Universidade Federal do ABC

Simulation Exercise #02: Coupled Differential Equations and Resolution Methods

INF301 – Systems Modeling and Simulation – W02

Prof. Luiz Henrique Bonani do Nascimento
Universidade Federal do ABC

Report Guidelines

Use a notebook on the Google Collaboratory platform to generate a report containing any explanations and comments you deem relevant, along with your codes and figures.

The graphs in the figures should be self-explanatory, with axis names and data captions. Use a font size appropriate for presentation in a document.

The language to be used is Python. However, the use of pre-built Python libraries is not permitted, except for those used in the examples.

Submit a single notebook file in ipynb format, with the file name in the format SE0N_NameSurname.ipynb, where N is the SE number, Name is your first name, and Surname is your last name.

Remember that plagiarism will not be tolerated under any circumstances!

UFABC

Problem 1 (70 points)

Consider the **initial value problem** given below.

$$\ddot{x} + 2\dot{x} + 5y = 3$$
$$\dot{x} + 2y = \dot{y}$$
$$x(0) = 0$$
$$\dot{x}(0) = 0$$
$$y(0) = 1$$

a) Rewrite the system as a system of first-order differential equations.

b) Show that the exact solution to this system is given by:

$$x(t) = 2\cos(t) + 6\sin(t) - 2 - 6t$$
$$y(t) = -2\cos(t) + 2\sin(t) + 3$$

UFABC

Problem 1 (continuation)

c) Approximate the solution in the interval [0, 10] using a time step of h = 0.1 and the Euler method.

d) Approximate the solution in the interval [0, 10] using a time step of h = 0.1 and the 4th-order Runge-Kutta method.

e) Compare and discuss the obtained results, providing the proper comments.

UFABC

Problem 2 (30 points)

Python has a routine called `odeint` (similar to Matlab's `ode45`) that can be imported from the `scipy.integrate` library to solve coupled differential equations using an iterative algorithm based on the fourth- and fifth-order Runge-Kutta methods.

Apply the `odeint` function to solve the system of differential equations in Problem 1 and compare the results. To use the function, see the example in Appendix 2.

Provide the proper comments.

UFABC

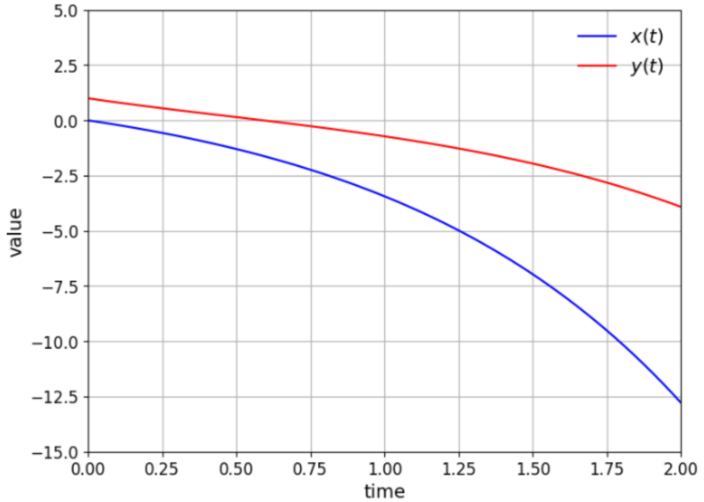
Appendix 1

To use the 4th Order Runge-Kutta method for a system of three differential equations, remember that:

$$
\begin{aligned}
&Kx_1 = f(t_k, x(k), y(k), z(k))\\
&Ky_1 = f(t_k, x(k), y(k), z(k))\\
&Kz_1 = f(t_k, x(k), y(k), z(k))\\
&Kx_2 = f(t_k + ½\,h, x(k) + ½\,h\,Kx_1, y(k) + ½\,h\,Ky_1, z(k) + ½\,h\,Kz_1)\\
&Ky_2 = f(t_k + ½\,h, x(k) + ½\,h\,Kx_1, y(k) + ½\,h\,Ky_1, z(k) + ½\,h\,Kz_1)\\
&Kz_2 = f(t_k + ½\,h, x(k) + ½\,h\,Kx_1, y(k) + ½\,h\,Ky_1, z(k) + ½\,h\,Kz_1)\\
&Kx_3 = f(t_k + ½\,h, x(k) + ½\,h\,Kx_2, y(k) + ½\,h\,Ky_2, z(k) + ½\,h\,Kz_2)\\
&Ky_3 = f(t_k + ½\,h, x(k) + ½\,h\,Kx_2, y(k) + ½\,h\,Ky_2, z(k) + ½\,h\,Kz_2)\\
&Kz_3 = f(t_k + ½\,h, x(k) + ½\,h\,Kx_2, y(k) + ½\,h\,Ky_2, z(k) + ½\,h\,Kz_2)\\
&Kx_4 = f(t_k + h, x(k) + h\,Kx_3, y(k) + h\,Ky_3, z(k) + h\,Kz_3)\\
&Ky_4 = f(t_k + h, x(k) + h\,Kx_3, y(k) + h\,Ky_3, z(k) + h\,Kz_3)\\
&Kz_4 = f(t_k + h, x(k) + h\,Kx_3, y(k) + h\,Ky_3, z(k) + h\,Kz_3)\\
&x(k+1) = x(k) + 1/6\,h\,(Kx_1 + 2\,Kx_2 + 2\,Kx_3 + Kx_4)\\
&y(k+1) = y(k) + 1/6\,h\,(Ky_1 + 2\,Ky_2 + 2\,Ky_3 + Ky_4)\\
&z(k+1) = z(k) + 1/6\,h\,(Kz_1 + 2\,Kz_2 + 2\,Kz_3 + Kz_4)
\end{aligned}
$$

UFABC

Appendix 2



Example: Solving the differential equation

$$\dot{x} = x - 2$$
$$\dot{y} = x - 2y$$
$$x(0) = 0$$
$$y(0) = 1$$

To do this, a function called `equt()` is created to describe the equation:

```
def equt(X, t):
  x, y = X # X is an array, so X[0] = x and X[1] = y
  dXdt = [x - 2, x - 2*y] #dx/dt = x - 2 and dy/dt = x - 2y
  return dXdt
```

To obtain the solution in the interval [0,2] with the initial conditions above, the `odeint()` function must be called:

```
sol = odeint(equt, [0, 1], np.linspace(0, 2, 101))
```

UFABC