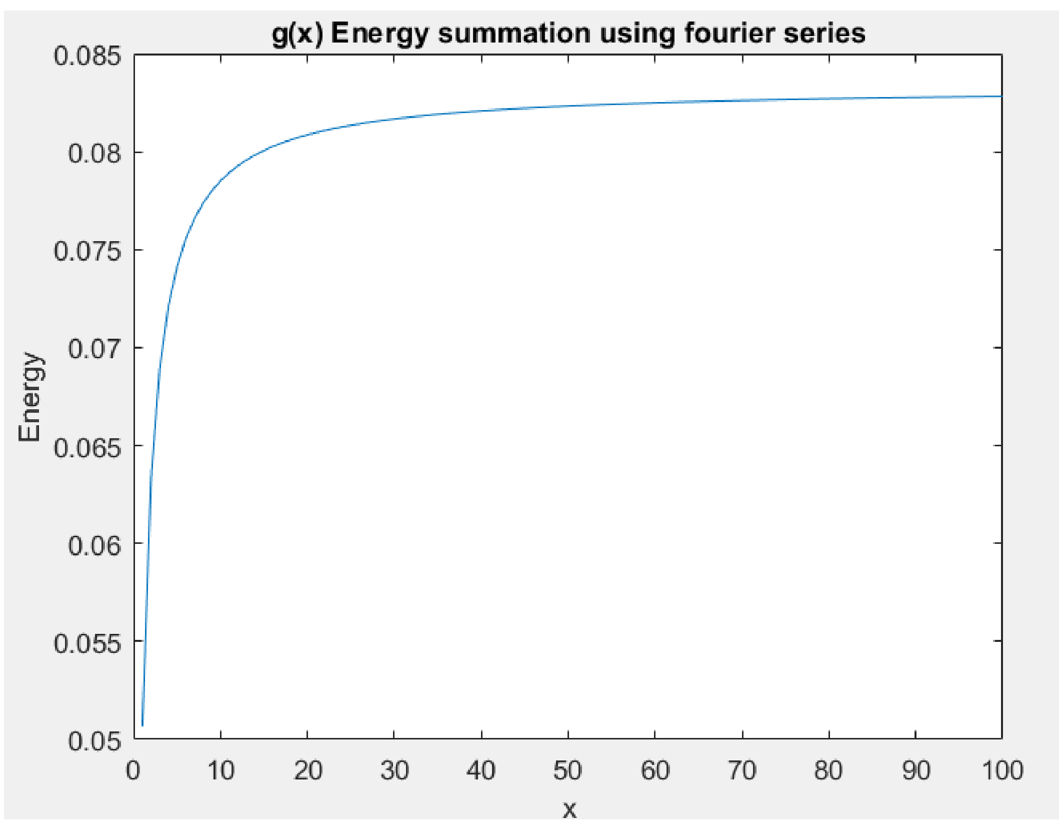
2. حال انرژی مجموع های جزئی سری فوریه تا جمله N ام را برای N های برابر 1 تا 100 داده شده به دست می آوریم و نمودار انرژی بر حسب N های داده شده رسم می کنیم.

مشاهده می کنیم که هرچقدر تقریب بهتری از تابع اصلی داشته باشیم، انرژی ما به انرژی تابع اصلی نزدیک تر می شود.

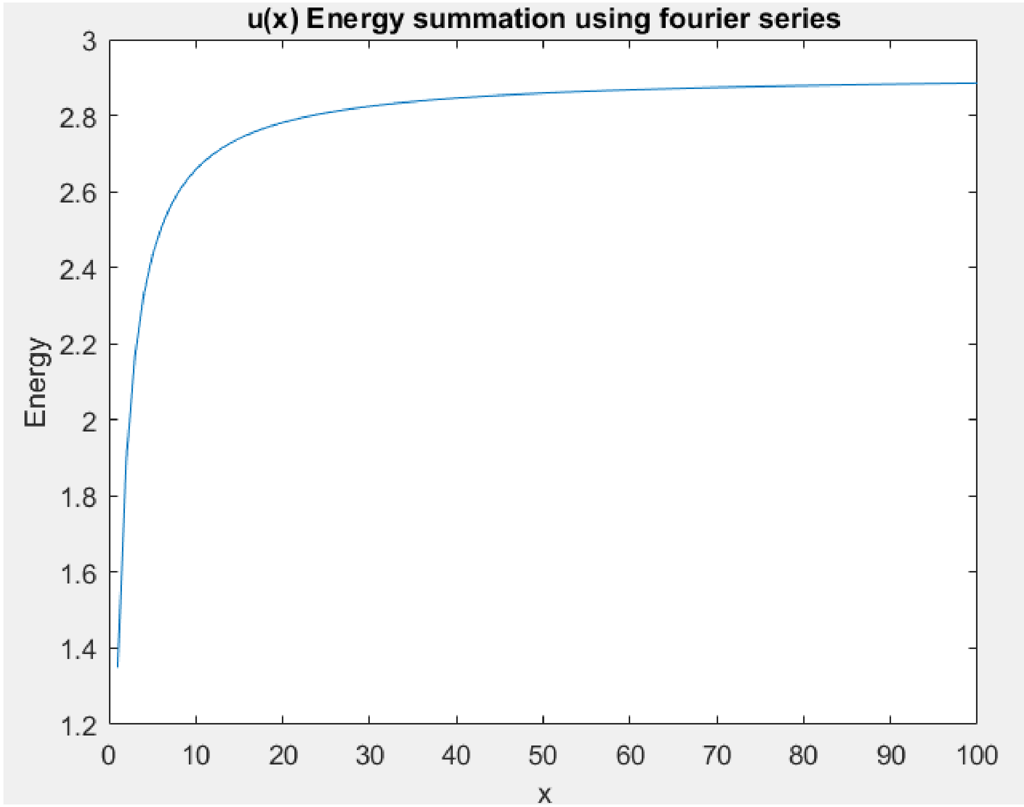
a)


f(x) Energy summation using fourier series
Energy
x
0.332
0.3322
0.3324
0.3326
0.3328
0.333
0.3332
0.3334
0 10 20 30 40 50 60 70 80 90 100

b)


g(x) Energy summation using fourier series
Energy
x
0.05
0.055
0.06
0.065
0.07
0.075
0.08
0.085
0 10 20 30 40 50 60 70 80 90 100

c)

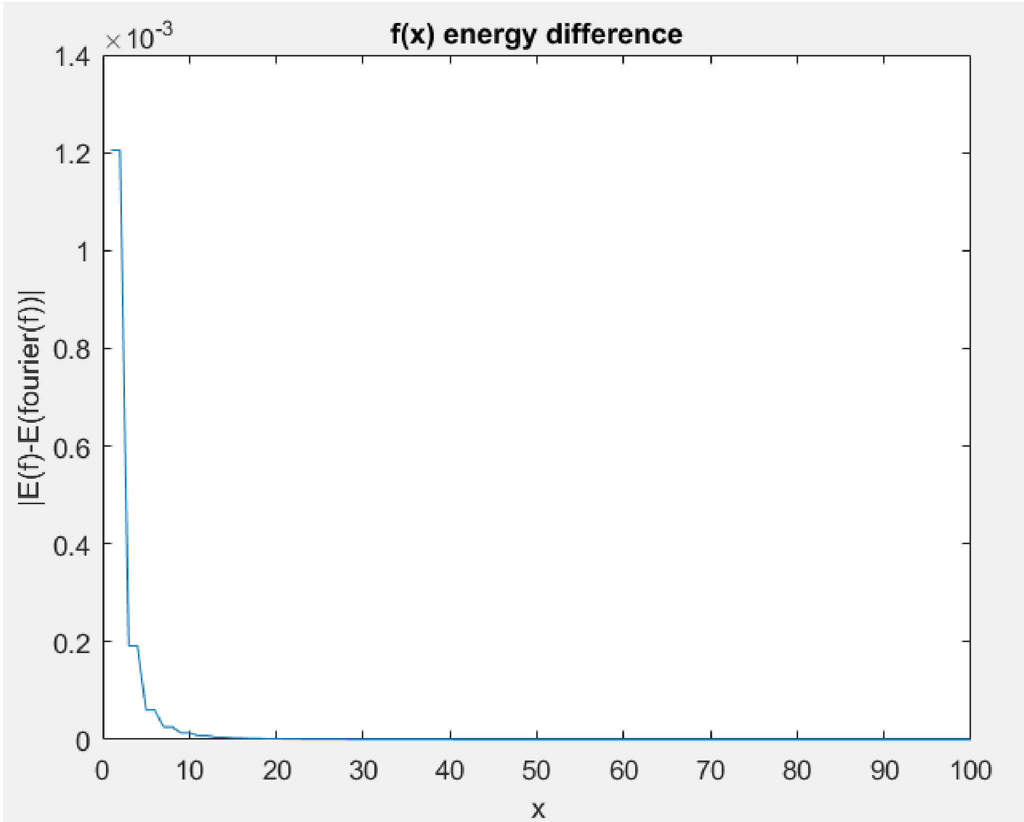


3. با بررسی منحنی های تفاضل انرژی اصلی سیگنال و تقریب های زده شده در هر مورد حداقل N را که به ازای آن این خطا کمتر از 0.1٪ از انرژی سیگنال اصلی می شود به دست می آوریم.

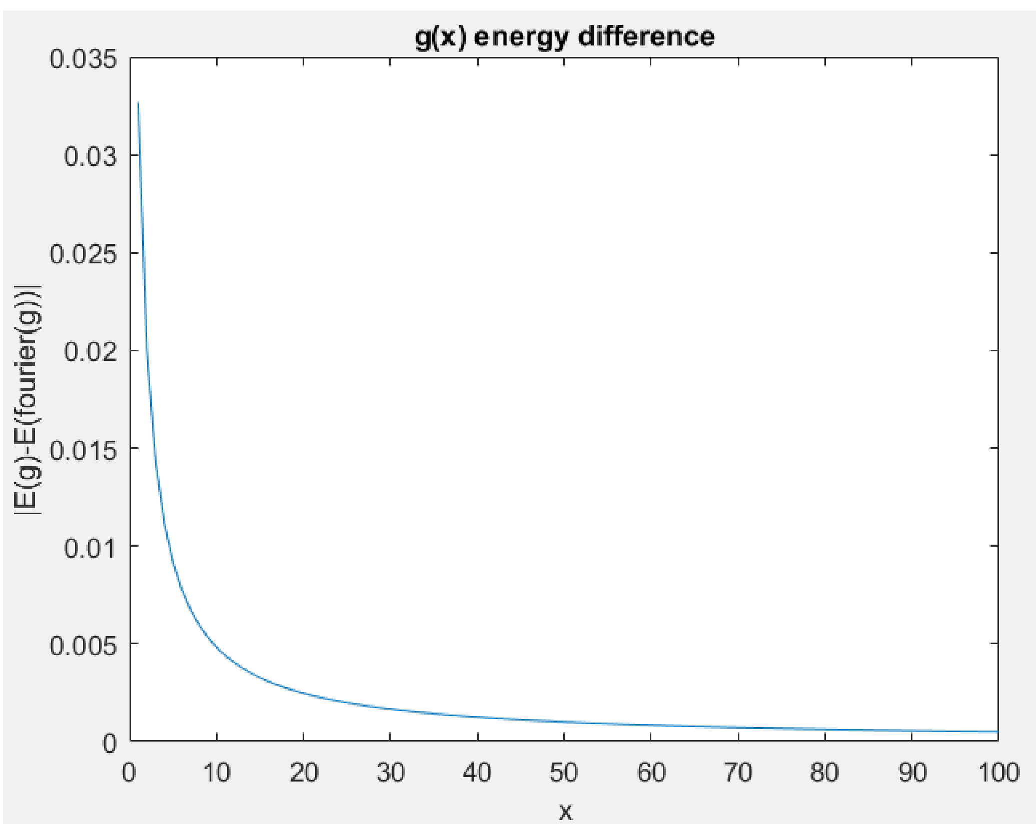
قطعه کد زیر برای a است. (b و c نیز به همین ترتیب است) با توجه به کد زیر، خواسته مسئله یعنی " حداقل N را که به ازای آن این خطا کمتر از 0.1٪ از انرژی سیگنال اصلی شود " را برای هر مورد بدست می آوریم.

```
ferr=eval(abs(Ef-Efs));
ffirstN=find((ferr./Ef)<.1);
figure(12);
cla;
title('f(x) energy difference');
xlabel('x');
ylabel('|E(f)-E(fourier(f))|');
plot(N,ferr);
fprintf('the least N for which energy using fourier series has 0.1percent err for f(x): %d  \n',N(ffirstN(1)));
```

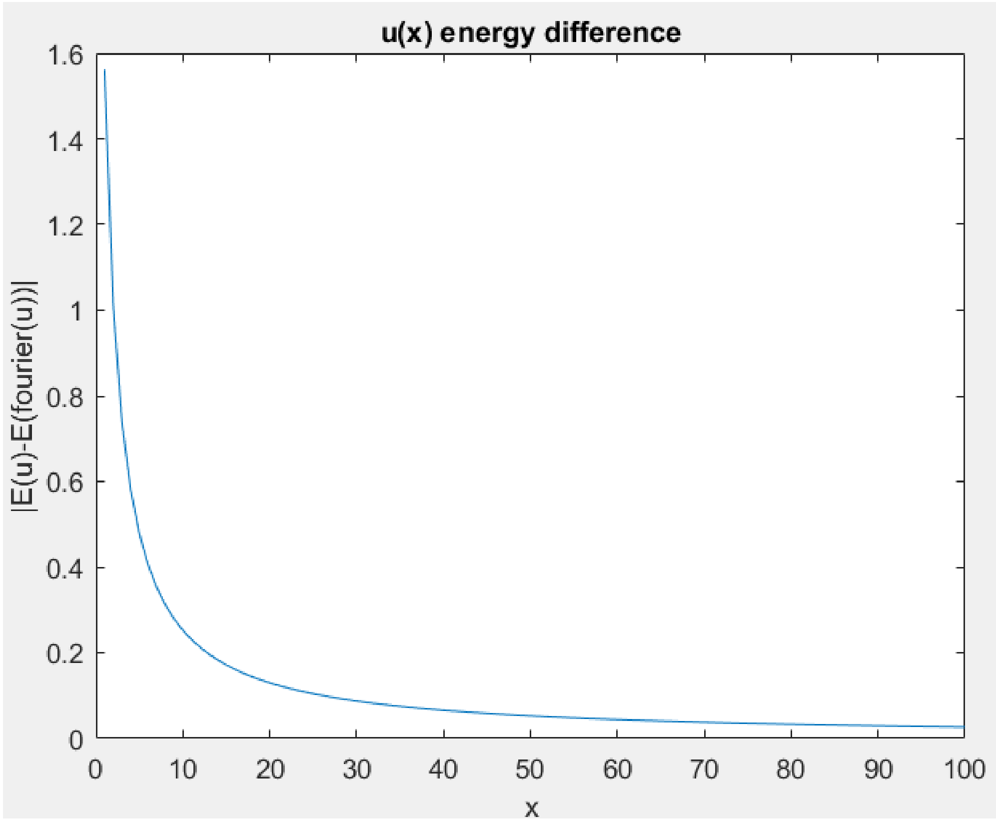
a)


f(x) energy difference
×10^-3
1.4
1.2
1
0.8
0.6
0.4
0.2
0
|E(f)-E(fourier(f))|
0
10
20
30
40
50
60
70
80
90
100
x

b)


g(x) energy difference
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
|E(g)-E(fourier(g))|
0
10
20
30
40
50
60
70
80
90
100
x

c)



- **قسمت سوم: بررسی اثر گیبس**

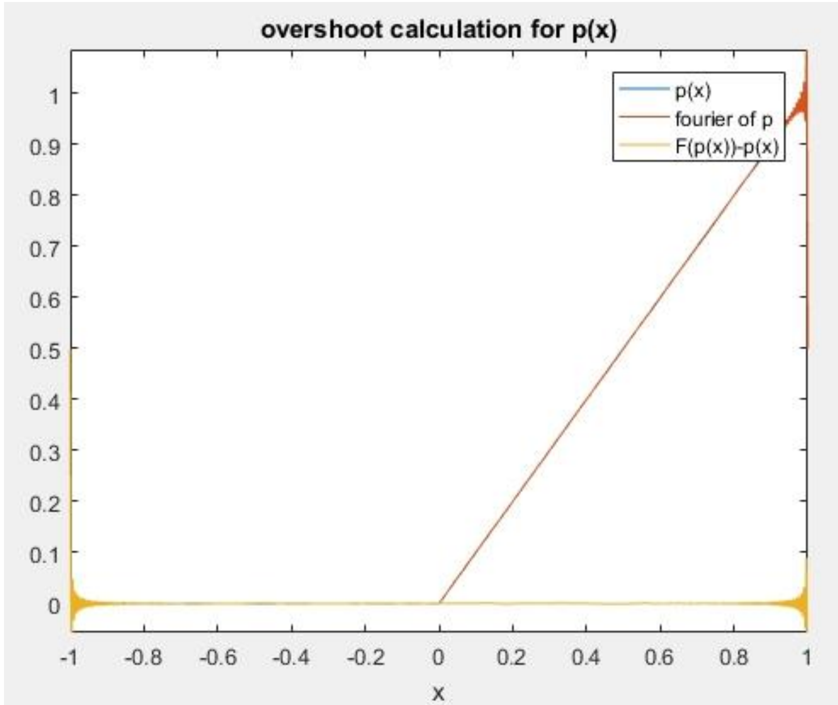
1. برای توابع داده شده سیگنال را گسترش متناوب می دهیم، سپس سری فوریه را برای این توابع محاسبه می کنیم و میزان overshoot را در هر مورد به دست می آوریم.

توابع 2 ضابطه ای را با استفاده از تابع sign می سازیم.

قطعه کد زیر برای بخش a است. (بخش b هم به همین روال انجام می گیرد) نحوه کشیدن و محاسبه overshoot را در کد زیر مشاهده می کنیم.

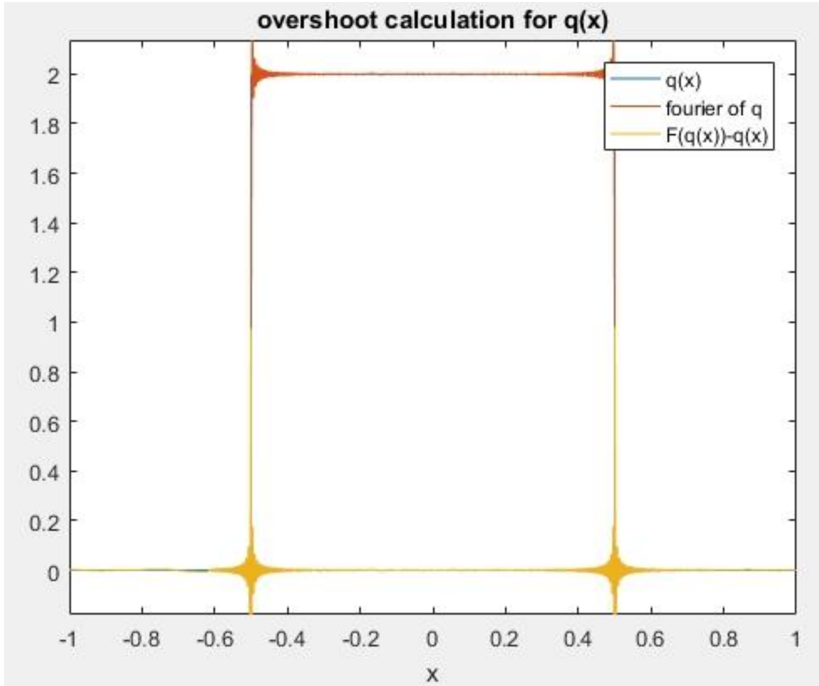
```
p=@(x)(x*(sign(x)+1)/2);
pfs=fouriers(p,x,300,2);
figure(15);
cla;
fplot(p(x),[-1,1]);
hold on
fplot(pfs,[-1,1]);
fplot(pfs-p(x),[-1,1]);
legend('p(x)','fourier of p','F(p(x))-p(x)');
title('overshoot calculation for p(x)');
xlabel('x');
x=(-1-eps):.00001:(1+eps);
pfsevaluatednumerically=(eval(pfs));
maxima=(max(pfsevaluatednumerically));
minima=(min(pfsevaluatednumerically));
fprintf('overshoot of p(x) fourier calculation at the end of the boundry: %f\n',maxima-1);
fprintf('overshoot of p(x) fourier calculation at the beginning of the boundry: %f\n',minima);
```

a)



overshoot calculation for p(x)
p(x)
fourier of p
F(p(x))-p(x)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x

b)

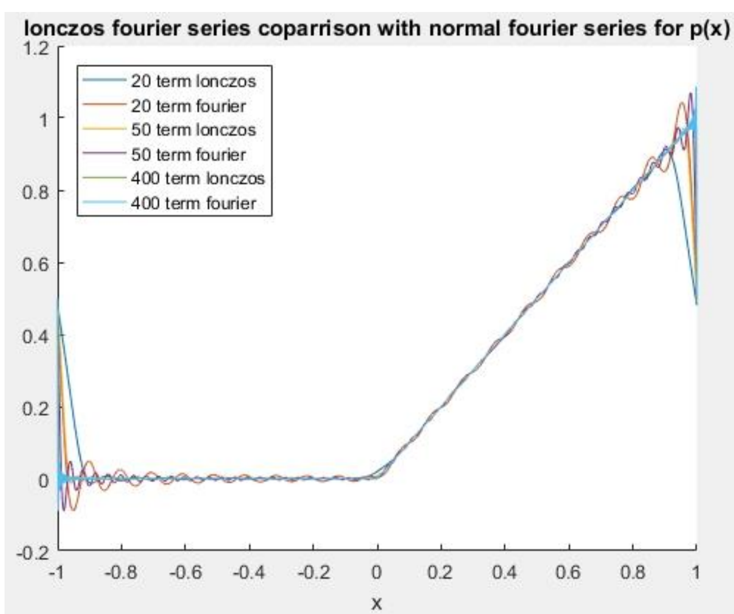


نتیجه گیری شهودی در مورد overshoot:

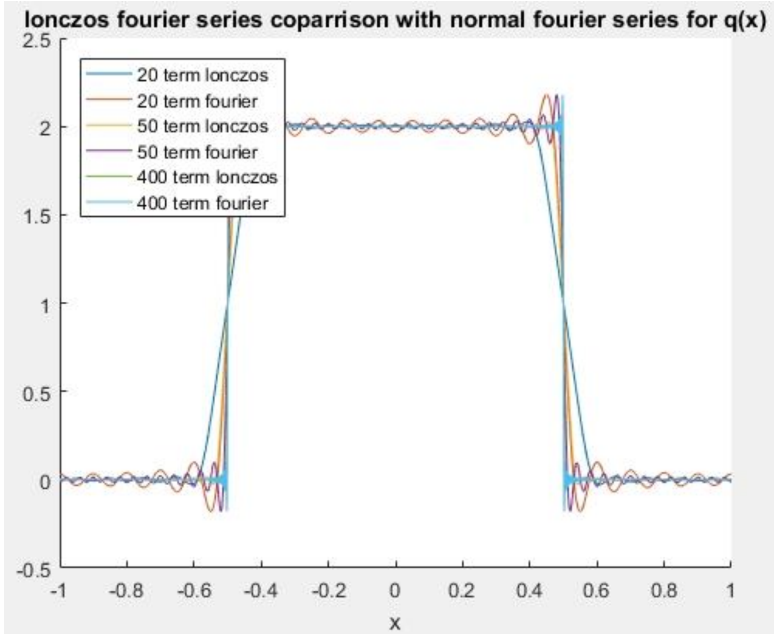
مقدار overshoot در هر یک از توابع بالا به میزان نصف جامپ تابع در آن نقطه است.

2. با استفاده از تقریب گفته شده مجموع های جزئی را در حالات {20 , 50 , 400} =N برای توابع بالا به همراه تقریب های مجموع های جزئی عادی در یک نمودار رسم می کنیم. مشاهده می کنیم در الگوریتم Lonczos که برای فیلتر کردن و حذف overshoot به کار می رود، با زیادتر شدن جملات N (یا همان ضرایبِ بهبود) مقدار overshoot کمتر می شود و به تقریب بهتری از تابع می رسیم.

a)



b)



- **قسمت چهارم: تخمین یک تابع با استفاده از سری فوریه**

1. سری فوریه را با روش گفته شده در سوال، با تقریب زدن ضرایب و جمع جملات تخمین می زنیم. بدین منظور ابتدا ضرایب تا جمله دهم سری (a5 و b5 با صرف نظر از a0) را تقریب زده و آن ها را چاپ می کنیم. سپس با توجه به اندازه ضرایب جملاتی که فکر می کنیم برای داشتن یک تخمین خوب از تابع کافی می باشند را جمع می کنیم.

```
clear
load x;
load y;
a=zeros(1,5);
b=zeros(1,5);
for i=1:5
a(i)=2/2001*sum(y.*cos(pi*i*x));
b(i)=2/2001*sum(y.*sin(pi*i*x));
end
fprintf('The coefficients are:\n \t\ta \t\t b\n');
disp([a' b'])
% Is is clear that the most significant coefficients are a2,b1 and b3.
%thus we estimate the main function using these coefficients
f=a(2)*cos(2*pi*x)+b(1)*sin(pi*x)+b(3)*sin(3*pi*x);
```

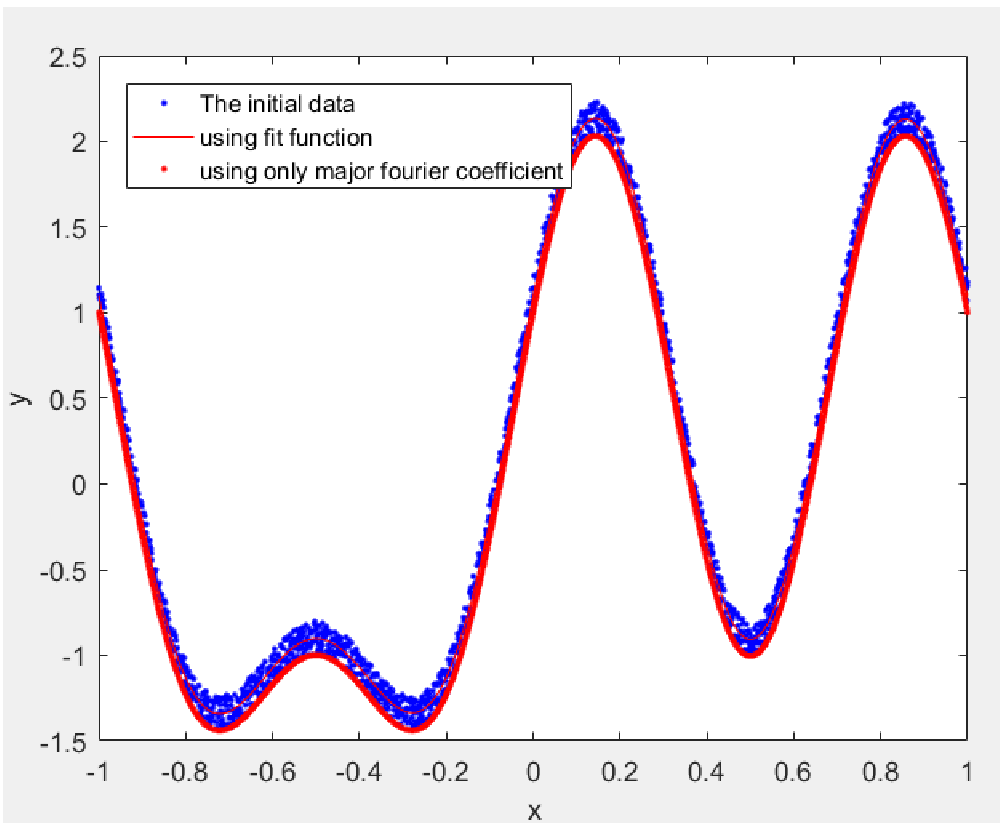
با اجرای دستور بالا نتیجه زیر را مشاهده می کنیم:

```
The coefficients are:
         a          b
      0.0002     0.9975
      1.0014    -0.0001
     -0.0025     1.0004
     -0.0008     0.0014
     -0.0016     0.0007
```

که معلوم است که ضرایب b1,b3،a2 مناسب اند.(نمودار این قسمت دی قسمت 3 آمده)

2. در این بخش با استفاده از تابع fit متلب و حالت fouriers5 آن ضرایب و دوره تناوب تابع برازش شده بر داده ها را می یابیم و این بار نیز با جمع زدن جملات موثر یک تخمین خوب برای تابع به دست می آوریم.

3. دو تابع تخمینی و داده های اصلی را در یک نمودار همزمان رسم می کنیم.



و حدس ما برای تابع اصلی:

می دانیم سه ضریب فوریه مقدار ناصفری هستند؛ و همچنین خیلی نزدیک به 1 هستند. تابع اصلی ما می تواند همان 3 جمله با ضریب 1 بوده باشد که مقداری نویز روی آن آمده باشد.