Pseudorandom Number Generators

M. Sami Fadali
University of Nevada

Outline
- Are they random?
- How do we generate “random” numbers for simulation?
- How do we generate “random” numbers for experiments?
- MATLAB

Pseudorandom Numbers
- Appear random but are deterministic.
- Needed for experiments and simulations.
- MATLAB

Example
- Obtain a sample realization of a Gauss-Markov process.
- Record of duration $T \gg 1/\beta$.
- Play record in a loop (i.e. periodic).
- Appears random but clearly deterministic.
Spectrum

- Discrete (line) spectrum for a periodic signal.
- Lines of spectrum are $1/T$ apart.
- Envelope of lines.

$$S(j\omega) = E\left\{ \frac{1}{T} |\mathcal{F}[X_T(t)]|^2 \right\} = \frac{2\sigma^2 \beta}{\omega^2 + \beta^2}$$

$$|\mathcal{F}[X_T(t)]|\text{ envelope} = \frac{2\sigma^2 \beta}{\sqrt{\omega^2 + \beta^2}}$$

Shift Register

- Generate binary sequence using a shift register with feedback.
- Each clock pulse:
  1. Shift right.
  2. Add $\text{XOR}(a_{\text{right}}, a_{\text{middle}})$ to left.

Is it random?

- Repeats after $2^n - 1$ bits.
- Almost half ones and half zeros (e.g. $n=10$, 512 ones and 512 zeros).
- Fixed sequence for fixed initial seed and feedback.
- Completely deterministic but appears random.

Plots of Binary Waveform

Seed = \{1, 1, 1, 1, 1\}
Sequence=\{1, 1, 1, 1, 0, 0, 0, 1, 1, 0, \ldots\}
“True” Random Number Generators

Drawbacks of Pseudorandom Generators

1. Not truly random.
2. Hackers can replicate them if they know the seed.

Hardware-based Generators:

1. Gold standard for security
2. Can be bulky, rigid, and expensive to manufacture.

Memristor

- Memristor: true random number generator.
- Used for low power logic.
- See IEEE Spectrum July 2012

Carbon Nanotubes

- Memory cell uses fluctuations in thermal noise to generate random bits.

Uniform Pseudorandom

**rand:** generates arrays of random numbers uniformly distributed in the interval (0,1)

- $Z = \text{rand}(m,n)$ or $Z = \text{rand}([m \ n])$
- returns an $m$-by-$n$ matrix of random entries
- $Z \sim U[0,1]$, uniform over the interval.
- $Y = (\text{max-min}) \cdot Z + \text{min}$ % $Y \sim U[\text{min}, \text{max}]$
- height = 1/(max - min)
**Theorem (Hogg et al., p. 288)**

If the random variable $U \sim \mathcal{U}(0,1)$ then, for any continuous distribution function $F$, the random variable $X = F^{-1}(U) \sim F$.

- Assume that $F$ is monotone increasing to simplify the proof (one-one).
- Use the theorem to generate many distributions using the uniform distribution.

**Proof**

\[
F_U(u) = u, \quad U \sim \mathcal{U}(0,1)
\]

\[
P(X \leq x) = P[F^{-1}(U) \leq x] = P[U \leq F(x)] = \int_0^F 1du = F(x)
\]

**Example: Exponential Distrib.**

\[
x \sim \mathcal{E}(1), \quad F(x) = 1 - e^{-x}, \quad x > 0
\]

Inverse of Exponential Distribution

\[
F^{-1}(u) = -\ln(1 - u)
\]

\[
X = -\ln(1 - U), \quad U \sim \mathcal{U}(0,1)
\]

\[
\Rightarrow X \sim \mathcal{E}(1)
\]

**Gaussian Pseudorandom**

- **randn**: Normally distributed random numbers and arrays.
  - $Z = \text{randn}(m, n)$ or $Z = \text{randn}([m, n])$ returns an $m$-by-$n$ matrix of random entries.
  - $Z = \text{randn}$ % returns a scalar.
    - $Z \sim \mathcal{N}(0,1)$,
    - $Y = \sigma \times Z + \text{mean}$, $Y \sim \mathcal{N}(\text{mean}, \sigma^2)$
Statistics Toolbox

- Generates random numbers with other distributions using `rand/randn`.
- Commands in the form `xxxrnd`
  - `xxx` = name of distribution

Examples: chi2rnd (Chi-square), binornd (binomial), poissrnd (Poisson), etc.

Gaussian

- $Z \sim \mathcal{N}(\text{mean, } \sigma^2)$
- `Z = normrnd(mean, sigma, m, n)`
  - $\text{mean}=1; \sigma=2; m=2; n=1$
  - `Z = normrnd(mean, sigma, m, n)`
  - $Z = [0.1349, -2.3312]$

Random

- Uses the name of the distribution as a parameter (Normal, Poisson, Chi2)
- `rn = random('Normal', mean, var, m, n)`
  - `rn = random('Normal', 1, 4, 2, 3)`
  - Mean 1, var = 4, 2 by 3 matrix
  - $rn = [0.6174, 2.1776, 3.8573; -2.3294, -4.3447, 7.4942]$

Uniform

- $Z \sim \mathcal{U}[\text{min, max}]$
  - $\text{height} = 1/(\text{max} - \text{min})$
- `Z = random('Uniform', min, max, m, n)`
  - `Z = random('Uniform', 1.1, 2, 1, 3)`
  - 1 by 3 array with uniform pdf
  - $Z = [1.9551, 1.3080, 1.6462]$
Interesting Web Sites

- [http://www.agner.org/random/](http://www.agner.org/random/)
  Programs for generating uniform and nonuniform distributions.
  MATLAB random number generator: for all commands
  - Controlling the random number generator: `rng`
  - `>> rng(sd) % sd = nonnegative integer seed`

Conclusion

- Pseudorandom not random.
- Monte Carlo simulation.
- Application in physics, engineering, biology, etc.