**Code Example 1: Solve initial value problem (IVP) of a linear ODE**

**Code:** example\_1.py

**Code description:** This Python script solves and visualizes the initial value problem (IVP) of a simple exponential growth ordinary differential equation (ODE) using Euler’s method. The implementation is done using PyTorch for numerical computation and Matplotlib for visualization.

**torch**: The PyTorch library is used for performing tensor operations, including differentiable calculations. It's mainly used here for handling numerical operations related to the ODE.

**matplotlib.pyplot**: This library is used for plotting the results, specifically a time-series plot of the water levels in two tanks.

A math problem with equations

Description automatically generated with medium confidence

**Code Example 2: Solve initial value problem (IVP) of a non-linear ODE**

**Code:** example\_2.py

**Code description:** The code provided is a simulation for solving a nonlinear ordinary differential equation (ODE) representing a two-tank system using the Euler method. Here's a breakdown of the code:

A screenshot of a math problem

Description automatically generated

A math problem with equations

Description automatically generated with medium confidence

**Code Example 3: Linearize a non-linear ODE**

**Code:** example\_3.py, example\_3a.py

**Code description:**

To solve the system using the matrix exponential, we need to express the system in a linearized state-space form and then apply the matrix exponential solution. We linearize the system by constructing the Jacobia matrix around the operating point.

The **matrix exponential** solution to a linearized ODE system is given as:

A white paper with black text

Description automatically generated

A screenshot of a math problem

Description automatically generated

A screenshot of a math problem

Description automatically generated

A white sheet with black text

Description automatically generated with medium confidence

**Code Example 4: Stability analysis of a linear state space model**

**Code:** example\_4.py

**Code description**

A white page with black text

Description automatically generated

1. **Stable System**:
   * The state matrix A has negative eigenvalues (−1.0 and −0.5), so the system's states decay to zero over time. This is reflected in the plot as the trajectory converging to the origin.
2. **Unstable System**:
   * The state matrix A has positive eigenvalues (1.0 and 2.0), so the system's states grow exponentially, which is shown in the plot where the trajectory diverges from the origin.
3. **Limit Cycle (Oscillatory System)**:
   * The state matrix A has purely imaginary eigenvalues (i and −i), which leads to oscillatory motion. The trajectory in the phase space forms a closed loop, showing a limit cycle.

**Summary:**

* The **stable system** decays to the origin, showing a convergence.
* The **unstable system** grows exponentially, showing divergence.
* The **limit cycle system** exhibits periodic behavior and forms a closed trajectory in the phase plane.