Table of Contents

section 0 - clean up

```
clear all;
close all;
clc

myFig = figure;
myFig.Visible = "off";
subplot(2,1,1)
hold on
title("function 1 error")
xlabel('n')
ylabel('error')
a = gca;
a.YScale = 'log';

subplot(2,1,2)
hold on
title("function 2 error")
xlabel('n')
ylabel('error')
a = gca;
a.YScale = 'log';
```

section 1 - use composite trap for f1

```
%clear all;

f1 = @(x) x/((x^2+9)^.5); % this is the given f1
F1 = @(x1,x2) (x2^2+9)^.5 - (x1^2+9)^.5; % this is the true definite integral
 of f1

a = 0; % set bounds
b = 4;
realSol = F1(a,b); % find real solution

n = 2; % use method to aproximate integral at different n values
I02 = CompositeTrapezoid(f1,a,b,n);
n = 4;
I04 = CompositeTrapezoid(f1,a,b,n);
```

```
n = 8;
I08 = CompositeTrapezoid(f1,a,b,n);
n = 16;
I16 = CompositeTrapezoid(f1,a,b,n);
n = 32;
I32 = CompositeTrapezoid(f1,a,b,n);

e(1) = I02 - realSol; % find the errors for each approximation
e(2) = I04 - realSol;
e(3) = I08 - realSol;
e(4) = I16 - realSol;
e(5) = I32 - realSol;

e = abs(e);
subplot(2,1,1)
plot([2,4,8,16,32],e)
```

section 2 - use composite trap for f2

```
%clear all;

f2 = @(x) cos(x)*exp(x);% this is the given f2
F2 = @(x1,x2) (exp(x2)/2)*(sin(x2)+cos(x2)) - (exp(x1)/2)*(sin(x1)+cos(x1));%
 this is the true definite integral of f2

a = 0; % set bounds
b = 2*pi;
realSol = F2(a,b); % find real solution

n = 2; % use method to aproximate integral at different n values
I02 = CompositeTrapezoid(f2,a,b,n);
n = 4;
I04 = CompositeTrapezoid(f2,a,b,n);
n = 8;
I08 = CompositeTrapezoid(f2,a,b,n);
n = 16;
I16 = CompositeTrapezoid(f2,a,b,n);
n = 32;
I32 = CompositeTrapezoid(f2,a,b,n);

e(1) = I02 - realSol; % find the errors for each approximation
e(2) = I04 - realSol;
e(3) = I08 - realSol;
e(4) = I16 - realSol;
e(5) = I32 - realSol;

e = abs(e);
subplot(2,1,2)
plot([2,4,8,16,32],e)
```

section 3 - use composite simpsons for f1

```
%clear all;
```

```
f1 = @(x) x/((x^2+9)^.5);% this is the given f1
F1 = @(x1,x2) (x2^2+9)^.5 - (x1^2+9)^.5;% this is the true definite integral
 of f1

a = 0; % set bounds
b = 4;
realSol = F1(a,b); % find real solution

n = 2; % use method to aproximate integral at different n values
I02 = CompositeSimpson(f1,a,b,n);
n = 4;
I04 = CompositeSimpson(f1,a,b,n);
n = 8;
I08 = CompositeSimpson(f1,a,b,n);
n = 16;
I16 = CompositeSimpson(f1,a,b,n);
n = 32;
I32 = CompositeSimpson(f1,a,b,n);

e(1) = I02 - realSol; % find the errors for each approximation
e(2) = I04 - realSol;
e(3) = I08 - realSol;
e(4) = I16 - realSol;
e(5) = I32 - realSol;

e = abs(e);
subplot(2,1,1)
plot([2,4,8,16,32],e)
```

section 4 - use composite simpsons for f2

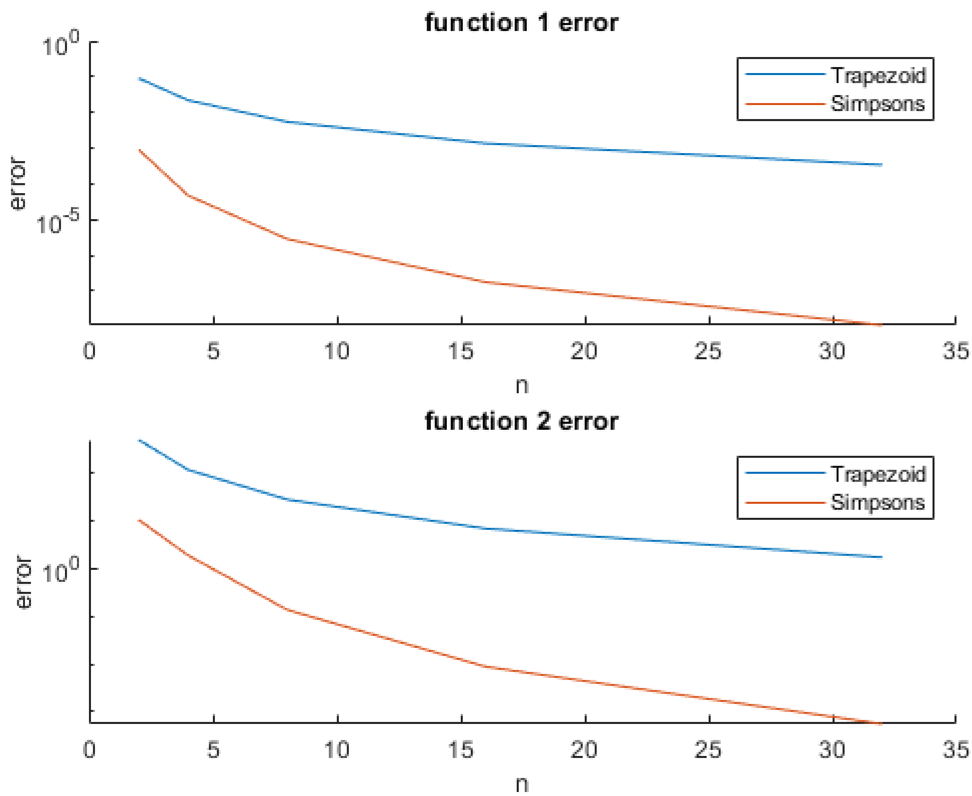
```
%clear all;

f2 = @(x) cos(x).*exp(x);% this is the given f2
F2 = @(x1,x2) (exp(x2)/2)*(sin(x2)+cos(x2)) - (exp(x1)/2)*(sin(x1)+cos(x1));%
 this is the true definite integral of f2

a = 0; % set bounds
b = 2*pi;
realSol = F2(a,b); % find real solution

n = 2; % use method to aproximate integral at different n values
I02 = CompositeSimpson(f2,a,b,n);
n = 4;
I04 = CompositeSimpson(f2,a,b,n);
n = 8;
I08 = CompositeSimpson(f2,a,b,n);
n = 16;
I16 = CompositeSimpson(f2,a,b,n);
n = 32;
I32 = CompositeSimpson(f2,a,b,n);
```

```
e(1) = I02 - realSol; % find the errors for each approximation
e(2) = I04 - realSol;
e(3) = I08 - realSol;
e(4) = I16 - realSol;
e(5) = I32 - realSol;

e = abs(e);
subplot(2,1,2)
plot([2,4,8,16,32],e)
```

section 5 - plot finishing

```
myFig.Visible = "on";
subplot(2,1,1)
legend('Trapezoid','Simpsons')
subplot(2,1,2)
legend('Trapezoid','Simpsons')

disp("f1 is f = x/((x^2+9)^.5)")
disp("f2 is f = cos(x)*exp(x)")
```

```
f1 is f = x/((x^2+9)^.5)
f2 is f = cos(x)*exp(x)
```



section 6 - discuss differences (Problem 2 f)

```
disp("In both cases (f1 and f2) Simpsons method converges more quickly than
 trapezoid, this makes sense since Simpsons method is known to be of higher
 order than Trapezoid.")
```

```
In both cases (f1 and f2) Simpsons method converges more quickly than
 trapezoid, this makes sense since Simpsons method is known to be of higher
 order than Trapezoid.
```

section 7 - function definition

```
function [I] = CompositeTrapezoid(f, a, b, n)
%Approximates the integral of a function using the Composite Trapezoid Rule
%   f: function to be integrated
%   a: lower bound of interval
%   b: upper bound of interval
%   n: number of panels used for the approximation
h = (b-a)/n; %width of a panel
x = linspace(a,b,n+1); %Create n+1 equally spaced points for the n panels
I = f(a) + f(b);
for i=1:n-1
        I = I + 2*f(x(i+1));
end
I = I*h/2;
end

function [I] = CompositeSimpson(f, a, b, n)
%Approximates the integral of a function using the Composite Simpson's Rule
%   f: function to be integrated
%   a: lower bound of interval
%   b: upper bound of interval
%   n: number of panels used for the approximation
h = (b-a)/(2*n); %width of a subinterval
x = linspace(a,b,2*n+1); %Create 2n+1 equally spaced points for the n panels
I = f(a) + f(b);
for i=1:n-1
        I = I + 2*f(x(2*i+1)) + 4*f(x(2*i));
end
I = (I + 4*f(x(2*n)))*h/3;
end
```

Published with MATLAB® R2022a