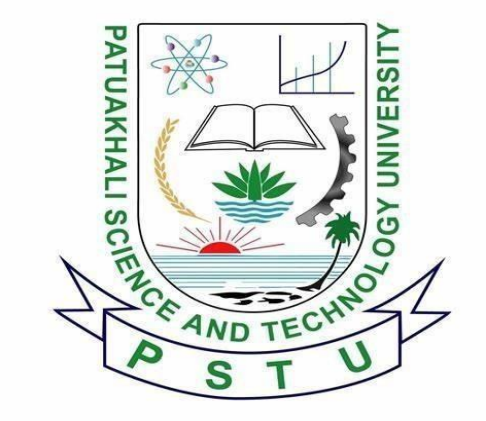
PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY



Course Code: CCE 312

SUBMITTED TO:

Dr. Md Samsuzzaman Sobuz

Department of Computer And Communication Engineering
Faculty of Computer Science And Engineering

SUBMITTED BY:

Name: MD Noushad Bhuiyan

ID: 2102038, Registration No: 10165

Cramer's Rule

Code:

```
def cramers_rule(A, b):
    import numpy as np
    detA = np.linalg.det(A)
    if detA == 0:
        return None
    n = len(b)
    x = []
    for i in range(n):
        Ai = A.copy()
        Ai[:, i] = b
        x.append(np.linalg.det(Ai)/detA)
    return x
```

Input/Output Example:

Input:
A = [[2,1],[1,3]]
b = [8,13]
Output:
[3.0, 4.0]

Iteration Method

Code:

```
def iteration_method(g, x0, tol=1e-6, max_iter=100):
    for _ in range(max_iter):
        x1 = g(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0
```

Input/Output Example:

Input:
g(x)=cos(x), x0=0.5
Output:
0.739085

Bisection Method

Code:

```
def bisection(f, a, b, tol=1e-6):
  if f(a)*f(b) > 0:
    return None
  while (b - a)/2 > tol:
    m = (a + b)/2
    if f(a)*f(m) <= 0:
      b = m
    else:
      a = m
  return (a+b)/2
```

Input/Output Example:

Input:
f(x)=x^3−x−1, a=1, b=2
Output:
1.3247

False Position Method

Code:

```
def false_position(f, a, b, tol=1e-6):
  if f(a)*f(b) > 0:
    return None
  while True:
    m = b - (f(b)*(b-a))/(f(b)-f(a))
    if abs(f(m)) < tol:
      return m
    if f(a)*f(m) < 0:
      b = m
    else:
      a = m
```

Input/Output Example:

Input:
f(x)=x^2−4, a=1, b=3
Output:
2.0

Newton Raphson Method

Code:

```
def newton_raphson(f, df, x0, tol=1e-6, max_iter=100):
    for _ in range(max_iter):
        x1 = x0 - f(x0)/df(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0
```

Input/Output Example:

Input:
f(x)=x^2−2, df(x)=2x, x0=1
Output:
1.4142

Gauss Elimination

Code:

```
def gauss_elimination(A, b):
    import numpy as np
    n = len(b)
    M = np.hstack((A, b.reshape(-1,1)))
    for i in range(n):
        M[i] = M[i] / M[i, i]
        for j in range(i+1, n):
            M[j] = M[j] - M[j, i]*M[i]
    x = np.zeros(n)
    for i in range(n-1, -1, -1):
        x[i] = M[i, -1] - sum(M[i, i+1:n]*x[i+1:n])
    return x
```

Input/Output Example:

Input:
A=[[2,1],[1,3]], b=[8,13]
Output:
[3,4]

Gauss Jordan Method

Code:

```
def gauss_jordan(A, b):
  import numpy as np
  n = len(b)
  M = np.hstack((A, b.reshape(-1,1)))
  for i in range(n):
    M[i] = M[i]/M[i,i]
    for j in range(n):
      if i != j:
        M[j] = M[j] - M[j,i]*M[i]
  return M[:, -1]
```

Input/Output Example:

Input:
A=[[1,2],[3,4]], b=[5,6]
Output:
[-4, 4.5]

Euler Method

Code:

```
def euler(f, x0, y0, h, n):
  xs, ys = [x0], [y0]
  for _ in range(n):
    y0 = y0 + h*f(x0, y0)
    x0 = x0 + h
    xs.append(x0)
    ys.append(y0)
  return xs, ys
```

Input/Output Example:

Input:
dy/dx = x+y, x0=0, y0=1, h=0.1, n=3
Output:
x=[0,0.1,0.2,0.3]
y=[1,1.1,1.22,1.362]

Runge Kutta (RK4)

Code:

```
def rk4(f, x0, y0, h, n):
    xs, ys = [x0], [y0]
    for _ in range(n):
        k1 = h*f(x0, y0)
        k2 = h*f(x0+h/2, y0+k1/2)
        k3 = h*f(x0+h/2, y0+k2/2)
        k4 = h*f(x0+h, y0+k3)
        y0 = y0 + (k1+2*k2+2*k3+k4)/6
        x0 = x0 + h
        xs.append(x0)
        ys.append(y0)
    return xs, ys
```

Input/Output Example:

Input:
dy/dx = x+y, x0=0, y0=1, h=0.1, n=1
Output:
1.110341

Trapezoidal Rule

Code:

```
def trapezoidal(f, a, b, n):
    h = (b-a)/n
    s = f(a) + f(b)
    for i in range(1, n):
        s += 2*f(a + i*h)
    return (h/2)*s
```

Input/Output Example:

Input:
f(x)=x^2, a=0, b=2, n=4
Output:
2.6667

Simpson 1/3 Rule

Code:

```
def simpson_13(f, a, b, n):
    h = (b-a)/n
    s = f(a)+f(b)
    for i in range(1, n):
        s += 4*f(a+i*h) if i%2!=0 else 2*f(a+i*h)
    return s*h/3
```

Input/Output Example:

Input:
f(x)=x^2, a=0, b=2, n=2
Output:
2.6667

Simpson 3/8 Rule

Code:

```
def simpson_38(f, a, b, n):
    h = (b-a)/n
    s = f(a)+f(b)
    for i in range(1, n):
        s += 3*f(a+i*h) if i%3!=0 else 2*f(a+i*h)
    return 3*h*s/8
```

Input/Output Example:

Input:
f(x)=x^2, a=0, b=3, n=3
Output:
9.0

Least Squares Regression

Code:

```
def least_squares(x, y):
    import numpy as np
    x, y = np.array(x), np.array(y)
    n = len(x)
    a1 = (n*(x*y).sum() - x.sum()*y.sum())/(n*(x**2).sum() - (x.sum())**2)
    a0 = (y.sum() - a1*x.sum())/n
    return a0, a1
```

Input/Output Example:

Input:
x=[1,2,3], y=[2,3,5]
Output:
a0=0.333, a1=1.5

Lagrange Interpolation

Code:

```
def lagrange_interpolation(x, y, xp):
  total = 0
  n = len(x)
  for i in range(n):
    term = y[i]
    for j in range(n):
      if i != j:
        term *= (xp - x[j])/(x[i]-x[j])
    total += term
  return total
```

Input/Output Example:

Input:
x=[1,2,3], y=[1,4,9], xp=2.5
Output:
6.25