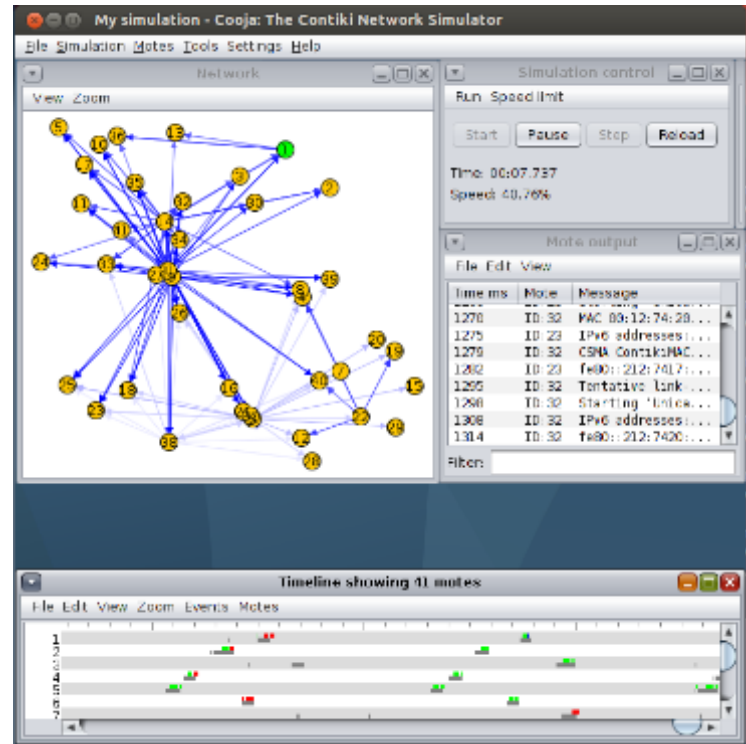# Chapter: Simulation

Chapter : Simulation

1.1 Introduction

1.2 System

- Discrete-Event Simulation
- Continuous Simulation

1.3 Model

- Type of Model
- Develop of Model
- Verification vs Validation

1.4 Monte Carlo Simulation

- Brief History
- Type of Monte Carlo
- Type of Random Number
- Probability Density Function
- Using Monte Carlo?
- Why use Monte Carlo
- Example
- Problems with Monte Carlo

Simulation?

A simulation is the imitation of the operation of real-world process or system over time.

- A method used to better understand problems and predict what will happen in the future by mimicking what is happening in the real world or what will happen in the future.
  ex) Military Simulation

Simulation is used before an existing system is altered or a new system built,

- to reduce the chances of failure to meet specifications,
- to eliminate unforeseen bottlenecks,
- to prevent under or over-utilization of resources,
- to optimize system performance.

Goal of Simulation

- Develop a simulation program that implements a computational model of the system of interest

- Run the simulation program and use the data collected to estimate the performance measures of interest

  (often involves the use of randomization)

- So simulation can be used as
  - Analysis tool for predicating the effect of changes
  - Design tool to predicate the performance of new system

- It is better to do simulation before Implementation.

Advantages of Simulation

- New policies and procedures can be explored without disrupting the ongoing operation of the real system

- New designs can be tested without committing resources for their acquisition

- Bottleneck analysis can be performed to discover where work in process, the system is delayed.

Advantages of Simulation

- A simulation study can help in understanding how the system operates

- "What if" questions can be answered

  What if all the oceans dried up?

  What if plastic was banned?

  What if you could shape your own future just as you wanted it to be?

  What if the South won the Civil War?

  What if no one hears a sound?

Disadvantages of Simulation

- Model building requires special training.
  - Modeling is the process of creating and analyzing a digital prototype of a physical model to predict its performance in the real world.

- Simulation results can be difficult to interpret.

- Simulation modeling and analysis can be time
  - Need proper statistical interpretation for output analysis

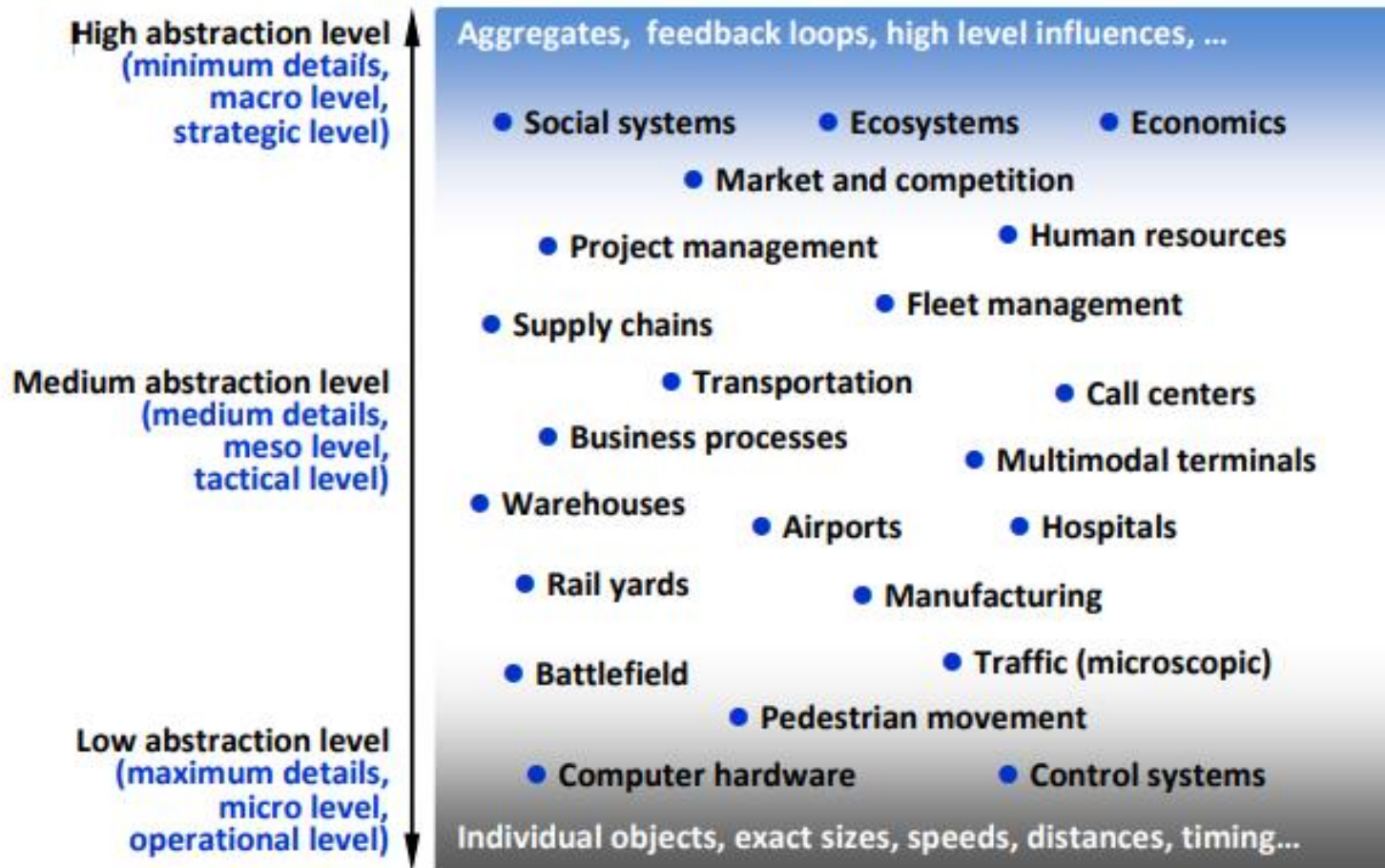- consuming and expensive.

When Simulation Is Not Appropriate

- When the problem can be solved by common sense

- When it is easier to perform direct experiments

- When cost of simulations exceeds (expected) savings for the real system

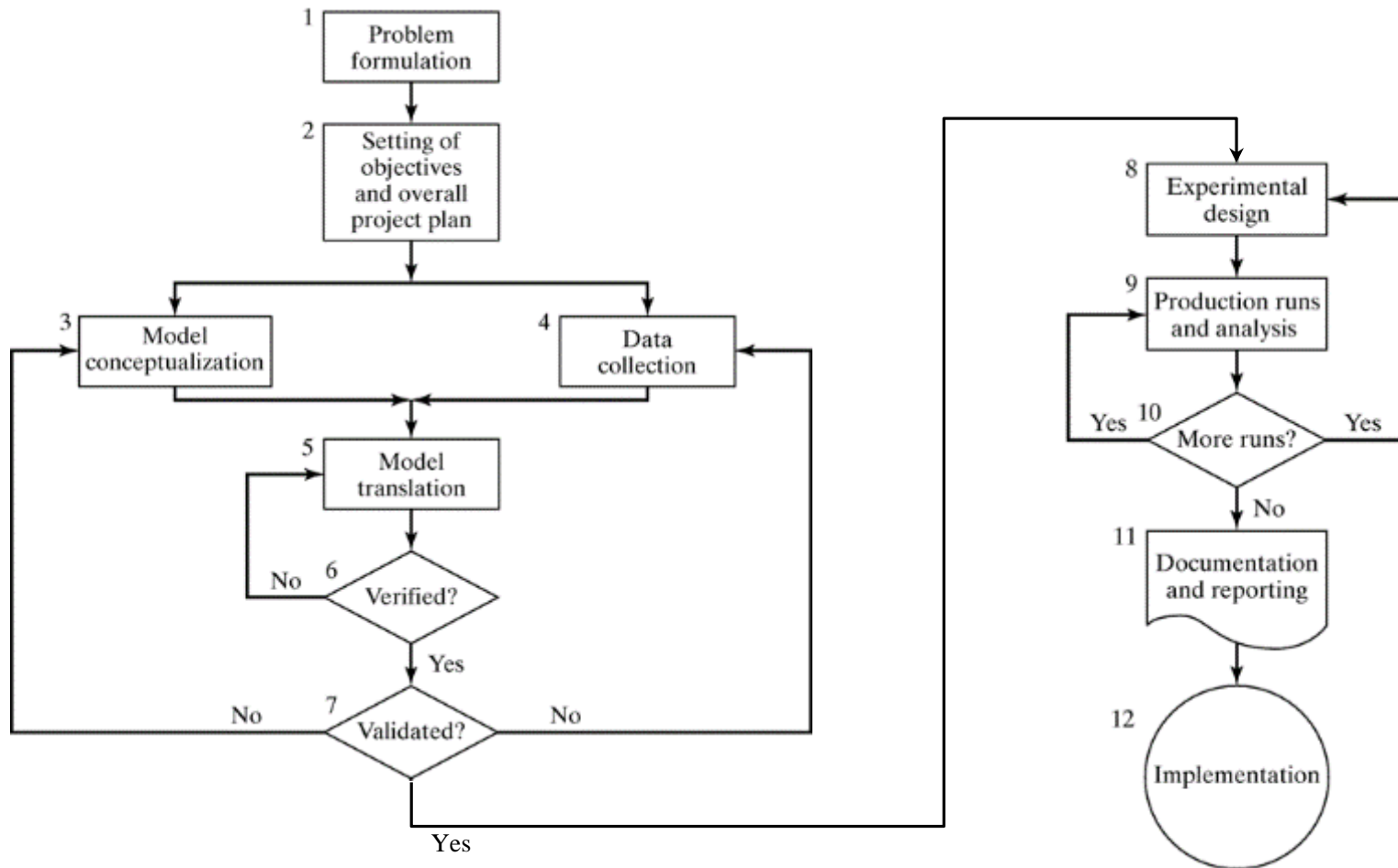- When system behavior is too complex (e.g., humans)

Application Areas of Simulation

- Construction Engineering and Project Management

- Business Process Simulation

- Automated Material Handling System (AMHS)
  - Automatically transporting the production process.

- Risk analysis
  - Insurance, fire disaster, Infectious disease

- War Simulation

- Computer Simulation

- Network Simulation

- Stock Simulation

Application Areas of Simulation

High abstraction level (minimum details, macro level, strategic level)

Medium abstraction level (medium details, meso level, tactical level)

Low abstraction level (maximum details, micro level, operational level)

Aggregates, feedback loops, high level influences, ...

- Social systems
- Ecosystems
- Economics
- Market and competition
- Project management
- Human resources
- Fleet management
- Supply chains
- Transportation
- Call centers
- Business processes
- Multimodal terminals
- Warehouses
- Airports
- Hospitals
- Rail yards
- Manufacturing
- Battlefield
- Traffic (microscopic)
- Pedestrian movement
- Computer hardware
- Control systems

Individual objects, exact sizes, speeds, distances, timing...

Steps in Simulation Study
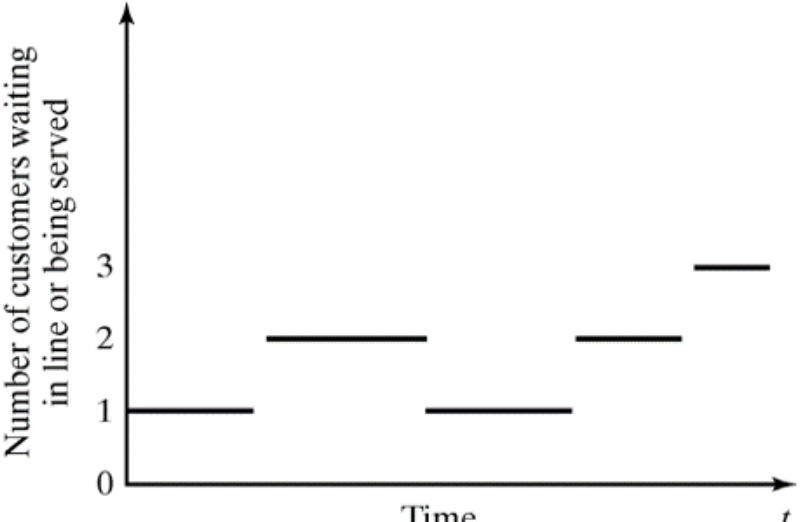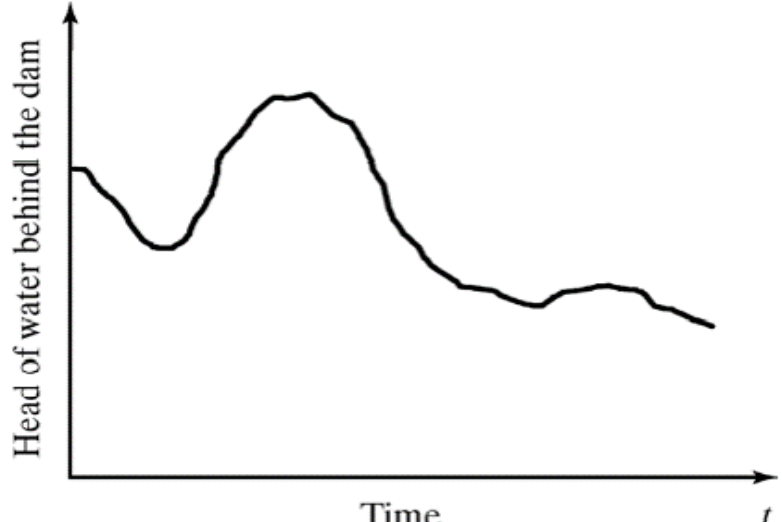
Chapter 1: Simulation

Definition of System

System is

set of objects, joined to accomplish some purpose
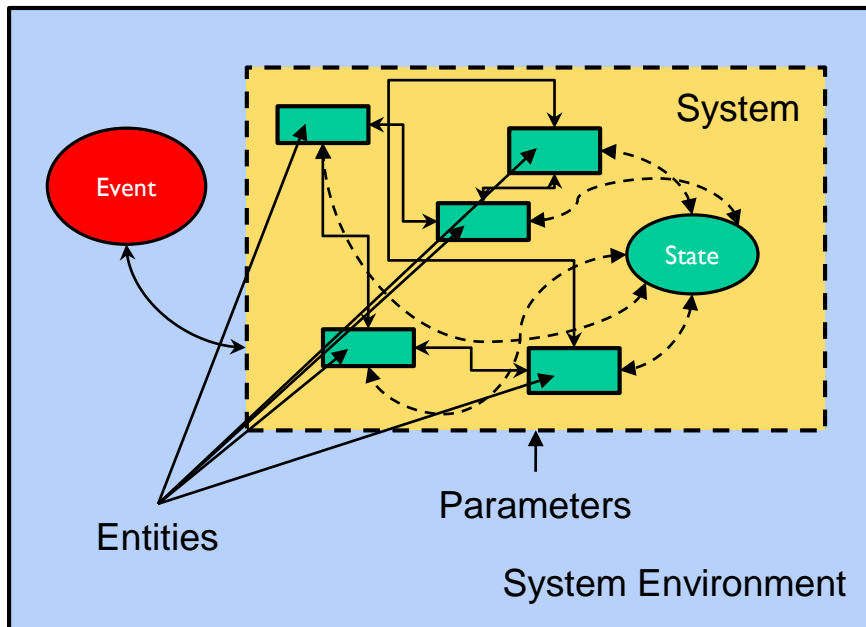

- *Entities:* System of Interest

- *State of system:*

  A set of variables that describe a system at any time

- Event:

  Immediate occurrences that may be associated with system state changes

- System Environment:

  Areas outside the system that affect system behavior

Discrete and Continues Systems

| Discrete System	Continues System
discrete sequence of events in time	continuously over time
Bank Customers, Inventory Control	Dam Water Level, Weather Temperature

Discrete-Event Simulation

- *A simulation model in which system state evolves over a discrete sequence of events in time*

  - *System state changes only when an event occurs*

  - System state does not change between the events
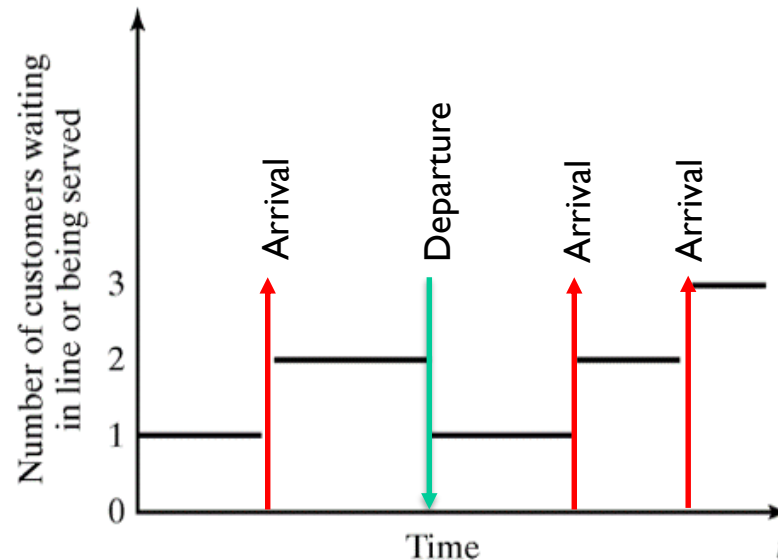
Continuous Simulation

- *A simulation model in which system state evolves continuously over time*

    - a system is modeled with the help of variables that change continuously according to a set of differential equation

    - The set of differential equations can be formulated in a conceptual model representing the system on an abstract level.
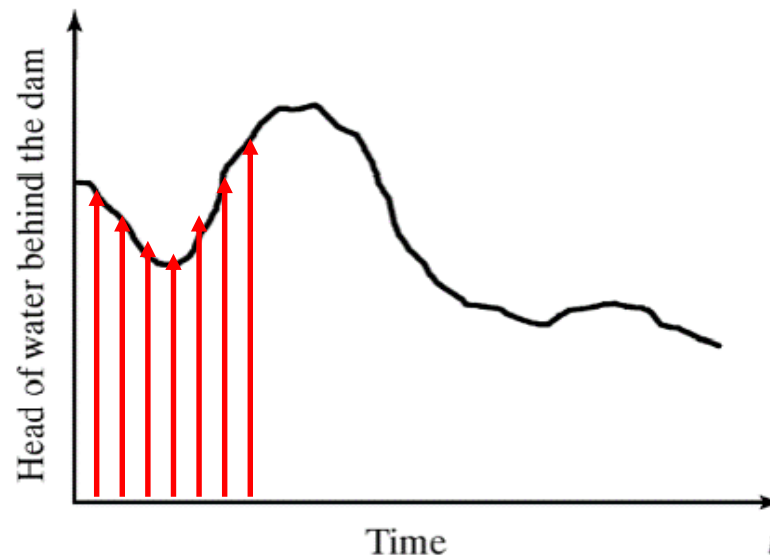
Chapter 1 : Simulation

1.1 Introduction

1.2 System

- Discrete-Event Simulation
- Continuous Simulation

1.3 Model

- Type of Model
- Develop of Model
- Verification vs Validation

1.5 Monte Carlo Simulation

- Brief History
- Type of Monte Carlo
- Type of Random Number
- Probability Density Function
- Using Monte Carlo?
- Why use Monte Carlo
- Example
- Problems with Monte Carlo
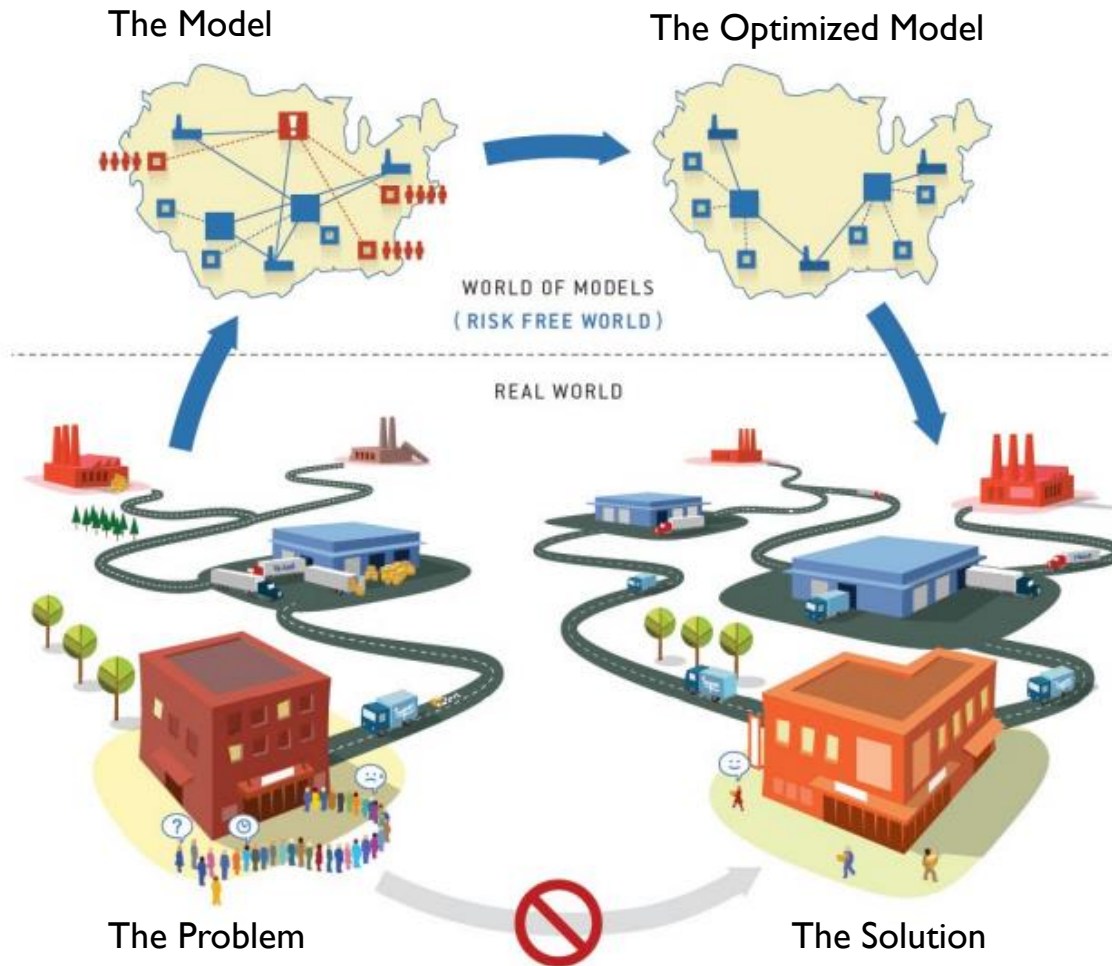
Definition of Model

Model is

- Abstracting and simplifying the system as an alternative
- A collection of system-related information collected for system research purposes
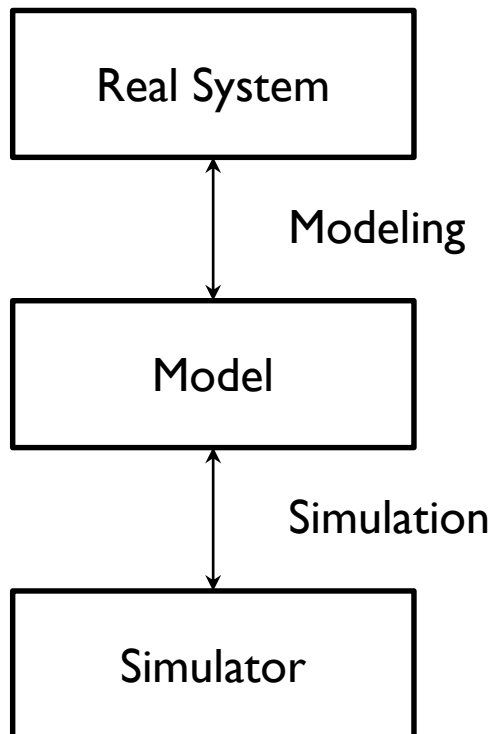- Observation of existing systems -> Deriving new systems

Role of Model

- A representation of the construction and working of some system of interest.
- To enable the analyst to predict the effect of changes to the system.
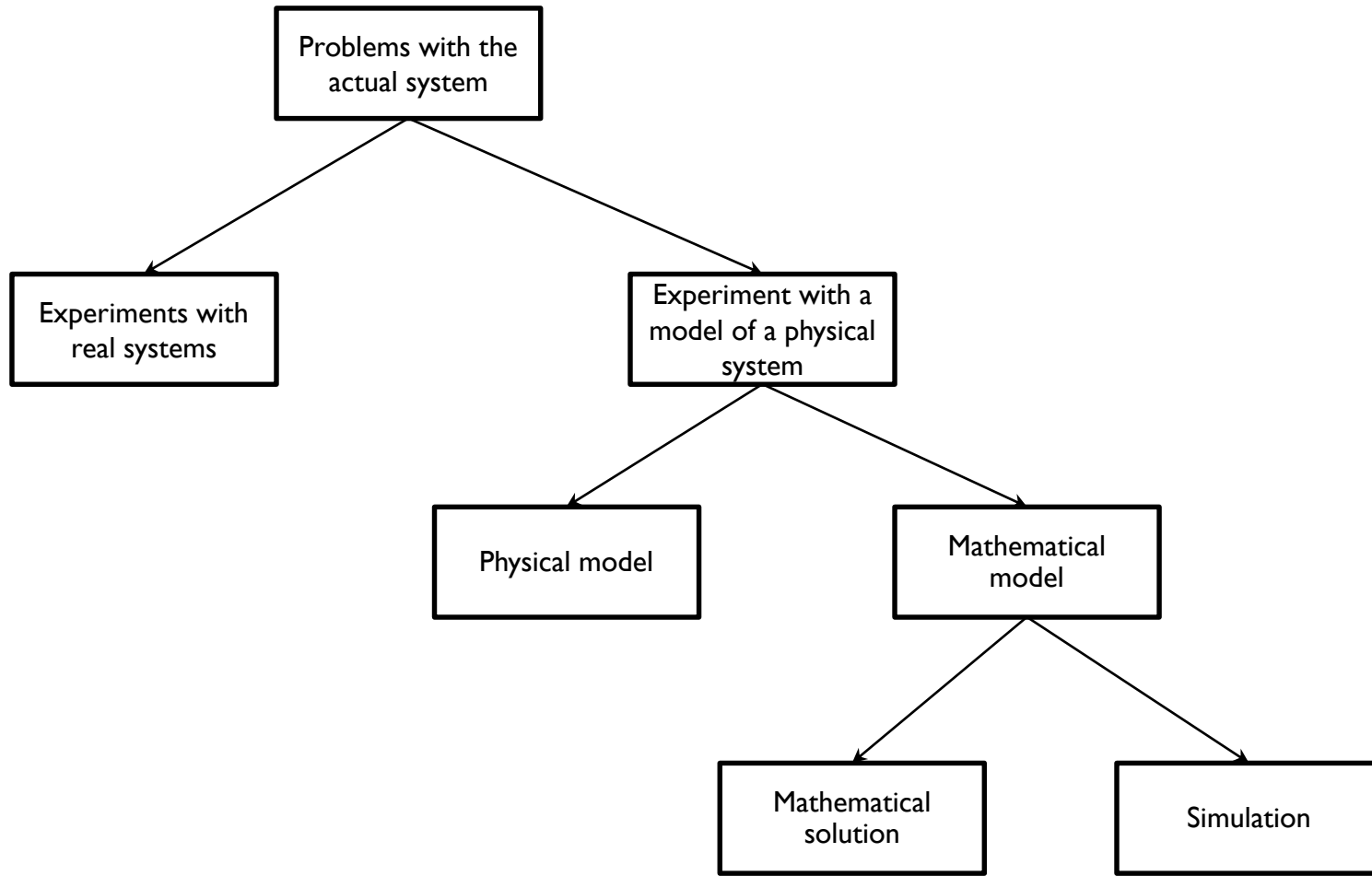
Definition of Model

The Model

The Optimized Model

WORLD OF MODELS
(RISK FREE WORLD)

REAL WORLD

The Problem

The Solution

Basic structure of Simulation and Modeling

Zeigler Theory


```
┌──────────────────┐
│   Real System    │
└──────────────────┘
         ↕
      Modeling
┌──────────────────┐
│      Model       │
└──────────────────┘
         ↕
     Simulation
┌──────────────────┐
│    Simulator     │
└──────────────────┘
```

- *Real System:* Source of data (part of the real world) Natural or artificial

- *Model: A set of commands that generate data(Artificial system for the real system)*

- *Simulator:* A device (computer) capable of executing a model's instructions

- *Modeling:* Model configuration of the real system

- *Simulation :* Validate that the simulator is performing correctly and accurately by the model

Type Of Model

Problems with the actual system
├── Experiments with real systems
└── Experiment with a model of a physical system
    ├── Physical model
    └── Mathematical model
        ├── Mathematical solution
   └── Simulation

Type Of Model

- *Deterministic or Stochastic*

  - *Does the model contains stochastic components?*

  - A model is deterministic if its behavior is entirely predictable. Given a set of inputs, the model will result in a unique set of outputs.

  - *Static or Dynamic*

  - *Is time a significant variable? (process=system+time)*
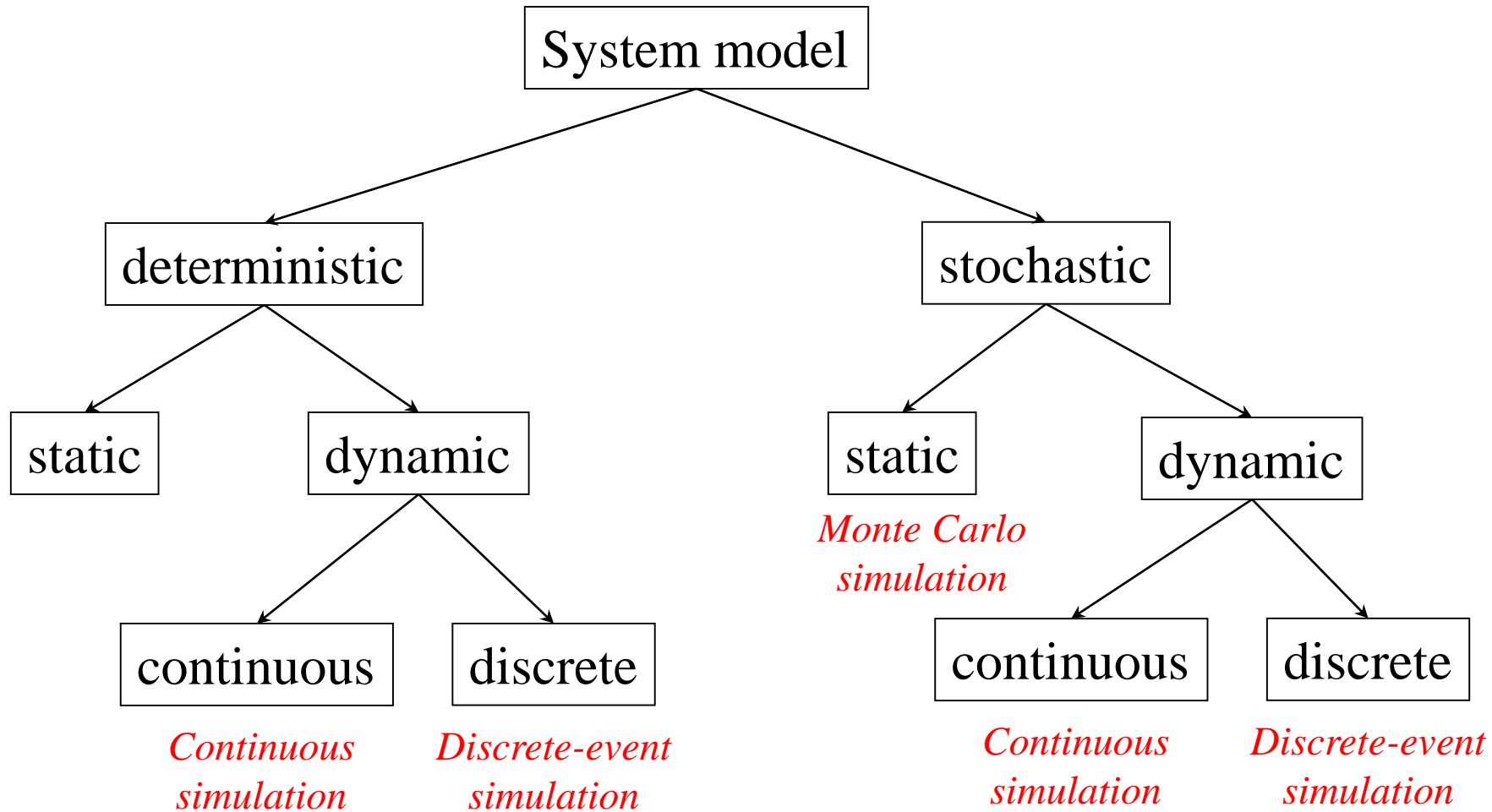
- *Continuous or Discrete*

  - Does the system state evolve continuously or only at discrete points in time?

  - *Continuous:* Over time..

  - *Discrete:* Depending on the specific event...

    - Queuing, Inventory

Continuous

Discrete

event 1   event 2   event 3   time
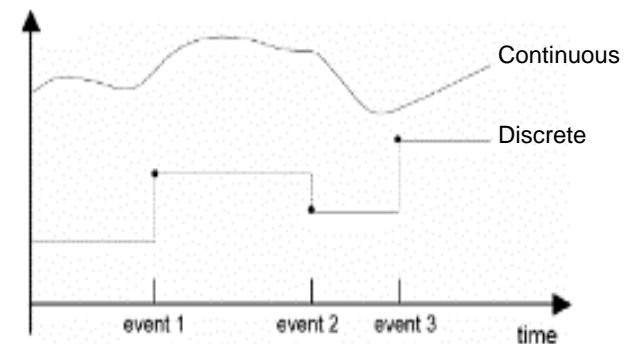
- 연봉 1,200만원을 받는 회사원이 저축계획을 세우려고 하는데 다음과 같다. 급여액 은 매년 5%씩 인상이 되는데 자신의 연봉에서 10%를 매년 저축하려고 한다. 은행 의 이율는 매년 복리 5%라고 할 때 앞으로 30년 후의 저축액은 얼마나 되겠는가?

- ```
salary =1200
savings =0
perinc =0.05
perint =0.09
year =0
while year < 30 do
  {
    year = year + 1
    savings = savings + (perint * savings+0.10*salary)
    salary = salary+perinc * salary // 연봉 계산 //
    print year, salary, savings
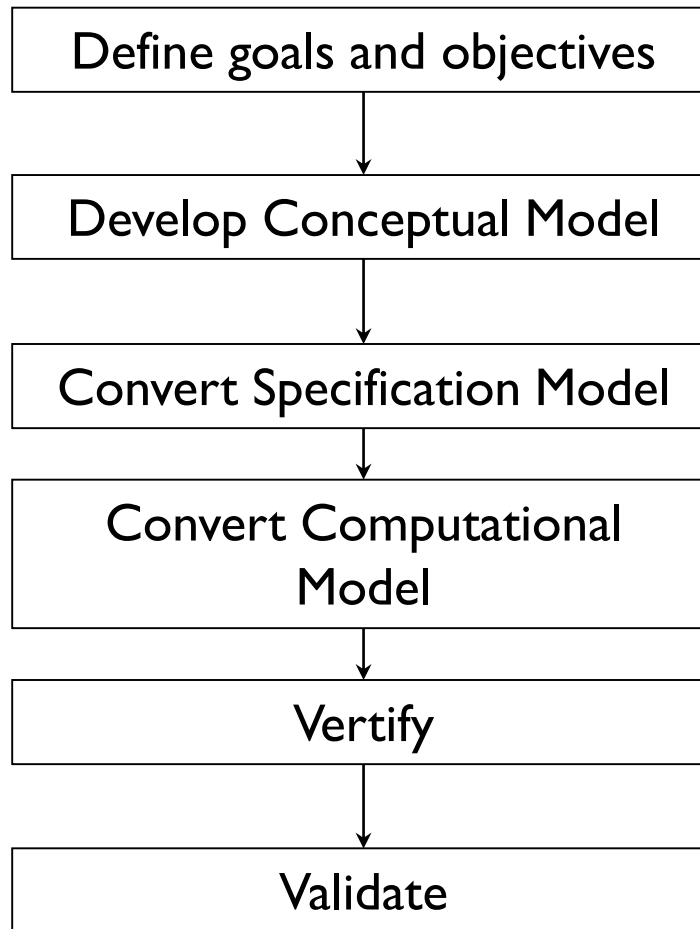 } stop
```

# Type Of Model

```
                          ┌──────────────┐
                          │ System model │
                          └──────────────┘
                    ┌───────────┴────────────┐
              ┌───────────────┐        ┌──────────────┐
              │ deterministic │        │  stochastic  │
              └───────────────┘        └──────────────┘
              ┌──────┴───────┐          ┌─────┴────────┐
        ┌────────┐   ┌──────────┐  ┌────────┐   ┌──────────┐
        │ static │   │ dynamic  │  │ static │   │ dynamic  │
        └────────┘   └──────────┘  └────────┘   └──────────┘
                    ┌─────┴──────┐  *Monte Carlo*  ┌─────┴──────┐
            ┌────────────┐ ┌──────────┐ *simulation* ┌────────────┐ ┌──────────┐
            │ continuous │ │ discrete │            │ continuous │ │ discrete │
            └────────────┘ └──────────┘            └────────────┘ └──────────┘
```

Continuous simulation   *Discrete-event simulation*   *Continuous simulation*   *Discrete-event simulation*

Develop Keypoint

- *Convexity*
  - Summarize complex interrelationships within the system

- *Relevance*
  - Include only aspects of the system related to the purpose of the study

- *Precision*
  - Considering the accuracy of the excerpts

- *Integrity*
  - a group of individual realities into larger ones

Develop Of Model

```
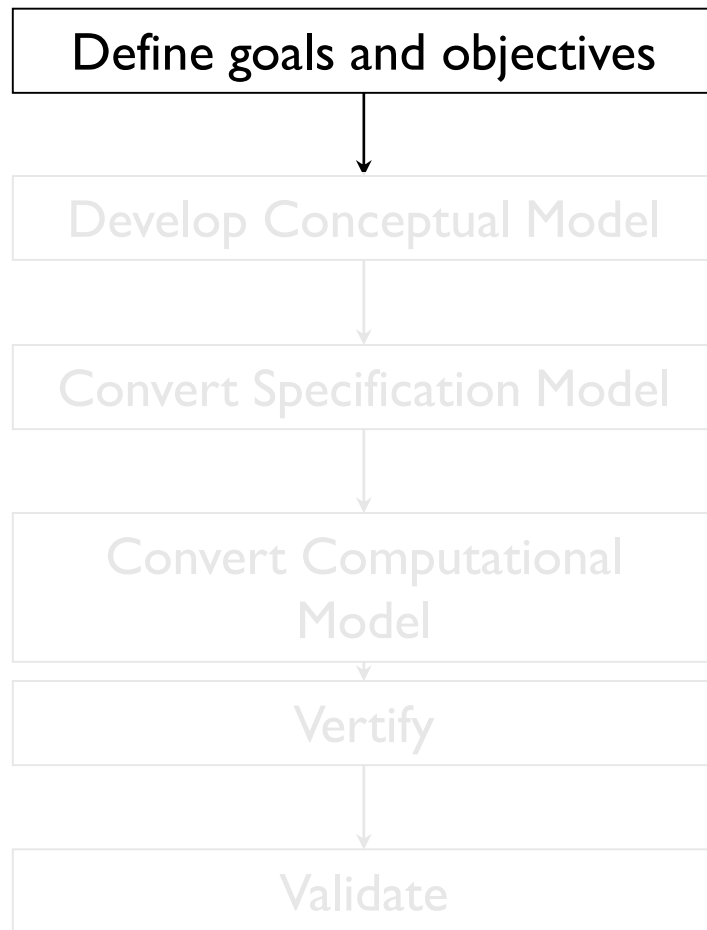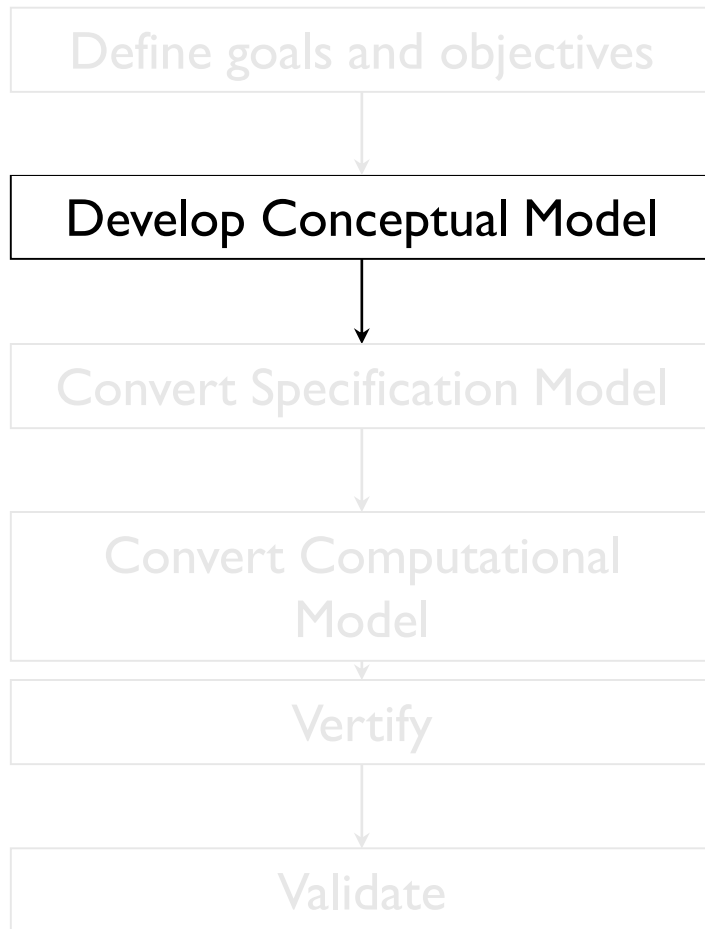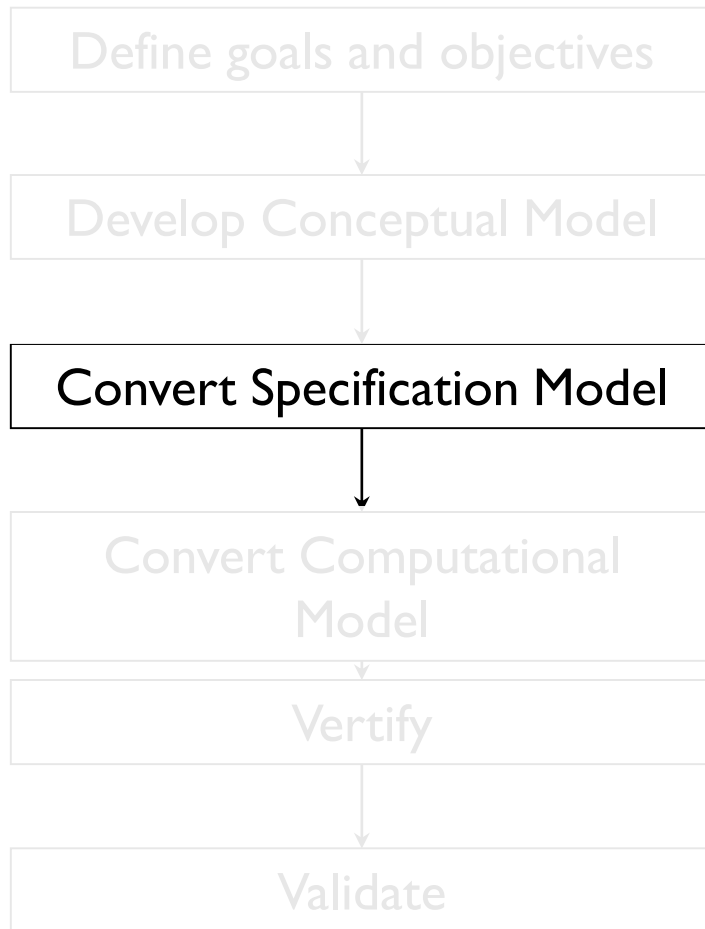┌─────────────────────────────┐
│  Define goals and objectives │
└─────────────────────────────┘
               │
               ▼
┌─────────────────────────────┐
│   Develop Conceptual Model   │
└─────────────────────────────┘
               │
               ▼
┌─────────────────────────────┐
│  Convert Specification Model │
└─────────────────────────────┘
               │
               ▼
┌─────────────────────────────┐
│     Convert Computational    │
│            Model             │
└─────────────────────────────┘
               │
               ▼
┌─────────────────────────────┐
│            Vertify           │
└─────────────────────────────┘
               │
               ▼
┌─────────────────────────────┐
│           Validate           │
└─────────────────────────────┘
```

Develop Of Model

| Define goals and objectives |

↓

Develop Conceptual Model

Convert Specification Model

Convert Computational Model

Vertify

Validate

- What does you (or the customer) hope to accomplish with the model

Develop Of Model

Define goals and objectives

Develop Conceptual Model

Convert Specification Model

Convert Computational Model

Vertify

Validate

- An abstract (i.e., not directly executable) representation of the system

- What should be included in model? What can be left out?

- What are the state variables, which are dynamic, and which are important?

Develop Of Model

Define goals and objectives

Develop Conceptual Model

Convert Specification Model

Convert Computational Model

Vertify

Validate

- A more detailed specification of the model including more specifics

- Specification
  - On Paper
  - Diagram, pseudocode, etc
  - How will the model receive input?

- Ex) Traffic Example: Road Geometry, Signal timing, driver behavior …

Develop Of Model

Define goals and objectives
Develop Conceptual Model
Convert Specification Model
Convert Computational Model
Vertify
Validate
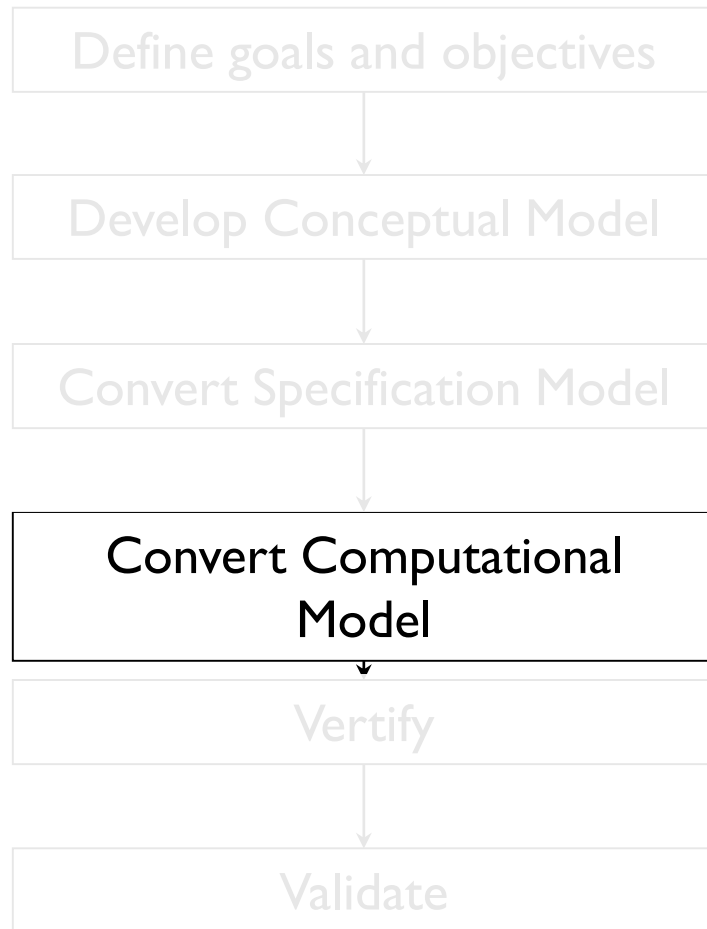
- A computer program

- General-purpose PL or simulation language?
  - FORTRAN, C++, Python, R , etc..

Develop Of Model
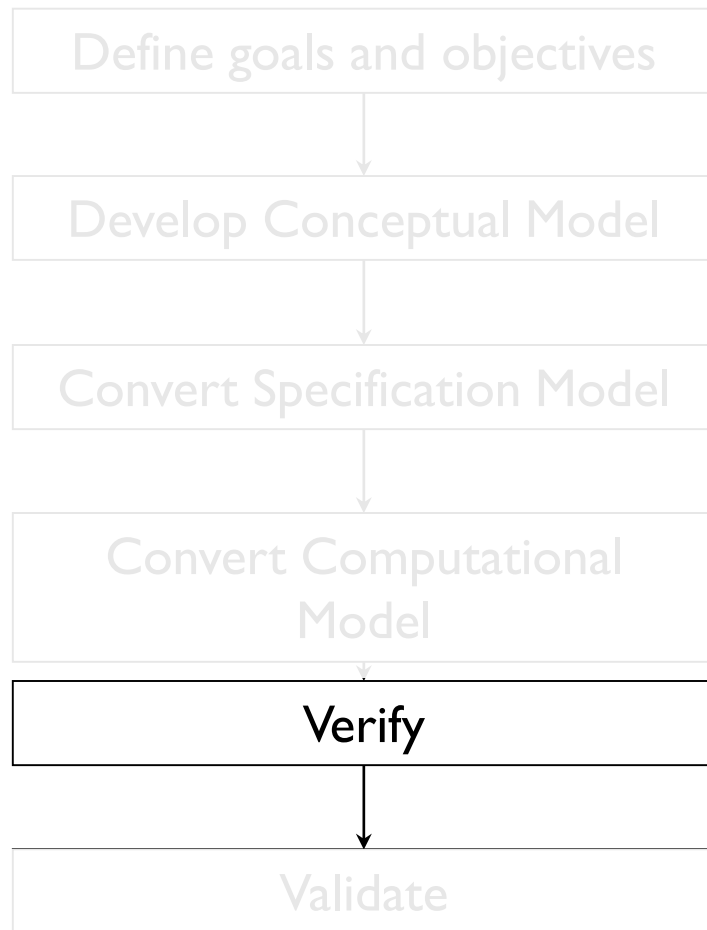
Define goals and objectives

↓

Develop Conceptual Model

↓

Convert Specification Model

↓

Convert Computational Model

↓

Verify

↓

Validate

- Computational model should be consistent with specification model

- Did we build the model right? (focus on process)

Develop Of Model
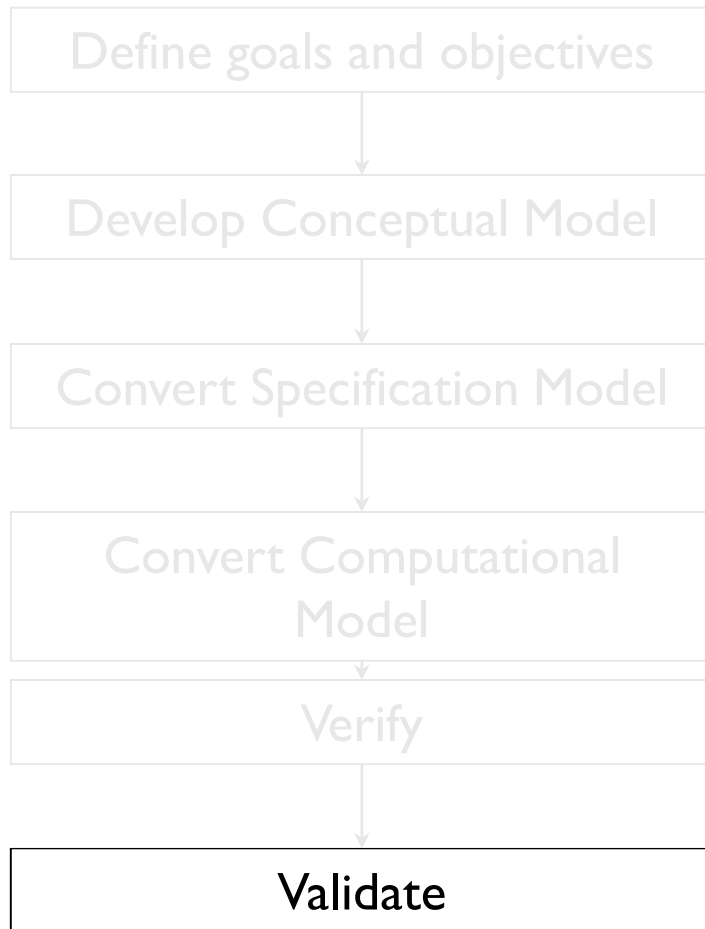
| Define goals and objectives |
| Develop Conceptual Model |
| Convert Specification Model |
| Convert Computational Model |
| Verify |
| **Validate** |

- Computational model should be consistent with the system being analyzed

- Did we build the model right? (focus on process)

- Can an expert distinguish simulation output from system output?

Verification vs Validation

Verification

- Computational model should be consistent with specification model

- Did we build the model right?

Validation

- Computational model should be consistent with the system being analyzed

- Did we build the model right?

- Can an expert distinguish simulation output from system output?

Develop Of Model
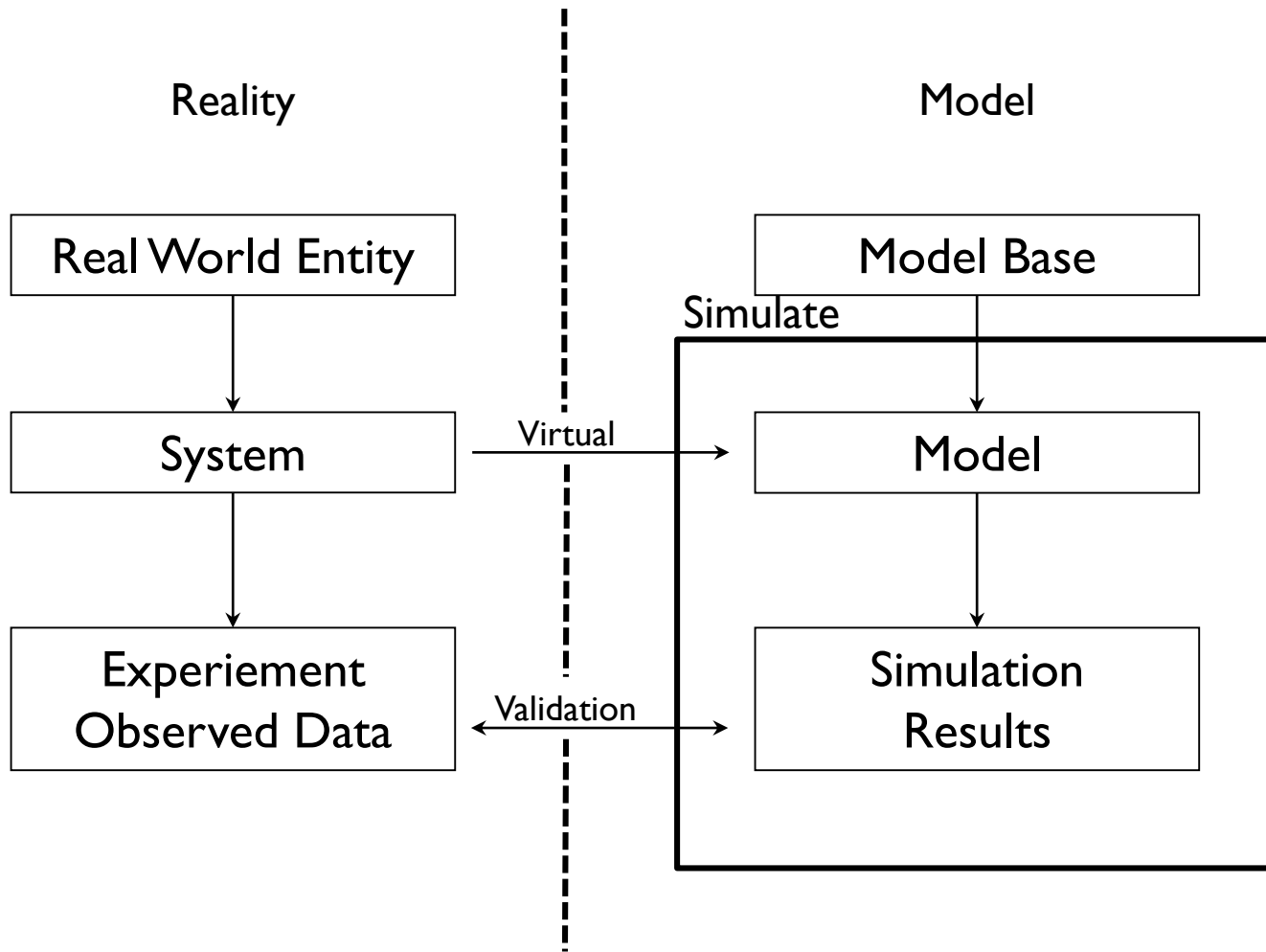
Reality

Model

Real World Entity		Model Base

Simulate

System → Virtual → Model

Experiement Observed Data ← Validation → Simulation Results

Chapter 1: Simulation

1.1 Introduction

1.2 System
- Discrete-Event Simulation
- Continuous Simulation

1.3 Model
- Type of Model
- Develop of Model
- Verification vs Validation

1.4 Monte Carlo Simulation
- Brief History
- Type of Monte Carlo
- Type of Random Number
- Probability Density Function
- Using Monte Carlo?
- Why use Monte Carlo
- Example
- Problems with Monte Carlo

Monte Carlo Simulation

- Name of casino in Monaco (2nd smallest country)

- Its use of randomness and the repetitive nature

- The process are analogous to the activities conducted at a casino

- All the gambling games including rullet, and black are nothing but a series of Markov processes

What is Monte Carlo

" Repetitive Random Sampling "

Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle.

- Monte Carlo methods are stochastic techniques
- It is based on the use of random numbers and probability statistics to simulate problems.

Type of Monte Carlo

- **Static Monte Carlo**
  - Finding the integrals

- **Dynamic Monte Carlo**
  - Dynamics for Materials
  - Dynamics for biological and ecological models
  - Dynamics of Magnets
  - Cell-phone switching
  - Spread of infectious diseases
  - War-game scenarios

Type of Random Numbers

- Pseudorandom numbers are numbers that appear random, but are obtained in a deterministic, repeatable, and predictable manner.

  "Values generated by algorithms to mimic random numbers"

- True random numbers are generated in non-deterministic ways. They are not predictable. They are not repeatable
  - Decay times of radioactive material
  - Electrical noise from a registor or semiconductor
  - Radio channel or audible noise
  - Keyboard timeings
  - Wireless Waveform
  - Guitar String quiver

Probability Density Function

- A probability density function (or probability distribution function = PDF) is a function f defined on an interval (a, b) and having the following properties:

1) $f(x) \geq 0 \qquad \forall x \in [a, b]$

2) $\int_a^b f(x)\, dx = 1$

3) $P(x_1 \leq x < x_2) = \int_{x_1}^{x_2} f(x)\, dx$

Probability Density Function(cont.)

- The cumulative distribution function (CDF) is drawn by accumulating the probability that each probability variable occurs, and has a value of 1 in the final probability variable.

- The probability density function means the slope of the tangent line of the probability variable in the probability distribution function.

- Integrating the probability density function for all probability variables has a value of 1.
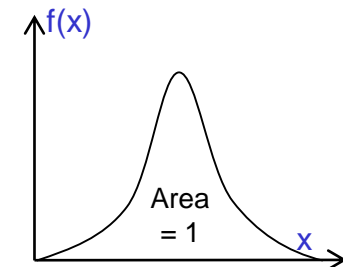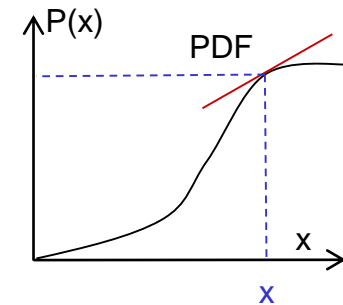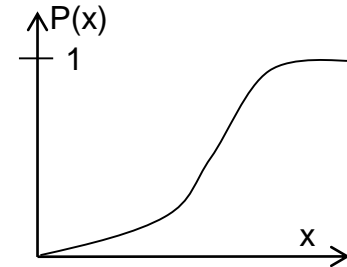
Using Monte Carlo?
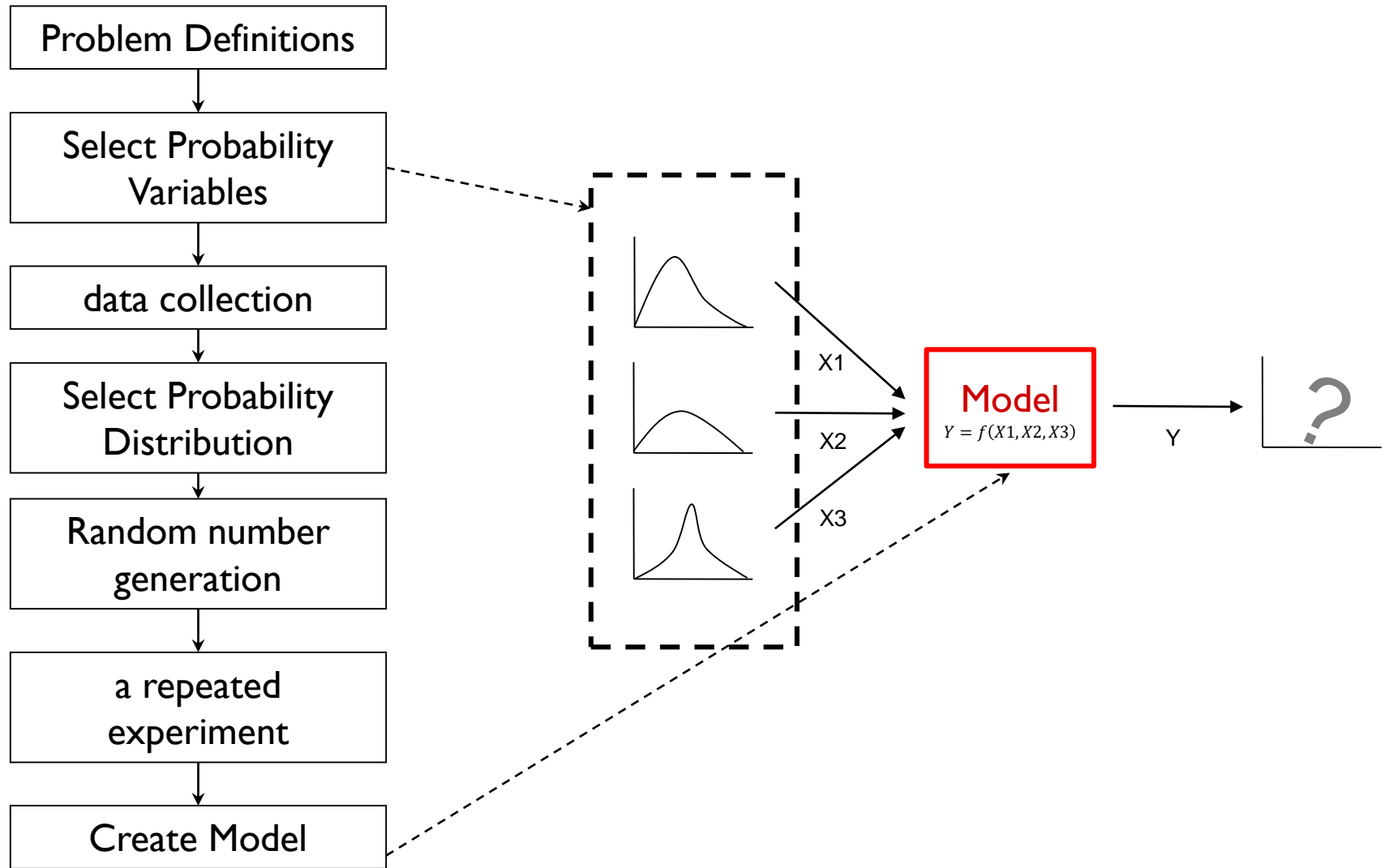
Why use Monte Carlo

- It allows us to examine complex system. And is usually easy to formulate.

- For example, solving equations which describe two atoms interactions. This would be doable without using Monte Carlo method. But solving the interactions for thousands of atoms using the same equations is impossible.

- However, the solutions are imprecise and it can be very slow if higher precision is desired.

Example1: Toss The Coin

What is the probability of flipping a coin face up?

$$\frac{Coin\ Up}{Total\ Toss}$$

Example1-Source : Toss The Coin

```python
import matplotlib.pyplot as plt
import random

n = 10000

up_Count = 0
res = [];

plt.xlim(1,n)
plt.ylim(0,1)

for i in range(1,n):
    coin =random.uniform(0,1)
    # print(coin)
    if coin > 0.5: #Coin Up
        up_Count  = up_Count+1

    res.append(up_Count/i)

plt.plot(range(1,n),res,'bo:', label=res[len(res)-1])
plt.legend(loc='lower right')

plt.axhline(y=res[len(res)-1], color="r", linestyle="--")
plt.show()
```

Example2: Computing Pi

$n = 3000, \pi \approx 3.1133$
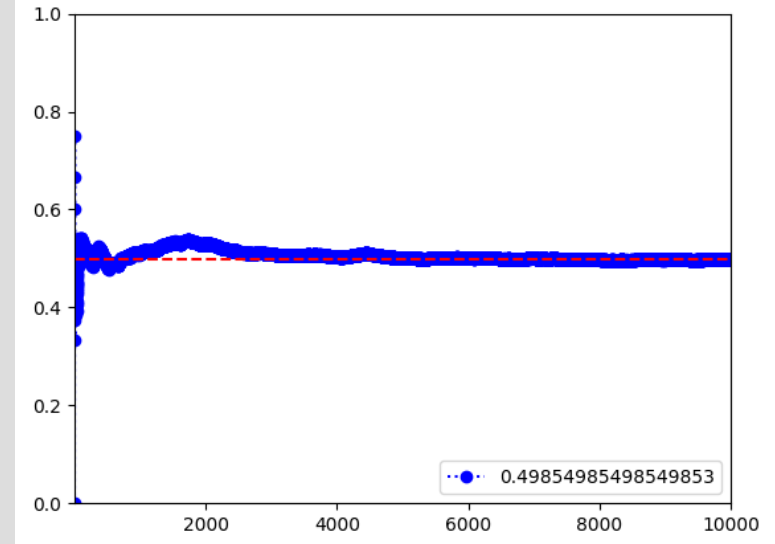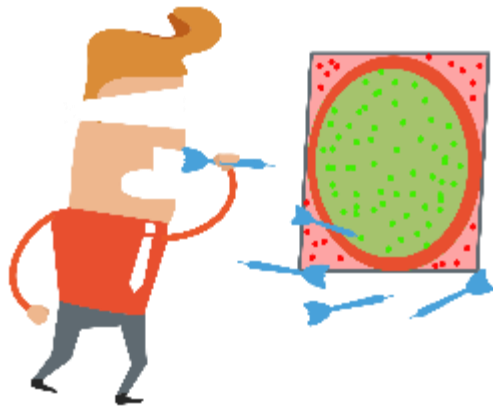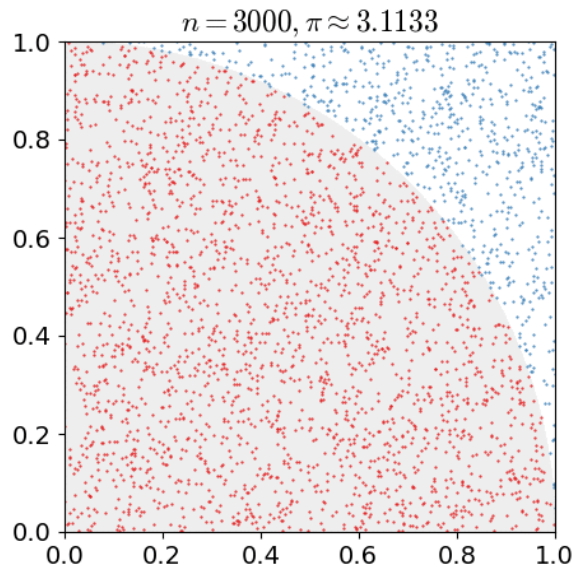
1. Draw a square, and then insert a quadrant in it.

2. Uniformly distribute a certain number of points over a square.

3. Count the number of points inside the quadrant (that is, less than 1 from the origin).

4. The ratio of the number inside to the total number represents the ratio of the two areas.

5. Multiply the ratio inside the quadrant by 4 to create π

Example2: Computing Pi

$$n = 3000, \pi \approx 3.1133$$

$$\frac{CircleArea}{4} = \frac{\pi R^2}{4} = \frac{Total\ Red\ Point}{Total\ Point}$$

$$\frac{A_{circle}}{A_{square}} = \frac{\pi R^2/4}{L^2}$$

$$\pi = \left(\frac{A_{circle}}{A_{square}}\right)\left(\frac{4L^2}{R^2}\right)$$

$$= \left(\frac{N_{in\ circle}}{N_{in\ square}}\right)\left(\frac{4L^2}{R^2}\right)$$

$$= 4\left(\frac{N_{in\ circle}}{N_{in\ square}}\right)$$

※ The total number of random points satisfying distance r ($\sqrt{x^2 + y^2}$) is less than the circle R

Example2: Computing Pi

Enter the total number of points: 100
The value of pi is:
3.08
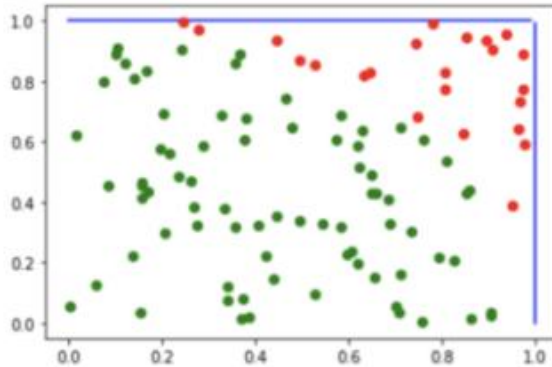
Enter the total number of points: 1000
The value of pi is:
3.112
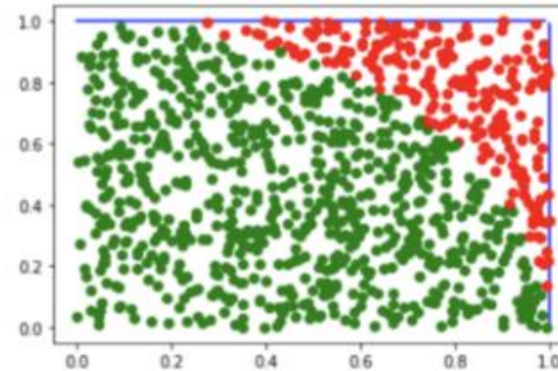
Enter the total number of points: 10000
The value of pi is:
3.1212
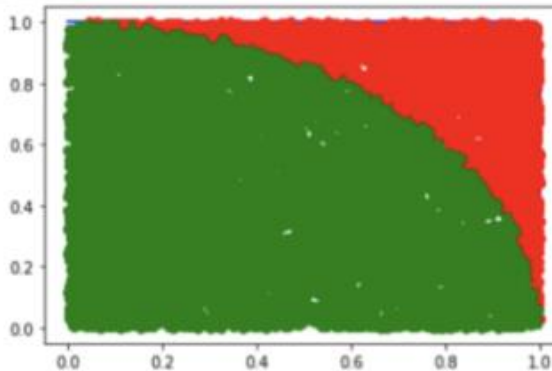
Enter the total number of points: 100000
The value of pi is:
3.14264
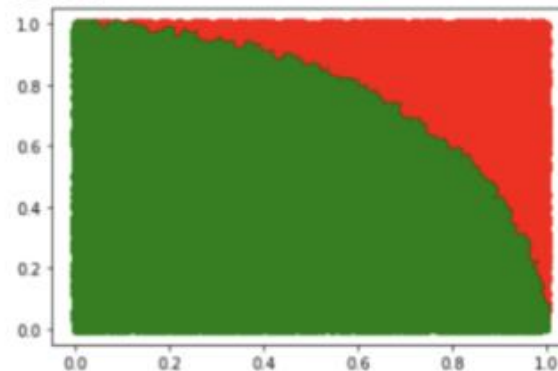
Example2: Computing Pi
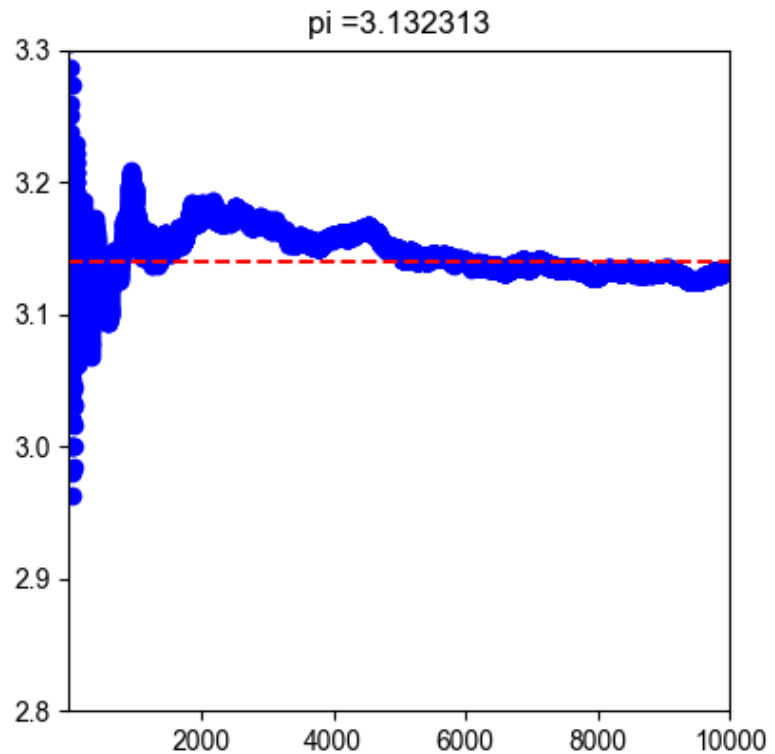
Example2-Source: Computing Pi

```python
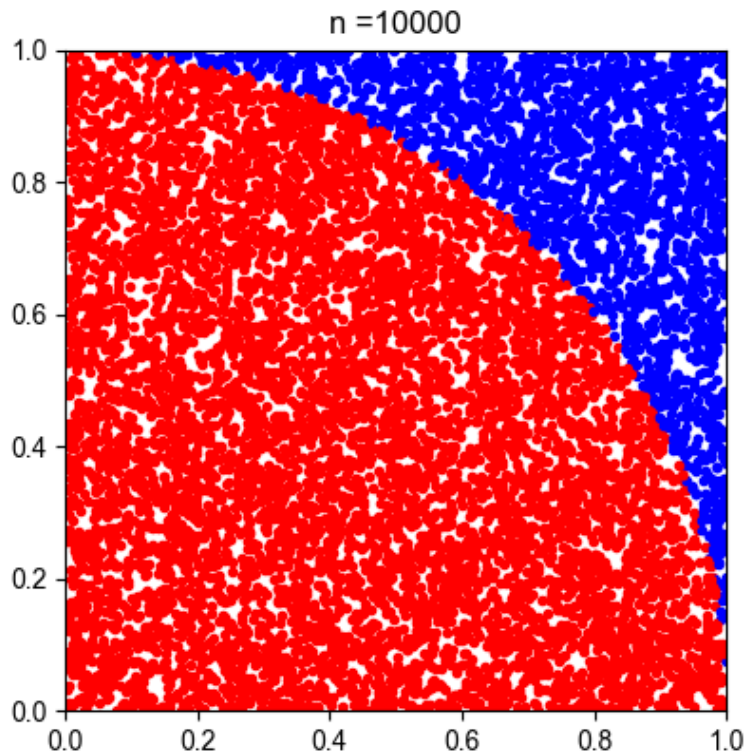import numpy as np
import matplotlib.pyplot as plt

fig, (gr1, gr2) = plt.subplots(1,2, figsize = (8,4), layout='constrained')

gr1.set_xlim(0,1)
gr1.set_ylim(0,1)

n = 10000
r_counter = 0
r_cal = 0

gr2.set_xlim(1,n)
gr2.set_ylim(2.8,3.3)

for i in range(1,n):
    x = np.random.rand(1)
    y = np.random.rand(1)

    if(x**2) + (y**2) < 1:
    # if((x-0.5)**2) + ((y-0.5)**2) <= (0.5) **2:
        gr1.scatter(x, y, c='r', marker='.')
        r_counter = r_counter + 1
    else:
        gr1.scatter(x, y, c='b', marker='.')

    r_cal =  4*(r_counter/i)

    gr1.set_title('n ={:d}'.format(i+1))
    gr2.set_title('pi ={:f}'.format(round(r_cal,20)))
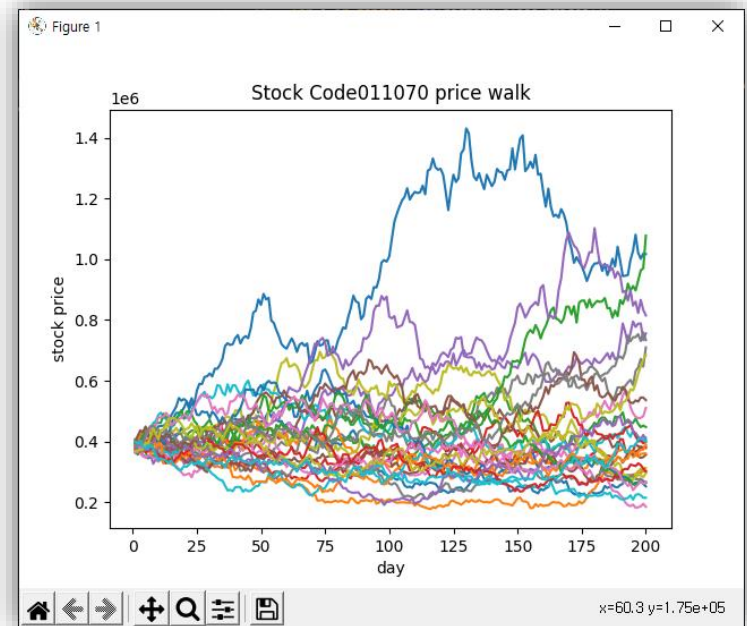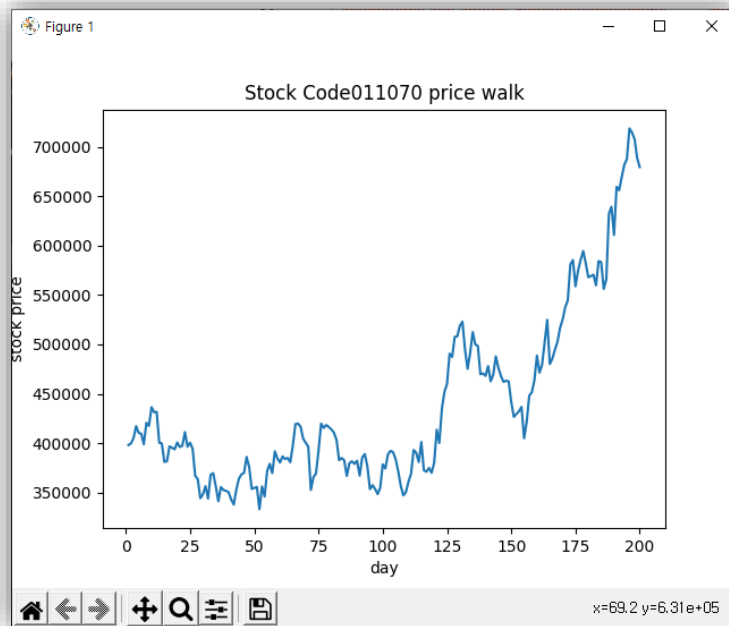    gr2.plot(i,r_cal,'bo:')

    plt.pause(0.01)
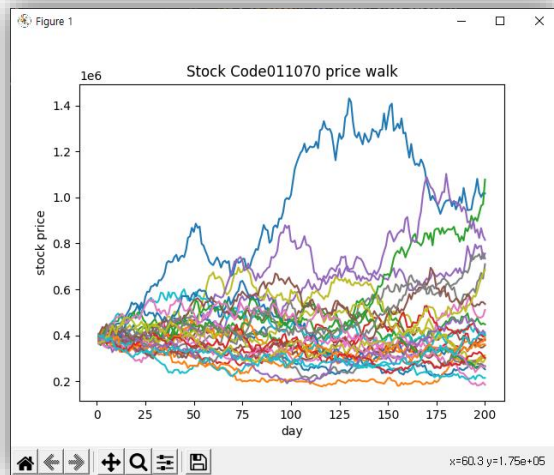
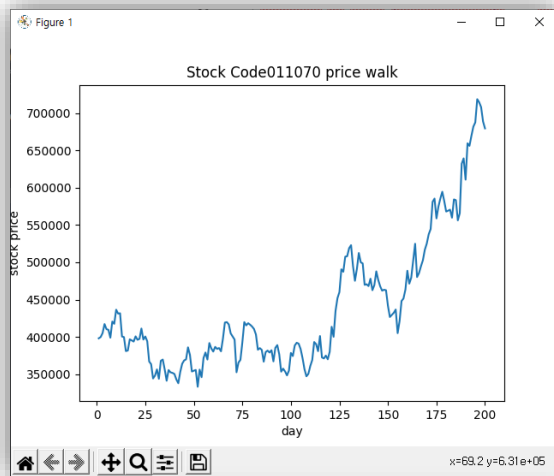gr2.axhline(y=3.14, color="r", linestyle="--")

plt.style.use('seaborn-whitegrid')
plt.show()
```

Example3: Stock Prediction

Use the closing price of stocks to predict the future

Example3: Stock Prediction

1. Load Stock Close Data

2. Load Data Standard Deviation

3. Price+ Price* random(0,….,Standard Deviation)

4. 3 Step Loop

5. Result

Example3-Source: Stock Prediction

```python
import statistics as stat
import matplotlib.pyplot as plt
import random as rand

DataCount = 1000
RepeatCount = 30
DayCount = 101
pastPricesce = [59400,60000,59600,59200,59400,60200,61800,62000,60400,62900,61900,62400,62700,61400,61800]

def randomWalk(stdev,pastPrices):
    days = [i for i in range(1,DayCount)]
    prices = []
    price = pastPrices[-1]
    for i in range(1,DayCount):
        price = price + price*rand.normalvariate(0,stdev)
        prices.append(price)
        print(price)
    return([days,prices])
history_close_prices = rand.sample(range(10000,15000),DataCount)

relative_prices = []

for i in range(0,len(history_close_prices)):
    if i == 0:
        pass
    else:
        relative_prices.append((history_close_prices[i]-history_close_prices[i-1])/(history_close_prices[i-1]))

std_prices = stat.stdev(relative_prices)

prices = randomWalk(std_prices,history_close_prices)

for i in range(0,RepeatCount):
    prices = randomWalk(std_prices,history_close_prices)
    plt.plot(prices[0],prices[1])
plt.title("Simulation")
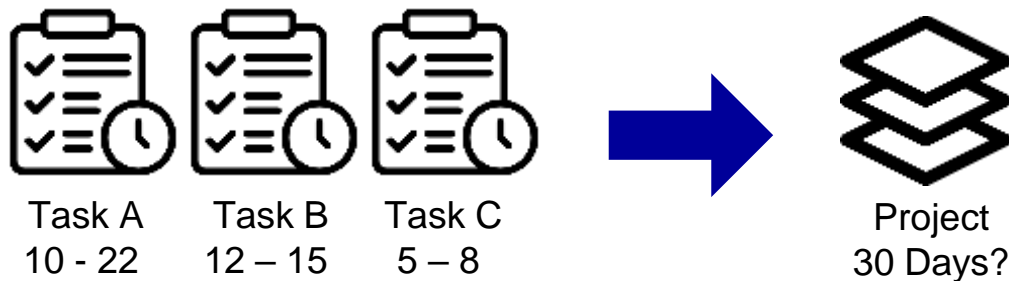plt.xlabel("day")
plt.ylabel("stock price")

plt.show()
```
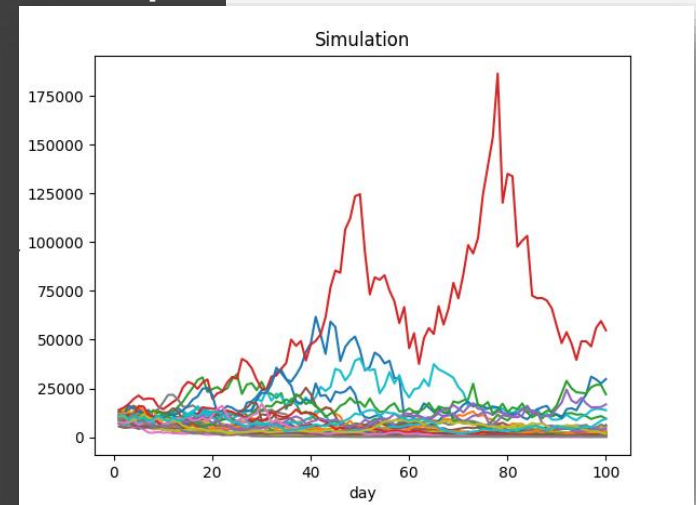
Homwork1: Project Complete Cal

What is the average probability that the project will end in 30 days?

Task A = 10 – 22, Task B = 12 – 15, Task C = 5 – 8

Project = A + B + C

Task A   Task B   Task C   Project
10 - 22   12 – 15   5 – 8   30 Days?

Homwork2 : 5 Random Cards

This program will run a simulation of being dealt 5 random cards from a deck of 52.

What is the probability of being dealt all hearts?

(1 thru 13 numbers are hearts)
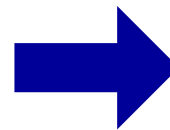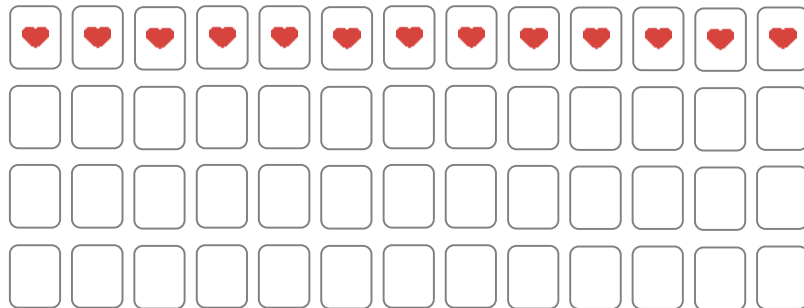
Probability
Heart = 5

Problems with Monte Carlo

- It has the same problems as any simulation.
  - If the model is wrong, the results may be wrong.

- It has the same problems as any statistical method.
  - Even a small bias in choosing the random sample to measure can affect the results.

- At best, you can say that you probably have the correct answer.

Chapter: Simulation

we've learned a lot!

- ## Goal of Simulation
  - A simulation is the imitation of the operation of real-world process or system over time.
  - Simulation is used before an existing system is altered or a new system built,
- ## Differences between systems and models
- ## Types and Developments of Models
  - Verification, Validation
- ## Monte Carlo Simulation

Homwork1: Solution

```python
import random as rand
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt

SimulationCount = 1000000

SimulationPack = []

A_DAY_PACK = []
B_DAY_PACK = []
C_DAY_PACK = []
T_DAY_PACK = []
count = 0

for i in tqdm(range(1,SimulationCount)):
    A_DAY = rand.randrange(10,23)
    B_DAY = rand.randrange(12,16)
    C_DAY = rand.randrange(5,9)

    SimulationPack.append([A_DAY,B_DAY,C_DAY,A_DAY+B_DAY+C_DAY])

for dayPACK in SimulationPack:
    if dayPACK[3] <= 40:
        A_DAY_PACK.append(dayPACK[0])
        B_DAY_PACK.append(dayPACK[1])
        C_DAY_PACK.append(dayPACK[2])
        T_DAY_PACK.append(dayPACK[3])
        count+=1

simulationdf = pd.DataFrame(SimulationPack, columns=['A_DAY', 'B_DAY', 'C_DAY', 'Total'])

plt.bar(simulationdf.groupby(['Total']).size().to_dict().keys(),simulationdf.groupby(['Total']).size().to_dict().values(), width=0.4)
plt.show()
```
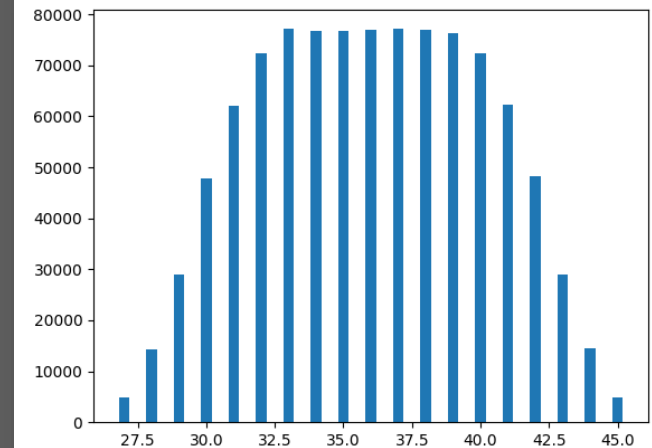
Homwork2 : Solution

```
import random as rand
from tqdm import tqdm

number_run = 10000000
MAX_CARDN = 52
NUM_DRAWN = 5

AllHeart = 0
for i in tqdm(range(1,number_run)):
    count = 0
    randPack = rand.sample(range(1,MAX_CARDN),NUM_DRAWN)
    for number in randPack:
        if number <= 13:
            count += 1
        if count == NUM_DRAWN:
            AllHeart += 1

print("ALLHEART = "+ str(AllHeart))
print (AllHeart/number_run)
```

```
100%|                                        | 9999999/9999999 [00:26<00:00, 374675.96it/s]
ALLHEART = 5371
0.0005371
```