Program-M5

Devansh Shukla
I18PH021

Write and execute a FORTRAN program to simulate motion of a particle following projectile motion.

1 Theory

1.1 Projectile motion

Considering the air-resistance is negligible and gravitaional accleration is constant; when we throw something from the Earth's surface, it could be descibed using the following parametric equations,

Assuming the initlal conditions are $x_0 = 0$, $y_0 = 0$, $t_0 = 0$

Let the particle be thrown at an initial velocity of v_0 and at an angle of θ

$$v_{0x} = v_0 \ \cos(\theta)$$
$$v_{0y} = v_0 \ \sin(\theta)$$

$$\boxed{\begin{aligned} x(t) &= v_{0x}t \\ y(t) &= v_{0y}t \ - \ \frac{1}{2}gt^2 \\ v_x(t) &= v_{0x} \\ v_y(t) &= v_{0y} \ - \ gt \end{aligned}} \tag{1}$$

1.1.1 Time of flight

At the maximum height(maxima) of the particle's trajectory, $v_y = 0$; so the time of flight is double the time taken to reach the maxima.

$$v_y(t) = v_{0y} \ - \ gt$$
$$v_y = 0$$
$$\implies t_f = \frac{v_{0y}}{g}$$

$$\begin{aligned} \text{Time of flight, } T_f &= 2t_f \\ T_f &= \frac{2v_{0y}}{g} \\ &\boxed{T_f = \frac{2v_0 \ \sin(\theta)}{g}} \end{aligned} \tag{2}$$

1.1.2 Range

Range is the maximum distance travelled by the particle with $t = T_f$

$$\begin{aligned} x &= v_{0x}T_f \\ \text{Range, } R &= v_0 \ \cos(\theta)\frac{2v_0 \ \sin(\theta)}{g} \\ &\boxed{R = \frac{v_0^2 \ \sin(2\theta)}{g}} \\ \frac{dR}{d\theta} &= 2\frac{v_0^2 \ \cos(2\theta)}{g} = 0 \\ \implies & \cos(2\theta) = 0 \\ &\boxed{\theta = (2n \ + \ 1)\frac{\pi}{4}} \end{aligned} \tag{3}$$

Therefore, the range is maximum for $\theta = 45°$.

1.1.3 Maxiumum height travelled

$$
\begin{aligned}
y(t) &= v_{0y}t - \frac{1}{2}gt^2 \\
H &= v_0 \sin(\theta)t_f - \frac{1}{2}gt_f^2; \text{ where } t_f = \frac{1}{2}T_f \\
H &= v_0 \sin(\theta)\frac{v_0 \sin(\theta)}{g} - \frac{1}{2}g\left(\frac{v_0 \sin(\theta)}{g}\right)^2 \\
&\boxed{H = \frac{v_0^2 \sin^2(\theta)}{2g}} \qquad (4) \\
\frac{dH}{d\theta} &= \frac{2v_0^2 \sin(\theta) \cos(\theta)}{2g} = 0 \\
\implies & \sin(2\theta) = 0 \\
&\boxed{\theta = \frac{n\pi}{2}}
\end{aligned}
$$

Therefore, the particle will achieve maximumu height at $\theta = 90°$.

1.1.4 Relation between R & H

$$
\boxed{\frac{H}{R} = \frac{1}{4} \tan(\theta)} \qquad (5)
$$

2 Numerical Solution

$$
\begin{aligned}
\theta &= 45° = \frac{\pi}{4} \\
v_0 &= 10\ m/s
\end{aligned} \qquad (6)
$$

$$
\begin{aligned}
\text{Time of flight, } T_f &= \frac{2v_0 \sin(\theta)}{g} \\
&= \frac{20 \sin(\pi/4)}{9.81} \\
&= 1.4416\ s
\end{aligned} \qquad (7)
$$

$$
\begin{aligned}
\text{Range, } R &= \frac{v_0^2 \sin(2\theta)}{g} \\
&= \frac{(10)^2 \sin(\pi/2)}{9.81} \\
&= 10.1937\ m
\end{aligned} \qquad (8)
$$

$$
\begin{aligned}
\text{Maximum height, } H &= \frac{v_0^2 \sin^2(\theta)}{2g} \\
&= \frac{(10)^2 \sin^2(\pi/4)}{2 \times 9.81} \\
&= 2.5484\ m
\end{aligned} \qquad (9)
$$

$$
\boxed{\begin{aligned}
T_f &= 1.4416\ s \\
R &= 10.1937\ m \\
H &= 2.5484\ m
\end{aligned}} \qquad (10)
$$

3 Program Algorithm

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

1. Program open.
2. Define variables (PI, g, x, y, vx, vy, theta, v0x, v0y, v0, t0, tf, dt, t, fmt1).
3. Open a writable data file.
4. Get input from user for inital velocity (v0) and initial theta(theta) and time period(t0, tf, dt).
5. If the time-increment is less-than or equal to 0, terminate the program with message Illegal value of dt

6. Print parameters to stdout for the user.
7. Write appropiate comments in the data file and initialize other parameters.
8. Define a do while loop with index t which runs from t0 to tf.
9. Compute the parameters using the functions theta(t), dtheta(t), x(t), y(t), vx(t), vy(t).
10. Write the parameters to stdout and data file.
11. Increment the index according to t = t + dt
12. End do-while loop.
13. Close data file.
14. Program close.

4 Program

4.1 Fortran program:

For computing the parameters

```
!=====================================
! projectile.f90
! Author: Devansh Shukla
!-------------------------------------

program projectile
    ! Program to compute motion of a particle moving in projectile motion.

    implicit none
    real*8, parameter :: PI=3.141592, g=9.81
    real*8 :: x, y, vx, vy
    real*8 :: theta, v0x, v0y, v0
    real*8 :: t0, tf, dt, t
    character(len=*), parameter :: fmt1 = "(F12.4, F12.4, F12.4, F12.4, F12.4)"

    open(unit=8, file="Projectile.dat")

    print *, "-----------------------------------------"
    print *, "Enter v0, theta(deg)"
    read *, v0, theta
    print *, "Enter t0, tf, dt"
    read *, t0, tf, dt
    print *, "-----------------------------------------"

    print *, "v0, theta =", v0, theta
    print *, "t0, tf, dt=", t0, tf, dt

    if (v0 .le. 0.0) stop "Illegal value of v0<=0"
    if (theta .lt. 0.0 .or. theta .gt. 90.0 ) &
    stop "Illegal value of theta"
    print *, "-----------------------------------------"

    theta = (PI/180.0) * theta ! theta radians
    v0x = v0*cos(theta)
    v0y = v0*sin(theta)

    print "(A12, A12, A12, A12, A12)", "time", "x(t)", "y(t)", "vx(t)", "vy(t)"
    t = t0
    do while(t <= tf)
        x = v0x * t
        y = v0y * t - 0.5*g*t*t
        vx = v0x
        vy = v0y - g*t
        write (*, fmt1) t, x, y, vx, vy
        write (8, fmt1) t, x, y, vx, vy
        if (y < 0.0) stop "y-ve"
        t = t + dt
    enddo
    print *, "-----------------------------------------"
    close(8)
end program projectile
```

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,
    "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()

# t, x, y, vx, vy
df = pd.read_csv("Projectile.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")
print(df)

gs = gridspec.GridSpec(2, 2)

ax = plt.subplot(gs[0, 0])
ax.plot(df[0], df[1], "o-", markersize=1.5, color="C0", label=r"$x(t)$")
ax.plot(df[0], df[2], "o-", markersize=1.5, color="C1", label=r"$y(t)$")
ax.set_xlim(left=0)
ax.set_ylim(bottom=0)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$Position(m)$")
ax.legend(loc="upper left")
plt.title("Position")

ax = plt.subplot(gs[0, 1])
ax.plot(df[0], df[3], "o-", markersize=1.5, color="C0", label=r"$v_{x}(t)$")
ax.plot(df[0], df[4], "o-", markersize=1.5, color="C1", label=r"$v_{y}(t)$")
ax.set_xlim(left=0)
ax.set_ylim(-8, 10)
ax.set_xlabel(r"$Time(s)$")
ax.set_ylabel(r"$Velocity(m/s)$")
ax.legend(loc="lower left")
plt.title("Velocity")

ax = plt.subplot(gs[1,:])
ax.plot(df[1], df[2], "o-", markersize=1.5, color="C0", label=r"trace")
ax.set_xlim(left=0, right=12)
ax.set_ylim(bottom=0, top=3)
ax.set_xlabel(r"$X$")
ax.set_ylabel(r"$Y$")
ax.legend(loc="upper right")
plt.title("Trajectory")

# plt.show()
plt.savefig("plots/projectile.pdf")

# %%
```

4.3 Python program: Animation

```
#!/usr/bin/env python
"""
Author: Devansh Shukla
"""
```

```python
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": (12, 3.5),
    "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)

# t, x, y, vx, vy
df = pd.read_csv("Projectile.dat", engine="python", delimiter=" ", header=None, skipinitialspace=True, comment="#")
print(df)

time = df[0].values
pos_x = df[1].values
pos_y = df[2].values
vel_x = df[3].values
vel_y = df[4].values
vel = np.sqrt(vel_x*vel_x + vel_y*vel_y)

fig, (ax1, ax2) = plt.subplots(1,2)

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

line_v, = ax2.plot([], [], '-', lw=2, label=r"$v(t)$")
line_vx, = ax2.plot([], [], '-', lw=2, label=r"$v_{x}(t)$")
line_vy, = ax2.plot([], [], '-', lw=2, label=r"$v_{y}(t)$")
ax2.legend()

line = [line1, line_v, line_vx, line_vy,]
ax1.set_xlim(left=0, right=pos_x[-1] + 2.0)
ax1.set_ylim(0, 5)
ax1.set_title("Trajectory")
ax1.set_xlabel(r"$X$")
ax1.set_ylabel(r"$Y$")
ax1.legend(loc="upper right")

ax2.set_xlim(0, time[-1]+0.1)
ax2.set_ylim(-10, 15)
# ax2.set_aspect(4)
ax2.set_ylabel(r"$v(m/s)$")
ax2.set_xlabel("Time(s)")
ax2.legend(loc="upper right")

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, vel

    line[0].set_data(pos_x[i], 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], vel[:i])
    line[2].set_data(time[:i], vel_x[:i])
    line[3].set_data(time[:i], vel_y[:i])
    return line, trace, time_text

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

capture_no = 0
ani = FuncAnimation(fig, animate, frames=len(time), interval=100, 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 v0, theta(deg)
10.0 45.0
 Enter t0, tf, dt
0.0 30.0 0.1
 ----------------------------------------
 v0, theta =   10.000000000000000        45.000000000000000
 t0, tf, dt=   0.0000000000000000        30.000000000000000        0.10000000000000000
 ----------------------------------------
        time        x(t)        y(t)        vx(t)       vy(t)
      0.0000      0.0000      0.0000      7.0711      7.0711
      0.1000      0.7071      0.6581      7.0711      6.0901
      0.2000      1.4142      1.2180      7.0711      5.1091
      0.3000      2.1213      1.6799      7.0711      4.1281
      0.4000      2.8284      2.0436      7.0711      3.1471
      0.5000      3.5355      2.3093      7.0711      2.1661
      0.6000      4.2426      2.4768      7.0711      1.1851
      0.7000      4.9497      2.5463      7.0711      0.2041
      0.8000      5.6569      2.5177      7.0711     -0.7769
      0.9000      6.3640      2.3909      7.0711     -1.7579
      1.0000      7.0711      2.1661      7.0711     -2.7389
      1.1000      7.7782      1.8431      7.0711     -3.7199
      1.2000      8.4853      1.4221      7.0711     -4.7009
      1.3000      9.1924      0.9029      7.0711     -5.6819
      1.4000      9.8995      0.2857      7.0711     -6.6629
      1.5000     10.6066     -0.4297      7.0711     -7.6439
STOP y-ve
```

5.2 Plots

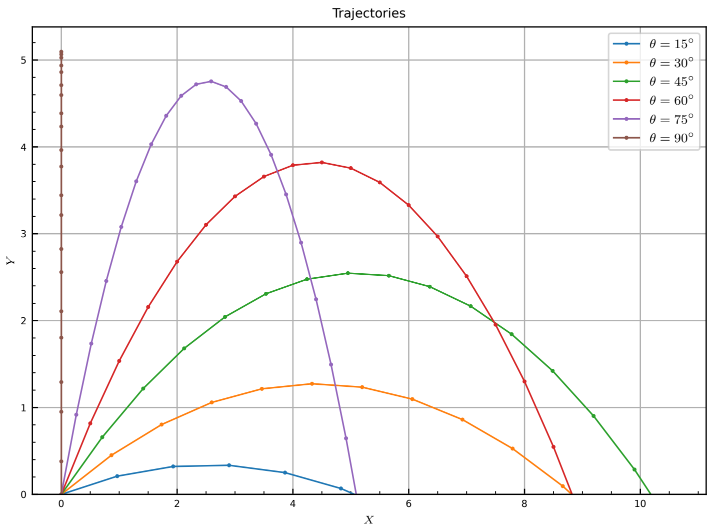


Figure 1: Trajectories at different θ and $v_0 = 10\ m/s$

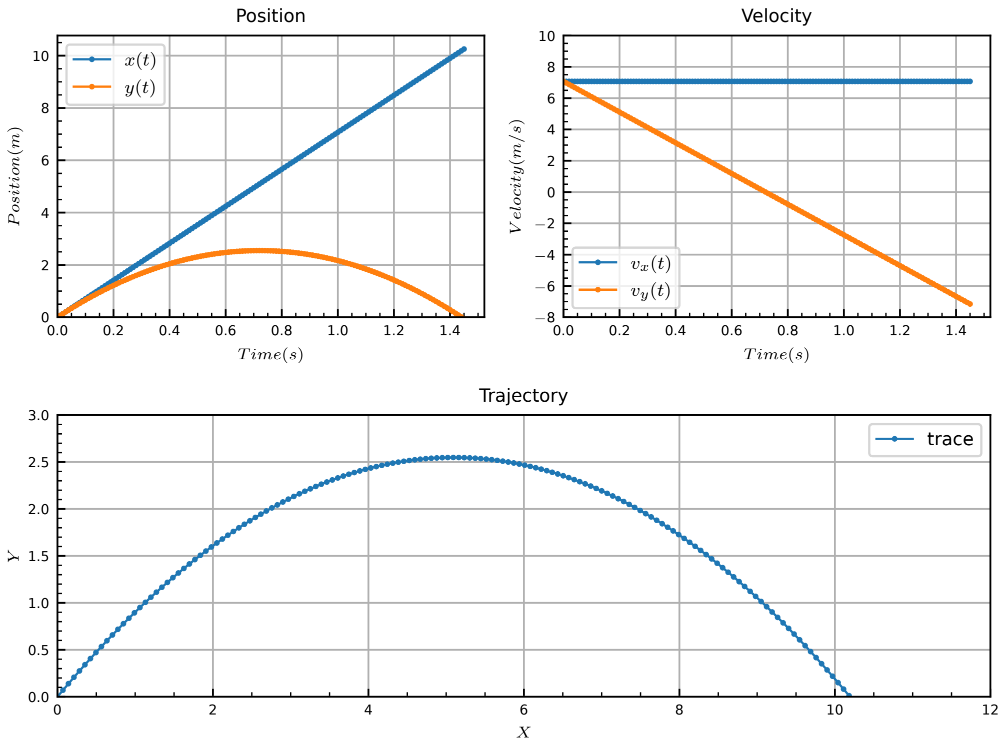
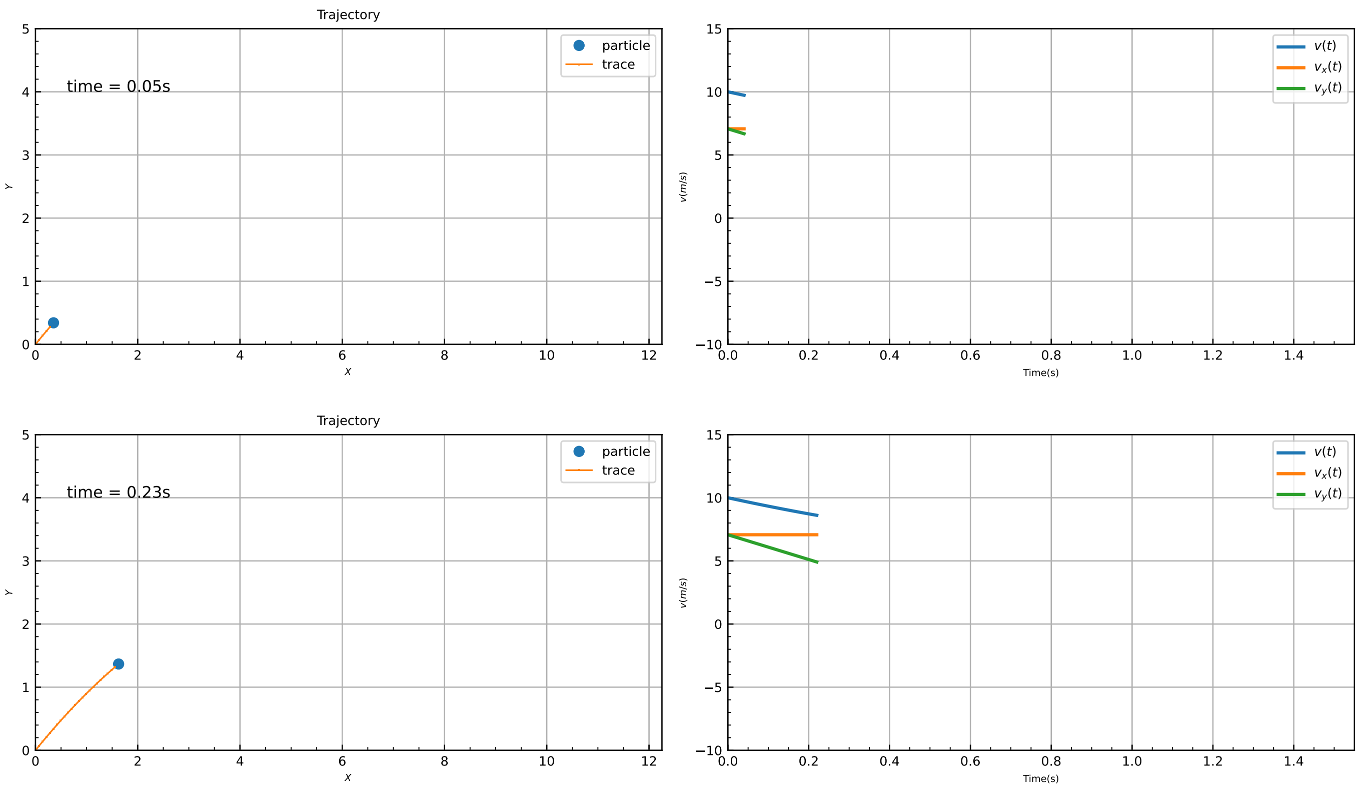


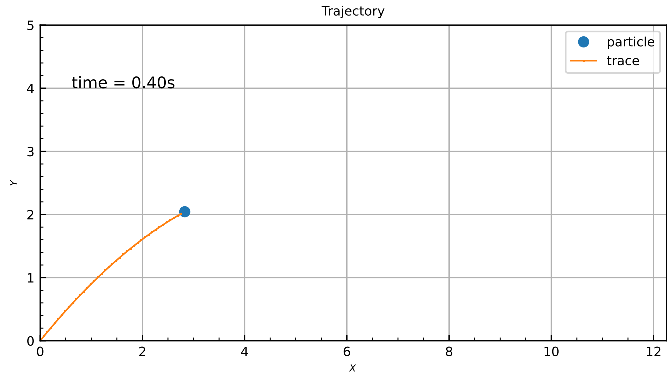
Figure 2: Trajectory at $\theta = 45°$ and $v_0 = 10\ m/s$

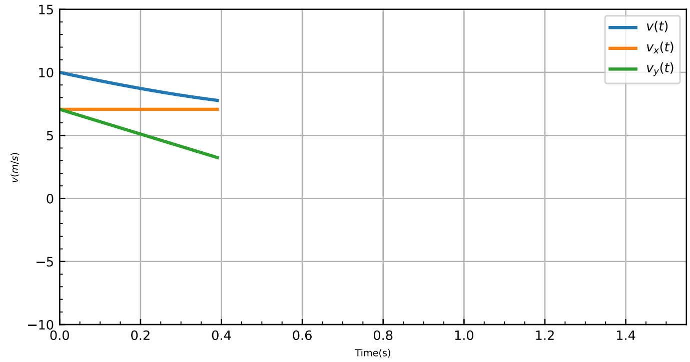
5.3 Animation

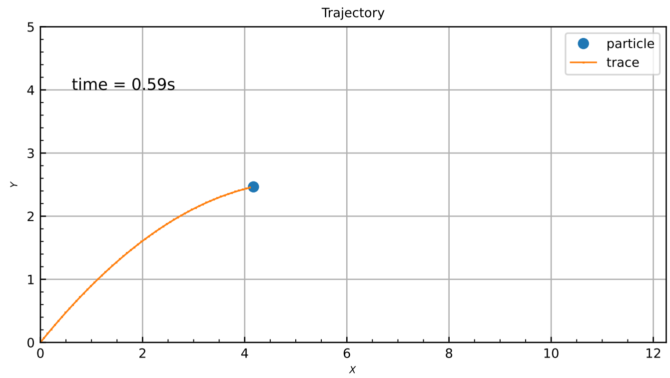
Note: Input parameters,

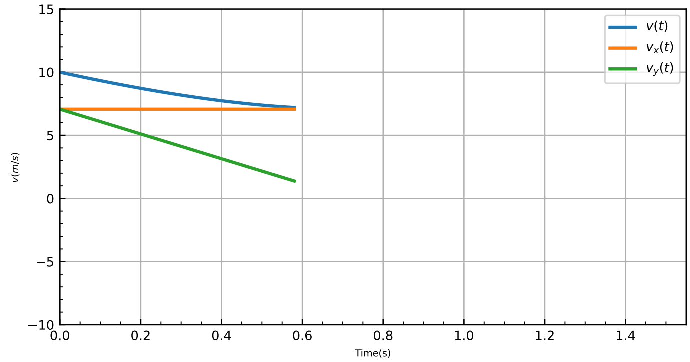
$$\begin{gathered} \theta_0 = 45^{\circ} \\ \text{Initial velocity, } v_0 = 10\ m/s \end{gathered} \tag{11}$$

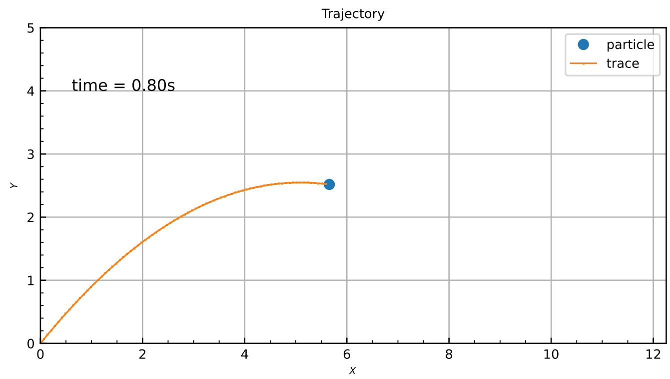


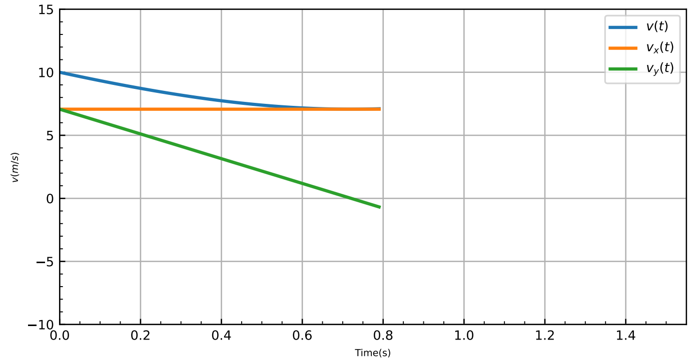

Trajectory
time = 0.40s
particle
trace
Y
x

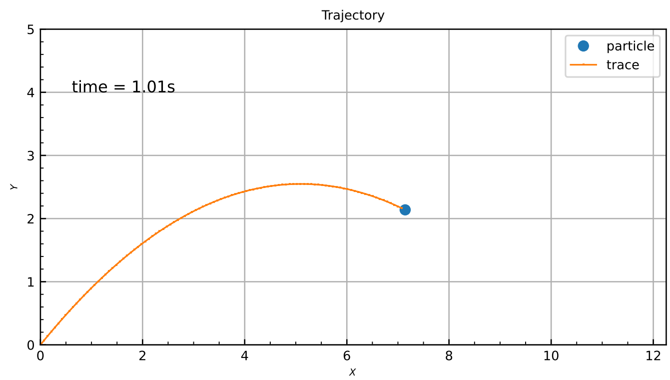

v(t)
$v_x(t)$
$v_y(t)$
v(m/s)
Time(s)

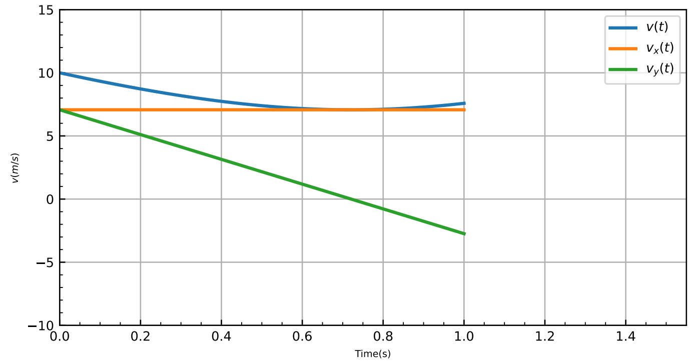

Trajectory
time = 0.59s
particle
trace
Y
x

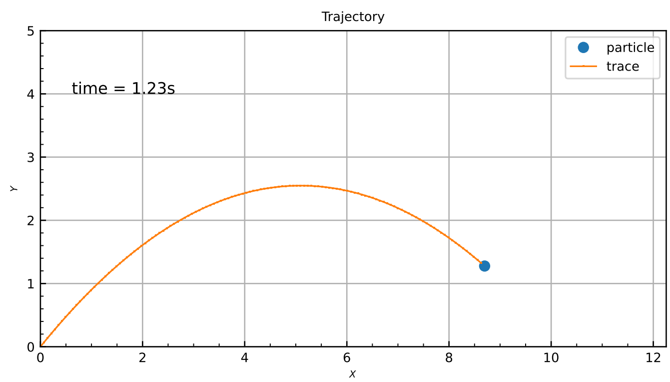

v(t)
$v_x(t)$
$v_y(t)$
v(m/s)
Time(s)

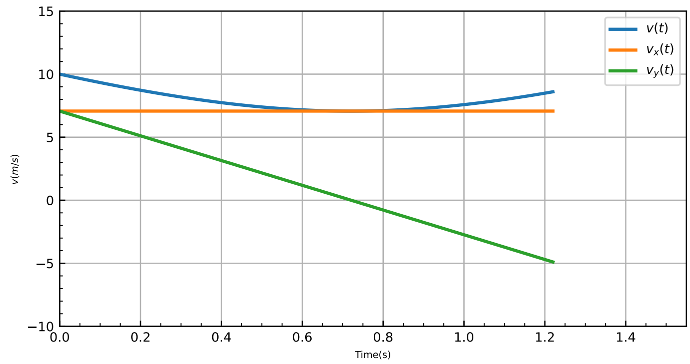

Trajectory
time = 0.80s
particle
trace
Y
x

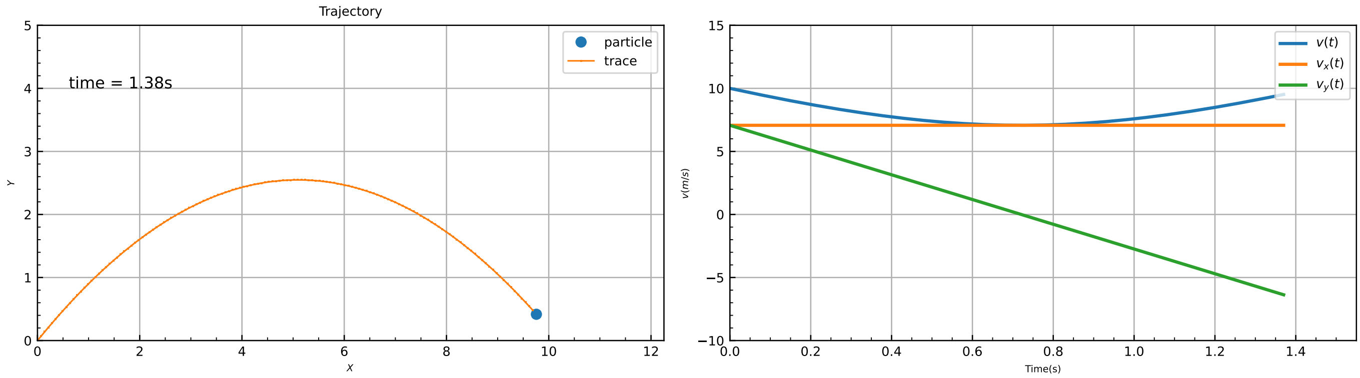

v(t)
$v_x(t)$
$v_y(t)$
v(m/s)
Time(s)

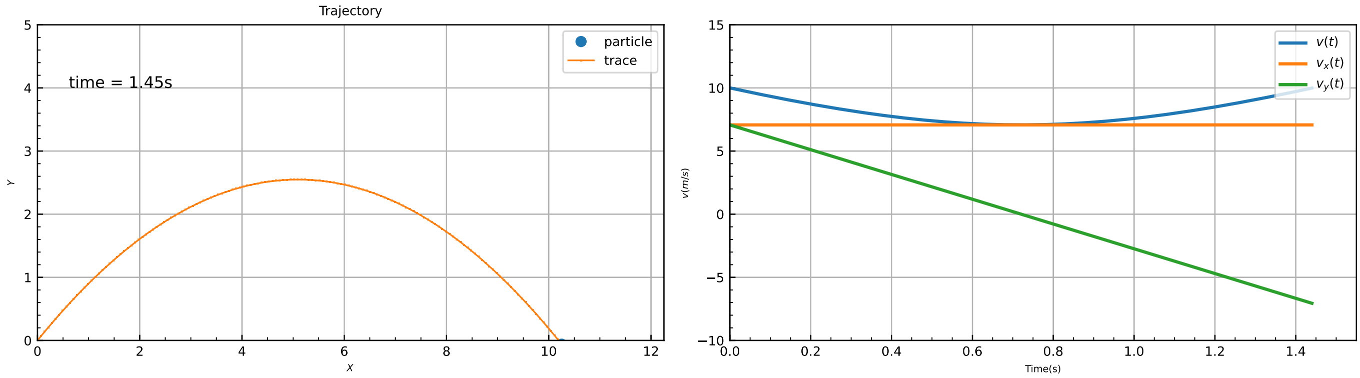

Trajectory
time = 1.01s
particle
trace
Y
x


v(t)
$v_x(t)$
$v_y(t)$
v(m/s)
Time(s)


Trajectory
time = 1.23s
particle
trace
Y
x


v(t)
$v_x(t)$
$v_y(t)$
v(m/s)
Time(s)





6 Remarks

The programs can be used to numerically trace and simulate of a particle moving in projectile motion, provided the required parameters are defined.

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