Generating Pseudo-Random Numbers and Random Variates

Quiz Study Guide — Appendix A

Manuel D. Rossetti, PhD P.E.

How to Use This Deck

This deck is a **checklist**, not a summary. Every slide is phrased as something you should be able to *state*, *derive*, or *compute* without looking it up.

- **Question formats:** multiple choice, true/false, matching, fill-in-the-blank, multi-slot fill-in-the-blank, and numeric answers.
- Work the **Drills** slides by hand first; the answers are in the appendix at the end.
- The **Common Traps** slides are built from the actual wrong answers — read them last.

> **Relationship to Chapter 2**
>
> This appendix is the **theory**; Chapter 2 is how the KSL implements it. Several questions can be answered from either, but the formulas live here.

What This Appendix Covers

Pseudo-random numbers

- Definition and why we use generators
- Linear congruential generators
- The Hull-Dobell full-period theorem
- Streams and sub-streams

Random variates

- Inverse transform
- Convolution
- Acceptance/rejection
- Mixtures, truncation, shifting

Objectives — Random Numbers

You should be able to:

- Define a pseudo-random number sequence and explain why generators beat dice
- Write the LCG recursion and name all four parameters
- **Crank the recursion by hand** and identify the period
- State all three Hull-Dobell conditions and apply them to the two special cases
- Explain why $m = 2^{31} - 1$ is a common choice
- Define stream and sub-stream, and quote the L'Ecuyer generator's numbers

Objectives — Random Variates

You should be able to:

- Name the four generation strategies — and what is **not** one of them
- Reproduce seven inverse-CDF formulas from memory
- Match each convolution target to its components
- State the requirements on a majorizing function and compute P_a
- Write the truncation CDF and its generation algorithm
- Generate a variate by hand from any of the above

What a Pseudo-Random Number Is

A **deterministic** sequence of numbers in $(0,1)$, produced by an algorithm, that nevertheless has the same relevant statistical properties as a true sequence of $U(0,1)$ random numbers.

Why generators instead of dice, coins, or electronic noise? They are fast, cheap, and **reproducible** — the same seed gives the same sequence. That reproducibility is what makes debugging, controlled comparison of alternatives, and variance reduction possible.

Re-running a simulation gives identical results because the generator **restarts from the same seed**. Seeding from the **system clock** destroys this, so it is recommended against in a simulation study.

The Linear Congruential Generator

$$R_{i+1} = (aR_i + c) \bmod m, \qquad U_i = R_i/m$$

Table 1: With $a > 0$, $c \geq 0$, $m > a$, $m > c$, $m > R_0$, and $0 \leq R_i \leq m - 1$

Parameter	Name and role
m	The **modulus** — bounds R_i and caps the achievable period
a	The constant **multiplier**
c	The **increment**; when $c = 0$ the LCG is **multiplicative**
R_0	The **seed** — determines the stream

The **period** is the length of the cycle of distinct values before the sequence repeats. If it equals m, the LCG has **full period**.

The Hull-Dobell Theorem

An LCG achieves full period **if and only if** all three hold:

1. $\gcd(c, m) = 1$
2. $(a - 1)$ is a multiple of **every prime** q that divides m
3. If 4 divides m, then 4 must also divide $(a - 1)$

Two special cases:

- **PMMLCG —** m prime, $c = 0$. Full period m is **not** achievable; the best is $m - 1$, with a well-chosen a. Taking $R_0 \in \{1, \dots, m - 1\}$ guarantees $U_i \in (0, 1)$.
- m **a power of 2 with** $c = 0$ **—** best achievable period is $m/4$, requiring R_0 **odd** and a of the form $8k + 3$ or $8k + 5$.

$m = 2^{31} - 1 = 2{,}147{,}483{,}647$ is common on 32-bit machines because it is the largest representable 2's-complement integer **and** it is **prime**.

Streams and the Modern Generator

- A **stream** is the **subsequence** of pseudo-random numbers generated starting from a given seed. A **sub-stream** is a non-overlapping segment *within* a stream.
- The small $m = 8$ example motivates streams: different seeds start you at different points of the **same single cycle**, so subsequences can be named by **stream number** rather than by huge integer seeds.

The L'Ecuyer Generator

Property	Value
Construction	Combination of **two MRGs**
Period	$\approx 3.1 \times 10^{57}$
Streams	$\approx 1.8 \times 10^{19}$, each of length $\approx 1.7 \times 10^{38}$
Initial state	A **6-element seed vector** — not a single integer

Streams are effectively independent as long as they do not overlap, so dedicating a fixed stream per source of randomness **synchronizes** pseudo-random number use across alternatives.

The Four Strategies

1. **Inverse transform** (inverse CDF)
2. **Convolution**
3. **Acceptance/rejection**
4. **Mixture, truncated, and shifted distributions**

⚠ Trap

Bootstrap resampling is not one of the four. That is the classic distractor.

Inverse Transform

$$X = F^{-1}(U), \qquad U \sim U(0,1)$$

It is **preferred when available** because it uses exactly **one** $U(0,1)$ per generated X — which is what makes common random numbers and antithetic variates work cleanly.

For **discrete** distributions it is a **table lookup in the CDF**: pick x_i with $F(x_{i-1}) < u \leq F(x_i)$. Because F is non-decreasing, you need only check the **upper** limit of each interval.

The Inverse-CDF Formulas

Table 2: Memorize all seven — the "+1" is the whole geometric distinction

Distribution	Inverse transform
Exponential, rate λ (mean $1/\lambda$)	$X = -\frac{1}{\lambda}\ln(1-U)$
Continuous Uniform(a, b)	$X = a + U(b-a)$
Discrete Uniform on $\{a, \ldots, b\}$	$X = a + \lfloor (b-a+1)U \rfloor$
Weibull, shape α, scale β	$X = \beta\left[-\ln(1-U)\right]^{1/\alpha}$
Geometric on $\{0, 1, 2, \ldots\}$	$X = \lfloor \ln(1-U)/\ln(1-p) \rfloor$
Shifted geometric on $\{1, 2, 3, \ldots\}$	$X = 1 + \lfloor \ln(1-U)/\ln(1-p) \rfloor$
Bernoulli(p)	$X = 1$ if $U \leq p$, else $X = 0$

Poisson via the Poisson Process

If interarrival times are exponential with mean $1/\lambda$, then the count of arrivals in $[0, t]$ is Poisson with mean λt.

Algorithm: generate exponential interarrival times by inverse transform, accumulate them until the cumulative sum **first exceeds** t, and return the **count of arrivals that fell within** $[0, t]$.

Conditions of the Poisson process: the probability of more than one event in a small subinterval is zero; the probability of one event is proportional to the subinterval length; subintervals are independent. As $n \to \infty$ with $\lambda = np$ held constant, these lead to the **Poisson** distribution.

Convolution

Some random variables are the **sum of i.i.d. simpler random variables**.

Target	Sum of
Binomial(n, p)	n i.i.d. Bernoulli(p)
Negative binomial (r, p) on $\{0, 1, 2, ...\}$	r i.i.d. geometric(p)
Erlang(r, λ)	r i.i.d. exponentials, each with rate λ
Chi-squared	squared independent standard normals
Normal	independent normals

Limitation: convolution is **not** a general method. It applies only when the target genuinely is such a sum.

Acceptance/Rejection

The majorizing function must satisfy $g(x) \geq f(x)$ for **all** x, and $c = \int g(x)\,dx < \infty$, so that $w(x) = g(x)/c$ is a valid PDF that is easy to sample.

The loop: generate $W \sim w(x)$ and $U \sim U(0,1)$; **accept** W if $U \cdot g(W) \leq f(W)$; otherwise reject and repeat.

$$P_a = \frac{1}{c}$$

The closer g is in shape to f, the smaller c, the **higher** the acceptance probability, and the more efficient the algorithm.

Textbook example. $f(x) = \frac{3}{4}(1 - x^2)$ on $[-1, 1]$ with flat $g(x) = 3/4$.

Mixtures, Truncation, Shifting

Mixture. $F_X(x) = \sum_{i=1}^{k} \omega_i F_{X_i}(x)$, with the ω_i behaving as a discrete distribution. **Sample an index $I \sim \omega_i$ first, then sample X from F_{X_I}** — do *not* sample every component and average.

The **hyper-exponential** is a mixture of exponentials with different rates; its defining feature is $c_v > 1$, useful for high variability — service times with mutually exclusive **phases**, such as credit-card versus cash checkout.

Truncation to $[a, b]$.

$$F^*(x) = \frac{F(x) - F(a)}{F(b) - F(a)}, \qquad X = F^{-1}(F(a) + U[F(b) - F(a)])$$

Shift. $Y = X + \delta$ has density $g(y) = f(y - \delta)$: generate X by any method and add δ. A shifted Weibull suits a known minimum, such as a minimum setup time.

Drills — LCG

1. $m = 8$, $a = 5$, $c = 1$, $R_0 = 5$. Generate the full cycle of R values, and report U_3 and U_5.
2. $a = 13$, $m = 64$, $c = 0$, $R_0 = 1$. Compute U_2.
3. $a = 13$, $m = 64$, $c = 0$. Can full period be achieved? What is the best achievable period, and what does it require of R_0 and a?

Drills — Variate Generation

4. Exponential with mean 10, $u = 0.943$. Report X to 4 decimal places.
5. Exponential interarrival with rate $\lambda = 4$ per hour, $u = 0.971$. Report T_1 in hours to 4 decimal places.
6. Uniform$(12, 22)$ with $u = 0.3734$.
7. Shifted Weibull, $\alpha = 3$, $\beta = 5$, $\delta = 5.5$, $u = 0.73$, to 2 decimals.
8. Erlang with $r = 2$ phases each of mean $\beta = 3$, using $u_1 = 0.9559$ and $u_2 = 0.5814$, to 3 decimals.
9. Exponential with mean 10 truncated to $[3, 6]$, $u = 0.23$, to 3 decimals.
10. Hyper-exponential, $\omega_1 = 0.7$, $\omega_2 = 0.3$, means 1.5 and 1.1, using $u_1 = 0.54$ and $u_2 = 0.12$, to 4 decimals.
11. For $f(x) = \frac{3}{4}(1 - x^2)$ on $[-1, 1]$ with $g(x) = 3/4$, compute c and P_a.

Common Traps

- A PMMLCG **cannot** achieve full period m — the best is $m-1$. With m a power of 2 and $c=0$, the best is $m/4$.
- The geometric on $\{0,1,\dots\}$ has **no “+1”**; the shifted one on $\{1,2,\dots\}$ does.
- Discrete uniform uses $a+\lfloor(b-a+1)U\rfloor$ — **not** the continuous formula $a+U(b-a)$.
- The acceptance probability is $1/c$, **not** c. A tighter g means a **smaller** c and a **higher** acceptance rate.
- The hyper-exponential has $c_v>1$, not $c_v<1$.
- Convolution is **not** general.
- **Bootstrap resampling is not one of the four strategies.**
- A mixture is sampled by **choosing a component first**, not by averaging components.

Appendix — Drill Answers (1 of 2)

1. $R: 5 \to 2 \to 3 \to 0 \to 1 \to 6 \to 7 \to 4 \to 5$ — all 8 values, so **full period**.
   $U_3 = 0/8 = \mathbf{0.0}$ and $U_5 = 6/8 = \mathbf{0.75}$.
2. $R_1 = 13$; $R_2 = 169 \bmod 64 = 41$; $U_2 = 41/64 = \mathbf{0.640625}$.
3. **No** — $c = 0$ fails condition 1. Best period is $m/4 = \mathbf{16}$, requiring R_0 **odd** and $a = 8k + 3$ or $8k + 5$. (Here $13 = 8(1) + 5$.)
4. $X = -10\ln(0.057) = \mathbf{28.6470}$
5. $T_1 = -\frac{1}{4}\ln(0.029) = \mathbf{0.8851}$ hours
6. $X = 12 + 0.3734(10) = \mathbf{15.734}$
7. $X = 5.5 + 5[-\ln(0.27)]^{1/3} = 5.5 + 5(1.09383) = \mathbf{10.97}$

8. $-3\ln(0.0441) = 9.3647$ and $-3\ln(0.4186) = 2.6122$; sum $= \mathbf{11.976}$
9. $F(3) = 0.259182$, $F(6) = 0.451188$; $W = 0.259182 + 0.23(0.192006) = 0.303343$; $X = -10\ln(0.696657) = \mathbf{3.615}$ — inside $[3, 6]$, as it must be.
10. $u_1 = 0.54 \leq 0.7$, so component 1 (mean 1.5); $X = -1.5\ln(0.88) = \mathbf{0.1918}$. Note it takes **two** pseudo-random numbers: one to choose, one to generate.
11. $c = \frac{3}{4} \cdot 2 = \mathbf{1.5}$ and $P_a = 1/c = \mathbf{2/3}$ — about two of every three candidates are accepted.