

Nama : NUR ANISAH

Nim : 200170015

MK : A2 - Metode Numerik

## 1. SOAL DAN PENYELESAIAN MANUAL

### METODE ELIMINASI GAUSS

Tentukan penyelesaian persamaan dibawah ini:

$$x_1 + 4x_2 + 5x_3 = 3$$

$$4x_1 + 2x_2 + x_3 = 5$$

$$x_1 + 4x_2 + 2x_3 = 7$$

Jawab

Persamaan diatas diubah kedalam bentuk matrik:

$$\begin{bmatrix} 1 & 4 & 5 \\ 4 & 2 & 1 \\ 1 & 4 & 2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 3 \\ 5 \\ 7 \end{Bmatrix} \quad (*)$$

Langkah 1 : Buat nilai 4 pada baris 2 menjadi nilai 0  
Dengan operasi Baris 2 - 4 baris 1 atau B2-4B1 hanya berlaku  
untuk semua elemen baris 2 sehingga persamaan (\*) menjadi:

$$\begin{bmatrix} 1 & 4 & 5 \\ 0 & -14 & -19 \\ 1 & 4 & 2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 3 \\ -7 \\ 7 \end{Bmatrix} \quad (**)$$

Langkah 2 : Buat nilai 1 dan 4 pada baris 3 menjadi nilai 0  
Dengan operasi B3-B1, berlaku untuk semua baris 3, sehingga  
persamaan (\*\*) menjadi :

$$\begin{bmatrix} 1 & 4 & 5 \\ 0 & -14 & -19 \\ 0 & 0 & -3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 3 \\ -7 \\ 1 \end{Bmatrix} \quad (***)$$

Persamaan (\*\*) sudah terbentuk matrik segitiga bawah sehingga dapat dilakukan back solving :

\* Hitung  $x_3$

$$-3x_3 = 4$$

$$x_3 = \frac{4}{-3}$$

$$x_3 = -1,333333$$

# Hitung  $x_2$

$$-14x_2 - 19x_3 = -7$$

$$-14x_2 - 19(-1,333333) = -7$$

$$-14x_2 = -7 - 25,333327$$

$$x_2 = \frac{-32,333327}{-14}$$

$$x_2 = 2,309524$$

# Hitung  $x_1$

$$x_1 + 4x_2 + 5x_3 = 3$$

$$x_1 + 4(2,309524) + 5(-1,333333) = 3$$

$$x_1 = 3 - 2,571431$$

$$x_1 = 0,428571$$

## 2. SOURCECODE

```
// PROGRAM PENYELESAIAN MATRIKS METODE ELIMINASI GAUSS

// Nama : NUR ANISAH

// Nim : 200170015

// MK : Metode Numerik

// Kelas : A2

#include<bits/stdc++.h>

#include <iostream>

#include <iomanip>

using namespace std;

#define N 3

// deklarasi variabel penampung inputan matriks

double matriks[3][4];

int i, j;

int forwardElim(double mat[N][N+1]);

void backSub(double mat[N][N+1]);

void gaussianElimination(double mat[N][N+1]){

    int singular_flag = forwardElim(mat);

    if (singular_flag != -1){

        printf("Matriks berbentuk sama\n");

        if (mat[singular_flag][N])

            printf("Sistem tidak konsisten");

        else

            printf("Matriks memiliki beberapa solusi ");

        return;

    }

    backSub(mat);
```

```

}

void swap_row(double mat[N][N+1], int i, int j){

    for (int k=0; k<=N; k++){
        double temp = mat[i][k];
        mat[i][k] = mat[j][k];
        mat[j][k] = temp;
    }
}

void print(double mat[N][N+1]){

    for (int i=0; i<N; i++, printf("\n"))
        for (int j=0; j<=N; j++)
            printf("%lf ", mat[i][j]);
    printf("\n");
}

int forwardElim(double mat[N][N+1]){

    for (int k=0; k<N; k++){

        int i_max = k;
        int v_max = mat[i_max][k];
        for (int i = k+1; i < N; i++)
            if (abs(mat[i][k]) > v_max)
                v_max = mat[i][k], i_max = i;

        if (!mat[k][i_max])
            return k;

        if (i_max != k)
            swap_row(mat, k, i_max);

        for (int i=k+1; i<N; i++){
            double f = mat[i][k]/mat[k][k];
            for (int j=k+1; j<=N; j++)

```

```

        mat[i][j] -= mat[k][j]*f;
        mat[i][k] = 0;
    }
}

return -1;
}

void backSub(double mat[N][N+1]){
    double x[N];
    for (int i = N-1; i >= 0; i--){
        x[i] = mat[i][N];
        for (int j=i+1; j<N; j++){
            x[i] -= mat[i][j]*x[j];
        }
        x[i] = x[i]/mat[i][i];
    }
    printf("\nSolusi untuk matriks tersebut adalah : \n");
    for (int i=0; i<N; i++)
        printf("%lf\n", x[i]);
}

// Program utama
int main(){
    printf("Masukkan nilai matriks : \n");

    // proses input array
    for(i = 0; i < N; i++){
        for(j = 0; j < N+1; j++){
            cout << "Baris " <<i+1<<, kolom "<<j+1<< " = ";
            cin >> matriks[i][j];
        }
    }
}

```

```

        cout << endl;
    }

    gaussianElimination(matriks);

    return 0;
}

```

### 3. SCREENSHOOT PROGRAM

```

D:\Matiks\project matriks\9. array matriks.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
[*] 9. array matriks.cpp
1 // PROGRAM PENYELESAIAN MATERIKS METODE ELIMINASI GAUSS
2
3 // Nama : NUR ANISAH
4 // Nim : 200170015
5 // MK : Metode Numerik
6 // Kelas : A2
7
8
9 #include<bits/stdc++.h>
10 #include <iostream>
11 #include <iomanip>
12
13 using namespace std;
14
15 #define N 3
16
17 // deklarasi variabel penampung inputan matriks
18 double matriks[3][4];
19 int i, j;
20
21 int forwardElim(double mat[N][N+1]);
22
23 void backSub(double mat[N][N+1]);
24
25 void gaussianElimination(double mat[N][N+1]){
26     int singular_flag = forwardElim(mat);
27     if (singular_flag != -1){
28         printf("Matriks berbentuk sama\n");
29         if (mat[singular_flag][N])
30             printf("Sistem tidak konsisten");
31         else
32             printf("Matriks memiliki beberapa solusi ");
33         return;
34     }
35     backSub(mat);
36 }
37
38 void swap_row(double mat[N][N+1], int i, int j){
39     for (int k=0; k<=N; k++){
40         double temp = mat[i][k];
41         mat[i][k] = mat[j][k];
42         mat[j][k] = temp;
43     }
44 }
45
46 void print(double mat[N][N+1]){
47     for (int i=0; i<N; i++)
48         for (int j=0; j<N; j++)
49             printf("%lf ", mat[i][j]);
50
51     printf("\n");
52 }
53
54 int forwardElim(double mat[N][N+1]){
55     for (int k=0; k<N; k++){
56         int i_max = k;
57         int v_max = mat[i_max][k];
58
59         for (int i = k+1; i < N; i++)
60             if (abs(mat[i][k]) > v_max)
61                 v_max = mat[i][k], i_max = i;
62
63         if (!mat[k][i_max])
64             return k;
65
66         if (i_max != k)
67             swap_row(mat, k, i_max);
68     }
69 }

```

```

68     for (int i=k+1; i<N; i++){
69         double f = mat[i][k]/mat[k][k];
70         for (int j=k+1; j<=N; j++){
71             mat[i][j] -= mat[k][j]*f;
72         }
73         mat[i][k] = 0;
74     }
75     return -1;
76 }
77 }
78 }
79 }
80
81 void backSub(double mat[N][N+1]){
82     double x[N];
83     for (int i = N-1; i >= 0; i--){
84         x[i] = mat[i][N];
85         for (int j=i+1; j<N; j++){
86             x[i] -= mat[i][j]*x[j];
87         }
88         x[i] = x[i]/mat[i][i];
89     }
90
91     printf("\nSolusi untuk matriks tersebut adalah : \n");
92     for (int i=0; i<N; i++)
93         printf("%lf\n", x[i]);
94     }
95 }
96
97 // Program utama
98 int main(){
99     printf("Masukkan nilai matriks : \n");
100
101    // proses input array
102    for(i = 0; i < N; i++){
103        for(j = 0; j < N+1; j++){
104            cout << "Baris " <<i+1<<, kolom " <<j+1<< " = ";
105            cin >> matriks[i][j];
106        }
107        cout << endl;
108    }
109
110    gaussianElimination(matriks);
111
112    return 0;
113 }
114

```

Compiler Resources Compile Log Debug Find Results

Type here to search

27°C B

#### 4. HASIL MENJALANKAN PROGRAM

```

D:\Matriks\project matriks\9.array matriks.exe
Masukkan nilai matriks :
Baris 1, kolom 1 = 1
Baris 1, kolom 2 = 4
Baris 1, kolom 3 = 5
Baris 1, kolom 4 = 3

Baris 2, kolom 1 = 4
Baris 2, kolom 2 = 2
Baris 2, kolom 3 = 1
Baris 2, kolom 4 = 5

Baris 3, kolom 1 = 1
Baris 3, kolom 2 = 4
Baris 3, kolom 3 = 2
Baris 3, kolom 4 = 7

Solusi untuk matriks tersebut adalah :
0.428571
2.309524
-1.333333

-----
Process exited after 39.61 seconds with return value 0
Press any key to continue . . .

```

Type here to search

27°C