Tugas Sistem Persamaan Linear Fisis

Rendy Putra Pratama - 140310230037

1 Sistem Pegas

Sistem yang ditunjukkan pada gambar terdiri dari **n** buah pegas linier yang menopang n buah benda bermassa **m**. Konstanta pegas disimbolkan dengan **k** berat benda adalah **w** dan **x** adalah perpindahan benda diukur dari posisi pegas saat tidak terdeformasi. Persamaan perpindahan diperoleh dengan menuliskan persamaan keseimbangan setiap benda dan mengganti $\vec{F}$ dengan gaya pegas diperoleh persamaan sebagai berikut :

$$\begin{aligned} (k_1 + k_2) \cdot x_1 - k_2 \cdot x_2 &= w_1 \\ -k_i \cdot x_{i-1} + (k_i + k_{i+1})x_i - k_{i+1} \cdot x_{i+1} &= w_i \\ i &= 1, 2, 3, ..., n-1 \\ -k_n \cdot x_{n-1} + k_n \cdot x_n &= w_n \end{aligned}$$

Tuliskan kode python untuk mendapatkan solusi dari sistem persamaan tersebut dengan metode **Gauss Seidel** jika jumlah benda dan pegas 5 buah Diketahui nilai konstanta pegas dan berat benda adalah sebagai berikut

$$\begin{aligned} k_1 = k_2 = k_3 &= 10\ {}^{N}/_{mm} \\ w_1 = w_3 = w_5 &= 100\ N \\ k_4 = k_5 &= 5\ {}^{N}/_{mm} \\ w_2 = w_4 &= 50\ N \end{aligned}$$

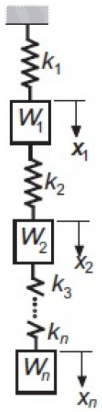

Figure 1: Sistem Pegas

1.1 Listing Program

```
# Jacobi Methods
'''
import numpy as np
# buat matriks persamaan linear [x,y]
A = np.array([[2.0,3.0],
              [1.0,-6.0]])
# matriks hasil dari persamaan linear
Y = np.array([5.0,9.0])

#matriks kosong untuk menempatkan hasil solusi
x = np.array([0.0,0.0])
# iterasi dimulai dari 0
i = 0
# pengondisian true pada while
e = 1

while e > 1e-6:

    solAB = [] # wadah solusi untuk variabel x dan y pada persamaan linear
    n = len(x)
    for i in range(n):
        sigma = 0
        for j in range(0,n):
            if j != i:
                sigma += A[i][j]*x[j]
        xn = (Y[i]-sigma)/A[i][i]
        solAB.append(xn)
    e = abs(max(x)-max(solAB))
    x = np.array(solAB)

    i += 1
print(f"akar persamaan {x} \t iterasi - {i}") '''

import numpy as np

# Flowchart

# Buat array berdasarkan persamaan yang akan dicari solusinya
A = np.array([
              [1.0,3.0,-1.0,0.3],
              [0.0,1.0,-9.8,0.0],
              [3.0,-4.0,2.0,0.0],
              [1.4,0.0,-2.4,0.0]
              ])
# Array hasil dari persamaan sebelumnya
Y = np.array([28,12,-2,3])
# Array untuk hasil x1,x2,x3
x = np.array([0.0,0.0,0.0,0.0])

# inisiasi awal
itr = 0
e = 1

while e > 0.000001:
    solveA = []
    n = len(x)
    for i in range(n):
        sigma = 0
        for j in range(0,n):
            if j != i:
                sigma += A[i][j]*x[j]
        xNow = (Y[i]-sigma)/A[i][i]
        solveA.append(xNow) # di append = biar array solveA kosong, dan bisa di input
    lagi dengan nilai baru
    e = abs(max(x)-max(solveA))
    x = np.array(solveA)
    itr += 1
print("Perhitungan dengan Metode Jacobi")
print(f"Solusi x1 = {x[0]:.3f}; x2 = {x[1]:.3f}; x3 = {x[2]:.3f}")
print(f"Matriks solusi X1,X2,X3,X4 = {x}")
```

Listing 1: Octave sample code

1.2 Output Program

2 Rangkaian Listrik