Math 104B: Homework 2

Raghav Thirumulu, Perm 3499720
rrajuthirumulu@umail.ucsb.edu

August 22, 2018

1.

```
function integral = composite_trap(a,b,h,f)
% Computer code for approximating the integral of a function using CTR
% Input:  a        --- lower bounds for the integral
%         b        --- upper bounds for the integral
%         h        --- width of subintervals
%         f        --- the desired function
% Output: integral --- approximation of integral
% Author: Raghav Thirumulu, Perm 3499720
% Date:   08/21/2018

result=0;
% Calculate the x values we want to use
x=[a+h:h:b-h];

% Calculate number of intervals for iteration
n=(b-a)/h;

% Iterate through the
for i=1:n-1
    result=result+f(x(i));
end

% Edit values so they are representative of CTR
result=h*(result+0.5*(f(a)+f(b)));
integral=result;
end
```

```
function integral = simpson(a,b,h,f)
% Computer code for approximating the integral of a function using Simpson
% Input:  a        --- lower bounds for the integral
%         b        --- upper bounds for the integral
%         h        --- width of subintervals
% Output: integral --- approximation of integral
% Author: Raghav Thirumulu, Perm 3499720
% Date:   08/21/2018

result=0;

% Calculate the x values we want to use
x=[a:h:b];

% Edit values so they are representative of Simpson's
result=h/3*(f(x(1))+2*sum(f(x(3:2:end-2)))+4*sum(f(x(2:2:end)))+f(x(end)));
integral=result;

end
```

2.

h	$T_h^{[0,1]}$	$S_h^{[0,1]}$
0.1	0.7462107961	0.7468249482
0.05	0.7466708369	0.7468241838
0.025	0.7467858112	0.7468241360
0.0125	0.7468145525	0.7468241330

The actual value of $\int_0^1 e^{-x^2}$ is 0.7468241328.

For Composite Trapezoidal rule:

$$E_{0.1}^{[0,1]} = |0.7462107961 - 0.7468241328| = 0.0006133367$$
$$E_{0.05}^{[0,1]} = |0.7466708369 - 0.7468241328| = 0.0001532959$$
$$E_{0.025}^{[0,1]} = |0.7467858112 - 0.7468241328| = 0.0000383216$$
$$E_{0.0125}^{[0,1]} = |0.7468145525 - 0.7468241328| = 0.0000095803$$

We can see that the error term reduces by approximately a factor of 4 each time h changes by a factor of 2. Therefore, the rate of convergence is $O(h^2)$.

For Simpson's rule:

$$E_{0.1}^{[0,1]} = |0.7468249482 - 0.7468241328| = 0.0000008154$$
$$E_{0.05}^{[0,1]} = |0.7468241838 - 0.7468241328| = 0.0000000510$$
$$E_{0.025}^{[0,1]} = |0.7468241360 - 0.7468241328| = 0.0000000032$$
$$E_{0.0125}^{[0,1]} = |0.7468241330 - 0.7468241328| = 0.0000000002$$

We can see that the error term reduces by approximately a factor of 16 each time h changes by a factor of 2. Therefore, the rate of convergence is $O(h^4)$.

3.

```
function [R,error,levels]=romberg(f,a,b,tol)
% Computer code for utilizing the Romberg algorithm to determine
% number of levels needed to achieve a certain error threshold
% Input:  f        --- the desired function
%         a        --- lower bounds of integral
%         b        --- upper bounds of integral
%         tol      --- desired user tolerance
% Output: R        --- approximation of integral
%         error    --- the approximate error we achieve
%         levels   --- # of levels needed for error to be less than tol
% Author: Raghav Thirumulu, Perm 3499720
% Date:   08/21/2018

% Set a really high default max iterations value
max_iterations=100;

% Use only one sub-interval to begin
n = 1;

% Use our CTR formula from question 1, set default levels to 0
I(1,1) = composite_trap(a,b,(b-a)/n,f);
levels = 0;

% Romberg algorithm, increase our levels till our error term goes
% under the value of the tolerance
while levels<max_iterations
    levels = levels+1;

    % Calculate number of subintervals depending on levels
    n = 2^levels;
    % Compute CTR for the correct level
    I(levels+1,1) = composite_trap(a,b,(b-a)/n,f);
    for k = 2:levels+1
        j = 2+levels-k;
        I(j,k) = (4^(k-1)*I(j+1,k-1)-I(j,k-1))/(4^(k-1)-1);
    end
    error = abs((I(1,levels+1)-I(2,levels))/I(1,levels+1))*100;
    if error<=tol, break; end
```

```
    end

    R = I(1,levels+1);
```

The actual value of $\int_{-1}^{1} e^x$ is 2.350402387.

For a tolerance of 10^{-6}, the approximation using the Romberg algorithm is 2.350402387329692, the approximate error is 1.773373533852801e-08, and the number of levels we need are 4.
For a tolerance of 10^{-8}, the approximation using the Romberg algorithm is 2.350402387287607, the approximate error is 1.757158550358556e-12, and the number of levels we need are 5.
For a tolerance of 10^{-10}, the approximation using the Romberg algorithm is 2.350402387287607, the approximate error is 1.757158550358556e-12, and the number of levels we need are 5.