Space Weather Summer School
DAY 6

July 25th, 2022, 9:00 am to 5:00 pm

Simone Servadio

MIT AEROASTRO

MIT | Astrodynamics, Space Robotics, and Controls Laboratory

Baseball tickets secured!!!

- The game starts at 6:40 pm
- I will send an email with the exact location where we meet.
- It is convenient to go drop your backpack and then head directly to the bus stop.
- Information about the bus is in the shared folder.
- Final ticket price is 20.50$. You can pay me either Venmo or Cash.

@SimoneServadio

Finite Differences

- What is the definition of derivative?

Finite Differences

- What is the definition of derivative?

$$\lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

Finite Differences

- What is the definition of derivative?

$$\lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

- We can approach from both sides, thus left and right hand derivative.

$$\boxed{\text{LHD}\,(\text{at } x = a) = \lim_{h \to 0} \frac{f(a-h) - f(a)}{-h}}$$

$$\boxed{\text{RHD}\,(\text{at } x = a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}}$$

Finite Differences

- What is the definition of derivative?

$$\lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

- We can approach from both sides, thus left and right hand derivative.

$$\boxed{\mathrm{LHD}\,(\text{at } x = a) = \lim_{h \to 0} \frac{f(a-h) - f(a)}{-h}}$$

$$\boxed{\mathrm{RHD}\,(\text{at } x = a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}}$$

- How do we translate this in a programming environment?

Finite Differences

- Using numpy, plot the following functions:

$$f(x) = \cos(x) + x\sin(x) \qquad f'(x) = x\cos(x) \qquad x \in [-6, 6]$$

Finite Differences

- Using numpy, plot the following functions:

$$f(x) = \cos(x) + x\sin(x) \qquad f'(x) = x\cos(x) \qquad x \in [-6, 6]$$

- Evaluate the derivative numerically using finite differences, h = 0.25:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h).$$

Forward

Initialization

while Condition

Body

Hint: use numpy.append(arr1,arr2)

Finite Differences

- Using numpy, plot the following functions:

$$f(x) = \cos(x) + x\sin(x) \qquad f'(x) = x\cos(x) \qquad x \in [-6, 6]$$

- Evaluate the derivative numerically using finite differences, h = 0.25:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h).$$

Forward

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h),$$

Backward

Initialization

while Condition

Body

Finite Differences

- Using numpy, plot the following functions:

$$f(x) = \cos(x) + x\sin(x) \qquad f'(x) = x\cos(x) \qquad x \in [-6, 6]$$

- Evaluate the derivative numerically using finite differences, h = 0.25:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h).$$

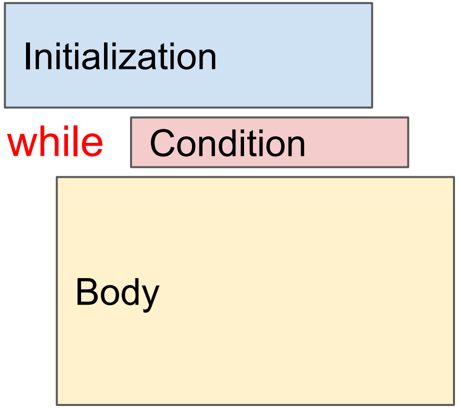
Forward

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h),$$

Backward

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2),$$

Central



Finite Differences

- Using numpy, plot the following functions:

$$f(x) = \cos(x) + x\sin(x) \qquad f'(x) = x\cos(x) \qquad x \in [-6, 6]$$

- Evaluate the derivative numerically using finite differences, h = 0.25:

$$f'(x) = \frac{f(x+h) - f(x)}{} \ldots$$

Forward

REVIEW EACH OTHER’S CODE

...on

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h),$$

Backward

Body

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2),$$

Central

Integration

- We can invert the forward finite difference formula to solve Ordinary Differential Equations (ODEs). Consider the problem:

$$\frac{dy(t)}{dt} = y'(t) = f(y(t), t), \qquad \text{with } y(t_0) = y_0$$

$$\frac{dy(t)}{dt} + 2y(t) = 0 \quad \text{or} \quad \frac{dy(t)}{dt} = -2y(t)$$

- Solve the ODE using the Euler Method (Runge-Kutta 1st Order) with a step-size of h = 0.2:

$$k_1 = f(y^*(\mathrm{t}_0), t_0) \qquad \text{approximation for derivative}$$

$$y^*(t_0 + h) = y^*(t_0) + k_1 h \qquad \text{approximate solution at next time step}$$

Integration

- Solve the same ODE with Runge-Kutta 2nd order:

$$\frac{dy(t)}{dt} = f(y(t), t)$$

$k_1 = f(y^*(t_0),\ t_0)$	estimate of derivative at $t = t_0$
$y_1\left(t_0 + \frac{h}{2}\right) = y^*(t_0) + k_1 \frac{h}{2}$	intermediate estimate of function at $t = t_0 + \frac{h}{2}$
$k_2 = f\left(y_1\left(t_0 + \frac{h}{2}\right),\ t_0 + \frac{h}{2}\right)$	estimate of slope at $t = t_0 + \frac{h}{2}$
$y^*\left(t_0 + h\right) = y^*(t_0) + k_2 h$	estimate of $y\left(t_0 + h\right)$

Integration

- Solve the same ODE with Runge-Kutta 4th order:

$$k_1 = f(y^*(t_0), t_0)$$

$$k_2 = f\left(y^*(t_0) + k_1\frac{h}{2}, t_0 + \frac{h}{2}\right)$$

$$k_3 = f\left(y^*(t_0) + k_2\frac{h}{2}, t_0 + \frac{h}{2}\right)$$

$$k_4 = f\left(y^*(t_0) + k_3 h, t_0 + h\right)$$

$$y^*(t_0 + h) = y^*(t_0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}h = y^*(t_0) + \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)h$$

$$= y^*(t_0) + mh \qquad \text{where } m \text{ is a weighted average slope approximation}$$

Integration

- Solve the same ODE with Runge-Kutta 2nd order:

$$f(t_{k+1}, x_{k+1}) = f(t_k, x_k) + hf'(t_k + \frac{h}{2}, x_k + \frac{h}{2} f'(t_k, x_k))$$

- Solve the same ODE with Runge-Kutta 4th order:

$$k_1 = \quad \ldots \quad \frac{h}{2}, t_0 + \frac{h}{2}\Big)$$

$$k_2 = \quad \ldots \quad , t_0 + h)$$

REVIEW EACH OTHER'S CODE

$$y^*(t_0 + h) = y^*(t_0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} h = y^*(t_0) + \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right) h$$

$$= y^*(t_0) + mh \qquad \text{where } m \text{ is a weighted average slope approximation}$$

The nonlinear pendulum: free, damped, controlled

- Consider now the nonlinear pendulum:

$$\ddot{\theta} + \frac{g}{l}\sin\theta = 0$$

- Solve the system using *odeint* and display the solution with two subplots.

The nonlinear pendulum: free, damped, controlled

- Consider now the nonlinear pendulum:

$$\ddot{\theta} + \frac{g}{l}\sin\theta = 0$$

- Solve the system using *odeint* and display the solution with two subplots.
- Code the Runge-Kutta 4th order function that is called like odeint and compare that you obtain the correct solution.

The nonlinear pendulum: free, damped, controlled

- Consider now the nonlinear pendulum:

$$\ddot{\theta} + \frac{g}{l}\sin\theta = 0$$

- Solve the system using *odeint* and display the solution with two subplots.
- Code the Runge-Kutta 4th order function that is called like odeint and compare that you obtain the correct solution.
- Repeat the process adding a damping coefficient of 0.3

The nonlinear pendulum: free, damped, controlled

- Add a PD controller to the pendulum with Kd = 2 and Kp = 2.

- Create a new ODE that includes control torque, asked as additional input.

- Considering a sensor that works at 100 Hz, and an actuators that gives constant torque. Bring the (free) pendulum to the stable equilibrium state.
    1. Propagate the state between sensors reading (use *odeint*)
    2. Evaluate the control at the current time step.
    3. Feed the control in the propagation of the next step

Hint: odeint(function,initial_state,time_vector,arguments=(arg1,arg2,))
np.vstack([arr1,arr2]) to stack two arrays

PD controller:
u = -Kp * x_pos - Kd * x_vel

The nonlinear pendulum: free, damped, controlled

- Add a PD controller to the pendulum with Kd = 2 and Kp = 2.
- Create a new ODE that includes control torque.
- Considering a sensor that works at 100 Hz, bring the (free) pendulum to the stable equilibrium state.
- Modify the gains to Kd = 5 and Kp = 10. Modify the code such that the control torque brings the pendulum to unstable equilibrium.

The nonlinear pendulum: free, damped, controlled

- Add a PD controller to the pendulum with Kd = 2 and Kp = 2.
- Create a new ODE that includes control torque.
- Considering a sensor that works at 100 Hz, bring the (free) pendulum to the stable equilibrium state.
- Modify the gains to Kd = 5 and Kp = 10. Modify the code such that the control torque brings the pendulum to unstable equilibrium.
- At the previous condition, at time 7.5 seconds, make the pendulum do one loop.

The nonlinear pendulum: free, damped, controlled

- Add a PD controller to the pendulum with Kd = 2 and Kp = 2.
- Create a new ODE that includes control torque.
- Considering a sensor that works at 100 Hz, bring the (free) pendulum to the stable equilibrium state.

REVIEW EACH OTHER’S CODE

- Modify t … that the control torque brings the pendulum to unstable equilibrium.
- At the previous condition, at time 7.5 seconds, make the pendulum do one loop.

Lorenz 63 system

- The Lorenz63 system is a simplified mathematical model for atmospheric convection.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

"The equations relate the properties of a two-dimensional fluid layer uniformly warmed from below and cooled from above. In particular, the equations describe the rate of change of three quantities with respect to time: x is proportional to the rate of convection, y to the horizontal temperature variation, and z to the vertical temperature variation. The constants σ, ρ, and β are system parameters proportional to the Prandtl number, Rayleigh number, and certain physical dimensions of the layer itself."

- Assignment 1: Propagate the system for 20 seconds from x0 = [5,5,5] and 3D plot the results. Use σ=10, ρ=28, and β=8/3 (they must be asked as additional input in the ODE function.

- Assignment 2: Randomly select, uniformly, 20 initial conditions in range
x = [-20,20], y = [-30,30], z = [0,50] and plot their solution in the same plot.
What do you notice?

Lorenz 63 system

- The Lorenz63 system is a simplified mathematical model for atmospheric convection.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \ldots$$

"The equations relate the properties of a two-dimensional fluid layer uniformly warmed from below and cooled from above. In particular, the equations describe the rate of change of three quantities with respect to time: x is proportional to the rate of convection, y to the horizontal temperature variation, and z to the vertical temperature variation. The constants σ, ρ, and β are system parameters proportional to the Prandtl number, Rayleigh number,

REVIEW EACH OTHER'S CODE

- Assignment 1: Propagate the system for 20 seconds from x0 = [5,5,5] and 3D plot the results. Use σ=10, ρ=28, and β=8/3 (they must be asked as additional input in the ODE function.

- Assignment 2: Randomly select, uniformly, 20 initial conditions in range x = [-20,20], y = [-30,30], z = [0,50] and plot their solution in the same plot. What do you notice?

Assignment: Predator vs. Prey System

- Load the hare and lynx population data in the .csv file.
- Plot the population of prey and predator as a function of time, and then plot the population from the data in a plot that has prey in the y axis and predator in the x axis.
- Code the Lotka-Volterra equations, where the constants of the system are additional arguments.
- In the predator_vs_prey figure, propagate in grey each point form the data file for the time length given by the "Year" data, using the coefficients below.
- Propagate the optimal initial condition in red: x0 = [34.9134, 3.8566] and plot it in both figures.
- Evaluate the average population of each species.
- Compare the average population from data and the time propagation of the optimal initial condition with the theoretical values (growth_coeff/shrinkage_coeff) of the other species.
- Plot the 3 average populations as a line in the time dependency figure, and in the predator/prey figure. What do you notice?

```
grow_hare = 0.48069
shrink_hare = 0.024822
grow_lynx = 0.92718
shrink_lynx = 0.027564
```

$$\frac{dx}{dt} = \alpha x - \beta x y,$$
$$\frac{dy}{dt} = \delta x y - \gamma y,$$

Announcements

- All the solutions will be posted after class.
- See you all at The Sink at 7:00 pm for the social event and dinner!
- Rockies colours are Silver - Purple - Black - White