POISSON PROCESS GENERATION

Homogeneous Poisson Processes with rate λ.

- Recall: interarrival times X_i are exponential RVs with rate λ:
  exponential pdf $f(x) = \lambda e^{-\lambda x}$; for $x \in [0, \infty)$,
  with exponential cdf $F(x) = 1 - e^{-\lambda x}$.
  So $X_i = -\ln(U_i)/\lambda,\ U_i \sim Uni(0,1)$; therefore
  RV $T_j = \sum_{i=1}^{j} X_i$ = the time for j^{th} event.
- Algorithm A, to generate all events in $(0, T)$:
  1) initialize $t = -\ln(U_0)/\lambda,\ n = 0$;
  2) while $t < T$, $n = n+1,\ S_n = t,\ t = t - \ln(U_n)/\lambda$ end.
  Output n is # of events in $(0, T)$, and event times $S_1, \ldots, S_n$.
- Example:
  a) in Text Problem 5.24, buses arrive at a sporting event according to a Poisson process with rate 5 per hour; need to simulate bus arrivals over one hour period.

  b) Example Matlab for $\lambda = 3$, $T = 2$

```
T = -log(rand)/3; n = 0;
while T < 2, n = n+1;
  S(n) = T; T = T - log(rand)/3;
end, disp(n), disp(S(1:n))
 7
.14181 .34328 .90224 1.0121 1.2183 1.4602 1.4988
```

  For $K = 10000$ Matlab runs, $E[n] = 6.0205$.

POISSON GENERATION CONTINUED

- Alternate Method: uses discrete Poisson RV $N(T)$, where $N(T)$ = total # events in $(0,T)$, and $mean(N) = T\lambda$.

  Recall Poisson pmf: $p_j = e^{-\lambda T}\frac{(\lambda T)^j}{j!}, j = 0, 1, \ldots$.

  If $n = N(T)$, then $TU_1, TU_2, \ldots, TU_n$ are event times, which can be sorted to get $S_1, \ldots, S_n$.

- Algorithm B, to generate all events in $(0,T)$:

  1) generate $N \sim Poisson(\lambda T)$;

  2) generate $U_1, U_2, \ldots, U_N \sim Uni(0,1)$;

  3) sort $TU_1, \ldots, TU_N$ to get arrival times $S_1, \ldots, S_n$.

  Output N is # of events in $(0,T)$, with event times $S_1, \ldots, S_n$.

  Step 1)? use Poisson RV algorithm from text Chapter 4:

  1a) generate $U \sim Uni(0,1)$;

  1b) set $N = 0,\ p = e^{-\lambda T},\ F = p$;

  1c) while $U > F$, set $N = N + 1$

  set $p = p\lambda T/N,\ F = F + p$

  end;

  1d) Output N, the # of Poisson process events in time T.

- Example: for $\lambda = 3$, $T = 2$

```
U = rand; N = 0;  p = exp(-6); F = p;
while U > F, N = N+1; p = 6*p/N; F = F+p; end
disp(N), disp(sort(2*rand(1,N)))
 6
.042814 .21492 .66472 1.6707 1.7381 1.8899
```

For $K = 10000$ Matlab runs, $E[N] = 5.9872$.

POISSON GENERATION CONTINUED

NonHomogeneous Processes with intensity function $\lambda(t)$ given.

- Interarrival times X_i are exponential RVs with rate $\lambda(t)$,
- "Thinning" Algorithm to generate all $S_i \in (0, T)$:
  1) initialize $t = 0,\ n = 0,\ \lambda = \max_{t\in[0,T]} \lambda(t)$;
  2) set $t = t - \ln(Uni(0,1))/\lambda$, if $t > T$, stop;
  3) if $Uni(0,1) \le \lambda(t)/\lambda$, set $n = n+1,\ S_n = t$;
  4) go to step 2.

  Output n is # of events in $(0, T)$, and event times $S_1, \ldots, S_n$.

- Example, with $\lambda(t) = 6/(t+2)$, $T = 2$; so $\lambda = 3$.

```
t = -log(rand)/3; n = 0;
while t < 2,
  if rand < 2/(t+2), n = n+1; S(n) = t; end
  t = t-log(rand)/3;
end
disp([n S(1:n)])
4  .096807 .73985 1.3257 1.6074
```

POISSON GENERATION CONTINUED

- If $\lambda(t) \not\approx \lambda$, it is more efficient to subdivide $[0, T]$; use $0 = t_0 < t_1 < \ldots < t_{k+1} = T$, with $\lambda_j = \max_{t \in [t_{j-1}, t_j]}(\lambda(t))$.
- Subdivision Algorithm to generate all $S_i \in (0, T)$:
  1) initialize $t = 0,\ n = 0,\ j = 1$;
  2) $X = -\ln(Uni(0,1))/\lambda_j$;
  3) set $t = t + X$, if $t > t_j$, go to step 6;
  4) if $Uni(0,1) \leq \lambda(t)/\lambda_j$, set $n = n + 1,\ S_n = t$;
  5) go to step 2;
  6) if $j = k + 1$ stop;
  7) set $X = (X + t - t_j)\lambda_j/\lambda_{j+1},\ t = t_j,\ j = j + 1$;
  8) goto step 3.

  Output n is # of events in $(0, T)$, and event times $S_1, \ldots, S_n$.

POISSON GENERATION CONTINUED

- Alternate Method, generates S_n's directly, using

  $F_s(x) = P\{s < x|s\}$
  $= 1 - P\{0 \text{ events in } (s, s+x)\}$
  $= 1 - e^{-\int_0^x \lambda(s+t)dt}$.

  Direct Algorithm: initialize $n = 0$, $S_0 = 0$;
  while $S_n < T$, generate $X_{n+1} \sim F_{S_n}$;
  set $S_{n+1} = S_n + X_{n+1}$, $n = n + 1$
  end.

  This method requires easily inverted F_{S_i}s.

- Example: with $\lambda(t) = b/(t + a)$. First compute

$$\int_0^x \lambda(s+y)dy = b\int_0^x (s+y+a)^{-1}dy = b(\ln(x+s+a) - \ln(s+a));$$

  then

$$F_s(x) = 1 - e^{-b(\ln(x+s+a)-\ln(s+a))} = 1 - (\frac{s+a}{x+s+a})^b.$$

  Then solve $U = F_s(X)$;

  Use $F_s^{-1}(U) = (s + a)[U^{-1/b} - 1]$, so event times are

$$S_1 = F_0^{-1}(U_1), \quad S_{n+1} = S_n + F_{S_n}^{-1}(U_n), \ i > 1.$$

POISSON GENERATION CONTINUED

- Example with $\lambda(t) = 6/(t+2)$, $T = 2$; so
  $F_s^{-1}(U) = (s+2)[U^{-1/6} - 1]$.

  Matlab

```
t = 2*(rand^(-1/6)-1); n = 0;
while t < 2, n = n+1;
  S(n) = t; t = t + (2+t)*(rand^(-1/6)-1);
end, disp([n S(1:n)])
 5   .23931 .59698 .77515 1.1221 1.8968
```

  Using $K = 100000$ runs, $E[n] \approx 4.1641$,
  compared to $\int_0^2 \frac{6}{2+x} dx = 6\ln(2) \approx 4.1589$.