Lecture 4: Solving ODEs with Matlab and Simulink

Juan Paredes
jparedes@umich.edu

Ordinary Differential Equations (ODEs)

- Let $t \geq 0, x \in \mathbb{R}^n, \mathrm{u} \in \mathbb{R}^m, t_f > 0.$

- For all $t \in [0, t_f]$, obtain $x(t)$ given $x(0) \in \mathbb{R}^n, u(t)$ and

$$\frac{dx}{dt} = \dot{x} = f(t, x, u)$$

- We'll focus on linear, time invariant models. Hence, the equation above can be represented as

$$\dot{x} = A\,x(t) + B\,u(t),$$

where $\mathrm{A} \in \mathbb{R}^{n \times n}$ and $\mathrm{B} \in \mathbb{R}^{n \times m}$ are constant matrices. This is also referred to as the **State Space Representation** of the system.

- We'll learn how to numerically solve this sort of ODE using Matlab and Simulink

Matlab and Simulink functions for solving linear ODEs

Matlab	Simulink
• ode45 • lsim	• State space block • Integrator block + Matlab function block

Example: Linear ODE

- Let

$$\dot{x} = -x + u,$$

where $x(0) = 2$ and

$$u(t) = \begin{cases} 5, & t \geq 1\ s \\ 0, & else \end{cases}.$$

- For all $t \in \{0, 0.01, \ldots, 9.99, 10\}$ s, obtain $x(t)$.
- Suppose $y = x$ and, thus, $A = -1, B = 1, C = 1, D = 0$.

Before proceeding…

- State Space Representation and System Output
- Numerical solutions and time intervals
- Fixed-step vs Variable-step numerical solvers
- Matlab function handles (for ode45)
- Matlab dynamic system models (for lsim)
- State space vs Transfer function for continuous time systems

State Space Representation and System Output

- In the state space representation of a system, x represents the state vector and u represents the input vector.

- Variable $y \in \mathbb{R}^p$ represents the output vector and the actual output of the system. Hence, the full system is represented by

$$\dot{x} = A\,x(t) + B\,u(t),$$
$$y = C\,x(t) + D\,u(t),$$

where $\mathrm{C} \in \mathbb{R}^{p\times n}$ and $\mathrm{D} \in \mathbb{R}^{p\times m}$ are constant matrices.

- Since the second equation is algebraic, we can first solve for x using the first equation and then use the second equation to obtain y (remember that u is given).

- There are Matlab and Simulink functions that do both simultaneously (lsim and the State Space block).

Numerical solutions and time intervals

- We mentioned that we would obtain $x(t)$ for all $t \in [0, t_f]$.
- We'll focus on obtaining numerical solutions.
- Due to the restrictions inherent to numerical solutions, we'll obtain solutions at the start and end points of time intervals within $[0, t_f]$.
- For example, we can obtain $x(t)$ for all $t \in \{0, t_s, 2t_s, \dots, t_f - t_s, t_f\}$, where $t_s > 0$ is the fixed size of the considered time intervals (step size).
- t_s can be made very small to increase the smoothness of the obtained plot, although this may increase the computational load (more simulation time).

Fixed-step vs Variable-step numerical solvers

- Fixed-step solvers keep the step size constant throughout the simulation.

- Variable-step solvers change the step size during the simulation depending on how fast the system states change (may lead to less time steps and less computation time).

- ode45 is a variable-step solver but can be supplied with a discrete time vector at which you require the states (“behaves” like a fixed-step solver).

- Will use fixed-step solver for Simulink.

- For sufficiently small step sizes, fixed and variable step solvers offer similar degrees of accuracy (may take more time).

Matlab function handles (for ode45)

- Function handles are used in Matlab to represent functions. For example:

$$f = @(x, y)\ x^2 + y^2$$

  Then, $f(2,1)$ yields 5.

- They can be used to make certain inputs constant. For example:

$$g = @(x)\ f(x, 2)$$

  Then, $g(2)$ yields 8.

- A function handle can also be used to pass a function to another function.

Matlab dynamic system models (for lsim)

- Dynamic system models are used in Matlab to represent systems that have internal dynamics or have memory from past states.
- Commands that yield dynamic system models include tf (for transfer functions) and ss(for state space models).
- Most Matlab functions that analyze linear systems use dynamic system models, including lsim.
- For example, $ssSym = ss(A, B, C, D)$ creates a dynamic system model $ssSym$ that represents a system described by a state space model with matrices A, B, C, D.

State space vs Transfer function for continuous time systems

- State space representations and transfer functions can be used to represent the same time-invariant linear system.
- Assuming $t \geq 0$, if $x(0) = 0$, then for a given input u, the same output y can be obtained via both representations.
- We'll mostly use State Space since it is easier to set initial conditions in this manner.

State Space

$$\dot{x} = A\,x(t) + B\,u(t),$$
$$\mathrm{y} = C\,x(t) + D\,u(t),$$
$$x(0) = 0$$

$$\equiv$$

(simulation-wise)

Transfer Function

$$\frac{Y(s)}{U(s)} = G(s) = C(sI - A)^{-1}B + D$$

Example linear ODE (reminder)

- Let

$$\dot{x} = -x + u,$$

where $x(0) = 2$ and

$$u(t) = \begin{cases} 5, & t \geq 1\ s \\ 0, & else \end{cases}.$$

- For all $t \in \{0, 0.01, \dots, 9.99, 10\}$ s, obtain $x(t)$.
- Suppose $\mathrm{y} = x$ and, thus, $A = -1, B = 1, C = 1, D = 0$.