More General Systems

- G/G/1 queue
More General Systems

- G/G/1 queue
  - matrix geometric techniques
  - other matrix techniques
  - bounds
More General Systems

- G/G/1 queue
  - matrix geometric techniques
  - other matrix techniques
  - bounds

- other service disciplines
  - priority queueing
More General Systems

- G/G/1 queue
  - matrix geometric techniques
  - other matrix techniques
  - bounds

- other service disciplines
  - priority queueing
  - last come first serve without preemptions (LCFS); with preemptions LCSFPR
More General Systems

- G/G/1 queue
  - matrix geometric techniques
  - other matrix techniques
  - bounds

- other service disciplines
  - priority queueing
  - last come first serve without preemptions (LCFS); with preemptions LCSFPR
  - processor sharing
M/M/1 Queue Revisited

- one minor modification, arrival rate $\lambda'$ when noone in system; $\lambda$ otherwise
\textbf{M/M/1 Queue Revisited}

- One minor modification, arrival rate \( \lambda' \) when no one in system; \( \lambda \) otherwise

- Infinitesimal generator \( Q \)
M/M/1 Queue Revisited

- one minor modification, arrival rate $\lambda'$ when no one in system; $\lambda$ otherwise

- infinitesimal generator $Q$

$$Q = \begin{bmatrix}
-\lambda' & \lambda' & 0 & 0 & 0 & \cdots \\
\mu & -(\lambda + \mu) & \lambda & 0 & \cdots \\
0 & \mu & -(\lambda + \mu) & \lambda & \cdots \\
0 & 0 & \mu & -(\lambda + \mu) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}$$
• balance equations:
• balance equations:

\[-\pi_0 \lambda' + \pi_1 \mu = 0\]
-π₀λ' + π₁μ = 0

π₀λ' − π₁(λ + μ) + π₂μ = 0
• balance equations:

\[-\pi_0 \lambda' + \pi_1 \mu = 0\]

\[\pi_0 \lambda' - \pi_1 (\lambda + \mu) + \pi_2 \mu = 0\]

\[\pi_{j-1} \lambda - \pi_j (\lambda + \mu) + \pi_{j+1} \mu = 0, \quad j = 2, 3, \ldots\]
balance equations:

\[-\pi_0 \lambda' + \pi_1 \mu = 0\]

\[\pi_0 \lambda' - \pi_1 (\lambda + \mu) + \pi_2 \mu = 0\]

\[\pi_{j-1} \lambda - \pi_j (\lambda + \mu) + \pi_{j+1} \mu = 0, \quad j = 2, 3, \ldots\]
• guess $\pi_j = \pi_1 r^{j-1}$ implies

$$\pi_1 r^j \mu - \pi_1 r^{j-1} (\lambda + \mu) + \pi_1 r^{j-2} \lambda = 0$$
• guess $\pi_j = \pi_1 r^{j-1}$ implies

$$\pi_1 r^j \mu - \pi_1 r^{j-1} (\lambda + \mu) + \pi_1 r^{j-2} \lambda = 0$$

or

$$r^2 \mu - r(\lambda + \mu) + \lambda = 0$$
M/M/1 Queue cont.

Two solutions to last equation, \( r = \frac{\lambda}{\mu}, 1 \). \( r = 1 \) does not make sense since \( \sum_{j=1}^{\infty} \pi_j = \infty \) Therefore, when \( \frac{\lambda}{\mu} < 1 \), solution \( r = \frac{\lambda}{\mu} \) makes sense, and \( \pi_j = \pi_1 \rho^{j-1} \) (where \( \rho \equiv \frac{\lambda}{\mu} \)).
Two solutions to last equation, \( r = \lambda/\mu, 1 \). \( r = 1 \) does not make sense since \( \sum_{j=1}^{\infty} \pi_j = \infty \). Therefore, when \( \lambda/\mu < 1 \), solution \( r = \lambda/\mu \) makes sense, and \( \pi_j = \pi_1 \rho^{j-1} \) (where \( \rho \equiv \lambda/\mu \)).
M/M/1 Queue cont.

Q: how to obtain $\pi_0$ and $\pi_1$?
Q: how to obtain $\pi_0$ and $\pi_1$?

\[
(\pi_0, \pi_1) \begin{bmatrix}
-\lambda' & \lambda' \\
\mu & -(\lambda + \mu) + \rho \mu 
\end{bmatrix} = [0, 0]
\]
**M/M/1 Queue cont.**

**Q:** how to obtain $\pi_0$ and $\pi_1$?

$$(\pi_0, \pi_1) \begin{bmatrix} -\lambda' & \lambda' \\ \mu & -(\lambda + \mu) + \rho \mu \end{bmatrix} = [0 \ 0]$$

along with

$$1 = \pi_0 + \pi_1 \sum_{j=1}^{\infty} \rho^{j-1}$$
**M/M/1 Queue cont.**

**Q:** how to obtain $\pi_0$ and $\pi_1$?

\[
(\pi_0, \pi_1) \begin{bmatrix}
-\lambda' & \lambda' \\
\mu & -(\lambda + \mu) + \rho \mu
\end{bmatrix} = [0 \ 0]
\]

along with

\[
1 = \pi_0 + \pi_1 \sum_{j=1}^{\infty} \rho^{j-1} = \pi_0 + \pi_1/(1 - \rho)
\]
\[ (\pi_0, \pi_1) \begin{bmatrix} -\lambda' & \lambda' \\ \mu & -(\lambda + \mu) + \rho \mu \end{bmatrix} = [0 \ 0] \]

along with

\[
1 = \pi_0 + \pi_1 \sum_{j=1}^{\infty} \rho^{j-1} = \pi_0 + \pi_1/(1 - \rho)
\]
Finally, $\pi_0$ and $\pi_1$ are solutions to
Finally, $\pi_0$ and $\pi_1$ are solutions to

$$
(\pi_0, \pi_1) \begin{bmatrix}
1 & \lambda' \\
1/(1 - \rho) & -(\lambda + \mu) + \rho \lambda
\end{bmatrix} = [1 \ 0]
$$
Finally, \( \pi_0 \) and \( \pi_1 \) are solutions to

\[
(\pi_0, \pi_1) \begin{bmatrix}
1 & \lambda' \\
1/(1 - \rho) & -(\lambda + \mu) + \rho \lambda
\end{bmatrix} = [1 \ 0]
\]
M/M/1 Queue cont.

Q: what is $E[N_q]$?
M/M/1 Queue cont.

Q: what is $E[N_q]$?

$E[N_q] = $
Q: what is $E[N_q]$?

$$E[N_q] = \sum_{j=1}^{\infty} (j - 1)\pi_j = \pi_1 \sum_{j=1}^{\infty} (j - 1)\rho^{j-1}$$
**M/M/1 Queue cont.**

**Q:** what is $E[N_q]$?

\[
E[N_q] = \sum_{j=1} \pi_j (j - 1) = \pi_1 \sum_{j=1} (j - 1) \rho^{j-1}
\]

\[
= \pi_1 \rho (1 - \rho)^{-2}
\]
Hyperexponential Distribution ($H_r$)

- An rv $X$ with an $r$ stage *hyperexponential* distribution ($H_r$) distr. has following density function
Hyperexponential Distribution \( (H_r) \)

- An rv \( X \) with an \( r \) stage hyperexponential distribution \( (H_r) \) distr. has following density function

\[
f_X(t) = \]

Hyperexponential Distribution ($H_r$)

- An rv $X$ with an $r$ stage hyperexponential distribution ($H_r$) distr. has following density function

$$f_X(t) = \sum_{i=1}^{r} \alpha_i \mu_i e^{-\mu_i t}, \quad t \geq 0$$
Hyperexponential Distribution \((H_r)\)

- An rv \(X\) with an \(r\) stage *hyperexponential* distribution \((H_r)\) distr. has following density function

\[
f_X(t) = \sum_{i=1}^{r} \alpha_i \mu_i e^{-\mu_i t}, \quad t \geq 0
\]

\((\sum_{i=1}^{r} \alpha_i = 1)\)
Hyperexponential Distribution \((H_r)\)

- An rv \(X\) with an \(r\) stage hyperexponential distribution \((H_r)\) distr. has following density function

\[
f_X(t) = \sum_{i=1}^{r} \alpha_i \mu_i e^{-\mu_i t}, \quad t \geq 0
\]

\((\sum_{i=1}^{r} \alpha_i = 1)\) with mean and coeff. of variation
Hyperexponential Distribution \((H_r)\)

- An rv \(X\) with an \(r\) stage \textit{hyperexponential} distribution \((H_r)\) distr. has following density function

\[
f_X(t) = \sum_{i=1}^{r} \alpha_i \mu_i e^{-\mu_i t}, \quad t \geq 0
\]

\((\sum_{i=1}^{r} \alpha_i = 1)\) with mean and coeff. of variation

\[
E[X] = \sum_{i=1}^{r} \frac{\alpha_i}{\mu_i}
\]
\[ C_X^2 = \frac{2 \sum_{i=1}^{r} \alpha_i / \mu_i^2}{\left( \sum_{i=1}^{r} \alpha_i / \mu_i \right)^2} - 1 \]
\[ C_X^2 = \frac{2 \sum_{i=1}^{r} \alpha_i/\mu_i^2}{\left( \sum_{i=1}^{r} \alpha_i/\mu_i \right)^2} - 1 \]

Can show \( C_X^2 \geq 1 \) using Cauchy-Schwarz inequality
M/H₂/1 Queue

• Poisson arrival process, $\lambda'$ when no jobs, $\lambda$ when one or more jobs
M/H₂/1 Queue

- Poisson arrival process, $\lambda'$ when no jobs, $\lambda$ when one or more jobs

- service times given by $\mathcal{H}_2$ distr. with parameters $\alpha, \mu_1; \bar{\alpha}, \mu_2$
M/H₂/1 Queue

- Poisson arrival process, \( \lambda' \) when no jobs, \( \lambda \) when one or more jobs
- Service times given by \( H_2 \) distr. with parameters \( \alpha, \mu_1; \bar{\alpha}, \mu_2 \)
- System state - \((n, s)\)
M/H₂/1 Queue

- Poisson arrival process, \( \lambda' \) when no jobs, \( \lambda \) when one or more jobs

- service times given by \( H_2 \) distr. with parameters \( \alpha, \mu_1; \bar{\alpha}, \mu_2 \)

- system state - \((n, s)\)
  - \(n\) no., in system
M/H₂/1 Queue

- Poisson arrival process, $\lambda'$ when no jobs, $\lambda$ when one or more jobs
- Service times given by $\mathcal{H}_2$ distr. with parameters $\alpha, \mu_1$; $\bar{\alpha}, \mu_2$
- System state - $(n, s)$
  - $n$ no., in system
  - $s$ exponential stage of job in service
M/H₂/1 **Queue**

- Poisson arrival process, \( \lambda' \) when no jobs, \( \lambda \) when one or more jobs

- service times given by \( H_2 \) distr. with parameters \( \alpha, \mu_1; \bar{\alpha}, \mu_2 \)

- system state - \((n, s)\)
  - \( n \) no,. in system
  - \( s \) exponential stage of job in service
  - when \( n = 0, s = 0 \)
**\( M/H_2/1 \) Queue**

- Poisson arrival process, \( \lambda' \) when no jobs, \( \lambda \) when one or more jobs

- service times given by \( H_2 \) distr. with parameters \( \alpha, \mu_1; \bar{\alpha}, \mu_2 \)

- system state - \((n, s)\)
  - \( n \) no. in system
  - \( s \) exponential stage of job in service
  - when \( n = 0, s = 0 \)
• state transition diagram
• state transition diagram
$M/H_2/1$ Queue

Infinitesimal generator $Q$
\textbf{M/H}_2/1 \textbf{ Queue}

Infinitesimal generator $Q$:

$$Q = \begin{bmatrix}
-\lambda' & \lambda' \alpha & \lambda' \bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots
\end{bmatrix}$$
**M/H_2/1 Queue**

Infinitesimal generator $Q$

\[
Q = \begin{bmatrix}
-\lambda' & \lambda'\alpha & \lambda'\bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots
\end{bmatrix}
\]
$M/H_2/1$ Queue

Infinitesimal generator $Q$

$$Q = \begin{bmatrix}
-\lambda' & \lambda'\alpha & \lambda'\bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & \cdots \\
\end{bmatrix}$$
M/H₂/1 Queue

Infinitesimal generator Q

\[
Q = \begin{bmatrix}
-\lambda' & \lambda'\alpha & \lambda'\bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
0 & \alpha\mu_1 & \bar{\alpha}\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & \cdots \\
\end{bmatrix}
\]
**M/H₂/1 Queue**

**Infinitesimal generator Q**

\[
Q = \begin{bmatrix}
-\lambda' & \lambda'\alpha & \lambda'\bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
0 & \alpha\mu_1 & \bar{\alpha}\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & \cdots \\
0 & \alpha\mu_2 & \bar{\alpha}\mu_2 & 0 & -a_2 & 0 & \lambda & \cdots \\
\end{bmatrix}
\]
**M/H₂/1 Queue**

Infinitesimal generator Q

\[
Q = \begin{bmatrix}
-\lambda' & \lambda'\alpha & \lambda'\bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
0 & \alpha\mu_1 & \bar{\alpha}\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & \cdots \\
0 & \alpha\mu_2 & \bar{\alpha}\mu_2 & 0 & -a_2 & 0 & \lambda & \cdots \\
0 & 0 & 0 & \alpha\mu_1 & \bar{\alpha}\mu_1 & -a_1 & 0 & \cdots \\
\end{bmatrix}
\]
**M/H₂/1 Queue**

Infinitesimal generator $Q$

\[ Q = \begin{bmatrix}
-\lambda' & \lambda'\alpha & \lambda'\bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
0 & \alpha\mu_1 & \bar{\alpha}\mu_1 & -a_1 & 0 & \lambda & 0 & \cdots \\
0 & \alpha\mu_2 & \bar{\alpha}\mu_2 & 0 & -a_2 & 0 & \lambda & \cdots \\
0 & 0 & 0 & \alpha\mu_1 & \bar{\alpha}\mu_1 & -a_1 & 0 & \cdots \\
0 & 0 & 0 & \alpha\mu_2 & \bar{\alpha}\mu_2 & 0 & -a_2 & \cdots 
\end{bmatrix} \]
**M/H₂/1 Queue**

**Infinitesimal generator Q**

\[
Q = \begin{bmatrix}
-\lambda' & \lambda' \alpha & \lambda' \bar{\alpha} & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mu_1 & -a_1 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
0 & \alpha \mu_1 & \bar{\alpha} \mu_1 & -a_1 & 0 & \lambda & 0 & \cdots \\
0 & \alpha \mu_2 & \bar{\alpha} \mu_2 & 0 & -a_2 & 0 & \lambda & \cdots \\
0 & 0 & 0 & \alpha \mu_1 & \bar{\alpha} \mu_1 & -a_1 & 0 & \cdots \\
0 & 0 & 0 & \alpha \mu_2 & \bar{\alpha} \mu_2 & 0 & -a_2 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots
\end{bmatrix}
\]
where \( a_i = \lambda + \mu_i, \ i = 1, 2 \)
Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \]
Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \quad B_{0,1} \equiv [\lambda' \alpha \quad \lambda' \bar{\alpha}], \]
Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \quad B_{0,1} \equiv [\lambda'\alpha \quad \lambda'\bar{\alpha}], \quad B_{1,0} \equiv \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \]
Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \quad B_{0,1} \equiv [\lambda' \alpha \quad \lambda' \bar{\alpha}], \quad B_{1,0} \equiv \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \]

\[ A_0 \equiv \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}, \]
M/H₂/1 Queue

Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \quad B_{0,1} \equiv [\lambda' \alpha \quad \lambda' \bar{\alpha}], \quad B_{1,0} \equiv \begin{bmatrix} \mu_1 \\
\mu_2 \end{bmatrix} \]

\[ A_0 \equiv \begin{bmatrix} \lambda & 0 \\
0 & \lambda \end{bmatrix}, \quad A_2 \equiv \begin{bmatrix} \alpha \mu_1 & \bar{\alpha} \mu_1 \\
\alpha \mu_2 & \bar{\alpha} \mu_2 \end{bmatrix}, \]
Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \quad B_{0,1} \equiv [\lambda' \alpha \quad \lambda' \bar{\alpha}], \quad B_{1,0} \equiv \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \]

\[ A_0 \equiv \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}, \quad A_2 \equiv \begin{bmatrix} \alpha \mu_1 & \bar{\alpha} \mu_1 \\ \alpha \mu_2 & \bar{\alpha} \mu_2 \end{bmatrix}, \]

\[ A_1 \equiv \begin{bmatrix} -(\lambda + \mu_1) & 0 \\ 0 & -(\lambda + \mu_2) \end{bmatrix} \]
Define the following submatrices

\[ B_{0,0} \equiv [-\lambda'], \quad B_{0,1} \equiv [\lambda' \alpha \: \lambda' \bar{\alpha}], \quad B_{1,0} \equiv \begin{bmatrix} \mu_1 \\
\mu_2 \end{bmatrix} \]

\[ A_0 \equiv \begin{bmatrix} \lambda & 0 \\
0 & \lambda \end{bmatrix}, \quad A_2 \equiv \begin{bmatrix} \alpha \mu_1 & \bar{\alpha} \mu_1 \\
\alpha \mu_2 & \bar{\alpha} \mu_2 \end{bmatrix}, \]

\[ A_1 \equiv \begin{bmatrix} -(\lambda + \mu_1) & 0 \\
0 & -(\lambda + \mu_2) \end{bmatrix} \]
M/H₂/1 Queue

Let \( \pi_i = (\pi_{i,1}, \pi_{i,2}) \), \( i = 1, 2, \ldots \) and \( \pi_0 = (\pi_{0,0}) \).
M/H_2/1 Queue

Let \( \pi_i = (\pi_{i,1}, \pi_{i,2}) \), \( i = 1, 2, \ldots \) and \( \pi_0 = (\pi_{0,0}) \).
M/H₂/1 Queue

Let $\pi_i = (\pi_{i,1}, \pi_{i,2})$, $i = 1, 2, \ldots$ and $\pi_0 = (\pi_{0,0})$.

$$Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
\end{bmatrix}$$
Let $\pi_i = (\pi_{i,1}, \pi_{i,2})$, $i = 1, 2, \ldots$ and $\pi_0 = (\pi_{0,0})$.

$Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & A_1 & A_0 & 0 & 0 & \cdots
\end{bmatrix}$
\textbf{M/H}_2/1 \textbf{ Queue}

Let \( \pi_i = (\pi_{i,1}, \pi_{i,2}) \), \( i = 1, 2, \ldots \) and \( \pi_0 = (\pi_{0,0}) \).

\[
Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & A_1 & A_0 & 0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & 0 & \cdots \\
\end{bmatrix}
\]
\textbf{M/H}_2/1 \textbf{ Queue}

Let $\pi_i = (\pi_{i,1}, \pi_{i,2})$, $i = 1, 2, \ldots$ and $\pi_0 = (\pi_{0,0})$.

$$Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & A_1 & A_0 & 0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & 0 & \cdots \\
0 & 0 & A_2 & A_1 & A_0 & \cdots
\end{bmatrix}$$
**M/H$_2$/1 Queue**

Let $\pi_i = (\pi_{i,1}, \pi_{i,2})$, $i = 1, 2, \ldots$ and $\pi_0 = (\pi_{0,0})$.

$$Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & 0 & \cdots \\
B_{1,0} & A_1 & A_0 & 0 & 0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & 0 & \cdots \\
0 & 0 & A_2 & A_1 & A_0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots
\end{bmatrix}$$
$\textbf{M/H}_2/1 \textbf{ Queue}$

Let $\pi_i = (\pi_{i,1}, \pi_{i,2})$, $i = 1, 2, \ldots$ and $\pi_0 = (\pi_{0,0})$.

\[
Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & A_1 & A_0 & 0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & 0 & \cdots \\
0 & 0 & A_2 & A_1 & A_0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]
M/H₂/1 Queue

Let \( \pi_i = (\pi_{i,1}, \pi_{i,2}) \), \( i = 1, 2, \ldots \) and \( \pi_0 = (\pi_{0,0}) \).

\[
Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & A_1 & A_0 & 0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & 0 & \cdots \\
0 & 0 & A_2 & A_1 & A_0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

This is equivalent to

\[
\pi_{j-1}A_0 + \pi_jA_1 + \pi_{j+1}A_2 = 0, \quad j = 2, 3, \ldots
\]
Let us conjecture the existence of matrix $R$ such that
Let us conjecture the existence of matrix $R$ such that

$$\pi_j = \pi_1 R^{j-1}, \quad j = 1, \ldots$$
Let us conjecture the existence of matrix $R$ such that

$$\pi_j = \pi_1 R^{j-1}, \quad j = 1, \ldots$$

If true, then
Let us conjecture the existence of matrix $R$ such that

$$\pi_j = \pi_1 R^{j-1}, \quad j = 1, \ldots$$

If true, then

$$\pi_1 R^{j-2} A_0 + \pi_1 R^{j-1} A_1 + \pi_1 R^j A_2 = 0, \quad j = 2, \ldots$$
Let us conjecture the existence of matrix $R$ such that

$$\pi_j = \pi_1 R^{j-1}, \quad j = 1, \ldots$$

If true, then

$$\pi_1 R^{j-2} A_0 + \pi_1 R^{j-1} A_1 + \pi_1 R^j A_2 = 0, \quad j = 2, \ldots$$

or

$$A_0 + RA_1 + R^2 A_2 = 0$$
Let us conjecture the existence of matrix $R$ such that

$$\pi_j = \pi_1 R^{j-1}, \quad j = 1, \ldots$$

If true, then

$$\pi_1 R^{j-2} A_0 + \pi_1 R^{j-1} A_1 + \pi_1 R^j A_2 = 0, \quad j = 2, \ldots$$

or

$$A_0 + RA_1 + R^2 A_2 = 0$$
M/H₂/1 Queue

Two roots: matrix of all ones and a second matrix R.
**$M/H_2/1$ Queue**

Two roots: matrix of all ones and a second matrix $R$. If the system is ergodic, this second matrix is the correct solution and has *spectral radius* less than one (similar to $\rho < 1$ in $M/M/1$ queue) i.e., all eigenvalues are less than one.
M/H₂/1 Queue

Two roots: matrix of all ones and a second matrix $R$. If system is ergodic, this second matrix is correct solution and has *spectral radius* less than one (similar to $\rho < 1$ in $M/M/1$ queue) i.e., all eigenvalues are less than one.

We will call resulting $R$ the *rate matrix*. 
Two roots: matrix of all ones and a second matrix $R$. If system is ergodic, this second matrix is correct solution and has spectral radius less than one (similar to $\rho < 1$ in $M/M/1$ queue) i.e., all eigenvalues are less than one.

We will call resulting $R$ the rate matrix. $\pi_0$ and $\pi_1$ are solutions to
**M/H₂/1 Queue**

Two roots: matrix of all ones and a second matrix \( \mathbf{R} \). If system is ergodic, this second matrix is correct solution and has *spectral radius* less than one (similar to \( \rho < 1 \) in \( M/M/1 \) queue) i.e., all eigenvalues are less than one.

We will call resulting \( \mathbf{R} \) the *rate matrix*. \( \pi_0 \) and \( \pi_1 \) are solutions to

\[
\pi_0 B_{0,0} + \pi_1 B_{1,0} = 0
\]
M/H_2/1 Queue

Two roots: matrix of all ones and a second matrix R. If system is ergodic, this second matrix is correct solution and has spectral radius less than one (similar to \( \rho < 1 \) in M/M/1 queue) i.e., all eigenvalues are less than one.

We will call resulting R the rate matrix. \( \pi_0 \) and \( \pi_1 \) are solutions to

\[
\begin{align*}
\pi_0 B_{0,0} + \pi_1 B_{1,0} &= 0 \\
\pi_0 B_{0,1} + \pi_1 A_1 + \pi_2 A_2 &= 0
\end{align*}
\]
or

\[(\pi_0, \pi_1) \begin{bmatrix} B_{0,0} & B_{0,1} \\ B_{1,0} & A_1 + RA_2 \end{bmatrix} = [0 \ 0]\]

Also need

\[1 = \pi_0 e + \pi_1 \sum_{j=1}^{\infty} R^{j-1} e = \pi_0 e + \pi_1 (I - R)^{-1} e\]
or

\[ (\pi_0, \pi_1) \begin{bmatrix} B_{0,0} & B_{0,1} \\ B_{1,0} & A_1 + RA_2 \end{bmatrix} = [0 \ 0] \]

Also need

\[ 1 = \pi_0 e + \pi_1 \sum_{j=1}^{\infty} R^{j-1} e = \pi_0 e + \pi_1 (I - R)^{-1} e \]

e column vector (of appropriate size) of ones.
M/H₂/1 Queue

Q: how to incorporate normalization condition into matrix equations?
**M/H₂/1 Queue**

**Q:** how to incorporate normalization condition into matrix equations?

\[
\begin{pmatrix}
\pi_0, \pi_1 \\
(I - R)^{-1}e, A_1 + RA_2
\end{pmatrix}
\begin{bmatrix}
1 & B_{0,1} \\
0 & 1
\end{bmatrix}
= \begin{bmatrix} 1 & 0 \end{bmatrix}
\]
**M/H_2/1 Queue**

**Q:** how to incorporate normalization condition into matrix equations?

\[(\pi_0, \pi_1) \begin{bmatrix} 1 & B_{0,1} \\ (I - R)^{-1}e & A_1 + RA_2 \end{bmatrix} = [1 \ 0]\]

**Q:** what is \(E[N_q]\)?
\textbf{M/H}_2/1 \textbf{ Queue}

**Q:** how to incorporate normalization condition into matrix equations?

\[
(\pi_0, \pi_1) \begin{bmatrix}
1 \\
(\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} \\
\end{bmatrix}
\begin{bmatrix}
\mathbf{B}_{0,1} \\
\mathbf{A}_1 + \mathbf{RA}_2 \\
\end{bmatrix} = [1 \ 0]
\]

**Q:** what is \( \mathbb{E}[N_q] \)?

\[
\mathbb{E}[N_q] =
\]
**M/H₂/1 Queue**

**Q:** how to incorporate normalization condition into matrix equations?

\[
(\pi_0, \pi_1) \begin{bmatrix}
1 & B_{0,1} \\
(I - R)^{-1}e & A_1 + RA_2
\end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]

**Q:** what is \(E[N_q]\)?

\[
E[N_q] = \sum_{j=1}^{\infty} (j - 1)\pi_j e
\]
**M/H_2/1 Queue**

**Q:** how to incorporate normalization condition into matrix equations?

\[
(\pi_0, \pi_1) \begin{bmatrix}
1 & B_{0,1} \\
(I - R)^{-1}e & A_1 + RA_2
\end{bmatrix} = [1 \ 0]
\]

**Q:** what is \(E[N_q]\)?

\[
E[N_q] = \sum_{j=1}^{\infty} (j - 1)\pi_j e = \pi_1 \sum_{j=1}^{\infty} (j - 1)R^{j-1}e
\]
\[ = \pi_1 R (I - R)^{-2} e \]
M/H₂/1 Queue

Q: how to solve for R?
M/H₂/1 Queue

Q: how to solve for R?

• use iterative procedure,
M/H₂/1 Queue

Q: how to solve for $R$?

- use iterative procedure,

$$R(0) = 0$$
**M/H₂/1 Queue**

**Q:** how to solve for $R$?

- use iterative procedure,

\[
R(0) = 0 \\
R(n + 1) = -A_0A_1^{-1} - R^2(n)A_2A_1^{-1}
\]
M/H₂/1 Queue

Q: how to solve for \( R \)?

- use iterative procedure,

\[
R(0) = 0 \quad R(n + 1) = -A_0A_1^{-1} - R^2(n)A_2A_1^{-1}
\]

- if system ergodic, guaranteed to converge
**M/H₂/1 Queue**

**Q:** how to solve for R?

- use iterative procedure,

\[
R(0) = 0 \\
R(n + 1) = -A_0 A_1^{-1} - R^2(n) A_2 A_1^{-1}
\]

- if system ergodic, guaranteed to converge
- there are other more efficient techniques
**M/H₂/1 Queue**

**Q:** how to solve for \( R \)?

- use iterative procedure,

\[
R(0) = 0 \\
R(n + 1) = -A_0 A_1^{-1} - R^2(n) A_2 A_1^{-1}
\]

- if system ergodic, guaranteed to converge
- there are other more efficient techniques
$M/H_2/1$ Queue

Q: when is system stable (ergodic)?
**M/H₂/1 Queue**

**Q:** when is system stable (ergodic)?

Calculate *expected drift* of repeating portion
M/H₂/1 Queue

Q: when is system stable (ergodic)?

Calculate expected drift of repeating portion

• assume \( A_i \) are \( m \times m \) matrices
M/H₂/1 Queue

Q: when is system stable (ergodic)?

Calculate expected drift of repeating portion

- assume $A_i$ are $m \times m$ matrices
- define $A = A_0 + A_1 + A_2$
**M/H₂/1 Queue**

**Q:** when is system stable (ergodic)?

Calculate *expected drift* of repeating portion

- assume $A_i$ are $m \times m$ matrices

- define $\mathbf{A} = \mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2$ - can be interpreted as an infinitesimal generator for an MC that describes behavior of states within a level far, far to the right.
M/H₂/1 Queue

Q: when is system stable (ergodic)?

Calculate expected drift of repeating portion

- assume \( A_i \) are \( m \times m \) matrices

- define \( A = A_0 + A_1 + A_2 \) - can be interpreted as an infinitesimal generator for an MC that describes behavior of states within a level far, far to the right. Let \( f = (f_1, \ldots, f_m) \) be solution of

\[
fA = 0
\]
• stability condition is
• stability condition is

\[ \text{drift to right} < \text{drift to left} \]

\[ fA_0 < fA_2 \]
General Case

- this is example of a quasi birth death (QBD) process
General Case

- this is example of a quasi birth death (QBD) process
- general matrix geometric solution
### General Case

- This is an example of a *quasi birth death (QBD)* process.
- General matrix geometric solution

\[
Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & 0 & \cdots \\
\end{bmatrix}
\]
General Case

- this is example of a \textit{quasi birth death (QBD)} process

- general matrix geometric solution

\[ Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & 0 & \cdots \\
\end{bmatrix} \]
General Case

- this is example of a *quasi birth death (QBD)* process

- general matrix geometric solution

\[ Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & 0 & \cdots \\
B_{2,0} & B_{2,1} & A_1 & A_0 & 0 & \cdots \\
\end{bmatrix} \]
General Case

- this is example of a \textit{quasi birth death (QBD)} process
- general matrix geometric solution

\[
Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & 0 & \cdots \\
B_{2,0} & B_{2,1} & A_1 & A_0 & 0 & \cdots \\
B_{3,0} & B_{3,1} & A_2 & A_1 & A_0 & \cdots \\
\end{bmatrix}
\]
General Case

- this is example of a quasi birth death (QBD) process

- general matrix geometric solution

\[ Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & 0 & \cdots \\
B_{2,0} & B_{2,1} & A_1 & A_0 & 0 & \cdots \\
B_{3,0} & B_{3,1} & A_2 & A_1 & A_0 & \cdots \\
B_{4,0} & B_{4,1} & A_3 & A_2 & A_1 & \cdots 
\end{bmatrix} \]
General Case

- this is example of a \textit{quasi birth death (QBD)} process

- general matrix geometric solution

\[ Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & 0 & \cdots \\
B_{2,0} & B_{2,1} & A_1 & A_0 & 0 & \cdots \\
B_{3,0} & B_{3,1} & A_2 & A_1 & A_0 & \cdots \\
B_{4,0} & B_{4,1} & A_3 & A_2 & A_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \]
General Case

- this is example of a quasi birth death (QBD) process
- general matrix geometric solution

\[
Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & 0 & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & 0 & \cdots \\
B_{2,0} & B_{2,1} & A_1 & A_0 & 0 & \cdots \\
B_{3,0} & B_{3,1} & A_2 & A_1 & A_0 & \cdots \\
B_{4,0} & B_{4,1} & A_3 & A_2 & A_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]
$B_{0,0}$ is $m' \times m'$. 
$B_{0,0}$ is $m' \times m'$, $B_{0,1}$ is $m' \times m$. 
$B_{0,0}$ is $m' \times m'$, $B_{0,1}$ is $m' \times m$, $B_{i,0}$ is $m \times m'$,
$B_{0,0}$ is $m' \times m'$, $B_{0,1}$ is $m' \times m$, $B_{i,0}$ is $m \times m'$, and all others are $m \times m$ matrices.
$\mathbf{B}_{0,0}$ is $m' \times m'$, $\mathbf{B}_{0,1}$ is $m' \times m$, $\mathbf{B}_{i,0}$ is $m \times m'$, and all others are $m \times m$ matrices.

- states divide into levels, $m'$ in level 0, $m$ in levels $i = 1, 2, \ldots$.
\(B_{0,0}\) is \(m' \times m'\), \(B_{0,1}\) is \(m' \times m\), \(B_{i,0}\) is \(m \times m'\), and all others are \(m \times m\) matrices

- States divide into levels, \(m'\) in level 0, \(m\) in levels \(i = 1, 2, \ldots\). Call states in level \(i\), \((i, j)\)

- \(\pi = (\pi_0, \pi_1, \ldots)\), where \(\pi_0 = (\pi_{0,1}, \ldots \pi_{0,m'})\),
$B_{0,0}$ is $m' \times m'$, $B_{0,1}$ is $m' \times m$, $B_{i,0}$ is $m \times m'$, and all others are $m \times m$ matrices

- states divide into levels, $m'$ in level 0, $m$ in levels $i = 1, 2, \ldots$ call states in level $i$, $(i, j)$

- $\pi = (\pi_0, \pi_1, \ldots)$, where $\pi_0 = (\pi_{0,1}, \ldots, \pi_{0,m'})$, $\pi_j = (\pi_{i,1}, \ldots, \pi_{i,m})$, $j = 1, \ldots$ and

$$\pi Q = 0$$
General Case

- balance equation for repeating portion
General Case

- balance equation for repeating portion

\[ \sum_{k=0}^{\infty} \pi_{j-1+k} A_k = 0, \quad j = 2, 3, \ldots \]
General Case

- balance equation for repeating portion

\[ \sum_{k=0}^{\infty} \pi_{j-1+k} A_k = 0, \quad j = 2, 3, \ldots \]

with solution
General Case

- balance equation for repeating portion

\[
\sum_{k=0}^{\infty} \pi_{j-1+k} A_k = 0, \quad j = 2, 3, \ldots
\]

with solution

\[
\pi_j = \pi_1 R^{j-1}, \quad j = 2, 3, \ldots
\]
General Case

- balance equation for repeating portion

\[ \sum_{k=0}^{\infty} \pi_{j-1+k} A_k = 0, \quad j = 2, 3, \ldots \]

with solution

\[ \pi_j = \pi_1 R^{j-1}, \quad j = 2, 3, \ldots \]
R is solution of
• R is solution of

\[ \sum_{k=0}^{\infty} R^k A_k = 0 \]
• $R$ is solution of

$$\sum_{k=0}^{\infty} R^k A_k = 0$$

• solution to boundary states
• R is solution of

\[ \sum_{k=0}^{\infty} R^k A_k = 0 \]

solution to boundary states

\[ (\pi_0, \pi_1) \begin{bmatrix} B_{0,0} & B_{0,1} \\ \sum_{k=1}^{\infty} R^{k-1} B_{k,0} & \sum_{k=1}^{\infty} R^{k-1} B_{k,1} \end{bmatrix} = 0 \]
General Case

- normalization condition yields
General Case

- normalization condition yields

\[(\pi_0, \pi_1) \times \begin{bmatrix} 1 & B^*_0,0 & B_{0,1} \end{bmatrix} \]
General Case

- normalization condition yields

\[ (\pi_0, \pi_1) \times \begin{bmatrix} 1 & B_{0,0}^* \sum_{k=1}^{\infty} R^{k-1} B_{0,0}^* \sum_{k=1}^{\infty} R^{k-1} B_{0,1} \end{bmatrix} \]

\[ \begin{bmatrix} (I - R)^{-1} e \sum_{k=1}^{\infty} R^{k-1} B_{k,0} \end{bmatrix}^* \begin{bmatrix} 1 & 0 \end{bmatrix} \]
General Case

- normalization condition yields

\[
(\pi_0, \pi_1) \times \\
\begin{bmatrix}
1 \\
(I - R)^{-1}e \\
\end{bmatrix} \left[
\begin{array}{c}
\sum_{k=1}^{\infty} R^{k-1} B_{k,0} \\
\sum_{k=1}^{\infty} R^{k-1} B_{k,1}
\end{array}
\right]^{*}
\]

\[= \begin{bmatrix} 1 & 0 \end{bmatrix} \]

where \(A^*\) is matrix \(A\) with leftmost column removed.
General Case

- normalization condition yields

\[(\pi_0, \pi_1) \times \left[ \begin{array}{c} 1 \\ (I - R)^{-1}e \end{array} \right] \left[ \sum_{k=1}^{\infty} R^{k-1}B_{0,k} \right]^* \sum_{k=1}^{\infty} R^{k-1}B_{1,k} \]

= \[1 \ 0\]

where \(A^*\) is matrix \(A\) with leftmost column removed.
• calculation of $R$
calculation of $R$

$R(0) = 0$
• calculation of $R$

\[
\begin{align*}
R(0) & = 0 \\
R(n + 1) & = - \sum_{k \neq 1} R^k(n)A_kA_1^{-1}
\end{align*}
\]
• calculation of $R$

$$R(0) = 0$$

$$R(n + 1) = - \sum_{k \neq 1} R^k(n) A_k A_1^{-1}$$

• when is system stable? look at expected drift of process in repeating portion
• calculation of $R$

$$R(0) = 0$$

$$R(n + 1) = -\sum_{k \neq 1} R^k(n) A_k A_1^{-1}$$

• when is system stable? look at expected drift of process in repeating portion

★ define $A = \sum_{k=0}^{\infty} A_k$ and $f$ such that $fA = 0$
• calculation of $R$

$$
R(0) = 0 \\
R(n + 1) = -\sum_{k\neq 1} R^k(n)A_kA_1^{-1}
$$

• when is system stable? look at expected drift of process in repeating portion

★ define $A = \sum_{k=0}^{\infty} A_k$ and $f$ such that $fA = 0$
★ compute drift to right and to left; condition of stability
is
is

drift to right < drift to left
is

\text{drift to right} \ < \ \text{drift to left}

fA_0 \ < \ f \sum_{k=2}^{\infty} (k - 1)A_k
Application to Multiprocessor with Failures

- 2 servers serving infinite capacity queue
Application to Multiprocessor with Failures

- 2 servers serving infinite capacity queue
- Poisson arrivals, $\lambda$, exponential service times, $\mu$
Application to Multiprocessor with Failures

- 2 servers serving infinite capacity queue
- Poisson arrivals, $\lambda$, exponential service times, $\mu$
- time between failures for single processor exponentially distr. with mean $1/\alpha$
Application to Multiprocessor with Failures

- 2 servers serving infinite capacity queue
- Poisson arrivals, $\lambda$, exponential service times, $\mu$
- time between failures for single processor exponentially distr. with mean $1/\alpha$
- single repairman, repair time exponentially distr. with mean $1/\gamma$
Application to Multiprocessor with Failures

- 2 servers serving infinite capacity queue
- Poisson arrivals, $\lambda$, exponential service times, $\mu$
- time between failures for single processor exponentially distr. with mean $1/\alpha$
- single repairman, repair time exponentially distr. with mean $1/\gamma$
- state of MC - $(n, u)$, $n$ - no. of jobs in system; $u$ - no. of processors operational
Application to Multiprocessor with Failures

- 2 servers serving infinite capacity queue
- Poisson arrivals, $\lambda$, exponential service times, $\mu$
- time between failures for single processor exponentially distr. with mean $1/\alpha$
- single repairman, repair time exponentially distr. with mean $1/\gamma$
- state of MC - $(n, u)$, $n$ - no. of jobs in system; $u$ - no. of processors operational
- let $\pi = (\pi_0, \pi_1, \ldots)$; $\pi_i = (\pi_{i,0}, \pi_{i,1}, \pi_{i,2})$, $i = 1, 2, \ldots$
Application to Multiprocessor with Failures

Infinitesimal generator

\[ Q = \begin{bmatrix}
  B_{0,0} & A_0 & 0 & \cdots \\
  B_{1,0} & B_{1,1} & A_0 & 0 & \cdots \\
  0 & A_2 & A_1 & A_0 & \cdots \\
  0 & 0 & A_2 & A_1 & \cdots \\
  \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \]
Application to Multiprocessor with Failures

\[
B_{0,0} = \begin{bmatrix}
-(\lambda + \gamma) & \gamma & 0 \\
\alpha & -(\lambda + \gamma + \alpha) & \gamma \\
0 & 2\alpha & -(\lambda + 2\alpha)
\end{bmatrix}
\]
Application to Multiprocessor with Failures

\[
B_{0,0} = \begin{bmatrix}
-(\lambda + \gamma) & \gamma & 0 \\
\alpha & -(\lambda + \gamma + \alpha) & \gamma \\
0 & 2\alpha & -(\lambda + 2\alpha)
\end{bmatrix}
\]

\[
B_{1,1} = \begin{bmatrix}
-(\lambda + \gamma) & \gamma & 0
\end{bmatrix}
\]
Application to Multiprocessor with Failures

\[
B_{0,0} = \begin{bmatrix}
-(\lambda + \gamma) & \gamma & 0 \\
\alpha & -(\lambda + \gamma + \alpha) & \gamma \\
0 & 2\alpha & -(\lambda + 2\alpha)
\end{bmatrix}
\]

\[
B_{1,1} = \begin{bmatrix}
-(\lambda + \gamma) & \gamma & 0 \\
\alpha & -(\lambda + \gamma + \alpha + \mu) & \gamma \\
0 & 2\alpha & -(\lambda + 2\alpha + \mu)
\end{bmatrix}
\]
$A_1 = \begin{bmatrix}
-(\lambda + \gamma) & \gamma & 0 \\
\alpha & -(\lambda + \gamma + \alpha + \mu) & \gamma \\
0 & 2\alpha & -(\lambda + 2\alpha + 2\mu)
\end{bmatrix}$
\[ A_1 = \begin{bmatrix}
  -(\lambda + \gamma) & \gamma & 0 \\
  \alpha & -(\lambda + \gamma + \alpha + \mu) & \gamma \\
  0 & 2\alpha & -(\lambda + 2\alpha + 2\mu)
\end{bmatrix} \]

\[ A_0 = \lambda I \quad A_2 = \text{diag}(0, \mu, 2\mu), \quad B_{1,0} = \text{diag}(0, \mu, \mu) \]

Can solve for \( \pi \) using matrix geometric technique
Application to Multiprocessor with Failures

Let \( I \) denote the number of processors that are operational, \( I = 0, 1, 2 \).
Application to Multiprocessor with Failures

Let $I$ denote the number of processors that are operational, $I = 0, 1, 2$.

**Q:** what is $P(I = 0)$
Application to Multiprocessor with Failures

Let \( I \) denote the number of processors that are operational, \( I = 0, 1, 2 \).

Q: what is \( P(I = 0) \)

\[
P(I = 0) = \sum_{j=0}^{\infty} \pi_j \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
\]
Let $I$ denote the number of processors that are operational, $I = 0, 1, 2$.

**Q:** what is $P(I = 0)$

\[
P(I = 0) = \sum_{j=0}^{\infty} \pi_j \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
\]
Q: what are conditions for stability?
Q: what are conditions for stability?

\[ A = A_0 + A_1 + A_2 = \begin{bmatrix} -\gamma & \gamma & 0 \\ \alpha & -(\gamma + \alpha) & \gamma \\ 0 & 2\alpha & -2\alpha \end{bmatrix} \]
Q: what are conditions for stability?

\[
A = A_0 + A_1 + A_2 = \begin{bmatrix}
-\gamma & \gamma & 0 \\
\alpha & -(\gamma + \alpha) & \gamma \\
0 & 2\alpha & -2\alpha 
\end{bmatrix}
\]

Let \( f = (f_1, f_2, f_3) \) be solution to
Q: what are conditions for stability?

\[ A = A_0 + A_1 + A_2 = \begin{bmatrix}
-\gamma & \gamma & 0 \\
\alpha & -(\gamma + \alpha) & \gamma \\
0 & 2\alpha & -2\alpha \\
\end{bmatrix} \]

Let \( f = (f_1, f_2, f_3) \) be solution to

\[ fA = 0 \quad \text{and} \quad fe = 1 \]
where

e = \begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix}
Stability Condition

Stability condition is:
Stability Condition

Stability condition is:

\[ f_{A_0} < f_{A_2} \]
Stability Condition

Stability condition is:

\[ fA_0 < fA_2 \]

or
Stability Condition

Stability condition is:

\[ fA_0 < fA_2 \]

or

\[ \lambda < \frac{\mu \gamma (\gamma + 2\alpha)}{2\alpha^2 + 2\gamma \alpha + \gamma^2} \]
Stability Condition

Stability condition is:

\[ fA_0 < fA_2 \]

or

\[ \lambda < \frac{\mu \gamma (\gamma + 2\alpha)}{2\alpha^2 + 2\gamma \alpha + \gamma^2} \]
Phase Type Distributions

Let $S$ be an rv with a phase-type distr.
Phase Type Distributions

Let $S$ be an rv with a phase-type distr. Determined by behavior of a $K$ state continuous time MC; $K - 1$ states ($k = 1, \ldots K - 1$) are transient;
Phase Type Distributions

Let $S$ be an rv with a phase-type distr. Determined by behavior of a $K$ state continuous time MC; $K - 1$ states ($k = 1, \ldots, K - 1$) are transient; one state ($K$) is an absorbing state.
Phase Type Distributions

Let $S$ be an rv with a phase-type distr. Determined by behavior of a $K$ state continuous time MC; $K - 1$ states ($k = 1, \ldots, K - 1$) are transient; one state ($K$) is an absorbing state. Initial distribution $\pi_k(0) = P(X(0) = k)$, $k = 1, \ldots, K$. By absorbing,

- $\lambda_{K,j} = 0$, $j \neq K$
Phase Type Distributions

Let $S$ be an rv with a phase-type distr. Determined by behavior of a $K$ state continuous time MC; $K - 1$ states ($k = 1, \ldots K - 1$) are transient; one state ($K$) is an absorbing state. Initial distribution $\pi_k(0) = P(X(0) = k), k = 1, \ldots, K$. By absorbing,

- $\lambda_{K,j} = 0, j \neq K$
- $\pi_K(t) \rightarrow 1$ as $t \rightarrow \infty$

$S$ is defined to be the time needed to reach $K$. 
Phase Type Distributions

Examples:
Phase Type Distributions

Examples:

- Erlang of order $r$: $K = r + 1$, $\lambda_{i,i+1} = r\mu$, $i = 1, \ldots, r$, all other rates are zero. $\pi_1(0) = 1$, $\pi_i(0) = 0$, $i \neq 1$
Phase Type Distributions

Examples:

- Erlang of order $r$: $K = r + 1$, $\lambda_{i,i+1} = r\mu$, $i = 1, \ldots, r$, all other rates are zero. $\pi_1(0) = 1$, $\pi_i(0) = 0$, $i \neq 1$

- $H_2$ distr.: $K = 3$, $\lambda_{i,3} = \mu_i$, $i = 1, 2$, all other rates are zero.
Phase Type Distributions

Examples:

- Erlang of order $r$: $K = r + 1$, $\lambda_{i,i+1} = r\mu$, $i = 1, \ldots, r$, all other rates are zero. $\pi_1(0) = 1$, $\pi_i(0) = 0$, $i \neq 1$

- $H_2$ distr.: $K = 3$, $\lambda_{i,3} = \mu_i$, $i = 1, 2$, all other rates are zero. $\pi_i(0) = \alpha_i$, $i = 1, 2$, $\pi_3(0) = 0$
\textbf{M/PH/1 Queue}

- Poisson arrivals - $\lambda$
M/PH/1 Queue

- Poisson arrivals - $\lambda$

- Phase-type distr. - $K + 1$ states, transition rates $\lambda_{ij}$, and initial distr. $\alpha_i$
**M/PH/1 Queue**

- Poisson arrivals - $\lambda$

- Phase-type distr. - $K + 1$ states, transition rates $\lambda_{ij}$, and initial distr. $\alpha_i$

- System state - $(n, s)$ $n$ - no. of jobs in system, $s$ state of MC associated with PH distr.
M/PH/1 Queue

- Poisson arrivals - $\lambda$

- Phase-type distr. - $K + 1$ states, transition rates $\lambda_{ij}$, and initial distr. $\alpha_i$

- system state - $(n, s)$ $n$ - no. of jobs in system, $s$ state of MC associated with PH distr.
\[ Q = \begin{bmatrix}
B_{0,0} & B_{0,1} & 0 & \cdots \\
B_{1,0} & B_{1,1} & A_0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & \cdots \\
0 & 0 & A_2 & A_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots 
\end{bmatrix} \]
\textbf{$\text{M/PH/1}$ Queue}

\[ B_{0,0} = \left[ -\lambda (1 - \alpha_{K+1}) \right] \]
$M/PH/1$ Queue

$$B_{0,0} = [-\lambda(1 - \alpha_{K+1})]$$

$$B_{0,1} = \lambda[\alpha_1 \ldots \alpha_K]$$
**M/PH/1 Queue**

\[ B_{0,0} = [-\lambda(1 - \alpha_{K+1})] \]

\[ B_{0,1} = \lambda[\alpha_1 \ldots \alpha_K] \]

\[ B_{1,0} = [\lambda_{1,K+1} \ldots \lambda_{K,K+1}]^T \]
\section*{M/PH/1 Queue}

\[ B_{0,0} = [-\lambda (1 - \alpha_{K+1})] \]

\[ B_{0,1} = \lambda [\alpha_1 \ldots \alpha_K] \]

\[ B_{1,0} = [\lambda_{1,K+1} \ldots \lambda_{K,K+1}]^T \]

\[ A_0 = \text{diag}(\lambda, \ldots, \lambda) \]
$$A_1 = \begin{bmatrix} -a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\ \lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K \end{bmatrix}$$
\[ A_1 = \begin{bmatrix} -a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\ \lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K \end{bmatrix} \]

where \( a_i = \lambda + \sum_{k \neq i} \lambda_{i,k} \).
\[
A_1 = \begin{bmatrix}
-a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\
\lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K
\end{bmatrix}
\]

where \( a_i = \lambda + \sum_{k \neq i} \lambda_{i,k} \).

\[
A_2 = \begin{bmatrix}
\lambda_{1,K+1} \alpha_1 & \lambda_{1,K+1} \alpha_2 & \cdots & \lambda_{1,K+1} \alpha_K \\
\lambda_{2,K+1} \alpha_1 & \lambda_{2,K+1} \alpha_2 & \cdots & \lambda_{2,K+1} \alpha_K \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{K,K+1} \alpha_1 & \lambda_{K,K+1} \alpha_2 & \cdots & \lambda_{K,K+1} \alpha_K
\end{bmatrix}
\]
\[ A_1 = \begin{bmatrix} -a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\ \lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K \end{bmatrix} \]

where \( a_i = \lambda + \sum_{k \neq i} \lambda_{i,k} \).

\[ A_2 = \begin{bmatrix} \lambda_{1,K+1} \alpha_1 & \lambda_{1,K+1} \alpha_2 & \cdots & \lambda_{1,K+1} \alpha_K \\ \lambda_{2,K+1} \alpha_1 & \lambda_{2,K+1} \alpha_2 & \cdots & \lambda_{2,K+1} \alpha_K \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,K+1} \alpha_1 & \lambda_{K,K+1} \alpha_2 & \cdots & \lambda_{K,K+1} \alpha_K \end{bmatrix} \]

Solution proceeds as before.
Markov Modulated Poisson Process (MMPP)

K state continuous time MC \( \{X(t)\} \) with transition rates \( \{\lambda_{i,j}\} \).
Markov Modulated Poisson Process (MMPP)

\( K \) state continuous time MC \( \{ X(t) \} \) with transition rates \( \{ \lambda_{i,j} \} \). \( K \) different arrival rates \( \{ \lambda_k \} \).
Markov Modulated Poisson Process (MMPP)

$K$ state continuous time MC $\{X(t)\}$ with transition rates $\{\lambda_{i,j}\}$. $K$ different arrival rates $\{\lambda_k\}$

Poisson arrival process with rate $\lambda_{X(t)}$, $t \geq 0$, i.e., $\{X(t)\}$ modulates the arrival rate
Markov Modulated Poisson Process (MMPP)

K state continuous time MC \{X(t)\} with transition rates \{\lambda_{i,j}\}. K different arrival rates \{\lambda_k\}

Poisson arrival process with rate \lambda_{X(t)}, t \geq 0, i.e., \{X(t)\} modulates the arrival rate

**Important** uses in modeling bursty traffic sources in networks.
MMPP/M/1 Queue

- arrivals according to a MMPP, transition rates \( \{\lambda_{i,j}\} \) and arrival rates \( \{\lambda_k\} \)
MMPP/M/1 Queue

- arrivals according to a MMPP, transition rates $\{\lambda_{i,j}\}$ and arrival rates $\{\lambda_k\}$

- exponential service times, $\mu$
**MMPP/M/1 Queue**

- arrivals according to a MMPP, transition rates \(\{\lambda_{i,j}\}\) and arrival rates \(\{\lambda_k\}\)

- exponential service times, \(\mu\)

- system state - \((n, s)\) \(n\) - no. of jobs in system, \(s\) state of MC modulating arrivals
MMPP/M/1 Queue

• arrivals according to a MMPP, transition rates $\{\lambda_{i,j}\}$ and arrival rates $\{\lambda_k\}$

• exponential service times, $\mu$

• system state - $(n, s)$ $n$ - no. of jobs in system, $s$ state of MC modulating arrivals
\[ Q = \begin{bmatrix}
B_{0,0} & A_0 & 0 & \cdots \\
A_2 & A_1 & A_0 & 0 & \cdots \\
0 & A_2 & A_1 & A_0 & \cdots \\
0 & 0 & A_2 & A_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \]
**MMPP/M/1 Queue**

\[
B_{0,0} = \begin{bmatrix}
-a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\
\lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K
\end{bmatrix}
\]
**MMPP/M/1 Queue**

\[ B_{0,0} = \begin{bmatrix} -a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\ \lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K \end{bmatrix} \]

\[ a_i = \lambda_i + \sum_{k \neq i} \lambda_{i,k} \]
**MMPP/M/1 Queue**

\[
B_{0,0} = \begin{bmatrix}
-a_1 & \lambda_{1,2} & \cdots & \lambda_{1,K} \\
\lambda_{2,1} & -a_2 & \cdots & \lambda_{2,K} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{K,1} & \lambda_{K,2} & \cdots & -a_K
\end{bmatrix}
\]

\[
a_i = \lambda_i + \sum_{k \neq i} \lambda_{i,k}
\]
$\Lambda_1 = \begin{bmatrix}
-(a_1 + \mu) & \lambda_{1,2} & \cdots & \lambda_{1,K} \\
\lambda_{2,1} & - (a_2 + \mu) & \cdots & \lambda_{2,K} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{K,1} & \lambda_{K,2} & \cdots & -(a_K + \mu)
\end{bmatrix}$
\[ \Lambda_1 = \begin{bmatrix} -(a_1 + \mu) & \lambda_{1,2} & \cdots & \lambda_{1,K} \\ \lambda_{2,1} & -(a_2 + \mu) & \cdots & \lambda_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,1} & \lambda_{K,2} & \cdots & -(a_K + \mu) \end{bmatrix} \]

\[ \Lambda_2 = \text{diag}(\mu, \ldots, \mu) \]
\[ A_1 = \begin{bmatrix}
-(a_1 + \mu) & \lambda_{1,2} & \cdots & \lambda_{1,K} \\
\lambda_{2,1} & -(a_2 + \mu) & \cdots & \lambda_{2,K} \\
\vdots & \vdots & & \vdots \\
\lambda_{K,1} & \lambda_{K,2} & \cdots & -(a_K + \mu)
\end{bmatrix} \]

\[ A_2 = \text{diag}(\mu, \ldots, \mu) \]

\[ A_0 = \text{diag}(\lambda_1, \ldots, \lambda_K) \]
\[ \Lambda_1 = \begin{bmatrix} - (a_1 + \mu) & \lambda_{1,2} & \cdots & \lambda_{1,K} \\ \lambda_{2,1} & - (a_2 + \mu) & \cdots & \lambda_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{K,1} & \lambda_{K,2} & \cdots & - (a_K + \mu) \end{bmatrix} \]

\[ \Lambda_2 = \text{diag}(\mu, \ldots, \mu) \]

\[ \Lambda_0 = \text{diag}(\lambda_1, \ldots, \lambda_K) \]

Solution proceeds as before.