Program-M3

Devansh Shukla
I18PH021

Write and execute a FORTRAN program to simulate motion of a particle in uniform circular motion in spherical coordinates.

1 Theory

1.1 Motion of a particle in uniform circular motion

Degrees of freedom, $dof = 1$

Choosing θ as the generalized coordinate.

$$\text{Lagrangian, } \mathcal{L} = T - V$$
$$\mathcal{L} = \frac{1}{2}mr^2\dot{\theta}^2 \tag{1}$$

Using Euler-Lagrange equation

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial}{\partial t}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} \tag{2}$$

$$\boxed{\ddot{\theta} = 0} \tag{3}$$

Therefore, Equation of motion is $\ddot{\theta} = 0$ or $\dot{\theta} = \omega =$ Constant

$$\boxed{\begin{aligned} &\theta = \omega t \\ &\dot{\theta} = \omega = \text{constant} \\ &\ddot{\theta} = \alpha = 0 \end{aligned}} \tag{4}$$

2 Numerical Solution

For $\omega = 0.5$, $R = 1.0$, time period, T will be:

$$\begin{aligned} T &= \frac{2\pi}{\omega} \\ T &= 12.5664\ s \end{aligned} \tag{5}$$

3 Program Algorithm

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

1. Program open.
2. Define variables (PI, theta0, R, omega, t0, tf, dt, t, fmt1).
3. Define functions (theta(t), x(t), y(t)).
4. Open a writable data file.
5. Get input from user for angular velocity (omega), radius of the circle(R), 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), x(t), y(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

```
!===============================================================
! Uniform circular motion in polar coordinates
! Author: Devansh Shukla
!---------------------------------------------------------------
program CircularMotion_Polar
    ! Program to compute motion of a particle moving in a circle with uniform velocity in polar coordinates.

    implicit none
    real*8, parameter :: PI = 3.141593
    real*8 :: R=0.0, omega=0.0, theta0=0.0, theta, x, y
    real*8 :: t0=0.0, tf=0.0, dt=0.0, t=0.0
    character(len=*), parameter :: fmt1 = "(F10.6, x, F10.6, x, F10.6, x, F10.6, x, F10.6)"

    theta(t) = theta0 + omega*(t-t0)
    x(t) = R*cos(theta(t))
    y(t) = R*sin(theta(t))

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

    ! Input
    print *, "Enter angular velocity(omega) and radius (R)"
    read *, omega, R
    if (omega .le. 0.0) stop "Illegal value of omega"
    if (R .le. 0.0) stop "Illegal value of R"

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

    print *, "Enter t0, tf, dt"
    read *, t0, tf, dt
    if (dt .le. 0.0) stop "Illegal value of dt"

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

    write (8, *) "# t0=", t0
    write (8, *) "# t R theta x y"
    print "(xA10,A10,xA10,A10,A10)", "time", "R", "theta(t)", "x(t)", "y(t)"
    ! Computing
    t = t0
    do while (t <= tf)
        write (*, fmt1) t, R, theta(t), x(t), y(t)
        write (8, fmt1) t, R, theta(t), x(t), y(t)
        t = t + dt
    enddo
    print *, "------------------------------------------------------------"
    close(8)

end program CircularMotion_Polar
```

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,
```

```python
    # Figure
    "figure.autolayout": True,
    "figure.titlesize": 9,
    "figure.dpi": 150,
    "savefig.format": "pdf",
    "lines.linewidth": 1,
    # Legend
    "legend.fontsize": 6,
    "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()

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

# Assuming t0=0, x_at_0=x0+ Rcos(wt0)=1
x0 = df[3].iloc[0] - 1
y0 = df[4].iloc[0]
theta = np.arctan((df[4].values - y0)/(df[3].values - x0))

gs = gridspec.GridSpec(2, 1)

fig = plt.figure()
ax = plt.subplot(gs[0, 0])
plt.plot(df[0], theta, "o-", markersize=2, color="C0", label=r"$\theta(t)$")
plt.hlines(y=np.pi/2, xmin=df[0].values[0], xmax=df[0].values[-1], color="red", label=r"$\pi/2$")
plt.hlines(y=-np.pi/2, xmin=df[0].values[0], xmax=df[0].values[-1], color="red", label=r"$-\pi/2$")
plt.xlim(left=0)
plt.ylim(-np.pi, np.pi)
plt.title(r"$\theta(t)$")
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$\theta(rad)$")
plt.legend(loc="upper right")

ax = plt.subplot(gs[1, 0])
plt.plot(df[0], df[3], "o-", markersize=2, color="C0", label=r"$x(t)$")
plt.plot(df[0], df[4], "o--", markersize=2, color="C1", label=r"$y(t)$")
plt.title("Position")
plt.xlim(left=0)
ax.set_ylim(-1.5, 1.5)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$Position(m)$")
plt.legend(loc="upper right")
plt.tight_layout()
plt.savefig("plots/ii_params.pdf")

fig, ax = plt.subplots(subplot_kw={"projection":"polar"})
plt.plot(df[2], df[1], "o-", markersize=2, color="C0", label="trace")
ax.set_rmax(1.5)
plt.title("Trajectory")
plt.legend(loc="upper right")
plt.tight_layout()
plt.savefig("plots/ii_polar.pdf")

fig, ax = plt.subplots()
plt.plot(df[3], df[4], "o-", markersize=2, color="C0", label="trace")
plt.xlim(-1.5, 1.5)
plt.ylim(-1.5, 1.5)
plt.title("Trajectory")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend(loc="upper right")
ax.set_aspect("equal")
plt.tight_layout()
plt.savefig("plots/ii_cart.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
import matplotlib.gridspec as gridspec

custom_rcparams = {
    "axes.labelsize": 6,
    "axes.titlesize": 8,
    "axes.grid": True,
    # Figure
    "figure.autolayout": True,
    "figure.titlesize": 9,
    # "figure.dpi": 200,
    "figure.figsize": (9, 4),
    "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)

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

# Extract data
time = df[0].values[::1]
radius = df[1].values[::1]
df_theta = df[2].values
x0 = df[3].iloc[0] - 1
y0 = df[4].iloc[0]
theta = np.arctan((df[4].values - y0)/(df[3].values - x0))
pos_x = df[3].values[::1]
pos_y = df[4].values[::1]

gs = gridspec.GridSpec(1, 2, width_ratios=[1.5, 1])
# Plot
fig = plt.figure()
ax1 = fig.add_subplot(gs[0, 0], projection="polar")
ax2 = fig.add_subplot(gs[0, 1], )
line1, = ax1.plot([], [], 'o', lw=2, label="particle")
line2, = ax2.plot([], [], '-', lw=2, label=r"$\theta(t)$")
trace, = ax1.plot([], [], ',-', lw=1, label="trace")
time_template = "time = %.1fs"
time_text = ax1.text(0, 1.0, '', transform=ax1.transAxes)

line = [line1, line2]

ax1.set_aspect("equal")
ax1.set_rmax(1.5)
ax1.legend()

ax2.hlines(np.pi/2, 0, 10*np.pi, color="red", label=r"$\theta=\pi/2$")
ax2.hlines(-np.pi/2, 0, 10*np.pi, color="red", label=r"$\theta=-\pi/2$")
ax2.set_ylim(-np.pi, np.pi)
ax2.set_xlim(0, 5*np.pi)
ax2.set_xlabel("Time(s)")
ax2.set_ylabel(r"$\theta(rad)$")
ax2.set_aspect(2)
ax2.legend(loc="upper right")

def init():
    line[0].set_data([], [])
    line[1].set_data([], [])
```

```python
    trace.set_data([], [])
    return line, trace

def animate(i):
    global time, radius, df_theta, pos_x, pos_y, theta
    # line[0].set_data(pos_x[i], pos_y[i])
    line[0].set_data(df_theta[i], radius[i])
    line[1].set_data(time[:i], theta[:i])
    # trace.set_data(pos_x[:i], pos_y[:i])
    trace.set_data(df_theta[:i], radius[:i])
    time_text.set_text(time_template % (time[i]))
    return line, trace, time_text

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

capture_no = 0
ani = FuncAnimation(fig, animate, frames=len(time), interval=1, 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 angular velocity(omega) and radius (R)
0.5 1.0
 Enter the value of theta0
0.0
 Enter t0, tf, dt
0.0 20.0 0.1
 -----------------------------------------------------------
 omega =    0.5000
 T =   12.5664
 theta0 =    0.0000
 R =    1.0000
 t0, tf, dt =    0.0000   20.0000    0.1000
 -----------------------------------------------------------
       time          R   theta(t)      x(t)      y(t)
  0.000000   1.000000   0.000000   1.000000   0.000000
  0.100000   1.000000   0.050000   0.998750   0.049979
  0.200000   1.000000   0.100000   0.995004   0.099833
  0.300000   1.000000   0.150000   0.988771   0.149438
  0.400000   1.000000   0.200000   0.980067   0.198669
  0.500000   1.000000   0.250000   0.968912   0.247404
  0.600000   1.000000   0.300000   0.955336   0.295520
  0.700000   1.000000   0.350000   0.939373   0.342898
  0.800000   1.000000   0.400000   0.921061   0.389418
  0.900000   1.000000   0.450000   0.900447   0.434966
  1.000000   1.000000   0.500000   0.877583   0.479426
  1.100000   1.000000   0.550000   0.852525   0.522687
  1.200000   1.000000   0.600000   0.825336   0.564642
  1.300000   1.000000   0.650000   0.796084   0.605186
  1.400000   1.000000   0.700000   0.764842   0.644218
  1.500000   1.000000   0.750000   0.731689   0.681639
  1.600000   1.000000   0.800000   0.696707   0.717356
  1.700000   1.000000   0.850000   0.659983   0.751280
  1.800000   1.000000   0.900000   0.621610   0.783327
  1.900000   1.000000   0.950000   0.581683   0.813416
  2.000000   1.000000   1.000000   0.540302   0.841471
  2.100000   1.000000   1.050000   0.497571   0.867423
  2.200000   1.000000   1.100000   0.453596   0.891207
  2.300000   1.000000   1.150000   0.408487   0.912764
  2.400000   1.000000   1.200000   0.362358   0.932039
  2.500000   1.000000   1.250000   0.315322   0.948985
  2.600000   1.000000   1.300000   0.267499   0.963558
  2.700000   1.000000   1.350000   0.219007   0.975723
  2.800000   1.000000   1.400000   0.169967   0.985450
  2.900000   1.000000   1.450000   0.120503   0.992713
  3.000000   1.000000   1.500000   0.070737   0.997495
  3.100000   1.000000   1.550000   0.020795   0.999784
  3.200000   1.000000   1.600000  -0.029200   0.999574
  3.300000   1.000000   1.650000  -0.079121   0.996865
```

```
  3.400000   1.000000   1.700000  -0.128844   0.991665
  3.500000   1.000000   1.750000  -0.178246   0.983986
  3.600000   1.000000   1.800000  -0.227202   0.973848
  3.700000   1.000000   1.850000  -0.275590   0.961275
  3.800000   1.000000   1.900000  -0.323290   0.946300
  3.900000   1.000000   1.950000  -0.370181   0.928960
  4.000000   1.000000   2.000000  -0.416147   0.909297
  4.100000   1.000000   2.050000  -0.461073   0.887362
  4.200000   1.000000   2.100000  -0.504846   0.863209
  4.300000   1.000000   2.150000  -0.547358   0.836899
  4.400000   1.000000   2.200000  -0.588501   0.808496
  4.500000   1.000000   2.250000  -0.628174   0.778073
  4.600000   1.000000   2.300000  -0.666276   0.745705
  4.700000   1.000000   2.350000  -0.702713   0.711473
  4.800000   1.000000   2.400000  -0.737394   0.675463
  4.900000   1.000000   2.450000  -0.770231   0.637765
  5.000000   1.000000   2.500000  -0.801144   0.598472
  5.100000   1.000000   2.550000  -0.830054   0.557684
  5.200000   1.000000   2.600000  -0.856889   0.515501
  5.300000   1.000000   2.650000  -0.881582   0.472031
  5.400000   1.000000   2.700000  -0.904072   0.427380
  5.500000   1.000000   2.750000  -0.924302   0.381661
  5.600000   1.000000   2.800000  -0.942222   0.334988
  5.700000   1.000000   2.850000  -0.957787   0.287478
  5.800000   1.000000   2.900000  -0.970958   0.239249
  5.900000   1.000000   2.950000  -0.981702   0.190423
  6.000000   1.000000   3.000000  -0.989992   0.141120
  6.100000   1.000000   3.050000  -0.995808   0.091465
  6.200000   1.000000   3.100000  -0.999135   0.041581
  6.300000   1.000000   3.150000  -0.999965  -0.008407
  6.400000   1.000000   3.200000  -0.998295  -0.058374
  6.500000   1.000000   3.250000  -0.994130  -0.108195
  6.600000   1.000000   3.300000  -0.987480  -0.157746
  6.700000   1.000000   3.350000  -0.978362  -0.206902
  6.800000   1.000000   3.400000  -0.966798  -0.255541
  6.900000   1.000000   3.450000  -0.952818  -0.303542
  7.000000   1.000000   3.500000  -0.936457  -0.350783
  7.100000   1.000000   3.550000  -0.917755  -0.397148
  7.200000   1.000000   3.600000  -0.896758  -0.442520
  7.300000   1.000000   3.650000  -0.873521  -0.486787
  7.400000   1.000000   3.700000  -0.848100  -0.529836
  7.500000   1.000000   3.750000  -0.820559  -0.571561
  7.600000   1.000000   3.800000  -0.790968  -0.611858
  7.700000   1.000000   3.850000  -0.759399  -0.650625
  7.800000   1.000000   3.900000  -0.725932  -0.687766
  7.900000   1.000000   3.950000  -0.690651  -0.723188
  8.000000   1.000000   4.000000  -0.653644  -0.756802
  8.100000   1.000000   4.050000  -0.615002  -0.788525
  8.200000   1.000000   4.100000  -0.574824  -0.818277
  8.300000   1.000000   4.150000  -0.533209  -0.845984
  8.400000   1.000000   4.200000  -0.490261  -0.871576
  8.500000   1.000000   4.250000  -0.446087  -0.894989
  8.600000   1.000000   4.300000  -0.400799  -0.916166
  8.700000   1.000000   4.350000  -0.354509  -0.935053
  8.800000   1.000000   4.400000  -0.307333  -0.951602
  8.900000   1.000000   4.450000  -0.259389  -0.965773
  9.000000   1.000000   4.500000  -0.210796  -0.977530
  9.100000   1.000000   4.550000  -0.161676  -0.986844
  9.200000   1.000000   4.600000  -0.112153  -0.993691
  9.300000   1.000000   4.650000  -0.062349  -0.998054
  9.400000   1.000000   4.700000  -0.012389  -0.999923
  9.500000   1.000000   4.750000   0.037602  -0.999293
  9.600000   1.000000   4.800000   0.087499  -0.996165
  9.700000   1.000000   4.850000   0.137177  -0.990547
  9.800000   1.000000   4.900000   0.186512  -0.982453
  9.900000   1.000000   4.950000   0.235381  -0.971903
 10.000000   1.000000   5.000000   0.283662  -0.958924
 10.100000   1.000000   5.050000   0.331234  -0.943549
 10.200000   1.000000   5.100000   0.377978  -0.925815
 10.300000   1.000000   5.150000   0.423777  -0.905767
 10.400000   1.000000   5.200000   0.468517  -0.883455
 10.500000   1.000000   5.250000   0.512085  -0.858934
 10.600000   1.000000   5.300000   0.554374  -0.832267
 10.700000   1.000000   5.350000   0.595278  -0.803520
 10.800000   1.000000   5.400000   0.634693  -0.772764
 10.900000   1.000000   5.450000   0.672522  -0.740077
 11.000000   1.000000   5.500000   0.708670  -0.705540
 11.100000   1.000000   5.550000   0.743046  -0.669240
 11.200000   1.000000   5.600000   0.775566  -0.631267
```

```
11.300000   1.000000   5.650000   0.806147  -0.591716
11.400000   1.000000   5.700000   0.834713  -0.550686
11.500000   1.000000   5.750000   0.861192  -0.508279
11.600000   1.000000   5.800000   0.885520  -0.464602
11.700000   1.000000   5.850000   0.907633  -0.419764
11.800000   1.000000   5.900000   0.927478  -0.373877
11.900000   1.000000   5.950000   0.945005  -0.327055
12.000000   1.000000   6.000000   0.960170  -0.279415
12.100000   1.000000   6.050000   0.972935  -0.231078
12.200000   1.000000   6.100000   0.983268  -0.182163
12.300000   1.000000   6.150000   0.991144  -0.132792
12.400000   1.000000   6.200000   0.996542  -0.083089
12.500000   1.000000   6.250000   0.999449  -0.033179
12.600000   1.000000   6.300000   0.999859   0.016814
12.700000   1.000000   6.350000   0.997769   0.066765
12.800000   1.000000   6.400000   0.993185   0.116549
12.900000   1.000000   6.450000   0.986119   0.166042
13.000000   1.000000   6.500000   0.976588   0.215120
13.100000   1.000000   6.550000   0.964616   0.263660
13.200000   1.000000   6.600000   0.950233   0.311541
13.300000   1.000000   6.650000   0.933474   0.358644
13.400000   1.000000   6.700000   0.914383   0.404850
13.500000   1.000000   6.750000   0.893006   0.450044
13.600000   1.000000   6.800000   0.869397   0.494113
13.700000   1.000000   6.850000   0.843616   0.536948
13.800000   1.000000   6.900000   0.815725   0.578440
13.900000   1.000000   6.950000   0.785796   0.618486
14.000000   1.000000   7.000000   0.753902   0.656987
14.100000   1.000000   7.050000   0.720124   0.693845
14.200000   1.000000   7.100000   0.684547   0.728969
14.300000   1.000000   7.150000   0.647258   0.762271
14.400000   1.000000   7.200000   0.608351   0.793668
14.500000   1.000000   7.250000   0.567924   0.823081
14.600000   1.000000   7.300000   0.526078   0.850437
14.700000   1.000000   7.350000   0.482916   0.875667
14.800000   1.000000   7.400000   0.438547   0.898708
14.900000   1.000000   7.450000   0.393083   0.919503
15.000000   1.000000   7.500000   0.346635   0.938000
15.100000   1.000000   7.550000   0.299322   0.954152
15.200000   1.000000   7.600000   0.251260   0.967920
15.300000   1.000000   7.650000   0.202570   0.979268
15.400000   1.000000   7.700000   0.153374   0.988168
15.500000   1.000000   7.750000   0.103794   0.994599
15.600000   1.000000   7.800000   0.053955   0.998543
15.700000   1.000000   7.850000   0.003982   0.999992
15.800000   1.000000   7.900000  -0.046002   0.998941
15.900000   1.000000   7.950000  -0.095871   0.995394
16.000000   1.000000   8.000000  -0.145500   0.989358
16.100000   1.000000   8.050000  -0.194765   0.980850
16.200000   1.000000   8.100000  -0.243544   0.969890
16.300000   1.000000   8.150000  -0.291714   0.956506
16.400000   1.000000   8.200000  -0.339155   0.940731
16.500000   1.000000   8.250000  -0.385748   0.922604
16.600000   1.000000   8.300000  -0.431377   0.902172
16.700000   1.000000   8.350000  -0.475928   0.879484
16.800000   1.000000   8.400000  -0.519289   0.854599
16.900000   1.000000   8.450000  -0.561352   0.827577
17.000000   1.000000   8.500000  -0.602012   0.798487
17.100000   1.000000   8.550000  -0.641167   0.767401
17.200000   1.000000   8.600000  -0.678720   0.734397
17.300000   1.000000   8.650000  -0.714576   0.699557
17.400000   1.000000   8.700000  -0.748647   0.662969
17.500000   1.000000   8.750000  -0.780846   0.624724
17.600000   1.000000   8.800000  -0.811093   0.584917
17.700000   1.000000   8.850000  -0.839313   0.543648
17.800000   1.000000   8.900000  -0.865435   0.501021
17.900000   1.000000   8.950000  -0.889394   0.457141
18.000000   1.000000   9.000000  -0.911130   0.412118
18.100000   1.000000   9.050000  -0.930589   0.366066
18.200000   1.000000   9.100000  -0.947722   0.319098
18.300000   1.000000   9.150000  -0.962485   0.271333
18.400000   1.000000   9.200000  -0.974844   0.222890
18.500000   1.000000   9.250000  -0.984765   0.173889
18.600000   1.000000   9.300000  -0.992225   0.124454
18.700000   1.000000   9.350000  -0.997205   0.074708
18.800000   1.000000   9.400000  -0.999693   0.024775
18.900000   1.000000   9.450000  -0.999682  -0.025219
19.000000   1.000000   9.500000  -0.997172  -0.075151
19.100000   1.000000   9.550000  -0.992170  -0.124895
```

```
19.200000   1.000000   9.600000  -0.984688  -0.174327
19.300000   1.000000   9.650000  -0.974745  -0.223323
19.400000   1.000000   9.700000  -0.962365  -0.271761
19.500000   1.000000   9.750000  -0.947580  -0.319519
19.600000   1.000000   9.800000  -0.930426  -0.366479
19.700000   1.000000   9.850000  -0.910947  -0.412523
19.800000   1.000000   9.900000  -0.889191  -0.457536
19.900000   1.000000   9.950000  -0.865213  -0.501405
----------------------------------------------------------
```

5.2 Plots

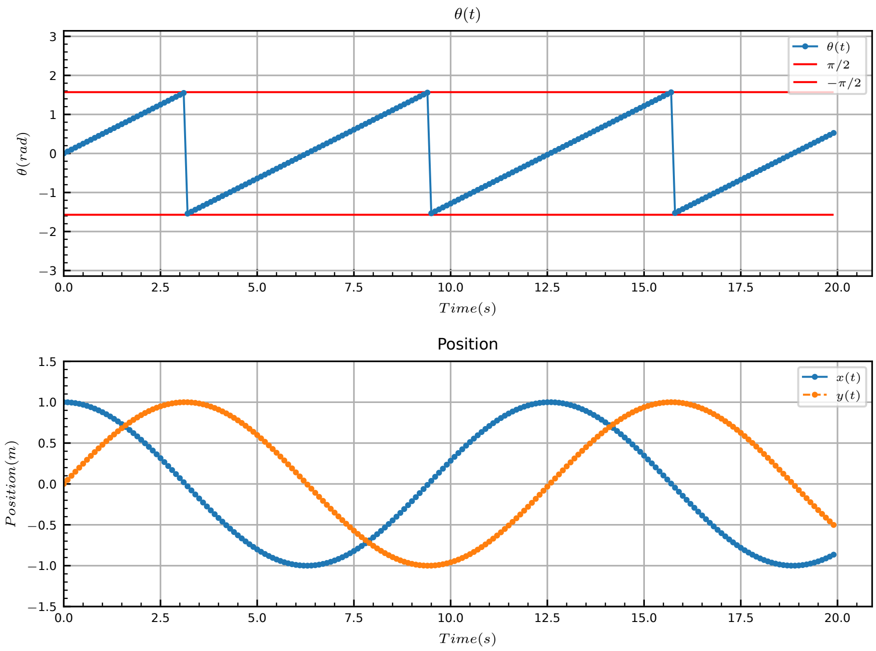


Figure 1: θ and position vs. *time*

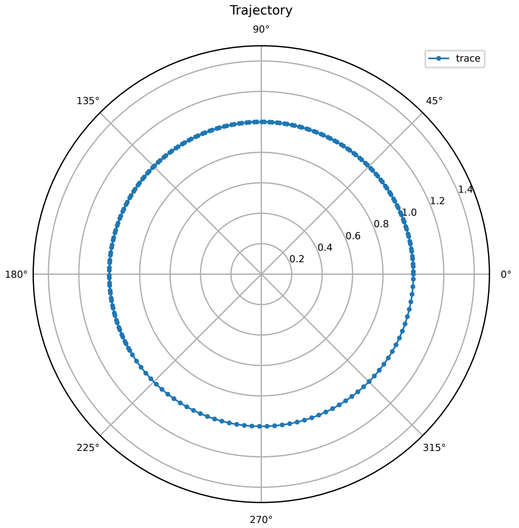


Figure 2: Trajectory in polar coordinates

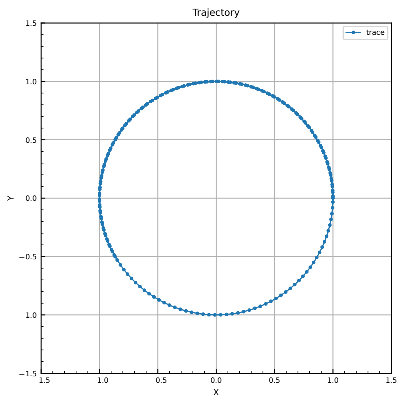
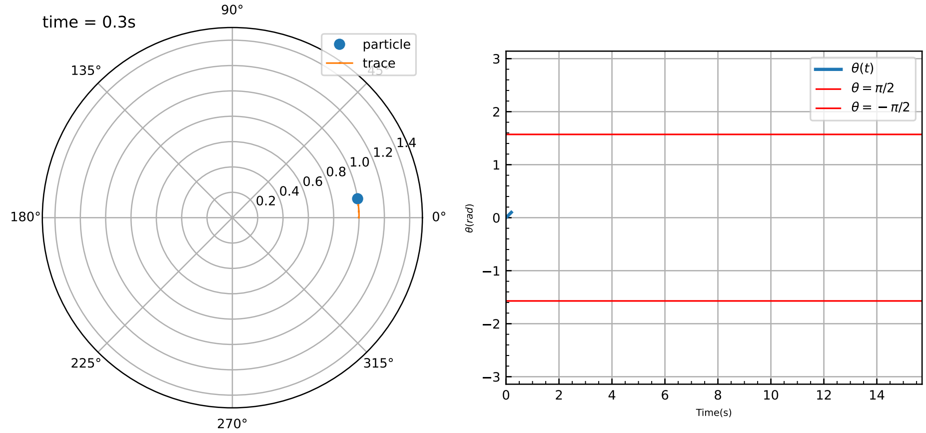


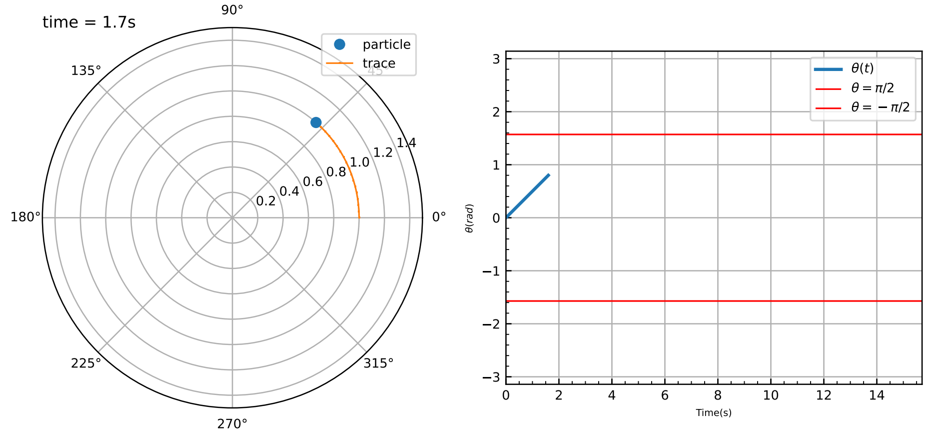
Figure 3: Trajectory in cartesian coordinates

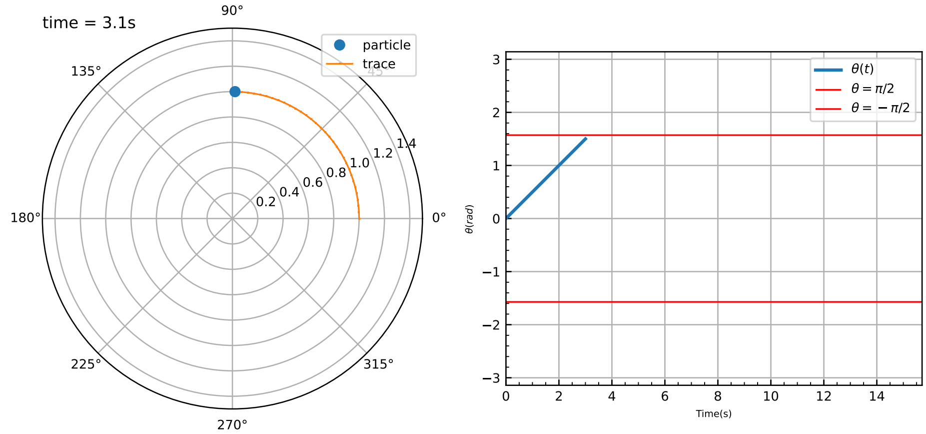
5.3 Animation

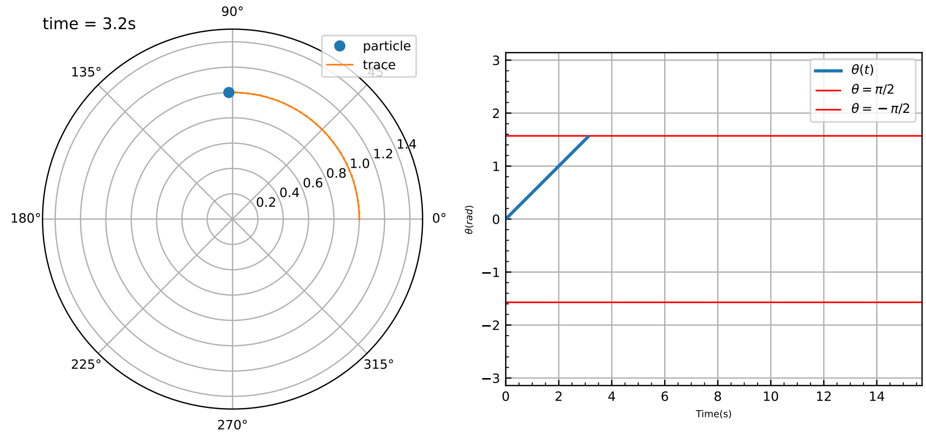
Note: Input parameters,

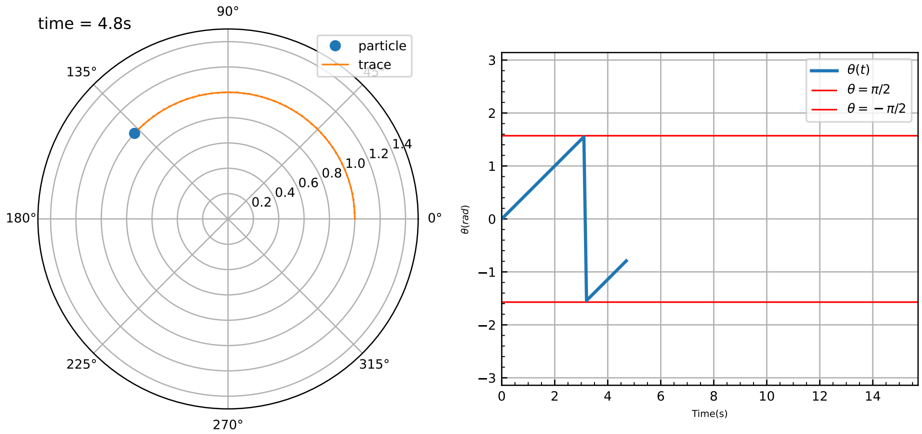
$$\begin{aligned} \omega &= 0.5\ rad/s \\ R &= 1\ m \\ \implies T &= 12.5664\ s \end{aligned} \tag{6}$$

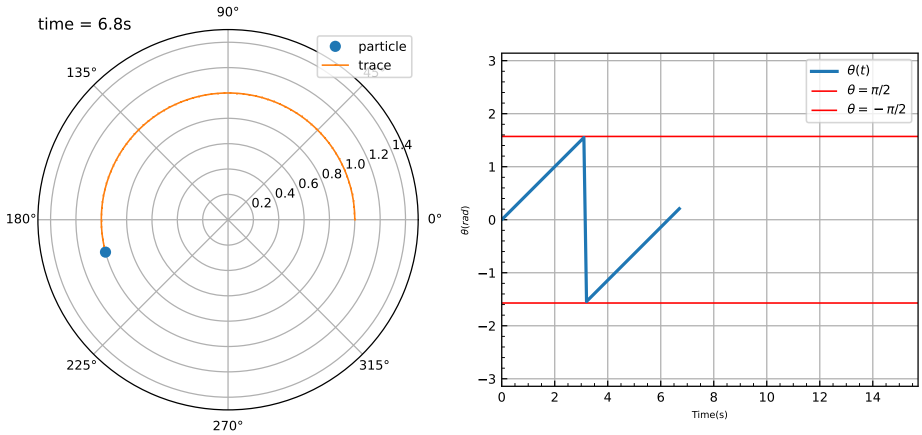

time = 0.3s
particle
trace
$\theta(t)$
$\theta = \pi/2$
$\theta = -\pi/2$
$\theta(rad)$
Time(s)

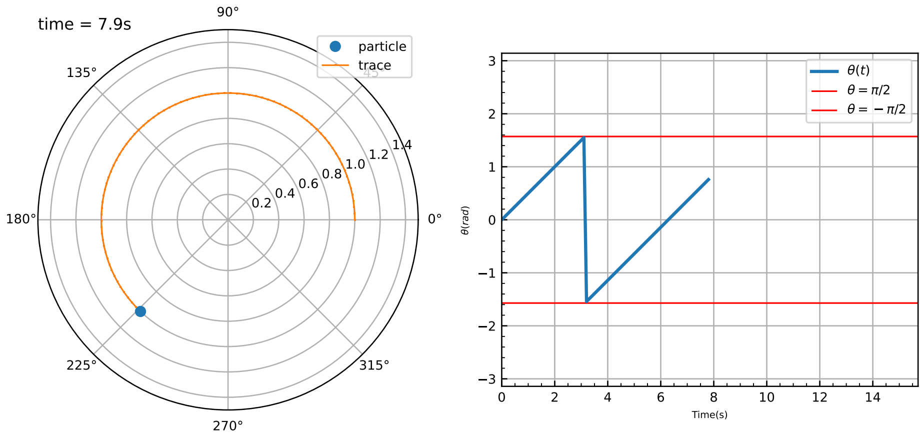

time = 1.7s
particle
trace
$\theta(t)$
$\theta = \pi/2$
$\theta = -\pi/2$
$\theta(rad)$
Time(s)

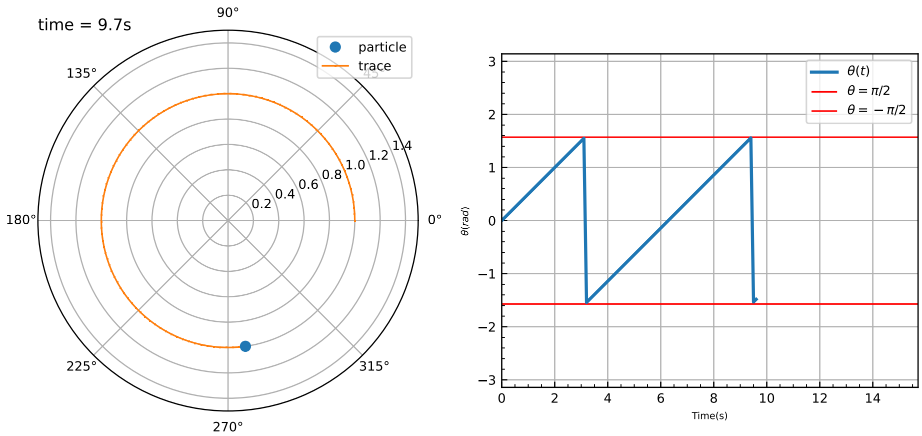

time = 3.1s
particle
trace
$\theta(t)$
$\theta = \pi/2$
$\theta = -\pi/2$
$\theta(rad)$
Time(s)


time = 3.2s
particle
trace
$\theta(t)$
$\theta = \pi/2$
$\theta = -\pi/2$
$\theta(rad)$
Time(s)


time = 4.8s
particle
trace
90°
135°
180°
225°
270°
315°
0°
0.2
0.4
0.6
0.8
1.0
1.2
1.4
θ(t)
θ = π/2
θ = −π/2
θ(rad)
Time(s)


time = 6.8s
particle
trace
90°
135°
180°
225°
270°
315°
0°
θ(t)
θ = π/2
θ = −π/2
θ(rad)
Time(s)


time = 7.9s
particle
trace
90°
135°
180°
225°
270°
315°
0°
θ(t)
θ = π/2
θ = −π/2
θ(rad)
Time(s)


time = 9.7s
particle
trace
90°
135°
180°
225°
270°
315°
0°
θ(t)
θ = π/2
θ = −π/2
θ(rad)
Time(s)

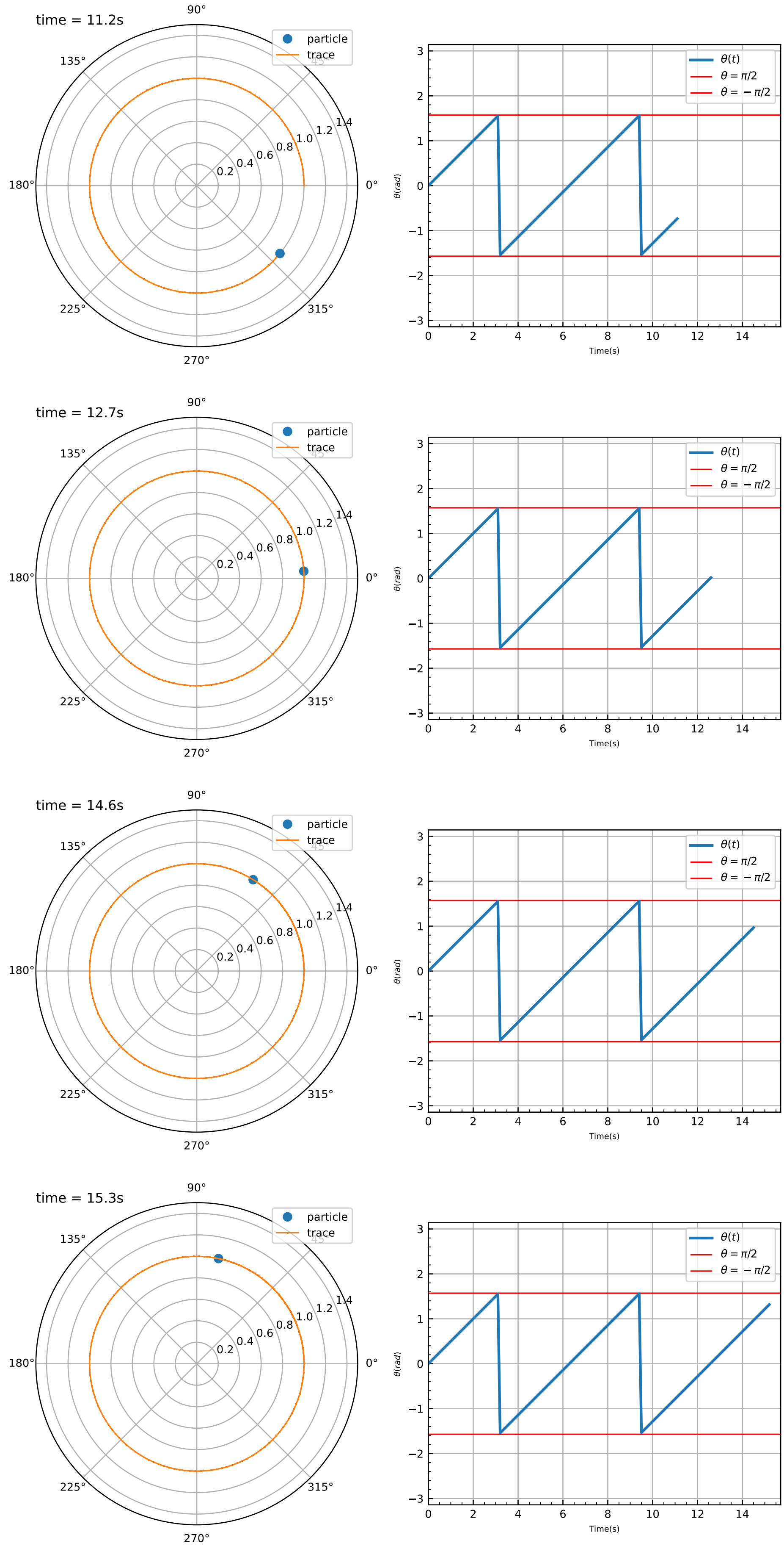


Figure 4: Animation for uniform circular motion in polar coordinates

6 Remarks

The programs can be used to trace and simulate the motion of any particle in uniform circular motion by defining the required parameters.

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

These programs can be useful in cases of circular motion, and with some modifications with satellite and planetary motions.