Simulation using Python

Jayendran Venkateswaran
Industrial Engineering & Operations Research
Indian Institute of Technology Bombay

Agenda

- Monte Carlo Simulation
- Simulation of Dynamic Systems
- Optimisation using Simulation

Monte Carlo Simulation

Monte Carlo Simulation

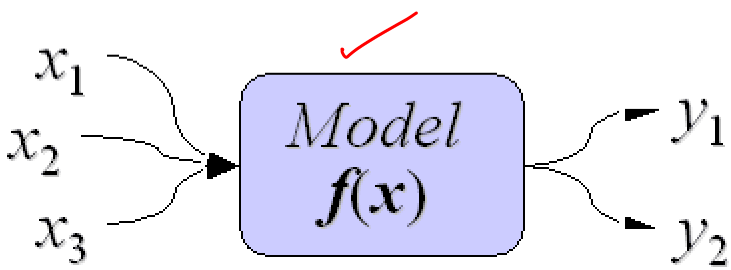
- Any simulation that uses random numbers
  - Very broad, includes all stochastic simulations

More restrictive definition

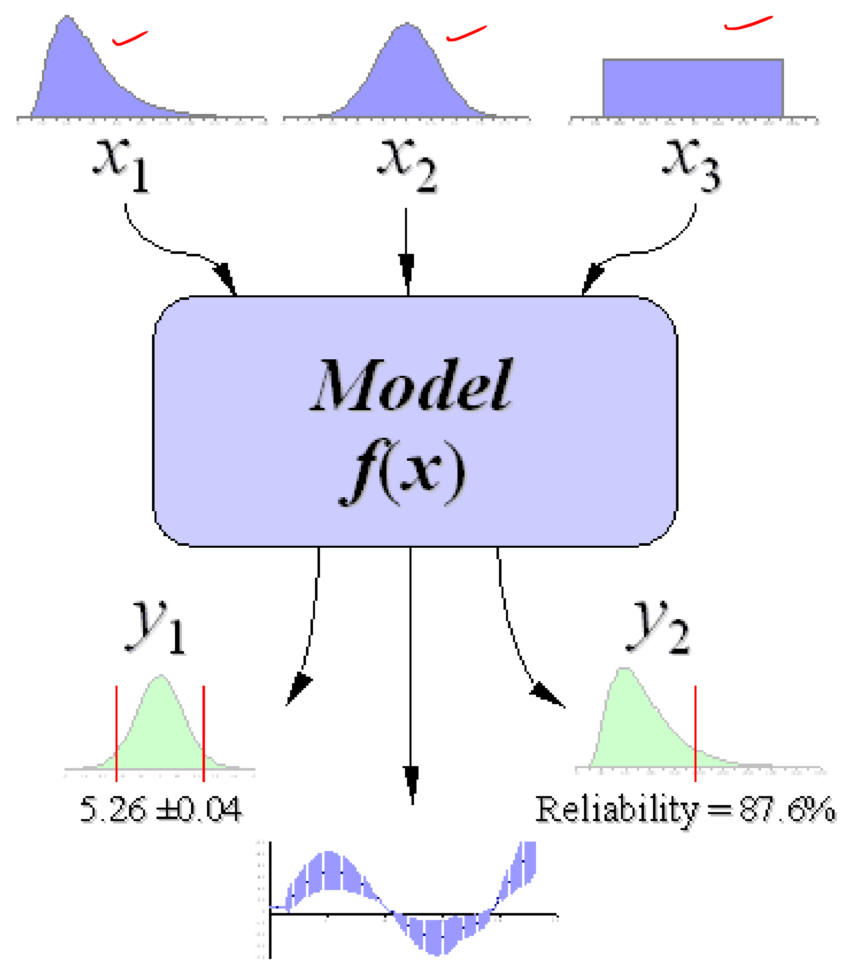
- Monte Carlo is a scheme employing random numbers, i.e. U(0,1) random variates, which is used for solving certain stochastic or deterministic problems where the passage of time plays no role!
  - Monte Carlo simulations are *static* rather than dynamic

MCS basics

- Deterministic model maps a set of input variables to a set of output variables



- Stochastic uncertainty propagation

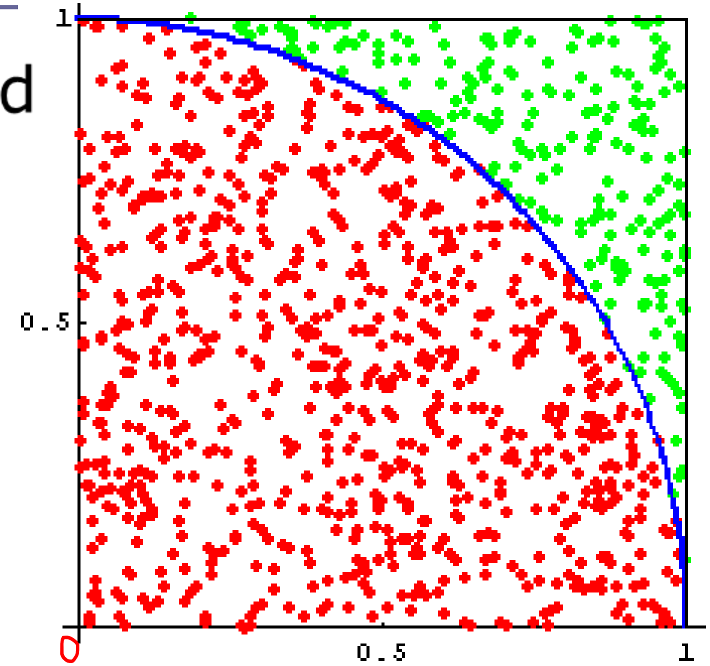


Source: http://www.vertex42.com/ExcelArticles/mc/MonteCarloSimulation.html

Example PI: Compute value of π

- Consider a quarter-circle inscribed inside a unit square
- $\text{Area}_{\text{quarter-circle}} = \pi/4$
- $\text{Area}_{\text{square}} = 1$
- $\text{Area}_{\text{region}} \sim$ Points inside region
  - $\pi \cong$ 4*points in circle/ Total points

quarter



$$\frac{\text{Area}_{\text{quarter circle}}}{\text{Area}_{\text{square}}} \simeq \frac{\text{Points in Circle}}{\text{Points in Square}}$$

$$\frac{\pi/4}{1} \simeq \frac{\text{Points in Circle}}{\text{Total points}}$$

- Set-up a MCS to compute π.
  - Generate (x, y) coordinate where $x, y \in [0,1]$
  - If $(x^2 + y^2 < 1)$ then
    - Count point inside quarter-circle
  - Repeat above two steps n times.
  - $\tilde{\pi} \approx 4 * points\ in\ circle / n$

Compute π for various values of *n*

SimPython2-class1.ipynb

- Estimate PI
  - Solved example;
  - Vary N & run multiple times; observe results
- Project planning
  - Solved example
  - Vary N & run multiple times; observe results
- A rat in a trap (Markov chain example)
  - Partially complete: You need complete this
- Newvendor Model
  - To Do

Simulation of Dynamic Systems

Simulation of Dynamic Systems

- Dynamic systems are often represented as differential equation (ODEs)
  - We can simulate them by numerically exercising the ODEs
- Using odeint of scipy
- odeint() function, takes as input
  - The set of equations as a function definition
  - Initial values
  - Timeline or timesteps
  - Parameter values
  - etc

General Structure of Simulation Model using odeint()

- Inputs
- Initial conditions
- Model (with the ODEs)
- Timeline
- Simulate using odeint function
- Plot the results

Example: Population dynamics

- The rate of change of population P(t) of a region is affected by its birth rate and death rate as follows

$$\frac{dP}{dt} = Births - Deaths = bP - dP$$

, b & d are known constants

Inputs→

```
b=0.02
d=0.05
```

Variable name

Initial values→

```
P0 = 100
```

Model with ODEs→

```
def popModel(P, t, b, d):
    dP_dt = b*P - d*P
    return dP_dt
```

Variable name

#First input is Equations/ state, Time, then optional input parameters

timeline→

```
t = np.linspace(0, 100, 1000)
```

Simulate with odeint→

```
Pop = odeint(popModel, P0, t, args=(b, d))
```

Results→

```
plt.plot(t, Pop)
```

SimDynamicModelsclass.ipynb

- Population Dynamics
  - Solved example
- Simple Harmonic Motion
  - Solved example
- The real non-linear simple pendulum
  - Partially complete. You need to complete this
- SIR epidemics model
  - Solved Example
- SIRS epidemics model
  - To Do

Optimisation using Simulation

Optimisation using Simulation

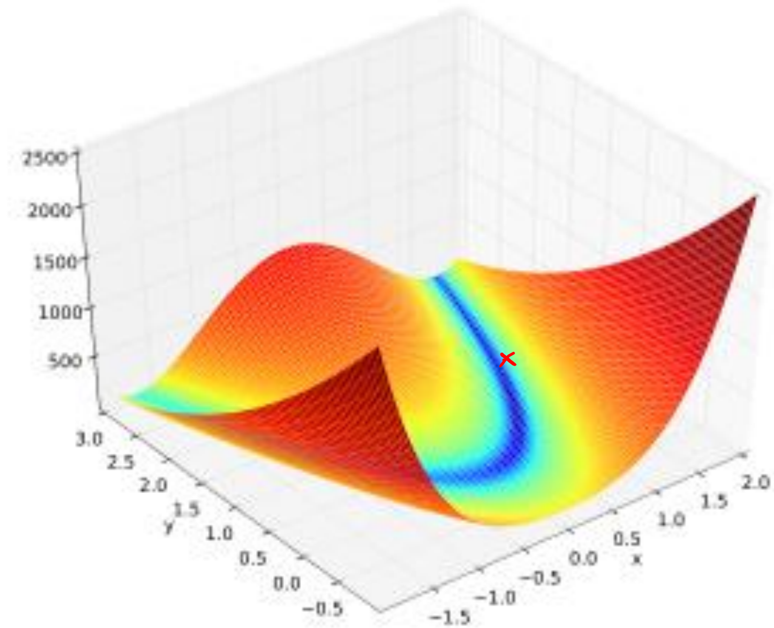
- We typically solve optimisation problems using general mathematical programming, or heuristics
- For NP-Hard problems, or non-linear problems it may be difficult to get the exact solution and hence we may settle for 'good enough' solutions.
- Now, can we simply randomly sample the search space, and by chance, get (near) optimum solution?

Example: Rosenbrock's function

- Minimize the Rosenbrock's function

$$f(\mathbf{x}) = \sum_{i=1}^{N-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \quad \text{where} \quad \mathbf{x} = [x_1, \ldots, x_N] \in \mathbb{R}^N$$

- Let N=2. The optimum is at (1,1) with value 0.



- Let's randomly sample the solution space and see if we can near optimum solutions
    1. Randomly choose a solution point
    2. Evaluate the function
    3. Repeat 1 and 2 for N samples
    4. Report the minimum obj and corresponding solution obtained

SimOptClass.ipynb

- Rosenbrock Function
  - Solved example

To Do: You can find some single objective unconstrained test functions at Wiki page

1. Through simulation, find the optimum solution of any one of the function: Beale or Goldstein-Price or Booth

2. 'Optimise' either Himmelblau's function OR Cross-in-Tray function. These functions have 4 alternate solutions. Do 20 sets of 'simulation-optimisation' runs, with N ~= 200000. Compute the number of times we are close to a particular known solution.