2022 Differential Equation Python Quiz 10 (Python)

Consider the spring model

$$y''(t) + 10y'(t) + 196y = 0, \qquad y(0) = 0.2, y'(0) = 0 \tag{1}$$

In this quiz we will try to solve this using both numerical and symbolic solutions of linear systems.
If you solve this using conventional 2nd order solutions you will receive 0.

1) Find(by hand) the 2x2 system of linear ODEs,such as (2) that represents the equation(1), then form a **symbolic** equivalent in sympy and *display()* it.(Hints show how to display it) 20pts

$$\begin{aligned}(x^{(1)})' &= \alpha_{1,2}x^{(2)} + \alpha_{1,1}x^{(1)} \\ (x^{(2)})' &= \alpha_{2,2}x^{(2)} + \alpha_{2,1}x^{(1)}\end{aligned} \tag{2}$$

   Where $\alpha_{i,j}$ represents the coefficient you chose(in exercise 1) for ith differential equation and jth term

2) Find the Euler approximations of y and y' in (1) with $t \in [0, 5]$ and step $h = 0.001$ as shown below (3) 30pts

$$\begin{aligned}x_n^{(1)} &= x_{n-1}^{(1)} + h(\alpha_{1,2}x_{n-1}^{(2)} + \alpha_{1,1}x_{n-1}^{(1)}) \\ x_n^{(2)} &= x_{n-1}^{(2)} + h(\alpha_{2,2}x_{n-1}^{(2)} + \alpha_{2,1}x_{n-1}^{(1)})\end{aligned} \tag{3}$$

   Where $\alpha_{i,j}$ represents the coefficient for ith differential equation and jth term.

3) Find the analytical/exact solution of IVP (2) using *dsolve_system()*.(If you use other method outside `dsolve_system` automatic 0) 20pts
   For this you should use the symbolic system of equation you created in part 1. Dont forget your initial values $y(0) = 0.2, y'(0) = 0$

4) Plot both the Euler approximation and the analytical solution of y and y' (one graph for y and another for y'). 15pts. See hints pdf for the graphs.
   For the graphs use:

```
plt.grid(True)
plt.axvline(x=0)
plt.axhline(y=0)
plt.xlim(0,1.5)
plt.ylim(-0.20,0.2)

```

5) Calculate distance as below then print it 15pts

$$d = \sum_{i=1}^{i=n} \sqrt{(\tilde{x}_i^{(1)} - x_i^{(1)})^2 + (\tilde{x}_i^{(2)} - x_i^{(2)})^2} \tag{4}$$

   Where $\tilde{x}_i^{(1)}$ is the value of the exact solution $\tilde{x}^{(1)}$ at time step $t = i$ and $x_i^{(1)}$ is the value of the approximation x^i at time step i. Similarly for $\tilde{x}^{(2)}$. n is the number of timesteps in your time vector.