# Lab 07 Mowen

## Table of Contents



Notation Definition: Integral: (f, xi, xf) |-> Integral(f, xi, xf) : (Fnct(R,R) X R X R) -> R where f: a function from R to R, xi: an initial value of x in the real numbers, xf: a final value of x in the real numbers,

# Problem 1

Approximate Integral( 1/(xlogx), e, e+2 ) (a) using the Composite Trapezoidal rule with n = 8. (b) using the Composite Simpson's rule with n = 8.

```
fprintf('Problem 1\n')
f = @(x) 1 / (x * log(x));
x0 = exp(1);
xn = exp(1) + 2;
n = 8;
trapezoidalApproximation = CompositeTrapezoidalRule(f, x0, xn, n);
fprintf('(a) Composite Trapezoidal Rule with n = %d: %0.6f\n', ...
        n, trapezoidalApproximation)

simpsonsApproximation = CompositeSimpsonsRule(f, x0, xn, n);
fprintf('(b) Composite   Simpson''s Rule with n = %d: %0.6f\n\n', ...
        n, simpsonsApproximation)
```

```
Problem 1
```

# Problem 2

To simulate the thermal characteristics of disc brakes, D. A. Secrist and R. W. Hornbeck needed to approximate numerically the "area averaged lining temperature," T, of the brake pad from the equation T = (Integral( T(r)*r*thetaP, re, ro )) / (Integral( T(r)*r, re, ro)) where re represents the radius at which the pad-disk contact begins, ro represents the outside radius of the pad-disk contact, thetaP represents the angle subtended by the sector brake pads, and T(r) is the temperature at each point of the pad, obtained numerically from analyzing the heat equation. Suppose re = 0:308 ft, ro = 0:478 ft, thetaP = 0:7051 radians, and the temperatures given in the data matrix have been calculated at the various points on the disk. Approximate T using the Composite Trapezoidal rule.

```
re = 0.308;      % ft. radius at which pad-disk coverage begins
 contact
ro = 0.478;      % ft. outside radius of the pad-disk contact
thetaP = 0.7051; % radians. angle subtented by the sector brak pads
```

```
%                        r    |  T(r) (deg F)
%                     ----------------
tempAtVariousRadius = [    re,  640;
                        0.325,  794;
                        0.342,  885;
                        0.359,  943;
                        0.376, 1034;
                        0.393, 1064;
                        0.410, 1114;
                        0.427, 1152;
                        0.444, 1204;
                        0.461, 1222;
                           ro, 1239];

fprintf('Problem 2\n')

h = tempAtVariousRadius(2:end,1) - tempAtVariousRadius(1:end-1,1);
n = length(h);
h = h(1);

% Composite Trapezoidal Rule to approximate integral in numerator
numeratorApprox = 0;
for i = 2: n
    r = tempAtVariousRadius(i,1);
    Tofr = tempAtVariousRadius(i,2);
    numeratorApprox = numeratorApprox + Tofr * r * thetaP;
end
Tinit = tempAtVariousRadius(1,2);
Tfinal = tempAtVariousRadius(n+1,2);
numeratorApprox = re*Tinit*thetaP + 2*numeratorApprox +
 ro*Tfinal*thetaP;

% Composite Trapezoidal Rule to approximate integral in denominator
denominatorApprox = 0;
for i = 2: n
    r = tempAtVariousRadius(i,1);
    denominatorApprox = denominatorApprox + r * thetaP;
end
denominatorApprox = re*thetaP + 2*denominatorApprox + ro*thetaP;

% Compute area averaged lining temperature using numerator
 approximation
% and denominator approximation.
AreaAveragedLiningTemperature = numeratorApprox/denominatorApprox;

fprintf('The area averaged lining temperature is %.0f degrees
 Fahrenheit\n\n',...
    AreaAveragedLiningTemperature)
```

```
Problem 2
The area averaged lining temperature is 1055 degrees Fahrenheit
```

# Problem 3

The Gateway Arch in St. Louis was constructed using the equation y = 211.49 - 20.96*(cosh(0.03291765*x)) for the central curve of the arch, where x and y are measured in meters and |x| <= 91.20. Setup an integral for the length of the central curve, and approximate the length to the nearest meter using the Composite Simpson's rule (with an even n required to achieve this accuracy from the error bound).

```
fprintf('Problem 3\n');
% Since |x| <= 91.20, it follows x in [-91.20, 91.20], which implies
%   xi = -91.20 and xf = 91.20
xi = -91.20;
xf = 91.20;

% To find the length of the curve given by
%   f(x) = y = 211.49 - 20.96*(cosh(0.03291765*x))
% we can use the arc length formula given by
%   ds = sqrt(1 + (dy/dx)^2)
%   s = Integral( ds , xi, xf)
syms x;
y = 211.49 - 20.96*(cosh(0.03291765*x));    % construction equation
 (ce)
dydx = diff(y);                             % derivative of ce
ds = sqrt(1 + dydx^2);                      % arc length of ce

% Need 4th derivative to determine error bound
dds4dx4 = diff(diff(diff(diff(ds))));       % 4th derivative of ds
% turn sym function to function handle
dds4dx4 = matlabFunction(dds4dx4);          % evaluatable 4th deriv of
 ds

% In order to approximate our answer to within 1 meter,
%   | ((xf-xi)/180) * (h^4) * ( ds(4)(ksi)) | <= 1
% => h <= ( 180 /( (xf-xi) * (ds(4)(ksi)) ) )^(1/4) and h = (xf-xi)/n
% => n >= (xf-xi)/h
% => n = ceil((xf-xi)/h)

xVals = linspace(xi, xf); % get values at which to test for ksi
yVals = dds4dx4(xVals);   % approximate ds(4)(x)

% ds(4)(ksi) is the max of ds(4)(x) on [-91.20,91,20]
ds4ofksi = max(yVals);
h = (180/((xf-xi)*abs(ds4ofksi))) ^(1/4);
n = ceil((xf-xi)/h);
% for Simpson's Rule, n has to be even
if mod(n,2) == 1
    n = n+1;
end

ds = matlabFunction(ds); % turn sym function to function handle

% Use Composite Simpson's Rule to approximate Integral(ds , xi, xf)
s = CompositeSimpsonsRule(ds, xi, xf, n);
```

```
fprintf('The arc length to the nearest meter is %.0f m\n', s);
```

*Problem 3*
*The arc length to the nearest meter is 451 m*

# Composite Trapezoidal Rule

```
function approximation = CompositeTrapezoidalRule(f, x0, xn, n)
% This function approximates a definite integral using the composite
% trapezoidal rule.
%    INPUTS - f: the integrand of the integral one wishes to
 approximate
%              x0: lower bound of integration
%              xn: upper bound of integration
%              n: number of intervals between x0 and xn
%   OUTPUTS - approximation: approximation using composite trapezoidal
 rule

h = (xn - x0)/n;    % step size

approximation = 0;  % initialize approximation variable

for j = 1: n-1
    xInterior = x0+j*h;
    approximation = approximation + f(xInterior);
end

approximation = (h/2)*(f(x0) + 2*approximation + f(xn));

end
```

*(a) Composite Trapezoidal Rule with n = 8: 0.440345*

# Composite Simpsons Rule

```
function approximation = CompositeSimpsonsRule(f, x0, xn, n)
% This function approximates a definite integral using the composite
% Simpson's rule.
%    INPUTS - f: the integrand of the integral one wishes to
 approximate
%              x0: lower bound of integration
%              xn: upper bound of integration
%              n: number of intervals between x0 and xn
%   OUTPUTS - approximation: approximation using composite Simpson's
 rule

h = (xn - x0)/n;        % step size

evenApproximation = 0;  % initialize approximation for even indices
oddApproximation = 0;   % initialize approximation for odd indices

for j = 1: n-1
```

```
    if mod(j, 2) == 0
        xInterior = x0 + h*j;
        evenApproximation = evenApproximation + f(xInterior);
    else
        xInterior = x0 + h*j;
        oddApproximation = oddApproximation + f(xInterior);
    end
end

approximation = (h/3)*(f(x0) + 2*evenApproximation +
 4*oddApproximation + f(xn));

end
```

```
(b) Composite   Simpson's Rule with n = 8: 0.439199
```

*Published with MATLAB® R2018b*