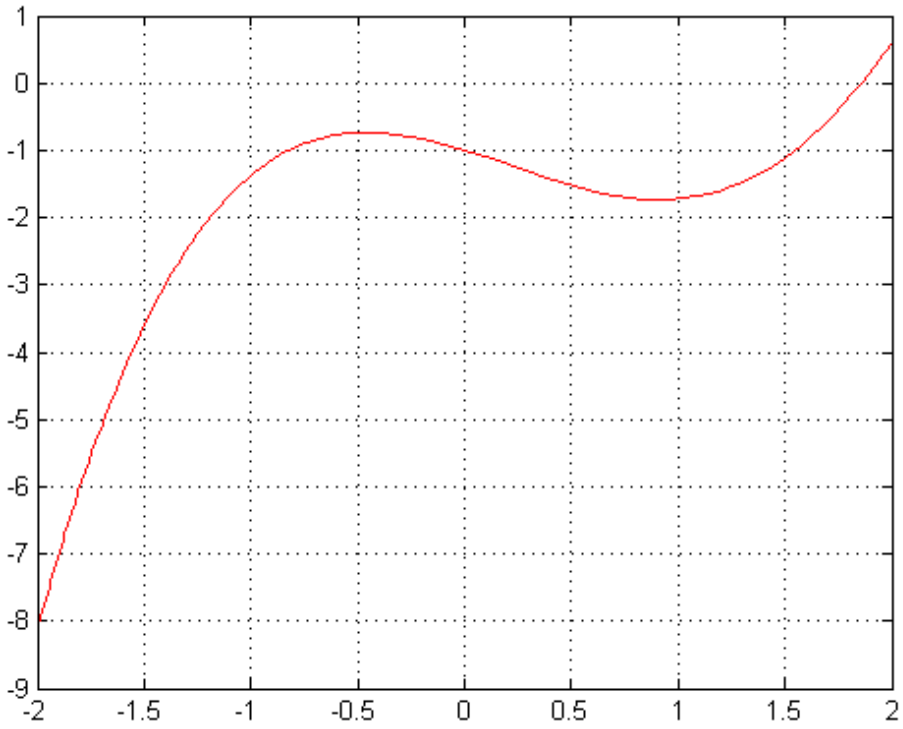
Class_Work_12

Contents

fplot- plot a function defined as string between specified limits

syntax: fplot('string,[xmin, xmax])

```
clc;clear all;clf;format compact
f1='x^3-exp(x)';
f2='t^3-exp(t)';
fplot(f1,[-2 2])
fplot(f1,[-2 2],'r')
grid on
```

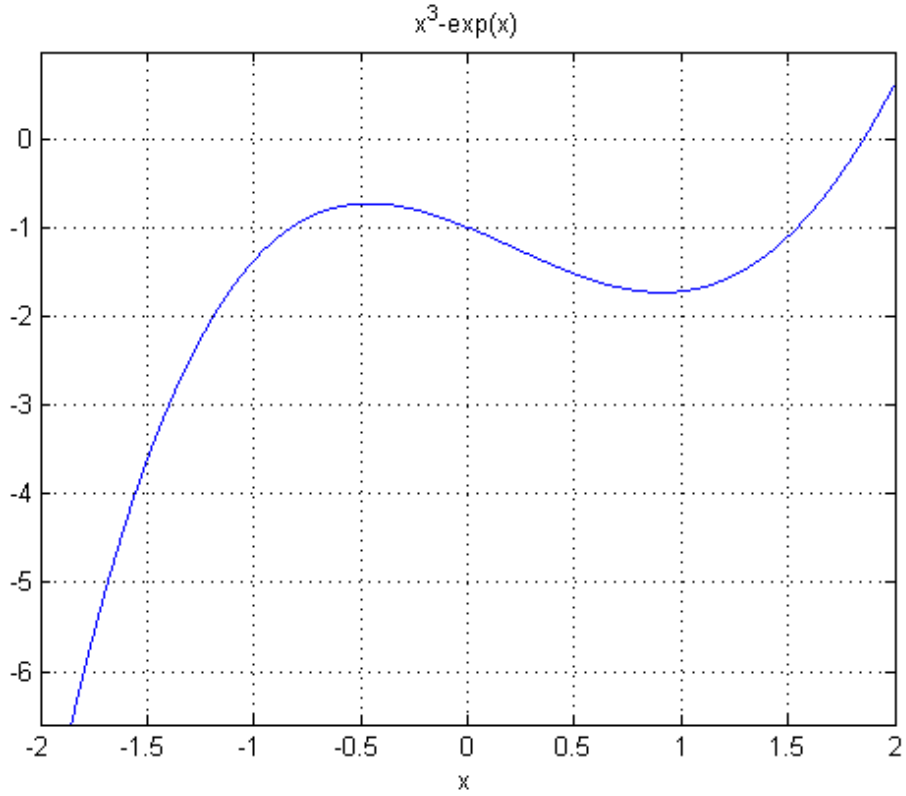


ezplot- Easy-to-use function plotter

plots the expression fun(x) over the default domain -2pi < x < 2pi

syntax: ezplot('string')

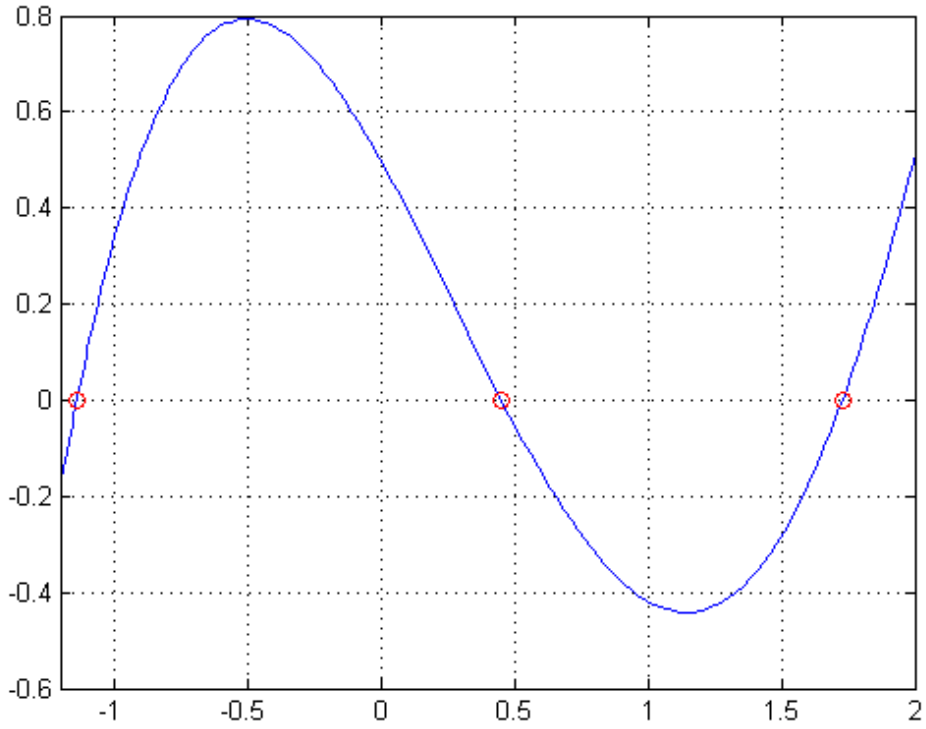
```
ezplot(f1)
ezplot(f1,[-2 2])
grid on
```



fzero- Find root of function of one variable

```
clc;clear all;clf;format compact
func='0.8*x^3-exp(x)+1.5'
fplot(func,[-1.2 2]);grid on
r1=fzero(func,-1)
r2=fzero(func,0.5)
r3=fzero(func,1.5)
hold on
plot([r1 r2 r3],[0 0 0],'ro')
hold off
```

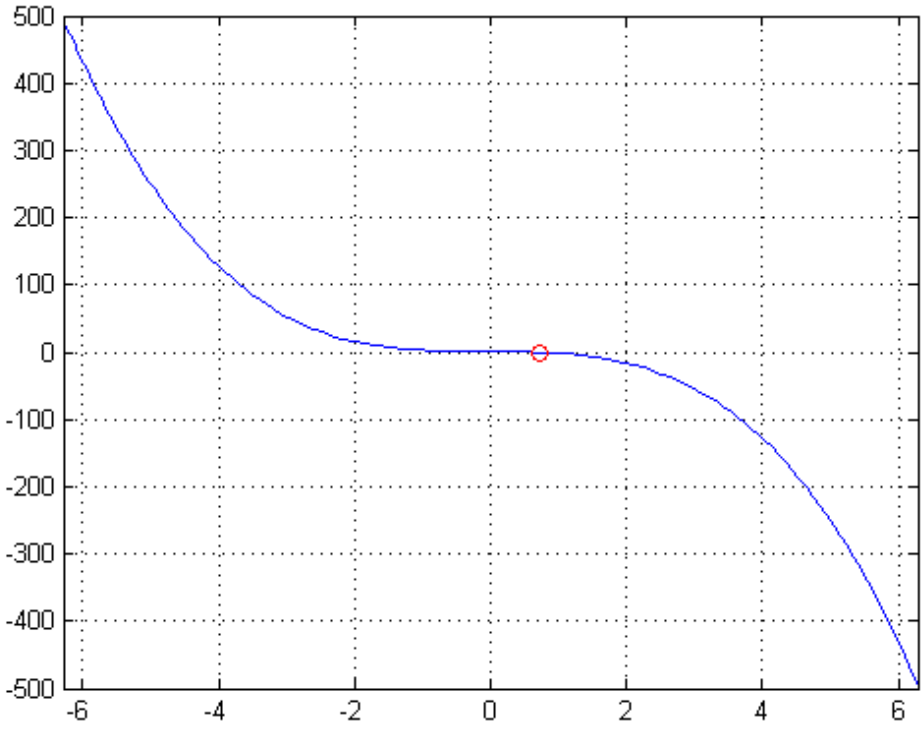
```
func =
0.8*x^3-exp(x)+1.5
r1 =
   -1.1382
r2 =
    0.4542
r3 =
    1.7290
```



class_assign_12,1

```
clc;clear all; format compact;clf
func='cos(x)-2*x^3';
fplot(func,[-2*pi 2*pi])
grid on
x0=fzero(func,0)
hold on
plot(x0,0,'ro')
hold off
```

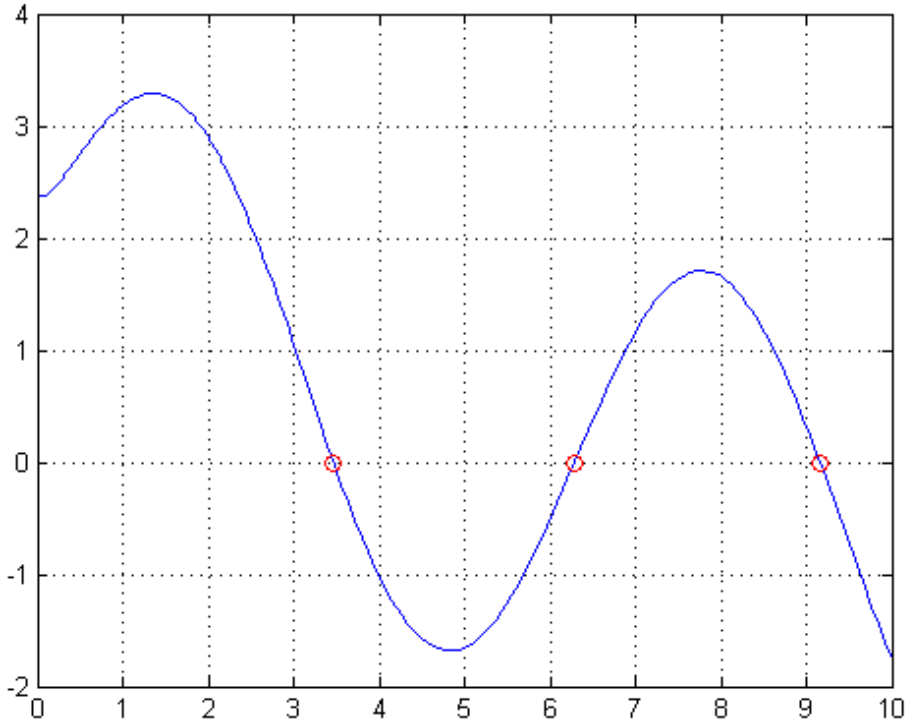
```
x0 =
    0.7214
```



class_assign_12,2

```
clc;clear all; format compact;clf
func='2*sin(x)-sqrt(x)+2.5';
fplot(func,[0 10])
grid on
r1=fzero(func,2)
r2=fzero(func,5.5)
r3=fzero(func,10)
hold on
plot([r1 r2 r3],[0 0 0],'ro')


r1 =
    3.4664
r2 =
    6.2869
r3 =
    9.1585
```



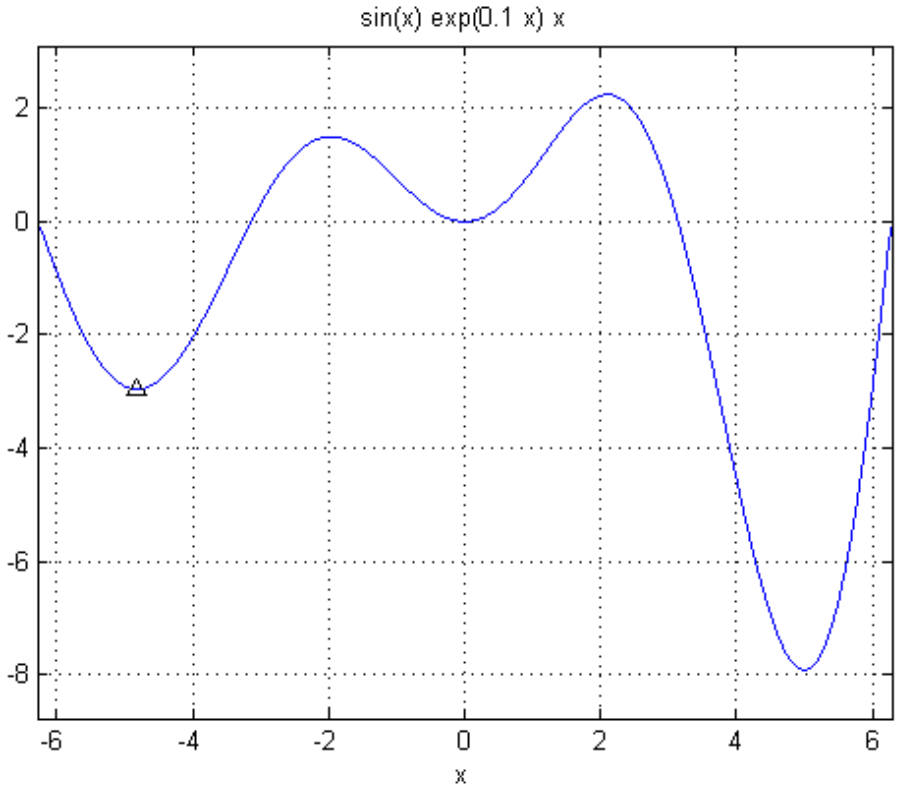
fminsearch, fminbnd

Find minimum of function

Syntax: [xmin1,ymin1] = fminsearch(fun,x0) Starts at the point x0 and finds a local minimum xmin1 and ymin1 of the function described in fun

```
clc;clear all; format compact;clf
h1='sin(x)*exp(0.1*x)*x'
 ezplot(h1)
 grid on
 hold on
 [Xmin1,Ymin1]=fminsearch(h1,-4)
 plot(Xmin1,Ymin1,'^k')
```

```
h1 =
sin(x)*exp(0.1*x)*x
Xmin1 =
   -4.8194
Ymin1 =
   -2.9594
```



[xmin1, ymin1] = fminbnd(fun,x1,x2)

Finds the minimum of a function of one variable within a fixed interval

Returns a values xmin1 and ymin1 that are a local minimizer of the function that is described in fun in the interval x1 < x < x2

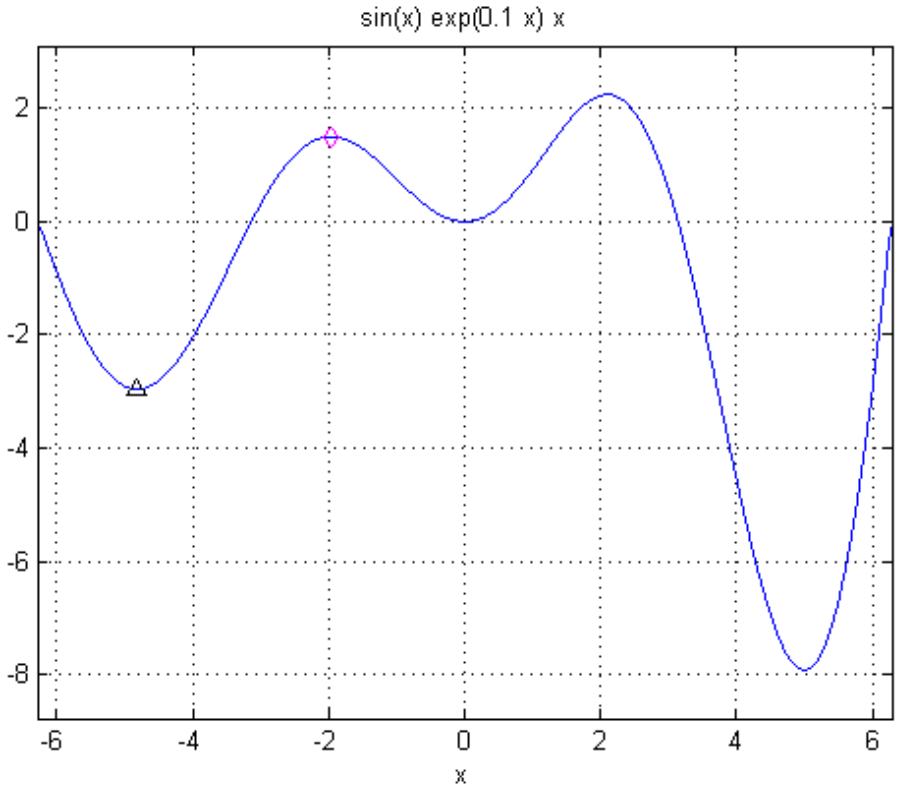
```
[Xmin2,Ymin2]=fminbnd(h1,-1,1)
```

```
Xmin2 =
  4.0389e-007
Ymin2 =
  1.6312e-013
```

maximum of a function

```
h1M=['-(',h1,')']
[Xmax1,Ymax1]=fminsearch(h1M,-2)
plot(Xmax1,-Ymax1,'dm')
```

```
h1M =
-(sin(x)*exp(0.1*x)*x)
Xmax1 =
   -1.9601
Ymax1 =
   -1.4906
```

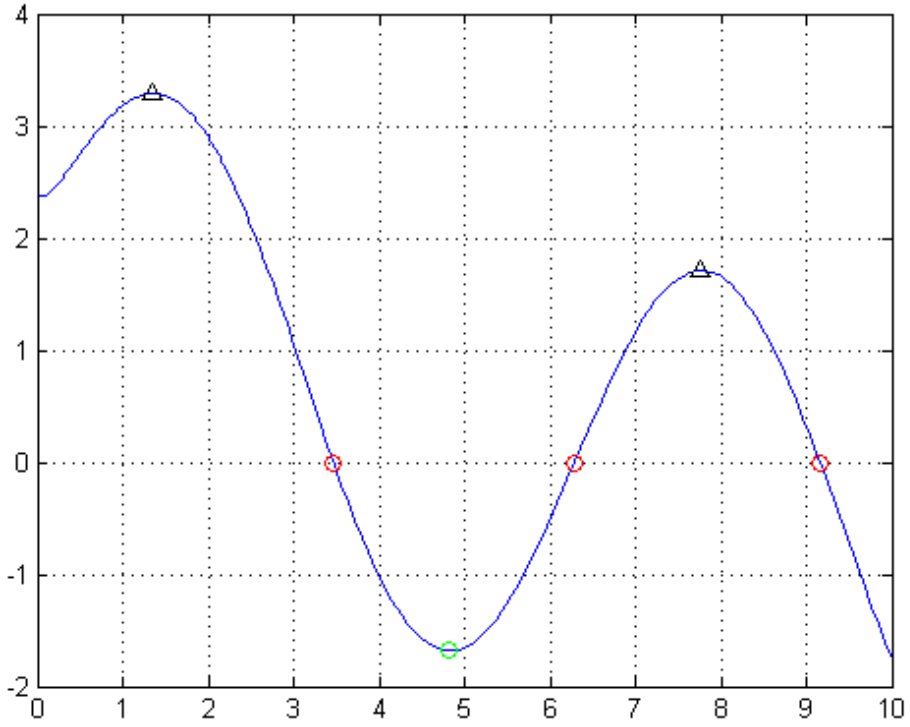


class_assign_12, 3

```
clc;clear all; format compact;clf
func='2*sin(x)-sqrt(x)+2.5';
fplot(func,[0 10])
grid on
r1=fzero(func,2)
r2=fzero(func,5.5)
r3=fzero(func,10)
hold on
plot([r1 r2 r3],[0 0 0],'ro')
[xmin1 ymin1]=fminsearch(func,4)
funcM=['-(' func ')']
[xmax1 ymax1]=fminsearch(funcM,1)
ymax1=-ymax1;
[xmax2 ymax2]=fminsearch(funcM,8)
ymax2=-ymax2;
plot([xmax1 xmax2],[ymax1 ymax2], 'k^')
plot(xmin1,ymin1,'go')
```

```
r1 =
    3.4664
r2 =
    6.2869
r3 =
    9.1585
xmin1 =
    4.8265
ymin1 =
   -1.6839
funcM =
-(2*sin(x)-sqrt(x)+2.5)
xmax1 =
    1.3543
ymax1 =
   -3.2896
xmax2 =
```

```
    7.7642
ymax2 =
   -1.7055
```



diff- Differences and approximate derivatives

syntax Y = diff(X) : calculates differences between adjacent elements of

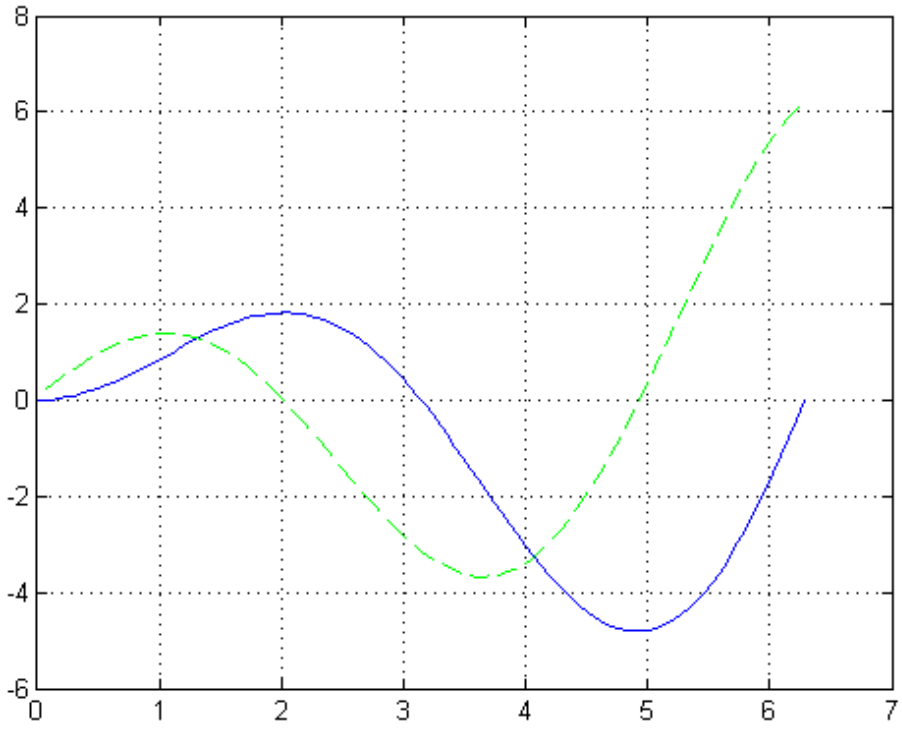
syntax: Y = diff(X,n): applies diff recursively n times

```
clc;clear all; format compact;clf
x=(1:5).^2;
dx=diff(x)
dxx=diff(dx)
dxx=diff(x,2)
```

```
dx =
     3     5     7     9
dxx =
     2     2     2
dxx =
     2     2     2
```

Example: approximate derivative of y=x.*sin(x);

```
clc,clear,clf
x=linspace(0,2*pi);
x_new=linspace(0,2*pi,99);
y=x.*sin(x);
plot(x,y),hold on
dy_dx=diff(y)./diff(x);
plot(x_new,dy_dx,'g--')
grid on
```



trapz - Trapezoidal numerical integration

Syntax: Z = trapz(X,Y) computes the integral of Y with respect to X using trapezoidal integration

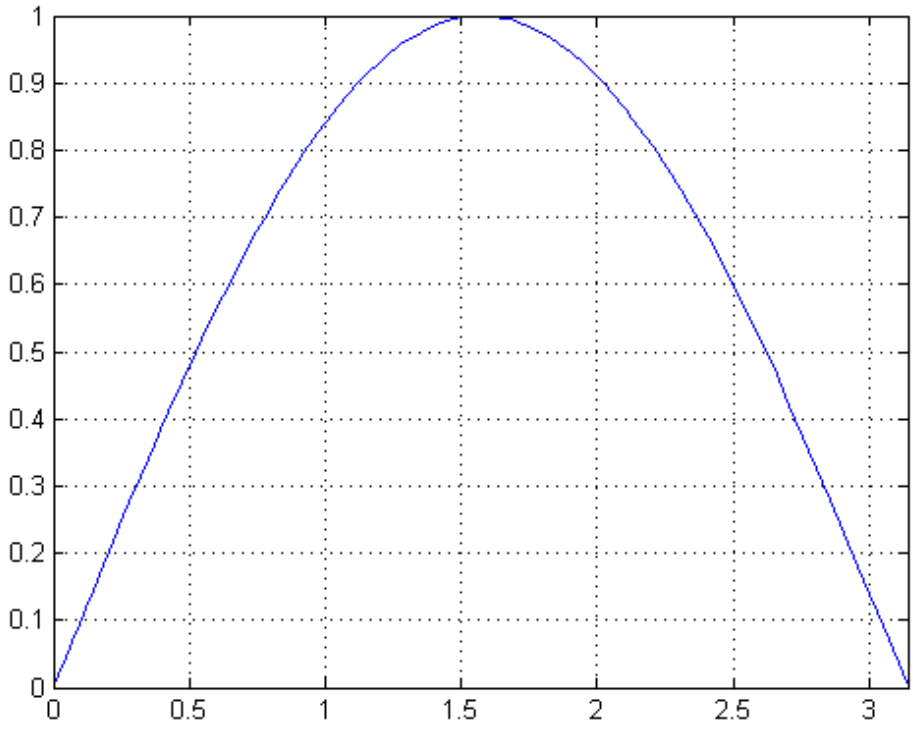
```
clc,clear all;
x=0:pi/10:pi;
y=sin(x);
z1=trapz(x,y)
```

```
z1 =
    1.9835
```

quad quadl on 'string_function'

```
clc;clear all;clf
% The function QUAD and the function QUAD8 evaluate a definite integral
%
%syntax:  intergral=quad(func,a,b)
% syntax: integral1=quadl(func,a,b)
% Remember: that any function called by QUAD WILL be called with a vector argument x. There
% you MUST write your function for x so that it works for a vector argument.
a=0;b=pi;
x=a:pi/1000:b;
y=sin(x);
z1=trapz(x,y)
f1='sin(x)'
z2=quad(f1,a,b)
z3=quadl(f1,a,b)
fplot(f1,[a,b])
grid on
```

```
z1 =
    2.0000
f1 =
sin(x)
z2 =
    2.0000
z3 =
    2.0000
```



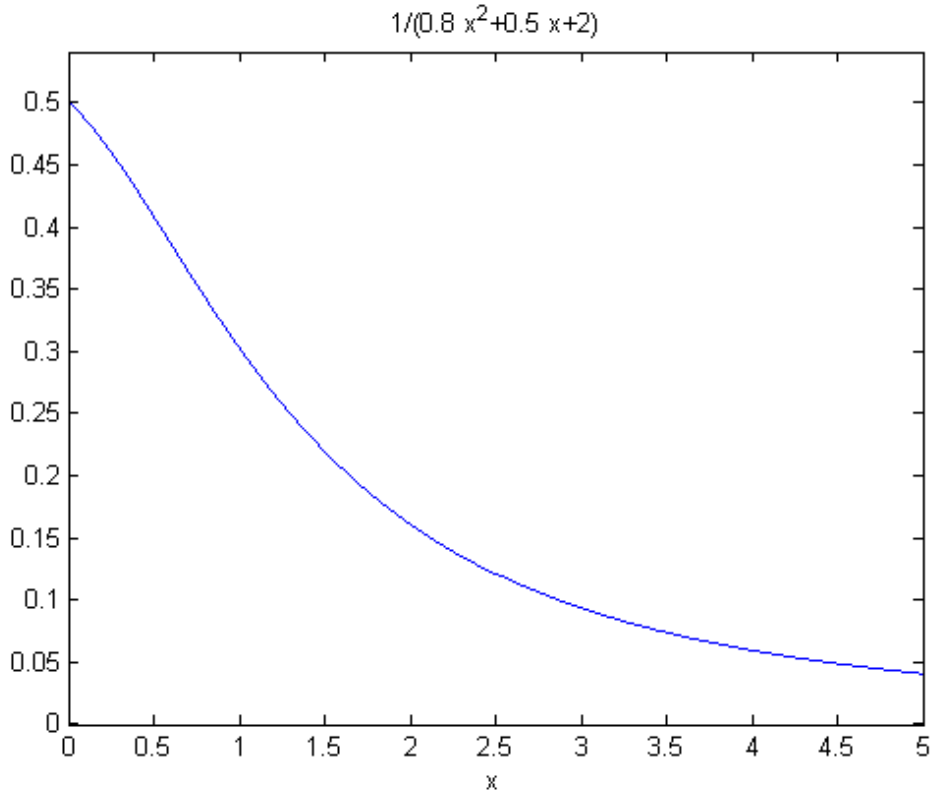
Example of an Integral

```
clc;clear all;clf;format long
a=0;b=5;
x=a:pi/1000:b;
y=1./(0.8*x.^2+0.5*x+2);
z1=trapz(x,y)
```

```
z1 =
   0.877351992117982
```

```
f1='1./(0.8*x.^2+0.5*x+2)';
z2=quad(f1,a,b)
z3=quadl(f1,a,b)
ezplot(f1,[a,b])
```

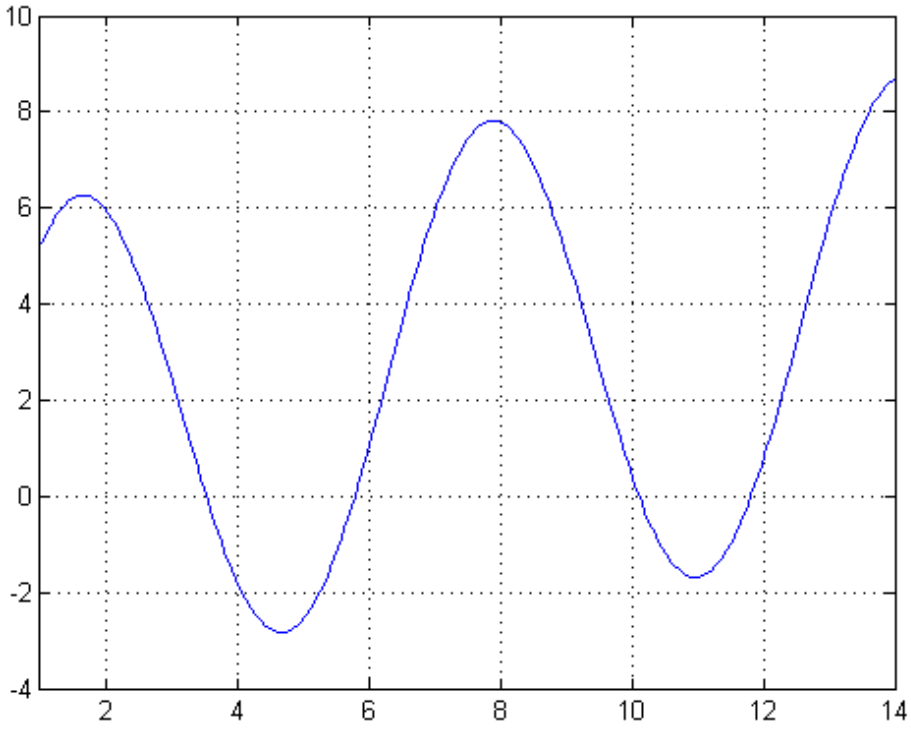
```
z2 =
   0.877421770128411
z3 =
   0.877422374618447
```



class_assign_12,4

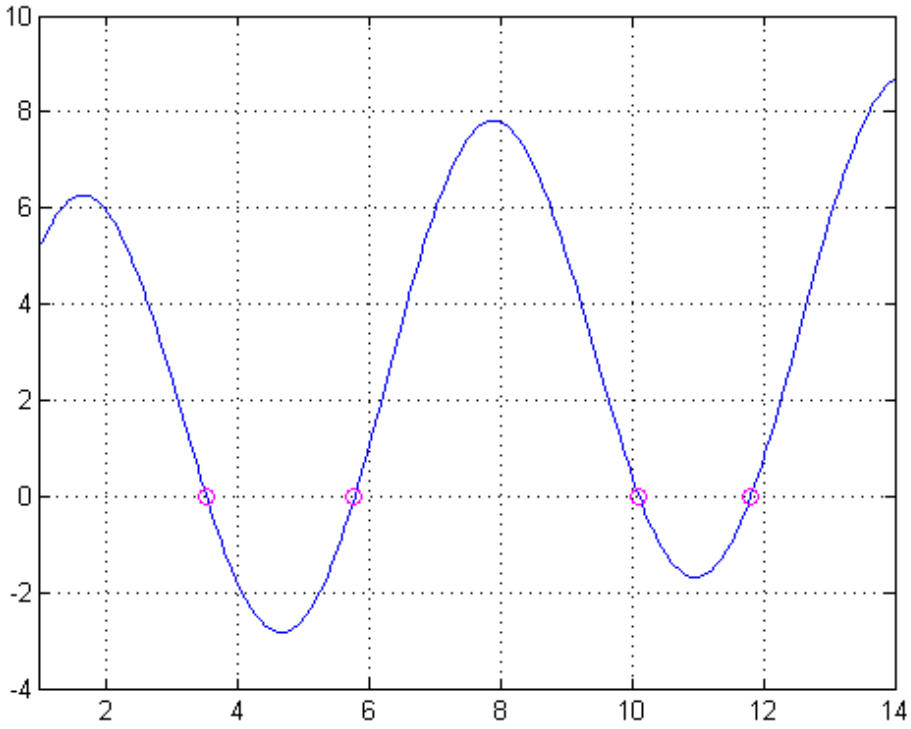
```
%%%%%%%%%%%%%%%
%a
%%%%%%%%%%%%%%%
clc;clear all;clf;format compact;
x1=1;x2=14;
f1='sqrt(x)+5*sin(x)'
fplot(f1,[1 14])
grid on
Int_f=quad(f1,1,14)


f1 =
sqrt(x)+5*sin(x)
Int_f =
  36.273294382662058
```




```
t=[2 6 10 12]
hold on
for i=1:4
    x0(i)=fzero(f1,t(i));
  plot(x0,0,'mo')
  end
s(1)=quad(f1,1,x0(1))
```

```
t =
     2     6    10    12
s =
  11.083948659005820
```




```
for j=1:3
    s(j+1)=quad(f1,x0(j), x0(j+1));
end
s(5)=quad(f1,x0(4),x2)
disp(['The area is '  num2str(sum(s))])
```

```
s =
  Columns 1 through 4
  11.083948659005820  -4.165322355987570  20.418299718768321  -1.881496781496155
  Column 5
  10.817865147580431
The area is 36.2733
```

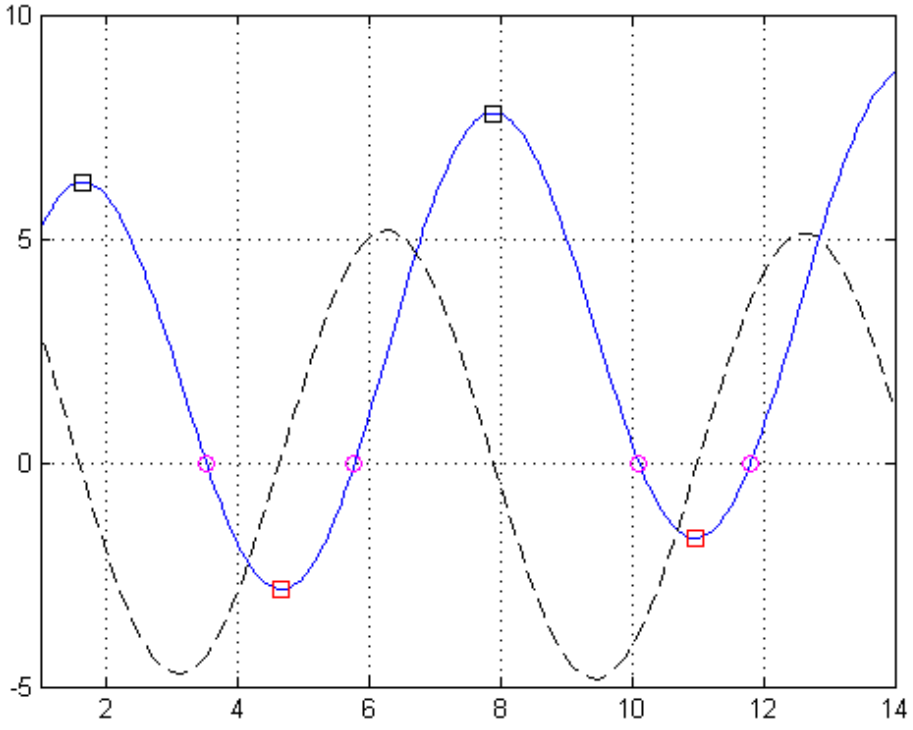
```
%%%%%%%%%%%%%%
%b
%%%%%%%%%%%%%%
hold on
minpoint=[4 10];
for i=1:2
[xmin(i) ymin(i)]=fminsearch(f1, minpoint(i));
plot(xmin(i),ymin(i),'rs')

end
minusf1=['-(' f1 ')']
maxpoint=[1.5 8];
for i=1:2
[xmax(i) ymax(i)]=fminsearch(minusf1, maxpoint(i));
plot(xmax(i),-ymax(i),'sk')
end
 %%%%%%%%%%%%%%
%c
%%%%%%%%%%%%%%

x=linspace(1,14);
dx=linspace(1,14,99);
df1=diff(sqrt(x)+5*sin(x))./diff(x);
```

```
plot(dx,df1,'k--')


minusf1 =
-(sqrt(x)+5*sin(x))
```



Published with MATLAB® 7.6