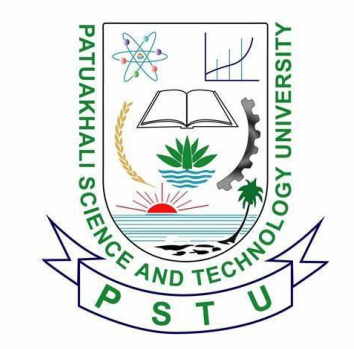

PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

COURSE CODE CCE 312
Numerical Methods Sessional

SUBMITTED TO:

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

SUBMITTED BY:

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

Assignment 04
Assignment title: Gauss Jordan
Date of submission: 14 Thu, 2025

Linear Equations

Sharafat Karim

CONTENTS

A linear equation is a mathematical statement that represents a straight line when graphed.

Then let's take input from user, in the following format,

```
Enter number of unknowns: 3
Enter row 1 in the format (ax + by + cz = d)
```

```
# User input (auto)
# N = int(input("Enter number of unknowns: "))

# arr = []
# for i in range(N):
#     arr.append(list(map(int, input().split())))

# User input (manual)
N = 3
arr = [[2, 1, -1, 8], [-3, -1, 2, -11], [-2, 1, 2, -3]]
```

```
# User input's output
print(N)

def print_arr(arr):
  for i in range(N):
    print(arr[i])
```

```
print_arr(arr)
```

```
3
[2, 1, -1, 8]
[-3, -1, 2, -11]
[-2, 1, 2, -3]
```

Gaussian Elimination

```
import copy

def gaussian_elimination(N, arr):
  arr = copy.deepcopy(arr)
  for i in range(N):
    for j in range(N, -1, -1):
      arr[i][j] /= arr[i][0+i]

    for j in range(i+1, N):
      for k in range(N, -1, -1):
        arr[j][k] -= arr[i][k] * arr[j][0+i]

    print_arr(arr)
    print()

  solve = [0 for j in range(N)]
  for i in range(N-1, -1, -1):
    for j in range(N):
      solve[i] = arr[i][N]
      for k in range(i+1, N):
        solve[i] -= arr[i][k] * solve[k]
    solve[i] /= arr[i][i]

  print(solve)

gaussian_elimination(N, arr)
```

```
[1.0, 0.5, -0.5, 4.0]
[0.0, 0.5, 0.5, 1.0]
[0.0, 2.0, 1.0, 5.0]

[1.0, 0.5, -0.5, 4.0]
[0.0, 1.0, 1.0, 2.0]
[0.0, 0.0, -1.0, 1.0]

[1.0, 0.5, -0.5, 4.0]
```

```
[0.0, 1.0, 1.0, 2.0]
[0.0, 0.0, 1.0, -1.0]

[2.0, 3.0, -1.0]
```

Gauss Jordan

```
import copy
arr = copy.deepcopy(arr)

def gauss_jordan(arr, N):
    for i in range(N):
        for j in range(N):
            if i != j:
                p = arr[j][i] / arr[i][i]
                for k in range(N+1):
                    arr[j][k] -= arr[i][k] * p
        print_arr(arr)
        print()

    for i in range(N):
        print(arr[i][3] / arr[i][i])

gauss_jordan(arr, N)
```

```
[2, 1, -1, 8]
[0.0, 0.5, 0.5, 1.0]
[0.0, 2.0, 1.0, 5.0]

[2.0, 0.0, -2.0, 6.0]
[0.0, 0.5, 0.5, 1.0]
[0.0, 0.0, -1.0, 1.0]

[2.0, 0.0, 0.0, 4.0]
[0.0, 0.5, 0.0, 1.5]
[0.0, 0.0, -1.0, 1.0]

2.0
3.0
-1.0
```

Numerical Methods

Numerical Methods