King Fahd University of Petroleum & Minerals Computer Engineering Dept

COE 540 – Computer Networks

Term 121

Dr. Ashraf S. Hasan Mahmoud

Rm 22-420

Ext. 1724

Email: ashraf@kfupm.edu.sa

Primer on Probability Theory

- **Source: Chapter 2 and 3 of:**

Alberto Leon-Garcia, Probability and Random Processes for Electrical Engineering, Addison Wisely

Expectation of a Random Variable

- **Expectation of the random variable X can be computed by**

$$E[X] = \sum_{\forall i} x_i P[X = x_i]$$

for discrete variables, or

$$E[X] = \int_{-\infty}^{\infty} t f_x(t) dt$$

for continuous variables.

n^{th} Expectation of a Random Variable

- **The n^{th} expectation of the random variable X can be computed by**

$$E[X^n] = \sum_{\forall i} x^n{}_i P[X = x_i]$$

for discrete variables, or

$$E[X^n] = \int_{-\infty}^{\infty} t^n f_x(t) dt$$

for continuous variables.

Expectation of a Function of the Random Variable

- **Let g(x) be a function of the random variable x, the expectation of g(x) is given by**

$$E[g(x)] = \sum_{\forall i} g(x_i) P[X = x_i]$$

for discrete variables, or

$$E[g(x)] = \int_{-\infty}^{\infty} g(t) f_x(t) dt$$

for continuous variables.

Some Important Random Variables – Discrete Random Variables

- **Bernoulli**
- **Binomial**
- **Geometric**
- **Poisson**

$$\sum_{n=1}^{M} n = \frac{1}{2} M(M+1)$$

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}; |r| < 1$$

$$\sum_{n=0}^{M} \binom{M}{n} r^n = (1+r)^M; |r| < 1$$

$$\sum_{n=0}^{M} r^n = \frac{1-r^{M+1}}{1-r}; |r| < 1, M = 1,2,\ldots$$

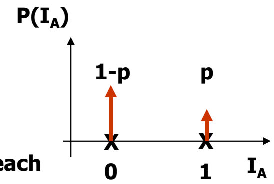
$$\sum_{n=0}^{\infty} n r^{n-1} = \frac{1}{(1-r)^2}; |r| < 1$$

$$\sum_{n=0}^{M} n r^{n-1} = \frac{1+(Mr-M-1)r^M}{(1-r)^2}; |r| < 1$$

Bernoulli Random Variable

- Let A be an event related to the outcomes of some random experiment. The indicator function for A is defined as

$$I_A(\zeta) = 0 \quad \text{if } \zeta \text{ not in A}$$
$$= 1 \quad \text{if } \zeta \text{ is in A}$$

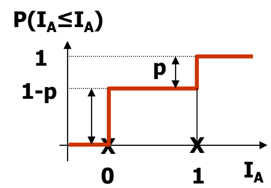


- I_A is random variable since it assigns a number to each outcome in S
- It is discrete r.v. that takes on values from the set {0,1}
- PMF is given by

$$p_I(0) = 1\text{-}p, \; p_I(1) = p$$

where P{A} = p

- Describes the outcome of a Bernoulli trial
- $E[X] = p$, $VAR[X] = p(1-p)$
- $G_X(z) = (1-p+pz)$



Binomial Random Variable

- Suppose a random experiment is repeated n independent times; let X be the number of times a certain event A occurs in these n trials

$$X = I1 + I2 + \ldots + In$$

i.e. X is the sum of Bernoulli trials (X's range = {0, 1, 2, …, n})

- X has the following pmf

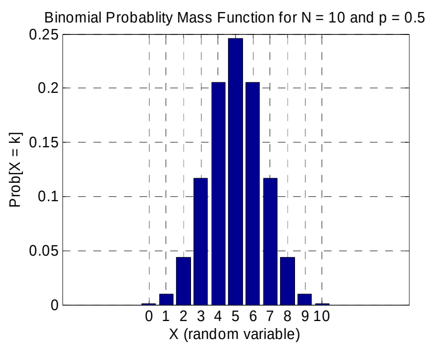
$$\Pr[X = k] = \binom{n}{k} p^k (1-p)^{n-k}$$

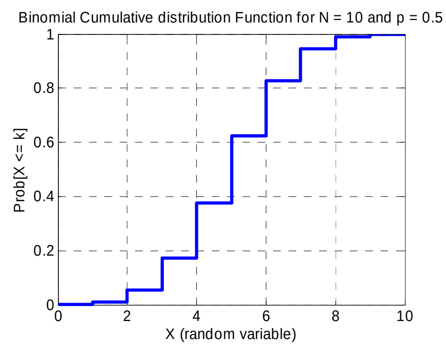
for k=0, 1, 2, …, n

- $E[X] = np$, $Var[X] = np(1-p)$
- $G_X(z) = (1-p + pz)^n$

Binomial Random Variable – cont'd

- **Example**





Geometric Random Variable

- **Suppose a random experiment is repeated - We count the number of M of independent Bernoulli trials UNTIL the first occurrence of a success**
- **M is called geometric random variable**
  - **Range of M = 1, 2, 3, …**
- **X has the following pmf**

$$\Pr[X = k] = (1-p)^{k-1} p$$

for k=1, 2, 3, …

- **E[X] = 1/p, Var[X] = (1-p)/p^2**
- **G_X(z) = pz/(1-(1-p)z))**

Geometric Random Variable - 2

- **Suppose a random experiment is repeated - We count the number of M of independent Bernoulli trials BEFORE the first occurrence of a success**
- **M is called geometric random variable**
  - **Range of M = 0, 1, 2, 3, …**
- **X has the following pmf**

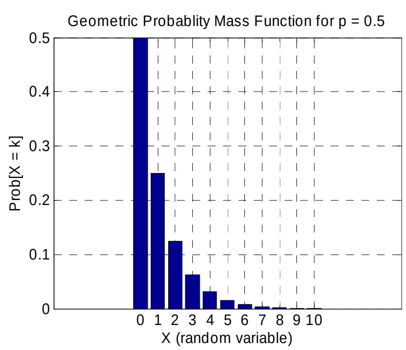
$$\Pr[X = k] = (1-p)^k p$$

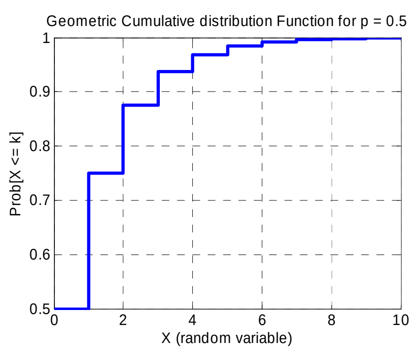
for k=0,1, 2, 3, …

- **E[X] = (1-p)/p,   Var[X] = (1-p)/p²**
- **$G_X(z)$ = pz/(1-(1-p)z))**

Geometric Random Variable – cont'd

- **Example: p = 0.5; X is number of failures BEFORE a success (2nd type)**
- **Note Matlab's version of geometric distribution is the 2nd type**





Poisson Random Variable

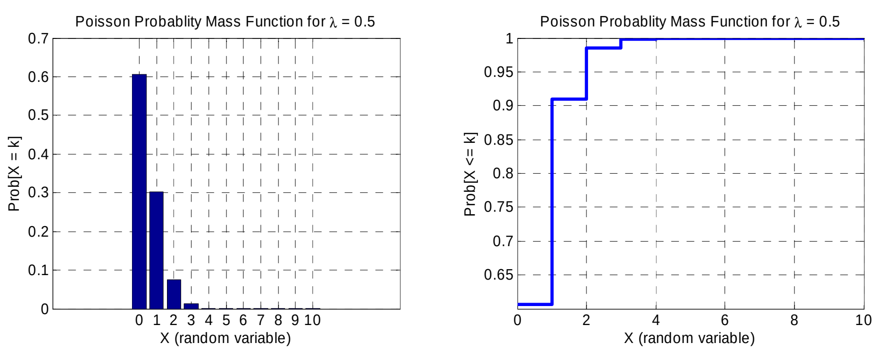
- **In many applications we are interested in counting the number of occurrences of an event in a certain time period**
- **The pmf is given by** $$\Pr[X = k] = \frac{\alpha^k}{k!}e^{-\alpha}$$

For k=0, 1, 2, … ; α is the average number of event occurrences in the specified interval

- **E[X] = α, Var[X] = α**
- **$G_X(z) = e^{\alpha(z-1)}$**
- **Remember: time between events is exponentially distributed!**
- **Poisson is the limiting case for Binomial as n→∞, p →0, such that np = α**

Poisson Random Variable – cont'd

- **Example:**



Matlab Code to Plot Distributions

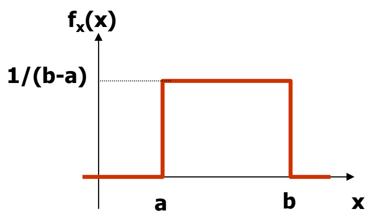
```
% plot distributions
% see "help stats"
clear all
FontSize = 14;
LineWidth = 3;
% Binomial
N = 10; X = [0:1:N]; P = 0.5;
ybp = binopdf(X, N, P); % get PMF
ybc = binocdf(X, N, P); % get CDF
figure(1); set(gca,'FontSize', FontSize);
bar(X, ybp);
title(['Binomial Probablity Mass Function for
    N = ' ...
     num2str(N) ' and p = ' num2str(P)]);
xlabel('X (random variable)');
ylabel('Prob[X = k]'); grid
figure(2); set(gca,'FontSize', FontSize);
stairs(X, ybc,'LineWidth', LineWidth);
title(['Binomial Cumulative distribution
    Function for N = ' ...
     num2str(N) ' and p = ' num2str(P)]);
xlabel('X (random variable)');
ylabel('Prob[X <= k]'); grid
% Geometric
P = 0.5; X = [0:10];
ygp = geopdf(X, P); % get pdf
ygc = geocdf(X, P); % get cdf
figure(3); set(gca,'FontSize', FontSize);
bar(X, ygp);
title(['Geometric Probablity Mass Function for
    p = ' num2str(P)]);
xlabel('X (random variable)');
ylabel('Prob[X = k]'); grid
figure(4); set(gca,'FontSize', FontSize);
stairs(X, ygc,'LineWidth', LineWidth);
title(['Geometric Cumulative distribution
    Function for p = ' num2str(P)]);
xlabel('X (random variable)');
ylabel('Prob[X <= k]'); grid
% Poisson
Lambda = 0.5; X     = [0:10];
ypp   = poisspdf(X, Lambda);
ypc   = poisscdf(X, Lambda);
figure(5); set(gca,'FontSize', FontSize);
bar(X, ypp);
title(['Poisson Probablity Mass Function for
    \lambda = ' num2str(Lambda)]);
xlabel('X (random variable)');
ylabel('Prob[X = k]'); grid
figure(6); set(gca,'FontSize', FontSize);
stairs(X, ypc,'LineWidth', LineWidth);
title(['Poisson Probablity Mass Function for
    \lambda = ' num2str(Lambda)]);
xlabel('X (random variable)');
ylabel('Prob[X <= k]'); grid
```

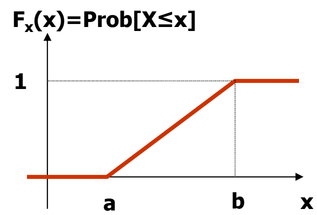

Some Important Random Variables – Continuous Random Variables

- **Uniform**
- **Exponential**
- **Gaussian (Normal)**
- **Rayleigh**
- **Gamma**
- **M-Erlang**
- **....**

Uniform Random Variables

- **Realizations of the r.v. can take values from the interval [a, b]**
- **PDF $f_X(x) = 1/(b-a)$ $a \le x \le b$**
- **$E[X] = (a+b)/2$, $Var[X] = (b-a)^2/12$**
- **$\Phi_X(\omega) = [e^{j\omega b} - e^{j\omega a}]/(j\omega(b-a))$**





Exponential Random Variables

- **The exponential r.v. X with parameter λ has pdf**

$$f_X(x) = \begin{cases} 0 & x < 0 \\ \lambda e^{-\lambda x} & x \ge 0 \end{cases}$$

- **And CDF given by**

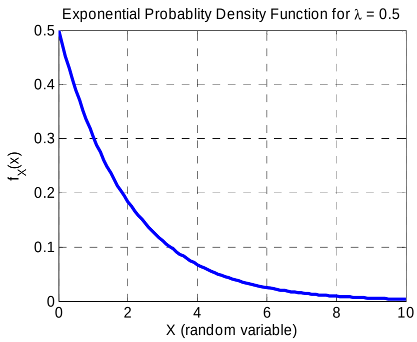
$$F_X(x) = \begin{cases} 0 & x < 0 \\ 1 - e^{-\lambda x} & x \ge 0 \end{cases}$$

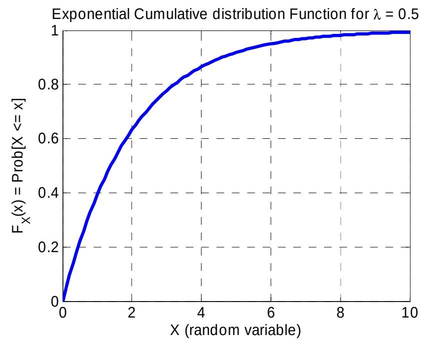
- **Range of X: $[0, \infty)$**
- **$E[X] = 1/\lambda$, $Var[X] = 1/\lambda^2$**
- **$\Phi_X(\omega) = \lambda/(\lambda - j\omega)$**

This means:
$Prob[X \le x] = 1 - e^{-\lambda x}$, or
$Prob[X > x] = e^{-\lambda x}$

Exponential Random Variables – cont'd

- **Example:**
  - **Note the mean is $1/\lambda = 2$**





Exponential Random Variables – Memoryless Property

- **The exponential r.v. is the only continuous r.v. with the memoryless property!!**
- **Memoryless Property:**
  $P[X>t+h / X>t] = P[X>h]$

i.e. the probability of having to wait h additional seconds given that one has already been waiting t second IS EXACTLY equal to the probability of waiting h seconds when one first begins to wait

<u>Proof</u>:

$$P[X>t+h / X>t] = \frac{P[(X > t+h) \cap (X > t)]}{P[(X > t)]}$$

$$= \frac{P[(X > t+h)}{P[X > t]} = \frac{e^{-\lambda(t+h)}}{e^{-\lambda t}}$$

$$= e^{-\lambda h}$$

$$= P[X > h]$$

Gaussian (Normal) Random Variable

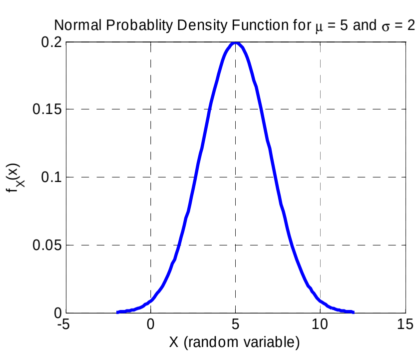
- **Rises in situations where a random variable X is the sum of a large number of "small" random variables – central limit theorem**
- **PDF** $$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

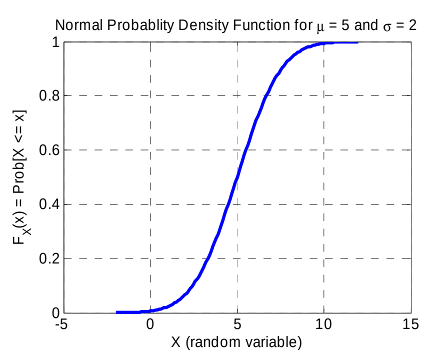
For -∞<x< ∞; μ and σ > 0 are real numbers

- **E[X] = μ, Var[X] =** σ^2
- $\Phi_X(\omega) = e^{j\mu\omega - \sigma^2\omega^2/2}$
- **Under wide range of conditions X can be used to approximate the sum of a large number of independent random variables**

Gaussian (Normal) Random Variable – cont'd

- **Example:**





Matlab Code to Plot Distributions

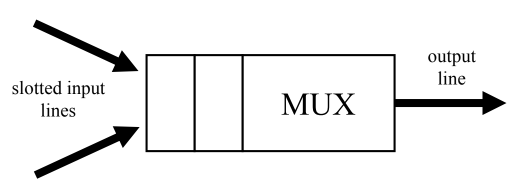
```
% plot distributions
% see "help stats"
clear all
FontSize = 14;
LineWidth = 3;
% exponential
X = [0:.1:10]; Lambda = 0.5;
yep = exppdf(X, 1/Lambda); % get PDF
yec = expcdf(X, 1/Lambda); % get CDF
figure(1); set(gca,'FontSize', FontSize);
plot(X, yep, 'LineWidth', LineWidth);
title(['Exponential Probablity Density Function for \lambda = ' ...
    num2str(Lambda)]);
xlabel('X (random variable)');
ylabel('f_X(x)'); grid
figure(2); set(gca,'FontSize', FontSize);
plot(X, yec,'LineWidth', LineWidth);
title(['Exponential Cumulative Distribution Function for \lambda = ' ...
    num2str(Lambda)]);
xlabel('X (random variable)');
ylabel('F_X(x) = Prob[X <= x]'); grid
% normal
X = [-2:.1:12]; Mu = 5; Sigma = 2;
ynp = normpdf(X, Mu, Sigma); % get PDF
ync = normcdf(X, Mu, Sigma); % get CDF
figure(3); set(gca,'FontSize', FontSize);
plot(X, ynp, 'LineWidth', LineWidth);
title(['Normal Probablity Density Function for \mu = ' ...
    num2str(Mu) ' and \sigma = ' num2str(Sigma)]);
xlabel('X (random variable)');
ylabel('f_X(x)'); grid
figure(4); set(gca,'FontSize', FontSize);
plot(X, ync,'LineWidth', LineWidth);
title(['Normal Probablity Density Function for \mu = ' ...
    num2str(Mu) ' and \sigma = ' num2str(Sigma)]);
xlabel('X (random variable)');
ylabel('F_X(x) = Prob[X <= x]'); grid
% Rayleigh
X = [0:.1:10]; Alpha = 2;
yrp = raylpdf(X, Alpha); % get PDF
yrc = raylcdf(X, Alpha); % get CDF
figure(5); set(gca,'FontSize', FontSize);
plot(X, yrp, 'LineWidth', LineWidth);
title(['Rayleigh Probablity Density Function for \alpha = ' ...
    num2str(Alpha)]);
xlabel('X (random variable)');
ylabel('f_X(x)'); grid
figure(6); set(gca,'FontSize', FontSize);
plot(X, yrc,'LineWidth', LineWidth);
title(['Rayleigh Probablity Density Function for \alpha = ' ...
    num2str(Alpha)]);
xlabel('X (random variable)');
ylabel('F_X(x) = Prob[X <= x]'); grid
```


Example 10: Multiplexer

<u>Problem</u>: Data in the form of fixed-length packets arrive in slots on both of the input lines of a multiplexer. A slot contains a packet with probability p, independent of the arrivals during other slots or on the other line. The multiplexer transmits one packet per time slot and has the capacity to store two messages only. If no room for a packet is found, the packet is dropped.

a) Draw the state diagram and define the matrix P

b) Compute the throughput of the multiplexer for p = 0.3



Example 10: Multiplexer – cont'd

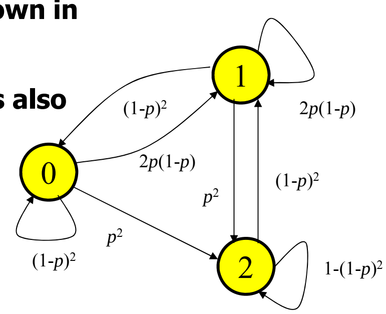
<u>Solution</u>: In any slot time, the arrivals pmf is given by

P(j cells arrive) = $(1-p)^2$ j=0
$2p(1-p)$ j=1
p^2 j=2

Let the state be the number of packets in the buffer, then the state diagram is shown in figure.

The corresponding transition matrix is also given below

$$P = \begin{bmatrix} (1-p)^2 & 2p(1-p) & p^2 \\ (1-p)^2 & 2p(1-p) & p^2 \\ 0 & (1-p)^2 & 1-(1-p)^2 \end{bmatrix}$$



Example 10: Multiplexer – cont'd

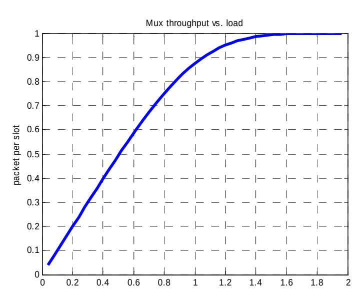
<u>Solution-cont'd</u>:

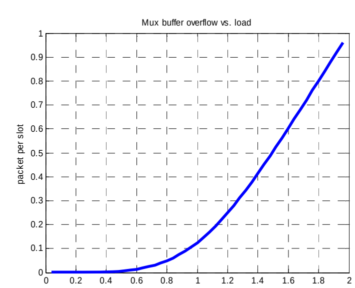
Load: average arrivals = 2*p* packets/slot

Throughput: $\pi_1 + \pi_2$

Buffer overflow = Prob(two packet arrivals while in state 2)
= Prob(two arrivals) X π_2
= $p^2 \pi_2$

The graphs below show the relation of load versus –throughput and buffer overflow for the MUX





Example 10: Multiplexer – cont'd

<u>Solution-cont'd</u>:

The matlab code used for plotted previous results is shown below.

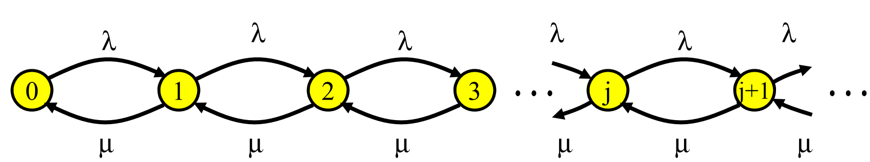
Make sure you understand the matrix formulation and the solution for the steady state probability vector π

```
clear all
Step        = 0.02;
ArrivalProb = [Step:Step:1-Step];
A    = zeros(4,3);
E    = zeros(4,1);
E(4) = 1;
for i=1:length(ArrivalProb)
    p = ArrivalProb(i);
    P = [(1-p)^2  2*p*(1-p)  p^2; ...
         (1-p)^2  2*p*(1-p)  p^2; ...
          0         (1-p)^2    1-(1-p)^2];
   A(1:3,:) = (P - eye(3))';
   A(4,:)   = ones(1,3);
   E(4)     = 1;
   SteadyStateP = A\E;
   % Prob(packet is lost) = Prob(2 arrivals) X
   %                        Prob(being in state 2);
   DropProb(i) = p^2*SteadyStateP(3);
   Throughput(i) = sum(SteadyStateP(2:3));
end
```

```
% matlab code continued
figure(1),
h = plot(2*ArrivalProb, Throughput);
set(h, 'LineWidth', 3);
title('Mux throughput vs. load');
ylabel('packet per slot');
xlabel('packet per slot');
grid
figure(2),
h = plot(2*ArrivalProb, DropProb);
set(h, 'LineWidth', 3);
title('Mux buffer overflow vs. load');
ylabel('packet per slot');
xlabel('packet per slot');
grid
```


Example 12:

- **<u>Problem</u>: The M/M/1 single-server queueing system**



The corresponding rate transition matrix is given by

$$\Gamma = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \cdots \\ 0 & 0 & \mu & -(\lambda+\mu) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Example 12: cont'd

- **Answer: The state transition rates:**
  - Customers arrive with rate $\lambda \rightarrow \gamma_{i,i+1} = \lambda$ for i = 0, 1, 2, …
  - When system is not empty, customers depart at rate $\mu \rightarrow \gamma_{i,i-1} = \mu$ for i = 1, 2, 3, …

- **The global balance equations:**

$$\lambda p_0 = \mu p_1 \quad \text{for } j = 0$$

$$(\lambda + \mu)p_j = \lambda p_{j-1} + \mu p_{j+1} \quad \text{for } j=1, 2, \ldots$$

$$\rightarrow \lambda p_j - \mu p_{j+1} = \lambda p_{j-1} - \mu p_j \quad \text{for } j=1,2, \ldots$$
$$= \text{constant}$$

Example 12: cont'd

- **Answer:**

For j = 1, we have

$$\lambda p_0 - \mu p_1 = \text{constant}$$

Therefore the constant is equal to zero.

Hence,

$$\lambda p_{j-1} = \mu p_j \text{ or}$$
$$p_j = (\lambda/\mu)\, p_{j-1} \quad \text{for } j=1,2, \ldots$$

By simple induction:

$$p_j = \rho^j p_0$$

where $\rho = \lambda/\mu$

Example 12: cont'd

- **Answer:**

To obtain p_0, we use the fact that

$$1 = \sum_j p_j = (1+\rho+\rho^2+\ldots)p_0$$

note the above series converges only for $\rho < 1$ or equivalently $\lambda < \mu$

Therefore, $p_0 = 1-\rho$

In general, the steady state pmf for the M/M/1 queue is given by

$$p_j = (1-\rho)\rho^j$$

References

- **Alberto Leon-Garcia, Probability and Random Processes for Electrical Engineering, Addison Wesley, 1989**