

Question 2

November 9, 2019

1 Dust Particle Trajectory

1.1 Problem Formulation:

The given problem is stated below:

Consider a dust particle that gets flushed down the sink. The particle takes a helical path down as the liquid swirls around while flowing down as it drains. This can be modeled using the following assumptions: The radial distance of the particle from the center of the sink decreases with increasing time. The vertical distance of the particle decreases with increasing time. The angular velocity of the particle is constant. Model the trajectory of the particle and show the same in a 3D line plot.

1.2 Proposed Solution Methodology:

First, let us find the equation of the particle. We know that x and y should be dependent on time t , and since it is following a helical path with reducing radius, we can assume that:

$$x(t) = f(t) \cos(t), y(t) = f(t) \sin(t)$$

such that $f(T) \rightarrow 0$ where T is the time the particle takes to reach the bottom.

Now, we also know that the particle's angular velocity is constant, hence we can say:

$$\omega = \frac{v}{r} = \frac{\sqrt{v_x^2 + v_y^2}}{x^2 + y^2} = \omega_0 \text{ (a constant)}$$

Now, let us replace $x(t), y(t), v_x(t) = x'(t)$, and $v_y(t) = y'(t)$ into the equation. This yields:

$$\frac{\sqrt{f(t)^2 + f'(t)^2}}{f(t)} = \omega_0$$

On squaring both sides and doing some manipulations, we arrive at:

$$f'(t) = \pm \sqrt{\omega_0^2 - 1} |f(t)|$$

When we solve the above differential equation, we get:

$$f(t) = Ae^{Kt}, \text{ where } K = \pm\sqrt{\omega_0 - 1}$$

Thus, we choose the equation of the curve to be:

$$(x, y, z) = \left(kr_0 e^{-\frac{t}{2k}} \cos(t), kr_0 e^{-\frac{t}{2k}} \sin(t), r_0 - \frac{t}{k} \right), t \in [0, kr_0]$$

1.3 Implementation

We will use implement this in Python using SageMath, as SageMath provides us with a ready library for plotting 3d parametric plots.

We will plot this using the `parametric_plot3d` function of SageMath: (using arbitrary values of r_0 and k)

```
[1]: r_0 = 10
      k=3
      pobject = parametric_plot3d((lambda t: (k*r_0*exp(-t/(2*k)))*cos(t), lambda t:
      ↪ (k*r_0*exp(-t/(2*k)))*sin(t), lambda t: r_0-t/k), (0, k*r_0))
      pobject.show(aspect_ratio=[2,2,2])
```

Graphics3d Object

1.4 Documentation

1.4.1 SageMath functions used:

1. `parametric_plot3d(xdata,ydata,options)`

Basically, `parametric_plot3d` draws a polygon by joining the points given in `xdata` and `ydata`

2. `show(options)`

`show()` is called on a plot object to display it. Here, we have used the `aspect_ratio` option to display lengths along all the axes on a 1:1:1 aspect ratio.

3. `exp(t)` This returns e^t , where e is the Euler constant $= 2.718$.

4. `cos(t)` This returns the cosine of t assuming t is expressed in radians.

5. `sin(t)` This returns the sine of t assuming t is expressed in radians.

1.4.2 Pure Python functions used:

1. `lambda t: f(t)`

This basically lets us create a nameless function without the use of the `def` keyword. It is meant to be used for use once, throw away functions.