CPE 345: Modeling and Simulation

Lecture 9

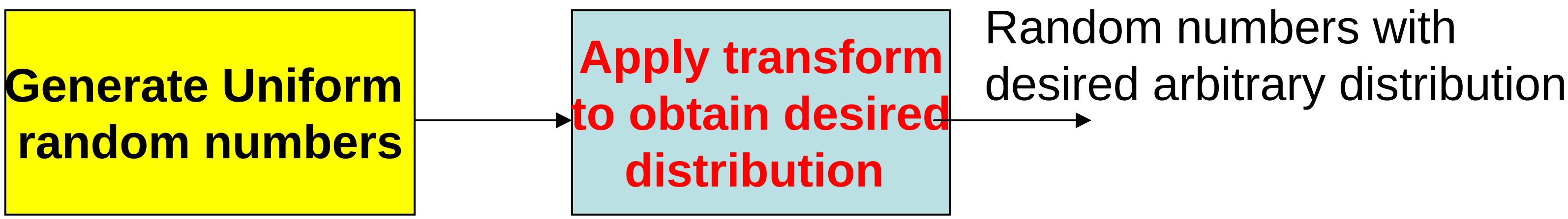
Today’s topics

- Announcements
  – Project due last day of classes
  – End of class – Extra Credit Quiz

- Today’s lecture: Random-Variate Generation

Random-Variate Generation

- Generation of random variables with other distribution than the uniform one
  - Generate continuous and discrete r.v. s with an arbitrary distribution



What transform techniques?

- Inverse transform
- Convolution Method
- Acceptance-Rejection

Why worry about generators?

- OMNET++ and other simulation environments have pre-implemented function for different distribution generation
  - you can understand how they work, and what to expect from them
  - not all distributions may have been already implemented
  - you might develop at some point your own simulation environment (maybe as part of a team)

Inverse transform technique

- Very straightforward technique
- Not always the most computationally efficient
- Explained for exponential distribution, then general principles can be applied to other distributions as well

- Generate R_i with uniform distribution in (0,1)
- Let X be the exponential r.v. that we want to generate
- CDF of exponential is

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)dx = 1 - e^{-\lambda x}, \quad x \geq 0 \qquad f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & ow \end{cases}$$

Inverse transform technique – cont.

- Set *F(X)=R,* then solve for *X = funct(R)*

$$1 - e^{-\lambda X} = R$$

$$e^{-\lambda x} = 1 - R$$

$$-\lambda X = \ln(1 - R)$$

$$X = -\frac{1}{\lambda}\ln(1 - R)$$

- Note that 1-R and R have the same distribution (uniform in (0,1) ➔

$$X = -\frac{1}{\lambda}\ln(R)$$

- Determine the exponential random numbers (X_i), using the uniformly generated random numbers R_i

Why this procedure results in a correct distribution?

- The resulted CDF for X:

$$P(X_1 \le x_0) = P\left(R_1 \le F(x_0)\right) = F(x_0)$$

$$R = F\left(X\right)$$

For a uniform distribution: U(0,1)

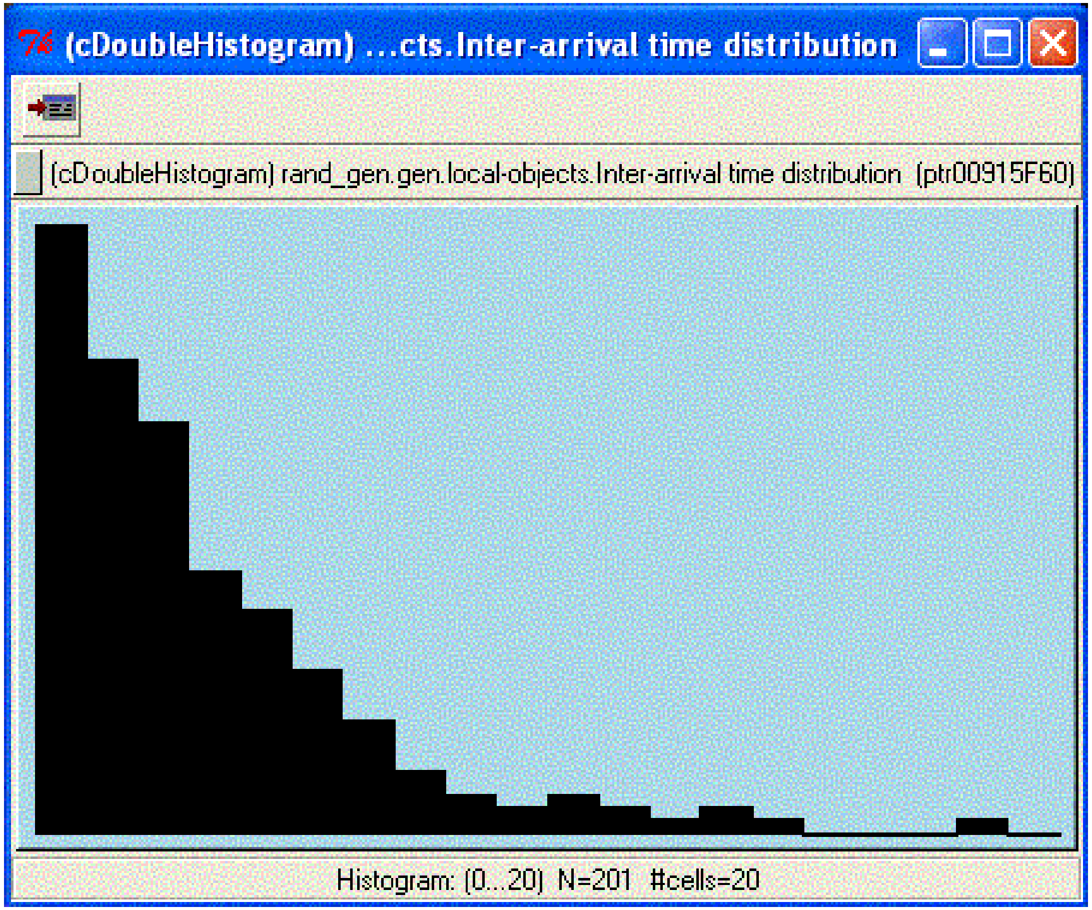
$$P(R \le a) = a, \quad 0 \le \mathrm{a} \le 1$$

Implement your own exponential generator using OMNET++ and the inverse transform

- Implement your own function: double my_exp(double mean)
- Declare it: Define_function (my_exp, 1)
- Use it in config file
- Collect statistics
  - Output vector
    - current values
  - Histogram
    - pdf approximation
- Study the code

Histogram range: [0, 20] →

Automatic range estimation
- the object collects the first few samples and estimates the range



Inverse transform technique for other distributions

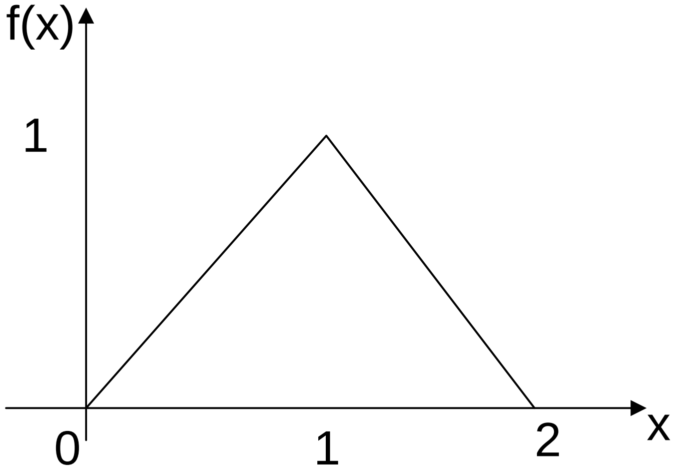
- When is it appropriate
  - When a closed form inverse for CDF exists
    - Simple example: Triangular distribution

$$f(x) = \begin{cases} x & x \le 0 \\ 2 - x & 1 < x < 2 \\ 0 & ow \end{cases}$$

$$F(x) = \begin{cases} 0 & x \le 0 \\ \frac{x^2}{2} & 0 < x \le 1 \\ 1 - \frac{(2-x)^2}{2} & 1 < x \le 2 \\ 1 & x > 2 \end{cases}$$

The inverse transform formula can then be determined to be:

$$X = \begin{cases} \sqrt{2R}, & 0 \le R \le \frac{1}{2} \\ 2 - \sqrt{2(1-R)}, & \frac{1}{2} < R \le 1 \end{cases}$$



When is it appropriate? - cont.

– When a good approximation for the CDF inverse can be found

- Example: simple approx. for Gaussian distribution

$$X = F^{-1}(R) \approx \frac{R^{0.35} - (1-R)^{0.135}}{0.1975}$$

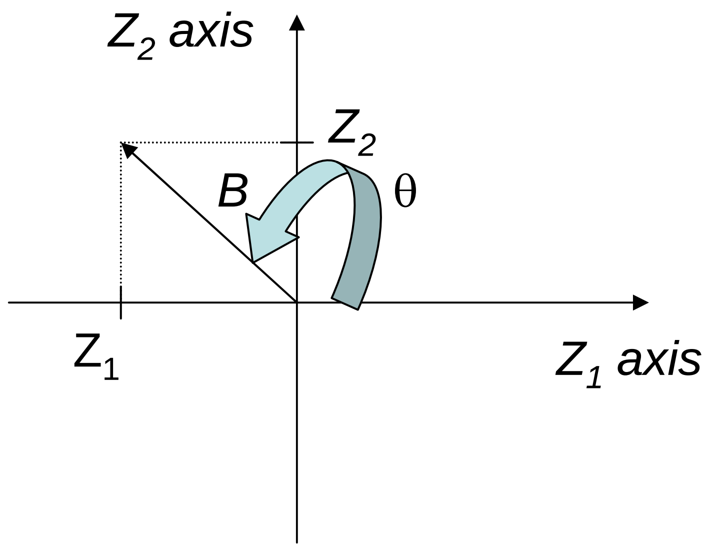
– Approximation with at least one decimal-place accuracy for

$$0.0013499 \leq R \leq 0.9986501$$

– More accurate approx. have also been found

Direct Transform – Normal Distribution

- Inverse CDF for Gaussian cannot be expressed in closed form
- Idea: use a direct transform to express the Gaussian random variable as a function of an exponential r.v., which can then be generated using the inverse transform method
- Consider two standard normal random variables Z_1 and Z_2 (zero mean, unit variance), expressed in polar coordinates



$$Z_1 = B\cos\theta$$

$$Z_2 = B\sin\theta$$

It is known that B^2 has a chi-square distribution with 2 degrees of freedom, equivalent to an exponential distribution with mean 2.

→ $B = (-2 \ln R)^{1/2}$ ← Determined using the inverse transform for B^2

Direct Transform – Normal Distribution

- *B* and the angle θ are mutually independent and θ can be considered as being uniformly distributed between [0, 2 π] radians.
- Two independent standard normal variates can then be generated using two independent uniform random numbers R_1 and R_2:

$$Z_1 = (-2\ln(R_1))^{1/2}\cos(2\pi R_2)$$

$$Z_1 = (-2\ln(R_1))^{1/2}\sin(2\pi R_2)$$

- To obtain a specified mean and variance, we then apply the transform

$$X_i = \mu + \sigma Z_i$$

- OMNET++ can be used again to generate Gaussian r.v.
  - double my_Gauss(double mean, double sigma)

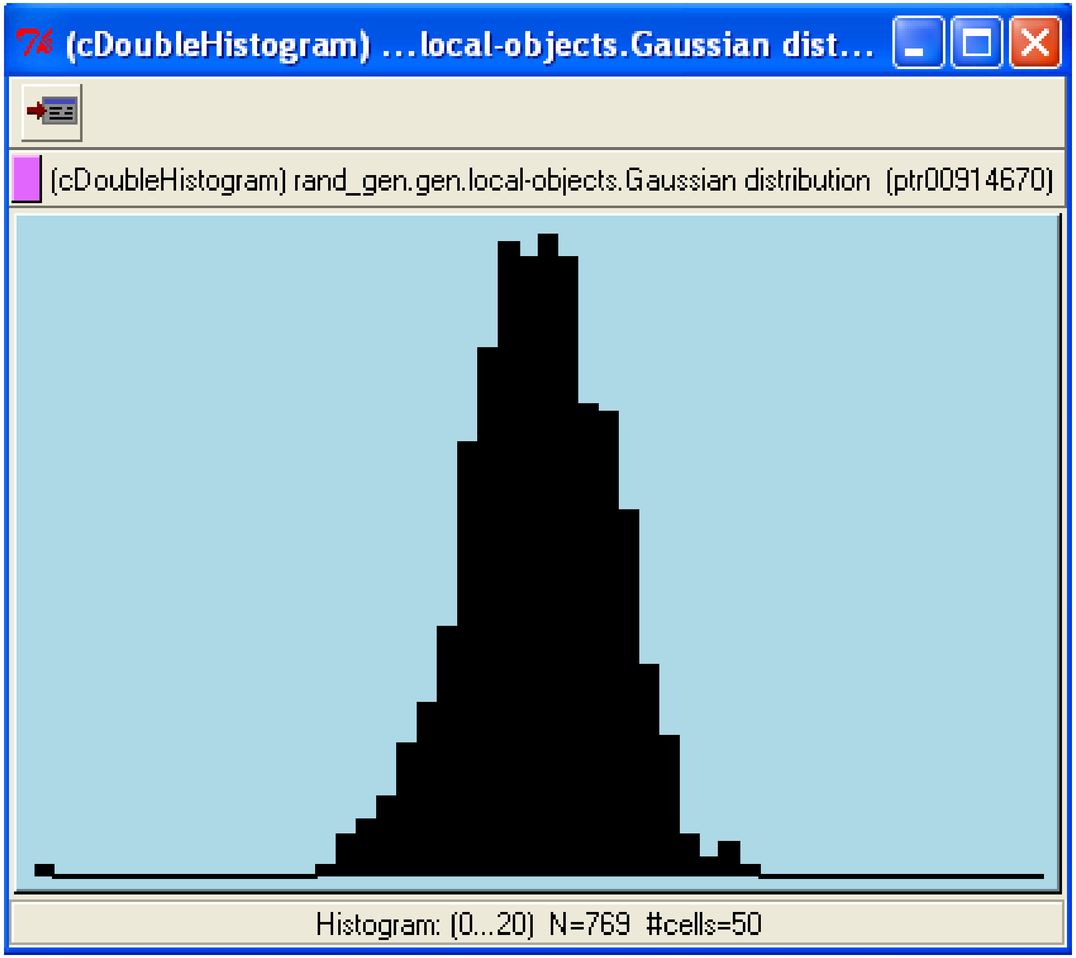
Example of Histogram for Gaussian in OMNET++

Generated:

Gaussian r.v. with

$$\mu = 10, \quad \sigma = \sqrt{2}$$

Histogram range: [0 20]
Number of bins: 50
Number of samples: 769



Convolution method

- Uses the property that the pdf for a sum of independent r.v.s is the convolution of the pdf-s for the initial r.v.

- Example: Erlang r.v. with parameters (k, λ) is the sum of k independent exponential r.v. with mean 1/ $k\lambda$.

$$X = \sum_{i=1}^{k} X_i = \sum_{i=1}^{k} -\frac{1}{k\lambda} \ln R_i = -\frac{1}{k\lambda} \ln\left(\prod_{i=1}^{k} R_i\right)$$

Inverse transform exponential

Property of logarithms

Acceptance-Rejection Technique

- Use when other methods fail
  - Generate r.v. and reject those that do not meet conditions
- Efficiency: depends on how many generated numbers are rejected
- Example:
  - Generate a uniform distribution between ¼ and 1:

Step 1: Generate a random number R

Step 2a: If $R \geq 1/4$, accept $X=R$, go to step 3

Step 2b: If $R < 1/4$, reject R, go to step 1

Step 3: If another uniform variate is needed between ¼ and 1, repeat from step 1; ow. STOP.

Example: Poisson distribution

- Poisson r.v. with mean α

$$p(n) = P(n = N) = \frac{e^{-\alpha}\alpha^n}{n!}, \quad n = 0,1,2,...$$

- N = number of arrivals in the unit time
- Inter-arrival times are exponentially distributed: A_1, A_2, …
- Relationship between Poisson (discrete) and exponential (continuous)
  - N=n if and only if there were exactly n arrivals during one unit of time and the n+1st arrival occurred after time 1, i.e.,

$$A_1 + A_2 + ... + A_n \leq 1 < A_1 + A_2 + ... + A_n + A_{n+1}$$

Acceptance-Rejection Technique

- Generate arrival times such that

$$A_1 + A_2 + \ldots + A_n \le 1 < A_1 + A_2 + \ldots + A_n + A_{n+1}$$

  - Which is equivalent to

$$\sum_{i=1}^{n} -\frac{1}{\alpha} \ln R_i \le 1 < \sum_{i=1}^{n+1} -\frac{1}{\alpha} \ln R_i$$

$$\sum_{i=1}^{n} \ln R_i \ge -\alpha > \sum_{i=1}^{n+1} \ln R_i$$

$$\ln \prod_{i=1}^{n} R_i \ge -\alpha > \ln \prod_{i=1}^{n+1} R_i$$

$$\prod_{i=1}^{n} R_i \ge e^{-\alpha} > \prod_{i=1}^{n+1} R_i$$

Acceptance-Rejection Routine for Poisson Distribution

- Step 1: Set $n=0, P=1$.
- Step 2: Generate a uniform random number R_{n+1}, and replace P with $P \times R_{n+1}$
- If $P < e^{-\alpha}$, then accept $N=n$. Otherwise, reject the current n, increase n by 1, and repeat from step 2.

Study example 8.10 – illustrates the generation of three Poisson variates

Alternate method for Poisson generation

- If arrival rate is large, the rejection technique may become quite expensive
- Approximate technique based on normal distribution:
  - When the arrival rate is large, the variable

$$Z = \frac{N - \alpha}{\sqrt{\alpha}}$$

Is approximately Gaussian distributed with mean 0, and unit variance

This suggest the following approach:

- first generate a Gaussian variate Z, then determine N:

$$N = \left| \alpha + \sqrt{\alpha} Z - 0.5 \right|$$

round to the nearest integer

Homework – due next week in class

- Web traffic models use a Pareto distribution:

$$pdf: \quad f(x) = \frac{ab^a}{x^{a+1}}, \quad \mathrm{x} \geq \mathrm{b}$$

$$CDF: \quad F(x) = 1 - \left(\frac{b}{x}\right)^a, \quad \mathrm{x} \geq \mathrm{b}$$

- Outline the steps for generating this distribution using the inverse transform method
- Modify the code for the exponential/gaussian example in OMNET++, to generate a Pareto random variable with a = 1.1 and b = 2.27 , and generate a histogram for 5000 samples. The homework should include your code and the obtained histogram.