Continuous Time Models

SYS-611: Simulation and Modeling

Paul T. Grogan, Ph.D.
Assistant Professor
School of Systems and Enterprises

Agenda

1. Continuous Time Simulation
2. Differential Equation Models
3. System Dynamics Models

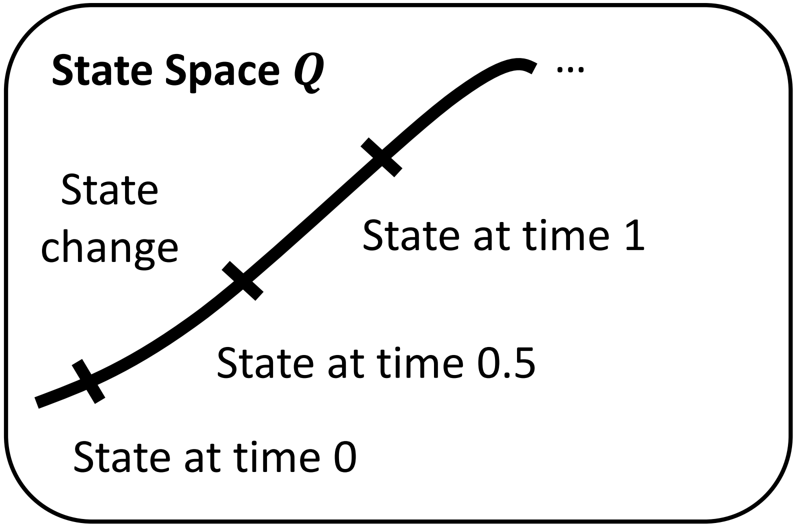
Reading: B.P. Zeigler, H. Praehofer, and T.G. Kim, “Modeling Formalisms and Their Simulators,” Ch. 3 in *Theory of Modeling and Simulation,* Academic Press, 2000, pp. 37-49.

H. Sayama, “Discrete-Time Models I: Modeling” Ch. 4 and “Cellular Automata I: Modeling,” Ch. 11 in *Introduction to Modeling and Analysis of Complex Systems*, Open SUNY Textbooks, 2015. (Free eBook online)

Continuous Time Simulation

Continuous Time Models

- Discrete or continuous state spaces (variables)
- **Dynamic**: time advances in continuous steps
  - Floating point/decimal units of some base
  - Step size can be refined with higher accuracy
- Applications:
  - Physical (electrical-mechanical) systems
  - Abstract systems



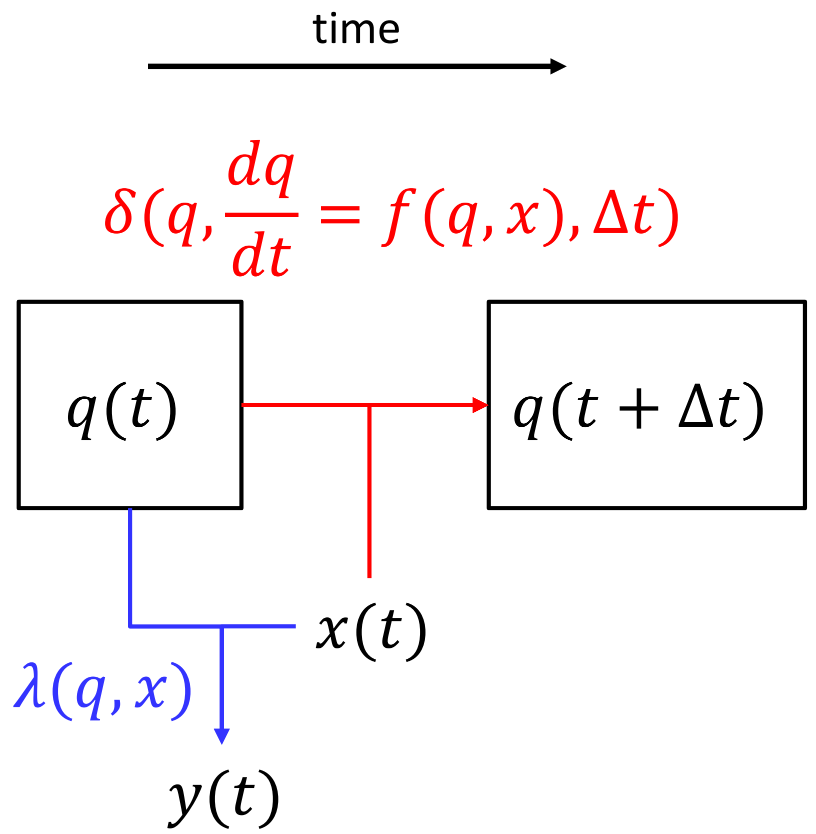
Continuous Time Notation

- $q(t)$: **state trajectory**, time history of states
- $x(t)$**: input trajectory**, time history of inputs
- $y(t)$**: output trajectory**, time history of outputs
- Next state determined by **state transition** function

$$\delta(q, \frac{dq}{dt}, \Delta t) = q(t+1)$$

- Outputs determined by **output function**

$$\lambda(q, x) = y(t)$$



Continuous Time Simulation

- Initialize time and state variables

$$t = 0, \qquad q(0) = q_0$$

- While terminal conditions not met:
  - Record output values

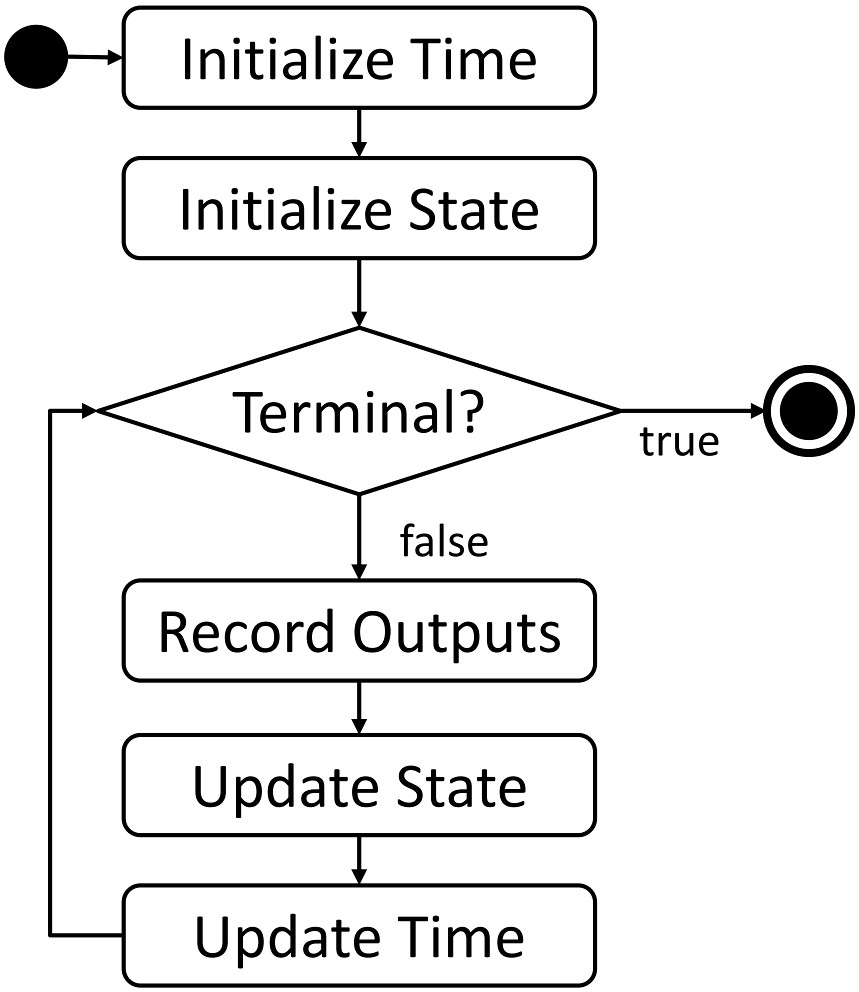
  $$y(t) = \lambda(y, x)$$

  - Compute next state

  $$q(t + \Delta t) = \delta(q, \frac{dq}{dt}, \Delta t)$$

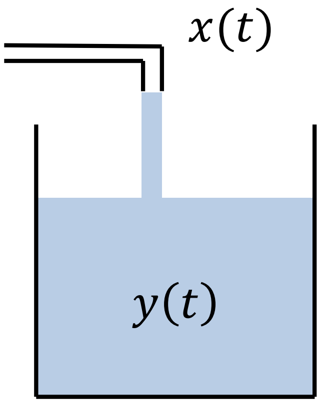
  - Increment time

  $$t = t + \Delta t$$



Differential Equation Models

Continuous Time Simulation



- Constant flow rate:

$$x(t) = 1 \text{ m}^3/\text{min}$$

$$q(0) = 5 \text{ m}^3$$

$$q(t) = 5 + \int_0^t 1 \cdot di = 5 + t$$

- Differential equations:
  - Volume: $y(t) = q(t)$
  - Flow: $\frac{dq}{dt} = x(t)$

$$y(t) = q(0) + \int_0^t x(i) \cdot di$$

- Linear flow rate:

$$x(t) = t \text{ m}^3/\text{min}$$

$$q(0) = 5 \text{ m}^3$$

$$q(t) = 5 + \int_0^t i \cdot di = 5 + \frac{t^2}{2}$$

Integration Methods

- Need a non-symbolic method to compute state transitions
- Approximate state transition function:

$$\delta\left(q, \frac{dq}{dt}, \Delta t\right), \text{where}$$

$$\frac{dq}{dt} = f(q, x)$$

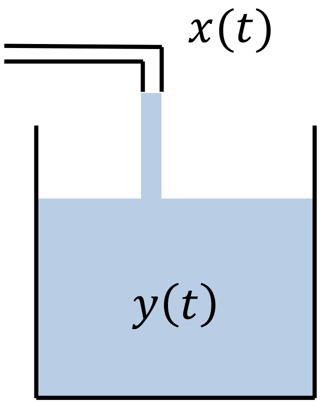
- Fundamental Theorem of Calculus:

$$\frac{dq}{dt} = \lim_{\Delta t \to 0} \frac{q(t + \Delta t) - q(t)}{\Delta t}$$

- Euler integration method:

$$q(t + \Delta t) = q(t) + \Delta t \frac{dq}{dt}$$

Euler Integration Example



$$q(0) = 5\ \mathrm{m}^3$$

$$y(t) = q(t)$$

$$\frac{dq}{dt} = x(t) = t$$

$$\Delta t = 1\ \mathrm{min}$$

$$q(t + \Delta t) = q(t) + \Delta t \frac{dq}{dt}$$

$$q(0) = 5$$

$$q(0 + 1) = 5 + 1 * 0 = 5$$

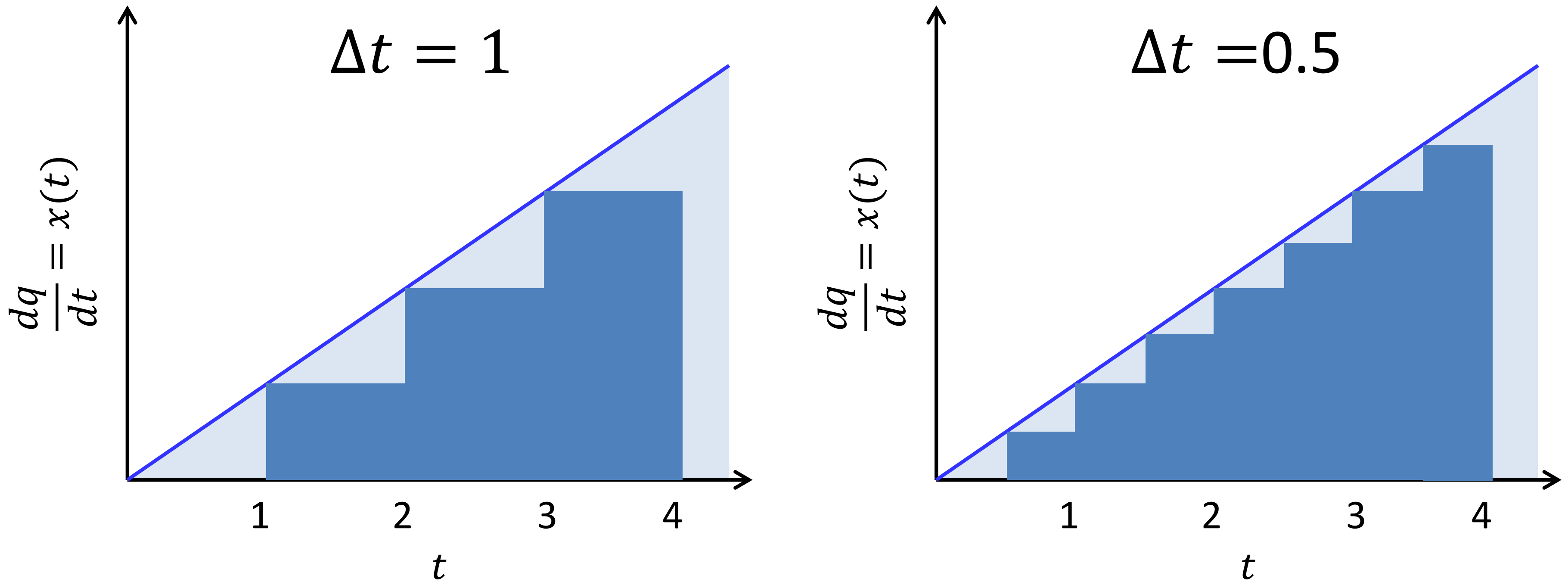
$$q(1 + 1) = 5 + 1 * 1 = 6$$

$$q(2 + 1) = 6 + 1 * 2 = 8$$

- Analytical solution:

$$q(t) = 5 + \frac{t^2}{2}$$

Euler Method Errors



Euler method has error from two sources:

1. Approximation of linear behavior
2. Mutual dependence of states and derivatives: $\frac{dq}{dt} = f(q, x)$

Euler Integration in Software

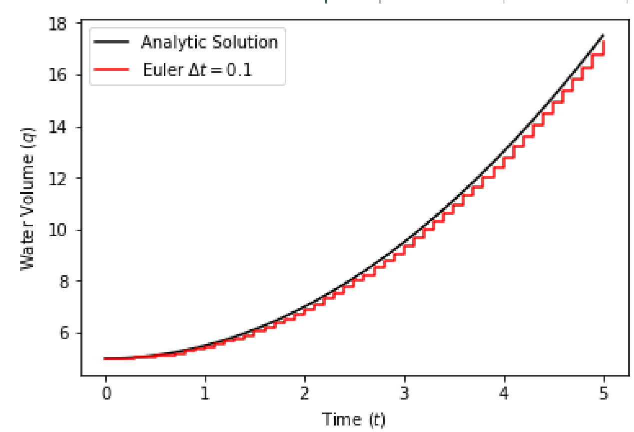
$$q(t+\Delta t) = q(t) + \Delta t \frac{dq}{dt}$$
$$= q(t) + \Delta t \cdot x(t)$$

```
def x(t):
  return t

delta_t = 0.1
num_steps = int(5.0/delta_t)
q = np.zeros(num_steps + 1)
t = np.zeros(num_steps + 1)

q[0] = 5.0
t[0] = 0.0
for i in range(num_steps):
  q[i+1] = q[i] + delta_t*x(t[i])
  t[i+1] = t[i] + delta_t
```

- In Excel:

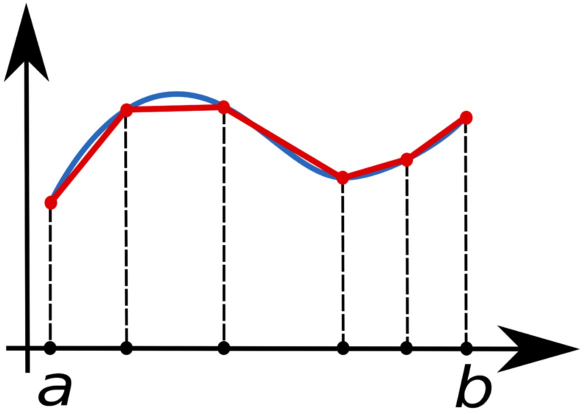
	A	B	C
1	delta_t	0.1	
2			
3	t	x	q
4	0	=A4	5
5	0.1	0.1	5.01
6	0.2	0.2	5.03
7	0.3	0.3	5.06
8	0.4	0.4	5.1
9	0.5	0.5	5.15
10	0.6	0.6	5.21

	A	B	C	D
1	delta_t	0.1		
2				
3	t	x	q	
4	0	0	5	
5	0.1	0.1	=C4+B4*B1	
6	0.2	0.2	5.01	
7	0.3	0.3	5.03	
8	0.4	0.4	5.06	
9	0.5	0.5	5.1	
10	0.6	0.6	5.15	

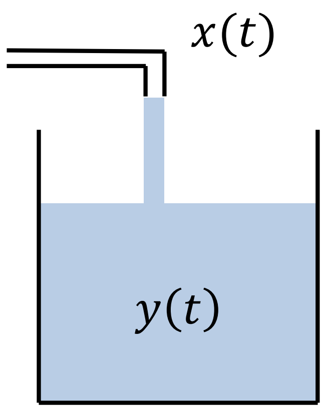


Numerical Integration in Python

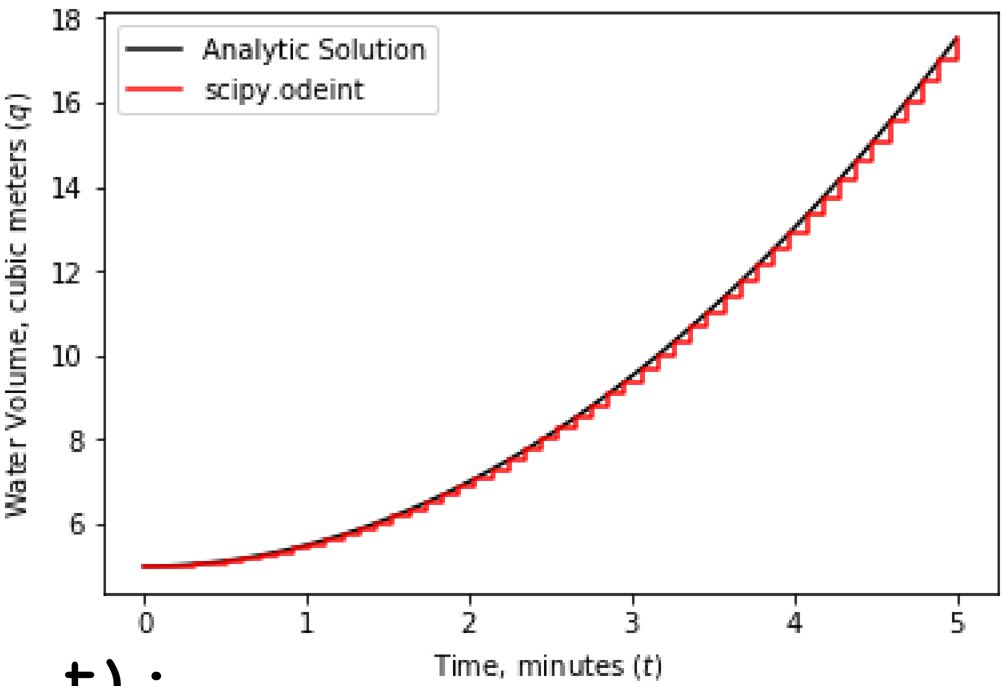
- **`scipy.integrate.quad`**:
  - Computes a definite integral for a callable function
- **`scipy.integrate.ode`** and **`scipy.integrate.odeint`**:
  - Computes integral for an ordinary differential equation with callable function and derivative (Jacobian)
  - Multiple integrators available
- **`numpy.trapz`**:
  - Evaluates integral for known discrete (y,x) points using a trapezoidal rule



Numerical Integration in Python



$$q(0) = 5\ \mathrm{m}^3$$
$$y(t) = q(t)$$
$$\frac{dq}{dt} = x(t) = t$$




```
def dq_dt(q, t):
    return t
t = np.linspace(0.0, 5.0)
q = integrate.odeint(dq_dt, 5.0, t)
```

System Dynamics Models

System Dynamics Models

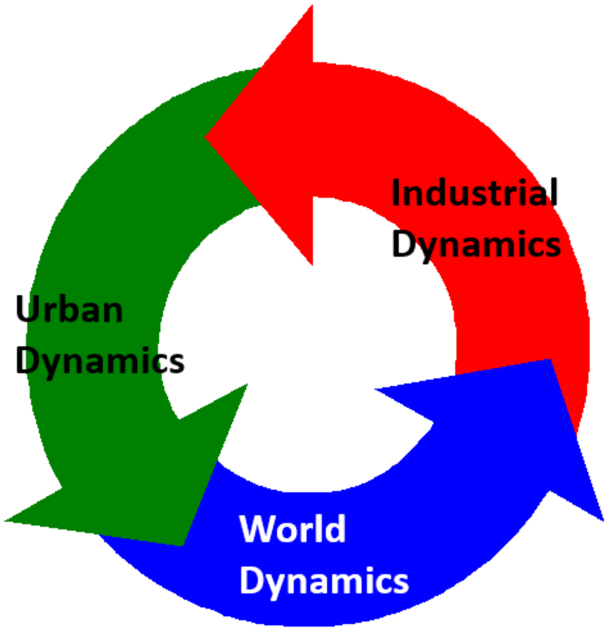
- **System Dynamics** (SD) models the dynamic behavior of complex systems over time
- Decomposes systems into components:
  - Stocks and flows
  - Feedback loops
  - Time delays
- Defines a system of differential equations in continuous-time simulation

Origins of System Dynamics

- 1940s-1950s: J.W. Forrester at MIT
  - Trained in electrical engineering
  - Dynamical systems and control theory
  - Developed Whirlwind digital computer
- 1958: Forrester left engineering for management
  - OR too limited in scope (operations, not strategy)
  - Dynamics and Controllability of Managed Systems

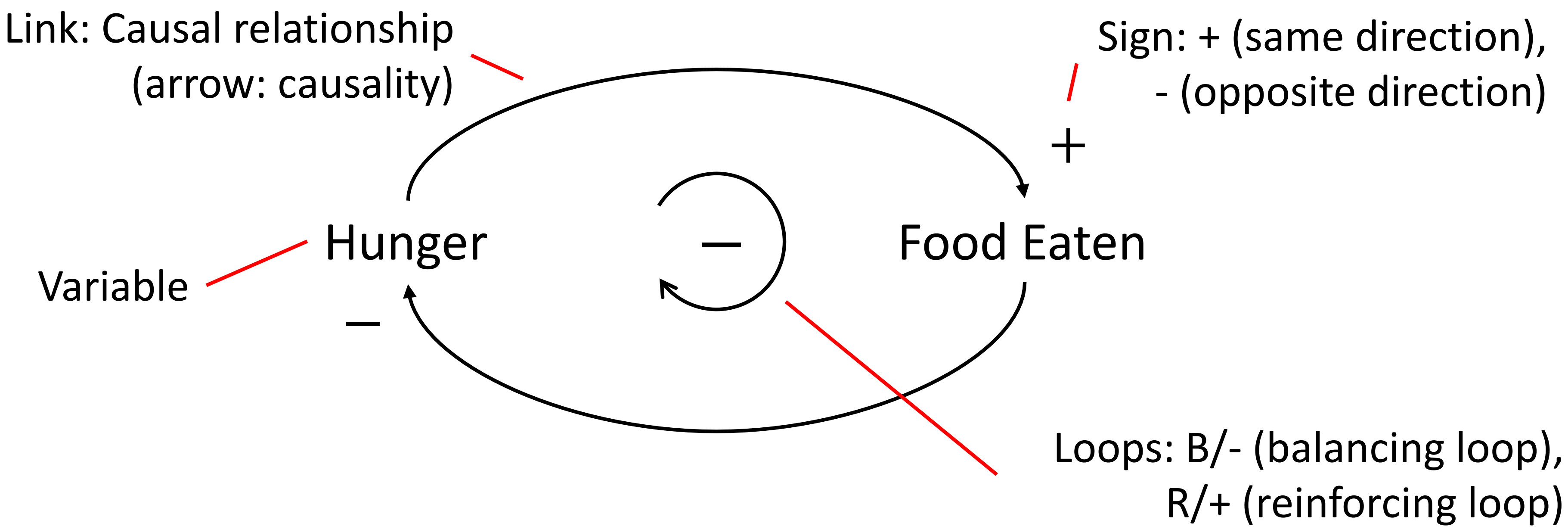
Evolution of SD Applications

- 1961: Industrial Dynamics
  - Application of SD to management (not so controversial)
- 1969: Urban Dynamics
  - Application of SD to urban planning (controversial)
- 1971: World Dynamics
  - Application of SD to societal planning (very controversial)



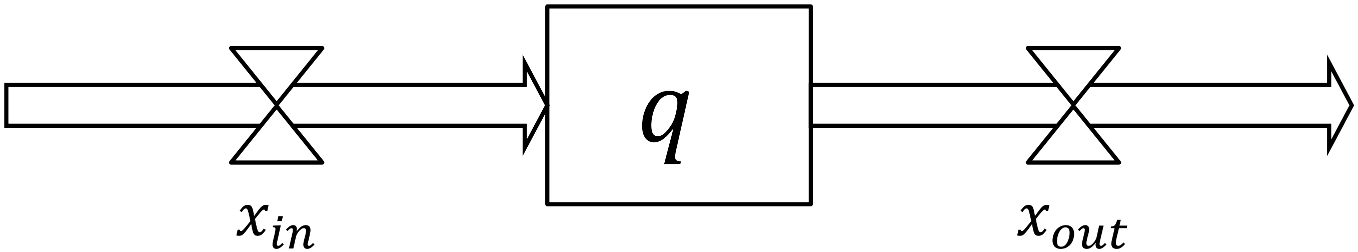
Causal Loop Diagrams

- **Causal Loop Diagrams** (CLDs) are graphical models of interrelationships between variables
  - Cause-and-effect linkages, first step to SD models



System Dynamics: Stock

- A **stock** is a simulation state variable
  - Accumulates information over time (has memory)

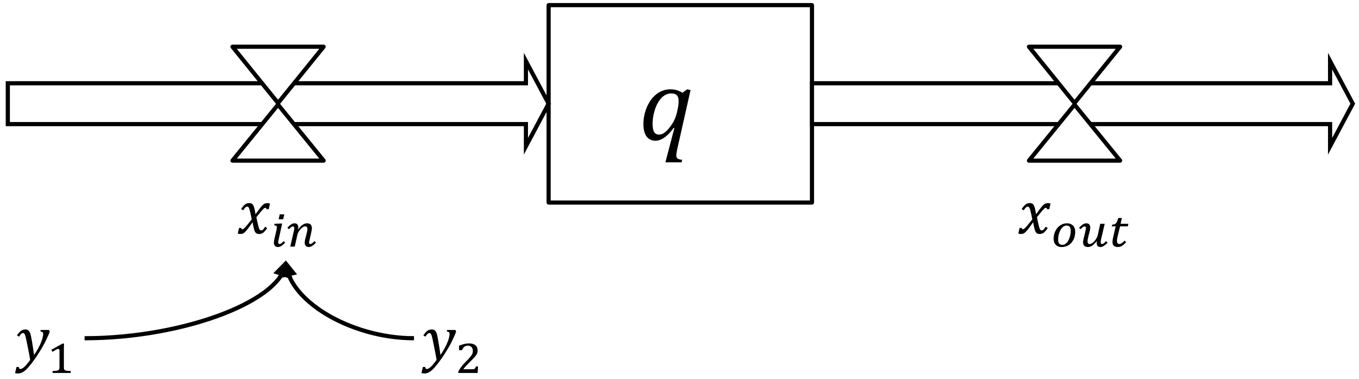


$$\frac{dq}{dt} = x_{in}(t) - x_{out}(t)$$

$$q(t) = q(0) + \int_0^t \big(x_{in}(i) - x_{out}(i)\big) di$$

System Dynamics: Flow

- A **flow** is a derived (dependent) variable
  - Does not accumulate information (memoryless)
  - Function of other stocks, flows, and constants



$$x_{in}(t) = f\left(y_1(t), y_2(t)\right)$$

System Dynamics: Simulator

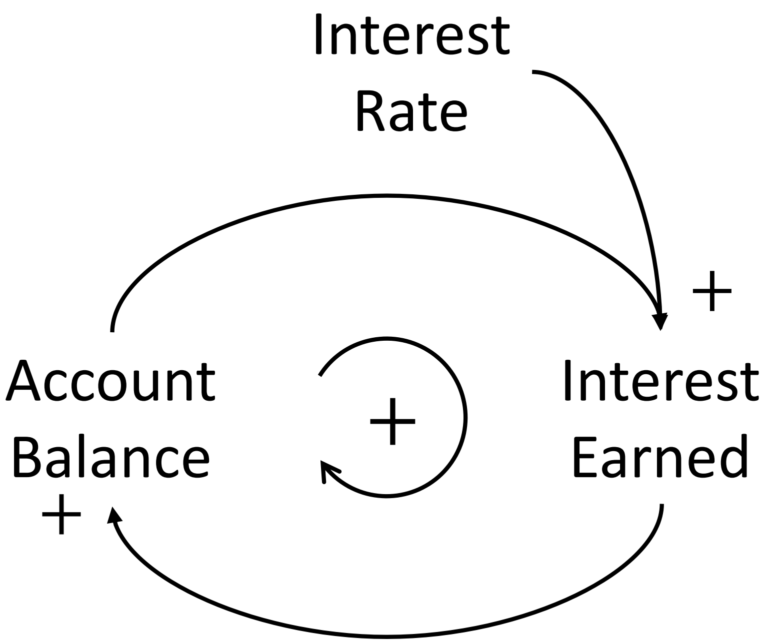
- A **SD simulator** runs continuous-time simulation
- Numerical integration of each stock variable
- For example, Euler integration:

$$q(t + \Delta t) = q(t) + \Delta t \frac{dq}{dt}$$

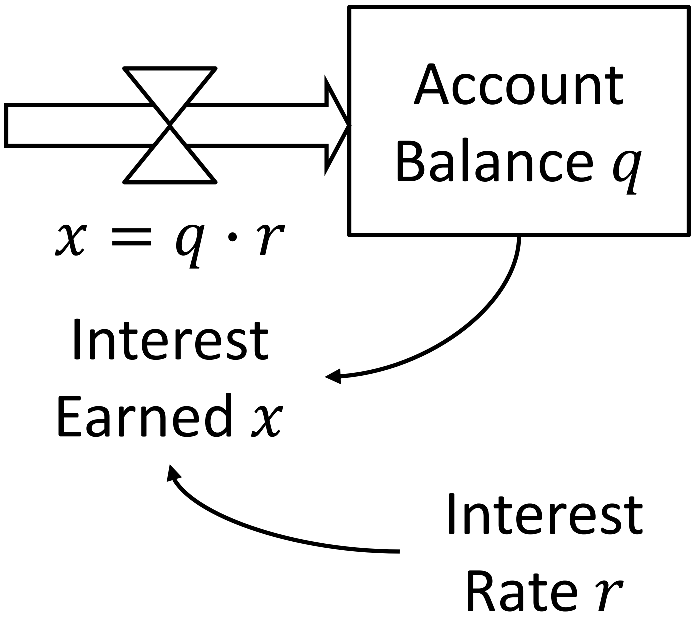
$$q(t + \Delta t) = q(t) + \Delta t\big(x_{in}(t) - x_{out}(t)\big)$$

Example: Savings Account

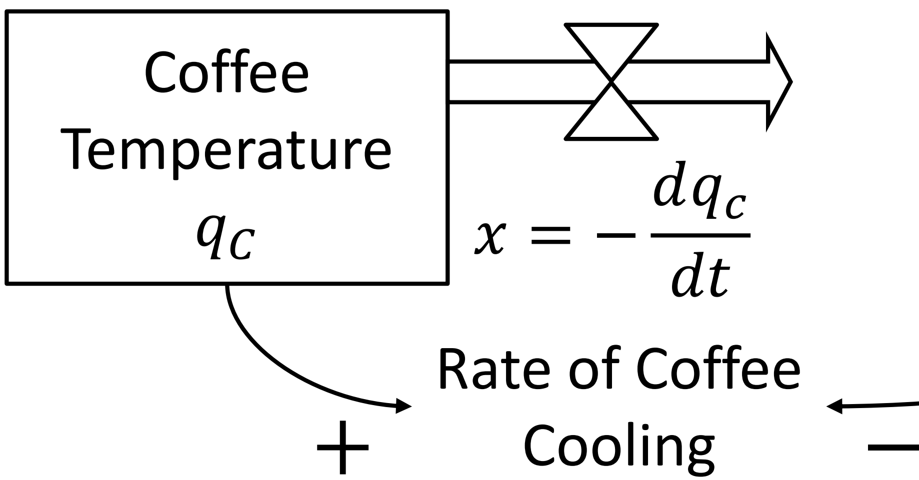
- Causal Link Diagram



- System Dynamics Model:



Example: Coffee Cooling



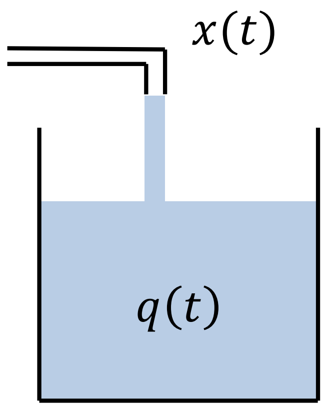
Newton's Law of Cooling:

$$x(q_R, q_C) = k \cdot (q_C - q_R)$$

$$q_C(t + \Delta t) = q_C(t) - \Delta t\big(k \cdot (q_C(t) - q_R(t))\big)$$

$$q_R(t + \Delta t) = q_R(t)$$

Example: Water Basin



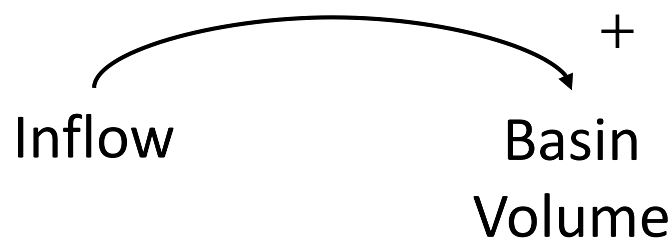
$$q(0) = 5\ \mathrm{m}^3$$

$$y(t) = q(t)$$

$$\frac{dq}{dt} = x(t) = t$$

$$\Delta t = 1\ \mathrm{min}$$

- Causal Link Diagram



- System Dynamics Model: