



Monte Carlo Simulations

2014 ISPA/SCEA National Conference

Eric Druker, Blake Boswell
Booz Allen Hamilton

This document is confidential and is intended solely for the use and information of the client to whom it is addressed.

Outline

- ▶ Introduction
 - What is a Monte Carlo Simulation?
- ▶ History of Monte Carlo Simulations
- ▶ The Inverse CDF Technique
- ▶ Correlation
- ▶ Building a Monte Carlo Simulation
 - Excel Demo

Introduction

- ▶ *“Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results” ~ [Wikipedia](#)*
- ▶ In the arena of cost risk analysis, Monte Carlo simulations are used to produce probabilistic cost estimates
 - Also known as S-Curves
- ▶ Cost risk analysis generally relies on the use of probabilistic distributions to describe the uncertainty surrounding a cost estimate
- ▶ To find the top level distribution of potential costs, the combined effects of multiple probabilistic distributions must be uncovered
 - Monte Carlo simulations are the most commonly accepted method by which this can be accomplished
 - Method of Moments can also be used, yet there are questions regarding its accuracy that must be answered before the results can be verified¹

¹*Taking a Second Look: The Potential Pitfalls of Popular Risk Methodologies.* Druker, Coleman, Braxton, Leonetti. SCEA 2007, NASA PM Challenge 2008

History of Monte Carlo Simulations

Buffon to Laplace

▶ Buffon's Needle Experiment

– French mathematician Georges-Louis Leclerc, Comte de Buffon (1707-1788) created an experiment to estimate Pi.

- The probability of a needle of length L being dropped from above, and intersecting two parallel lines of distance D apart was known to be equal to $2L/(\pi \cdot D)$
- Italian mathematician Mario Lazzarini found Pi to within 6 decimals (3.1415929) using this method in 1901
- <http://www.ms.uky.edu/~mai/java/stat/buff.html>

▶ Seeing this method of deriving Pi, French mathematician Pierre-Simon Laplace was inspired to propose a method (now known as the Monte Carlo Method) for numerically evaluating unknown values through the translation of random events.



History of Monte Carlo Simulations

Fermi and the Manhattan Project

- ▶ In the 1930s, Enrico Fermi used a random method to examine neutron diffusion
- ▶ The name Monte Carlo method was coined by mathematicians working on the Manhattan Project
- ▶ While contemplating his probability of winning a game of solitaire (as mathematicians often do), Stanislaw Ulam wondered if he could use the new “fast” computers to simply create random card draws, and then see if they were winning draws
- ▶ During the 1950s, the same method was expanded by scientists at Los Alamos in the development of the hydrogen bomb

The Inverse CDF Technique

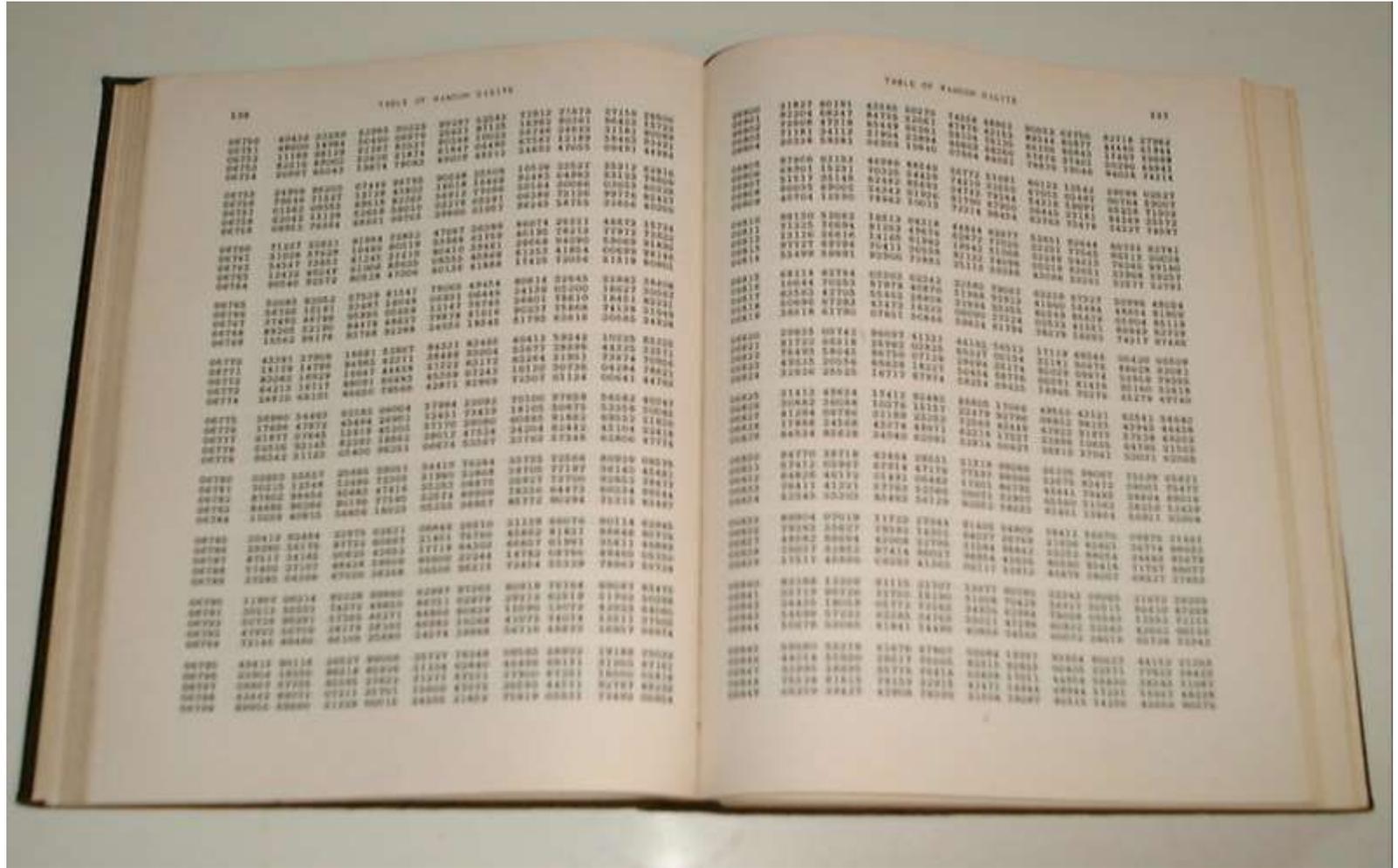
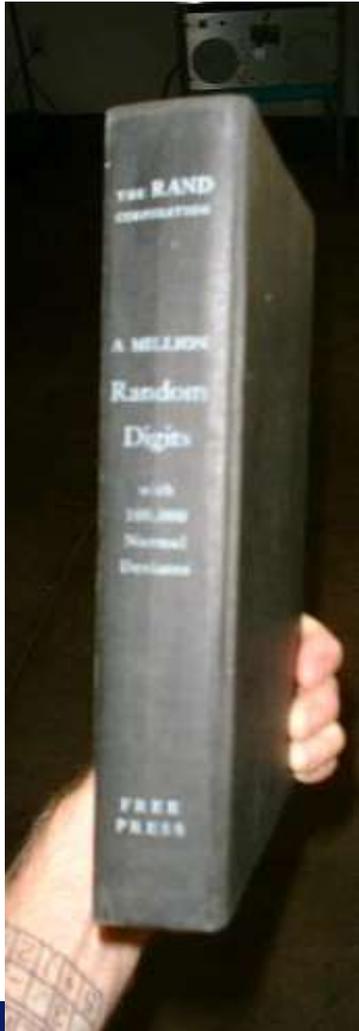
- ▶ One of the most common techniques used in Monte Carlo simulations is the Inverse CDF Technique
 - For this technique, events in a simulation must be assigned some sort of Probability Distribution Function
- ▶ The Inverse CDF Technique is then used to map a $U(0,1)$ random number to a point on the desired CDF
 - Every simulation must contain some sort of Uniform $(0,1)$ random number generator
 - In Excel this function is “=RAND()”
 - There is an entire area of computer science dedicated to the production of “the most random” numbers
- ▶ The random number generator in Excel 2003 (and later) passes the DIEHARD tests as well as additional tests developed by the NIST¹
 - This is not true of the random number generators in previous versions of Excel

¹<http://support.microsoft.com/kb/828795>

Random Number Generation

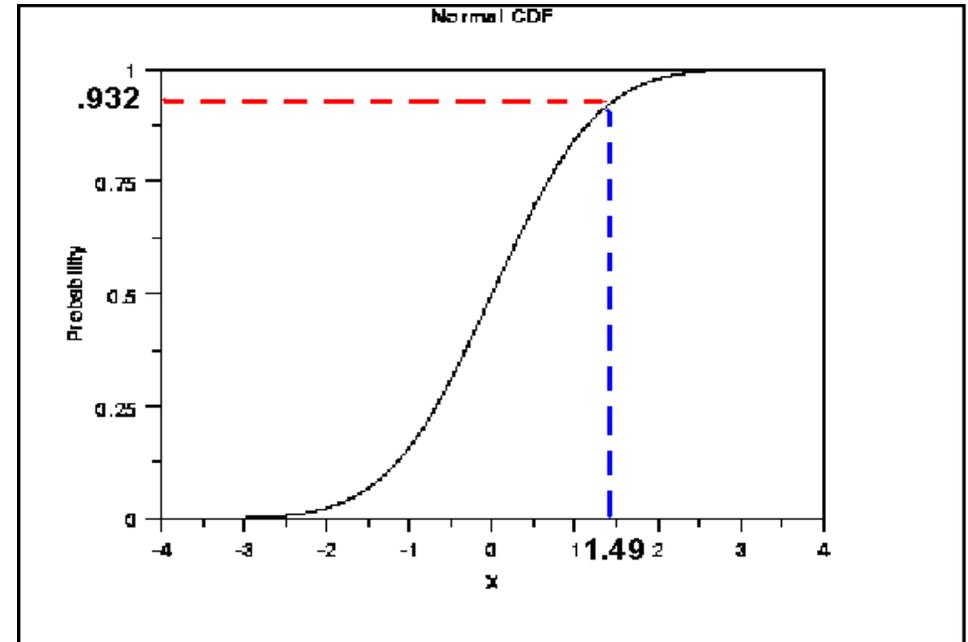
- ▶ Before computing was common place, the difficulty in simulations was that no one had a way of producing huge quantities of random numbers
- ▶ In 1955 RAND published a book: A Million Random Digits with 100,000 Normal Deviates
- ▶ This book contained 1,000,000 numbers drawn from a Standard Normal Distribution
 - The methods used to produce these numbers were advanced for the day, yet would be considered primitive now
- ▶ Pictures of the book are on the next slide

A Million Random Digits With 100,000 Normal Deviates



Standard Normal Example

- ▶ If the RAND() function returns a value of .932, it will be plugged into the inverse function of the CDF, and the value 1.49 will be returned as the appropriate percentile of the PDF



Creating a Monte Carlo Simulation

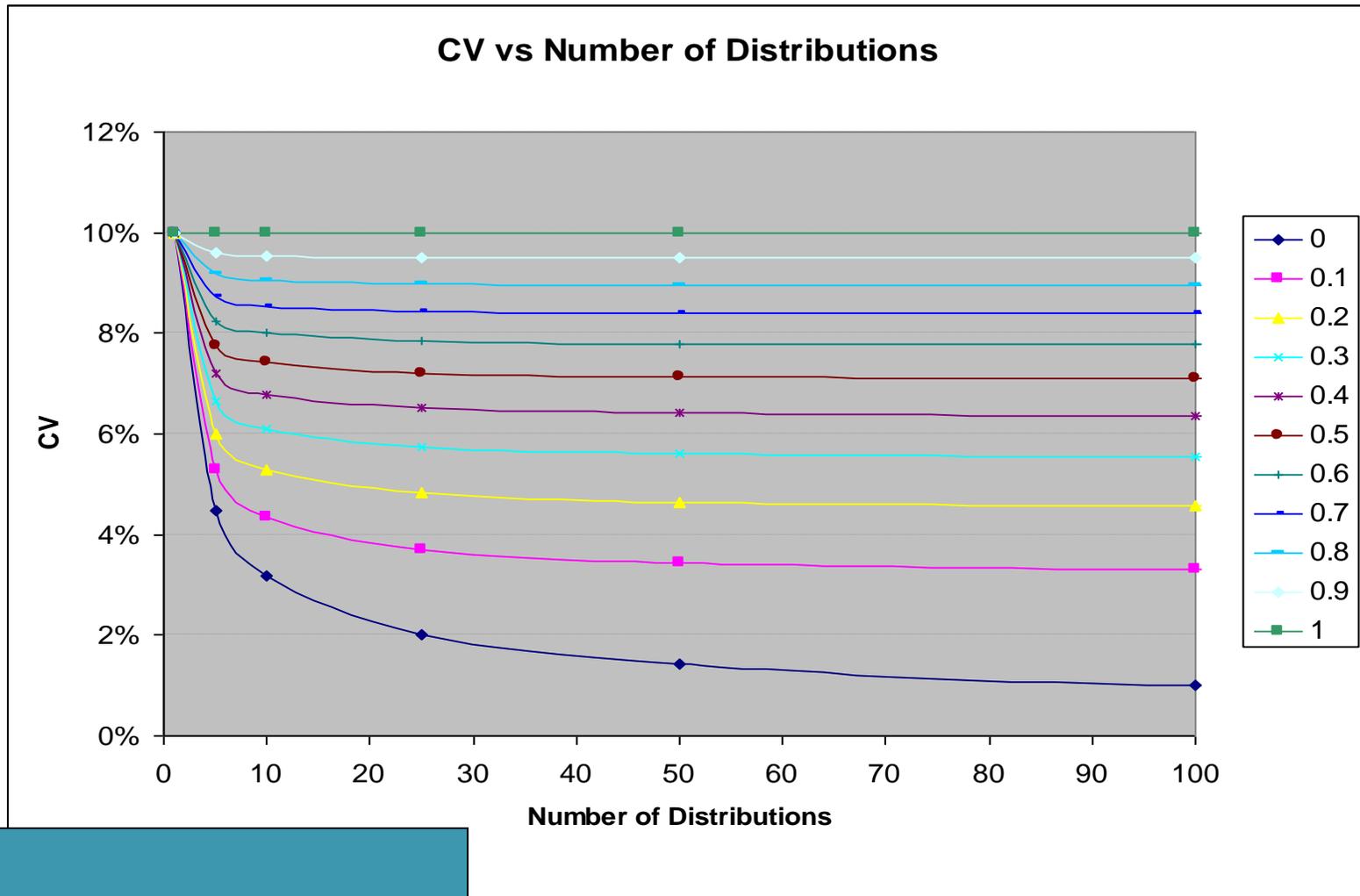
- ▶ What follows are the general steps in developing a Monte Carlo simulation for cost risk
 1. Determine what parameters will have uncertainty associated with them
 - Inputs Based Analysis – Cost *Drivers* have uncertainty distributions
 - Outputs Based Analysis – *Costs* have uncertainty distributions
 2. Determine the distributions to be used for each of the parameters
 - Generally developed by applying statistical analysis and curve-fitting techniques to historical data
 3. Build a cost model that relates the parameters to a final cost
 - If a cost model already exists, the Monte Carlo simulation can be built into it (as will be demonstrated in the upcoming analysis)
 4. Place in a random number generator; one random number must be generated per simulation run per distribution
 5. Replace fixed cost values with their associated distributions
 - In particular, replace the value with the inverse of the CDF
 - The value for each run should be a function of the random number associated with that distribution
 6. Functionally correlate costs where appropriate
 - Costs estimated as a function of other costs (G&A, SEPM, OH, etc.) should be linked to their appropriate cost drivers
 7. Run the simulation a large number of times, storing the total cost for each run
 - In Excel, this can be done using a VBA macro
 8. Sort the results from all the runs and chart against their cumulative probability

Correlation

- ▶ One of the most difficult things to account for in a Monte Carlo simulation is correlation between the independent variables
 - If correlation is not accounted for, the Coefficient of Variation of the top level distribution will be artificially shrunk
 - This is commonly referred to as the “square root of n problem”: as the number of uncorrelated distributions increases, assuming all standard deviations equal, the top level coefficient of variation will decrease inversely as the square root of n , the number of distributions
- ▶ Example:
 - Assume an inputs based risk analysis is being performed at the outset of the development of a missile system
 - Cost is being estimated as a function of warhead weight and velocity
 - When examining the data, it is found that weight growth and change in velocity from the initial technical assessment are negatively correlated
 - As weight grows, velocity decreases
 - Whatever variables are used to express the uncertainty in the projections for warhead weight and velocity must be correlated
- ▶ The main method by which correlation can be accounted for is the Lurie-Goldberg method¹
 - This method correlates uncertainty distributions by correlating the random number draws
 - It also preserves the marginal distributions

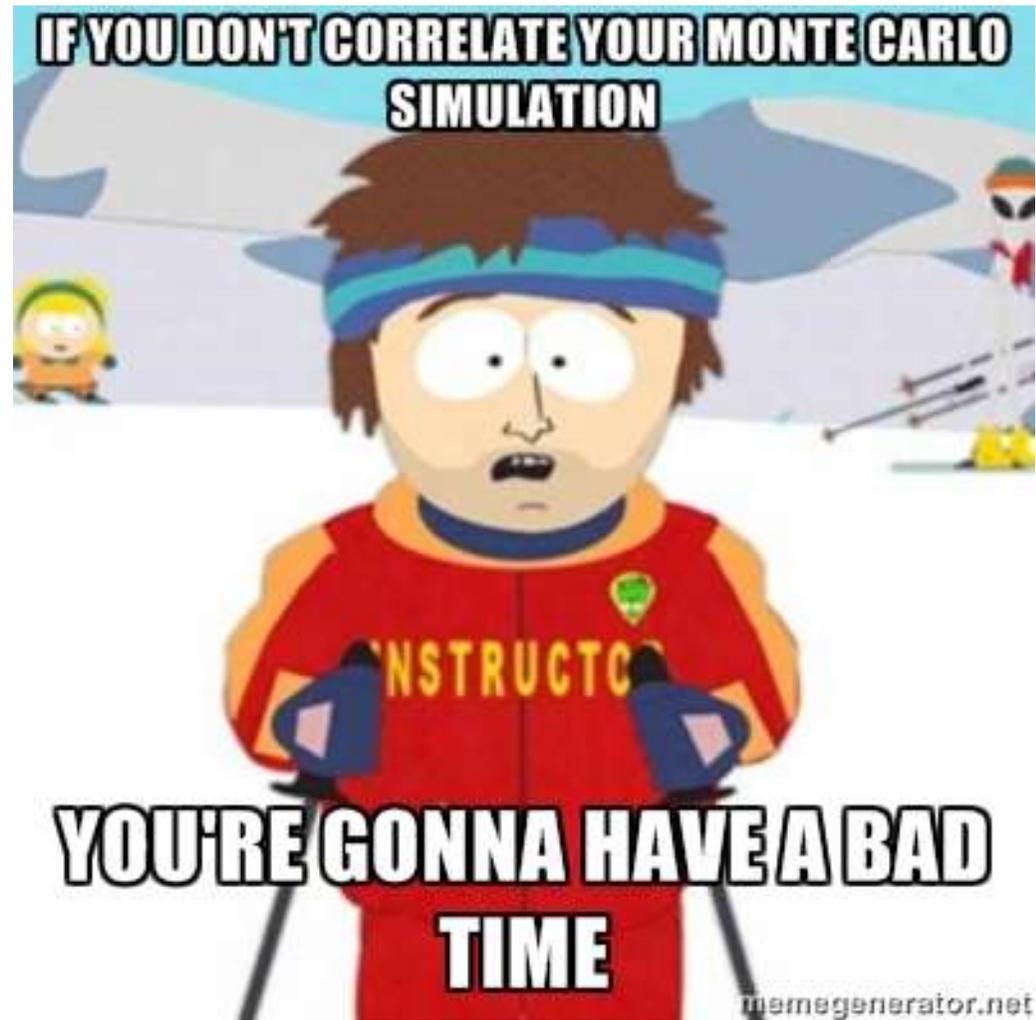
¹Goldberg, Matthew S, Lurie, Phillip M. *Correlating Random Variables*, 32nd DoDCAS, Williamsburg, VA. February 1999

“Square Root of N Problem”



Assumes iid Normal Distributions w/ CV of 10%

Correlation



Variance Reduction

▶ In terms of Monte Carlo methods, variance reduction refers to increasing the precision of the results

▶ There are several methods by which this can be accomplished:

– Increasing the number of runs of the simulation (the brute force way)

- Make n “pilot” runs
- Calculate the mean and variance of the values for the desired measure

This is a rough value that will be used to determine the necessary number of runs

Note that determining the variance of the results is at best only approximate, because it generally takes an order of magnitude more runs to determine the variance than it does the mean

- Make an additional $n^* - n$ runs where

$$n^* = \min_{i \geq n} \left\{ t_{i-1, \alpha/2} \frac{S}{\sqrt{i}} \leq h^* \right\}$$

- And h^* is the desired halfwidth for the desired $(1 - \alpha) * 100$ confidence level

– Latin Hypercube Sampling (or stratified sampling)

- Force observations to fall into each decile of the distribution
- Allows the simulation to converge on the true distribution with less runs