CDS 230
Modeling and Simulation I

Module 8

Dynamical Systems

Dr. Hamdi Kavak
http://www.hamdikavak.com
hkavak@gmu.edu

GEORGE MASON UNIVERSITY

Center for Social Complexity

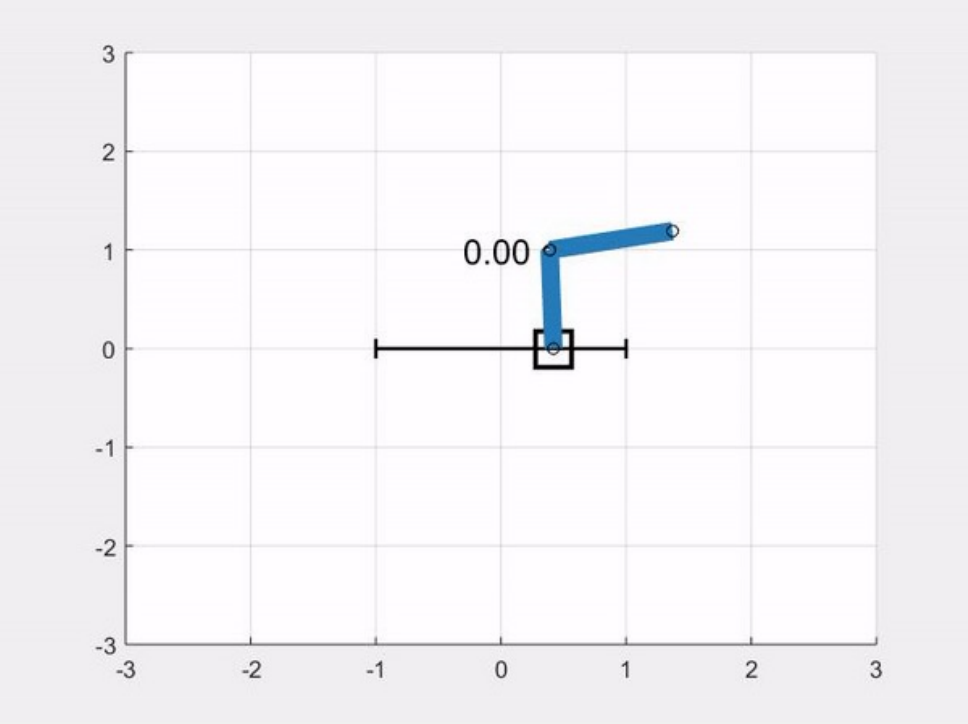
Basics

- **Dynamical system:** a physical or artificial system in which the system state ***changes over time*** (vs. static systems with ***no state changes***).
- **State:** a set of abstract attributes of interest that describe a system.

Example dynamical system	State
The weather	Temperature, precipitation, wind, pollution
Planetary motion	Location (x, y, z), velocity
Organism growth	Number of organisms (i.e., population)
Disease outbreak	?
Stock market	?
Traffic	?

How to study dynamical systems

- Using mathematical concepts, but how?
- Consists of two parts
  - **State space:** all possible state values
  - **Dynamics:** a rule that transforms the current state to the next state and so on. In other words, it is a **function** that maps current state to the next state.
- Once an initial system state ($t = 0$) is chosen, the dynamics can determine all future states ($t = 1, t = 2, \dots, t = n$).

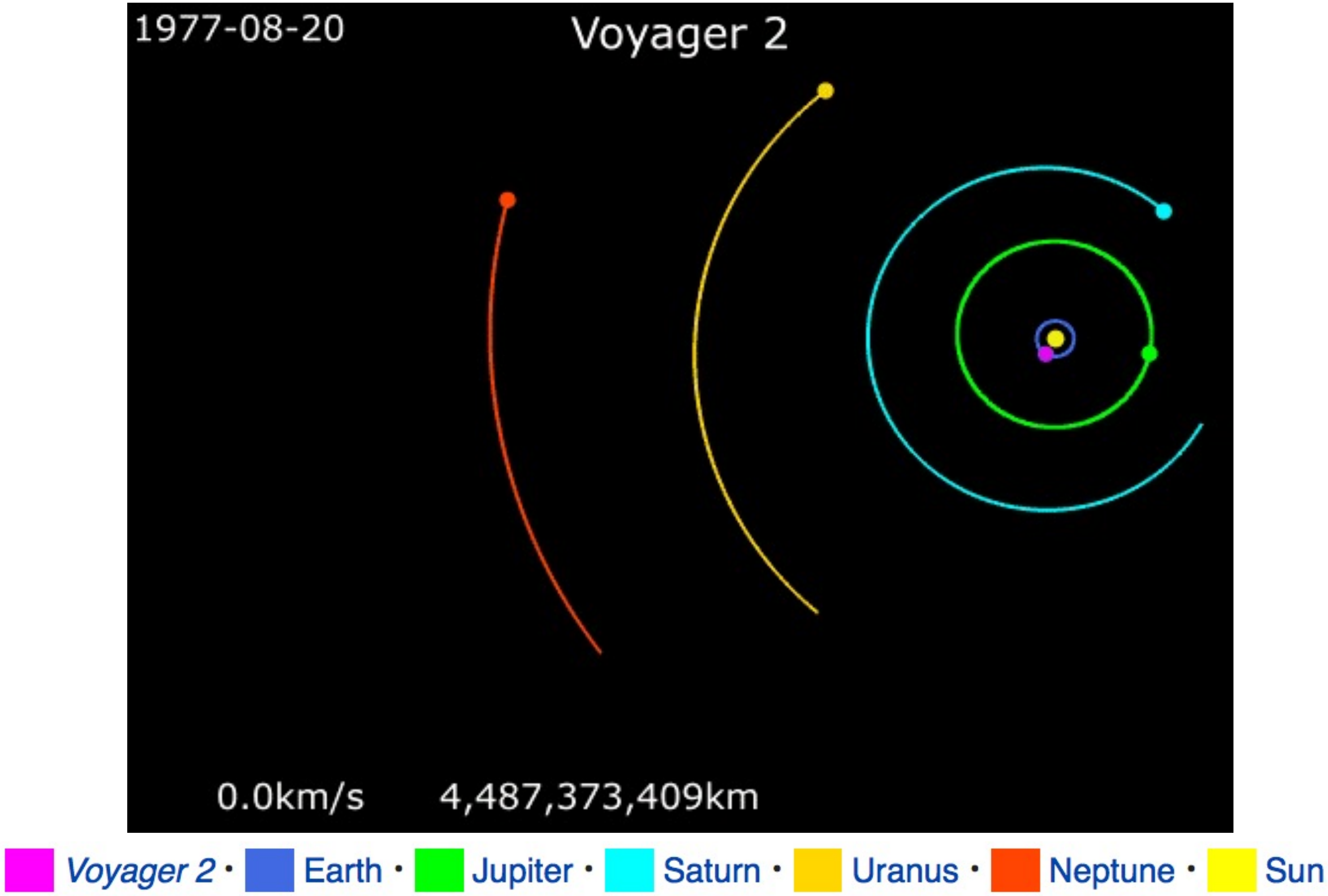


Source: https://openocl.org/tutorials/tutorial-01-modeling-double-cartpole/

Assumptions in dynamical systems

- Current state of the system exclusively determines the very next state of the system.
- The state transition from current to the next state is independent of the time it occurs.
- We sometimes don’t know the source of the changes on system states but we know that some dynamics (transition rule) exist. E.g.: gravity.

Gravity assist example



Source: https://en.wikipedia.org/wiki/Gravity_assist#/media/File:Animation_of_Voyager_2_trajectory.gif

Characteristics of dynamical systems

- Stationary vs. non-stationary
  - **Stationary:** have a stable state. Given two states of the system at different times (t_1 vs. t_2) that we cannot differentiate if $t_1 > t_2$ or $t_1 < t_2$.
    - E.g.: constant speed, no movement, periodic behavior
  - **Non-stationary:** have a non-stable state. Given two states of the system at different times (t_1 vs. t_2) that we can differentiate if $t_1 > t_2$ or $t_1 < t_2$.
    - E.g.: environment with friction, free fall
  - A system can become stationary or non-stationary at different times.
- Deterministic vs. Stochastic
  - **Deterministic:** same condition always leads to the same next state.
  - **Stochastic:** same conditions leads to different next states at different times.

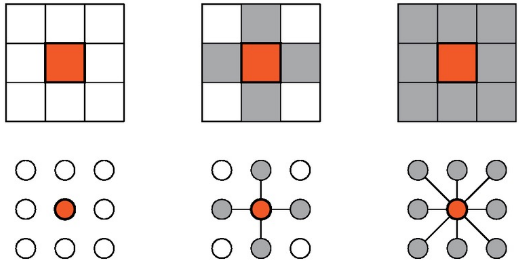
Math concepts to model dynamical systems

- Differential equations
  - Ordinary differential equations
    - mechanical systems, population growth, exponential decay,..
  - Partial differential equations
    - fluid dynamics, heat, diffusion, …
  - Higher order differential equations
    - wave motion, flow mechanics, electro-magnetic dynamics
- Difference equations
  - Population dynamics, spread of infectious diseases
- Cellular automata (automaton)
  - Growth of biological organisms, growth of crystals

$$y'(t) = f(t, y(t))$$

$$f(x,y,w)\frac{\partial w}{\partial x} + g(x,y,w)\frac{\partial w}{\partial x} = h(x,y,w)$$

$$y_{t+n} = a_1 y_{t+n-1} + \cdots + a_n y_t + b$$



Ordinary differential equations (ODEs)

- Informally, differential equations are used to solve how a quantity changes with respect to another variable.
  - E.g., how position changes with respect to time, how infections change over time
- ODE: Derivatives of **one** or **more** unknown functions with respect to **one** independent variable type.
  - First order examples

$$\frac{dy}{dx} = \cos x \qquad \frac{dx}{dt} - \frac{dy}{dt} = 2x + y$$

  - Second order example

$$\frac{d^2y}{dx^2} = \cos x$$

- Partial differential equations:

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0$$

Example – chaotic difference equations

- Write a Python script for the following equation .
  - $x_{n+1} = r\, x_n\, (1 - x_n)$
- Start with r=2, and $x_0 = 0.1$
- Compute for 100 iterations.
- Plot the results.
- What value does it settle on?

Example – chaotic difference equations

- If the value of r is 2.5 what is the final value of the graph?
- If the value of r is 2.8 what is the final value of the graph?
- Before trying it, predict the value for r = 3.
- r=3.3?
- r=3.5?
- r=4.0?

ODEs in Python

- ODEs mostly do **not** have closed form solutions.
- We approximate ODEs by solving them numerically.
- There is a useful Python package for this task: `scipy`
  - Install `scipy` using the Anaconda UI or conda command
  - `conda install scipy`
- SciPy (pronounced Sigh Pie) has a wide range of features in addition to solving ODEs.
  - Integration, optimization, interpolation, Fourier transforms, signal processing, statistics,…

Check if you successfully installed SciPy

```
from scipy import misc
import matplotlib.pyplot as plt

face = misc.face()
plt.imshow(face)
plt.show()
```

- What do you see?

Solving ODEs in Python

- We have the following first order ODE. Simulate y from $t = 0$ to 3.

$$\frac{dy}{dt} + y = t, \qquad y(0) = 1$$

We usually call this the initial condition.

- Let's rearrange the formula

$$\frac{dy}{dt} = t - y$$

- Define a Python function based on the rearranged formula

```
def dy_dt(y, t):
    return t - y
```

Solving ODEs in Python

$$\frac{dy}{dt} = t - y, \qquad y(0) = 1$$

- Import SciPy's integration submodule

```
import scipy.integrate as sci
```

- Set values

```
y0 = 1.0  # the initial condition
t = np.linspace( 0, 3, 30 )
```

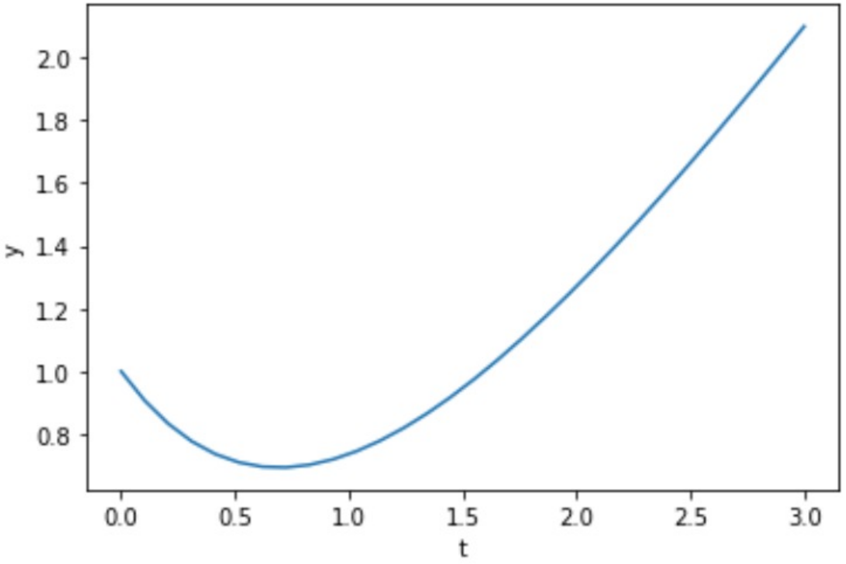
- Call the solver

```
y = sci.odeint(dy_dt, y0, t)
```

Solving ODEs in Python

- How to plot this result =>

```
plt.plot(t,y[:,0])
plt.xlabel("t")
plt.ylabel("y")
plt.show()
```




```
y
```

```
array([[1.        ],
       [0.90689362],
       [0.83310408],
       [0.77673302],
       [0.73606858],
       [0.70956715],
       [0.69583683],
       [0.69362253],
       [0.70179241],
       [0.71932603],
       [0.74530311],
       [0.77889385],
       [0.81935001],
       [0.86599686],
       [0.91822601],
       [0.97548883],
       [1.03729064],
       [1.10318535],
       [1.17277072],
       [1.24568405],
       [1.32159828],
       [1.40021848],
       [1.48127871],
       [1.56453918],
       [1.64978366],
       [1.73681716],
       [1.82546385],
       [1.91556522],
       [2.00697828],
       [2.09957414]])
```

Verifying that our solution is good enough

- Remember the ODE

$$\frac{dy}{dt} + y = t, \qquad y(0) = 1$$

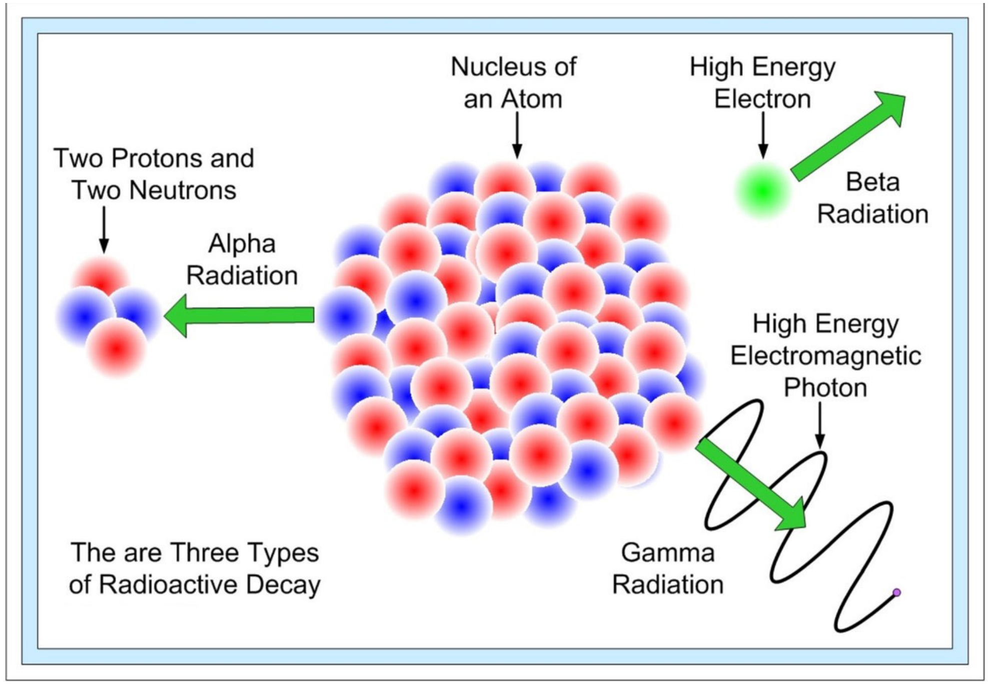
- The closed form solution for this ODE is as follows

$$y = t - 1 + 2e^{-t}$$

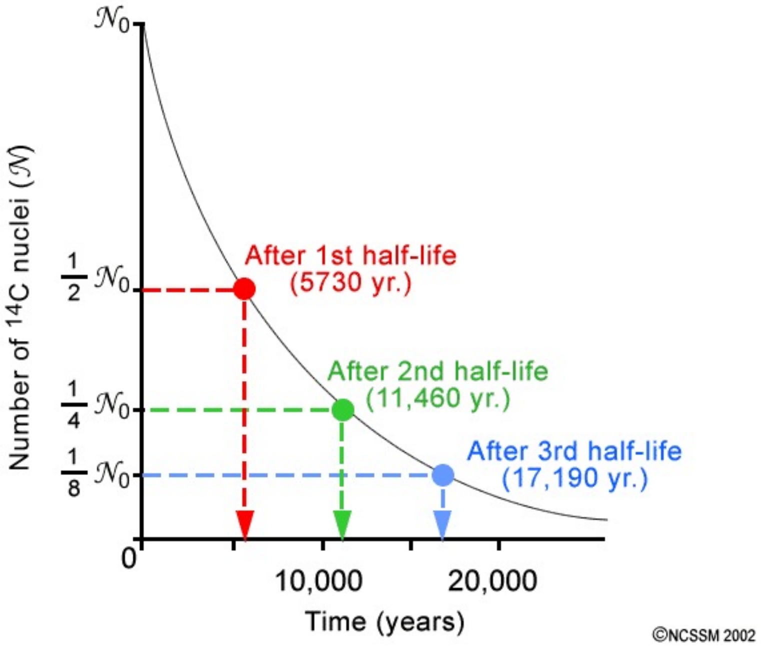
- Let's compare numerical vs. exact solution
  - Plot both on the same figure
  - Plot the difference

Radioactive decay

- “Radioactive decay takes place when an unstable atomic nucleus breaks up by emitting ionizing radiation.”



Source: https://www.toppr.com/guides/physics/nuclei/radioactivity-law-of-radioactive-decay/



Source: http://www.dlt.ncssm.edu/tiger/chem2.htm

Variables for radioactive decay model

- System states
  - N represents the amount of radioactive material.
- Other variables
  - λ represents the positive decay constant which is dependent on the type of the material.
- Dynamics

- (minus) sign indicates the decay

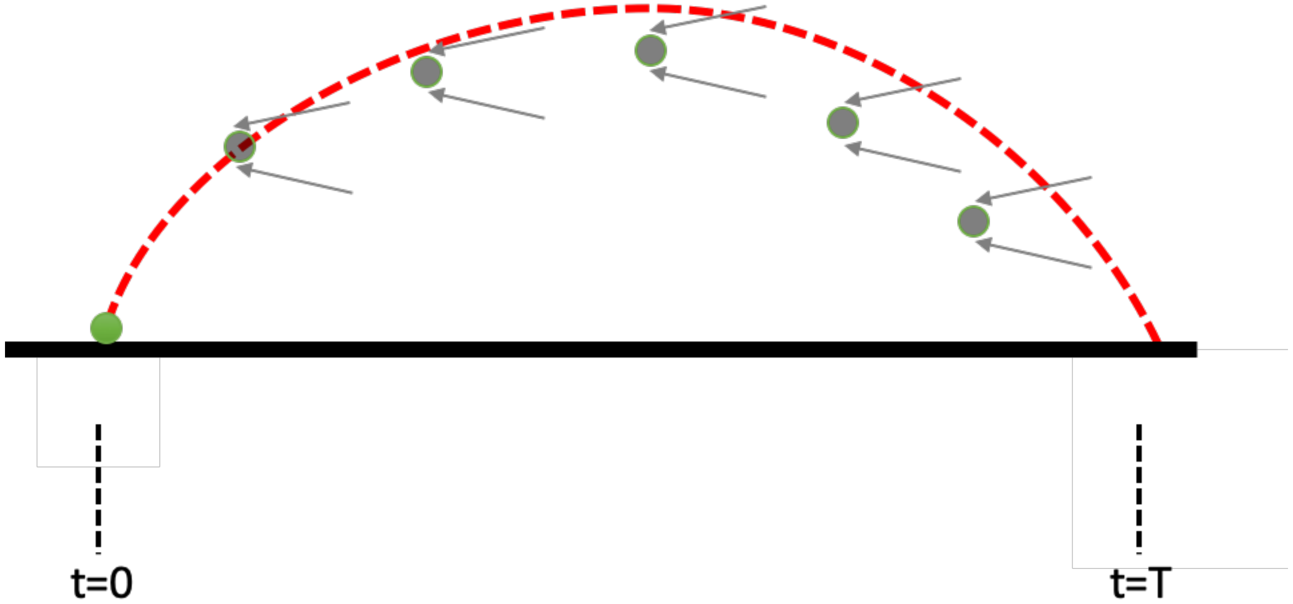
$$\frac{dN}{dt} = -\lambda N$$

Example radioactive decay question

- The mass of iodine isotope was 100g initially.
- Decay constant (λ)=0.0866 days.
- Find the decay of the material over 30 days.
- Note: $N(t) = N_0 e^{-\lambda t}$ is the closed form solution.

Let's solve this problem in Python

Model of projectile motion with air resistance



Variables for projectile motion model

- System states
  - Position (x, y)
  - Velocity (horizontal speed, vertical speed)
- Other variables
  - Mass
  - Air resistance coefficient
  - Gravity

Dynamics for projectile motion model

- $u = (x, y)$ - two dimensional position of our mass (m).
- Our mass is under the influence of two forces
  - Gravity mg or the vector of $(0, -9.81m)$.
  - Air resistance $F = -ku'$.
- Newton's Second Law of Motion (F=ma)
  - $F + mg = m \cdot u''$ can be rearranged by plugging $F = -ku'$ and leaving u'' alone.
    - $u'' = -\frac{k}{m}u' + g$ (second order ODE).
- Transform to a first-order ODE by plugging $\mathrm{v} = (u, u')$ thus $v' = (u', u'')$
  - $v' = (u', u'') = (u', -\frac{k}{m}u' + g)$ which can be represented as a function of v.

Example projectile motion question

- Object position (x, y): $(0, 0)$
- Velocity $(horizontal, vertical)$: $(10, 10)$
- Try different air resistance coefficients $k = 0.1$, $k = 0.5$, $k = 1.0$
- Mass (m): 1
- Project the position of the object over three seconds.

Let's solve this problem in Python

Sources

- Dynamical Systems Theory: What in the World is it? by Mike Hochman
- Dynamical Systems. James Meiss (2007), Scholarpedia, 2(2):1629.
- Figure in page 7: https://www.researchgate.net/figure/Cellular-Automata-grid-and-neighbours-presented-as-graphs-From-the-left-the-active-cell_fig7_258434100
- Österlund, Toni. (2013). DESIGN POSSIBILITIES OF EMERGENT ALGORITHMS FOR ADAPTIVE LIGHTING SYSTEM. Nordic Journal of Architectural Research. 25. 159-184.
- IPython Interactive Computing and Visualization Cookbook, Second Edition (2018), by Cyrille Rossant
- Sam Dolan's "Solving Ordinary Differential Equations"
- https://www.sciencedirect.com/topics/earth-and-planetary-sciences/radioactive-decay