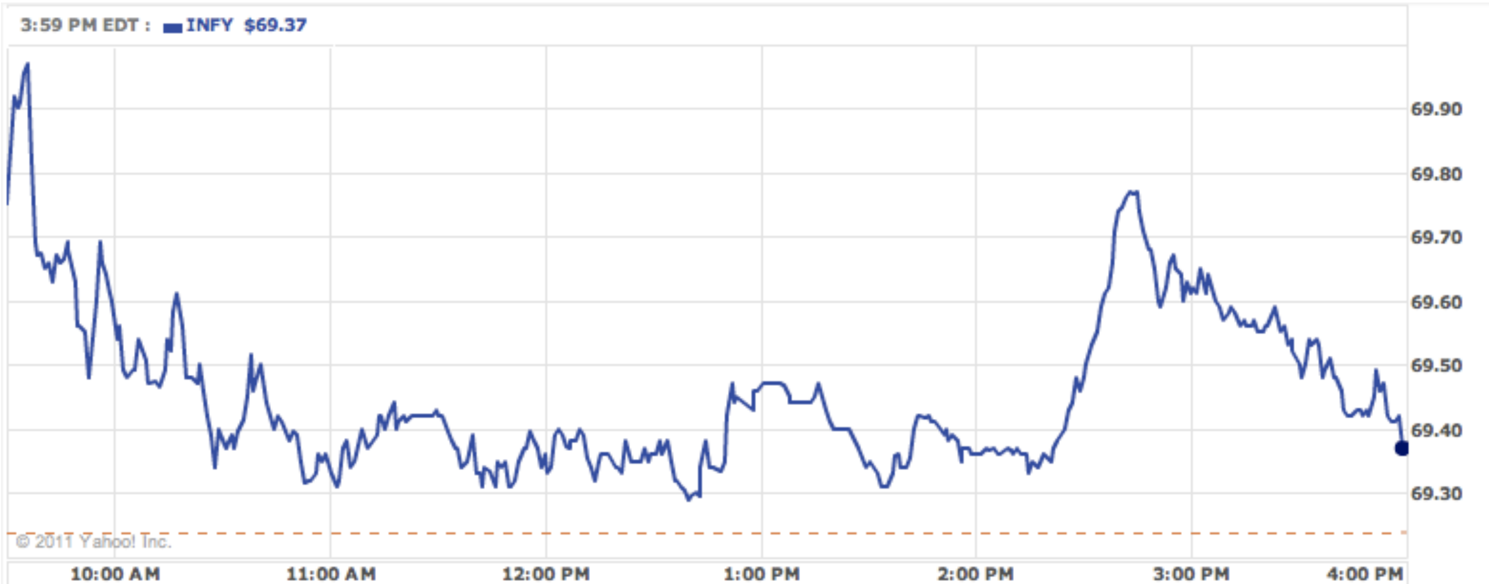
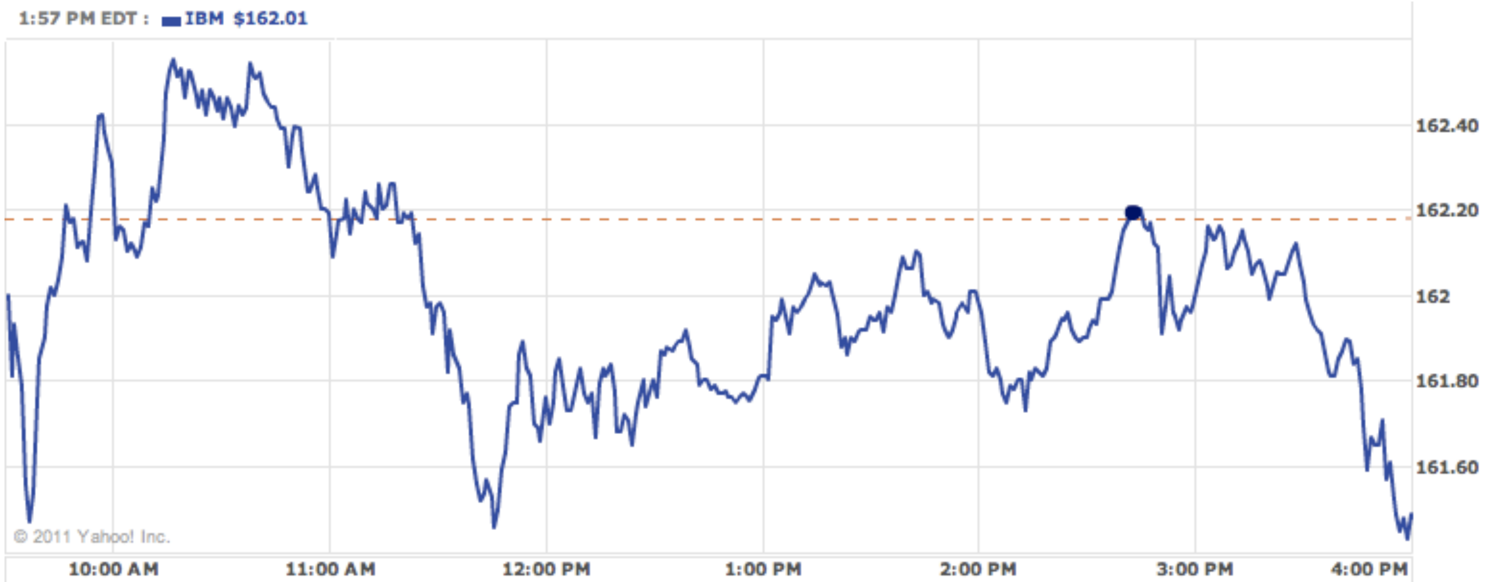
Random Numbers

IDC 102
Chaitanya Athale

Stock Market Values



Infosys



IBM

Yahoo.com

Outline

- Randomness
- Stochastic simulations
- Random number types
- How to generate a random number
- Transformations
- Quasi random numbers

REFERENCES

- Scientific Computing: An Introductory Survey
  http://www.cse.illinois.edu/heath/scicomp/
- The Nature of Mathematical Modelling- N. Gershenfeld.

Why do we need them?

- Games of chance
- Cryptography
- Statistical data sampling
- Computer simulations of diffusion, reaction, explosions
- Social simulations
- Systems with many coupled degrees of freedom with uncertainty in inputs
- Data optimization
- Evolutionary algorithms (genetic algorithms)

Anything where unpredictable outcomes are desired

Radioactive Decay

Scintillation counting

Radioactive decay

Beta particle/electron

Scintillant

Decay of fluor to ground state

Photons

Photomultiplier tube

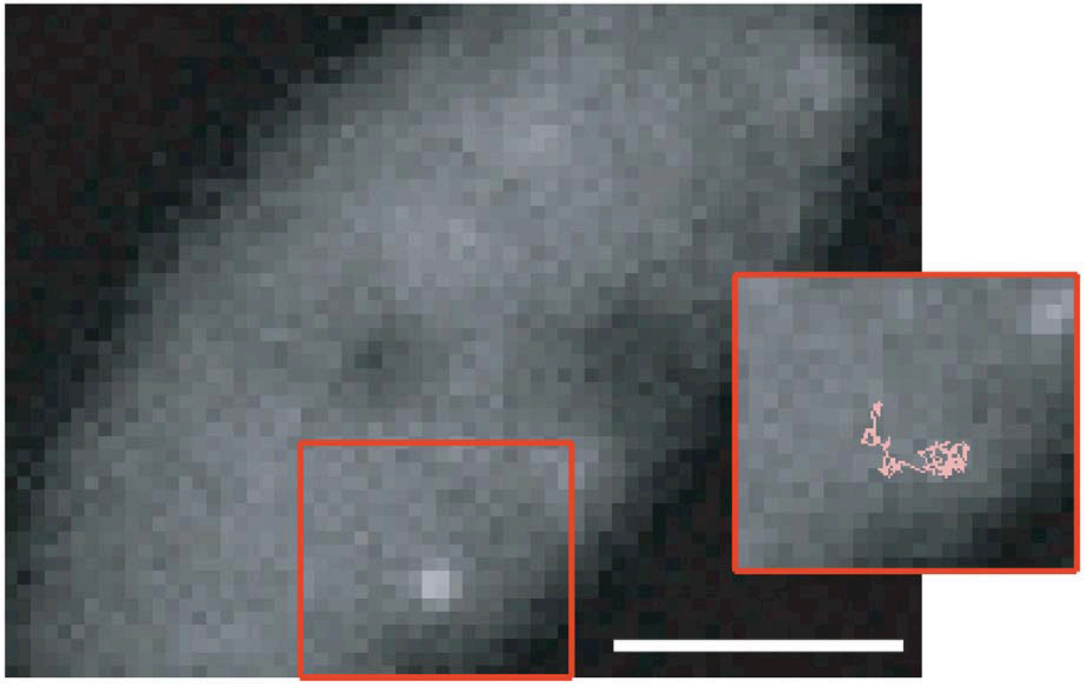
Electrons, current

Evolutionary Algorithms

- Evolutionary algorithms inspired by biological evolution
  - Reproduction
  - Random mutation
  - Recombination
  - Selection
- Artificial Evolution
- No assumptions about fitness landscape
- Computationally complex
- Fitness function- selection of offspring for higher fitness
- Ex: Genetic algorithms, swarm optimization, stochastic hill climbing

Random Walks (drunken walk)

- On a one-dimensional plane
- Take step left -1, right +1, or 0
- Over time, average displacement?
- Does one move at all?
- Brownian Motion
- Mean squared displacement
- Diffusion



Bancaud (2009)

Properties of a good random number generator (RNG)

- Fast
- As random as possible

Contradiction

Typical computers 10^9 operations/s

Pseudo random number generators (PRNGs)

True random number generators (TRNGs)

Random Variables

- Discrete
- Continuous

Unknown numerical variable that can take on or represent any possible element in sample space

Random Variables

$$X(s)$$

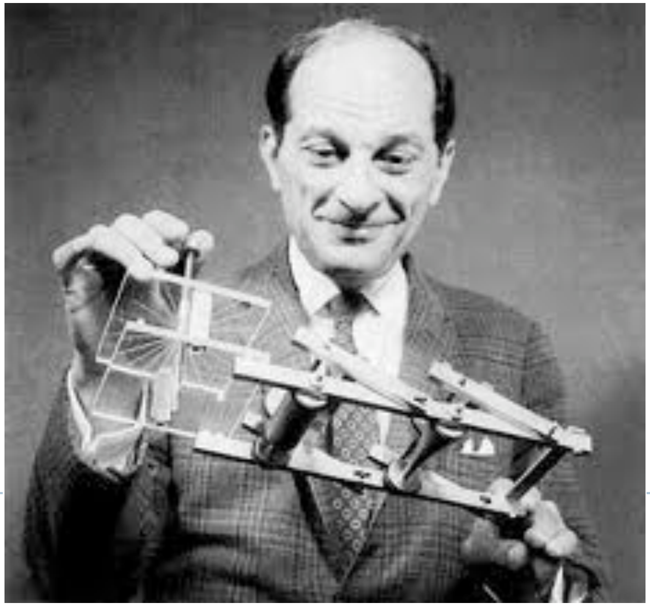
Where

$$s \in S$$

Sample space S in real numbers

Stochastic Simulations

- Mimics behaviour of system by exploiting randomness to obtain statistical sample of random outcomes
- Monte Carlo simulations (1940s)
- Used to study
  - Non deterministic processes
  - Complex deterministic systems that cannot be analytically treated
  - High dimensionality



Stanislaw Ulam

Stochastic simulations

Requirements for stochastic simulations

- Knowledge of probability distribution
- Supply of random numbers for making random choices

- Probability distribution: physical system
  - Diffusion of particle
  - Radioactive decay
- Large number of trials: Probability distribution more accurate with increasing no. of trials

Randomness

- Randomness associated with unpredictability
- No shorter description than itself
- Physical processes: flipping coin, roll of dice, etc.
- Even deterministic systems in chaotic regime due to sensitivity to initial conditions

Repeatability

- Lack of repeatability
- Hard to test
- Independence of trials

How to Generate a Random Number

- Physical processes: examples?
- Computer algorithms: deterministic, output appears random
- Pseudorandom
- Predictable and repeatable (reproducible)
- Finite number possible- eventually repeats

RNGs

Properties of a good RNG

- Random pattern: passes statistical test of randomness
- Long period: goes on as long as possible before repeating
- Efficiency: Executes rapidly and requires little storage
- Repeatability: Produces the same sequence if started with the same initial conditions
- Portability: Runs on different kinds of computers, producing the same sequence

RNGs

- Early RNGs complex
- Mid-square method
- 5341
- Square
- $28\underline{5262}81$
- Take 5262

- Von Neumann (1949)
- For n digit numbers, period length < 8^n
- Sensitive to zeros

Need for simple methods with well understood theoretical basis preferred

Congruential generators

- Congruential random number generators

$$x_k = \left(a x_{k-1} + c\right)\left(\operatorname{mod} M\right)$$

Where a and c are given integers

Starting integer x_0 is called a **seed**

Integer M is approximately the largest integer represented on the machine

Quality of the generator depends on choice of a and c.

Period cannot exceed M

Congruential generators

- Reasonable random numbers only if **a** and **c** chosen very carefully
- Default random numbers with many systems- congruential- some very poor
- Congruential RNGs produce numbers between 0 and M
- Random floating point number uniformly distributed over interval [0,1), random numbers divided by M

Example

Linear Congruential Generator

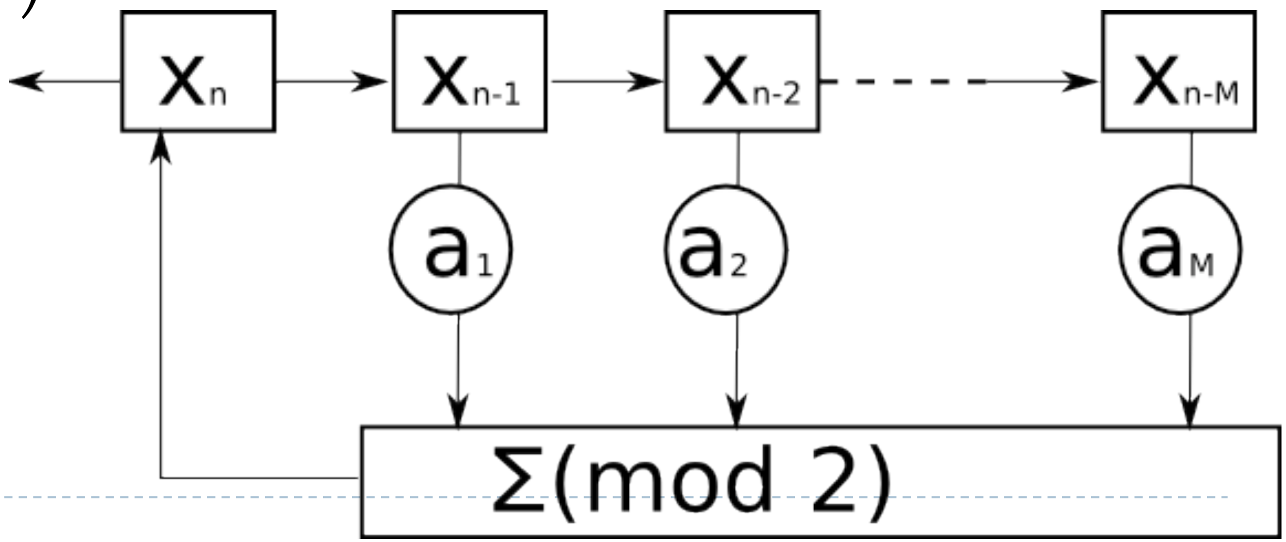
- No. of possibilities set by M
- If x_n is even, x_{n+1} will be odd
- x_n oscillates at every step

- Solution: Degrees of freedom-
  - run multiple, parallel generators and shuffle entries

Linear Feedback

- Linear congruential generator numbers not equally random
- Linear shift feedback registers (LSFR) provide alternative
- Recursion relation

$$x_n = \sum_{i=1}^{M} a_i x_{n-1} (\mathrm{mod}\, 2)$$



- Repeat time 2^n

Fibonacci Generators

- Fibonacci generators produce floating point random numbers on the interval [0, 1) directly as difference, sum, or product of previous values
- Typical example is a subtractive generator

$$x_k = x_{k-17} - x_{k-5}$$

- This generator has lags of 17 and 5
- Lags need to be chosen carefully to generate good subtractive generators
- The method might generate negative numbers- in which case the remedy is to add 1! Interval again [0, 1)

Fibonacci Generators

- Require more storage than congruential. Require special procedures to get started.
- Do not require division to obtain floating point results
- Well designed Fibonacci Generators have very good statistical properties
- Fibonacci Generators have a much longer period than congruential generators, since repetition of one member of sequence does not mean all others will also repeat in the same order

Sampling on Other Intervals

- If random number required to sample other distribution on some interval [a, b), modify values x_k generated on [0,1) by transformation:

$$(b-a)x_k + a$$

to obtain random numbers uniformly distributed on desired interval.

Non-Uniform Distributions

- Sampling non-uniform distributions more difficult
- If cumulative distribution function of probability distribution function (pdf) is invertible with ease, random samples can be generated with desired distribution, by generating uniform random numbers, and inverting them
- Eg.: $$f(t) = \lambda e^{-\lambda t}, t > 0$$

We can take $$x_k = -\log(1 - y_k)/\lambda$$

Where y_k is uniform

- Many important distributions are not easily invertible. Special methods needed.

Normal Distribution

- Important random number distribution- normal with given mean and variance

- Most available routines assume mean = 0, variance = 1

- If other mean μ and variance σ^2 are required, each value x_k produced can be modified by $\sigma \cdot x_k + \mu$

Quasi Random Sequences

- For some applications achieving a coverage of the sampled volume more important than “true randomness”
- “Truly random” sequences show clumping
- Perfectly uniform coverage can be achieved by sample points on a regular grid. Doesn’t scale well for higher dimensions
- Compromise- quasi random sequences

Quasi Random Sequences

- Not random
- Carefully constructed to sample volume and appear random
- Avoid each other
- Eliminate clumping

Example

NEXT

- Random number generators in common use
- Distributions and transformations
- Application of RNGs to scientific simulation problem