Chapter 5.1

Discrete-Time Markov Chain

Performance Evaluation of the Internet of Things (IoT)

Module Course: Performance Evaluation of Distributed Systems

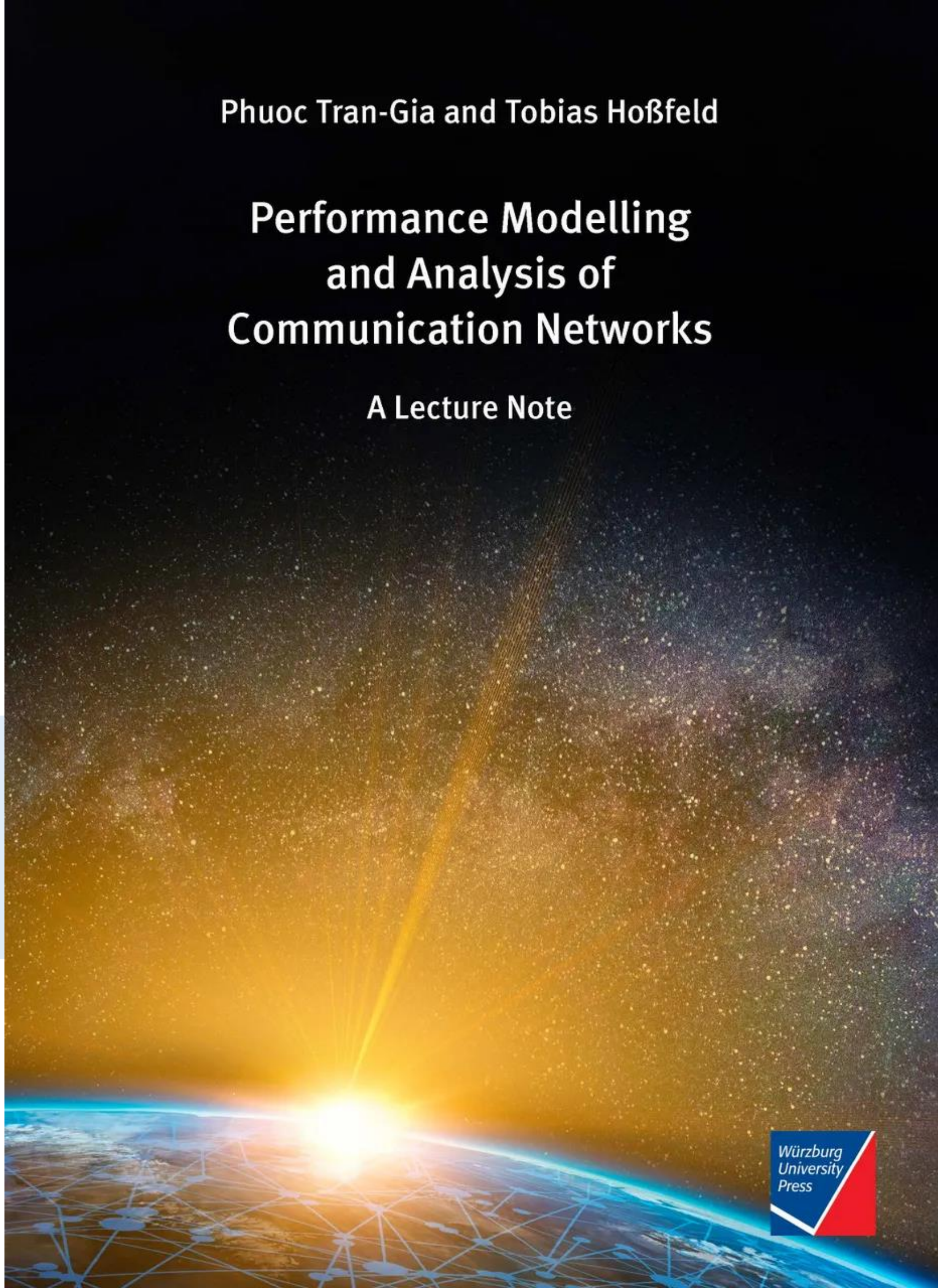
Prof. Tobias Hoßfeld, Summer Semester 2022

Disclaimer and Copyright Notice

The book must be cited and the disclaimer attached when using lectures slides or scripts.

Tran-Gia, P. & Hossfeld, T. (2021).
Performance Modeling and Analysis of Communication Networks - A Lecture Note. Würzburg University Press.
https://doi.org/10.25972/WUP-978-3-95826-153-2

Website to download book, exercises, slides and scripts:
https://modeling.systems/



Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chapter 5

5 Analysis of Non-Markovian Systems

Discrete-Time Markov Chain (DTMC)

- **Discrete-time Markov process**
  - Stochastic process $\{X(t),\ t > 0\}$ has Markov property at discrete (not necessarily equidistant) points in time $\{t_n,\ n = 0,1,\ldots\}$
  - Sequence of r.v.s $\{X(t_0), X(t_1), \ldots\}$: $X(t_{n+1})$ only depends on current $X(t_n)$ (Markov property)

- **Discrete-time Markov chain**
  - finite or countable state space S: discrete state space, e.g., analysis of M/GI/1 or GI/M/1
  - transition probability $p_{ij} \geq 0$ for $i \neq j$ and $i, j \in S$
  - initial state $X(0)$

- Fundamental: **Markov property**
  - $P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n, \ldots, X(t_0) = x_0) = P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n)$
  - **transition probability**: $p_{ij} = P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n)$
  - transition probability matrix (stochastic matrix): $\boldsymbol{P} = \{p_{ij}\}$

Continuous-Time Markov Chain (CTMC)

- **Continuous-time Markov process**
  - Stochastic process $\{X(t), t \geq 0\}$ with Markov property
- **Continuous-time Markov chain (CMTC)** is defined by
  - discrete state space S is finite or countable; e.g. number of customers in system
  - transition rates $q_{ij} \geq 0$ for $i \neq j$ and $i, j \in S$
  - initial state $X(0)$, i.e. probability distribution of initial state

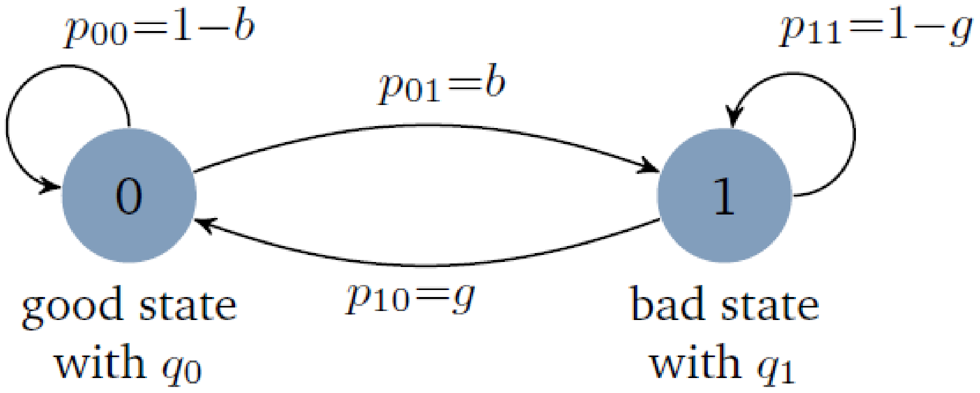
- Probability $x(i,t) = P(X(t) = i)$ that the system is in state $[X = i]$ at time t
- State vector $\boldsymbol{X}(t) = (x(t,0), x(t,1), \dots)$

- Definition of **rate matrix $\boldsymbol{Q}$** with $q_{ii} = -\sum_{i \neq j} q_{ij}$
  - allows compact notation (Kolmogorov equations.)
  - row-wise sums of $\boldsymbol{Q}$ are 0

Example: Bursty Channel (Gilbert-Elliot Model)

- Gilbert-Elliot model has two different states
  - good state [X = 0] with low packet loss probability q_0
  - bad state [X = 1] with a high packet loss probability q_1
- DTMC
  - probability b: change from good to bad
  - probability g: change from bad to good

- Transition diagram with transition probabilities

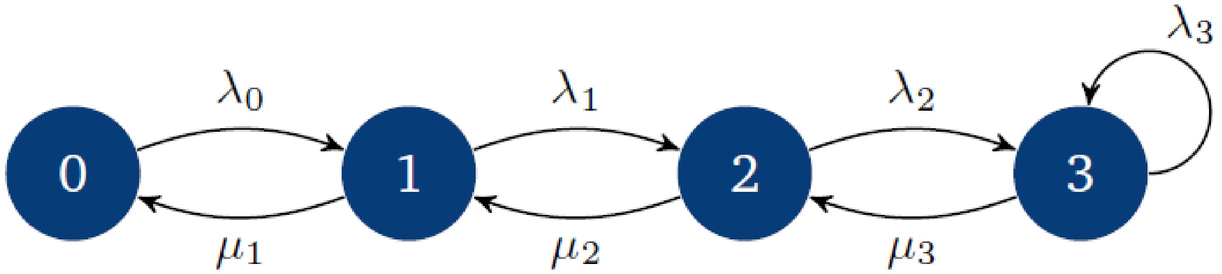


- Transition matrix

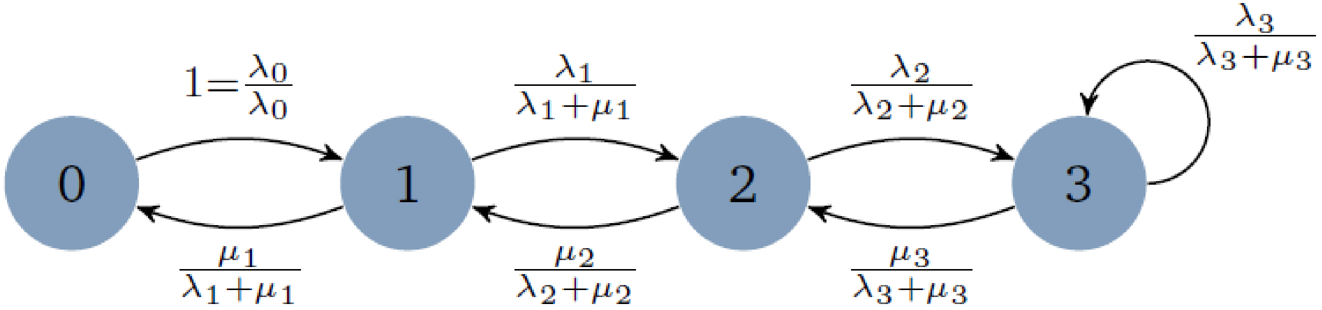
$$\mathcal{P} = \begin{pmatrix} 1-b & b \\ g & 1-g \end{pmatrix}$$

DTMC of a CTMC

$$p_{ij} = \frac{q_{ij}}{q_i} = \frac{q_{ij}}{\sum_{i \neq j} q_{ij}}$$



(a) Continuous-time Markov chain (CTMC) with transition rates.



(b) Embedded discrete-time Markov chain (DTMC) with transition probabilities.

Chapter 5.2

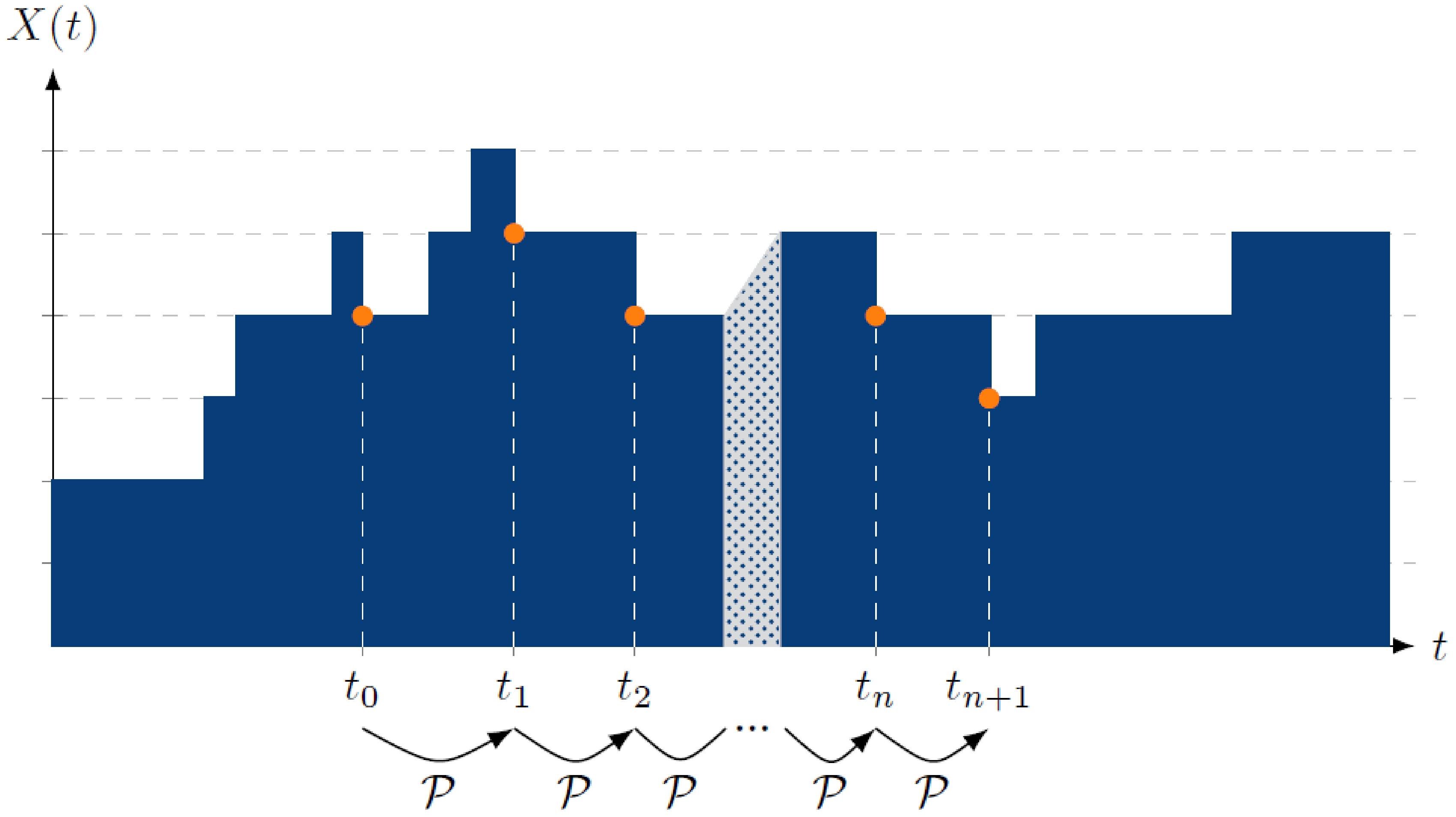
Method of Embedded Markov Chain

Performance Evaluation of the Internet of Things (IoT)

Module Course: Performance Evaluation of Distributed Systems

Prof. Tobias Hoßfeld, Summer Semester 2022

Regeneration Points of the Embedded Markov Chain



Method of Embedded Markov Chain

- **State probabilities** at embedding times

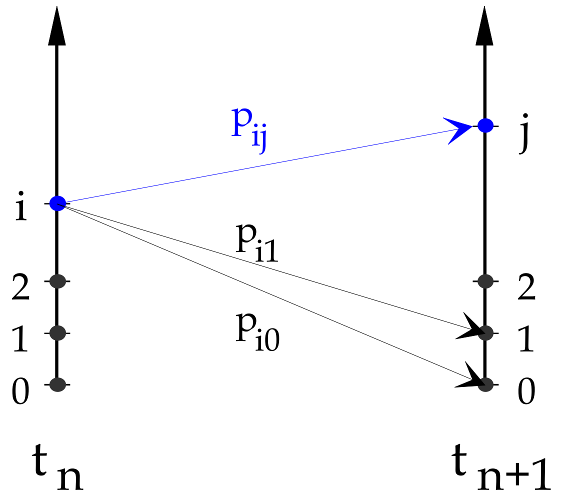
$$\mathcal{X}_n = \{x(i,n),\ i=0,1,\dots\}$$

$$x(i,n) = P(X(t_n)=i)$$

- **Transition probability matrix** $\mathcal{P}$

$$\mathcal{P} = \{p_{ij}\}$$

$$p_{ij} = P(X(t_{n+1})=j \mid X(t_n)=i), \qquad i,j=0,1,\dots$$



$$\sum_j p_{ij} = 1$$

Method of Embedded Markov Chain: Matrix Form

- At each embedding time t_n, state probability vector

$$\mathbf{X}_n = \left(x(0,n), x(1,n), \dots\right) ,$$
$$x(i,n) = P(X(t_n) = i) .$$

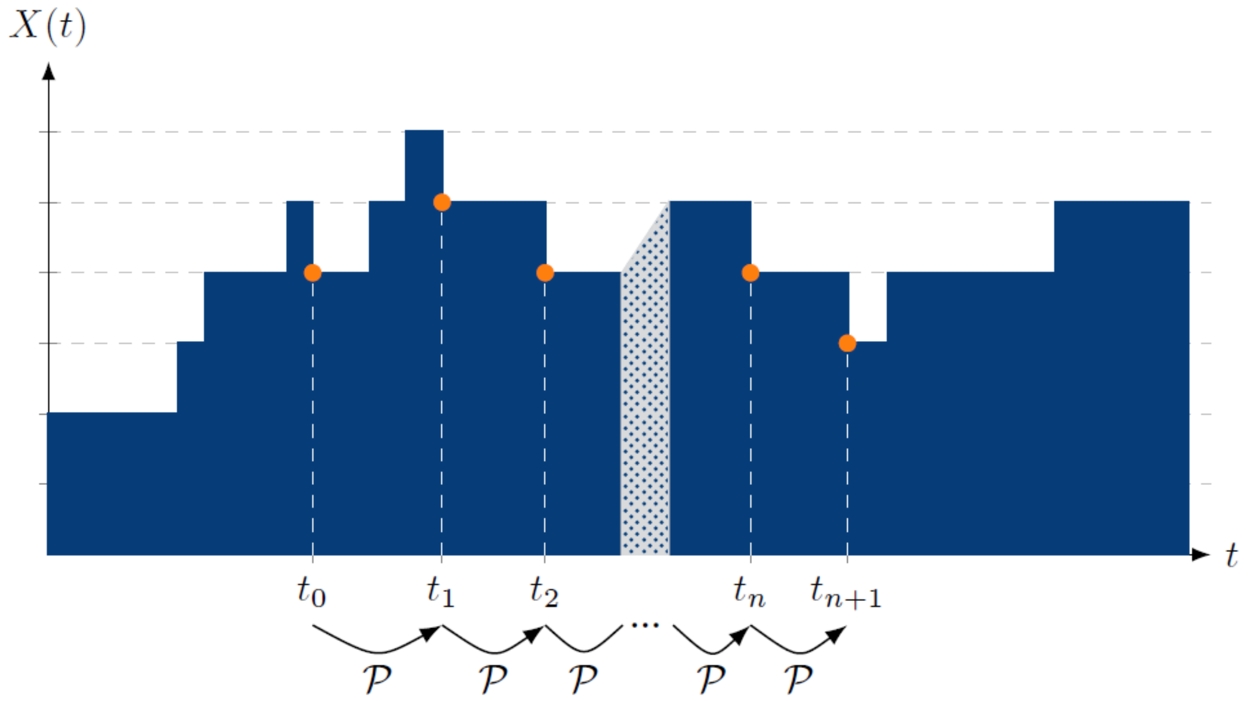
- Non-stationary analysis

$$\mathbf{X}_{n+1} = \mathbf{X}_n \cdot \mathcal{P}$$

- Steady state for $n \to \infty$

$$\mathbf{X}_{n+1} = \mathbf{X}_n = \mathbf{X} , \qquad \mathbf{X} = \mathbf{X} \cdot \mathcal{P}.$$

stationary probability vector of the embedded Markov chain is left-eigenvector of the transition probability matrix with eigenvalue 1



Power Method

Numerical solution

Power Method

- Numerical robust method for non-stationary analysis of embedded Markov chains
- An implementation of the power method is provided at https://modeling.systems

```
def powerMethod(X0, P, stopFunction):
    Z = P.shape[0]  # P is a quadratic matrix of size Z x Z
    X_old = numpy.zeros(Z)
    X1 = X0
    while stopFunction(X1, X_old): # test if steady state is reached
        X_old = X1
        X1 = X_old @ P # compute Xn+1 = Xn*P via matrix multiplication
    return X1
```

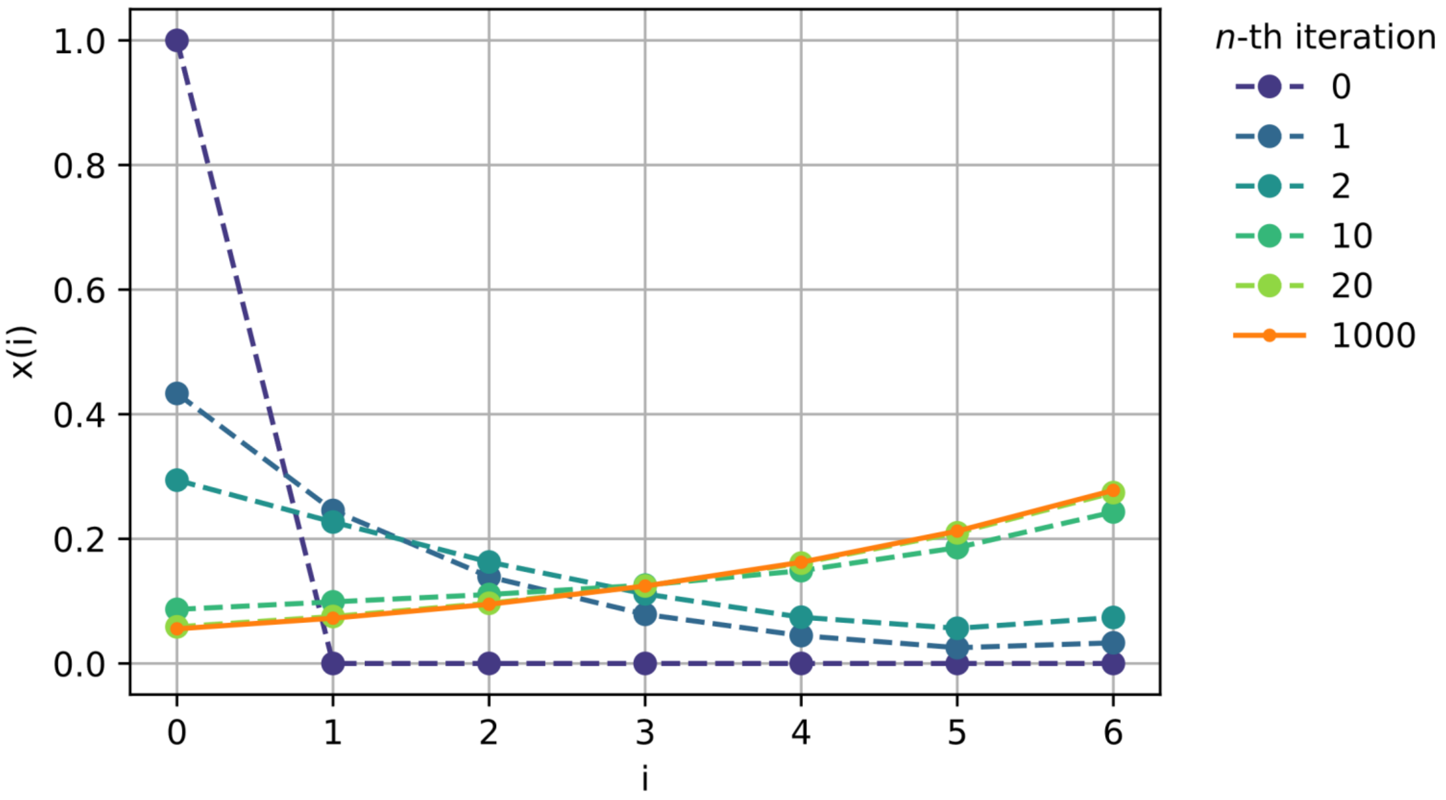
```
def stopFunction_mean(X1, X_old, epsilon=1e-6):
    i = numpy.arange(len(X1)) # with i = (0, 1, ...,  len(X1)-1)
    EX_old = X_old @ i  # expected value E[X_n]
    EX1 = X1 @ i # expected value E[X_n+1]
    return abs(EX_old-EX1) > epsilon

Xmax = 6 # number of states 0, 1, ..., Xmax
X0 = numpy.zeros(Xmax+1)
X0[0] = 1 # initialization: empty system
X =  powerMethod(X0, P, stopFunction_mean)
```

$$\boxed{|\mathrm{E}[X(t_{n+1})] - \mathrm{E}[X(t_n)]| < \varepsilon = 10^{-6})}$$

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Illustration of Power Method

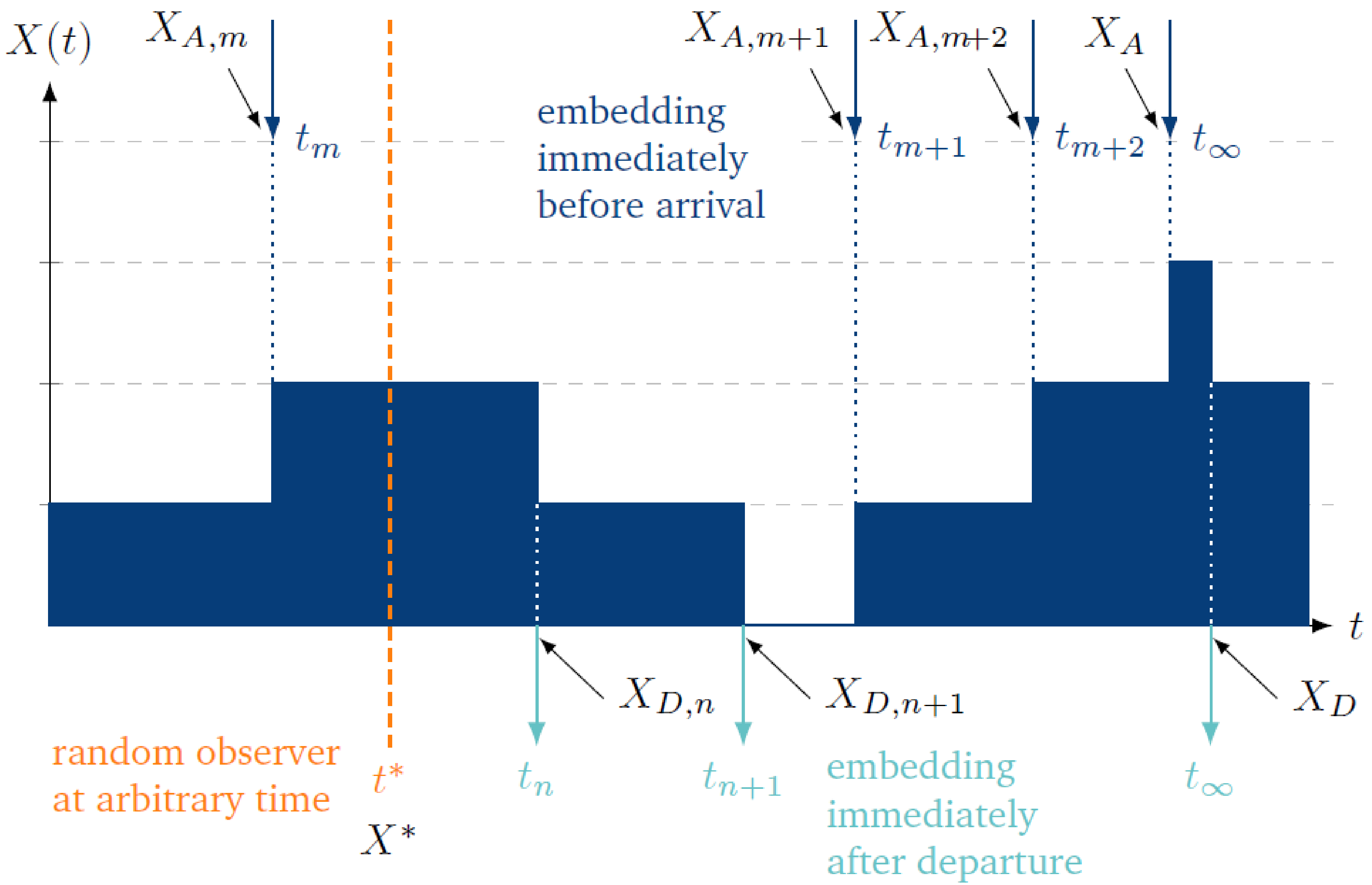
- Analysis of M/GI/1-5 quickly converges



Notion of Embedding Times

Kleinrock's (Burke's) result

Notion of Embedding Times



Variables of System State at Embedding Times

- X_A system state as seen by an arriving customer immediately before or after an arrival
- X_D system state as seen by a departing customer immediately before or after service end
- X^* system state as seen by a random observer, i.e., at an arbitrary time
- X system state at defined embedding times

- Examples for embedding times
  - $X = X_D$ with embedding immediately after departure for M/GI/1
  - $X = X_A$ with embedding immediately before arrival for GI/M/1
  - $X = X^*$ for birth-and-death processes

- In general: $X^* \neq X_A$
- PASTA property: $X^* = X_A$

- In general: $X_A \neq X_D$
- Kleinrock's (Burke's) result shows for which systems $X_A = X_D$

Kleinrock's (Burke's) Result

- System state can change at most by $+1$ or -1. State distribution as seen by an arriving customer will be the same as that seen by a departing customer.

$$x_A(i) = x_D(i) \text{ for } i = 0,1,\ldots$$

- Example: analysis of M/GI/1 queue utilizes this result

- Note: result cannot be used for queues with batch arrivals or batch services

Kleinrock's (Burke's) Result: Derivation

Lecture