RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	BSC101A
Course Name	Engineering Mathematics – 1
Programme	B.Tech
Department	Mathematics
Faculty	

Name of the Student	Satyajit Ghana
Reg. No	17ETCS002159
Semester/Year	1st - 2017
Course Leader/s	

Declaration Sheet				
Student Name				
Reg. No				
Programme		Semester/Year		
Course Code				
Course Title				
Course Date		to		
Course Leader				

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

< The table numbers have to be based on the chapter number>

< The Figure numbers have to be based on the chapter number>

Symbol	Description	Units
A	Current	Amp
g	Acceleration due to gravity - 9.81	m/s^2
V	Voltage	Volts
w	Width	mm

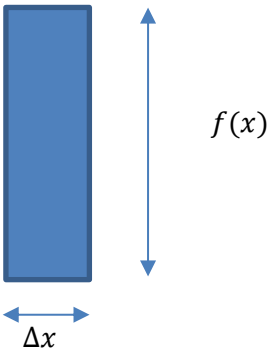
< Arrange in alphabetical order>

Question No. 1

Solution to Question No. 1 Part A:

A.1 Description of the concept of Riemann sum using the three methods: left rectangles, right rectangles and mid-point rectangles with an example:

The Riemann Sum can be interpreted as one of a method that helps to approximate the area under a given curve $f(x)$, Riemann achieves this by diving the area under the function into shapes like rectangles, trapezoids, parabolas, etc. and then finding the sum of these shapes.



$$Area\ under\ the\ curve\ = \sum_{i=1}^{n} f(x_i)\Delta x_i$$

Given a function $f(x)$, for which the area under the curve of $f(x)$ is to be calculated under the interval $a \leq x \leq b$

We define Δx such that $\Delta x = \frac{b-a}{n}$, hence the interval $[a, b]$ is essentially is divided into n parts, now as the value of n increases or $\lim_{n\to\infty} \Delta x \to 0$ the Riemann sum obtained is more close the original integral, i.e.

$$\int_a^b f(x^*)dx = \lim_{\Delta x\to 0} \sum_{i=0}^{n} f(x_i)\Delta x = \Delta x(f(x_0) + f(x_1) + ... + f(x_n))$$

The integral sign $\int$ refers to sum like notation just like the summation sign Σ ,the only difference is that integration is the sum over an infinite number of terms which is evident as our Δx approaches 0 we will have infinite number of terms for which summation is to be calculated or simply integrated.

The collections of points $\{x_0, x_1, x_2, x_3. . . . x_n\}$ is known as the partition of $[a, b]$

1. Left rectangles

$$Approximation\ of \int_a^b f(x)dx = \sum_{i=0}^{n-1} f(x_i)\Delta x = \Delta x(f(x_0) + f(x_1) + ... + f(x_{n-1}))$$

2. Right rectangles

$$Approximation\ of \int_a^b f(x)dx = \sum_{i=1}^{n} f(x_i)\Delta x = \Delta x(f(x_1) + f(x_2) + ... + f(x_n))$$

3. Mid-Point rectangles

$$Approximation\ of \int_a^b f(x)dx = \sum_{i=1}^{n} f(\overline{x_i})\Delta x = \Delta x(f(\overline{x_1}) + f(\overline{x_2}) + ... + f(\overline{x_n}))$$

Where $\overline{x_i}$ represents the midpoint of the interval i.e. $\frac{(x_i + x_{i-1})}{2}$

A.2 Description of the geometrical meaning of Riemann sums with sketch of the left rectangles, right rectangles and mid-point rectangles using an example:

Riemann Sums, in a geometric way can be interpreted as breaking up the given function $f(x)$ into the same shape, either rectangles or trapezoids, then finding the sum of these shapes, the resultant will be the area under the curve $f(x)$, an observation can be made that this is an approximation and not the exact value, but as the number of shape slices of the function increase the sum of slices approaches more and more close to the exact value of the area under the curve.

Let's consider the function $f(x) = \sqrt{x}$ in the interval $0 \leq x \leq 5$ for which we would like to compute the Riemann sum, taking $n = 5$, $\Delta x = \frac{5-0}{5} = 1$

1. Left rectangles

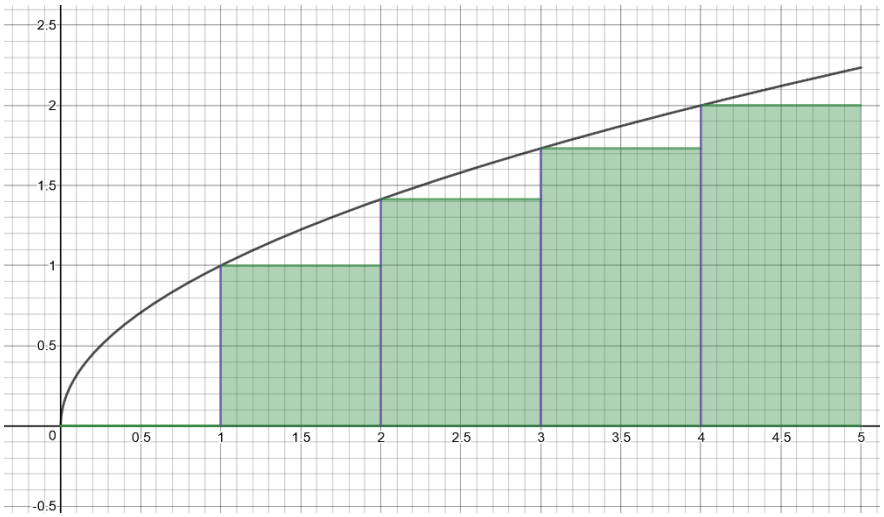


Figure A.1 left rectangles Riemann sum

$$f(0) \times 1 = \sqrt{0} = 0.000000000$$
$$f(1) \times 1 = \sqrt{1} = 1.000000000$$
$$f(2) \times 1 = \sqrt{2} = 1.414213562$$
$$f(3) \times 1 = \sqrt{3} = 1.732050808$$
$$f(4) \times 1 = \sqrt{4} = 2.000000000$$
$$\sum_{i=0}^{n-1} f(x_i)\Delta x = 6.146264370$$

2. Right rectangles

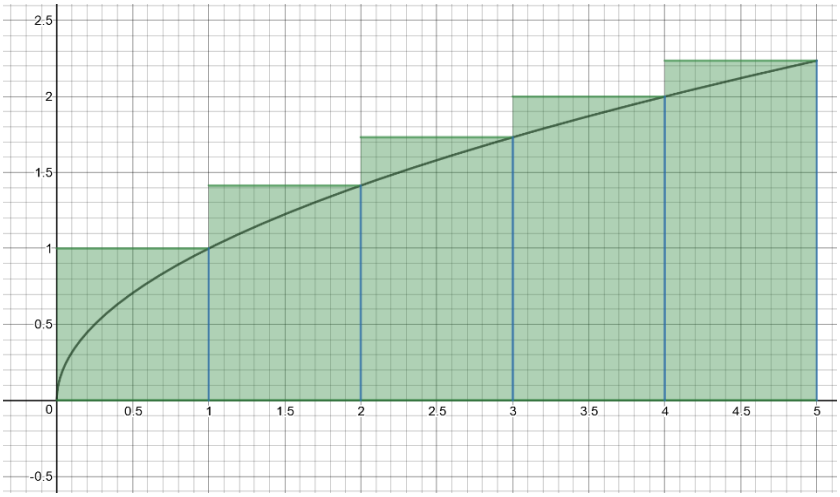


Figure A.2 right rectangles Riemann sum

$$f(1) \times 1 = \sqrt{1} = 1.000000000$$
$$f(2) \times 1 = \sqrt{2} = 1.414213562$$
$$f(3) \times 1 = \sqrt{3} = 1.732050808$$
$$f(4) \times 1 = \sqrt{4} = 2.000000000$$
$$f(5) \times 1 = \sqrt{5} = 2.236067977$$
$$\sum_{i=1}^{n} f(x_i)\Delta x = 8.382332347$$

3. Mid-Point rectangles

Figure A.3 middle-point rectangles Riemann sum

$$f(0.5) \times 1 = \sqrt{0.5} = 0.707106781$$
$$f(1.5) \times 1 = \sqrt{1.5} = 1.224744871$$
$$f(2.5) \times 1 = \sqrt{2.5} = 1.581138830$$
$$f(3.5) \times 1 = \sqrt{3.5} = 1.870828693$$
$$f(4.5) \times 1 = \sqrt{4.5} = 2.121320344$$
$$\sum_{i=1}^{n} f(\overline{x_i})\Delta x = 7.505139520$$

A.3 MATLAB function to compute the Riemann sum using left rectangles, right rectangles and mid-point rectangles.

```
function [] = riemannSums (func, a, b, n)
%Author : Satyajit Ghana
%Matlab function to find left, right and middle reiemann sum for f(x) from a
to b with n partitions
%USAGE : riemannSums(@(x) f(x), lowerBound, upperBound, numberOfPartitions)
leftSum = 0;
rightSum = 0;
middleSum = 0;
dx = (b-a)/n;
fprintf('\nGiven function : ');
disp(func);
    %left sum calculation
    %run from 0 to n-1 (lefts)
    for i=0:1:n-1
        xleft = a + i*dx;
        leftSum = leftSum + func(xleft);
    end
fprintf('Left Sum : %.9f\n', leftSum);
    %right sum calculation
    %run from 1 to n (right)
    for i=1:1:n
        xright = a + i*dx;
        rightSum = rightSum + func(xright);
    end
fprintf('\nRight Sum : %.9f\n', rightSum);
    %middle sum calculation
    %run from 1 to n (middle)
    for i=1:1:n
        xmiddle = ( a + i*dx + a+(i-1)*dx ) / 2;
        middleSum = middleSum + func(xmiddle);
    end
fprintf('\nMiddle Sum : %.9f\n', middleSum);
end
```

A.4 Comparison of the results of the above three methods and conclusion.

All the three methods approximately compute the area under the curve of $f(x)$, from the observations made in A.2 , left rectangle Riemann Sums is underestimating if f is a monotonically increasing and overestimating if f is a monotonically decreasing function.

The right rectangle Riemann Sums is overestimating for if f is monotonically increasing and underestimating if f is monotonically decreasing.

The middle rectangle Riemann Sums is more closer to the actual area under the curve.

Another observation to make is that as the number of partitions increase all the three type of sumations, namely left rectangle Riemann Sum, right rectangle Riemann Sum and middle rectangle Riemann sums converge to the actual area under the curve $f(x)$.

Solution to Question No. 1 Part B:

B1.1 Compute velocity and acceleration at $\boldsymbol{t = \frac{\pi}{2}}$ and $= \frac{3\pi}{2}$:

Given the Displacement as a function of time, the equation of velocity can be computed by finding the derivative of $s(t)$, using the fact that velocity is change is displacement over time, i.e. $v(t) = \frac{d\,s(t)}{dt}$.

Hence

$$v(t) = \frac{1}{2}\frac{d(t^2 + 5t\sin t)}{dt}$$

$$\boldsymbol{v(t) = \frac{1}{2}\,(2t + 5\sin t + 5t\cos t)}$$

$$v(t)\,|_{t=\frac{\pi}{2}} = \frac{1}{2}\,(2 \times \frac{\pi}{2} + 5 \times \sin\frac{\pi}{2} + 5 \times \frac{\pi}{2} \times \cos\frac{\pi}{2})$$

$$\boldsymbol{v(t)\,|_{t=\frac{\pi}{2}} = \frac{5 + \pi}{2}}$$

$velocity\ remains\ constant\ after\ t = \pi\ and\ is\ equal\ to\ v(\pi)$

$$v(t)\,|_{t=\frac{3\pi}{2}} = v(t)\,|_{t=\pi}$$

$$v(t)\,|_{t=\pi} = \frac{1}{2}\,(2 \times \pi + 5 \times \sin\pi + 5 \times \pi \times \cos\pi)$$

$$\boldsymbol{v(t)\,|_{t=\frac{3\pi}{2}} = -\frac{3\pi}{2}}$$

Now that the velocity is calculated as a function of time, finding the acceleration is easy, just find the derivative of the velocity, as change in velocity with respect to time is acceleration.

$$a(t) = \frac{d(v(t))}{dt}$$

$$a(t) = \frac{d\left(\frac{1}{2}(2t + 5\sin t + 5t\cos t)\right)}{dt}$$

$$\boldsymbol{a(t) = \frac{1}{2}(2 + 10\cos t - 5t\sin t)}$$

$$a(t)\,|_{t=\frac{\pi}{2}} = \frac{1}{2}(2 + 10 \times \cos\frac{\pi}{2} - 5 \times \frac{\pi}{2} \times \sin\frac{\pi}{2})$$

$$\boldsymbol{a(t)\,|_{t=\frac{\pi}{2}} = 1 - \frac{5\pi}{4}}$$

The velocity of the particle after $t = \pi$ *is constant,*

hence the acceleration being derivative of velocity, and derivative of a constant is zero, acceleration will be zero after $t = \pi$

$$\boldsymbol{a(t)\,|_{t=\pi} = 0}$$

B1.2 Plot the graph of displacement curve:

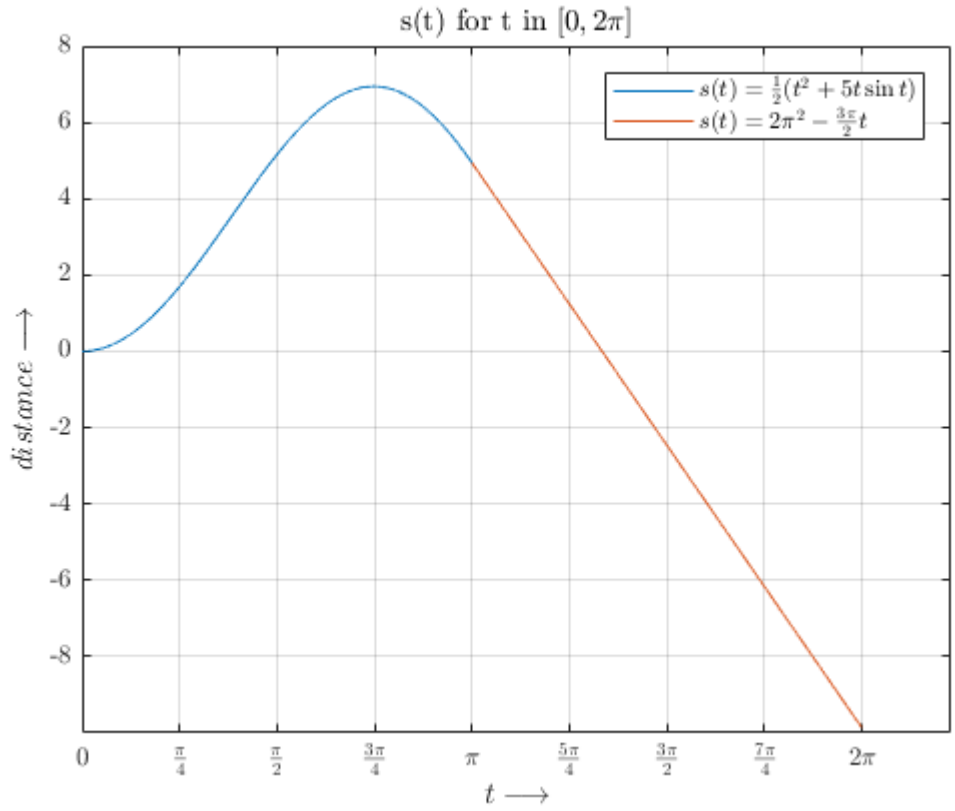


Figure B1.1 Displacement vs Time curve for the particle

MATLAB Code for the figure

```
format long;
X = linspace(0,pi,5000);
C = linspace(pi,2*pi,5000);
Y=0.5*(X.^2+5.*X.*sin(X));
W=2*pi*pi-(3/2)*pi.*C;
plot(X,Y)
hold on
plot(C,W)
xticks([0 pi/4 pi/2 3*pi/4 pi 5*pi/4 3*pi/2 7*pi/4 2*pi]);
yticks(-8:2:26);
set(gca,'TickLabelInterpreter','latex')
set(gca, 'XTick-
label',{'0','$\frac{\pi}{4}$','$\frac{\pi}{2}$','$\frac{3\pi}{4}$','$\pi$','$
\frac{5\pi}{4}$','$\frac{3\pi}{2}$','$\frac{7\pi}{4}$','$2\pi$'});
title('s(t) for t in $[0,2\pi]$','Interpreter','latex')
legend({'$s(t)=\frac{1}{2}(t^2+5t\sin{t})$','$s(t)=2\pi^2-\frac{3\pi}{2}t$'},
'interpreter', 'latex')
xlabel('$t \longrightarrow$','Interpreter','latex');
ylabel('$distance \longrightarrow$','Interpreter','latex');
grid on
```

B1.3 Show that the particle comes to rest at least once before attaining uniform velocity

Given that

$$s(t) = \frac{1}{2}(t^2 + 5t\sin t)$$

When the slope of this curve goes to zero or becomes parallel to x-axis the particle comes to rest, i.e. the particle will come to rest when the velocity becomes zero.

For the function $s(t)$ where $t \epsilon [1.51632,3.14159]$

1. $s(t)$ is continuous as it's a combination of polynomial and the ever continuous sine curve, and also this is evident from *Figure B1.1*
2. $s'(t) = \frac{1}{2}(2t + 5sint + 5tcost)$, as $s'(t)$ is continuous in the interval $t \epsilon (1.51632,3.15159)$, $s(t)$ is derivable in the range $(1.51632,3.14159)$
3. $s(1.51632) = 4.9348$; $s(3.14159) = 4.9348$; $s(b) = s(a)$

As $s(t)$ for $t \epsilon [1.51632, \pi]$ satisfies all the three conditions for Rolle's Theorem we can conclude that there will be at least one $c \epsilon (1.51632, \pi)$ for which $s'(c) = 0$.

The velocity function being:

$$v(t) = \frac{1}{2}(2t + 5\sin t + 5t\cos t)$$

The particle will come to rest when the velocity becomes zero.

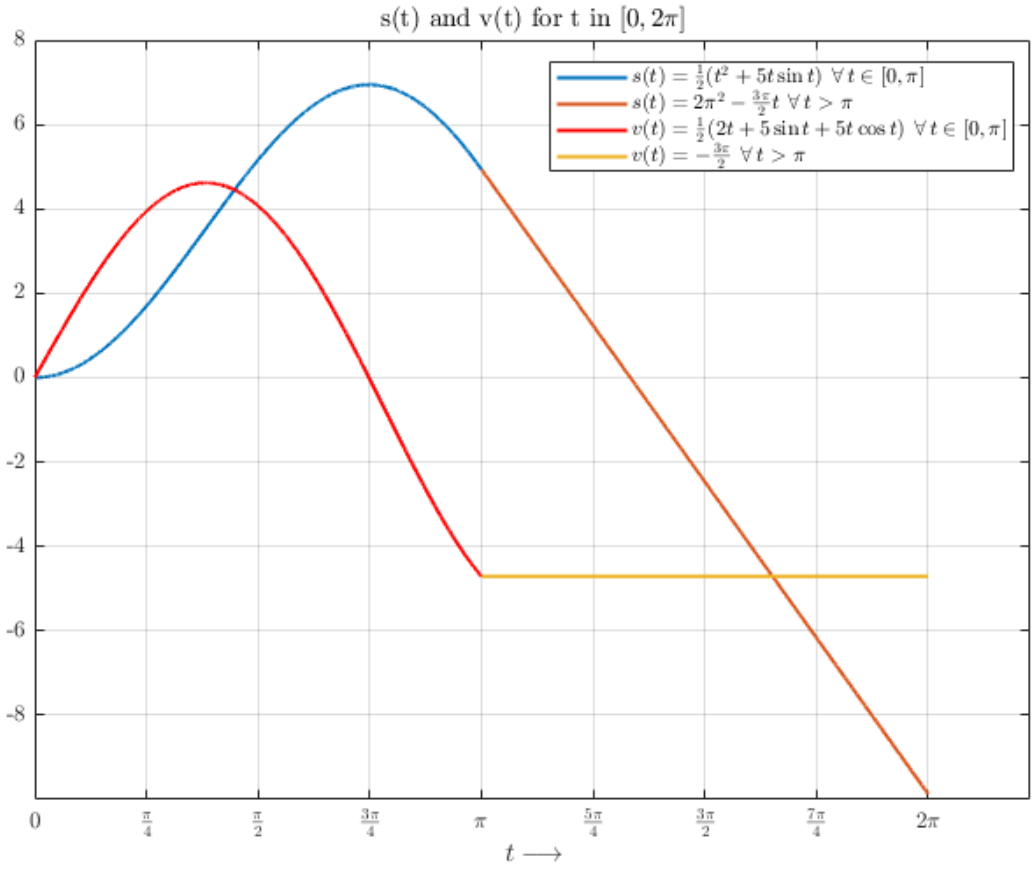


Figure B1.2 Displacement and Velocity vs Time Curve for the Particle0

The particle comes to rest when $s'(t) = v(t) = 0$ which is true when $t = \pm 2.35003646166806$ as $\epsilon\ (0, \pi)$, hence for $t = 2.35003646166806\ second$ the particle comes to rest.

B1.4 Conclude and comment on the results

Velocity is a derivative of displacement w.r.t. time and acceleration is derivative of velocity w.r.t time, Hence when the velocity becomes constant at $= \pi$, acceleration becomes zero.

Solution to Question No. 2 Part B:

B2.1 Obtain the Maclaurin series for the kinetic energy as a function of the velocity:

Maclaurin series for a function $f(x)$ is:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)x^2}{2!} + \frac{f'''(0)x^3}{3!} + \ldots + \frac{f^{(n)}(0)x^n}{n!} = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!}$$

$$given\ m = m_0\left(1 - \frac{v^2}{c^2}\right)^{-\frac{1}{2}}\ and\ K = mc^2 - m_0c^2$$

$Where\ m\ is\ the\ relativistic\ mass\ of\ the\ object\ and\ m_0\ is\ the\ rest\ mass\ of\ the\ object.$
$v\ is\ the\ velocity\ of\ the\ object\ relative\ to\ the\ observer, and\ c\ is\ speed\ of\ light.$

$$and\ hence\ we\ have\ K = m_0c^2\left(1 - \frac{v^2}{c^2}\right)^{-\frac{1}{2}} - m_0c^2$$

$$K = m_0c^2\left[\left(1 - \frac{v^2}{c^2}\right)^{-\frac{1}{2}} - 1\right]$$

$$Let\ f(x) = (1-x)^{-\frac{1}{2}} - 1$$

$Now\ we\ can\ compute\ the\ Maclaurin\ Series\ for\ f$

$$f(0) = (1-x)^{-\frac{1}{2}} - 1 = 1 - 1 = 0$$

$$f'(0) = \frac{1}{2}(1-x)^{-\frac{3}{2}} = \frac{1}{2}$$

$$f''(0) = \frac{1}{2}\cdot\frac{3}{2}(1-x)^{-\frac{5}{2}} = \frac{3}{4}$$

$$f'''(0) = \frac{1}{2}\cdot\frac{3}{2}\cdot\frac{5}{2}(1-x)^{-\frac{7}{2}} = \frac{15}{8}$$

$$f^n(0) = \frac{1\cdot 3\cdot 5\cdot 7\cdot\ldots\cdot(2n-1)}{2^n}(1-x)^{-\frac{2n+1}{2}}$$

$Hence\ the\ Maclaurin\ Series\ for\ f\ is$

$$f(x) = \frac{1}{2}x + \frac{3}{8}x^2 + \frac{5}{16}x^3 + \frac{35}{128}x^4 + \frac{63}{256}x^5 + \frac{231}{1024}x^6 + O(x^7)$$

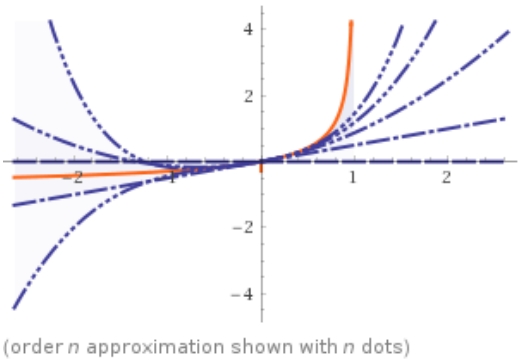


Figure B2.1 n order approximation for f(x)

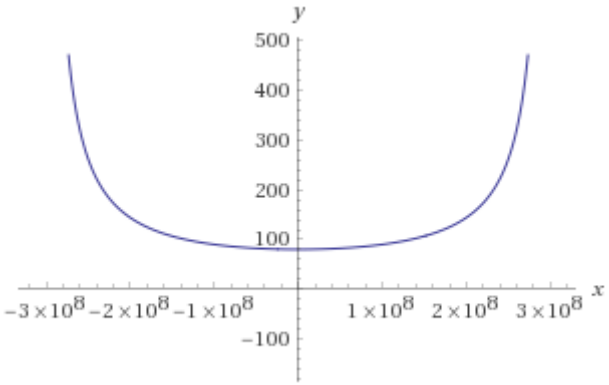


Figure B2.2 graphical plot for f(x)

Now we can use this to analyze Kinetic Energy

$$K = mc^2 - m_0c^2 = m_0c^2 \times f\left(\frac{v^2}{c^2}\right)$$

$$K = m_0c^2\left[\frac{1}{2}\frac{v^2}{c^2} + \frac{3}{8}\left(\frac{v^2}{c^2}\right)^2 + \frac{5}{16}\left(\frac{v^2}{c^2}\right)^3 + \ldots\right]$$

$$K = \frac{1}{2}m_0v^2 + m_0\left[\frac{3}{8}\left(\frac{v^2}{c^2}\right)^2 + \frac{5}{16}\left(\frac{v^2}{c^2}\right)^3 + \ldots\right]$$

B2.2 Show that when v is very small compared to c, the expression for K agrees with the classical Newtonian physics by $K = \frac{1}{2}m_0v^2$:

Previously we formulated our Relativistic Kinetic Energy to be

$$K = \frac{1}{2}m_0v^2 + m_0\left[\frac{3}{8}\left(\frac{v^2}{c^2}\right)^2 + \frac{5}{16}\left(\frac{v^2}{c^2}\right)^3 + \ldots\right]$$

In this if we assume that there is no particle that can go faster than light, and that the velocities that we deal with in Newtonian Physics have fairly low velocity compared to speed of light, which is said to be the ultimate speed that a particle could ever reach, then $< c$, which implies that $\frac{v^2}{c^2} < 1$ which is not too small to ignore though $\left(\frac{v^2}{c^2}\right)^n \ll 1$ and its exponents tend to be a very small quantity and diminish hence can be ignored.

Hence now our Kinetic Energy which is true according to Newtonian Physics is

$$K = \frac{1}{2}m_0v^2$$

B2.3 Comment on the results and conclude.

Classical Newtonian Kinetic Energy can be obtained from the Relativistic Kinetic energy when v is considered to be very small compared to c .

Solution to Question No. 3 part B:

B3.1 Choose any algorithm to search the index of a given element in vector X and explain:

To Search from an unsorted array there is only one way which would be linear as there is no order in it. The time complexity here is $O(n)$ which is linearly increasing. Hence the algorithm being:

1. Start
2. Take the Vector X and Element to search as input
3. Take an element from Vector X and compare with Element to search
4. If element is found then store the index in Vector indexes
5. Repeat Step 3 and 4 until the last element of Vector X
6. Display Vector Indexes
7. Stop

B3.2 Write a MATLAB function to implement the above chosen algorithm. The function should accept the array and an element as input and should output the index of that element in the given array:

```
function [  ] = searchArray( arr , element )

%Author : Satyajit Ghana
%searchArray to be used to search for an element in an array
%displays all the indexes for the element, if not found
%prints element not found
%   USAGE : searchArray(array , element)
%lengthOfArray = length(arr);

index = [];
found = false;
    for i=1:1:length(arr)
        if arr(i) == element
            index = [index, i];
            found = true;
        end
    end
    if found == false
        fprintf('Element not found\n');
    else
        fprintf('Element found at index : ');
        for i=1:1:length(index)
            fprintf('%d ,',index(i));
        end
        fprintf('\b\b\n');
    end
end
```

B3.3 Use MATLAB built-in function 'find' to search the index of the required element. Compare and comment on the results:

The built-in $find()$ function has more scalability compared to the user defined function in *B3.2* , $I = find(X, K)$ returns at most first K indices for which the X is satisfied, here X can also be a relational statement.

Comparing the runtimes of the two, along with another algorithm binary search which sorts the given vector using $sort()$ function and then binary searched for the element. A 50,000,000 element vector was taken with random integers, and a random element was searched in this vector using the in-built $find$ function and the user-defined search function, the run time being :

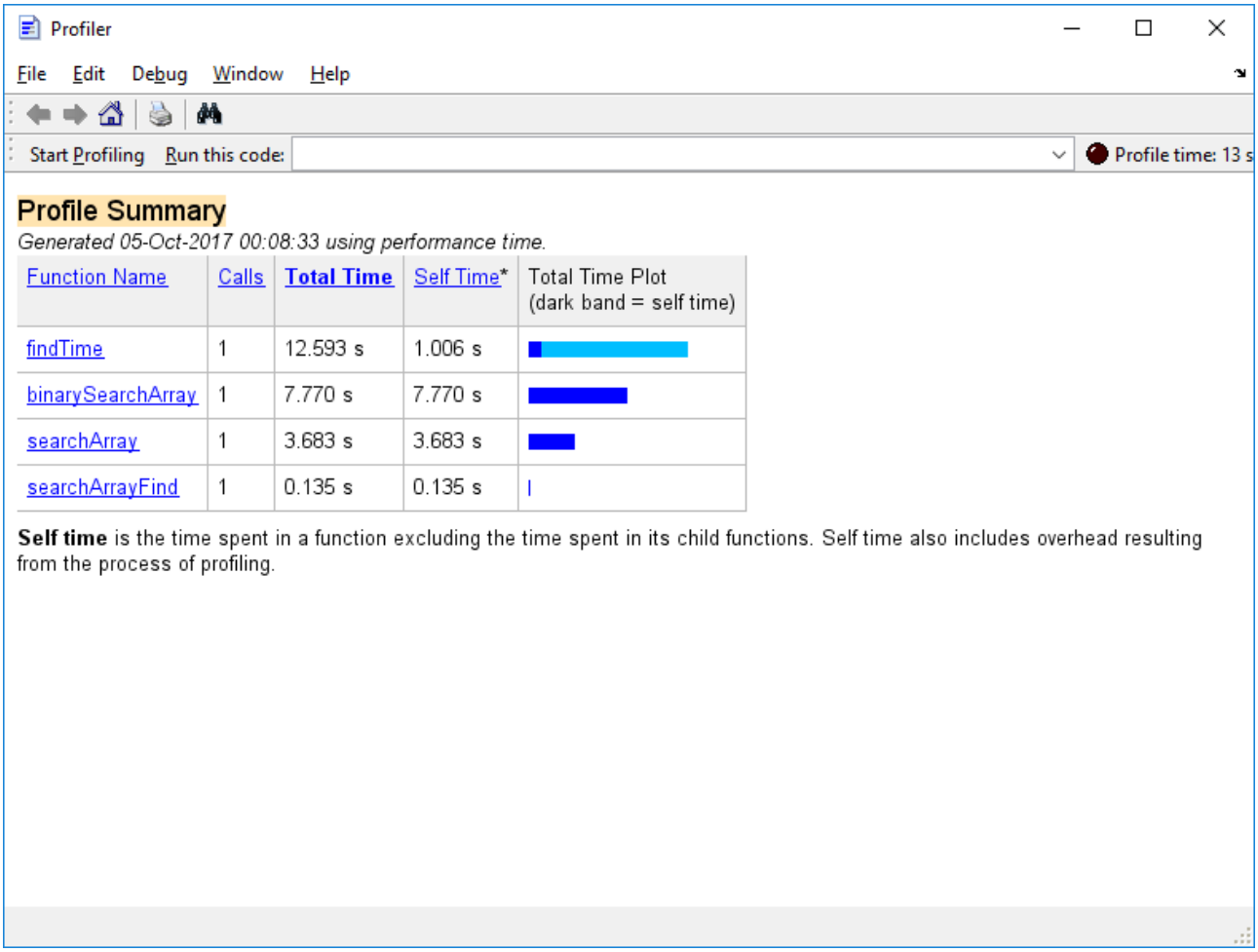


Figure B3.1 Run Time comparison for build-in find function and user defined search function

There is a large difference between the run time of *searchArray* and *searchArrayFind*, $find()$ function is way more quicker in searching for the element and provides more options to search for.

Solution to Question No. 4 Part B:

B4.1 Obtain Equations for the temperature at the four intersection points:

$$Temperature\ at\ x_1 = \frac{200 + 200 + x_3 + x_2}{4}$$
$$Temperature\ at\ x_2 = \frac{200 + 100 + x_4 + x_1}{4}$$
$$Temperature\ at\ x_3 = \frac{200 + x_1 + x_4 + 100}{4}$$
$$Temperature\ at\ x_4 = \frac{x_2 + 100 + 100 + x_3}{4}$$

The equations hence formed are :

$$4x_1 - x_2 - x_3 = 400$$
$$-x_1 + 4x_2 - x_4 = 300$$
$$-x_1 + 4x_3 - x_4 = 300$$
$$-x_2 - x_3 + 4x_4 = 200$$

This can be written in an Matrix form as :

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 400 \\ 300 \\ 300 \\ 200 \end{pmatrix}$$

B4.2 Solve the resultant system to find the temperature at each intersection point in the grid:
Students are expected to provide the solution to the question considering the points mentioned in the marking scheme of the assignment question

$The\ Augumented\ matrix\ is$:

$$\begin{pmatrix} 4 & -1 & -1 & 0 & 400 \\ -1 & 4 & 0 & -1 & 300 \\ -1 & 0 & 4 & -1 & 300 \\ 0 & -1 & -1 & 4 & 200 \end{pmatrix}$$

$R_2 \rightarrow 4R_2 + R_1$

$R_3 \rightarrow 4R_3 + R_1$

$$\begin{pmatrix} 4 & -1 & -1 & 0 & 400 \\ 0 & 15 & -1 & -4 & 1600 \\ 0 & -1 & 15 & -4 & 1600 \\ 0 & -1 & -1 & 4 & 200 \end{pmatrix}$$

$R_3 \to 15R_3 + R_2$

$R_4 \to 15R_4 + R_2$

$$\begin{pmatrix} 4 & -1 & -1 & 0 & 400 \\ 0 & 15 & -1 & -4 & 1600 \\ 0 & 0 & 224 & -64 & 25600 \\ 0 & 0 & -16 & 56 & 4600 \end{pmatrix}$$

$R_3 \to \frac{R_3}{32}; R_4 \to \frac{R_4}{8}$

$$\begin{pmatrix} 4 & -1 & -1 & 0 & 400 \\ 0 & 15 & -1 & -4 & 1600 \\ 0 & 0 & 7 & -2 & 800 \\ 0 & 0 & -2 & 7 & 575 \end{pmatrix}$$

$R_4 \to 7R_4 + 2R_3$

$$\begin{pmatrix} 4 & -1 & -1 & 0 & 400 \\ 0 & 15 & -1 & -4 & 1600 \\ 0 & 0 & 7 & -2 & 800 \\ 0 & 0 & 0 & 45 & 5625 \end{pmatrix}$$

$rank\ of\ augumented\ matrix = rank\ of\ matrix, hence\ we\ have\ an\ unique\ solution$

$Now\ we\ have$

$$45x_4 = 5625$$

$$7x_3 - 2x_4 = 800$$

$$15x_2 - x_3 - 4x_4 = 1600$$

$$4x_1 - x_2 - x_3 = 400$$

$The\ Solution\ being:$

$x_1 = 175$

$x_2 = 150$

$x_3 = 150$

$x_4 = 125$

B4.3 Write a MATLAB script to check for the consistency of the system and obtain the solution.

```
function [] = linearEquationSolver ( array, constants )
%Author : Satyajit Ghana
%linearEquationSolver can be used to find a solution
%to a linear equation, if solution exists it prints the
%solution, else it prints solution doesn't exist
%
%   USAGE : linearEquationSolver ( array, constants)
sizeOfMatrix = size(array);
numberOfVariables = sizeOfMatrix(2);
augumentedMatrix = [array,constants];

if rank(array) == rank(augumentedMatrix)
    fprintf('The System is Consistent ');
    if rank(array) == numberOfVariables
        fprintf('and has a Unique Solution\n');
        solution = array^(-1)*constants;
        for i=1:1:length(solution)
            fprintf('X%d = %f\n',i,solution(i));
        end
    else
        freeVariables = numberOfVariables - rank(array);
        fprintf('and has Infinitely Many Solutions with %d free
variables', freeVariables);
    end
else
    fprintf('The System is Inconsistent');
end

end
```

```
>> array = [4 -1 -1 0; -1 4 0 -1; -1 0 4 -1; 0 -1 -1 4]

array =

     4    -1    -1     0
    -1     4     0    -1
    -1     0     4    -1
     0    -1    -1     4

>> constants = [400; 300; 300; 200]

constants =

   400
   300
   300
   200

>> linearEquationSolver(array, constants)
The System is Consistent and has a Unique Solution
X1 = 175.000000
X2 = 150.000000
X3 = 150.000000
X4 = 125.000000
```

B4.4 Comment on the results obtained and conclude

The given problem turns out to be a 4 variable equation with x_1, x_2, x_3, x_4 whose values are to be computed, to find the value of these 4 equations are setup which is then solved to get the values of respective variables. The system is consistent and has a Unique Solution which was verified with the results obtained using the formed MATLAB function.

1. James Stewart, 2015, *Calculus – Early Transcendentals*, Cengage Learning, pp 379-386