Program-M4

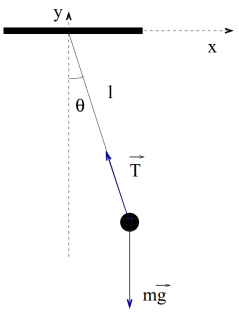
Devansh Shukla
I18PH021

Write and execute a FORTRAN program to simulate motion of a simple pendulum.

1 Theory

1.1 Motion of a Simple Pendulum

Assuming the string is massless and inextendable.
Choosing θ as the generalized coordinate:



$$\mathcal{L} = T - V$$
$$\mathcal{L} = \frac{1}{2}mr^2\dot{\theta}^2 - mgh$$
$$\mathcal{L} = \frac{1}{2}mr^2\dot{\theta}^2 - mgr(1 - \cos(\theta)) \tag{1}$$

Let the length of the string be l

$$\mathcal{L} = \frac{1}{2}ml^2\dot{\theta}^2 - mgl(1 - \cos(\theta)) \tag{2}$$

$$\text{Using Euler-Lagrange equation, } \frac{\partial}{\partial t}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = \frac{\partial \mathcal{L}}{\partial \theta}$$
$$ml^2\ddot{\theta} = mgl \sin\theta$$
$$\ddot{\theta} = \frac{g}{l} \sin\theta$$

$$\text{which has a solution of the form, } \theta = A\cos(\omega t);\ \omega = \sqrt{\frac{g}{l}}$$

$$\boxed{\begin{aligned} &\therefore \theta = A\cos(\omega t) \\ &T = 2\pi\frac{l}{g} \\ &\dot{\theta} = -A\omega \sin(\omega t) \\ &\ddot{\theta} = -A\omega^2 \cos(\omega t) \end{aligned}} \tag{3}$$

2 Numerical Solution

For length, $l = 2.0$; $\theta_0 = \pi/4 = 0.785$; the time-period will be:

$$\begin{aligned} \omega &= \sqrt{\frac{g}{l}} \\ \omega &= 2.2143\ s^{-1} \\ T &= \frac{2\pi}{\omega} \\ T &= 2.8375\ s \end{aligned} \tag{4}$$

3 Program Algorithm

NOTE: Blue-colored text represents variables in the algorithm, eg. variable.

1. Program open.

2. Define variables (PI , g , omega , theta0 , l , t0 , tf , dt , t , FMT , fmt1).
3. Define functions (theta(t) , dtheta(t) , ddtheta(t) , x(t) , y(t) , vx(t) , vy(t)).
4. Open a writable data file.
5. Get input from user for angular velocity (omega), length(l), initial position (theta0) and time period(t0 , tf , dt).
6. If the time-increment is less-than or equal to 0, terminate the program with message Illegal value of dt
7. Print parameters to stdout for the user.
8. Write appropiate comments in the data file and initialize other parameters.
9. Define a do while loop with index t which runs from t0 to tf .
10. Compute the parameters using the functions theta(t) , dtheta(t) , x(t) , y(t) , vx(t) , vy(t) .
11. Write the parameters to stdout and data file.
12. Increment the index according to t = t + dt
13. End do-while loop.
14. Close data file.
15. Program close.

4 Program

4.1 Fortran program:

For computing the parameters

```
!=============================================================
! Simple Pendulum
! Author: Devansh Shukla
!=============================================================
program simple_pendulum
    ! Program to compute motion of a simple pendulum.

    implicit none
    real*8, parameter :: PI=3.141593, g=9.80665 !m/s^2
    real*8 :: l=0.0, omega=0.0, theta0=PI/4.0, theta, dtheta, ddtheta
    real*8 :: t0=0.0, tf=0.0, dt=0.0, t=0.0
    real*8 :: x, y, vx, vy
    character(LEN=*), parameter :: FMT = "(F10.6, x, F10.6, x, F10.6, x, F10.6, x, F10.6, x, F10.6, x, F10.6, x, F10.6,
        x, F10.6)"
    character(len=*), parameter :: fmt1="(A10, A10, A10, xA10, xxxxA10, A10, A10, A10, xA10)"

    theta(t) = theta0*cos(omega*(t-t0))
    dtheta(t) = -omega * theta0 * sin(omega*(t-t0))
    ddtheta(t) = -omega * omega * theta0 * cos(omega*(t-t0))
    x(t) = l * sin(theta(t))
    y(t) = -l * cos(theta(t))
    vx(t) = l * dtheta(t) * cos(theta(t))
    vy(t) = l * dtheta(t) * sin(theta(t))

    open(UNIT=8, FILE="SimplePendulum.dat")

    print *, "Enter length (l)"
    read *, l
    omega = sqrt(g/l)

    print *, "Enter the value of theta0"
    read *, theta0

    print *, "Enter time: initial (t0), final (tf), increment (dt)"
    read *, t0, tf, dt
    if (dt .le. 0.0) stop "Illegal value of dt"

    print *, "-------------------------------------------------------"
    print "(x,A,F10.4,F10.4,F10.4)", "l, omega, T =", l, omega, 2.0*PI / omega
    print "(x,A,F10.4,F10.4,F10.4)", "theta0 =", theta0
    print "(x,A,F10.4,F10.4,F10.4)", "t0, tf, dt =", t0, tf, dt
    print *, "-------------------------------------------------------"
    print *, "Computing..."

    write (8, *) "# t0=", t0
    write (8, *) "# t l theta(t) dtheta(t) ddtheta(t) x(t) y(t) vx(t) vy(t)"

    print fmt1, "time", "length", "theta", "dtheta", "ddtheta(t)", "x(t)", "y(t)", "vx(t)", "vy(t)"
    t = t0
    do while (t <= tf)
        write (*, FMT) t, l, theta(t), dtheta(t), ddtheta(t), x(t), y(t), vx(t), vy(t)
```

```
        write (8, FMT) t, l, theta(t), dtheta(t), ddtheta(t), x(t), y(t), vx(t), vy(t)
        t = t + dt
    enddo
    print *, "-----------------------------------------------------"
    close(8)

end program simple_pendulum
```

4.2 Python program: Plots

```
#!/usr/bin/env python
"""
Author: Devansh Shukla
"""
# In[0]
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec

custom_rcparams = {
    "axes.labelsize": 7,
    "axes.titlesize": 8,
    "axes.grid": True,
    # Figure
    "figure.autolayout": True,
    "figure.titlesize": 9,
    "figure.figsize": (8, 8),
    "savefig.format": "pdf",
    "lines.linewidth": 1,
    # Legend
    "legend.fontsize": 8,
    "legend.frameon": True,
    # Ticks
    "xtick.labelsize": 6,
    "ytick.labelsize": 6,
    "xtick.minor.visible": True,
    "xtick.direction": "in",
    "ytick.direction": "in",
    "ytick.minor.visible": True,
    # TeX
    "pgf.texsystem": "lualatex",
}
mpl.rcParams.update(custom_rcparams)
mpl.use("pgf")
plt.ioff()

# Read data file
df = pd.read_csv("SimplePendulum.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")
print(df)

# Plot
gs = gridspec.GridSpec(3, 2)

fig = plt.figure()
ax = plt.subplot(gs[0, 0])
plt.plot(df[0], df[5], "o-", markersize=1.5, color="C0", label=r"$x(t)$")
plt.plot(df[0], df[6], "o-", markersize=1.5, color="C1", label=r"$y(t)$")
plt.title("Position")
ax.set_xlim(left=0)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$Position(m)$")
plt.legend(loc="upper right")

ax = plt.subplot(gs[0, 1])
plt.plot(df[0], df[7], "o-", markersize=1.5, label=r"$v_{x}(t)$")
plt.plot(df[0], df[8], "o-", markersize=1.5, label=r"$v_{y}(t)$")
plt.title("Velocity")
ax.set_xlim(left=0)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$Velocity(m/s)$")
plt.legend(loc="upper right")

ax = plt.subplot(gs[1, 0])
plt.plot(df[0], df[2], "o-", markersize=1.5, color="C0", label=r"$\theta(t)$")
plt.title(r"$\theta$ vs $t$")
ax.set_xlim(left=0)
```

```
ax.set_ylim(-1.0, 1.0)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$\theta(rad)$")
plt.legend(loc="upper right")

ax = plt.subplot(gs[1, 1])
plt.plot(df[0], df[3], "o-", markersize=1.5, color="C0", label=r"$\theta'(t)$")
plt.title(r"$\theta'$ vs $t$")
ax.set_xlim(left=0)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$\theta'(rad/s)$")
plt.legend(loc="upper right")

ax = plt.subplot(gs[2, 0])
plt.plot(df[0], df[4], "o-", markersize=1.5, color="C0", label=r"$\theta''(t)$")
plt.title(r"$\theta''$ vs $t$")
ax.set_xlim(left=0)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$\theta''(rad/s^2)$")
plt.legend(loc="upper right")

ax = plt.subplot(gs[2, 1])
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim(-2.5, 2.5)
ax.set_ylim(-2.5,0)
ax.text(0.05, 1.025, 'X', transform=ax.transAxes)
ax.text(-0.025, 0.5, 'Y', transform=ax.transAxes)
# ax.set_aspect("equal")
plt.plot(df[5], df[6], "o-", markersize=1.5, color="C1", label="trace")
plt.title("Trajectory trace")
plt.legend(loc="upper right")


plt.suptitle("Simple Pendulum")
plt.savefig("plots/simple_params.pdf")
# plt.show()

# %%
```

4.3 Python program: Animation

```
#!/usr/bin/env python
"""
Author: Devansh Shukla
"""
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation, FFMpegWriter

custom_rcparams = {
    "axes.labelsize": 6,
    "axes.titlesize": 8,
    "axes.grid": True,
    # Figure
    "figure.autolayout": True,
    "figure.titlesize": 9,
    "figure.figsize": (10, 3.5),
    # "figure.dpi": 150,
    "savefig.format": "pdf",
    "lines.linewidth": 1,
    # Legend
    "legend.fontsize": 8,
    "legend.frameon": True,
    # Ticks
    "xtick.labelsize": 8,
    "ytick.labelsize": 8,
    "xtick.minor.visible": True,
    "xtick.direction": "in",
    "ytick.direction": "in",
    "ytick.minor.visible": True,
}
mpl.rcParams.update(custom_rcparams)

df = pd.read_csv("SimplePendulum.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")
```

```
time = df[0].values
theta = df[2].values
pos_x = df[5].values
pos_y = df[6].values
vel_x = df[7].values
vel_y = df[8].values

gs = mpl.gridspec.GridSpec(1, 2, width_ratios=[1, 1])
fig = plt.figure()
ax1 = fig.add_subplot(gs[0, 0])
ax2 = fig.add_subplot(gs[0, 1])

line1, = ax1.plot([], [], 'o-', lw=2, label="Pendulum")
trace, = ax1.plot([], [], ',-', lw=1, label="trace")
_, = ax1.plot([], [], ',-', color="C4", lw=1, label="velocity vector")
time_template = "time = %.1fs"
time_text = ax1.text(0.05, 0.8, '', transform=ax1.transAxes)

patch = plt.Arrow(pos_x[0], pos_y[0], 0.1, 0.1, width=0.1, color="C4")
ax1.add_patch(patch)
ax1.legend()

line_theta, = ax2.plot([], [], '-', lw=2, label=r"$\theta(t)$")
ax2.legend()

line = [line1, line_theta,]

ax1.spines["top"].set_color("none")
ax1.spines["bottom"].set_position("zero")
ax1.spines["left"].set_position("zero")
ax1.spines["right"].set_color("none")
ax1.set_ylim(top=0.0, bottom=(min(pos_y)-1.0))
ax1.set_xlim(left=-(abs(max(pos_x))+1.0), right=(abs(max(pos_x))+1.0))
ax1.set_aspect("equal")
ax1.text(0.05, 1.025, "X", transform=ax1.transAxes)
ax1.text(-0.025, 0.5, "Y", transform=ax1.transAxes)

ax2.set_xlim(0, time[-1])
ax2.set_aspect(4)
ax2.set_ylabel(r"$\theta(rad)$")
ax2.set_xlabel("Time(s)")
ax2.set_ylim(-1.2, 1.2)
ax2.set_title(r"$\theta(t)$")

def init():
    line[0].set_data([], [])
    trace.set_data([], [])
    return line, trace

def animate(i):
    global time, pos_x, pos_y, vel_x, vel_y

    line[0].set_data([0, pos_x[i]], [0, pos_y[i]])
    trace.set_data(pos_x[:i], pos_y[:i])
    time_text.set_text(time_template % (time[i]))

    line[1].set_data(time[:i], theta[:i])
    global ax1, patch
    ax1.patches.remove(patch)
    patch = plt.Arrow(pos_x[i], pos_y[i], vel_x[i]/10, vel_y[i]/10, width=0.1, color="C4") # Velocity rescaled for arrow
          length
    ax1.add_patch(patch)

    global captures
    if time[i] in captures:
        toggle_capture()
    return line, trace, time_text

def toggle_capture(*args, **kwargs):
    global ani, capture_no
    ani.pause()
    plt.gcf().savefig(f"plots/simple_capture_{capture_no}.pdf")
    capture_no += 1
    ani.resume()

capture_no = 0
captures = np.arange(2.5, 18.0, 1.0)
captures = np.insert(captures, 0, [0.5, 1.0, 1.5])
```

```
ani = FuncAnimation(fig, animate, frames=len(time), interval=10, init_func=init, blit=False, repeat=False)
fig.canvas.mpl_connect('button_press_event', toggle_capture)
writer = FFMpegWriter(fps=10)
ani.save('animation.mp4', writer=writer)
plt.show()
```

5 Results

5.1 Terminal Output

```
 Enter length (l)
2.0
 Enter the value of theta0
0.785
 Enter time: initial (t0), final (tf), increment (dt)
0.0 20.0 0.1
 ------------------------------------------------------
 l, omega, T =    2.0000    2.2143    2.8375
 theta0 =    0.7850
 t0, tf, dt =    0.0000   20.0000    0.1000
 ------------------------------------------------------
 Computing...
      time     length      theta      dtheta    ddtheta(t)       x(t)       y(t)      vx(t)      vy(t)
  0.000000   2.000000   0.785000  -0.000000  -3.849110   1.413650  -1.414777  -0.000000  -0.000000
  0.100000   2.000000   0.765833  -0.381773  -3.755128   1.386275  -1.441611  -0.550368  -0.529243
  0.200000   2.000000   0.709268  -0.744903  -3.477771   1.302557  -1.517678  -1.130523  -0.970278
  0.300000   2.000000   0.618067  -1.071657  -3.030583   1.158921  -1.630000  -1.746801  -1.241966
  0.400000   2.000000   0.496684  -1.346078  -2.435402   0.953025  -1.758335  -2.366857  -1.282847
  0.500000   2.000000   0.351046  -1.554767  -1.721292   0.687760  -1.878027  -2.919894  -1.069307
  0.600000   2.000000   0.188265  -1.687530  -0.923126   0.374310  -1.964661  -3.315425  -0.631660
  0.700000   2.000000   0.016291  -1.737887  -0.079881   0.032581  -1.999735  -3.475312  -0.056622
  0.800000   2.000000  -0.156478  -1.703376   0.767265  -0.311681  -1.975564  -3.365130   0.530911
  0.900000   2.000000  -0.321607  -1.585685   1.576943  -0.632183  -1.897457  -3.008769   1.002443
  1.000000   2.000000  -0.471030  -1.390559   2.309613  -0.907609  -1.782203  -2.478258   1.262084
  1.100000   2.000000  -0.597451  -1.127528   2.929498  -1.125074  -1.653544  -1.864417   1.268552
  1.200000   2.000000  -0.694697  -0.809436   3.406326  -1.280306  -1.536495  -1.243694   1.036325
  1.300000   2.000000  -0.758019  -0.451816   3.716812  -1.374968  -1.452399  -0.656217   0.621233
  1.400000   2.000000  -0.784324  -0.072133   3.845795  -1.412693  -1.415732  -0.102121   0.101902
  1.500000   2.000000  -0.772328   0.311072   3.786974  -1.395609  -1.432577   0.445635  -0.434135
  1.600000   2.000000  -0.722617   0.679087   3.543224  -1.322699  -1.500156   1.018737  -0.898228
  1.700000   2.000000  -0.637618   1.013940   3.126446  -1.190566  -1.607033   1.629435  -1.207162
  1.800000   2.000000  -0.521482   1.299279   2.556994  -0.996331  -1.734164   2.253163  -1.294512
  1.900000   2.000000  -0.379880   1.521170   1.862676  -0.741618  -1.857418   2.825449  -1.128128
  2.000000   2.000000  -0.219728   1.668777   1.077397  -0.435928  -1.951914   3.257309  -0.727467
  2.100000   2.000000  -0.048846   1.734893   0.239506  -0.097652  -1.997615   3.465647  -0.169416
  2.200000   2.000000   0.124422   1.716288  -0.610082   0.248203  -1.984539   3.406040   0.425987
  2.300000   2.000000   0.291614   1.613871  -1.429877   0.574996  -1.915562   3.091471   0.927970
  2.400000   2.000000   0.444565   1.432644  -2.179846   0.860130  -1.805596   2.586775   1.232260
  2.500000   2.000000   0.575807   1.181456  -2.823367   1.089023  -1.677507   1.981900   1.286633
  2.600000   2.000000   0.678930   0.872574  -3.329014   1.255921  -1.556490   1.358153   1.095884
  2.700000   2.000000   0.748899   0.521081  -3.672094   1.361665  -1.464878   0.763320   0.709538
  2.800000   2.000000   0.782296   0.144142  -3.835854   1.409820  -1.418593   0.204479   0.203214
  2.900000   2.000000   0.777492  -0.239836  -3.812296   1.402989  -1.425350  -0.341850  -0.336487
  3.000000   2.000000   0.734720  -0.612101  -3.602573   1.340759  -1.484037  -0.908381  -0.820681
  3.100000   2.000000   0.656070  -0.954476  -3.216923   1.220015  -1.584792  -1.512646  -1.164475
  3.200000   2.000000   0.545381  -1.250241  -2.674182   1.037488  -1.709859  -2.137736  -1.297110
  3.300000   2.000000   0.408060  -1.484953  -2.000851   0.793659  -1.835785  -2.726053  -1.178546
  3.400000   2.000000   0.250812  -1.647149  -1.229812   0.496381  -1.937422  -3.191224  -0.817613
  3.500000   2.000000   0.081316  -1.728910  -0.398718   0.162452  -1.993391  -3.446394  -0.280865
  3.600000   2.000000  -0.092151  -1.726243   0.451848  -0.184042  -1.991514  -3.437836   0.317701
  3.700000   2.000000  -0.261118  -1.639277   1.280347  -0.516322  -1.932204  -3.167417   0.846395
  3.800000   2.000000  -0.417334  -1.472261   2.046324  -0.810649  -1.828346  -2.691801   1.193487
  3.900000   2.000000  -0.553170  -1.233349   2.712372  -1.050774  -1.701727  -2.098823   1.295971
  4.000000   2.000000  -0.661993  -0.934209   3.245966  -1.229380  -1.577538  -1.473749   1.148497
  4.100000   2.000000  -0.738488  -0.589448   3.621049  -1.346342  -1.478974  -0.871778   0.793598
  4.200000   2.000000  -0.778921  -0.215903   3.819304  -1.405024  -1.423343  -0.307304   0.303349
  4.300000   2.000000  -0.781317   0.168186   3.831051  -1.408430  -1.419973   0.238819  -0.236878
  4.400000   2.000000  -0.745558   0.544061   3.655715  -1.356764  -1.469418   0.799454  -0.738163
  4.500000   2.000000  -0.673392   0.893368   3.301859  -1.247282  -1.563422   1.396712  -1.114282
  4.600000   2.000000  -0.568341   1.199049   2.786762  -1.076470  -1.685590   2.021106  -1.290740
  4.700000   2.000000  -0.435537   1.446177   2.135579  -0.843794  -1.813287   2.622335  -1.220276
  4.800000   2.000000  -0.281464   1.622683   1.380108  -0.555524  -1.921300   3.117661  -0.901440
  4.900000   2.000000  -0.113646   1.719949   0.557243  -0.226803  -1.987099   3.417707  -0.390089
  5.000000   2.000000   0.059722   1.733223  -0.292835   0.119372  -1.996434   3.460267   0.206899
  5.100000   2.000000   0.230173   1.661859  -1.128612   0.456292  -1.947254   3.236062   0.758292
  5.200000   2.000000   0.389384   1.509341  -1.909276   0.759237  -1.850286   2.792713   1.145948
  5.300000   2.000000   0.529580   1.283117  -2.596704   1.010342  -1.726038   2.214709   1.296387
  5.400000   2.000000   0.643915   0.994234  -3.157326   1.200663  -1.599503   1.590280   1.193739
  5.500000   2.000000   0.726806   0.656799  -3.563766   1.328972  -1.494601   0.981653   0.872868
```

5.600000	2.000000	0.774204	0.287291	-3.796175	1.398295	-1.429955	0.410814	0.401718
5.700000	2.000000	0.783796	-0.096246	-3.843205	1.411946	-1.416478	-0.136331	-0.135894
5.800000	2.000000	0.755112	-0.475084	-3.702559	1.370740	-1.456390	-0.691907	-0.651216
5.900000	2.000000	0.689554	-0.830721	-3.381106	1.272386	-1.543060	-1.281853	-1.056998
6.000000	2.000000	0.590322	-1.145792	-2.894541	1.113257	-1.661523	-1.903760	-1.275562
6.100000	2.000000	0.462263	-1.404910	-2.266628	0.891950	-1.790091	-2.514917	-1.253110
6.200000	2.000000	0.311631	-1.595422	-1.528027	0.613222	-1.903670	-3.037156	-0.978349
6.300000	2.000000	0.145780	-1.708024	-0.714808	0.290529	-1.978786	-3.379814	-0.496230
6.400000	2.000000	-0.027189	-1.737218	0.133318	-0.054372	-1.999261	-3.473152	0.094456
6.500000	2.000000	-0.198831	-1.681578	0.974933	-0.395047	-1.960596	-3.296896	0.664302
6.600000	2.000000	-0.360763	-1.543821	1.768939	-0.705977	-1.871255	-2.888884	1.089902
6.700000	2.000000	-0.505078	-1.330674	2.476562	-0.967752	-1.750273	-2.329044	1.287762
6.800000	2.000000	-0.624728	-1.052547	3.063247	-1.169754	-1.622244	-1.707487	1.231221
6.900000	2.000000	-0.713871	-0.723019	3.500343	-1.309529	-1.511666	-1.092964	0.946815
7.000000	2.000000	-0.768153	-0.358185	3.766506	-1.389617	-1.438390	-0.515210	0.497740
7.100000	2.000000	-0.784924	0.024141	3.848739	-1.413543	-1.414884	0.034156	-0.034124
7.200000	2.000000	-0.763365	0.405288	3.743025	-1.382713	-1.445028	0.585652	-0.560396
7.300000	2.000000	-0.704528	0.766643	3.454528	-1.295348	-1.523835	1.168238	-0.993069
7.400000	2.000000	-0.611286	1.090561	2.997334	-1.147842	-1.637821	1.786144	-1.251792
7.500000	2.000000	-0.488193	1.361223	2.393771	-0.938062	-1.766363	2.404414	-1.276912
7.600000	2.000000	-0.341261	1.565412	1.673313	-0.669351	-1.884667	2.950280	-1.047810
7.700000	2.000000	-0.177663	1.693157	0.871141	-0.353460	-1.968519	3.333012	-0.598464
7.800000	2.000000	-0.005390	1.738220	0.026429	-0.010780	-1.999971	3.476390	-0.018738
7.900000	2.000000	0.167147	1.698400	-0.819574	0.332739	-1.972127	3.349460	0.565123
8.000000	2.000000	0.331521	1.575642	-1.625555	0.650963	-1.891097	2.979691	1.025684
8.100000	2.000000	0.479706	1.375939	-2.352154	0.923037	-1.774261	2.441276	1.270042
8.200000	2.000000	0.604465	1.109046	-2.963890	1.136645	-1.645612	1.825059	1.260591
8.300000	2.000000	0.699707	0.787994	-3.430890	1.287987	-1.530062	1.205679	1.014926
8.400000	2.000000	0.760779	0.428462	-3.730348	1.378972	-1.448598	0.620669	0.590837
8.500000	2.000000	0.784701	0.048006	-3.847642	1.413227	-1.415200	0.067938	0.067844
8.600000	2.000000	0.770302	-0.334793	-3.777043	1.392705	-1.435400	-0.480563	-0.466268
8.700000	2.000000	0.718288	-0.701244	-3.521998	1.316193	-1.505867	-1.055980	-0.922972
8.800000	2.000000	0.631197	-1.033451	-3.094963	1.180223	-1.614644	-1.668655	-1.219702
8.900000	2.000000	0.513283	-1.315191	-2.516791	0.982079	-1.742275	-2.291423	-1.291621
9.000000	2.000000	0.370303	-1.532705	-1.815716	0.723796	-1.864435	-2.857630	-1.109366
9.100000	2.000000	0.209240	-1.675373	-1.025974	0.415434	-1.956378	-3.277664	-0.696007
9.200000	2.000000	0.037960	-1.736228	-0.186130	0.075902	-1.998559	-3.469954	-0.131783
9.300000	2.000000	-0.135174	-1.712296	0.662803	-0.269526	-1.981756	-3.393352	0.461508
9.400000	2.000000	-0.301707	-1.604748	1.479370	-0.594302	-1.909661	-3.064524	0.953704
9.500000	2.000000	-0.453507	-1.418834	2.223694	-0.876242	-1.797832	-2.550825	1.243242
9.600000	2.000000	-0.583161	-1.163634	2.859428	-1.101330	-1.669452	-1.942632	1.281546
9.700000	2.000000	-0.684337	-0.851611	3.355526	-1.264319	-1.549677	-1.319721	1.076707
9.800000	2.000000	-0.752095	-0.498000	3.687764	-1.366340	-1.460519	-0.727339	0.680437
9.900000	2.000000	-0.783125	-0.120071	3.839916	-1.410995	-1.417425	-0.170191	0.169419
10.000000	2.000000	-0.775913	0.263722	3.804553	-1.400736	-1.427564	0.376481	-0.369405
10.100000	2.000000	-0.730811	0.634637	3.583401	-1.334947	-1.489267	0.945144	-0.847207
10.200000	2.000000	-0.650020	0.974560	3.187261	-1.210405	-1.592143	1.551639	-1.179613
10.300000	2.000000	-0.537487	1.266892	2.635476	-1.023959	-1.717996	2.176516	-1.297245
10.400000	2.000000	-0.398707	1.497358	1.954992	-0.776455	-1.843127	2.759822	-1.162631
10.500000	2.000000	-0.240457	1.654703	1.179039	-0.476293	-1.942458	3.214192	-0.788124
10.600000	2.000000	-0.070465	1.731244	0.345511	-0.140813	-1.995037	3.453895	-0.243781
10.700000	2.000000	0.102969	1.723242	-0.504890	0.205574	-1.989407	3.428230	0.354254
10.800000	2.000000	0.271374	1.631089	-1.330636	0.536111	-1.926807	3.142793	0.874445
10.900000	2.000000	0.426527	1.459284	-2.091402	0.827424	-1.820816	2.657088	1.207446
11.000000	2.000000	0.560852	1.216218	-2.750039	1.063815	-1.693605	2.059793	1.293832
11.100000	2.000000	0.667788	0.913760	-3.274382	1.238501	-1.570387	1.434957	1.131694
11.200000	2.000000	0.742114	0.566681	-3.638827	1.351695	-1.474083	0.835334	0.765980
11.300000	2.000000	0.780200	0.191928	-3.825576	1.406844	-1.421545	0.272834	0.270013
11.400000	2.000000	0.780187	-0.192197	-3.825509	1.406824	-1.421564	-0.273220	-0.270388
11.500000	2.000000	0.742074	-0.566937	-3.638631	1.351636	-1.474137	-0.835742	-0.766292
11.600000	2.000000	0.667724	-0.913991	-3.274067	1.238401	-1.570466	-1.435392	-1.131887
11.700000	2.000000	0.560766	-1.216412	-2.749619	1.063671	-1.693696	-2.060231	-1.293861
11.800000	2.000000	0.426425	-1.459432	-2.090899	0.827237	-1.820901	-2.657480	-1.207295
11.900000	2.000000	0.271259	-1.631183	-1.330074	0.535890	-1.926868	-3.143074	-0.874135
12.000000	2.000000	0.102848	-1.723278	-0.504296	0.205333	-1.989432	-3.428343	-0.353846
12.100000	2.000000	-0.070586	-1.731220	0.346108	-0.141055	-1.995020	-3.453817	0.244198
12.200000	2.000000	-0.240573	-1.654620	1.179610	-0.476519	-1.942403	-3.213940	0.788458
12.300000	2.000000	-0.398813	-1.497221	1.955508	-0.776649	-1.843045	-2.759446	1.162815
12.400000	2.000000	-0.537577	-1.266707	2.635912	-1.024112	-1.717904	-2.176082	1.297250
12.500000	2.000000	-0.650089	-0.974336	3.187597	-1.210514	-1.592060	-1.551202	1.179448
12.600000	2.000000	-0.730855	-0.634385	3.583620	-1.335013	-1.489208	-0.944731	0.846912
12.700000	2.000000	-0.775931	-0.263455	3.804644	-1.400762	-1.427538	-0.376092	0.369038
12.800000	2.000000	-0.783117	0.120341	3.839875	-1.410983	-1.417437	0.170575	-0.169799
12.900000	2.000000	-0.752060	0.498259	3.687592	-1.366289	-1.460567	0.727741	-0.680766
13.000000	2.000000	-0.684277	0.851847	3.355233	-1.264226	-1.549752	1.320152	-1.076927
13.100000	2.000000	-0.583079	1.163836	2.859026	-1.101194	-1.669543	1.943073	-1.281608
13.200000	2.000000	-0.453407	1.418990	2.223204	-0.876063	-1.797919	2.551231	-1.243124
13.300000	2.000000	-0.301595	1.604852	1.478816	-0.594086	-1.909728	3.064830	-0.953420
13.400000	2.000000	-0.135054	1.712343	0.662213	-0.269287	-1.981788	3.393500	-0.461112

```
13.500000   2.000000   0.038082   1.736214  -0.186729   0.076146  -1.998550   3.469911   0.132205
13.600000   2.000000   0.209358   1.675301  -1.026552   0.415664  -1.956329   3.277440   0.696363
13.700000   2.000000   0.370411   1.532578  -1.816245   0.723997  -1.864357   2.857273   1.109582
13.800000   2.000000   0.513375   1.315014  -2.517245   0.982240  -1.742184   2.290995   1.291659
13.900000   2.000000   0.631270   1.033233  -3.095320   1.180340  -1.614558   1.668214   1.219566
14.000000   2.000000   0.718337   0.700996  -3.522240   1.316267  -1.505802   1.055562   0.922698
14.100000   2.000000   0.770326   0.334528  -3.777158   1.392738  -1.435368   0.480170   0.465910
14.200000   2.000000   0.784697  -0.048277  -3.847625   1.413222  -1.415205  -0.068322  -0.068226
14.300000   2.000000   0.760749  -0.428724  -3.730201   1.378929  -1.448639  -0.621066  -0.591180
14.400000   2.000000   0.699651  -0.788235  -3.430618   1.287902  -1.530133  -1.206105  -1.015170
14.500000   2.000000   0.604387  -1.109254  -2.963508   1.136516  -1.645701  -1.825501  -1.260685
14.600000   2.000000   0.479609  -1.376105  -2.351680   0.922865  -1.774351  -2.441693  -1.269959
14.700000   2.000000   0.331410  -1.575756  -1.625011   0.650753  -1.891169  -2.980021  -1.025428
14.800000   2.000000   0.167027  -1.698458  -0.818988   0.332503  -1.972167  -3.349642  -0.564743
14.900000   2.000000  -0.005512  -1.738218   0.027028  -0.011024  -1.999970  -3.476384   0.019163
15.000000   2.000000  -0.177782  -1.693096   0.871725  -0.353695  -1.968477  -3.332820   0.598839
15.100000   2.000000  -0.341371  -1.565294   1.673853  -0.669558  -1.884593  -2.949943   1.048056
15.200000   2.000000  -0.488289  -1.361055   2.394241  -0.938231  -1.766273  -2.403995   1.276984
15.300000   2.000000  -0.611363  -1.090350   2.997710  -1.147968  -1.637733  -1.785702   1.251687
15.400000   2.000000  -0.704581  -0.766400   3.454792  -1.295430  -1.523765  -1.167814   0.992818
15.500000   2.000000  -0.763393  -0.405024   3.743165  -1.382754  -1.444988  -0.585256   0.560049
15.600000   2.000000  -0.784926  -0.023870   3.848747  -1.413546  -1.414881  -0.033773   0.033741
15.700000   2.000000  -0.768128   0.358450   3.766383  -1.389581  -1.438425   0.515603  -0.498095
15.800000   2.000000  -0.713820   0.723266   3.500094  -1.309453  -1.511732   1.093384  -0.947082
15.900000   2.000000  -0.624654   1.052762   3.062884  -1.169634  -1.622331   1.707928  -1.231346
16.000000   2.000000  -0.504985   1.330848   2.476103  -0.967588  -1.750364   2.329469  -1.287713
16.100000   2.000000  -0.360655   1.543945   1.768407  -0.705774  -1.871332   2.889235  -1.089676
16.200000   2.000000  -0.198713   1.681647   0.974353  -0.394815  -1.960643   3.297109  -0.663940
16.300000   2.000000  -0.027067   1.737227   0.132719  -0.054128  -1.999267   3.473182  -0.094032
16.400000   2.000000   0.145900   1.707974  -0.715397   0.290766  -1.978751   3.379655   0.496622
16.500000   2.000000   0.311743   1.595315  -1.528577   0.613436  -1.903601   3.036842   0.978623
16.600000   2.000000   0.462362   1.404751  -2.267112   0.892127  -1.790003   2.514508   1.253216
16.700000   2.000000   0.590403   1.145589  -2.894937   1.113391  -1.661433   1.903319   1.275488
16.800000   2.000000   0.689612   0.830484  -3.381392   1.272476  -1.542986   1.281424   1.056770
16.900000   2.000000   0.755145   0.474823  -3.702723   1.370789  -1.456344   0.691506   0.650883
17.000000   2.000000   0.783803   0.095976  -3.843239   1.411955  -1.416468   0.135947   0.135514
17.100000   2.000000   0.774184  -0.287558  -3.796076   1.398266  -1.429983  -0.411204  -0.402083
17.200000   2.000000   0.726760  -0.657050  -3.563539   1.328903  -1.494663  -0.982068  -0.873156
17.300000   2.000000   0.643845  -0.994456  -3.156983   1.200551  -1.599587  -1.590719  -1.193895
17.400000   2.000000   0.529490  -1.283299  -2.596261   1.010186  -1.726130  -2.215141  -1.296372
17.500000   2.000000   0.389278  -1.509475  -1.908756   0.759041  -1.850367  -2.793083  -1.145753
17.600000   2.000000   0.230056  -1.661938  -1.128039   0.456064  -1.947307  -3.236305  -0.757950
17.700000   2.000000   0.059600  -1.733244  -0.292237   0.119129  -1.996449  -3.460333  -0.206480
17.800000   2.000000  -0.113767  -1.719909   0.557836  -0.227043  -1.987071  -3.417582   0.390494
17.900000   2.000000  -0.281578  -1.622586   1.380668  -0.555743  -1.921236  -3.117372   0.901742
18.000000   2.000000  -0.435639  -1.446027   2.136077  -0.843979  -1.813202  -2.621938   1.220416
18.100000   2.000000  -0.568426  -1.198853   2.787175  -1.076612  -1.685499  -2.020666   1.290700
18.200000   2.000000  -0.673455  -0.893136   3.302167  -1.247380  -1.563344  -1.396279   1.114080
18.300000   2.000000  -0.745597  -0.543804   3.655903  -1.356821  -1.469367  -0.799048   0.737845
18.400000   2.000000  -0.781329  -0.167916   3.831109  -1.408447  -1.419957  -0.238434   0.236501
18.500000   2.000000  -0.778906   0.216171   3.819230  -1.405003  -1.423365   0.307691  -0.303721
18.600000   2.000000  -0.738447   0.589703   3.620846  -1.346280  -1.479030   0.872188  -0.793905
18.700000   2.000000  -0.661927   0.934437   3.245644  -1.229276  -1.577618   1.474185  -1.148681
18.800000   2.000000  -0.553083   1.233539   2.711946  -1.050626  -1.701818   2.099260  -1.295989
18.900000   2.000000  -0.417230   1.472404   2.045816  -0.810460  -1.828430   2.692188  -1.193325
19.000000   2.000000  -0.261003   1.639367   1.279782  -0.516099  -1.932263   3.167689  -0.846076
19.100000   2.000000  -0.092030   1.726274   0.451252  -0.183800  -1.991536   3.437938  -0.317289
19.200000   2.000000   0.081437   1.728882  -0.399314   0.162695  -1.993372   3.446304   0.281280
19.300000   2.000000   0.250928   1.647063  -1.230380   0.496605  -1.937365   3.190961   0.817940
19.400000   2.000000   0.408164   1.484812  -2.001363   0.793850  -1.835702   2.725672   1.178719
19.500000   2.000000   0.545469   1.250053  -2.674613   1.037638  -1.709768   2.137301   1.297103
19.600000   2.000000   0.656137   0.954250  -3.217253   1.220121  -1.584710   1.512210   1.164301
19.700000   2.000000   0.734763   0.611848  -3.602784   1.340823  -1.483979   0.907970   0.820380
19.800000   2.000000   0.777509   0.239568  -3.812379   1.403013  -1.425326   0.341462   0.336116
19.900000   2.000000   0.782286  -0.144412  -3.835804   1.409806  -1.418608  -0.204864  -0.203593
-----------------------------------------------------
```

5.2 Plots

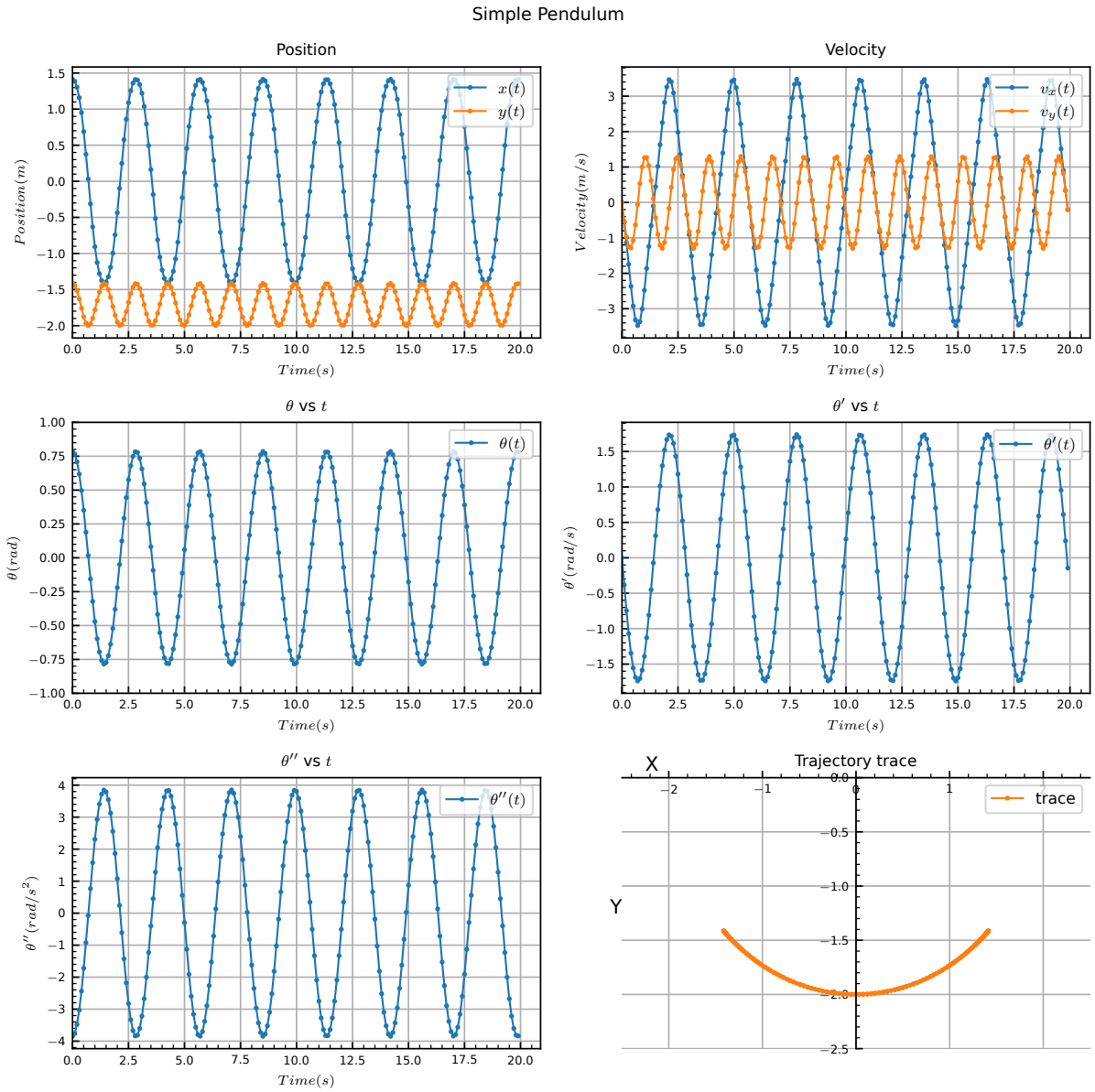
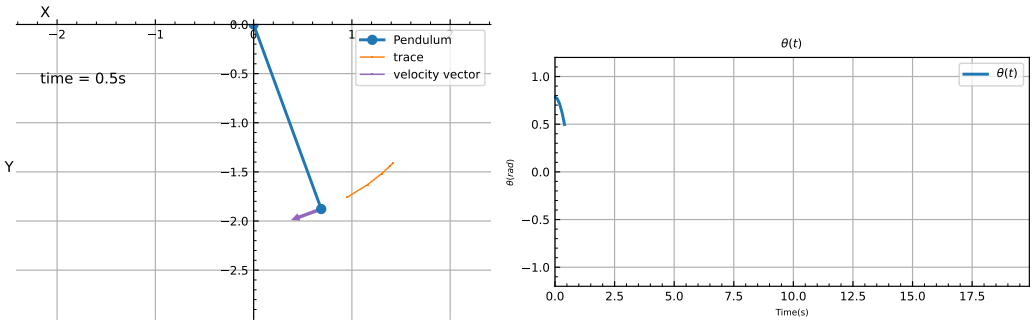


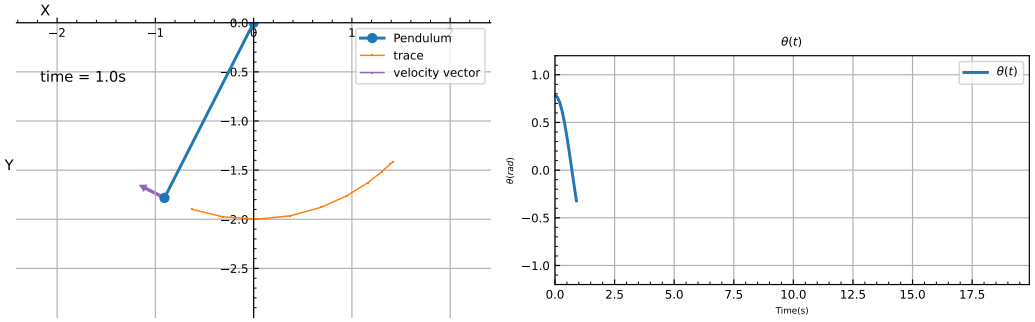
Figure 1: Simple Pendulum

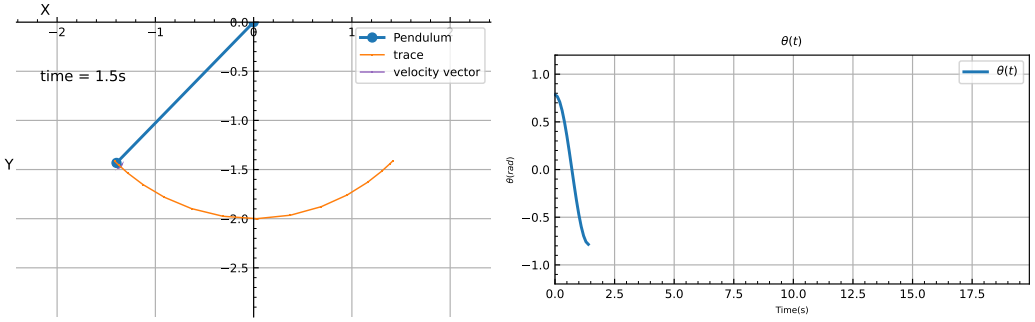
5.3 Animation

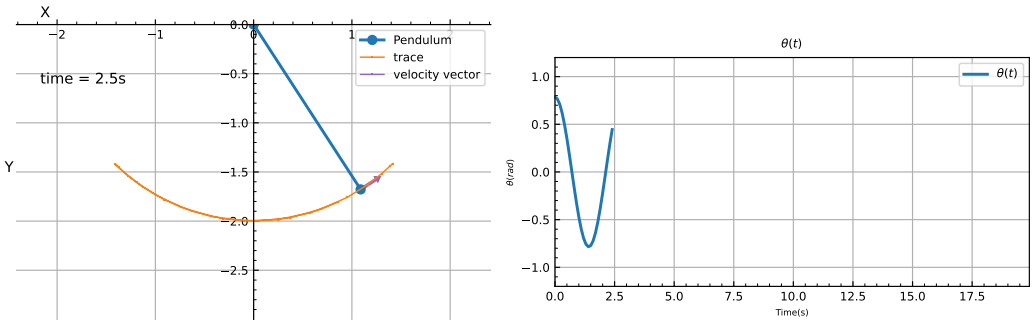
Note: Input parameters,

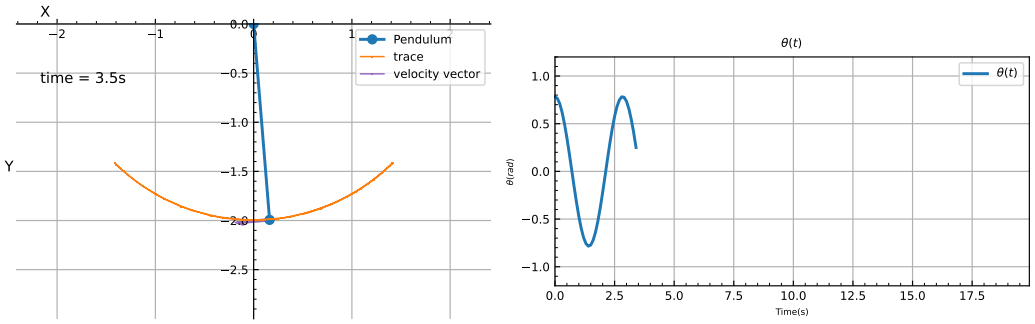
$$\begin{aligned} \theta_0 &= \frac{\pi}{4} = 0.785 \\ \text{length, } l &= 2\ m \\ \implies T &= 2.837\ s \end{aligned} \tag{5}$$

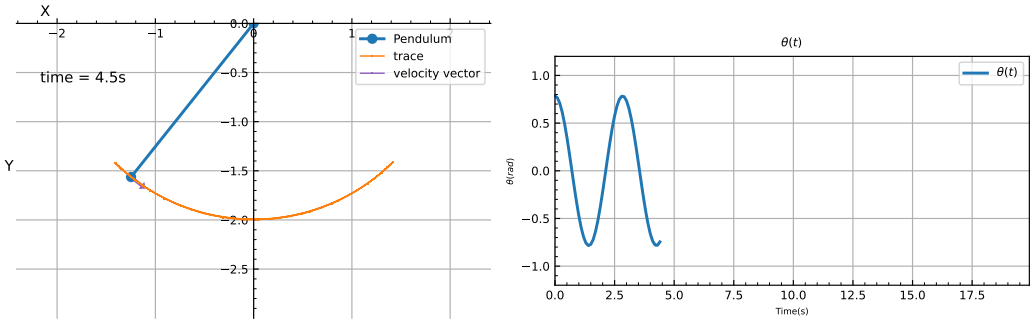


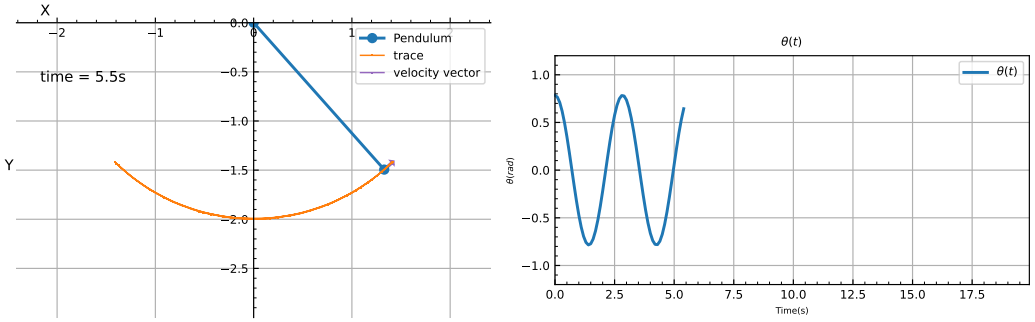

X
Y
time = 1.0s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)

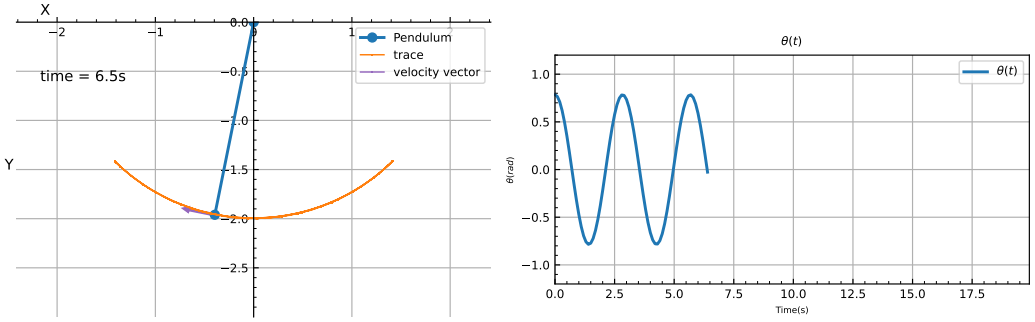

X
Y
time = 1.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)

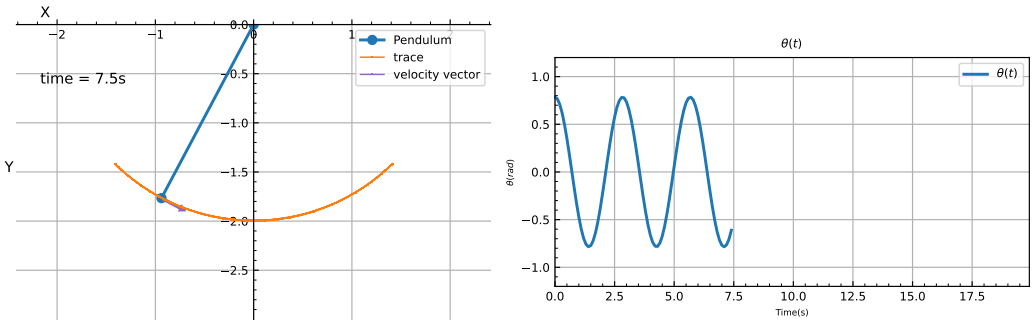

X
Y
time = 2.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)

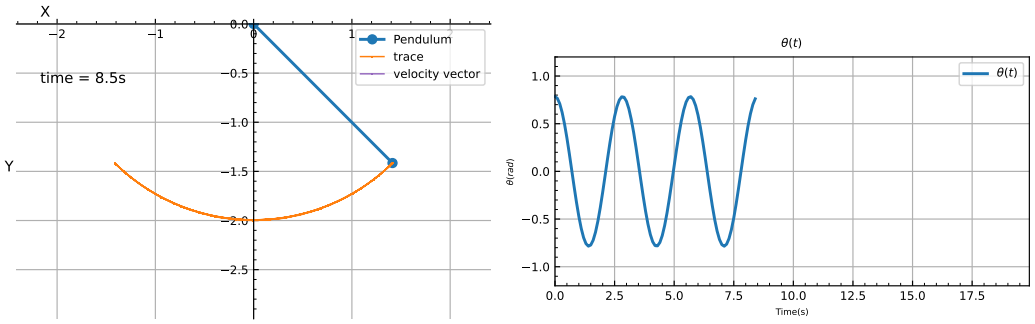

X
Y
time = 3.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)

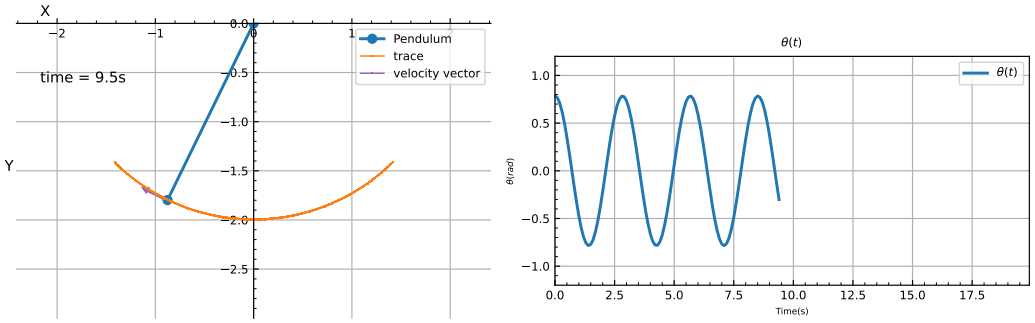

X
Y
time = 4.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)

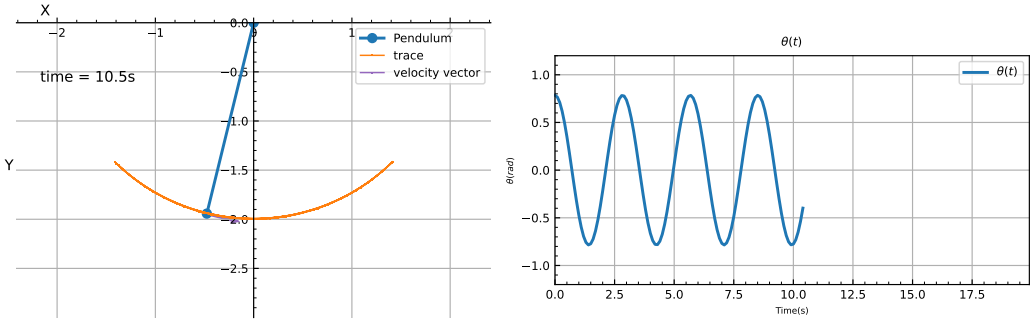

X
Y
time = 5.5s
Pendulum
trace
velocity vector
θ(t)
θ(t)
θ(rad)
Time(s)

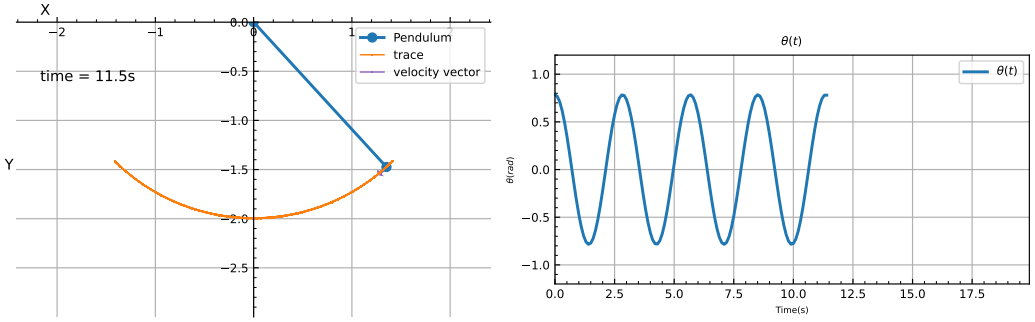

X
Y
time = 6.5s
Pendulum
trace
velocity vector
θ(t)
θ(t)
θ(rad)
Time(s)

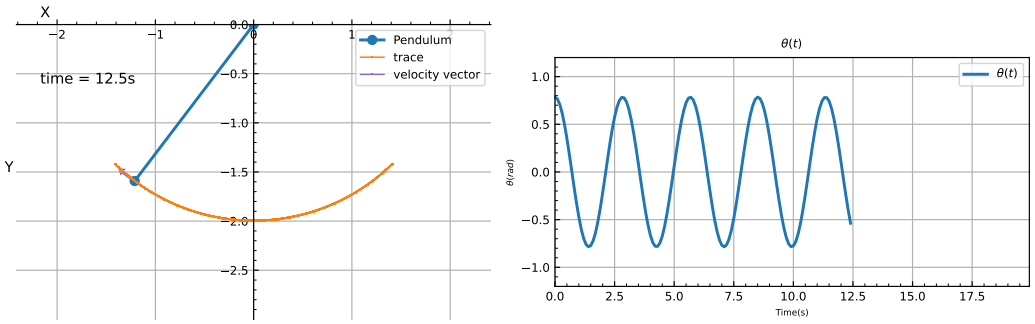

X
Y
time = 7.5s
Pendulum
trace
velocity vector
θ(t)
θ(t)
θ(rad)
Time(s)

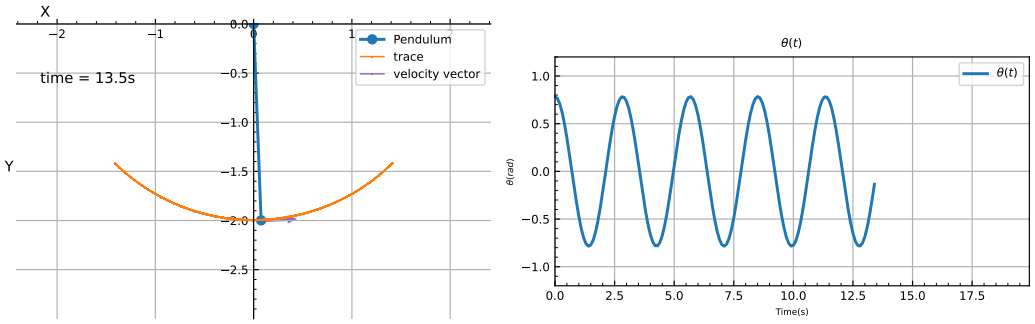

X
Y
time = 8.5s
Pendulum
trace
velocity vector
θ(t)
θ(t)
θ(rad)
Time(s)

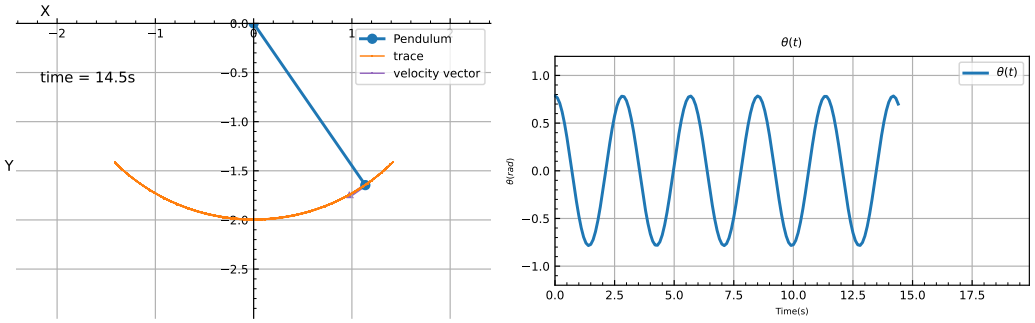

X
Y
time = 9.5s
Pendulum
trace
velocity vector
θ(t)
θ(t)
θ(rad)
Time(s)


X
Y
time = 10.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)


X
Y
time = 11.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)


X
Y
time = 12.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)


X
Y
time = 13.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)


X
Y
time = 14.5s
Pendulum
trace
velocity vector
θ(t)
θ(rad)
Time(s)

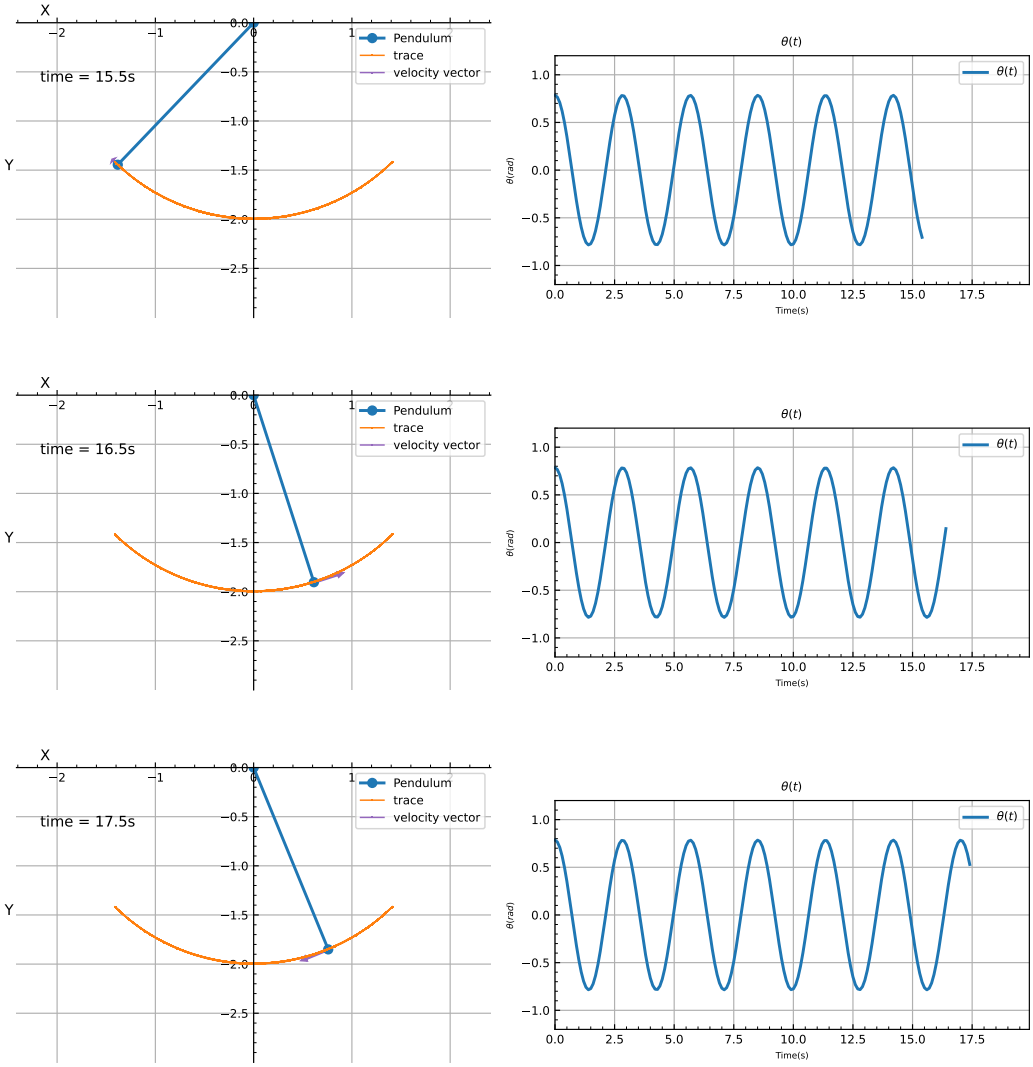


Figure 2: Animation of simple pendulum

6 Remarks

The programs can be used to numerically trace and simulate the motion of any simple pendulum or any simple harmonic motion, provided the required parameters are defined.

The parameters computed numerically and via the programs are in agreement.

With some modifications, programs can be used to simulate motion of forced damped and damped harmonic motion.