Laxmi Charitable Trust's

Sheth L.U.J. & Sir M.V. College of Arts, Science & Commerce

Dr. S Radhakrishnan Marg, Andheri (E), Mumbai – 400 069.

Certificate

This is to certify that, Mr./Ms. NEERAJ APPARI *Seat No.* F129 *studying in* **F.Y.B.Sc. SEM-II** *Computer Science has satisfactorily completed the Practicals in the Subject of* CALCULAS *as prescribed by University of Mumbai, during academic year* **2019-2020**.

Signature
Subject in charge
Date: - 03/02/2020

Signature
Co-ordinator B.Sc. C.S
Date: -

Signature
External Examiner
Date: -

Scanned with CamScanner

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

	Date	Practical	Sign
1	2/1/20	Find the derivative of the following	
2	2/1/20	Find the partial derivative of the following.	
3	20/2/20	Simpson's 1/3rd rule	
4	20/2/20	Euler's Method	
5	1/3/20	Newton's Raphson Method	
6	1/3/20	Absolute Maxima	
7	1/3/20	Absolute Minima	
8	1/3/20	Gradient function	

Scanned with CamScanner

Practical: 01

Aim: find the derivatives

1) $x^2 - 3x$
>f1 <- expression (x^2-3*x)
>D(f1,'x')

2) $x^2 + 2x + 3$
>f1 <- expression (x^2+2*x+3)
>D(f1,'x')

3) $\cos(x^2)y$
>f1 <- expression (cos(x^2)*y)
>D(f1,'x')

4) $x^3 + y^3$
>f1 <- expression(x^3+y^3)
>D(f1,'x')

5) $x\sin(x^2)y$
>f1 <- expression (sin(x^2)*y)
>D(f1,'y')

ALNRAO

1] 2 * x - 3

2] 2 * x + 2

3] - (sin (x ^ 2) * (2 * x) * y)

4) 3 + x ^ 2

5) cos (x ^ 2) * 2 * x) * y

Practical 02

Aim - find the partial derivatives

1) cos (x2) . y
>f <- expression(cos(x^2)*y)
>D(f, 'x')
-(sin(x^2)*(2*x)*y)
>D(f, 'y')
- cos (x^2)

2) $x^2 + y^3$
>f <- expression(x^2 + y^3)
>D(f, 'x')
- 2 * x
> D(f, 'y')
- 3 * y ^2

3) $x^2 - 3x$
>f <- expression(x^3 - 3*x)
>D(f, 'x')
>D(f, 'y'

4) sin (x2) . y
>f <- expression (sin(x^2)*y
>D(f, 'x')
>D(f, 'y'

ALNDAO

1) (sin(x^2)*(2*x))*y
cos(x^2)

2) 2*x
3*y^2

3) 2*x-3
0

4) cos(x^2)*(2*x)*y
sin(x^2)

5) x^2+2x+3
f <- expression(x^2+2*x+3)
> D(f,'x')
> D(f,'y')

6) f ~~sin (x^2~~ $2x^3.y^2+2.y+4x$
f <- expression(2*x^3*y^2+2*y+4*x)
D(f,'x')
D(f,'y')

7) $(x^4.\sin(xy^3)$
f <- expression(x^4.*sin*(x*y^3)
D(f,'x')
D(f,'y')

8) ~~$(x^3*y^2*z^4+2x$~~ $(x^3.y^2.z^4+2xy+2)$
f <- expression(x^3*y^2*z^4+2*x*y+2)
D(f,'x')
D(f,'y')

9) $(2xy^3+4x^3y)$
f <- expression(2*x*y^3+4*x^3*y)
> D(f,'x')
> D(f,'y')

10) (x^3+y^3+x^4+y)
> D(f,'x')
> D(f,'y')

5) 2*x+2
0

6) 2*x^3*(2*y)+2
2*(3*x^2)*y^2+4

7) 4*x^3+sin(x*y^3)+x^4*(cos(x*y^3)*y^3)
x^4*cos

8) 3*x^2*y^2*z^4+2*y
x^3*(2*y)*z^4+2*z
x^3*y^2*(4*z^3)+x*y+1

9) 2*y^3+4*(3*x^2)*y
2*x*(3*y^2)+4*x*y

10) 3*x^2*y^3+4*z^3*y
x^3*(3*y^2)+z^4

Scanned with CamScanner

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

A(

11) fL- $y^3 \cdot e^x + y$
fL- expression (y^3*e^x+y)
D(f, 'x')
D(f, 'y')

ALNERAO

ii) y^3*(e^x*log(e))
3*y^2*e^x+1

Scanned with CamScanner

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

Practical - 3

Aim - Write a Simpsons 1/3rd rule

1)
```
v2 <-c(0.05,1,1.5,2)
v1 <-c(0.3989,0.3521,0.2420,0.1296,0.0540)
>h <-0.5
>a <-v1[c(1)]
>b <-v1[c(2)]*4
>e <-v1[c(3)]*2
>f <-v1[c(5)]
>g <-h*(a+b+d+f)
>i <-g/3
print(i)

[1] 0.4772833
```

2)
```
v2 <-c(4,5,5.7340,6,5.73330,5.4)
v1 <-c(0,30,60,90,120,150,180)
>h <-0.7
>a <-v1[c(1)]
>b <-v1[c(2)]*4
>d <-v1[c(3)]*2
>e <-v1[c(4)]*4
>f <-v1[c(5)]*2
>k <-v1[c(6)]*4
>j <-v1[c(7)]
>g <-h*(a+b+d+e+f+k+j)
```

Scanned with CamScanner

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

```
>i <- g/3
>print(i)

[1] 378

v2 <- c(0,0.0005,0.04,0.187,0.33)
>v1 <- c(0,0.25,0.5,0.75,1)
>h <- 0.5
>a <- v1[c(1)]
>b <- v1[c(2)]*4
>d <- v1[c(3)]*2
>e <- v1[c(4)]*4
>f <- v1[c(5)]
>g <- h* (a+b+d+e+f)
>j <- g/3
>print(i)

[1] 2.66667
```

late

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

ALNRAO

Practical No. 4

Aim: Write a Code of Eulers method

```
euler <- function (dy.dx = function(x,y) {}, h = 1
+ nsteps <- (end - start)/h
+ ys <- numeric(nsteps + 1)
+ ys [1] <- y0
+ for (i in 1 : nsteps {
+ x <- start + (i - 1)*h
+ ys [i+1] <- ys [i] + h* dy.dx (x, ys [i])
+ }
+ ys
+ }
> dy.dx <- function (x,y) {3*x - y + 8}
> euler(dy.dx , start = 0, end = 0.5, h = 0.1, y0 = 3)
[1] 3.0000  3.50000  3.98000  4.44200  4.88780  5.31902
```

10/10

ALNRAO

P

2)
```
euler <- function( dy.dx= function(x,y) {}, h=1e-7, y0=1,
   start=0 , end=1){
+nsteps <-(end-start))/h
+ys <- numeric(nsteps+1)
+ys [1] <- y0
+for (i in 1: nsteps){
+x <-start + (i-1)*h
+ys[i+1]<-ys [i]+h*dy.dx(x,ys[i]
+}
+ys
+}
>dy.dx <- function (x,y){ x+3^y}
>euler(dy.dx, start=0 , end=1.5 , h=0.2, y0=1)

[1] 1.000000e+00    1.600000e+00    2.799909e+00
    7.214281e+00    5.608358e+02    7.70905e+266
```

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

```
> euler <- function (dy_dx = function(x,y) {} h=1e-7, y0 = 1,
  start=0, end=1) {
+ nsteps <- (end - start)/h
+ ys <- numeric (nsteps+1)
+ ys [1] <- y0
+ for (i in 1 : nsteps) {
+ x <- start + (i-1)*h
+ ys [i+1] <- ys[i] + h* dy_dx (x,ys[i])
+ }
+ ys
+ }
> dy_dx <- function (x,y) {x + y + x*y}
> euler (dy_dx, start=0, end= 0.5, h=0.25, y0=1)
```

[1] 1.000000   1.200000   1.669104   1.94097
    2.523369   3.301934

Practical No - 05

Aim - Newton Raphson method

```
> NRM <- function(x)
+ {
+ fx <- (x^2 - 2*x - 2)
+ fx0 <- (2*x - 2)
+ x1 <- x - (fx/fx0)
+ it <- 1
+ while (abs(fx) > 0.0001 & it < 20)
+ {
+ x <- x1
+ fx <- (x^2 - 2*x - 2)
+ fx0 <- (2*x - 2)
+ x1 <- x - (fx/fx0)
+ it <- it + 1
+ }
+ list (a = x1, iteration = it)
+ }
> NRM (2)
$ 'a'
[1] 2.732051

$iteration
[1] 4
```

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

```
> NRM <- function(x)
+ {
+ fx <- (x^3 - 3.5*x - 1)
+ fx0 <- (3*x^2 - 5)
+ it <- 1
+ while (abs(fx) > 0.0001 & it < 20)
+ {
+ x <- x1
+ fx <- (x^3 - 8x - 1)
+ fx0 <- (3*x^2 - 8)
+ x1 <- x - (fx/fx0)
+ it <- it + 1
+ }
+ list(a = x1, iteration = it)
+ }
> NRM(3)
$ 'a'
[1] ~~2.953679~~ 1.324718

$ iteration
[1] 36
```

ALNdAO

```
3) >NRM<-function(x)
+{
+fx<-(x^3-x-1)
+fxa<-(3*x^2-1)
+xf<-x-(fx/fxa)
+it<-1
+while(abs(fx)>0.0001 & it<20)
+{
+x<-xf
+fx<-(x^3-5*x-11)
+fxa<-(3*x^2-5)
+xf<-x-(fx/fxa)
+it<-it+1
+}
+list(a=xf,iteration=it)
+}
>NRM(3)
$'a'
[1] 2.953672
$iteration
[1] 3
```

Practical No - 6

Aim - Absolute Maxima

```
absmax <- function(x){x[which ~~abs~~.max(abs(x))]}
absmax(c(-10,0,9)
[1] -10

absmax <- function(x) {x [which.max(abs(x))]}
absmax(c(~~-10,0,9~~) (+10,1,0)
[1] 10

absmax <- function (x) {x [which.max (abs (x))]}
absmax(c(-5,-7,8,10,0))
[1] 10

absmax <- function (x) {x [which.max (abs (x))]}
~~absmax (c(-5, -7, 8, 10,0)~~
absmax (c(10,-10,0,-25))
[1] -25

absmax <- function (x){x [which.max(abs (x))]}
absmax(c(2,3,-4,-6,10))
[1] 10
```

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

ALNRAO

Practical No. 7

Aim. Absolute Minima

1) >absmin<-function (x) {x[which.min(abs(x))]}
>absmin(c(-10,0,9))
[1] 0

2) >absmin <-function (x) {x[which.min (abs(x))]}
>absmin (c(10,1,0))
[1] 0

3) >absmin<-function (x) {x[which.min(abs (x))]}
>absmin (c(-5,-7,8,10,0))
[1] 0

4) >absmin<-function (x) {x[which.min(abs(x))]}
>absmin (c(10,-10,0,-25))
[1] 0

5) >absmin<-function (x) {x[which.min(abs(x))]}
>absmin(c(2,3,-4,-6,10))
[1] 2

Scanned with CamScanner

SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE
Department of Computer Science

ALNRAO

Practical 8

Aim : Find gradient of functions

Iutits :

1) > library ("numDeriv")
> myfunc <- function (lst) {lst [1]^2 * lst [2]}
> grad (myfunc, c (3,2))
[1] 12 9

2) > myfunct <- function ~~(ls~~ (lst) {lst [1]^2 * lst [2]}
grad (myfunct, c(-2,0))

Output
[1] 0,4

Scanned with CamScanner