

Nama : Alfianri Manihuruk

NIM : 120450088

Kelas : RB

Matkul: Pemograman Berbasis Fungsi

Non-Linear Pendulum

Suppose that the pendulum is described by the nonlinear second order differential equation as follows:

$$\frac{d^2\alpha}{dt^2} + \frac{g}{L}\sin\alpha = 0.$$

■ $\alpha(t)?$

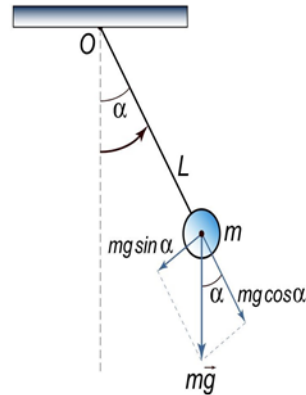


Figure 1.

Solusi Persamaan Diferensial Orde 2

Salah satu metode untuk menyelesaikan persamaan diferensial tersebut adalah dengan menggunakan metode numerik menggunakan metode euler untuk PD orde 2.

Code tersebut adalah metode euler yang dapat digunakan untuk menyelesaikan persamaan diferensial non linear pendulum. Misal metode tersebut disimpan di modul bernama solver.py

Definisikan fungsi Func sebagai fungsi yang me return nilai $-g/L * \sin(a)$!

```
def euler(t,h,y,dy,Func):
    d2y = Func(t,y,dy)
    y_next = y + (h * dy)
    dy_next = dy + (h * d2y)
    return ( y_next, dy_next )

def cauchy_euler(params,Func):
    # Initial Condition
    t0 = params['t0']
    t_akhir = params['t_akhir']
    h = params['h']
    y0 = params['y0']
    dy0 = params['dy0']

    res_euler = []
    t = []
    step = int((t_akhir - t0) / h)

    for i in range(step):
        tm = (i + 1) * h
        (y_next, dy_next) = euler(tm, h, y0, dy0, Func)
        res_euler.append(y_next)
        t.append(tm)
        y0 = y_next
        dy0 = dy_next

    return (t,res_euler)
```

1. Apakah modul solver.py dapat digunakan untuk menyelesaikan persamaan differensial orde 2 selain kasus non linear pendulum? Jelaskan mengapa!

Dapat, karena sistem pengerjaannya sama, tinggal menyesuaikan dengan permasalahan yang ada.

2. Implementasikan solver tersebut dengan cara membuat file solver.py

```
1  """
2  sesuai soal, pada bagian ini kita membuat modul dengan nama (solver.py),
3  untuk fungsinya sendiri telah tersedia pada soal
4  """
5
6
7  def euler(t, h, y, dy, Func): # defenisikan fungsi euler
8      d2y = Func(t, y, dy)
9      y_next = y + (h * dy)
10     dy_next = dy + (h * d2y)
11     return (y_next, dy_next)
12
13  def euler_cromer(t, h, y, dy, Func): # defenisikan fungsi euler_cromer
14      d2y = Func(t, y, dy)
15      dy_next = dy + (h * d2y)
16      y_next = y + (h * dy_next)
17
18     return (y_next, dy_next)
19
```

3. Untuk menyelesaikan persamaan dengan solver.py, bentuk fungsi harus diubah menjadi:

$$\frac{d^2\alpha}{dt^2} = -\frac{g}{L} * \sin(\alpha)$$

Definisikan fungsi Func sebagai fungsi yang me return nilai $-g/L * \sin(a)$!

```
def Func(t,u,du):
    return -w0 * sin(u) - k*du #rumus func yang me return nilai -g/l * sin(a)
```

$u = \alpha$

$-w0 = -g/L$

Parameter	Deskripsi	Value
g	Konstanta Gravitasi	9.8 m/s ²
L	Panjang Tali Pendulum	1 m
t0	Waktu Awal	0 detik
tn	Waktu Akhir	4 detik
h	Step Size	0.001
α_0	Nilai awal alpha	$0.5 * 3.14$

4. Menggunakan Parameter Parameter yang ada dalam tabel diatas, buatlah program yang menggunakan solver.py untuk menemukan solusi persamaan diferensial non linear tersebut! Hint (Solusi Akhir berupa plot).

```

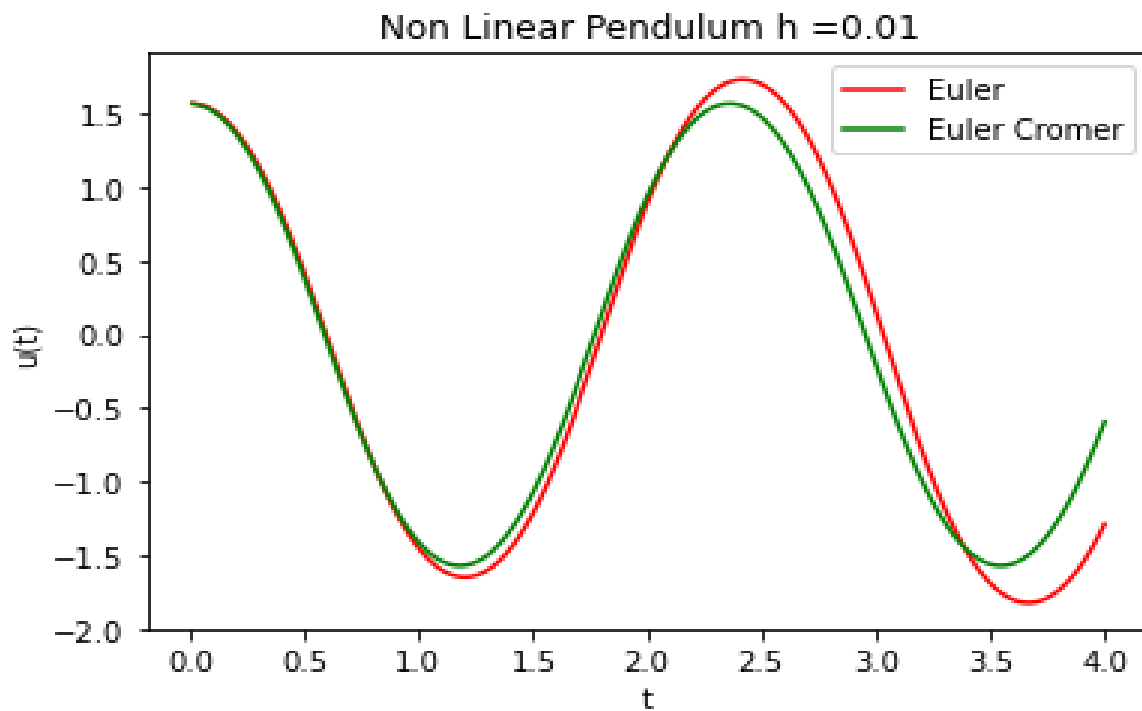
1 from solver import * # import module solver yang telah dibuat tadi
2 from math import *
3 import matplotlib.pyplot as plt
4
5 # membuat semua parameter yang dibutuhkan
6 g = 9.81 # gaya gravitasi
7 l = 1 # panjang pendulum
8 k = 0 # koefisien
9 u0 = 0.5 * pi # nilai alpha
10 du0 = 0 # diffrensial
11 t0 = 0 # waktu awal
12 t_akhir = 4 # waktu akhir
13 h = 0.01 # step
14 w0 = g/l
15
16 def Func(t,u,du):
17     return -w0 * sin(u) - k*du #rumus func yang me return nilai -g/l * sin(a)
18
19 #membuat List kosong
20 res_euler = []
21 res_eulercromer = []
22 t = []
23 step = int((t_akhir - t0) / h) #rumus step 1 yang akan menginput angka yang diketahui
24
25 for i in range(step): #melakukan perulangan untuk rumus yang telah di buat (step)
26     tm = (i + 1) * h #rumus step di kali ketinggian
27     (u_next, du_next) = euler(tm, h, u0, du0, Func) # substitusi nilai pada rumus euler
28     res_euler.append(u_next) # menambahkan nilai ke list res_euler
29     t.append(tm) # menambahkan hasilnya ke list t
30     u0 = u_next
31     du0 = du_next
32
33 # update setiap nilai
34 t = []
35 u0 = 0.5 * pi
36 du0 = 0
37 d2u0 = Func(t0,u0,du0)

```

```

39 for i in range(step):
40     tm = (i + 1) * h
41     '''setelah step 1 dilakukan selanjutnya nilai yang di hasilkan di
42     masukkan/subtitusi ke per 2 yaitu euler_cromer
43     '''
44     (u_next, du_next) = euler_cromer(tm, h, u0, du0, Func) # masukkan nilai ke fungsi euler cromer
45     res_eulercromer.append(u_next) # menambahkan nilai yang di hasilkan ke list res_eulercromer
46     t.append(tm) # update nilai t
47     u0 = u_next
48     du0 = du_next
49
50 # visualisasi hasil dari fungsi euler dan dan fungsi euler_cromer
51 plt.title('Non Linear Pendulum h =0.01')
52 plt.plot(t, res_euler, color='r', label = 'Euler')
53 plt.plot(t, res_eulercromer, color='g', label = 'Euler Cromer')
54 plt.xlabel('t')
55 plt.ylabel('u(t)')
56 plt.legend()
57
58 plt.show()

```



$$\frac{d^2y}{dx^2} = -y - \frac{dy}{dx} + \sin^2(x)$$

5. Perhatikan Persamaan Diferensial Diatas! Buatlah program untuk menyelesaikan PD tersebut dengan menggunakan solver.py sebagai modul dengan parameter berikut!

Parameter	Deskripsi	Value
X_0	X awal	0
X_n	X akhir	50
h	Step Size	0.05
y(X_0) = y_0	Nilai awal Y	1
y'(X_0) = y'_0	Nilai awal dy/dx	-9/2

```

1  #import module yang telah dibuat dan Library yang di butuhkan
2  from solver import *
3  from math import *
4  import matplotlib.pyplot as plt
5  from metode_euler import *
6
7
8  # example y'' = -y - y' + sin^2(t)
9  #membuat fungsi Func
10 def Func(t,y,dy):
11     return (-1 *y) + (-1 * dy) + sin(t)**2
12
13 #membuat fungsi eksak(mereturn rumus )
14 def eksak(t):
15     return ((5/13)*exp(-0.5*t) * cos( 0.5 * sqrt(3) * t ) ) - ( 108/(13 * sqrt(3)) * exp(-0.5*t) * sin(0.5 * sqrt(3) * t )
16
17 #membuat fungsi untuk mencari solusi analik dengan memasukan setiap parameter ke rumus yang telah di buat tadi
18 def solusi_analitik(params):
19     t0 = params['t0']
20     t_akhir = params['t_akhir']
21     h = params['h']
22     step = int((t_akhir - t0) / h)
23     t = []
24     res_eksak = []
25     for i in range(step):
26         tm = (i + 1) * h
27         y_next = eksak(tm)
28         res_eksak.append(y_next)
29         t.append(tm)
30     return (t,res_eksak)

```

```

32 # membuat nilai paramer dalam bentuk dictionary
33 params = {
34     't0' : 0,
35     "t_akhir" : 50,
36     "h" : 5 * 10**-2,
37     "y0" : 1,
38     "dy0" : -9/2
39 }
40
41 params2 = {
42     't0' : 0,
43     "t_akhir" : 50,
44     "h" : 5 * 10**-1,
45     "y0" : 1,
46     "dy0" : -9/2
47 }
48
49 #membaut list kosong untuk setiap nilai yang di hasilkan
50 res_eksak = []
51 res_euler = []
52 res_euler_2 = []
53 res_eulercromer = []
54 res_eulercromer_2 = []
55 t = []
56
57 plt.subplot(2,1,1)
58
59
60 # visualisasi Plot Euler
61 (t,res_eksak) = solusi_analitik(params)
62 plt.plot(t,res_eksak,color='k',label='Solusi Analitik')
63
64 (t,res_euler) = cauchy_euler(params,Func)
65 plt.plot(t,res_euler,color='g', label = 'h = 0.05')
66
67 (t,res_euler_2) = cauchy_euler(params2,Func)
68 plt.plot(t,res_euler_2,color='r',label = 'h = 0.5')
69 plt.title('Metode Euler')
70 plt.xlabel('t')
71 plt.ylabel('y(t)')

```

```

72 plt.ylim(-4,2)
73 plt.legend()
74 plt.subplot(2,1,2)
75
76 # visualisasi plot euler cromer
77 (t,res_eksak) = solusi_analitik(params)
78 plt.plot(t,res_eksak,color='k',label='Solusi Analitik')
79
80 (t,res_eulercromer) = cauchy_eulercromer(params,Func)
81 plt.plot(t,res_eulercromer,color='g', label = 'h = 0.05')
82
83 (t,res_eulercromer_2) = cauchy_eulercromer(params2,Func)
84 plt.plot(t,res_eulercromer_2,color='r',label = 'h = 0.5')
85 plt.title('Metode Euler-Cromer')
86 plt.xlabel('t')
87 plt.ylabel('y(t)')
88 plt.ylim(-4,2)
89 plt.legend()
90 plt.figure()

```

```

93 # visualisasi plot comparison
94 (t,res_eksak) = solusi_analitik(params2)
95 plt.plot(t,res_eksak,color='k',label='Solusi Analitik')
96
97 (t,res_euler_2) = cauchy_euler(params2,Func)
98 plt.plot(t,res_euler_2,color='r',label = 'Euler ')
99
100 (t,res_eulercromer_2) = cauchy_eulercromer(params2,Func)
101 plt.plot(t,res_eulercromer_2,color='c',label = 'Euler-Cromer')
102 plt.title('Euler vs Euler-Cromer at h= 0.5')
103 plt.xlabel('t')
104 plt.ylabel('y(t)')
105 plt.ylim(-4,2)
106 plt.legend()
107
108
109
110 plt.show()

```

Hasil dalam berupa grafik

