Homework #8

Math 131-05D-Spring2017

4/18/17 at 4:30pm

By Yeash Patel

Problem 1.

Consider the function$F(x) - e^{-x^2} sin(x)$. Estimate the derivative of the function at the point x=0 using each of the four formulas listed below for all of the following values of $h = 10^{-n}$, n = 1,2,… ,6.

$f`(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$ forward difference

$f`(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}$ 3-point centered difference

$f`(x_0) \approx \frac{-f(x_0+2h)+4f(x_0+h)-3f(x_0)}{2h}$ 3-point one side difference

$f`(x_0) \approx \frac{-f(x_0+2h)-8f(x_0+h)-8f(x_0-h)+f(x_0-2h)}{12h}$ 5-point centered difference

Plot your error verses h for all methods on the same graph. You may have cause to use loglog plot with the commands:

loglog(x1,y1,'-k',x2,y2,'-b',x3,y3,'-g',x4,y4,'-m','linewidth',2);

legend('FD','3pt CD','3pt 1SD','5pt CD');

Explain your findings. Is the error decreasing with h? Why or why not? Which method has the largest error? Which method has the smallest error? Why?

Please note that for the graph forward difference (FD) and 3point center difference (3PCD) are on top of each other hence why there are three lines instead of four. I feel this is due to the similarities of the two formulas. One can see the error decrease values the code displays in the command window however in the loglog graph this is oddly not the case for some reason. I would also assume that the smallest error would be form the 5-point centered difference (FPCD).

Please see appendix A problem one for the code or refer to the pr1.m in the file.

Octave input and out put

>> pr1

```
x0 = 0
warning: axis: omitting non-positive data in log plot
warning: called from
    __line__ at line 120 column 16
    line at line 56 column 8
    __plt__>__plt2vv__ at line 500 column 10
    __plt__>__plt2__ at line 246 column 14
    __plt__ at line 113 column 17
    loglog at line 60 column 10
    pr1 at line 28 column 1
xx:
 Columns 1 through 4:
  1.0000e-001  1.0000e-002  1.0000e-003  1.0000e-004
 Columns 5 and 6:
  1.0000e-005  1.0000e-006
FD:
   0.98840   0.99988   1.00000   1.00000   1.00000   1.00000
FD error:
 Columns 1 through 4:
  1.1599e-002  1.1666e-004  1.1667e-006  1.1667e-008
 Columns 5 and 6:
  1.1667e-010  1.1666e-012
3PCD
   0.98840   0.99988   1.00000   1.00000   1.00000   1.00000
3PCD error:
```

Columns 1 through 4:

1.1599e-002 1.1666e-004 1.1667e-006 1.1667e-008

Columns 5 and 6:

1.1667e-010 1.1666e-012

3POSD

1.0224   1.0002   1.0000   1.0000   1.0000   1.0000

3POSD error:

Columns 1 through 4:

2.2404e-002 2.3324e-004 2.3333e-006 2.3333e-008

Columns 5 and 6:

2.3333e-010 2.3335e-012

5PCD

0.99974   1.00000   1.00000   1.00000   1.00000   1.00000

5PCD error:

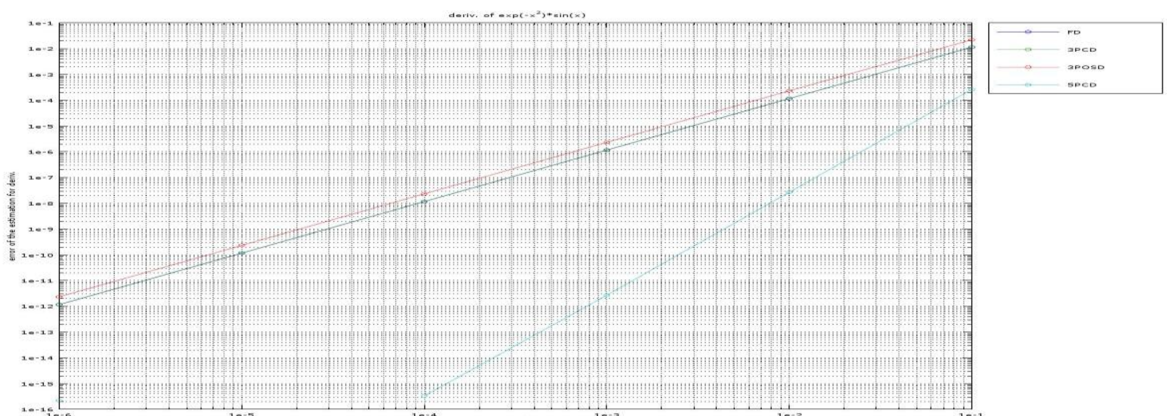
Columns 1 through 4:

2.6489e-004 2.6995e-008 2.7001e-012 3.3307e-016

Columns 5 and 6:

0.0000e+000 2.2204e-016

exact: 1



Problem 2.

Create a function file called trap_int.m that inputs a function f, a pair of endpoints, a;b, and a number n of subintervals, and outputs the approximation to the integral of f from a to b using the trapezoid rule on n points. Your function header should look like this function I = trap_int(f,a,b,n)

> To display the use of this we will attempt to estimate the integral $\int_0^4 x^2\,dx$ with four divisions. We know that the exact value of this integral will be about 21.3333 out function returned 22 which I found to be acceptable for the composite trapezoid method.
>
> Please see appendix A problem two for the code or refer to the trap_int.m in the file.

Octave input and out put

```
>> trap_int(@(x) x^2,0,4,4)

ans =  22
```

Problem 3

Create a function file called Simp_int.m that inputs a function f, a pair of endpoints, a;b, and a number n of subintervals, and outputs the approximation to the integral of f from a to b using Simpsons rule on n points. Your function header should look like this function I = Simp_int(f,a,b,n).

> To display the use of this we will attempt to estimate the integral $\int_0^4 x^2\,dx$ with four divisions. We know that the exact value of this integral will be about 21.3333 out function returned 21.333 which is accurate and expected of the composite Simpson method.
>
> Please see appendix A problem three for the code or refer to the Simp_int.m in the file.

Octave input and out put

```
>> Simp_int(@(x) x^2,0,4,4)

ans =  21.333
```

Problem 4

Consider the integral

$$I = \int_{-2}^{2} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

(a) Use the code you developed in Problems 2 and 3 to compute the integral for the number of points n = 10; 100; 1000; 10000.

(b) Compute the error in your computation for each n and for each method. Note that the ’exact’ answer to the given integral can be computed by using the error function erf in MATLAB, namely,

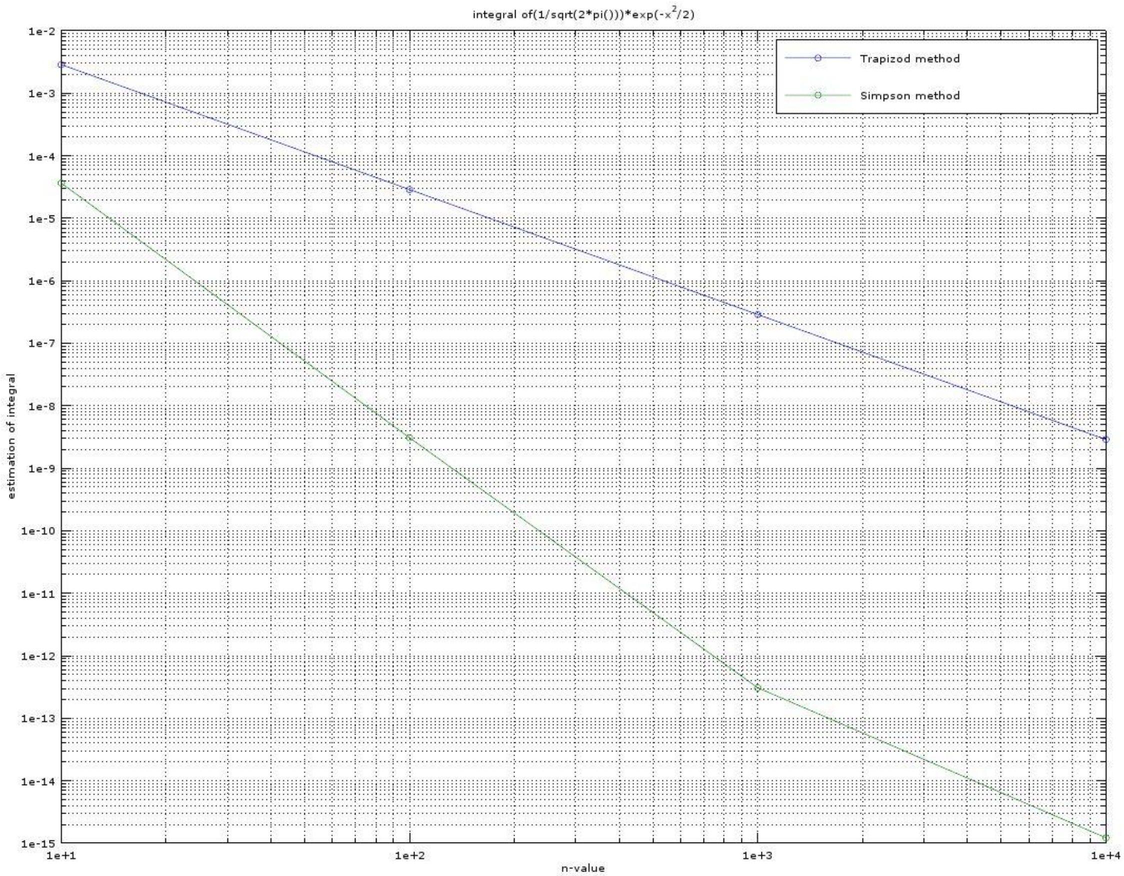
$$I_{exact} = \frac{\mathrm{erf}\,(\sqrt{2} - \mathrm{erf}\,(-\sqrt{2})}{2}$$

(c) Make a loglog plot of error verses the number of points for all methods on the same plot. Make sure to create a legend, showing which plot corresponds to which method. Comment on your results: which method works better? Why?

> For ease I decided to write a script that will call on the function used in problems 2 and 3 to answer this problem. Simpsons tends to work better because it uses a smooth curve rather than the linear lines of a trapezoid method therefore giving having less error.
>
> Please see appendix A problem four for the code or refer to the pr4.m in the file.

Octave input and out put

```
>> pr4

for part A:

  Trapizod method:

   0.95163   0.95447   0.95450   0.95450

  Simpson method:

   0.95446   0.95450   0.95450   0.95450

exact value:

 0.95450
```


integral of(1/sqrt(2*pi()))*exp(-x^2/2)
Trapizod method
Simpson method
estimation of integral
n-value
1e-2
1e-3
1e-4
1e-5
1e-6
1e-7
1e-8
1e-9
1e-10
1e-11
1e-12
1e-13
1e-14
1e-15
1e+1
1e+2
1e+3
1e+4

Appendix A

PROBLEM 1

```
Pr1.m
clc
clear all
close all
f=@(x) exp(-x^2)*sin(x);
df=@(x) exp(-x^2)*cos(x)-2*x*exp(-x^2)*sin(x);

%x0=input('x0:')
x0=0

for n=1:6
  h=10^(-n);
  xx(n)=h;

  fowdiffnum(n)=(f(x0+h)-f(x0))/h;
  error1(n)=abs(fowdiffnum(n)-df(x0));

  ThreePointCentDiff(n)=(f(x0+h)-f(x0-h))/(2*h);
  error2(n)=abs(ThreePointCentDiff(n)-df(x0));

  ThreePointOneSideDiff(n)=(-f(x0+2*h)+4*f(x0+h)-3*f(x0))/(2*h);
  error3(n)=abs(ThreePointOneSideDiff(n)-df(x0));

  FivePointCentDiff(n)=(-f(x0+2*h)+8*(f(x0+h))-8*(f(x0-h))+(f(x0-2*h)))/(12*h);
                    %(-f(x0+2*h)+8*(f(x0+h))-8*(f(x0-h))+(f(x0-2*h)))/(2*h)
  error4(n)=abs(FivePointCentDiff(n)-df(x0));
end

loglog(xx,error1,'-o',xx,error2,'-o',xx,error3,'-o',xx,error4,'-o')
legend('FD','3PCD','3POSD','5PCD',"location", "northeastoutside");
grid on
xlabel('xx-value')
ylabel('error of the estimation for deriv.')
title(' deriv. of exp(-x^2)*sin(x)')


disp('xx:')
disp(xx)
disp('FD:')
disp(fowdiffnum)
disp('FD error:')
disp(error1)
disp('3PCD')
```

```
disp(ThreePointCentDiff)
disp('3PCD error:')
disp(error2)
disp('3POSD')
disp(ThreePointOneSideDiff)
disp('3POSD error:')
disp(error3)
disp('5PCD')
disp(FivePointCentDiff)
disp('5PCD error:')
disp(error4)
disp('exact:')
disp(df(x0))
```

PROBLEM 2

trap_int.m

```
function [I] = trap_int (f,a,b,n)
x=linspace(a,b,(n+1));
h=(b-a)/n;
part1=f(a)+f(b);
for i=1:n
  if i==1
    part2=0;
   endif
  if i==n
    break;
  endif
   part2=(f(x(i+1)))+part2;
endfor
I=h/2*(part1+2*part2);

endfunction
```

PROBLEM 3

Simp_int.m

```
function [I] = Simp_int (f,a,b,n)
x=linspace(a,b,(n+1));
h=(b-a)/n;
part1=f(a)+f(b);
for i=1:n
  if i==1
    part2=0;
    part3=0;
```

```
   endif
  if i==n
    break;
  endif
  if rem(i,2)==0
    part2=part2+f(x(i+1));
  endif
  if rem(i,2)!=0
    part3=part3+f(x(i+1));
  end
endfor
 I=h/3*(part1+2*part2+4*part3);
endfunction
```

PROBLEM 4

pr4.m

```
clc
clear all
close all
f= @(x) (1/sqrt(2*pi()))*exp(-x^2/2);
a=-2;
b=2;
for i=1:4
n=10^i;
xx(i)=n;

trap(i)=trap_int (f,a,b,n);
simp(i)=Simp_int (f,a,b,n);
endfor
disp('for part A:')
disp('  Trapizod method: ')
disp(trap)
disp('  Simpson method: ')
disp(simp)
exact=(erf(sqrt(2))-erf(-sqrt(2)))/2;
disp('exact value:')
disp(exact)
for j=1:4

error1(j)=abs(trap(j)-exact);
error2(j)=abs(simp(j)-exact);
endfor
loglog(xx,error1,'-o',xx,error2,'-o')
legend('Trapizod method','Simpson method')
grid on
```

```
xlabel('n-value')
ylabel('estimation of integral')
title('integral of(1/sqrt(2*pi()))*exp(-x^2/2)')
```