ps1_1

February 12, 2023

[800]:
```
import math
import numpy as np
import matplotlib.pyplot as plt
```

[801]:
```
import math
import numpy as np
```

[836]:
```
gravity = 32.2  # ft/s^2
initial_position = math.pi/4
initial_velocity = 0
arm_length = 2 # feet
end_time = 4  # seconds
time_step = 0.05  # seconds
time_range = np.arange(0,end_time + time_step,time_step)
```

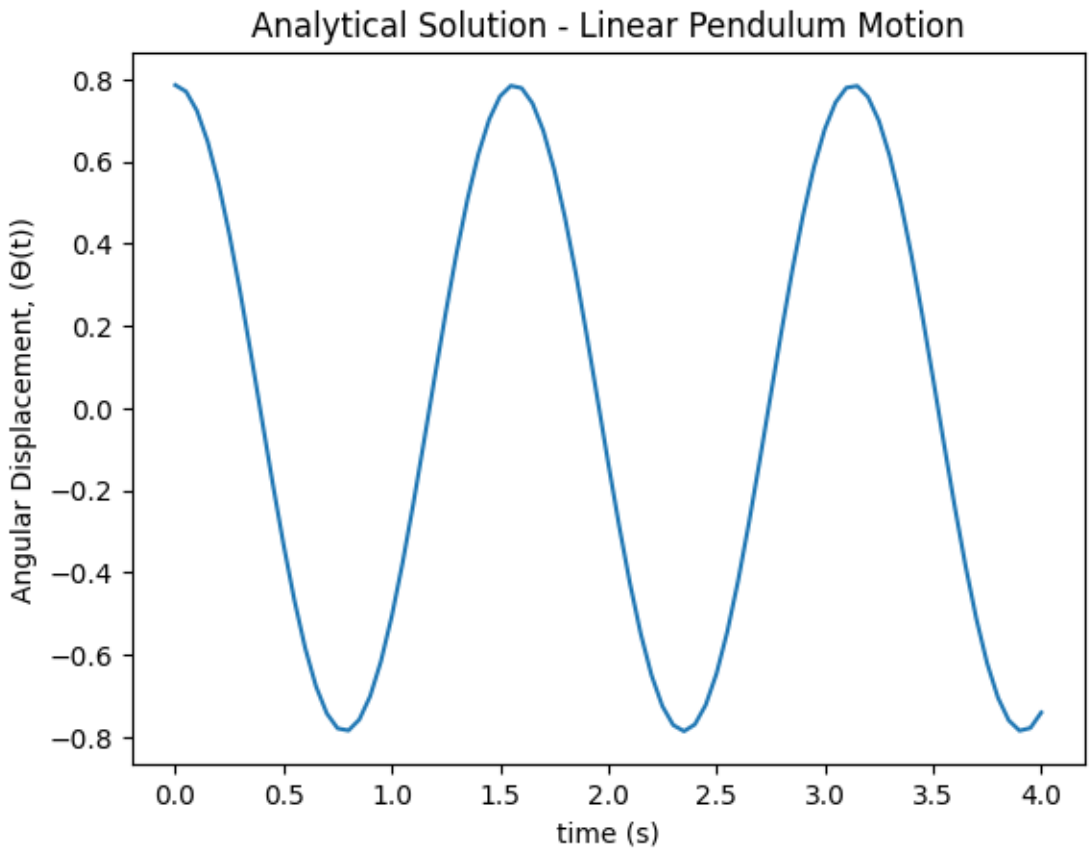
$$\theta(t) = A\sin(\lambda t) + B\cos(\lambda t)$$

[837]:
```
def pendulum_analytical_angle_sol(velocity_naut, theta_naut, length, end_time):
    lambd = math.sqrt(gravity/length)
    A = velocity_naut / lambd
    B = theta_naut
    angle_in_time = []
    for time in time_range:
        angle_in_time.append((A * math.sin(lambd * time)) + (B * math.cos(lambd␣
 ↪* time)))
    return angle_in_time
```

[838]:
```
analytical_solution_output =␣
 ↪pendulum_analytical_angle_sol(initial_velocity,initial_position,arm_length,end_time)
```

[839]:
```
plt.xlabel('time (s)')
plt.ylabel('Angular Displacement, ( (t))')
plt.title('Analytical Solution - Linear Pendulum Motion')
plot_time = np.linspace(0, end_time, len(analytical_solution_output))
plt.plot(plot_time,analytical_solution_output)
```

[839]:
```
[<matplotlib.lines.Line2D at 0x22ac03d4970>]
```



$$v_{i+1} = v_i - \Delta t \lambda^2 \theta_i$$

$$\theta_{i+1} = \theta_i + \Delta t v_i$$

[840]:
```
def pendulum_numerical_angle_sol_foward_difference(velocity_naut, theta_naut,␣
↪length, end_time, time_step):
    lambd = math.sqrt(gravity/length)
    velocity_series = [velocity_naut]
    angle_series = [theta_naut]

    for i in range(len(time_range) - 1):
        velocity_series.append(velocity_series[i] - (time_step * lambd**2 *␣
↪angle_series[i]))
        angle_series.append(angle_series[i] + (time_step * velocity_series[i]))

    return angle_series
```

$$v_{i+1} = v_{i-1} - \Delta t \lambda^2 \theta_{i-1}$$

$$\theta_{i+1} = \theta_{i-1} + \Delta t v_i$$

[841]:
```
def pendulum_numerical_angle_sol_backward_difference(velocity_naut, theta_naut,␣
↪length, end_time, time_step):
    lambd = math.sqrt(gravity/length)
    velocity_series = [velocity_naut]
    angle_series = [theta_naut]

    for i in range(1, len(time_range)):
        velocity_series.append(velocity_series[i-1] - (time_step * lambd**2 *␣
↪angle_series[i-1]))
        angle_series.append(angle_series[i-1] + (time_step *␣
↪velocity_series[i]))

    return angle_series
```

$$v_{i+1} = v_{i-1} - 2\Delta t \lambda^2 \theta_{i-1}$$

$$\theta_{i+1} = \frac{\theta_{i-1} + \Delta t v_{i-1} + v_i}{2}$$

[842]:
```
def pendulum_numerical_angle_sol_central_difference(velocity_naut, theta_naut,␣
↪length, end_time, time_step):
    lambd = math.sqrt(gravity/length)
    velocity_series = [velocity_naut]
    angle_series = [theta_naut]

    for i in range(1, len(time_range)):
        velocity_series.append(velocity_series[i-1] - (2 * time_step * lambd**2␣
↪* angle_series[i-1]))
        angle_series.append((angle_series[i-1] + (time_step *␣
↪(velocity_series[i-1] + velocity_series[i])) / 2))

    return angle_series
```

[843]:
```
numerical_solution_output_forward_difference =␣
↪pendulum_numerical_angle_sol_foward_difference(initial_velocity,initial_position,arm_length
```

[844]:
```
numerical_solution_output_backward_difference =␣
↪pendulum_numerical_angle_sol_backward_difference(initial_velocity,initial_position,arm_leng
```

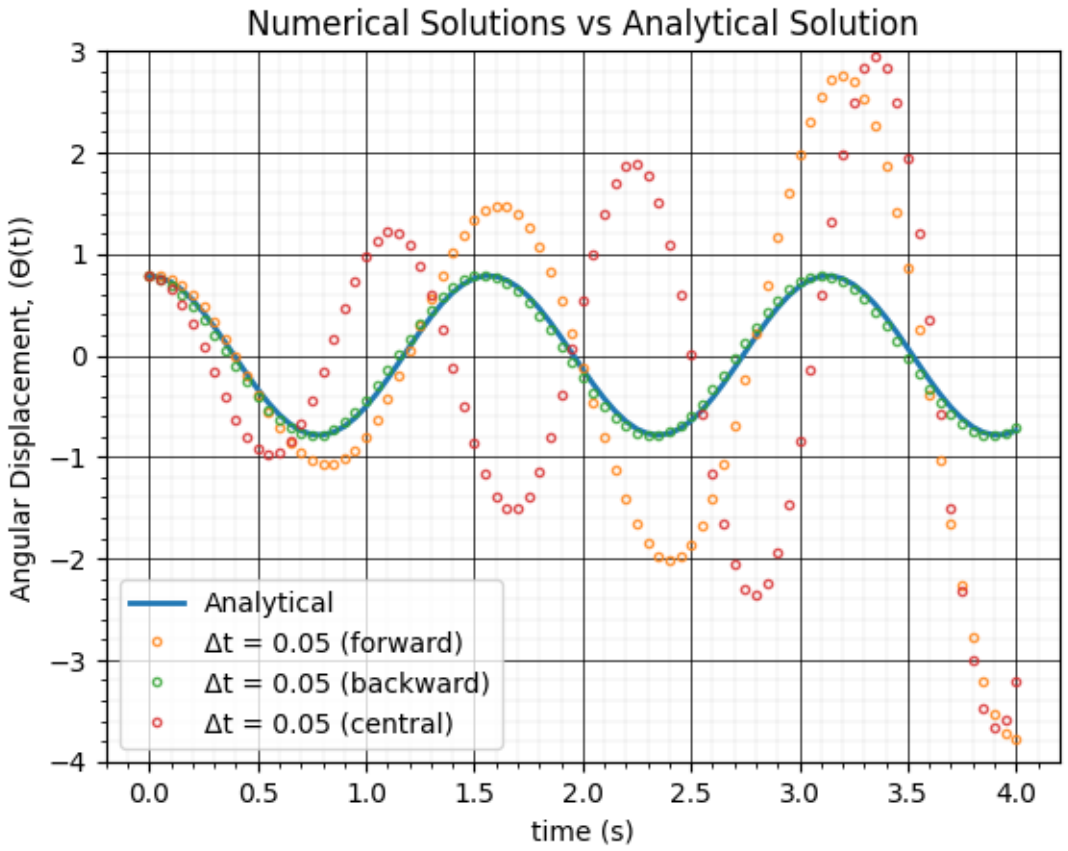
[845]:
```
numerical_solution_output_central_difference =␣
↪pendulum_numerical_angle_sol_central_difference(initial_velocity,initial_position,arm_lengt
```

[865]:
```
plt.xlabel('time (s)')
plt.ylabel('Angular Displacement, ( (t))')
plt.title('Numerical Solutions vs Analytical Solution')
plt.grid(color = 'black', which = 'major', linestyle = '-', linewidth = 0.5)
plt.grid(color = 'black', which = 'minor', linestyle = '--', linewidth = 0.05)
plt.minorticks_on()
plt.ylim([-4, 3])

plot_time = np.linspace(0, end_time, len(analytical_solution_output))
plt.plot(plot_time,analytical_solution_output, label='Analytical', linewidth=2)
plt.plot(plot_time,numerical_solution_output_forward_difference,"o" ,␣
 ↪label=f'Δt = {time_step} (forward)', markersize=3, alpha = 0.8, mfc='none')
plt.plot(plot_time,numerical_solution_output_backward_difference,"o" ,␣
 ↪label=f'Δt = {time_step} (backward)', markersize=3, alpha = 0.8, mfc='none')
plt.plot(plot_time,numerical_solution_output_central_difference,"o" ,␣
 ↪label=f'Δt = {time_step} (central)', markersize=3, alpha = 0.8, mfc='none')

plt.legend(loc="lower left")
```

[865]:
```
<matplotlib.legend.Legend at 0x22ac21d1070>
```



[866]:
```
forward_error_plot = []
def calculate_error_forward(numerical_solution, analytical_solution):
    for i in range(len(time_range)):
        forward_error_plot.append(abs(numerical_solution[i] -␣
 ↪analytical_solution[i]))
        print( f'error at time {i*time_step:.2f} = {abs(numerical_solution[i] -␣
 ↪analytical_solution[i]):.5f}')
```

[867]:
```
backward_error_plot = []
def calculate_error_backward(numerical_solution, analytical_solution):
    for i in range(len(time_range)):
        backward_error_plot.append(abs(numerical_solution[i] -␣
 ↪analytical_solution[i]))
        print( f'error at time {i*time_step:.2f} = {abs(numerical_solution[i] -␣
 ↪analytical_solution[i]):.5f}')
```

[868]:
```
central_error_plot = []
def calculate_error_central(numerical_solution, analytical_solution):
    for i in range(len(time_range)):
        central_error_plot.append(abs(numerical_solution[i] -␣
 ↪analytical_solution[i]))
        print( f'error at time {i*time_step:.2f} = {abs(numerical_solution[i] -␣
 ↪analytical_solution[i]):.5f}')
```

[869]:
```
calculate_error_forward(numerical_solution_output_forward_difference,analytical_solution_outpu
```

```
error at time 0.00 = 0.00000
error at time 0.05 = 0.01575
error at time 0.10 = 0.03077
error at time 0.15 = 0.04318
error at time 0.20 = 0.05121
error at time 0.25 = 0.05335
error at time 0.30 = 0.04840
error at time 0.35 = 0.03561
error at time 0.40 = 0.01474
error at time 0.45 = 0.01387
error at time 0.50 = 0.04932
error at time 0.55 = 0.09015
error at time 0.60 = 0.13438
error at time 0.65 = 0.17959
error at time 0.70 = 0.22305
error at time 0.75 = 0.26184
error at time 0.80 = 0.29300
error at time 0.85 = 0.31373
error at time 0.90 = 0.32149
error at time 0.95 = 0.31425
error at time 1.00 = 0.29057
error at time 1.05 = 0.24976
```

```
error at time 1.10 = 0.19197
error at time 1.15 = 0.11827
error at time 1.20 = 0.03063
error at time 1.25 = 0.06810
error at time 1.30 = 0.17423
error at time 1.35 = 0.28339
error at time 1.40 = 0.39068
error at time 1.45 = 0.49087
error at time 1.50 = 0.57863
error at time 1.55 = 0.64879
error at time 1.60 = 0.69659
error at time 1.65 = 0.71794
error at time 1.70 = 0.70967
error at time 1.75 = 0.66972
error at time 1.80 = 0.59736
error at time 1.85 = 0.49327
error at time 1.90 = 0.35963
error at time 1.95 = 0.20013
error at time 2.00 = 0.01987
error at time 2.05 = 0.17475
error at time 2.10 = 0.37622
error at time 2.15 = 0.57625
error at time 2.20 = 0.76601
error at time 2.25 = 0.93655
error at time 2.30 = 1.07919
error at time 2.35 = 1.18587
error at time 2.40 = 1.24962
error at time 2.45 = 1.26487
error at time 2.50 = 1.22782
error at time 2.55 = 1.13670
error at time 2.60 = 0.99198
error at time 2.65 = 0.79646
error at time 2.70 = 0.55531
error at time 2.75 = 0.27597
error at time 2.80 = 0.03203
error at time 2.85 = 0.35739
error at time 2.90 = 0.68739
error at time 2.95 = 1.00836
error at time 3.00 = 1.30626
error at time 3.05 = 1.56721
error at time 3.10 = 1.77812
error at time 3.15 = 1.92727
error at time 3.20 = 2.00491
error at time 3.25 = 2.00380
error at time 3.30 = 1.91957
error at time 3.35 = 1.75117
error at time 3.40 = 1.50101
error at time 3.45 = 1.17508
```

```
error at time 3.50 = 0.78285
error at time 3.55 = 0.33710
error at time 3.60 = 0.14648
error at time 3.65 = 0.64980
error at time 3.70 = 1.15297
error at time 3.75 = 1.63512
error at time 3.80 = 2.07514
error at time 3.85 = 2.45263
error at time 3.90 = 2.74872
error at time 3.95 = 2.94699
error at time 4.00 = 3.03426
```

[870]:
```
calculate_error_backward(numerical_solution_output_backward_difference,analytical_solution_out
```

```
error at time 0.00 = 0.00000
error at time 0.05 = 0.01586
error at time 0.10 = 0.03118
error at time 0.15 = 0.04535
error at time 0.20 = 0.05778
error at time 0.25 = 0.06796
error at time 0.30 = 0.07545
error at time 0.35 = 0.07995
error at time 0.40 = 0.08125
error at time 0.45 = 0.07928
error at time 0.50 = 0.07409
error at time 0.55 = 0.06587
error at time 0.60 = 0.05494
error at time 0.65 = 0.04172
error at time 0.70 = 0.02672
error at time 0.75 = 0.01056
error at time 0.80 = 0.00614
error at time 0.85 = 0.02270
error at time 0.90 = 0.03844
error at time 0.95 = 0.05273
error at time 1.00 = 0.06498
error at time 1.05 = 0.07469
error at time 1.10 = 0.08144
error at time 1.15 = 0.08494
error at time 1.20 = 0.08503
error at time 1.25 = 0.08169
error at time 1.30 = 0.07504
error at time 1.35 = 0.06530
error at time 1.40 = 0.05288
error at time 1.45 = 0.03824
error at time 1.50 = 0.02196
error at time 1.55 = 0.00471
error at time 1.60 = 0.01285
error at time 1.65 = 0.02999
```

```
error at time 1.70 = 0.04603
error at time 1.75 = 0.06030
error at time 1.80 = 0.07222
error at time 1.85 = 0.08130
error at time 1.90 = 0.08716
error at time 1.95 = 0.08952
error at time 2.00 = 0.08829
error at time 2.05 = 0.08349
error at time 2.10 = 0.07528
error at time 2.15 = 0.06399
error at time 2.20 = 0.05005
error at time 2.25 = 0.03401
error at time 2.30 = 0.01650
error at time 2.35 = 0.00177
error at time 2.40 = 0.02009
error at time 2.45 = 0.03769
error at time 2.50 = 0.05388
error at time 2.55 = 0.06798
error at time 2.60 = 0.07943
error at time 2.65 = 0.08773
error at time 2.70 = 0.09254
error at time 2.75 = 0.09364
error at time 2.80 = 0.09097
error at time 2.85 = 0.08461
error at time 2.90 = 0.07480
error at time 2.95 = 0.06191
error at time 3.00 = 0.04646
error at time 3.05 = 0.02904
error at time 3.10 = 0.01036
error at time 3.15 = 0.00885
error at time 3.20 = 0.02780
error at time 3.25 = 0.04574
error at time 3.30 = 0.06194
error at time 3.35 = 0.07572
error at time 3.40 = 0.08652
error at time 3.45 = 0.09389
error at time 3.50 = 0.09751
error at time 3.55 = 0.09722
error at time 3.60 = 0.09300
error at time 3.65 = 0.08501
error at time 3.70 = 0.07354
error at time 3.75 = 0.05905
error at time 3.80 = 0.04209
error at time 3.85 = 0.02335
error at time 3.90 = 0.00356
error at time 3.95 = 0.01647
error at time 4.00 = 0.03595
```

[871]:
```
calculate_error_central(numerical_solution_output_central_difference,analytical_solution_outpu
```

```
error at time 0.00 = 0.00000
error at time 0.05 = 0.01586
error at time 0.10 = 0.06280
error at time 0.15 = 0.13764
error at time 0.20 = 0.23361
error at time 0.25 = 0.34077
error at time 0.30 = 0.44683
error at time 0.35 = 0.53822
error at time 0.40 = 0.60133
error at time 0.45 = 0.62381
error at time 0.50 = 0.59587
error at time 0.55 = 0.51144
error at time 0.60 = 0.36904
error at time 0.65 = 0.17234
error at time 0.70 = 0.06971
error at time 0.75 = 0.34321
error at time 0.80 = 0.63007
error at time 0.85 = 0.90919
error at time 0.90 = 1.15801
error at time 0.95 = 1.35419
error at time 1.00 = 1.47752
error at time 1.05 = 1.51172
error at time 1.10 = 1.44602
error at time 1.15 = 1.27643
error at time 1.20 = 1.00658
error at time 1.25 = 0.64792
error at time 1.30 = 0.21940
error at time 1.35 = 0.25352
error at time 1.40 = 0.74040
error at time 1.45 = 1.20783
error at time 1.50 = 1.62180
error at time 1.55 = 1.95024
error at time 1.60 = 2.16552
error at time 1.65 = 2.24679
error at time 1.70 = 2.18188
error at time 1.75 = 1.96864
error at time 1.80 = 1.61557
error at time 1.85 = 1.14167
error at time 1.90 = 0.57549
error at time 1.95 = 0.04657
error at time 2.00 = 0.68270
error at time 2.05 = 1.28856
error at time 2.10 = 1.82050
error at time 2.15 = 2.23883
error at time 2.20 = 2.51096
error at time 2.25 = 2.61408
```

```
error at time 2.30 = 2.53714
error at time 2.35 = 2.28214
error at time 2.40 = 1.86427
error at time 2.45 = 1.31122
error at time 2.50 = 0.66140
error at time 2.55 = 0.03868
error at time 2.60 = 0.73777
error at time 2.65 = 1.38370
error at time 2.70 = 1.92732
error at time 2.75 = 2.32645
error at time 2.80 = 2.54927
error at time 2.85 = 2.57709
error at time 2.90 = 2.40612
error at time 2.95 = 2.04810
error at time 3.00 = 1.52977
error at time 3.05 = 0.89109
error at time 3.10 = 0.18237
error at time 3.15 = 0.53954
error at time 3.20 = 1.21560
error at time 3.25 = 1.78940
error at time 3.30 = 2.21171
error at time 3.35 = 2.44477
error at time 3.40 = 2.46565
error at time 3.45 = 2.26859
error at time 3.50 = 1.86593
error at time 3.55 = 1.28769
error at time 3.60 = 0.57957
error at time 3.65 = 0.20029
error at time 3.70 = 0.98579
error at time 3.75 = 1.70809
error at time 3.80 = 2.30130
error at time 3.85 = 2.70809
error at time 3.90 = 2.88482
error at time 3.95 = 2.80579
error at time 4.00 = 2.46605
```

[872]:
```
time_index = []

for i in range(len(time_range)):
    time_index.append(i*time_step)
```

[873]:
```
# plot errors
plt.plot(time_index, forward_error_plot,"-" , label='forward error',␣
 ↪markersize=3)
plt.plot(time_index,backward_error_plot,"-" , label='backward error',␣
 ↪markersize=3)
```

```
plt.plot(time_index,central_error_plot,"-" , label='central error',␣
 ↪markersize=3)
plt.xlabel('index')
plt.ylabel('error (ft)')
plt.title('Error Between Numerical and Analytical')
plt.grid(color = 'black', which = 'major', linestyle = '-', linewidth = 0.5)
plt.grid(color = 'black', which = 'minor', linestyle = '--', linewidth = 0.05)
plt.legend(loc="upper left")
plt.xlim([0, end_time])
```

[873]: (0.0, 4.0)