Implementation / Programming: Random Number Generation

OSMAN BALCI
Professor

Department of Computer Science
Virginia Polytechnic Institute and State University (Virginia Tech)
Blacksburg, VA 24061, USA

https://manta.cs.vt.edu/balci

Basic Concepts in Number Theory
Background for Random Number Generation

1. **For any pair of integers** n **and** m, $m \neq 0$, **there exists a unique pair of integers such that** $n = mq + r$ **with** $0 \leq r < m$, **where** m **is the modulus,** q **the quotient, and** r **the residue (remainder). This is commonly written as**

$$r \equiv n \pmod{m}$$

and is read as "r is congruent to n modulo m."

Congruent is defined as having the difference divisible by a given modulus. For example:

$12 \equiv 2 \pmod{5} \rightarrow$ **since** $12 - 2$ **is divisible by** 5.

$7 \equiv -1 \pmod{8} \rightarrow$ **since** $7 + 1$ **is divisible by** 8.

Basic Concepts in Number Theory

2. **Commonly, the residue is the remainder after division by** m**. There are** m **distinct residues (modulo** m**):** 0, 1, 2, ..., m – 1.

3. **Two integers** a **and** b **are Congruent Modulo m if their difference is an integral multiple of** m**. For example, let** a = 15**,** b = 8**, and** m = 7**.**

$$1 \equiv 15 \pmod 7 \text{ and } 1 \equiv 8 \pmod 7$$

4. **An integer** x **is a Prime Number if it is neither** 0 **nor** ±1 **and if its only divisors are** ±1 **and** ±x**. Example positive prime numbers:**

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ...

5. **An integer** g **is the Greatest Common Divisor (GCD) of two integers** a **and** b **if** g **is a common divisor of** a **and** b **and is a multiple of every other common divisor of** a **and** b**. This is usually written as** (a, b) = g **or** GCD (a, b) = g**.**

(12, 16) = 4

Basic Concepts in Number Theory

6. **The integers** a **and** b **are said to be Relatively Prime if** $(a, b) = 1$**.**

7 **and** 12 **are relatively prime because** $(7, 12) = 1$

7. **A Residue Class is a class of all integers which are mutually congruent for a given modulus. There are** m **distinct residue classes (modulo** m**), corresponding to the terms of a complete residue system modulo** m**. Collectively they comprise the class of all integers:**

$$\ldots 36 \equiv 29 \equiv 22 \equiv 15 \equiv 08 \equiv 1 \equiv -6 \pmod 7$$
$$\ldots 37 \equiv 30 \equiv 23 \equiv 16 \equiv 09 \equiv 2 \equiv -5 \pmod 7$$
$$\ldots 38 \equiv 31 \equiv 24 \equiv 17 \equiv 10 \equiv 3 \equiv -4 \pmod 7$$
$$\ldots 39 \equiv 32 \equiv 25 \equiv 18 \equiv 11 \equiv 4 \equiv -3 \pmod 7$$
$$\ldots 40 \equiv 33 \equiv 26 \equiv 19 \equiv 12 \equiv 5 \equiv -2 \pmod 7$$
$$\ldots 41 \equiv 34 \equiv 27 \equiv 20 \equiv 13 \equiv 6 \equiv -1 \pmod 7$$
$$\ldots 42 \equiv 35 \equiv 28 \equiv 21 \equiv 14 \equiv 7 \equiv -0 \pmod 7$$

Basic Concepts in Number Theory

8. **A Complete Residue System is a set of m numbers, for a given modulus m, congruent in some order to the residues $0,1,2,\ldots,m-1$.**

A CRS for 7:	4	8	14	9	10	12	13	(mod 7)
Residue:	4	1	0	2	3	5	6	

9. **A subset of a complete residue system, containing all terms which are relatively prime to m, is termed a Reduced Residue System.**

4, 8, 9, 10, 12, 13

10. **Euler's Phi-function, $\phi(.)$, denotes the number of positive integers less than m and relatively prime to m, so a reduced residue system contains $\phi(m)$, terms if m is not a prime, $\phi(m) = m - 1$ if m is a prime.**

$\phi(7) = 7 - 1 = 6$

Basic Concepts in Number Theory

11. Power Residues are the residues of the successive powers of a number (i.e., $x^n \pmod m$ **for** $n = 1, 2, 3, \ldots$ **). For** $x=4$ **and** $n = 1, 2, 3, 4, 5, 6, \ldots$ **the values** $4, 2, 1, 4, 2, 1, \ldots$ **are the power residues (mod 7).**

12. Order of x modulo m is defined to be the least positive exponent h **with** $x^h \equiv 1 \pmod m$ **when** x **and** m **are relatively prime.**

Order of 4 modulo 7 = 3 $\rightarrow$ $(4^3 \equiv 1 \pmod 7)$

13. The Primitive Root of m is a number x **whose order** h **is equal to** $\phi(m)$**. For the reduced residue system in the example above (4, 8, 9, 10, 12, 13); a primitive root of 7 is 10. Note that 12 is also a primitive root.**

$4^3 \equiv 8^1 \equiv 9^3 \equiv 10^6 \equiv 12^6 \equiv 13^2 \equiv 1 \pmod 7$

Basic Concepts in Number Theory

14. **If** $(x, m) = 1$**, then** $x^n \pmod m$**, where** $n = 0, 1, 2, 3, \ldots$**, repeats by returning to the starting point since** $x^r \equiv x^s \pmod m$ **implies** $x^{r-s} \equiv 1 \pmod m$**; the division is possible because** $(x, m) = 1$**.**
15. **If** $a \equiv b \pmod m$ **and** $c \equiv d \pmod m$**, then** $a \pm c \equiv b \pm d \pmod m$**.**
16. **If** $a \equiv b \pmod m$, **then** $ac \equiv bc \pmod{mc}$.
17. **If** $a \equiv b \pmod m$ **and** $b \equiv c \pmod m$, **then** $a \equiv c \pmod m$.
18. **If** $a \equiv b \pmod m$ **and** d **is any divisor of** m**, then** $a \equiv b \pmod d$.
19. **If** $a \equiv b \pmod m$ **and** $c \equiv d \pmod m$, **then** $ac \equiv bd \pmod m$.
20. **If** $a \equiv b \pmod m$, **with** d **as any common divisor of** a **and** b**, and** $(m, d) = g$ **then**

$$(a/d) \equiv (b/d) \pmod{(m/g)}.$$

Pseudo-Random Number Generation
Linear Congruential Generators (LCGs)

$$Z_i \equiv (a\, Z_{i-1} + c\,) \pmod{m}$$

where

- **a** is the multiplier
- Z_0 is the **seed** or starting value
- **c** is the constant
- **m** is the modulus

- **a, Z_0, c, and m are positive integers**
- $0 \le Z_i \le m - 1$ **for** $i = 1, 2, 3, \ldots$
- **To obtain random numbers on** $[0, 1]$, **we let** $U_i = Z_i / m$
- $0 < m$, $a < m$, $c < m$, **and** $Z_0 < m$

Why Z_0 cannot be equal to m?

Linear Congruential Generators

- **The length of a cycle is called the period (p) of a generator.**
- **Since** $0 \le Z_i \le m - 1$ **then** $p \le m$**.**
- **If** $p = m$**, the LCG is said to have Full Period.**
- **Question: How should we choose m, a, and c values so that the corresponding LCG will have full period?**
- **Theorem 1:**

$Z_i \equiv (a Z_{i-1} + c) \pmod m$ **has full period (length m) if and only if**

1. **(c, m) = 1**
2. **If q is a prime number that divides m, then q divides a – 1**
3. **If 4 divides m, then 4 divides a – 1**

Linear Congruential Generators

- **Theorem 2:**

  $Z_i \equiv a\, Z_{i-1} \pmod{m}$ has full period if and only if

  1. **m is a prime number**
  2. **a is a primitive root modulo m**

- **If a and m are chosen to satisfy Theorem 2,**
  - $Z_i = 1, 2, 3, \ldots, m - 1$ **(cycle)**
  - **Period =** $m - 1$
  - $0 \leq Z_0 \leq m - 1$

Mixed Congruential Generators

$$Z_i \equiv (a\, Z_{i-1} + c\,) \pmod{m}$$

- Since this LCG's full period is m, we want m as large as possible.
- If a computer has a word size of b bits, the largest number that can be represented is $2^{b-1} - 1$.
- If b=4, the largest number is $111 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7$ since one bit (leftmost) is used for sign.
- However, m is chosen to be 2^{b-1} to avoid explicit division by m on *some* computers by taking advantage of "integer overflow."
- **Example:** Let b=5. Thus we choose $2^4=16=m$ when in fact 15 is the largest number that can be represented by a word of 5 bits. Now let's see how we avoid the division by the overflow.
- Any attempt to store an integer larger than 15 will result in loss of the leftmost bits in overflow.

Mixed Congruential Generators

Consider $Z_i \equiv (5\ Z_{i-1} + 3)\ (\text{mod } 16)$

Let $Z_{i-1} = 8$. Then $5 \times 8 + 3 = 43$. $43 \div 16 = 2$ with a remainder of 11.

2^5	2^4	2^3	2^2	2^1	2^0	
32	16	8	4	2	1	
1	0	1	0	1	1 $)_2$	$= 43\ )_{10}$
dropped		$= 11\ )_{10}$				

- **Tested values for a 36-bit computer:**
  - $Z_i \equiv (5^{15}\ Z_{i-1} + 1)\ (\text{mod } 2^{35})$
- **Tested values for a 32-bit computer:**
  - $Z_i \equiv (\pi \times 10^8\ Z_{i-1} + 453806245)\ (\text{mod } 2^{31})$

Multiplicative Congruential Generators

$$Z_i \equiv a\, Z_{i-1} \pmod{m}$$

- **From Theorem 2:**
  - **m must be a prime number, and**
  - **a must be a primitive root modulo m to achieve the full period of** $m - 1$**.**
- **The modulus m must be as large as possible; however, we cannot choose** $m = 2^{b-1}$ **since it is divisible by 2 for all values of the word size b.**
- **Hence the largest prime number is** $2^{b-1} - 1$ **which is used for m.**
- **Tested values for 32-bit computers:**
  - $Z_i \equiv 7^5\, Z_{i-1} \pmod{2^{31} - 1}$ ☜ **Use this one**

Additive Congruential Generators

$$Z_{i+1} \equiv \left(\sum_{j=0}^{k} \delta_j Z_{i-j} \right) \pmod{m}$$

Quadratic Congruential Generators

$$Z_i \equiv (a' Z^2_{i-1} + a Z_{i-1} + c) \pmod{m}$$

Combinational Techniques

Table Shuffling Approach

1. Create a one-dimensional array (Table) of K (= 128) elements.
2. Generate K random numbers and fill in the Table using the following LCG 1:

$$U_i \equiv (2^{17} + 3)\, U_{i-1} \pmod{2^{35}}$$

3. Generate an integer random number N over [1, K] using the following LCG 2:

$$V_i \equiv ((2^7 + 1)\, V_{i-1} + 1) \pmod{2^{35}}$$

4. Deliver the N^{th} element of the Table as the random number.
5. Replace the N^{th} element of the Table by using the LCG 1 in Step 2.

Combinational Techniques

Bit String Perturbation Approach

Let $Z_i^{(1)}$ and $Z_i^{(2)}$ be the values of Z_i produced by the first and second LCG's given on the previous slide, respectively.

1. Using the bits of $Z_i^{(1)}$, "rotate circularly" the bits of $Z_i^{(2)}$ to obtain $Z_i^{(3)}$ between 0 and m – 1.

2. Using a bitwise addition modulo 2 (Exclusive OR = XOR) on the bits of $Z_i^{(1)}$ and $Z_i^{(3)}$, obtain $Z_i^{(4)}$.

3. Deliver $U_i = Z_i^{(4)} / m$

TAUSWORTHE GENERATORS

$$b_i \equiv (c_1 b_{i-1} + c_2 b_{i-2} + \ldots + c_q b_{i-q}) \pmod 2$$

- where C_j is a constant 0 or 1 and b_k is a bit.
- The maximum period = $2^q - 1$.
- Usually, only two of the C_j's are nonzero.

$$b_i \equiv (b_{i-r} + b_{i-q}) \pmod 2$$

for integers r and q satisfying $0 < r < q$.

- The following Exclusive OR operation is equivalent to the congruent relation given above:

$$b_i = \begin{cases} 0 & \text{if } b_{i-r} = b_{i-q} \\ 1 & \text{if } b_{i-r} \neq b_{i-q} \end{cases}$$

TAUSWORTHE GENERATORS

Example: Let r = 3 and q = 4.

$$b_i \equiv (b_{i-3} + b_{i-4}) \pmod 2$$

i = 1	1	0	1	1
i = 2	0	1	1	0
i = 3	1	0	0	0
i = 4	0	1	0	0
i = 5	1	1	0	1
i = 6	1	1	1	0
i = 7	1	1	0	0
i = 8	1	0	0	1
i = 9	0	0	1	1
i = 10	0	0	1	0
i = 11	0	1	0	1
i = 12	1	0	1	0
i = 13	0	0	0	1
i = 14	0	1	1	1
i = 15	1	1	1	1
i = 16	1	0	1	1

Initial Values

Assume 4 bits are used to store an integer number Initial starting 4 values must then be given.

$$b_i = \begin{cases} 0 \text{ if } b_{i-r} = b_{i-q} \\ 1 \text{ if } b_{i-r} \neq b_{i-q} \end{cases}$$

← End of cycle

Note that the period is $2^q - 1 = 15$

Desired Properties of a Random Number Generator

1. Random numbers must be independent and identically distributed over [0, 1].
2. Random numbers must be uniformly distributed.
3. Random numbers must be reproducible.
4. Must have high execution speed and minimum amount of storage.
5. Random numbers must have a considerably large period (cycle).
6. The generator should be generally available.

Testing for Randomness

1. **Frequency Test:** Uses either the Chi-square or Kolmogorov-Smirnoff test to compare the distribution of the set of numbers generated against a uniform distribution.
2. **Serial Test:** Tallies the frequency of occurrence of all possible combinations of 2, 3, 4, etc., digits and then runs a Chi-square test against expected values.
3. **Gap Test:** Counts the number of digits that appear between repetitions of a particular digit and then uses a Chi-square test against expected values.
4. **Runs Test:** Tests the number of runs above and below some constant (usually the mean) or runs up and down. The test involves counting the actual number of occurrences of runs of different lengths and comparing these counts to expected values by Chi-square.
5. **Spectral Test:** Measures the independence of adjacent sets of n numbers based on Fourier analysis.

Testing for Randomness

6. **Poker Test:** Analogous to testing poker hands. This test counts combinations of five or more digits for all digits different, one pair, two pairs, three of a kind, full house, etc., and tests against expected occurrences.
7. **Autocorrelation Test:** Tests the correlation between X_n and X_{n+k} where k is the lag in the generation order (k = 1, 2, 3, ...).
8. **D^2 or Distance Test:** Successive pairs of random numbers are regarded as coordinates for points in the unit square, and the square of the distance between the two points is tested against theoretical probabilities given by a set of equations.
9. **Order Statistic Test:** Tests the maximum or minimum value of n consecutive numbers or the range of n consecutive values.
10. **Yule's Test:** Consists of taking the sum of five decimal digits from normalized random numbers and comparing it with the theoretical expected values.