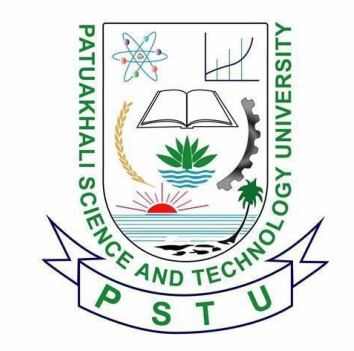

PATUAKHALI SCIENCE AND TECHNOLOGY

COURSE CODE CCE
312 Numerical Methods

SUBMITTED TO:

Prof. Dr. Md
Department of Computer and Communication
Faculty of Computer Science and

SUBMITTED BY:

Md. Sharafat
Karim ID: **2102024**,
Registration No: **10151**
Faculty of Computer Science and

Assignment 05
Assignment title: Cramer's rule
Date of submission: 16 Sat, 2025

Cramers' rule

Another one easiest way is to use cramer's rule. It's not so dynamic but it works for small systems of equations. The idea is to express the solution in terms of determinants.

```
import numpy as np

main_array = np.array(arr)
D = np.array(main_array[:,:-1])

last_col  = np.array(main_array[:,-1])

D1 = np.array([last_col, main_array[:,1], main_array[:,2]])
D2 = np.array([main_array[:,0], last_col, main_array[:,2]])
D3 = np.array([main_array[:,0], main_array[:,1], last_col])

D_det = np.linalg.det(D)
D1_det = np.linalg.det(D1)
D2_det = np.linalg.det(D2)
D3_det = np.linalg.det(D3)

print("x = ", D1_det / D_det)
print("y = ", D2_det / D_det)
print("z = ", D3_det / D_det)
```

```
x =  2.0
y =  2.9999999999999996
z =  -0.9999999999999998
```

Numerical Methods
Mastering Matplotlib!

Numerical Methods
Root Finding