Discrete-Event Simulation:

A First Course

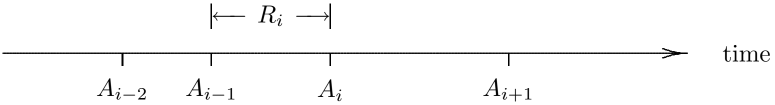
Section 7.3: Continuous RV Applications

Section 7.3: Continuous RV Applications

- **Arrival Process Models**
  - Model *interarrival* times as RV sequence $R_1, R_2, R_3, \ldots$
  - Construct corresponding *arrival* times $A_1, A_2, A_3, \ldots$ defined by

$$A_0 = 0 \qquad \text{and} \qquad A_i = A_{i-1} + R_i \qquad i = 1, 2, \ldots$$

  - By induction, $A_i = R_1 + R_2 + \cdots + R_i \qquad i = 1, 2, \ldots$
  - Since $R_i > 0$, $\qquad 0 = A_0 < A_1 < A_2 < A_3 < \cdots$



Example 7.3.1

- Programs ssq2 and ssq3 generate job arrivals in this way, where $R_1, R_2, R_3, \ldots$ are *Exponential*$(1/\lambda)$

  In both programs, the arrival rate is equal to $\lambda = 0.5$ jobs per unit time
- Programs sis3 and sis4 generate demand instances in this way, with *Exponential*$(1/\lambda)$ interdemand times

  The demand rate corresponds to an average of
  - $\lambda = 30.0$ *actual* demands per time interval in sis3
  - $\lambda = 120.0$ *potential* demands per time interval in sis4

  0-13-142917-5

Definition 7.3.1

- If $R_1, R_2, R_3, \ldots$ is an *iid* sequence of positive interarrival times with $E[R_i] = 1/\lambda > 0$, then the corresponding sequence of arrival times $A_1, A_2, A_3, \ldots$ is a *stationary arrival process* with *rate* λ
- Stationary arrival processes also known as
  - Renewal processes
  - Homogeneous arrival processes
- Arrival rate λ has units "arrivals per unit time"
  - If average interarrival *time* is 0.1 minutes,
  - then the arrival *rate* is 10.0 arrivals per minute
- Stationary arrival processes are "convenient fiction"
- If the arrival rate λ *varies* with time, the arrival process is *nonstationary* (see Section 7.5)

Stationary Poisson Arrival Process

- As in `ssq2`, `ssq3`, `sis3` and `sis4`, with lack of information it is usually most appropriate to assume that the interarrival times are *Exponential*$(1/\lambda)$
- If $R_1, R_2, R_3, \ldots$ is an *iid* sequence of *Exponential*$(1/\lambda)$ interarrival times, the corresponding sequence $A_1, A_2, A_3, \ldots$ of arrival times is a stationary *Poisson* arrival process with rate λ

  Equivalently, for $i = 1, 2, 3, \ldots$ the arrival time A_i is an *Erlang*$(i, 1/\lambda)$ random variable

Algorithm 7.3.1

Algorithm 7.3.1

Given $\lambda > 0$ and $t > 0$, this algorithm generates a realization of a stationary Posson arrival process with rate λ over $(0, t)$

```
a_0 = 0.0; /* a convention */
n = 0;
while(a_n < t) {
  a_{n+1} = a_n + Exponential(1 / λ);
  n++;
}
return a_1, a_2, a_3, ..., a_{n-1};
```

Random Arrivals

- We now demonstrate the interrelation between *Uniform, Exponential and Poisson* random variables
- In the following discussion,
  - $t > 0$ defines a fixed time interval $(0, t)$
  - n represents the number of arrivals in the interval $(0, t)$
  - $r > 0$ is the length of a small subinterval located *at random* interior to $(0, t)$
- Correspondingly,
  - $\lambda = n/t$ is the arrival rate
  - $p = r/t$ is the probability that a particular arrival will be in the subinterval
  - $np = nr/t = \lambda r$ is the expected number of arrivals in the subinterval

Theorem 7.3.1

Theorem (7.3.1)

Let $A_1, A_2, A_3, \ldots$ be an iid sequence of Uniform$(0, t)$ random variables ("unsorted" arrivals). Let the discrete random variable X be the number of A_i that fall in a fixed subinterval of length $r = pt$ interior to $(0, t)$. Then X is a Binomial(n, p) random variable

Proof.

- Each A_i is in the subinterval with probability $p = r/t$
- Define $X_i = \begin{cases} 1 & \text{if } A_i \text{ is in the subinterval} \\ 0 & \text{otherwise} \end{cases}$
- Because $X_1, X_2, \cdots, X_n$ is an *iid* sequence of *Bernoulli*(p) RVs, and $X = X_1 + X_2 + \cdots + X_n$,
  X is a *Binomial*(n, p) random variable

□

Random Arrivals Produce Poisson Counts

- Recall that $\textit{Poisson}(\lambda r) \approx \textit{Binomial}(n, \lambda r/n)$ for large n
- Theorem 7.3.1 can be restated as **Theorem 7.3.2**:

Theorem (7.3.2)

- *Let $A_1, A_2, A_3, \ldots$ be an iid sequence of $\textit{Uniform}(0, t)$ random variables*
- *Let the discrete random variable X be the number of A_i that fall in a fixed subinterval of length $r = pt$ interior to $(0, t)$*
- *If n is large and r/t small, X is indistinguishable from a $\textit{Poisson}(\lambda r)$ random variable with $\lambda = n/t$*

Example 7.3.2

- Suppose $n = 2000$ $\mathit{Uniform}(0, t)$ random variables are generated and tallied into a continuous-data histogram with 1000 bins of size $r = t/1000$
- If bin counts are tallied into a discrete-data histogram
  - Since $\lambda r = (n/t)(t/1000) = 2$,
  - from Thm 7.3.2, the relative frequencies will agree with the pdf of a $\mathit{Poisson}(2)$ random variable

More on Random Arrivals

- If many arrivals occur *at random* with a rate of λ, the number of arrivals X that will occur in an interval of length r is $\mathit{Poisson}(\lambda r)$
- The probability of x arrivals in an interval with length r is

$$\Pr(X = x) = \frac{\exp(-\lambda r)(\lambda r)^x}{x!} \qquad x = 0, 1, 2, \cdots$$

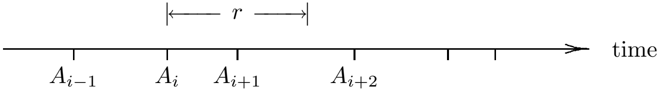
- The probability of no arrivals is: $\Pr(X = 0) = \exp(-\lambda r)$
- The probability of *at least one arrival* is

$$\Pr(X > 0) = 1 - \Pr(X = 0) = 1 - \exp(-\lambda r)$$

- For a fixed λ, the probability of at least one arrival increases with increasing interval length r

Random Arrivals Produce Exponential Interarrivals

- If R represents the time between consecutive arrivals, the possible values of R are $r > 0$
- Consider arrival time A_i selected at random and an interval of length r beginning at A_i



- $R = A_{i+1} - A_i$ will be less than r iff there is at least one arrival in this interval
- The cdf of R is

$$\Pr(R \leq r) = \Pr(\text{at least one arrival}) = 1 - \exp(-\lambda r) \qquad r > 0$$

- R is an *Exponential*$(1/\lambda)$ random variable

Theorem 7.3.3

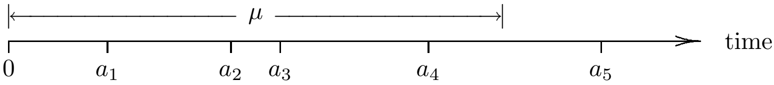
Theorem (7.3.3)

If arrivals occur at random with rate λ, *the corresponding interarrival times form an iid sequence of Exponential*$(1/\lambda)$ *RVs.*

- Proof on previous slide
- Theorem 7.3.3 justifies the use of *Exponential* interarrival times in programs `ssq2`, `ssq2`, `sis2`, `sis4`
  - If we know only that arrivals occur at random with a constant rate λ, the function `GetArrival` in `ssq2` and `ssq3` is appropriate
  - If we know only that demand instances occur at random with a constant rate λ, the function `GetDemand` in `sis3` and `sis4` is appropriate

Generating *Poisson* Random Variates

- Observation:
  - If arrivals occur at random with rate $\lambda = 1$,
  - the number of arrivals X in an interval of length μ will be a *Poisson*(μ) random variate (Thm. 7.3.2)



Example 7.3.3: Generating a *Poisson*(μ) Random Variate

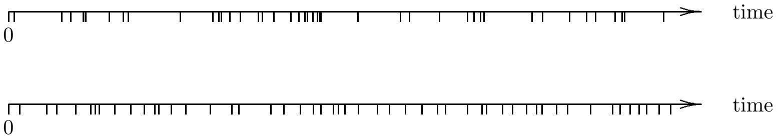
```
a₀ = 0.0;
x = 0;
while (a < μ) {
    a += Exponential(1.0);
    x++;
}
return x-1;
```

Summary of Poisson Arrival Processes

- Given a fixed time interval $(0, t)$, there are two ways of generating a realization of a stationary Poisson arrival process with rate λ
  - Generate the number of arrivals: $n = Poisson(\lambda t)$
    Generate a $Uniform(0, t)$ random variate sample of size n and *sort* to form $0 < a_1 < a_2 < a_3 < \cdots < a_n$
  - Use Algorithm 7.3.1
- Statistically, the two approaches are equivalent
- The first approach is computationally more expensive, especially for large n
- The second approach is always preferred

Summary of Arrival Processes

- The *mode* of the exponential distribution is 0
  - A stationary Poisson arrival process exhibits "clustering"
- The top axis shows a stationary Poisson arrival process with $\lambda = 1$
- The bottom axis shows a stationary arrival process with *Erlang*$(4, 1/4)$ interarrival times



- The stationary Poisson arrival process generalizes to
  - a stationary arrival process when exponential interarrival times are replaced by any continuous RV with positive support
  - a nonstationary Poisson arrival process when λ varies over time

Service Process Models

- No well-defined "default", only application-dependent guidelines:
  - *Uniform*(a, b) service times are usually inappropriate since they rarely "cut off" at a maximum value b
  - Service times are positive, so they can*not* be *Normal*(μ, σ) unless truncated to positive values
  - Positive probability models "with tails", such as the *Lognormal*(a, b) distribution, are candidates
  - If service times are the sum of n *iid Exponential*(b) sub-task times, then the *Erlang*(n, b) model is appropriate

Program ssq4

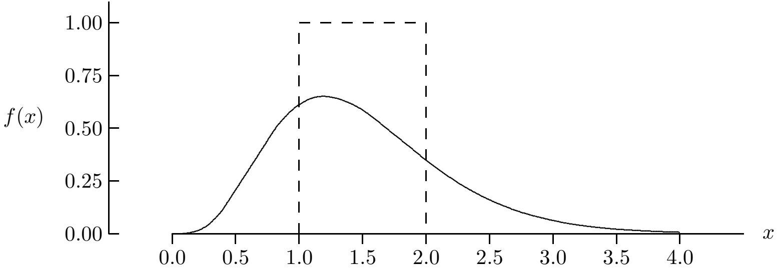
- Program `ssq4` is based on program `ssq3`, but with a more realistic *Erlang*$(5, 0.3)$ service time model

  The corresponding service rate is $2/3$
- As in program `ssq3`, `ssq4` uses *Exponential*(2) random variate interarrivals

  The corresponding arrival rate is $1/2$

Example 7.3.4

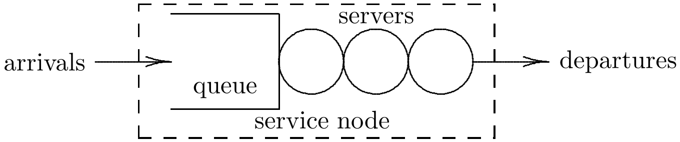
- For both ssq3 and ssq4, the arrival rate is $\lambda = 0.5$ and the service rate is $\nu = 2/3 \simeq 0.667$
- The distribution of service times for two programs is very different



- The solid line is the *Erlang*$(5, 0.3)$ service time pdf in ssq4
- The dashed line represents the *Uniform*$(1, 2)$ pdf in ssq3

Erlang Service Times

- Some service processes can be naturally decomposed into a series of independent "sub-processes"



- The total service time is the sum of each sub-process service time
- If sub-process times are independent, a random variate service time can be generated by generating sub-process times and summing
- In particular, if there are n sub-processes, and each service sub-processes is *Exponential*(b), then the total service time will be *Erlang*(n, b) and the service rate will be $1/nb$

Truncation

- Let X be a continuous random variable with possible values $\mathcal{X}$ and cdf $F(x) = \Pr(X \le x)$
- Suppose we wish to restrict the possible values of X to $(a, b) \subset \mathcal{X}$
- Truncation in the continuous-variable context is similar to, but simpler than, truncation in the discrete-variable context
- X is less or equal to a with probability $\Pr(X \le a) = F(a)$
- X is greater or equal to b with probability

$$\Pr(X \ge b) = 1 - \Pr(X < b) = 1 - F(b)$$

- X is between a and b with probability

$$\Pr(a < X < b) = \Pr(X < b) - \Pr(X \le a) = F(b) - F(a)$$

Two Cases for Truncation

- If a and b are specified, the cdf of X can be used to determine the left-tail, right-tail truncation probabilities

$$\alpha = \Pr(X \leq a) = F(a) \qquad \text{and} \qquad \beta = \Pr(X \leq b) = 1 - F(b)$$

- If α and β are specified, the idf of X can be used to determine left and right truncation points

$$a = F^{-1}(\alpha) \qquad \text{and} \qquad b = F^{-1}(1 - \beta)$$

Both transformations are exact

Example 7.3.5

- Use a *Normal*(1.5, 2.0) random variable to model service times
- Truncate distribution so that
  - Service times are non-negative $(a = 0)$
  - Service times are less than 4 $(b = 4)$

Example 7.3.5

```
α = cdfNormal(1.5, 2.0, a);          /*a is 0.0 */
β = 1.0 - cdfNormal(1.5, 2.0, b); /*b is 4.0 */
```

- The result: $\alpha = 0.2266$ and $\beta = 0.1056$
- Note: the *truncated Normal*(1.5, 2.0) random variable has a mean of 1.85, not 1.5, and a standard deviation of 1.07, not 2.0

Constrained Inversion

- Once α and β are determined, the corresponding truncated random variate can be generated by using constrained inversion

Constrained Inversion

```
u = Uniform(α, 1.0 - β);
return F^{-1}(u);
```

Example 7.3.6

- The idf capability in `rvms` can be used to generate the truncated *Normal*$(1.5, 2.0)$ random variate in Example 7.3.5

Example 7.3.6

```
α = 0.2266274;
β = 0.1056498;
u = Uniform(α, 1.0 - β);
return idfNormal(1.5, 2.0, u);
```

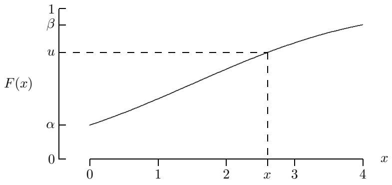
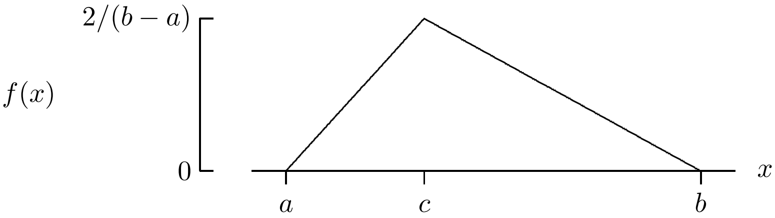


Figure shows $u = 0.7090$ and $x = 2.601$

Triangular Random Variable

- *Triangular*(a, b, c) model should be considered in situations where the finite range of possible values along with the mode is known
- The distribution is appropriate
  - As an alternative to truncating a "traditional" model such as *Erlang*(n, b) or *Lognormal*(a, b)
  - If no other data is available
- Assume that the pdf of the random variable has shape



Properties of the Triangular Distribution

- X is *Triangular*(a, b, c) iff $a < c < b$, $\mathcal{X} = (a, b)$, and the pdf of X is

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & a < x \le c \\ \frac{2(b-x)}{(b-a)(b-c)} & c < x < b \end{cases}$$

- $\mu = \frac{1}{3}(a + b + c)$ and $\sigma = \frac{1}{6}\sqrt{(a-b)^2 + (a-c)^2 + (b-c)^2}$
- The cdf is

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

- The idf is

$$F^{-1}(u) = \begin{cases} a + \sqrt{(b-a)(c-a)u} & 0 < u \le \frac{c-a}{b-a} \\ b - \sqrt{(b-a)(b-c)(1-u)} & \frac{c-a}{b-a} < u < 1 \end{cases}$$