University of Colorado
Department of Computer Science

Numerical Computation CSCI 3656

Spring 2016

Problem Set 11 Solutions

1. *[10 pts]* Apply Richardson extrapolation starting from $h = 0.1$ to get $f'(1)$ for $f(x) = e^x + 0.5$ and using center differences. Compare the resulting estimate to the true value and to the results that you obtained for this function with that method in PS10.

   Here is some MATLAB Code for the first derivative using Richardson Extrapolation:

```
function approx = richardson(f, x, h)
    step = h; n = 2;
    outerSlope = (f(x+step) - f(x-step)) / (2*step);
    step = step/2;
    innerSlope = (f(x+step) - f(x-step)) / (2*step);
    approx = (innerSlope) + ( (1/((2^n)-1)) * (innerSlope - outerSlope) );
end
```

   See also example 5.4 on page 250 of the textbook. $f'(1) = 2.71828126$. This is correct to six decimal places! The value obtained in PS10 using centered differences, with the same h, yields $f'(1) = 2.71832713$.

2. In your favorite programming language, write a program that uses the trapezoid rule to compute $\int_{x_1}^{x_2} f(x)dx$ from a data set.

   Your program should take the following inputs:

   - a series of (evenly spaced) values for x
   - a series of values for $f(x)$ at those x values
   - a lower bound x_1 for the integration
   - an upper bound x_2 for the integration
   - a step size h, which must be a multiple of the sample interval in the data table

   The x and $f(x)$ can be in individual lists, or together in a table or matrix or whatever data structure you want, but please explain your choice in a comment. Your program should complain if the data are not evenly spaced, and it should not allow the user to specify bound values that fall between the table entries —

i.e., $x = 1.03$ is not a legal lower-bound argument for the table in part (a) below (otherwise the integration gets difficult).

The output should be a single number: the value $\int_{x_1}^{x_2} f(x)dx$ computed using panels built from points that are h apart. Please turn in a copy of your code.

See the last page of these solutions for the code.

(a) *[10 pts]* Use your code to compute $\int_{1.0}^{1.8} f(x)dx$ from the table at the top of the next page *using* $h = 0.1$. (These data are in a file on the course webpage if you don't want to type them in.)

Trapezoidal rule with a $h = 0.1$ gives 3.73414131 as an estimate for the integral.

x	$f(x)$	x	$f(x)$
0.9000	2.9596	1.4000	4.5552
0.9500	3.0857	1.4500	4.7631
1.0000	3.2183	1.5000	4.9817
1.0500	3.3577	1.5500	5.2115
1.1000	3.5042	1.6000	5.4530
1.1500	3.6582	1.6500	5.7070
1.2000	3.8201	1.7000	5.9739
1.2500	3.9903	1.7500	6.2546
1.3000	4.1693	1.8000	6.5496
1.3500	4.3547	1.8500	6.8598
		1.9000	7.1859

(b) *[10 pts]* Repeat part (a) of this problem with $h = 0.2$.

Trapezoidal rule with a $h = 0.2$ gives 3.74246279 as an estimate for the integral.

3. *[10 pts]* Repeat the previous problem using Simpson's 1/3 rule. This should only require a few minor modifications of the code. Don't forget to handle the problem that arises if you don't have the right number of data points to divide neatly into the three-point-wide panels (with points spaced h apart) that this method uses. Please also turn in a copy of this code.

   Using an h of 0.1, Composite Simpson's yields an estimate of 3.73136748. Using an h of 0.2, it yields 3.73139511.

4. *[10 pts]* The function tabulated above is actually the same one as in the first problem above ($f(x) = e^x + 0.5$).

   (a) Using calculus, figure out what $\int_{1.0}^{1.8} f(x)dx$ *should* be.

   $\int_1^{1.8}(e^x + 0.5)dx = e^x|_1^{1.8} + 0.5x|_1^{1.8} = 3.73136564$.

(b) Comment on how close your four answers from parts (a) and (b) of problems 2 and 3 are to this number. Which one of each (a)/(b) pair is better? Why? Is Simpson's 1/3 rule *generally* better than trapezoidal, for the same h?

The estimates generated with a smaller h are better than those generated with a larger h. Simpson's rule is closer than trapezoidal rule for a given h. Simpson's rule is generally better because the error is $O(h^4)$ for composite Simpson's and $O(h^2)$ for composite trapezoidal.

Here is the code:

```
For composite trapezoidal rule

function Trapezoid = Trapezoid(x,y,x1,x2,h)

xr = round(x,5);
sigma = 0;
index = find(xr == x1,1);
stop = find(xr == x2,1);
step = h/(x(2) - x(1));

for i = round(index + step):round(step):round(stop - step)
sigma = sigma + y(i);
end

Trapezoid = h/2*(y(index) + y(stop) + 2*sigma);
end
```

```
For composite Simpson's Rule

function Simp = Simpsons(x,y,x1,x2,h)

xr = round(x,5);
sigma1 = 0;
sigma2 = 0;
index = find(xr == x1,1);
stop = find(xr == x2,1);
step = h/(x(2) - x(1));

for i = round(index + step):round(2*step):round(stop - step)
sigma1 = sigma1 + y(i);
end

for k = round(index + 2*step):round(2*step):round(stop - 2*step)
sigma2 = sigma2 + y(k);
end

Simp = h/3*(y(index) + y(stop) + 4*sigma1 + 2*sigma2);
end
```