

# Monte Carlo Simulation: Variance Reduction

SYS-611: Simulation and Modeling

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

## Agenda



- 1. Buffon's Needle Model Implementation
- 2. Variance Reduction Methods

Reading: S.M. Ross, "Variance Reduction Techniques," Ch. 9 in Simulation, 2012.



## **Buffon's Needle Implementation**

#### **Excel Implementation (1)**



 Define columns for the elementary random variables (using process generators)

$$d \sim U(0, t/2) \Rightarrow f(d) = \frac{2}{t}, F(d) = \frac{2 \cdot d}{t} \Rightarrow d = \frac{r \cdot t}{2}$$
$$\theta \sim U(0, \pi/2) \Rightarrow f(\theta) = \frac{2}{\pi}, F(\theta) = \frac{2 \cdot \theta}{\pi} \Rightarrow \theta = \frac{r \cdot \pi}{2}$$

4	А	В	С	D	E	F
1	r1	d	r2	theta		
2	0.566645	=A2*3/2	0.358502	0.563134		
3	0.483976	0.725964	0.795111	1.248957		
4	0.181968	0.272952	0.729009	1.145125		
5	0.888497	1.332746	0.801691	1.259293		
6	0.215872	0.323808	0.109275	0.171648		
7	0.020329	0.030493	0.873607	1.372259		

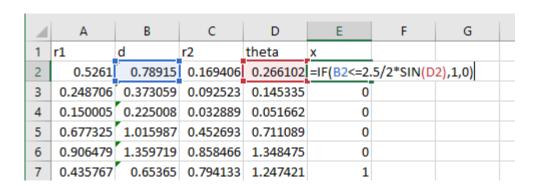
$\mathcal{A}$	Α	В	С	D	E	F
1	r1	d	r2	theta		
2	0.733606	1.100409	0.80425	=C2*PI()/2		
3	0.971178	1.456768	0.316316	0.496868		
4	0.584758	0.877136	0.22941	0.360356		
5	0.5151	0.772651	0.110767	0.173992		
6	0.557662	0.836492	0.286458	0.449967		
7	0.657947	0.98692	0.03508	0.055104		

# **Excel Implementation (2)**



Define a column with the derived state variable

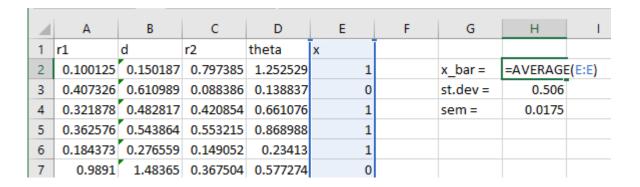
$$X = \begin{cases} 1, & \text{if } d \leq \frac{l}{2} \sin \theta \\ 0, & \text{otherwise} \end{cases}$$



# **Excel Implementation (3)**



- Fill down the equations to generate samples
- Compute statistics and plots as needed



#### Python Implementation (1)



 Define functions to sample the elementary random variables (using process generators)

$$d \sim U(0, t/2) \Rightarrow f(d) = \frac{2}{t}, F(d) = \frac{2 \cdot d}{t} \Rightarrow d = \frac{r \cdot t}{2}$$

$$\theta \sim U(0, \pi/2) \Rightarrow f(\theta) = \frac{2}{\pi}, F(\theta) = \frac{2 \cdot \theta}{\pi} \Rightarrow \theta = \frac{r \cdot \pi}{2}$$
import numpy as np
def generate\_d():
 return np.random.rand()\*3/2.
def generate\_theta():
 return np.random.rand()\*np.pi/2

# Python Implementation (2)



Define a function for the derived state variable

$$X = \begin{cases} 1, & \text{if } d \leq \frac{l}{2}\sin\theta \\ 0, & \text{otherwise} \end{cases}$$

```
def generate_x():
    d = generate_d()
    theta = generate_theta()
    if d <= 3/2.*np.sin(theta):
        return 1
    else:
        return 0</pre>
```

# Python Implementation (3)



- Iterate over a list to generate samples
- Compute statistics and plots as needed

```
samples = [generate_x() for i in range(30)]
print np.mean(samples)
print np.std(samples,ddof=1)
import scipy.stats as stats
print stats.sem(samples)
```



#### **Variance Reduction Methods**

#### Variance Reduction Methods



- Basic Monte Carlo simulation requires many samples for accurate estimates (can be improved)
  - Antithetic variables: leverage correlation in observations to get better estimates of population mean
  - Control variables: replace the estimation of unknown quantity with the difference between two quantities, one of which has a known expected value
  - Importance sampling: purposefully draw samples from a different (better) distribution and correct for known bias
  - Stratified sampling: purposefully draw samples from segmented regions of the sample space

#### **Antithetic Variables: Theory**



 Antithetic variables help make more accurate estimates of expected value or population mean

$$\bar{X} \sim \text{normal}(\mu, \frac{\sigma}{\sqrt{N}})$$

• Reduce  $\sigma$  to improve the accuracy of estimates

$$X = \frac{X_1 + X_2}{2}$$
,  $Var(X) = \frac{Var(X_1) + Var(X_2) + 2 \cdot Cov(X_1, X_2)}{4}$ 

 Carefully pick X<sub>1</sub> and X<sub>2</sub> to have negative correlation while not violating the quality of the estimate of X

#### **Example: Antithetic Distance**



Basic process generator samples d as:

$$d = \frac{r \cdot t}{2}$$

Antithetic form would decompose into two parts:

$$d = \frac{d_1 + d_2}{2}$$

Simplest negative correlation from re-using r:

$$d_1 = \frac{r \cdot t}{2}, \qquad d_2 = \frac{(1-r) \cdot t}{2}$$

#### **Antithetic Buffon's Needle**



Generate two random values for distance and angle:

$$r_1 \sim U(0,1), \qquad r_2 \sim U(0,1)$$

Generate antithetic distance and angle samples

$$d_1 = \frac{r_1 \cdot t}{2}$$
,  $d_2 = \frac{(1 - r_1) \cdot t}{2}$   $\theta_1 = \frac{r_2 \cdot \pi}{2}$ ,  $\theta_2 = \frac{(1 - r_2) \cdot \pi}{2}$ 

Compute antithetic derived variable:

$$X_1 = 1 \text{ if } d_1 \le \frac{l}{2} \sin \theta_1 \text{ else } 0, \qquad X_2 = 1 \text{ if } d_2 \le \frac{l}{2} \sin \theta_2 \text{ else } 0$$
 
$$X = \frac{X_1 + X_2}{2}$$

#### Impact of Antithetic Variable



#### Using Buffon's Needle Monte Carlo simulation:

• Basic Monte Carlo simulation (N = 10000):

$$\bar{x} = 0.533$$
,  $s_x = 0.499$ ,  $SEM = 0.005$ 

• Antithetic Monte Carlo simulation (N = 10000):

$$\bar{x}=0.527$$

$$s_x = 0.156$$
,

$$\bar{x} = 0.527$$
,  $s_x = 0.156$ ,  $SEM = 0.0016$ 

• Antithetic Monte Carlo simulation (N = 850):

$$\bar{x} = 0.522$$
,

$$s_x = 0.147$$
,

$$\bar{x} = 0.522$$
,  $s_x = 0.147$ ,  $SEM = 0.005$