Lecture Outline

- Model Formulation
- Graphical Solution Method
- Linear Programming Model
- Solution
- Solving Linear Programming Problems with Excel
- Sensitivity Analysis

Linear Programming (LP)

A model consisting of linear relationships representing a firm's objective and resource constraints

LP is a mathematical modeling technique used to determine a level of operational activity in order to achieve an objective, subject to restrictions called constraints

Types of LP

Linear Programming Model Type	OM Application
Aggregate Production Planning	Determines the resource capacity needed to meet demand over an immediate time horizon, including units produced, workers hired and fired and inventory. (See Chapter 13.)
Product Mix	Mix of different products to produce that will maximize profit or minimize cost given resource constraints such as material, labor, budget, etc.
Transportation	Logistical flow of items (goods or services) from sources to destinations, for example, truckloads of goods from plants to warehouses. (See Supplement 10.)
Transshipment	Flow of items from sources to destinations with intermediate points, for example shipping from plant to distribution center and then to stores. (See Supplement 10.)

Types of LP (cont.)

Linear Programming Model Type	OM Application
Assignment	Assigns work to limited resources, called "Loading," for example, assigning jobs or workers to different machines. (See Chapter 16.)
Multiperiod Scheduling	Schedules regular and overtime production, plus inventory to carry over, to meet demand in future periods.
Blend	Determines "recipe" requirements, for example, how to blend different petroleum components to produce different grades of gasoline and other petroleum products.
Diet	Menu of food items that meets nutritional or other requirements, for example, hospital or school cafeteria menus.
Investment/Capital Budgeting	Financial model that determines amount to invest in different alternatives given return objectives and constraints for risk, diversity, etc., for example, how much to invest in new plant, facilities or equipment.

Types of LP (cont.)

Linear Programming Model Type	OM Application
Data Envelopment Analysis (DEA)	Compares service units of the same type—banks, hospitals, schools—based on their resources and outputs to see which units are less productive or inefficient.
Shortest Route	Shortest routes from sources to destinations, for example, the shortest highway truck route from coast to coast.
Maximal Flow	Maximizes the amount of flow from sources to destinations, for example, the flow of work-in process through an assembly operation.
Trim-Loss	Determines patterns to cut sheet items to minimize waste, for example, cutting lumber, film, cloth, glass, etc.
Facility Location	Selects facility locations based on constraints such as fixed, operating, and shipping costs, production capacity, etc.
Set Covering	Selection of facilities that can service a set of other facilities, for example, the selection of distribution hubs that will be able to deliver packages to a set of cities.

LP Model Formulation

- Decision variables
  - mathematical symbols representing levels of activity of an operation
- Objective function
  - a linear relationship reflecting the objective of an operation
  - most frequent objective of business firms is to *maximize profit*
  - most frequent objective of individual operational units (such as a production or packaging department) is to *minimize cost*
- Constraint
  - a linear relationship representing a restriction on decision making

LP Model Formulation (cont.)

Max/min $z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

subject to:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \ (\leq, =, \geq)\ b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \ (\leq, =, \geq)\ b_2 \\ \vdots \\ a_{m1}x1 + a_{m2}x_2 + \ldots + a_{mn}x_n \ (\leq, =, \geq)\ b_m \end{cases}$$

x_j = decision variables
b_i = constraint levels
c_j = objective function coefficients
a_{ij} = constraint coefficients

LP Model: Example

	RESOURCE REQUIREMENTS		
PRODUCT	*Labor (hr/unit)*	*Clay (lb/unit)*	*Revenue ($/unit)*
Bowl	1	4	40
Mug	2	3	50

There are 40 hours of labor and 120 pounds of clay available each day

Decision variables

x_1 = number of bowls to produce

x_2 = number of mugs to produce

LP Formulation: Example

Maximize $Z = \$40\ x_1 + 50\ x_2$

Subject to

$$x_1 + 2x_2 \leq 40 \text{ hr} \quad \text{(labor constraint)}$$
$$4x_1 + 3x_2 \leq 120 \text{ lb} \quad \text{(clay constraint)}$$
$$x_1, x_2 \geq 0$$

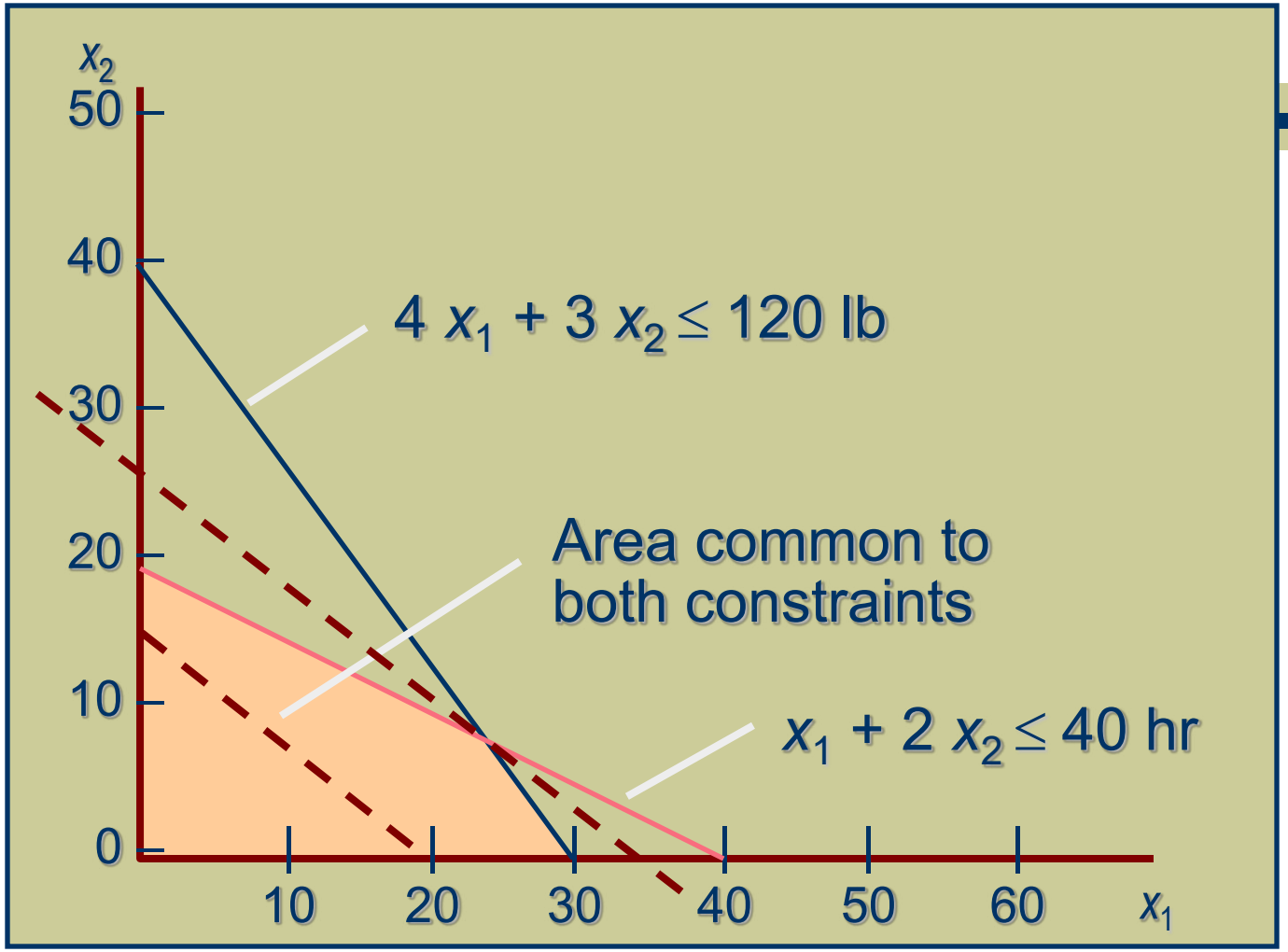
Solution is $x_1 = 24$ bowls $x_2 = 8$ mugs

Revenue = \$1,360

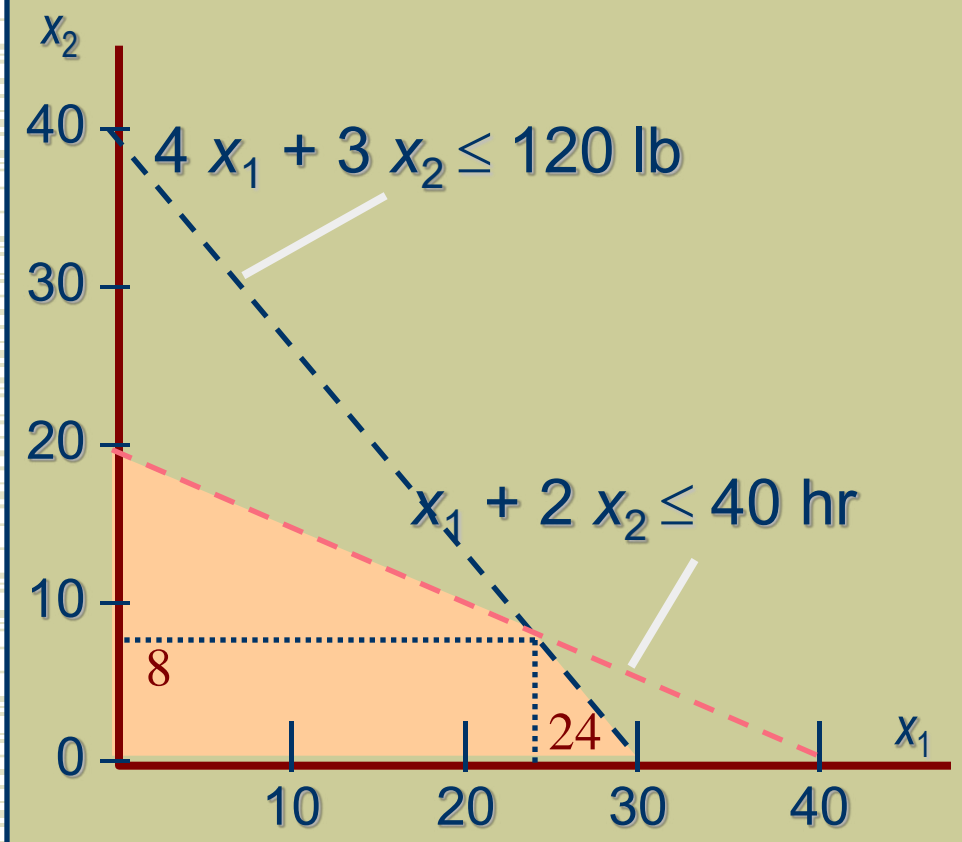
Graphical Solution Method

1. Plot model constraint on a set of coordinates in a plane
2. Identify the feasible solution space on the graph where all constraints are satisfied simultaneously
3. Plot objective function to find the point on boundary of this space that maximizes (or minimizes) value of objective function

Graphical Solution: Example



Computing Optimal Values

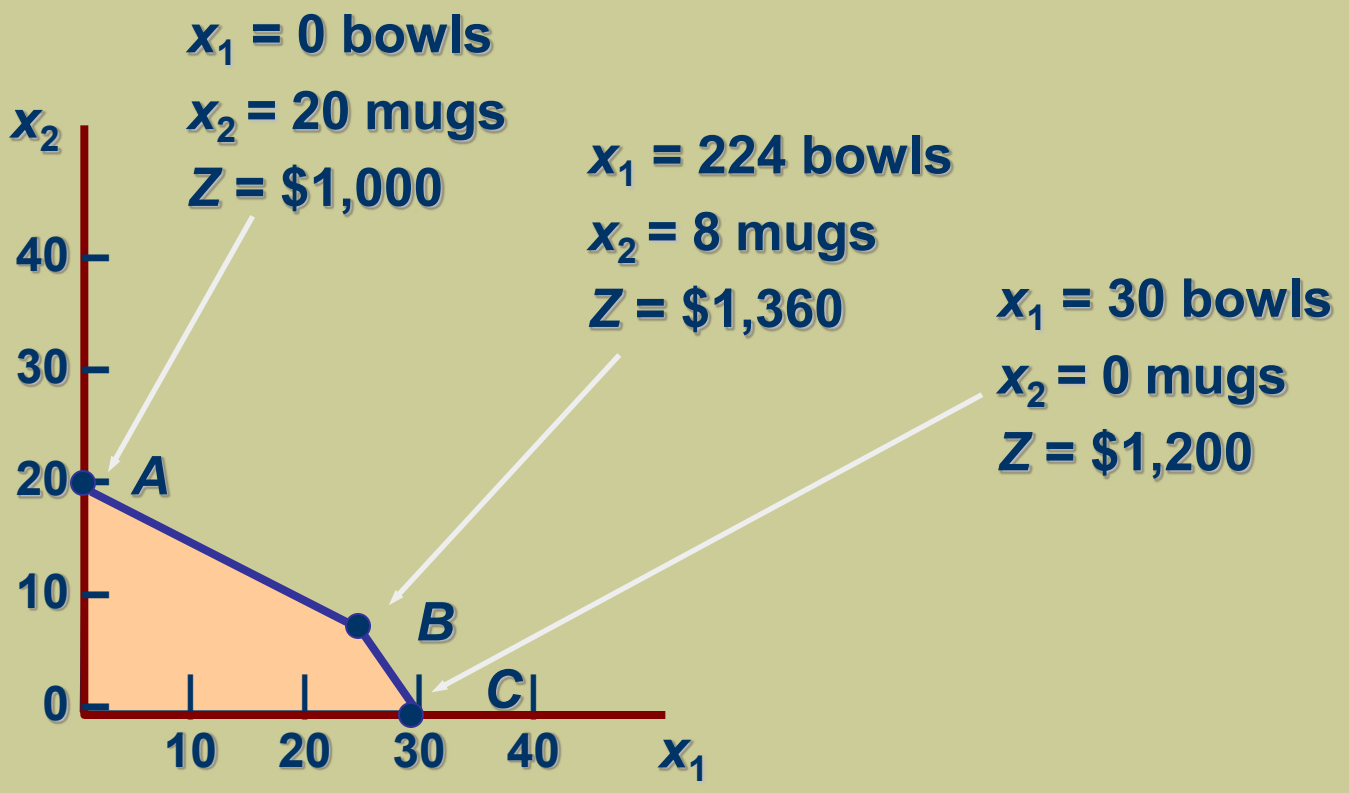


$$
\begin{array}{rrcr}
x_1 + & 2x_2 & = & 40 \\
4x_1 + & 3x_2 & = & 120 \\
\hline
4x_1 + & 8x_2 & = & 160 \\
-4x_1 - & 3x_2 & = & -120 \\
\hline
 & 5x_2 & = & 40 \\
 & x_2 & = & 8
\end{array}
$$

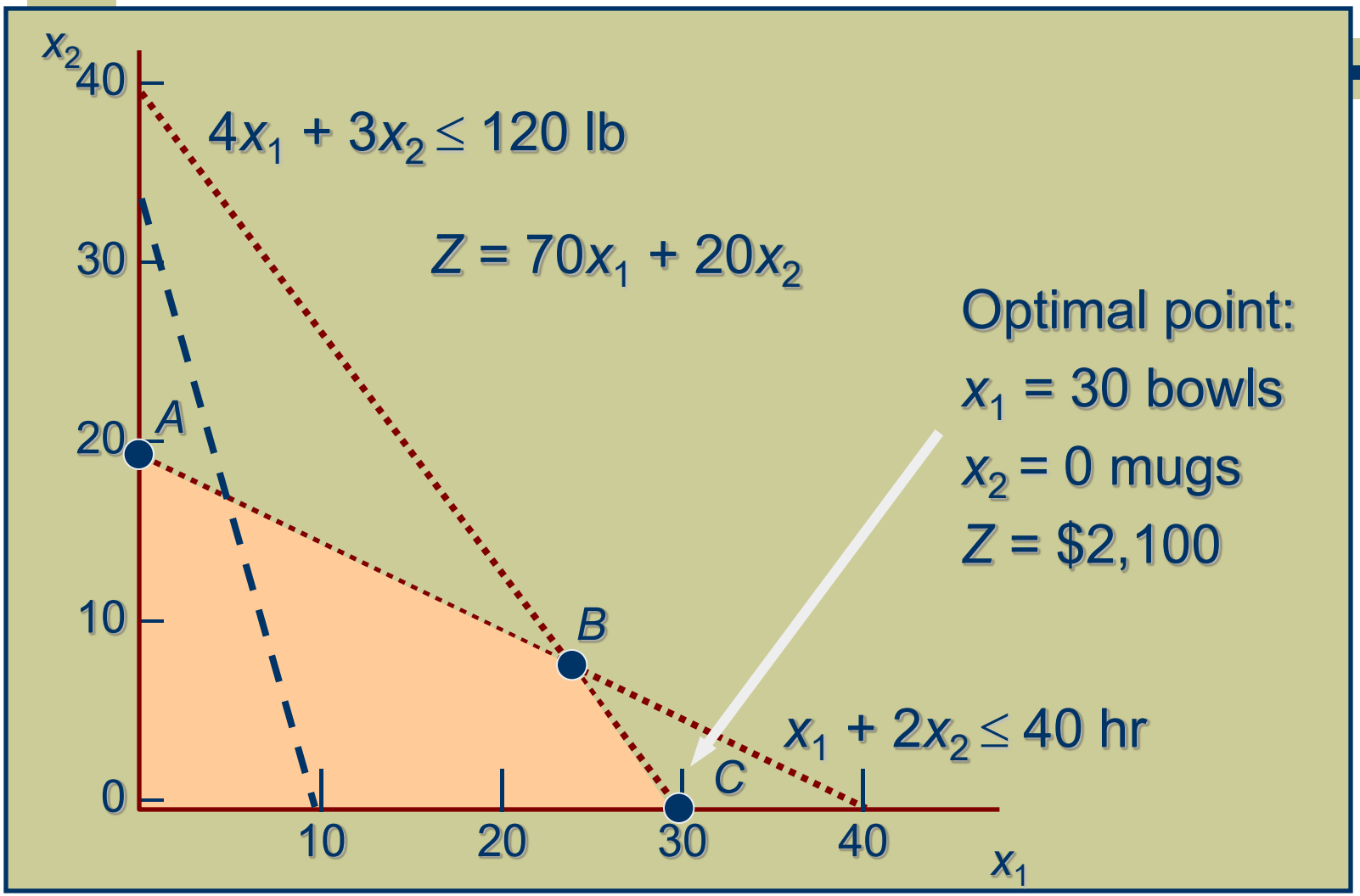
$$
\begin{array}{rrcr}
x_1 + & 2(8) & = & 40 \\
 & x_1 & = & 24
\end{array}
$$

$Z = \$50(24) + \$50(8) = \$1,360$

Extreme Corner Points



Objective Function



Minimization Problem

	CHEMICAL CONTRIBUTION	
Brand	*Nitrogen (lb/bag)*	*Phosphate (lb/bag)*
Gro-plus	2	4
Crop-fast	4	3

Minimize $Z = \$6x_1 + \$3x_2$

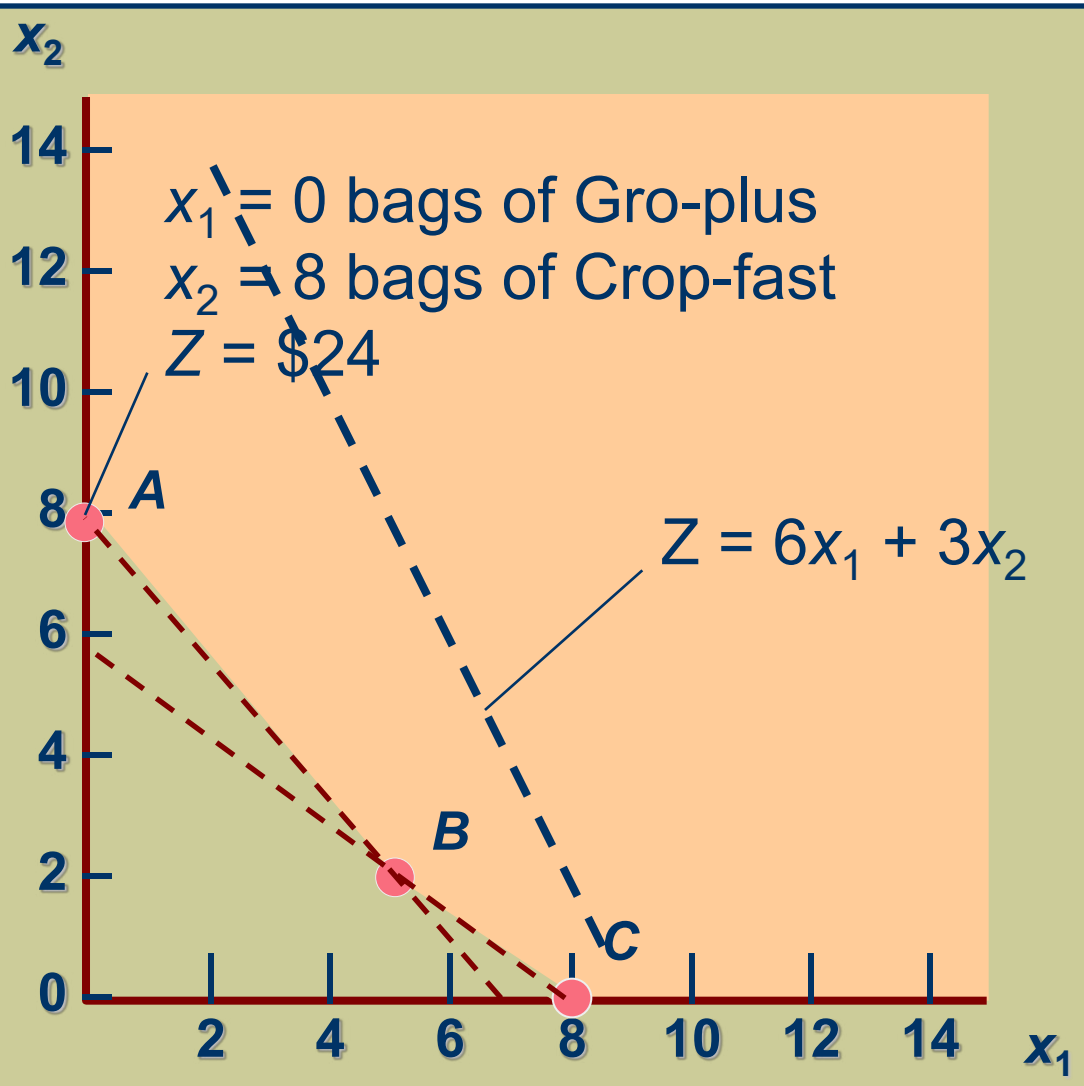
subject to

$2x_1 + 4x_2 \geq 16$ lb of nitrogen

$4x_1 + 3x_2 \geq 24$ lb of phosphate

$x_1, x_2 \geq 0$

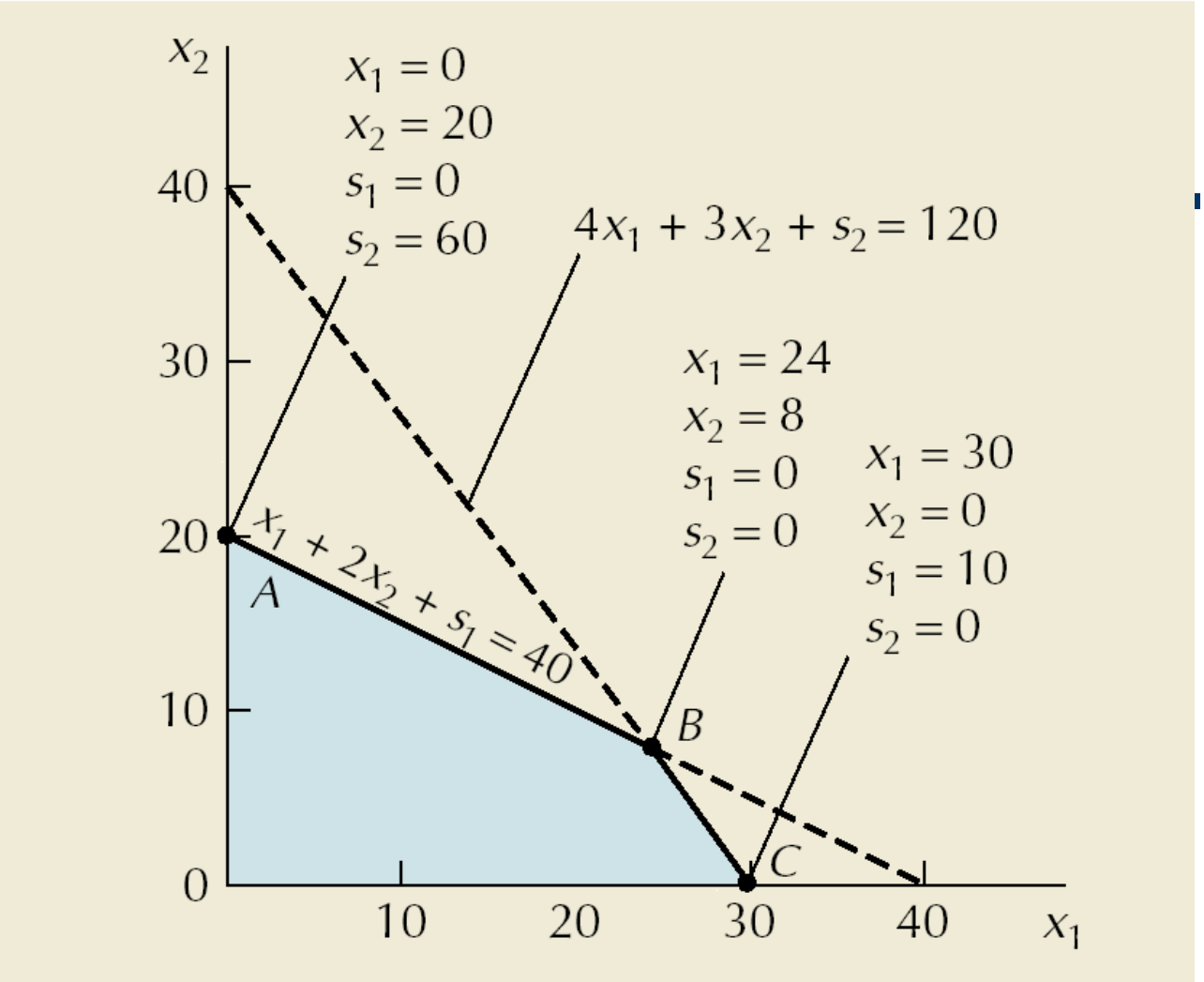
Graphical Solution



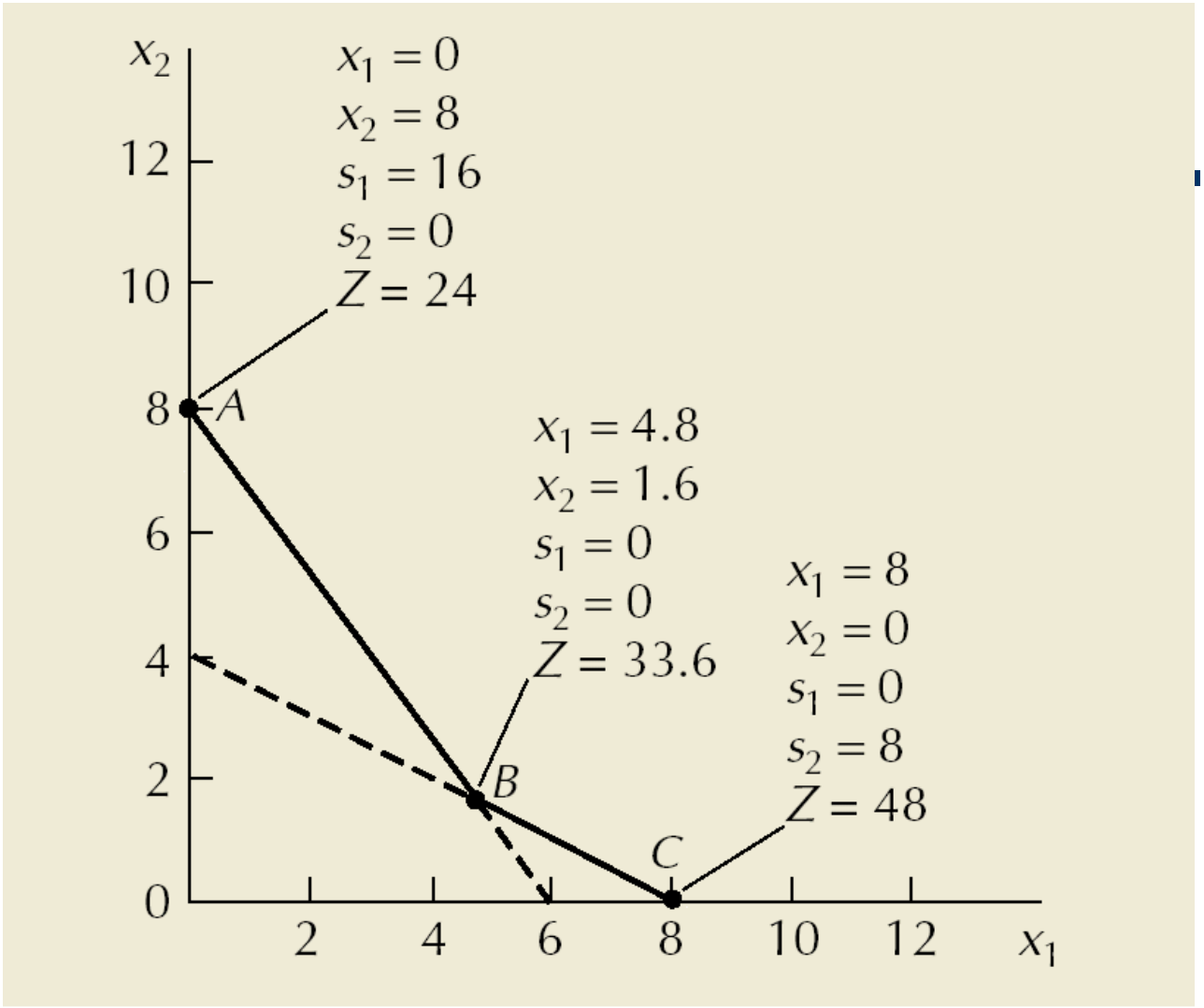
Simplex Method

- A mathematical procedure for solving linear programming problems according to a set of steps
- Slack variables added to ≤ constraints to represent unused resources
  - $x_1 + 2x_2 + s_1 = 40$ hours of labor
  - $4x_1 + 3x_2 + s_2 = 120$ lb of clay
- Surplus variables subtracted from ≥ constraints to represent excess above resource requirement. For example
  - $2x_1 + 4x_2 \geq 16$ is transformed into
  - $2x_1 + 4x_2 - s_1 = 16$
- Slack/surplus variables have a 0 coefficient in the objective function
  - $Z = \$40x_1 + \$50x_2 + 0s_1 + 0s_2$

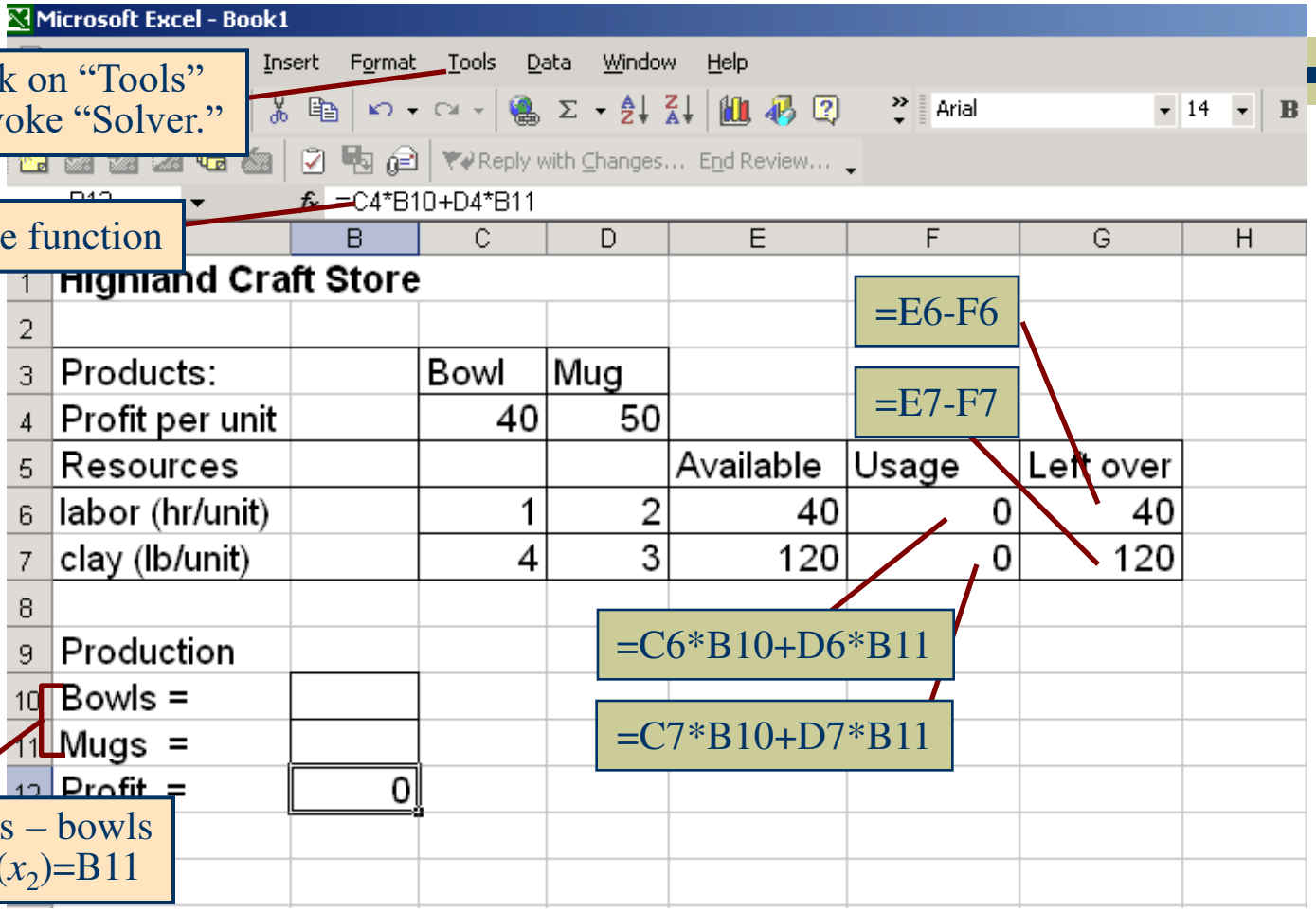
Solution Points with Slack Variables



Solution Points with Surplus Variables



Solving LP Problems with Excel



	A	B	C	D	E	F	G	H
1	**Highland Craft Store**							
2								
3	Products:		Bowl	Mug				
4	Profit per unit		40	50				
5	Resources				Available	Usage	Left over	
6	labor (hr/unit)		1	2	40	0	40	
7	clay (lb/unit)		4	3	120	0	120	
8								
9	Production							
10	Bowls =							
11	Mugs =							
12	Profit =		0					

Solving LP Problems with Excel (cont.)

After all parameters and constraints have been input, click on "Solve."

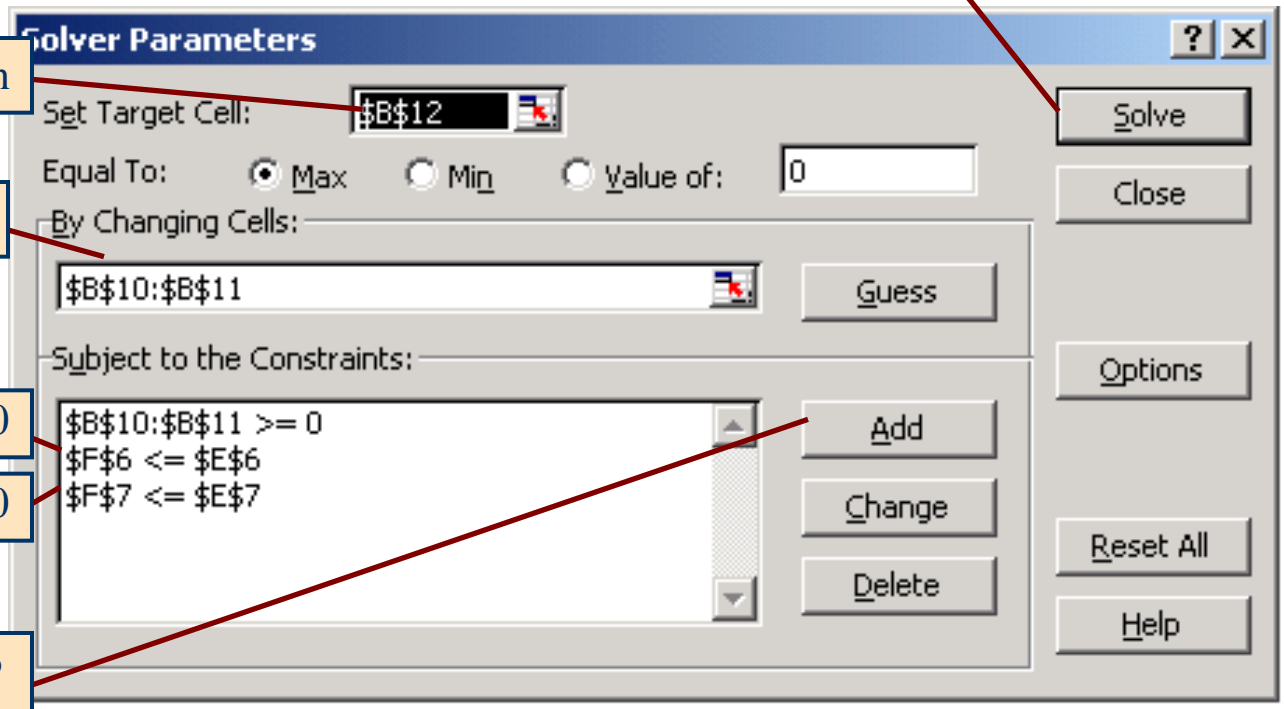
Objective function

Decision variables

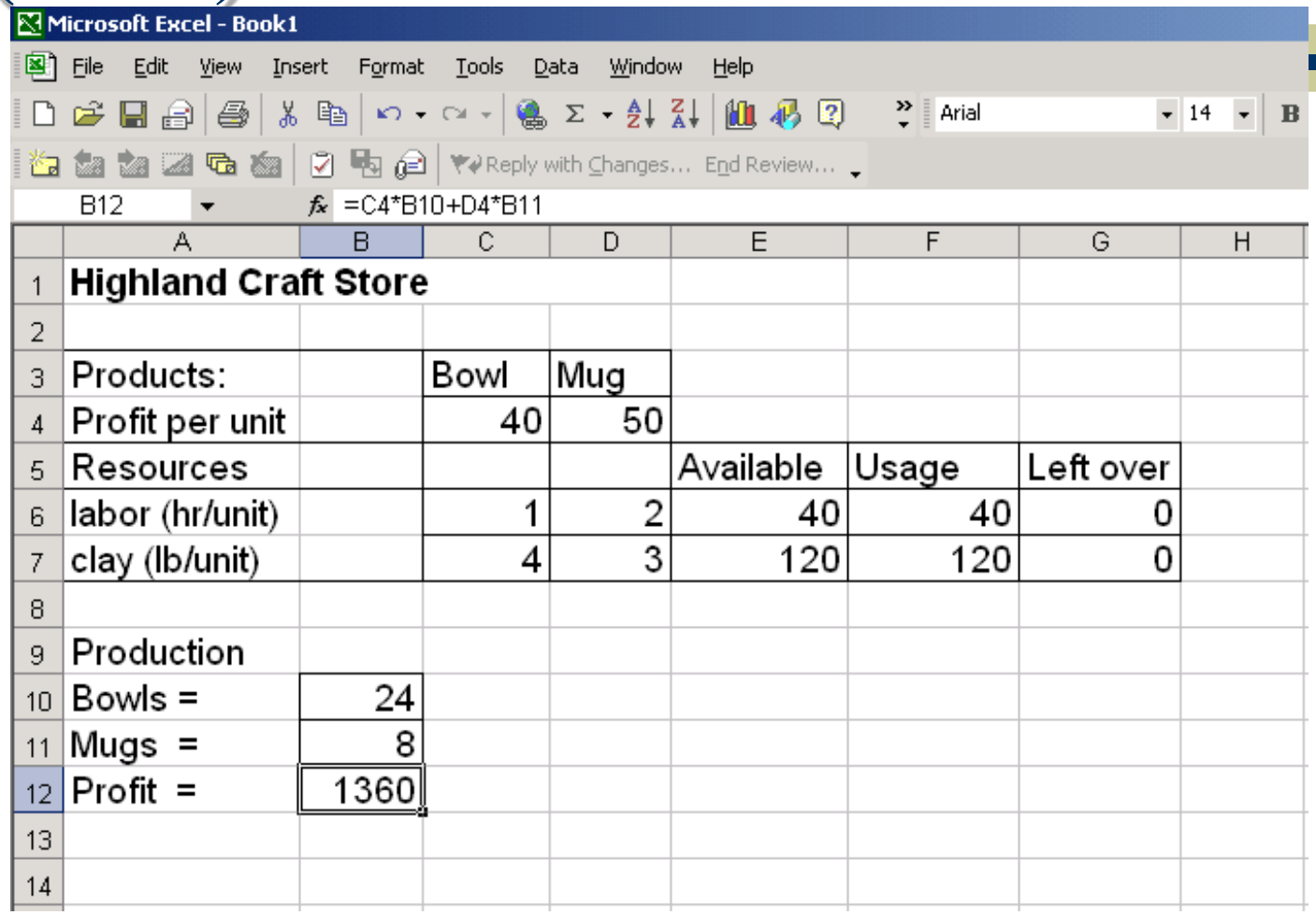
C6*B10+D6*B11≤40

C7*B10+D7*B11≤120

Click on "Add" to insert constraints

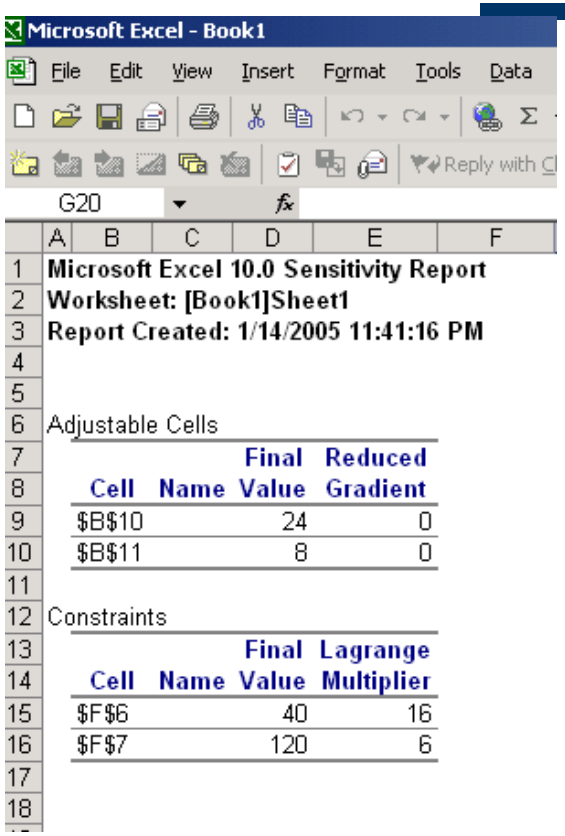


Solving LP Problems with Excel (cont.)



	A	B	C	D	E	F	G	H
1	**Highland Craft Store**							
2								
3	Products:		Bowl	Mug				
4	Profit per unit		40	50				
5	Resources				Available	Usage	Left over	
6	labor (hr/unit)		1	2	40	40	0	
7	clay (lb/unit)		4	3	120	120	0	
8								
9	Production							
10	Bowls =	24						
11	Mugs =	8						
12	Profit =	1360						
13								
14								

Sensitivity Analysis



Microsoft Excel - Book1

File Edit View Insert Format Tools Data

G20

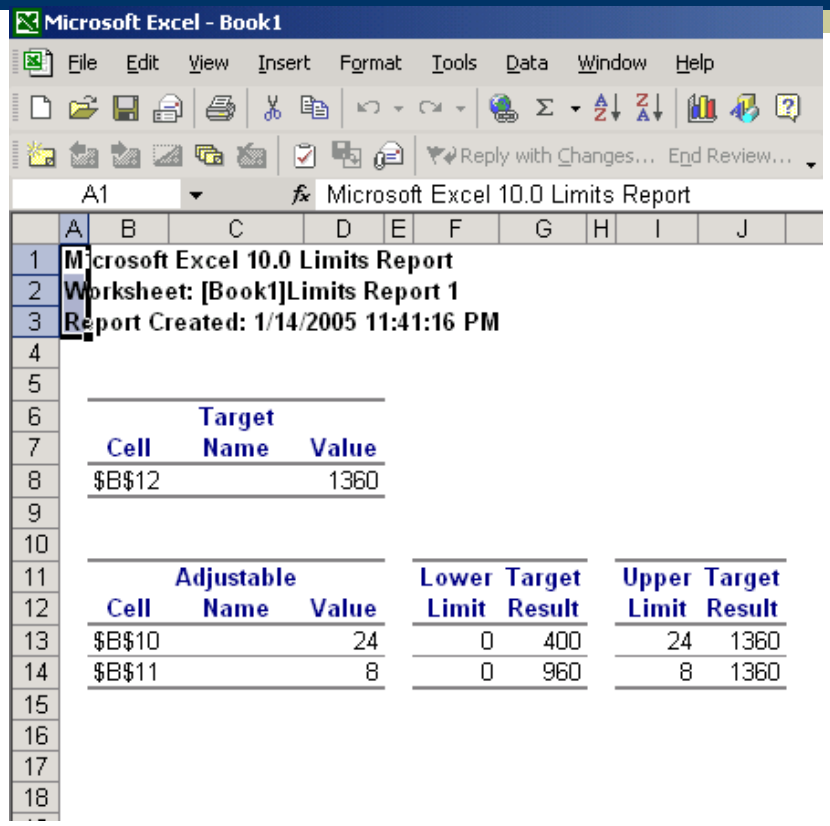
Microsoft Excel 10.0 Sensitivity Report
Worksheet: [Book1]Sheet1
Report Created: 1/14/2005 11:41:16 PM

Adjustable Cells

Cell	Name	Final Value	Reduced Gradient
$B$10		24	0
$B$11		8	0

Constraints

Cell	Name	Final Value	Lagrange Multiplier
$F$6		40	16
$F$7		120	6



Microsoft Excel - Book1

File Edit View Insert Format Tools Data Window Help

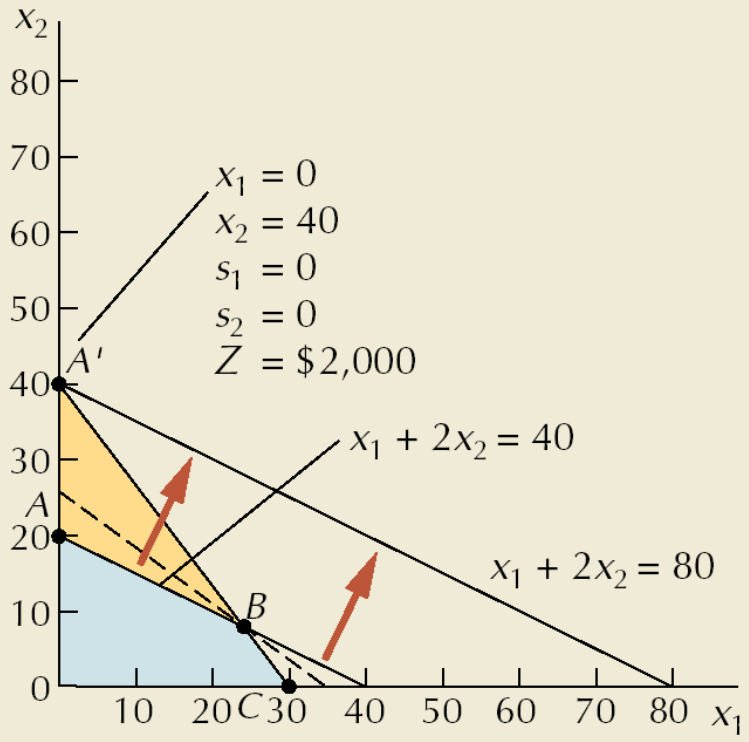
A1 Microsoft Excel 10.0 Limits Report

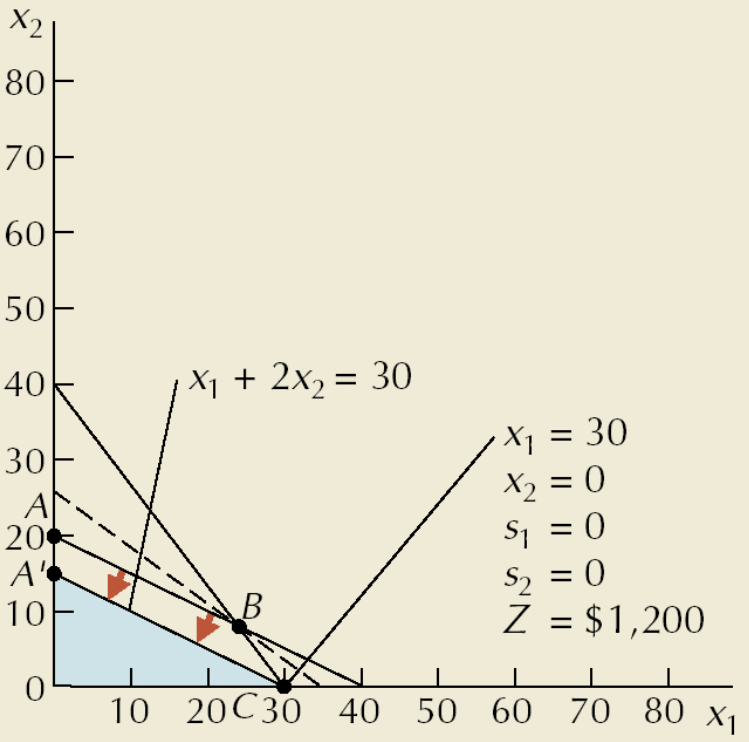
Microsoft Excel 10.0 Limits Report
Worksheet: [Book1]Limits Report 1
Report Created: 1/14/2005 11:41:16 PM

Cell	Name	Target Value
$B$12		1360

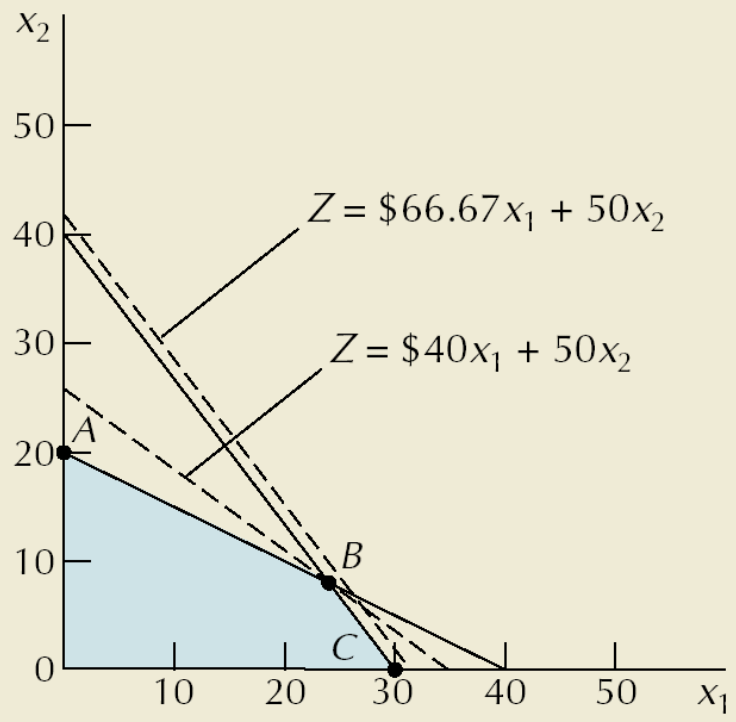
Cell	Name	Adjustable Value	Lower Limit	Target Result	Upper Limit	Target Result
$B$10		24	0	400	24	1360
$B$11		8	0	960	8	1360

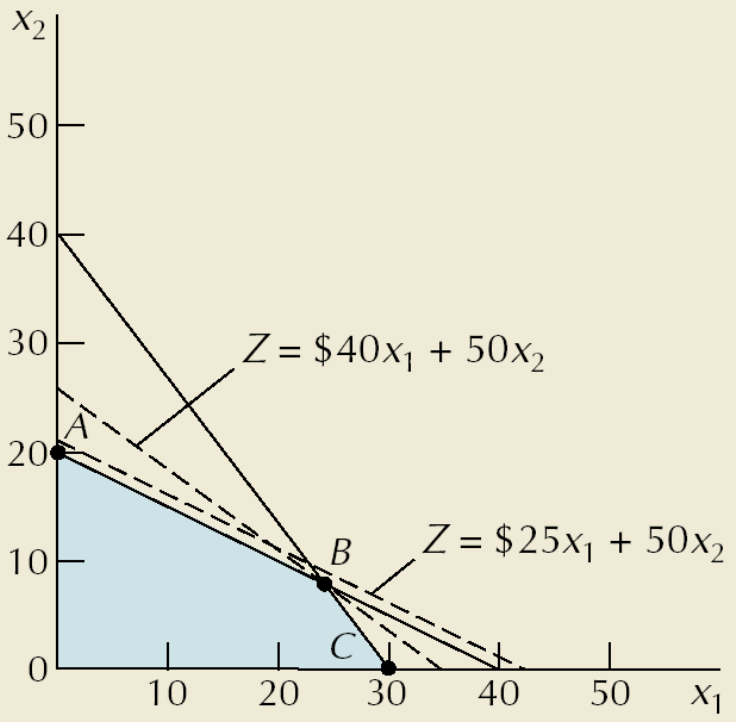
Sensitivity Range for Labor Hours





Sensitivity Range for Bowls





TRANSPORTATION PROBLEM

OBJECTIVES

- Work Center Defined
- Typical Scheduling and Control Functions
- Job-shop Scheduling
- Examples of Scheduling Rules
- Shop-floor Control
- Principles of Work Center Scheduling
- Issues in Scheduling Service Personnel

Work Center

- A work center is an area in a business in which productive resources are organized and work is completed
- Can be a single machine, a group of machines, or an area where a particular type of work is done

Capacity and Scheduling

- Infinite loading (Example: MRP)
- Finite loading
- Forward scheduling
- Backward scheduling (Example: MRP)

Types of Manufacturing Scheduling Processes and Scheduling Approaches

Type of Process	Typical Scheduling Approach
Continuous process	Finite forward of process, machine limited
High-volume manufacturing	Finite forward of line, machined limited
Med-volume manufacturing	Infinite forward of process, labor and machined limited
Low-volume manufacturing	Infinite forward of jobs, labor and some machine limited

Typical Scheduling and Control Functions

- Allocating orders, equipment, and personnel
- Determining the sequence of order performance
- Initiating performance of the scheduled work
- Shop-floor control

Work-Center Scheduling Objectives

- Meet due dates
- Minimize lead time
- Minimize setup time or cost
- Minimize work-in-process inventory
- Maximize machine utilization

Priority Rules for Job Sequencing

1. First-come, first-served (FCFS)
2. Shortest operating time (SOT)
3. Earliest due date first (DDate)
4. Slack time remaining (STR) first
5. Slack time remaining per operation (STR/OP)

Priority Rules for Job Sequencing (Continued)

6. Critical ratio (CR)

$$CR = \frac{(\text{Due date} - \text{Current date})}{\text{Number of days remaining}}$$

7. Last come, first served (LCFS)
8. Random order or whim

Example of Job Sequencing: First-Come First-Served

Suppose you have the four jobs to the right arrive for processing on one machine

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)
A	4	5
B	7	10
C	3	6
D	1	4

What is the FCFS schedule?

Do all the jobs get done on time?

Answer: FCFS Schedule

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)	Flow Time (days)
A	4	5	4
B	7	10	11
C	3	6	14
D	1	4	15

No, Jobs B, C, and D are going to be late

Example of Job Sequencing: Shortest Operating Time

Suppose you have the four jobs to the right arrive for processing on one machine

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)
A	4	5
B	7	10
C	3	6
D	1	4

What is the SOT schedule?

Do all the jobs get done on time?

Answer: Shortest Operating Time Schedule

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)	Flow Time (days)
D	1	4	1
C	3	6	4
A	4	5	8
B	7	10	15

No, Jobs A and B are going to be late

Example of Job Sequencing: Earliest Due Date First

Suppose you have the four jobs to the right arrive for processing on one machine

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)
A	4	5
B	7	10
C	3	6
D	1	4

What is the earliest due date first schedule?

Do all the jobs get done on time?

Answer: Earliest Due Date First

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)	Flow Time (days)
D	1	4	1
A	4	5	5
C	3	6	8
B	7	10	15

No, Jobs C and B are going to be late

Example of Job Sequencing: Critical Ratio Method

Suppose you have the four jobs to the right arrive for processing on one machine

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)
A	4	5
B	7	10
C	3	6
D	1	4

What is the CR schedule?

Do all the jobs get done on time?

In order to do this schedule the CR's have be calculated for each job. If we let today be Day 1 and allow a total of 15 days to do the work. The resulting CR's and order schedule are:

CR(A)=(5-4)/15=0.06 (Do this job last)

CR(B)=(10-7)/15=0.20 (Do this job first, tied with C and D)

CR(C)=(6-3)/15=0.20 (Do this job first, tied with B and D)

CR(D)=(4-1)/15=0.20 (Do this job first, tied with B and C)

No, but since there is three-way tie, only the first job or two will be on time

Example of Job Sequencing: Last-Come First-Served

Suppose you have the four jobs to the right arrive for processing on one machine

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)
A	4	5
B	7	10
C	3	6
D	1	4

What is the LCFS schedule?

Do all the jobs get done on time?

Answer: Last-Come First-Served Schedule

Jobs (in order of arrival)	Processing Time (days)	Due Date (days hence)	Flow Time (days)
D	1	4	1
C	3	6	4
B	7	10	11
A	4	5	15

No, Jobs B and A are going to be late

Example of Job Sequencing: Johnson's Rule (Part 1)

Suppose you have the following five jobs with time requirements in two stages of production. What is the job sequence using Johnson's Rule?

	Time in Hours	
Jobs	Stage 1	Stage 2
A	1.50	1.25
B	2.00	3.00
C	2.50	2.00
D	1.00	2.00

Example of Job Sequencing: Johnson's Rule (Part 2)

First, select the job with the smallest time in either stage.

That is Job D with the smallest time in the first stage. Place that job as early as possible in the unfilled job sequence below.

	Time in Hours	
Jobs	Stage 1	Stage 2
A	1.50	1.25
B	2.00	3.00
C	2.50	2.00
D	1.00	2.00

Drop D out, select the next smallest time (Job A), and place it 4th in the job sequence.

Drop A out, select the next smallest time. There is a tie in two stages for two different jobs. In this case, place the job with the smallest time in the first stage as early as possible in the unfilled job sequence.

Then place the job with the smallest time in the second stage as late as possible in the unfilled sequence.

Job Sequence	1	2	3	4
Job Assigned	D	B	C	A

Shop-Floor Control: Major Functions

1. Assigning priority of each shop order

2. Maintaining work-in-process quantity information

3. Conveying shop-order status information to the office

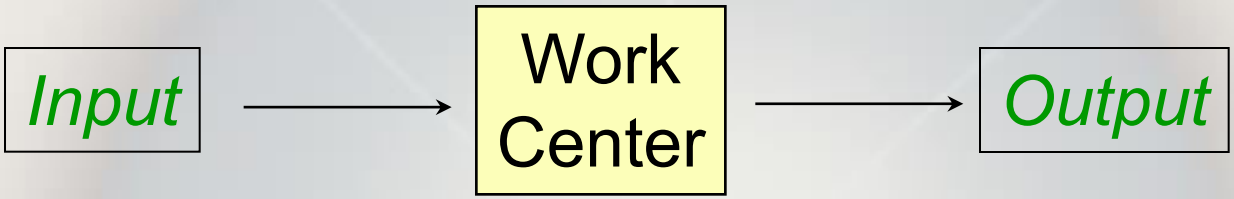
Shop-Floor Control: Major Functions (Continued)

4. Providing actual output data for capacity control purposes

5. Providing quantity by location by shop order for WIP inventory and accounting purposes

6. Providing measurement of efficiency, utilization, and productivity of manpower and machines

Input/Output Control



- ***Planned input* should never exceed *planned output***
- **Focuses attention on bottleneck work centers**

Principles of Work Center Scheduling

1. There is a direct equivalence between work flow and cash flow
2. The effectiveness of any job shop should be measured by speed of flow through the shop
3. Schedule jobs as a string, with process steps back-to-back
4. A job once started should not be interrupted

Principles of Job Shop Scheduling (Continued)

5. Speed of flow is most efficiently achieved by focusing on bottleneck work centers and jobs

6. Reschedule every day

7. Obtain feedback each day on jobs that are not completed at each work center

8. Match work center input information to what the worker can actually do

Principles of Job Shop Scheduling (Continued)

9. When seeking improvement in output, look for incompatibility between engineering design and process execution

10. Certainty of standards, routings, and so forth is not possible in a job shop, but always work towards achieving it

Personnel Scheduling in Services

- Scheduling consecutive days off
- Scheduling daily work times
- Scheduling hourly work times

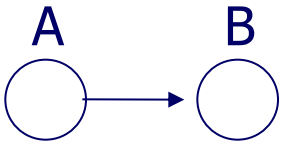
End of Chapter 17

PERT / CPM advantages

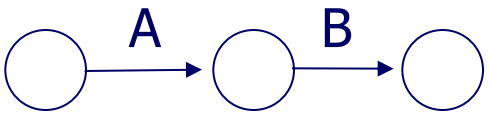
1. *Disciplined planning*
2. *Realistic objectives*
3. *Unambiguous communication*
4. *Allows management by exception of critical tasks*

Drawing networks

- ***Activities on nodes (AoN)***

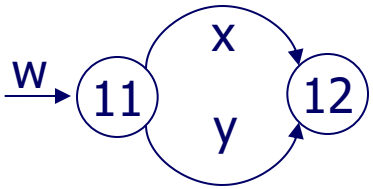


- ***Activities on arrows (AoA)***

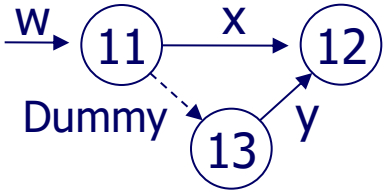


- Dummy activities – In AoA, any 2 events in network can be directly connected to only one activity.

Wrong:

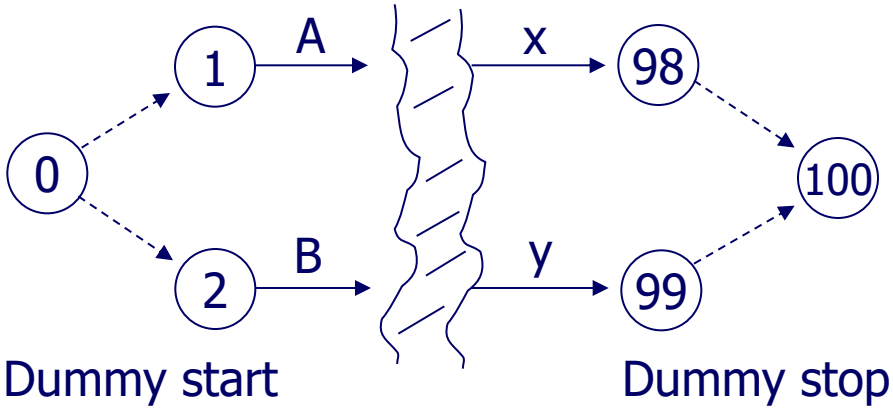


Right:

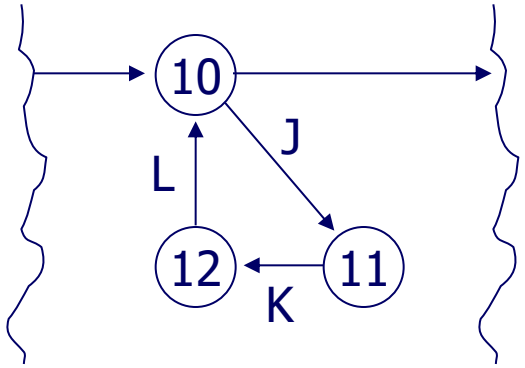


Drawing networks (cont.)

- Many computer programs require one initial event and one terminal event



- What's wrong with this?



Drawing exercise 1

Activity	Predecessors
A	-
B	A
C	A
D	B,C
E	C
F	D
G	E,F

Draw this format in AoN format.

Repeat in AoA format.

Drawing exercise 2

Activity	Predecessors
A	-
B	-
C	A,B
D	A,B
E	C
F	C,D
G	F

Scheduling calculations:

1. Forward pass – Start each activity as early as possible.

   ES = Max. EF of immediate predecessors

   EF = ES + Task time

2. Backward pass – Work backward from the project completion time in step 1, starting each activity as late as possible.

   LS = LF – Task time

   LF = Min. LS of immediate successors

3. Compute slack times.

   Slack = LS – ES

   or

   LF – EF

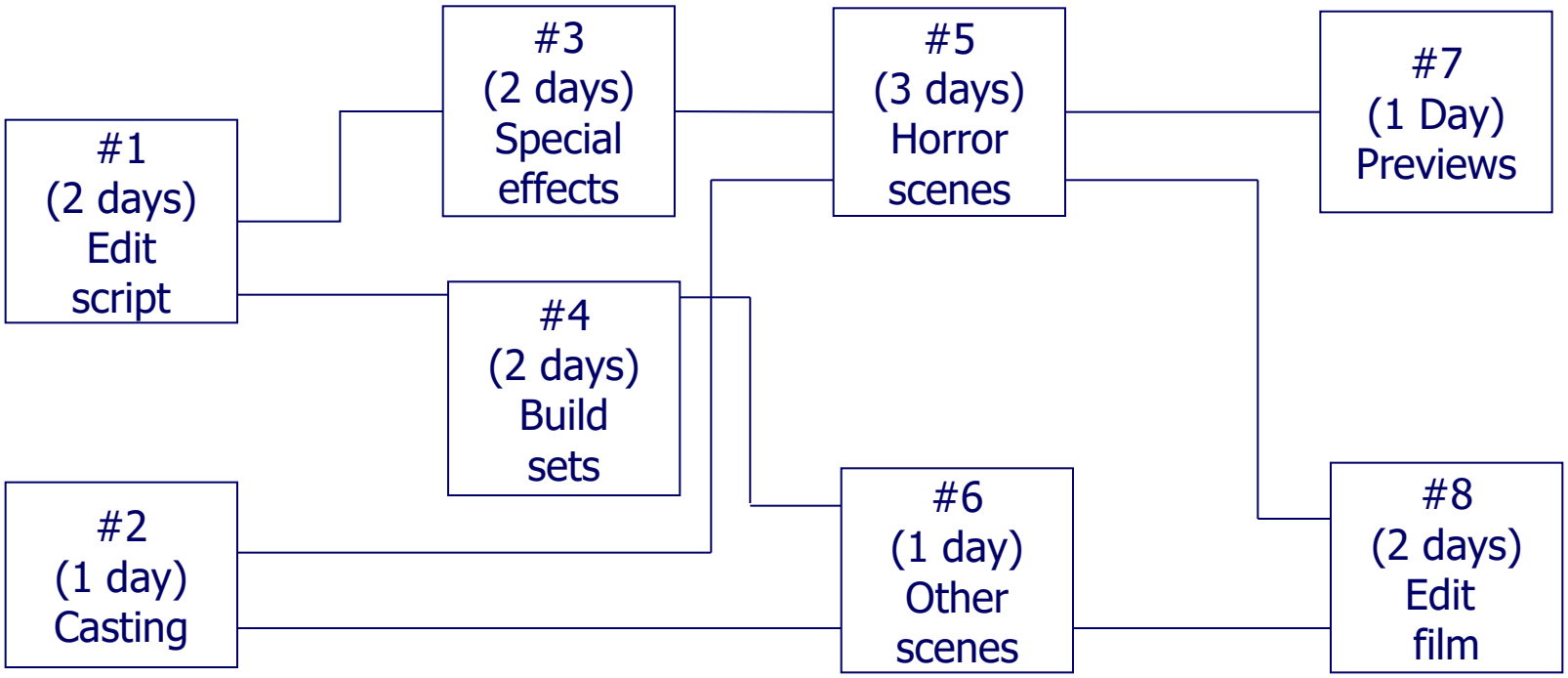
Black Lagoon project

Work breakdown table:

			Preceding tasks		Succeeding tasks	
Description	**Task #**	**Days**	**A**	**B**	**A**	**B**
Edit script	1	2			3	4
Casting	2	1			5	6
Special effects	3	2	1		5	
Build sets	4	2	1		6	
Horror scenes	5	3	3	4	7	8
Other scenes	6	1	2	4	8	
Previews	7	1	5			
Edit film	8	2	5	6		

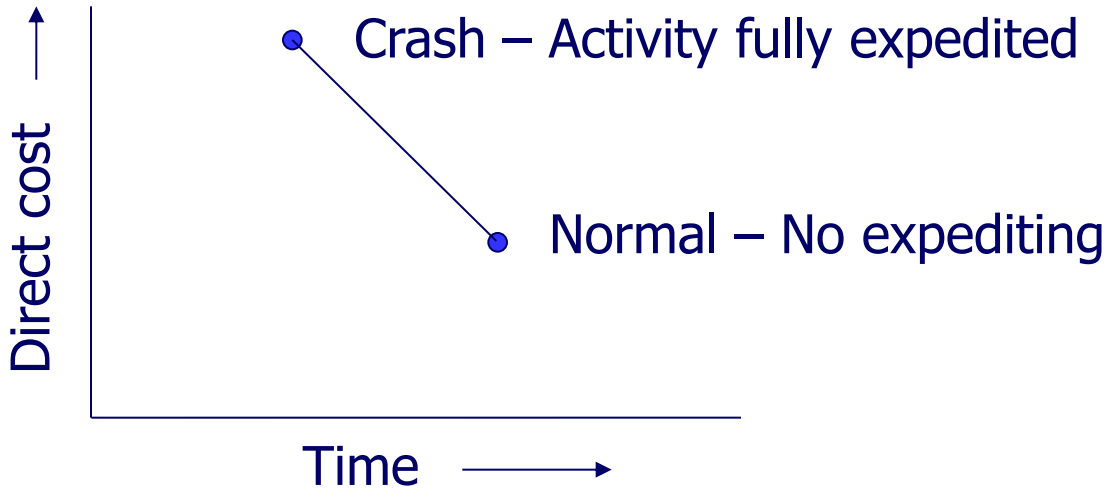
Black Lagoon project (cont.)

Precedence diagram:



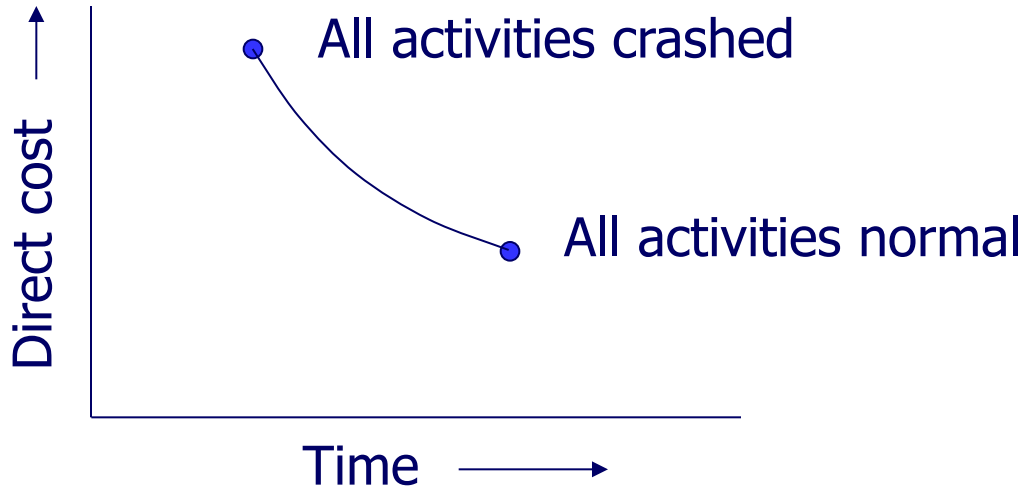
Time-cost tradeoffs

- **Activity direct costs**



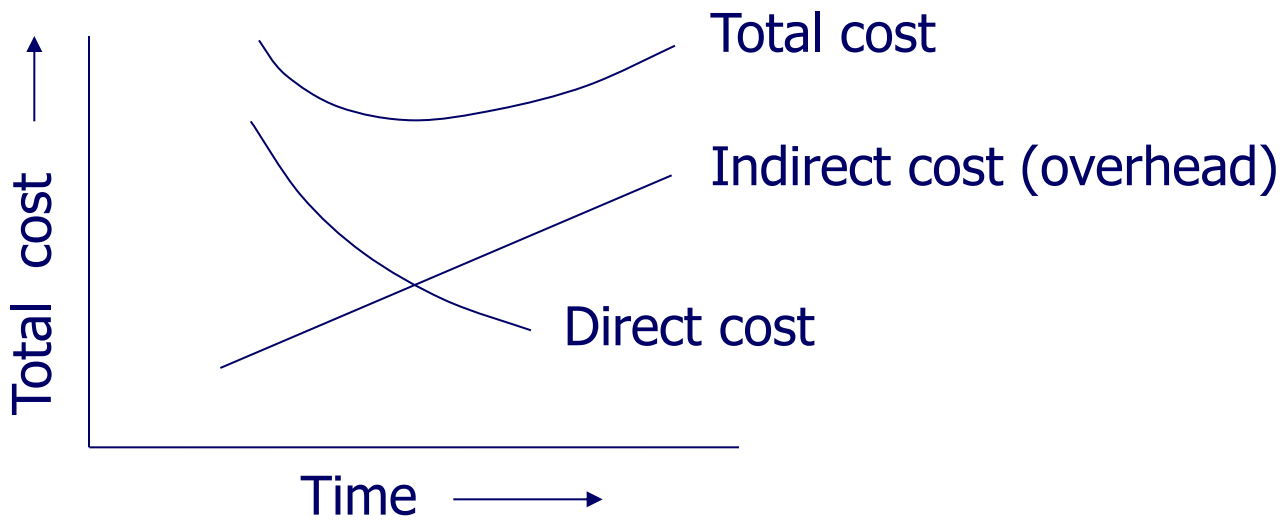
Time-cost tradeoffs (cont.)

- **Project direct costs**



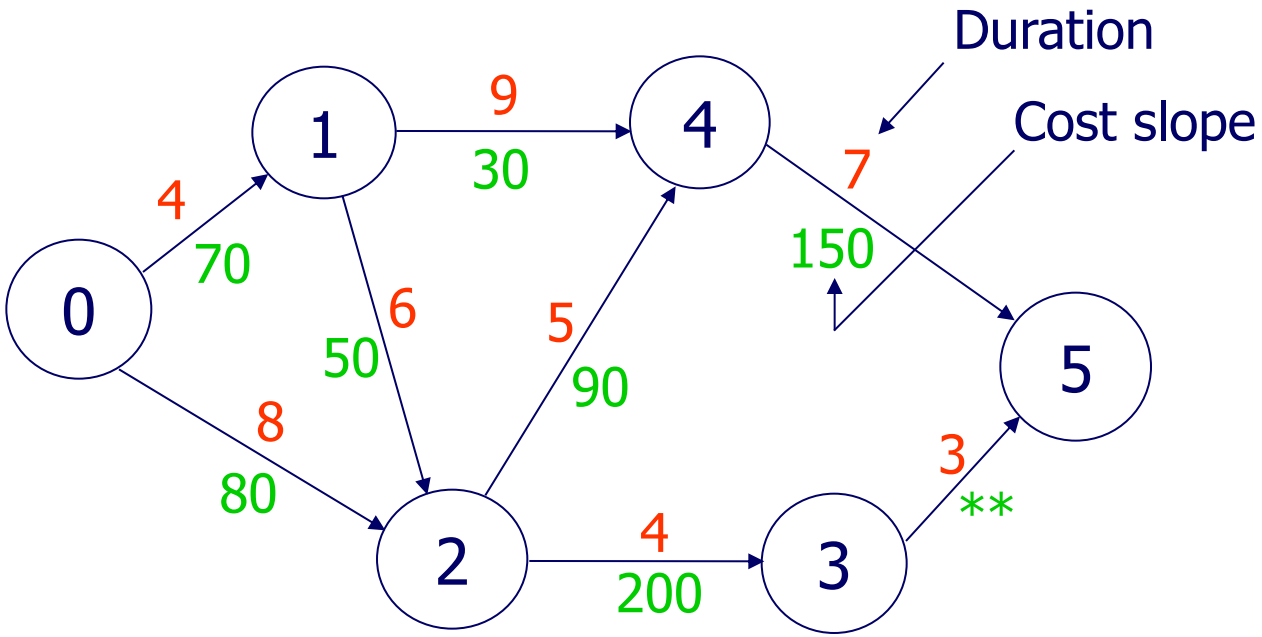
Time-cost tradeoffs (cont.)

- **Project total costs**



Network crashing problem

Consider the following AoA network:



** Activity 3-5 cannot be expedited

The duration of each task is shown above the arrows. The cost slope for each task is shown below the arrows. The initial critical path is 0-1-2-4-5 @ 22 days. The initial cost is $3,050.

Network crashing problem (cont.)

Cost and time data are:

	Normal		Crash		
Activity	**Time**	**Cost**	**Time**	**Cost**	**Cost Slope**
0-1	4 days	$210	3 days	$280	70
0-2	8	400	6	560	80
1-2	6	500	4	600	50
1-4	9	540	7	600	30
2-3	4	500	1	1,100	200
2-4	5	150	4	240	90
3-5	3	150	3	150	**
4-5	7	600	6	750	150
		3,050		4,280	

** Activity 3-5 cannot be expedited

Network crashing problem (cont.)

Develop schedules ranging from 22 to 17 days for this project. Record the critical paths and total costs below:

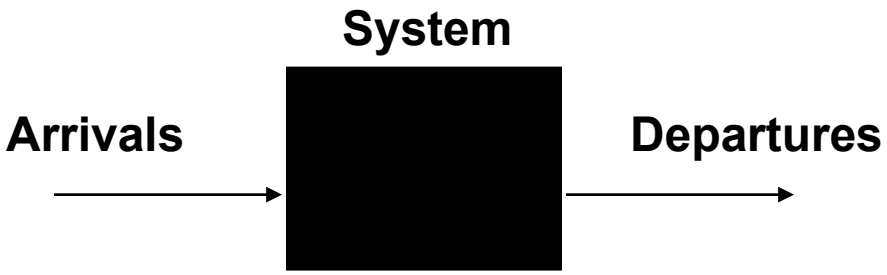
Schedule Duration	Critical Path(s)	Total Cost
22	0-1-2-4-5	$3,100
21		
20		
19		
18		
17		

Queuing Theory

Queuing theory

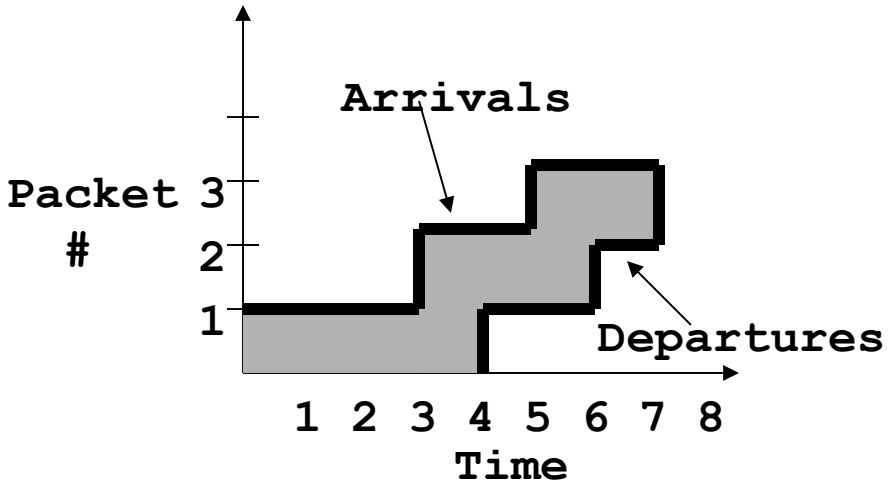
- View network as collections of queues
  - FIFO data-structures
- Queuing theory provides probabilistic analysis of these queues
- Examples:
  - Average length
  - Probability queue is at a certain length
  - Probability a packet will be lost

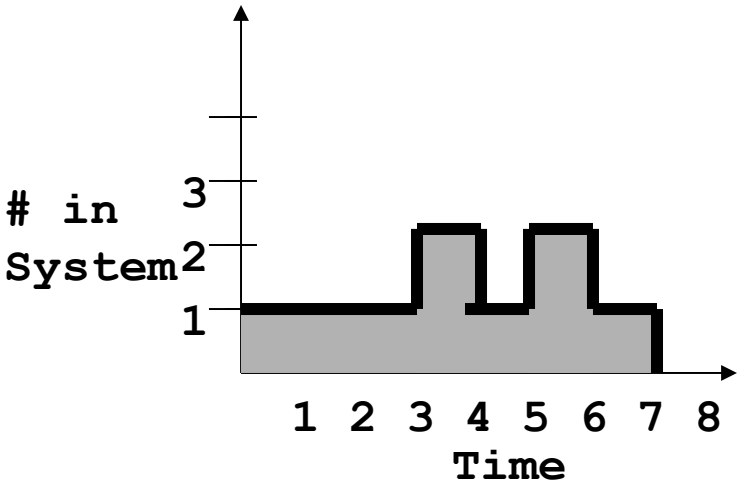
Little's Law

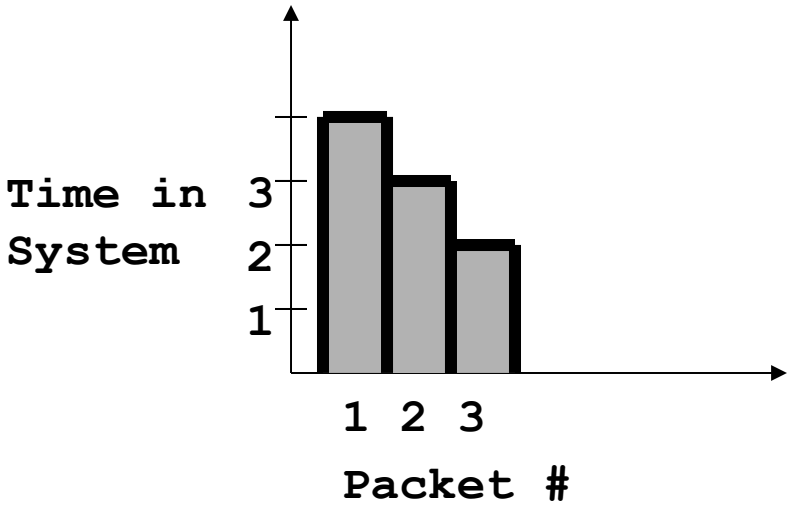


- Little's Law:
  Mean number tasks in system = arrival rate x mean response time
  - Observed before, Little was first to prove
- Applies to any system in equilibrium, as long as nothing in black box is creating or destroying tasks

Proving Little's Law







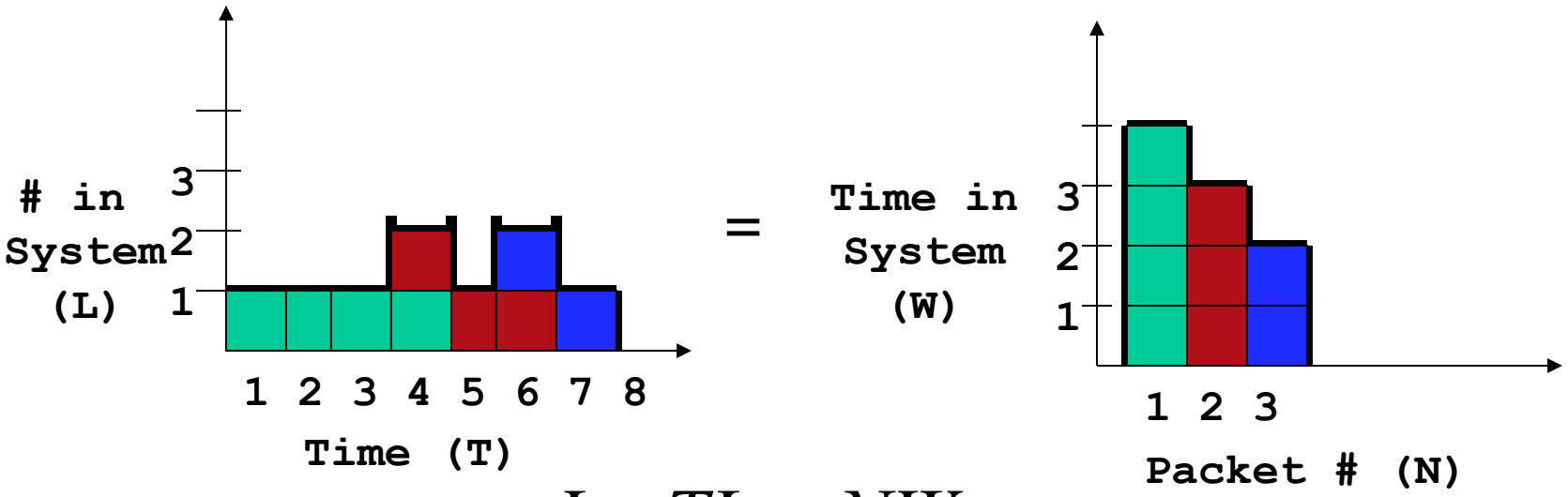
J = Shaded area = 9

Same in all cases!

Definitions

- J: “Area” from previous slide
- N: Number of jobs (packets)
- T: Total time
- $\forall$ λ: Average arrival rate
  - N/T
- W: Average time job is in the system
  - = J/N
- L: Average number of jobs in the system
  - = J/T

Proof: Method 1: Definition



$$J = TL = NW$$

$$L = (\frac{N}{T})W$$

$$L = (\lambda)W$$

Proof: Method 2: Substitution

$$L = (\lambda)W$$

$$L = (\tfrac{N}{T})W$$

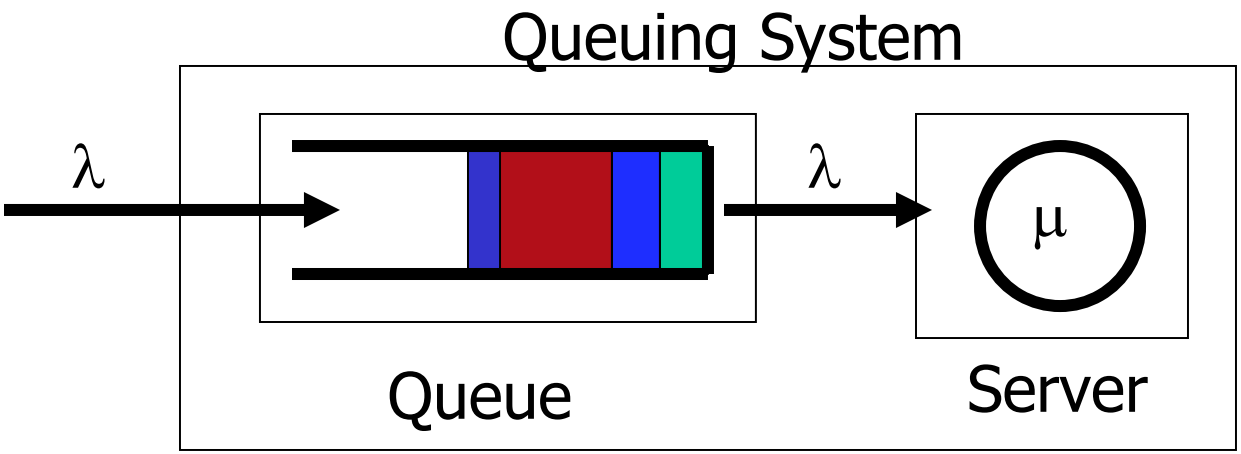
$$\tfrac{J}{T} = (\tfrac{N}{T})(\tfrac{J}{N})$$

$$\tfrac{J}{T} = \tfrac{J}{T}$$ Tautology

Example using Little's law

- Observe 120 cars in front of the Lincoln Tunnel
    - Observe 32 cars/minute depart over a period where no cars in the tunnel at the start or end (e.g. security checks)
- What is average waiting time before and in the tunnel?

Model Queuing System



Queuing System

Server System

Strategy:

Use Little's law on both the complete system and its parts to reason about average time in the queue

Kendal Notation

- Six parameters in shorthand
  - First three typically used, unless specified

1. Arrival Distribution
   - Probability of a new packet arrives in time t
2. Service Distribution
   - Probability distribution packet is serviced in time t
3. Number of servers
4. Total Capacity (infinite if not specified)
5. Population Size (infinite)
6. Service Discipline (FCFS/FIFO)

Distributions

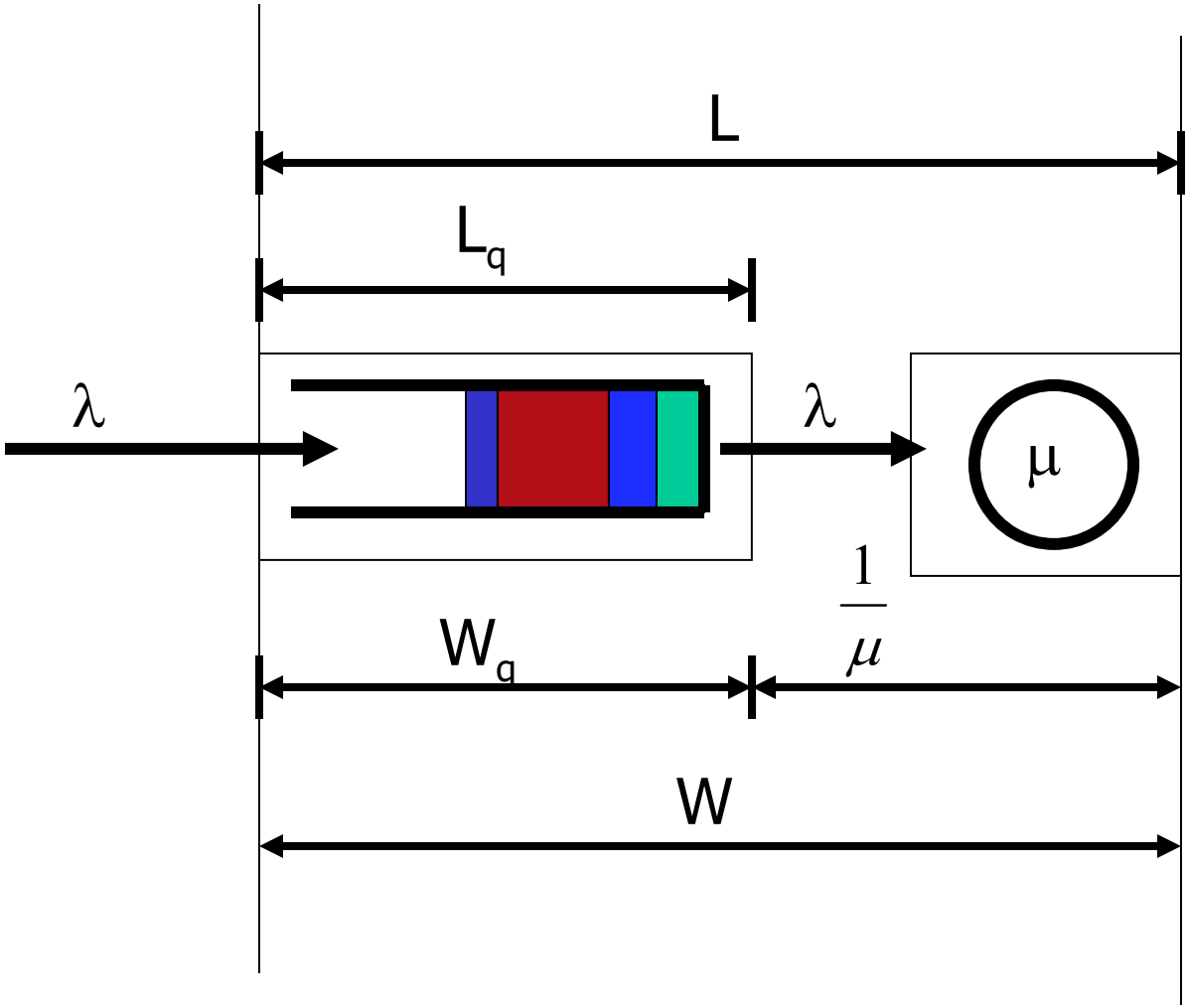
- M: Exponential
- D: Deterministic (e.g. fixed constant)
- E_k: Erlang with parameter *k*
- H_k: Hyperexponential with param. *k*
- G: General (anything)

- M/M/1 is the simplest ‘realistic’ queue

Kendal Notation Examples

- M/M/1:
  - Exponential arrivals and service, 1 server, infinite capacity and population, FCFS (FIFO)
- M/M/m
  - Same, but M servers
- G/G/3/20/1500/SPF
  - General arrival and service distributions, 3 servers, 17 queue slots (20-3), 1500 total jobs, Shortest Packet First

M/M/1 queue model



Analysis of M/M/1 queue

- Goal: A closed form expression of the probability of the number of jobs in the queue (P_i) given only λ and μ

Solving queuing systems

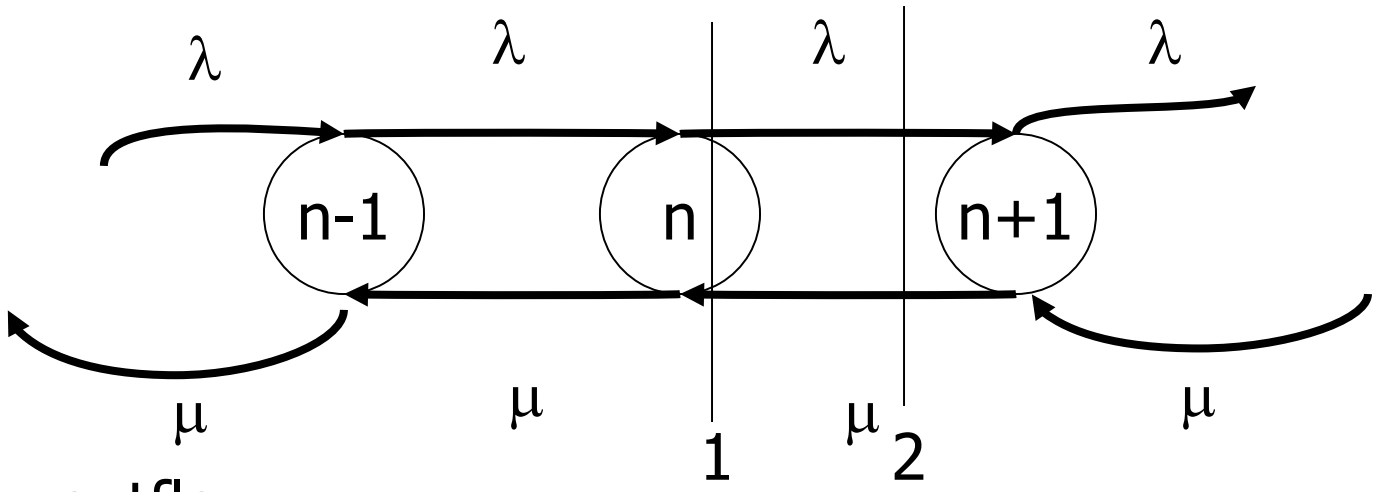
- Given:
  - λ: Arrival rate of jobs (packets on input link)
  - μ: Service rate of the server (output link)
- Solve:
  - L: average number in queuing system
  - L_q average number in the queue
  - W: average waiting time in whole system
  - W_q average waiting time in the queue
- 4 unknown's: need 4 equations

Solving queuing systems

- 4 unknowns: L, L_q W, W_q
- Relationships using Little's law:
  - $L=\lambda W$
  - $L_q=\lambda W_q$ (steady-state argument)
  - $W = W_q + (1/\mu)$
- If we know any 1, can find the others
- Finding L is hard or easy depending on the type of system. In general:

$$L = \sum_{n=0}^{\infty} nP_n$$

Equilibrium conditions



inflow = outflow

1: $(\lambda + \mu)P_n = \lambda P_{n-1} + \mu P_{n+1}$

2: $\lambda P_n = \mu P_{n+1}$

stability: 3: $\lambda \le \mu, \rho = \frac{\lambda}{\mu}, \rho \le 1$

Solving for P_0 and P_n

1: $P_1 = \rho P_0 \ , \ P_2 = (\rho)^2 P_0 \ , \ P_n = (\rho)^n P_0$

2: $\sum_{n=0}^{\infty} P_n = 1 \ , \ P_0 \sum_{n=0}^{\infty} \rho^n = 1 \ , \ P_0 = \frac{1}{\sum_{n=0}^{\infty} \rho^n}$,

3: $\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho}, \rho < 1$ (geometric series)

4: $P_0 = \frac{1}{\sum_{n=0}^{\infty} \rho^n} = \frac{1}{\frac{1}{(1-\rho)}} = 1-\rho$ 5: $P_n = (\rho)^n (1-\rho)$

Solving for L

$$L = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n\rho^n(1-\rho) = (1-\rho)\rho\sum_{n=1}^{\infty} n\rho^{n-1}$$

$$(1-\rho)\rho\frac{d}{d\rho}\left(\sum_{n=0}^{1}\rho^n\right) = (1-\rho)\rho\frac{d}{d\rho}\left(\frac{1}{1-\rho}\right)$$

$$(1-\rho)\rho\left(\frac{1}{(1-\rho)^2}\right) = \frac{\rho}{(1-\rho)} = \frac{\lambda}{\mu-\lambda}$$

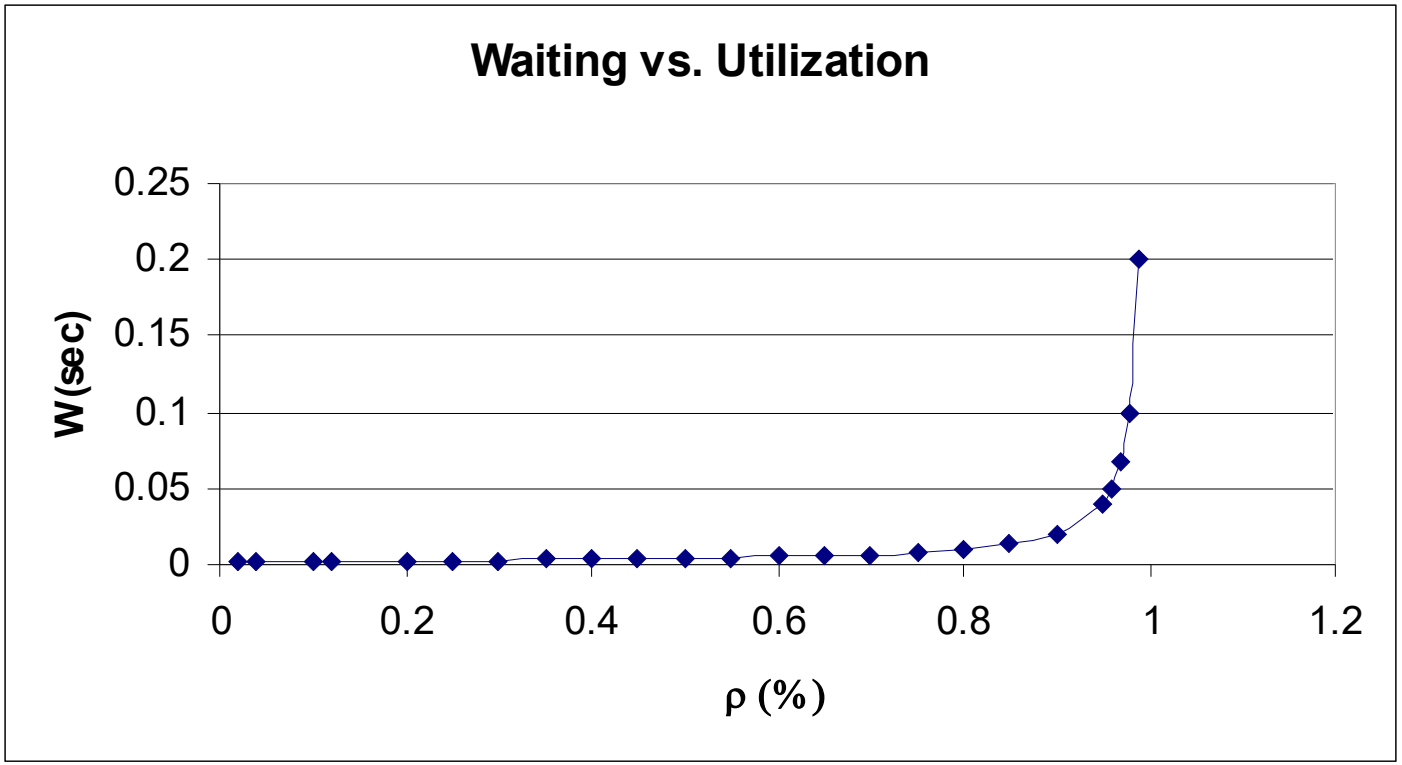
Solving W, W_q and L_q

$$W = \frac{L}{\lambda} = \left(\frac{\lambda}{\mu-\lambda}\right)\left(\frac{1}{\lambda}\right) = \frac{1}{\mu-\lambda}$$

$$W_q = W - \frac{1}{\mu} = \left(\frac{\lambda}{\mu-\lambda}\right) - \left(\frac{1}{\mu}\right) = \frac{\lambda}{\mu(\mu-\lambda)}$$

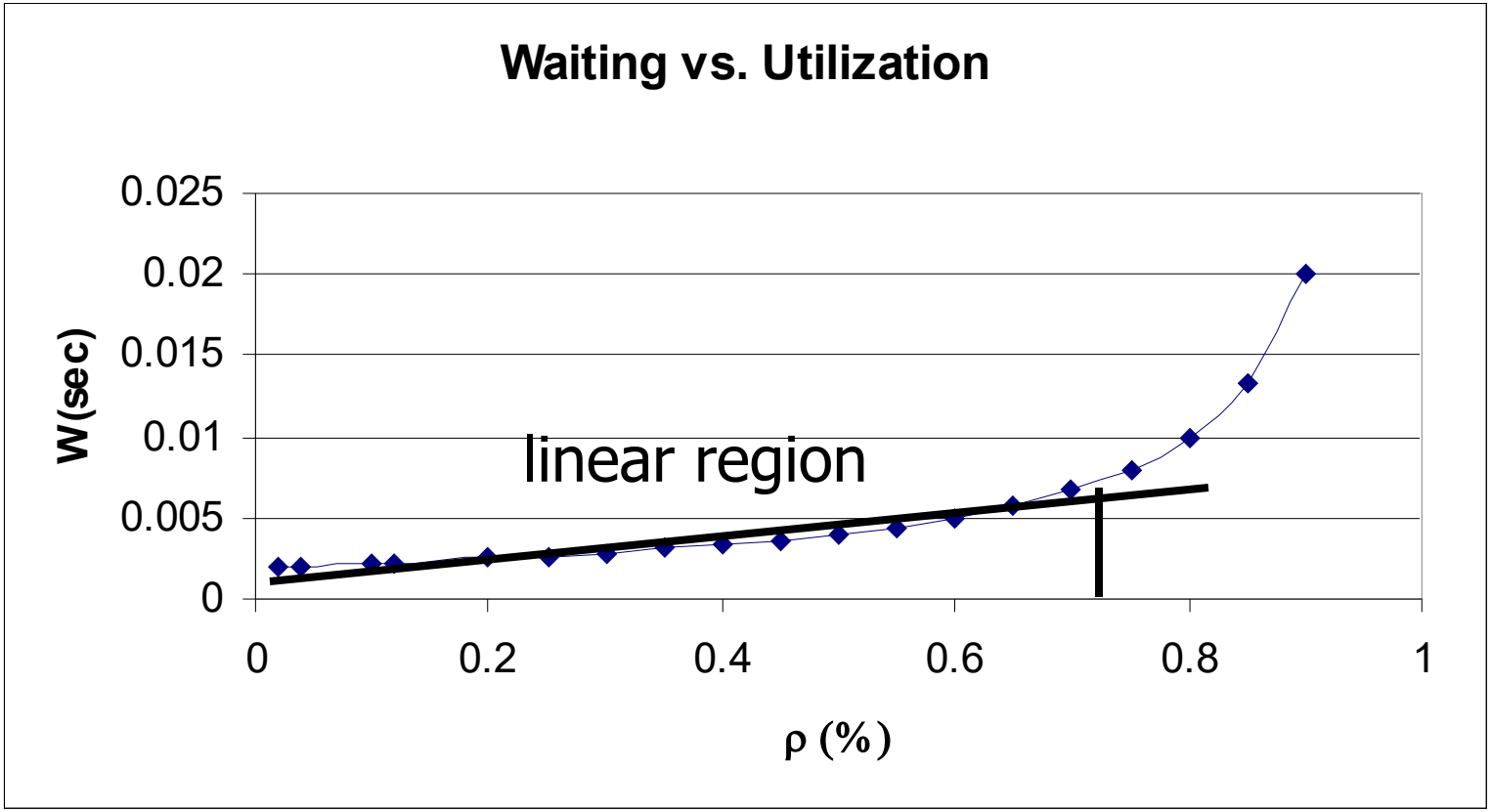
$$L_q = \lambda W_q = \lambda \frac{\lambda}{\mu(\mu-\lambda)} = \frac{\lambda^2}{\mu(\mu-\lambda)}$$

Response Time vs. Arrivals

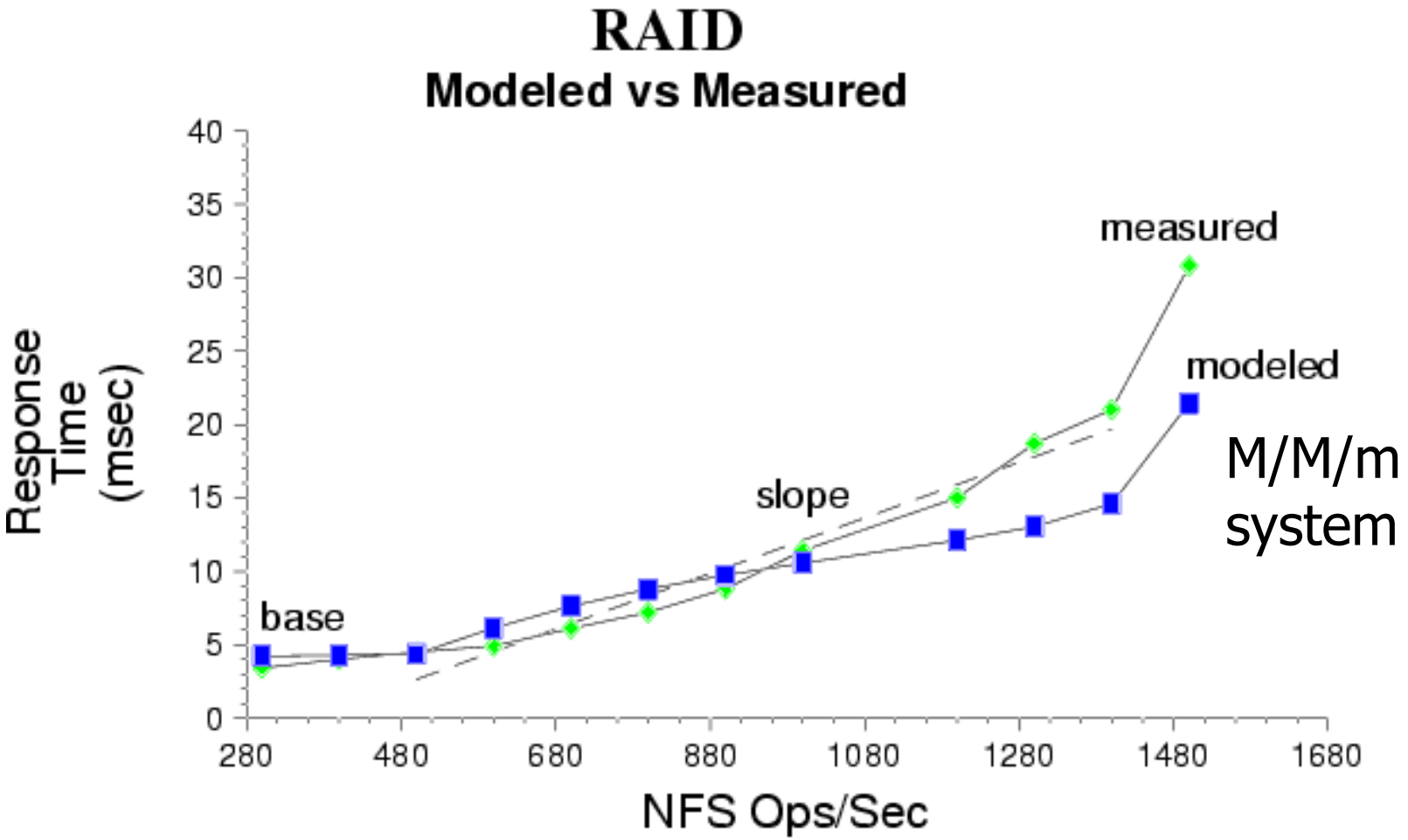


$$W = \frac{1}{\mu - \lambda}$$

Stable Region



Empirical Example



Example

- Measurement of a network gateway:
  - mean arrival rate (λ): 125 Packets/s
  - mean response time per packet: 2 ms
- Assuming exponential arrivals & departures:
  - What is the service rate, μ ?
  - What is the gateway's utilization?
  - What is the probability of n packets in the gateway?
  - mean number of packets in the gateway?
  - The number of buffers so P(overflow) is $<10^{-6}$?

Example (cont)

The service rate, $\mu = \frac{1}{0.002} = 500\,pps$

utilization = $\rho = (\frac{\lambda}{\mu}) = 0.25\%$

P(n) packets in the gateway =

$$P_0 P_n = (1-\rho)(\rho^n) = (0.75)(0.25^n)$$

Example (cont)

Mean # in gateway (L) =

$$\frac{\rho}{1-\rho} = \frac{0.25}{1-0.25} = 0.33$$

to limit loss probability to less than 1 in a million:

$$\rho^n \leq 10^{-6}$$

Properties of a Poisson processes

- Poisson process = exponential distribution between arrivals/departures/service

$$P(\text{arrival} < t) = 1 - e^{-\lambda t}$$

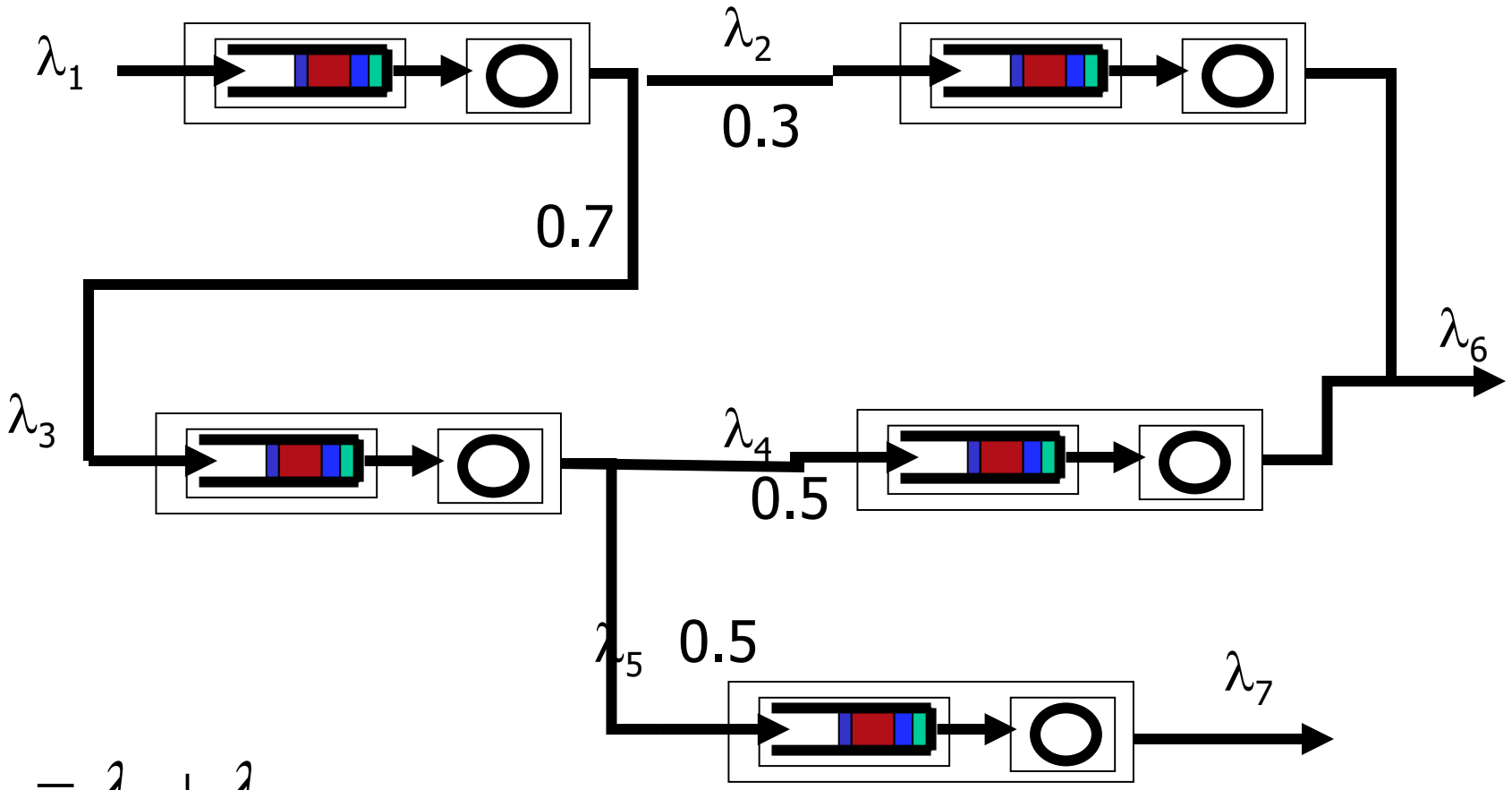
- Key properties:
  - memoryless
    - Past state does not help predict next arrival
  - Closed under:
    - Addition
    - Subtraction

Addition and Subtraction

- Merge:
  - two poisson streams with arrival rates λ_1 and λ_2:
    - new poisson stream: $\lambda_3=\lambda_1+\lambda_2$
- Split :
  - If any given item has a probability P_1 of "leaving" the stream with rate $\lambda_{1:}$

    $\forall\ \lambda_2=(1-P_1)\lambda_1$

Queuing Networks



$$\lambda_6 = \lambda_2 + \lambda_4$$

$$\lambda_7 = \lambda_5$$

Bridging Router Performance and Queuing Theory

Sigmetrics 2004

Slides by N. Hohn*, D. Veitch*, K. Papagiannaki, C. Diot

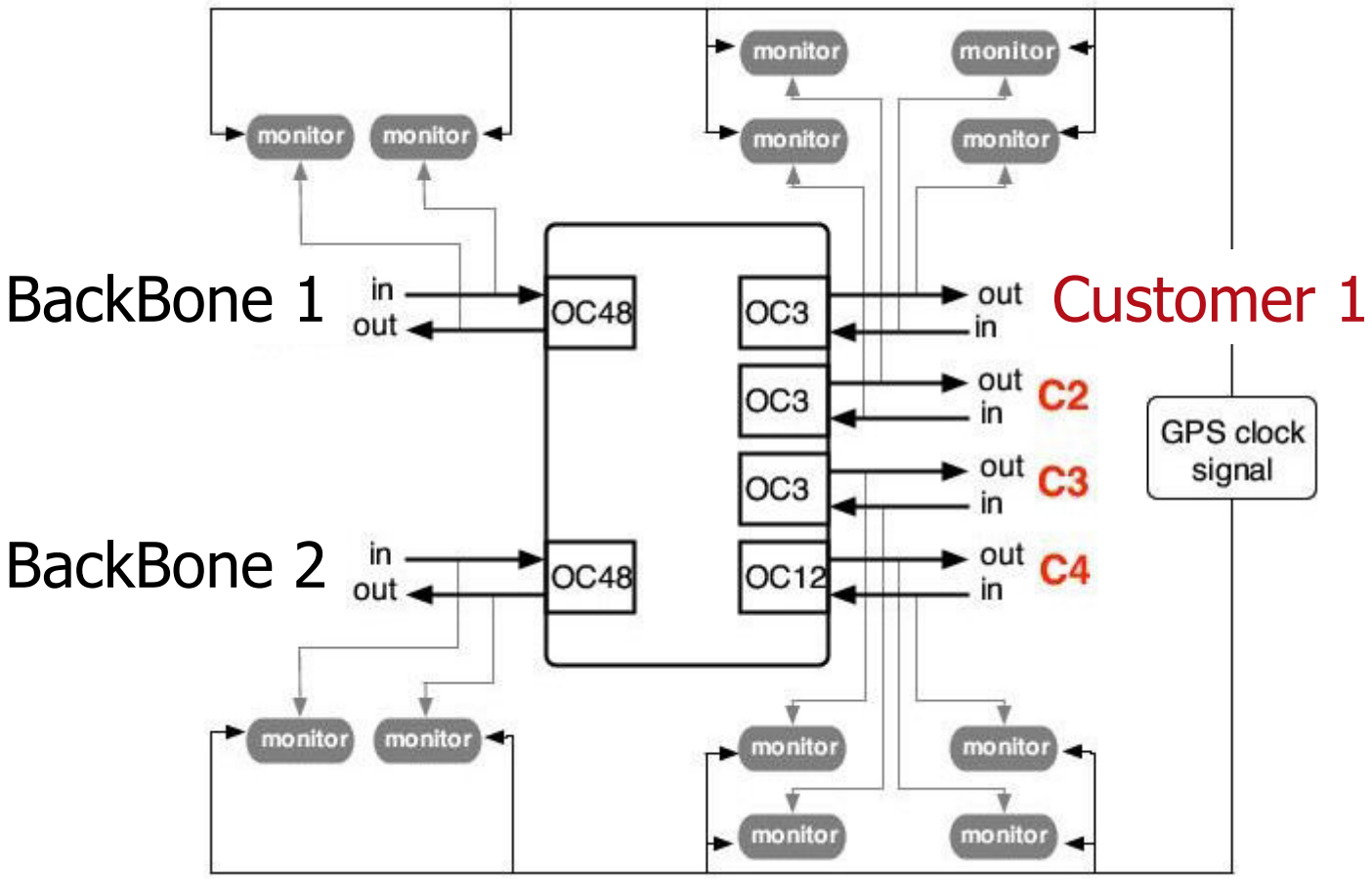
Motivation

- End-to-end packet delay is an important metric for performance and Service Level Agreements (SLAs)
- Building block of end-to-end delay is through router delay
- Measure the delays incurred by *all* packets crossing a single router

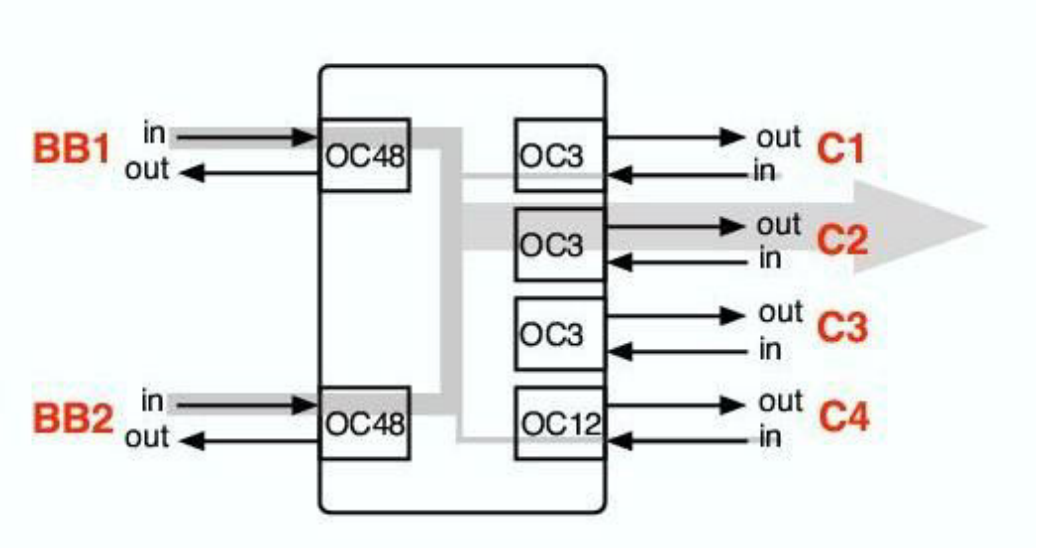
Overview

- Full Router Monitoring
- Delay Analysis and Modeling
- Delay Performance: Understanding and Reporting

Measurement Environment



Packet matching



Set	Link	Matched pkts	% traffic C2-out
C4	In	215987	0.03%
C1	In	70376	0.01%
BB1	In	345796622	47.00%
BB2	In	389153772	52.89%
C2	out	735236757	**99.93**%

Overview

Full Router Monitoring

- Delay Analysis and Modeling

Delay Performance: Understanding and Reporting

Definition of delay

Store & Forward Datapath

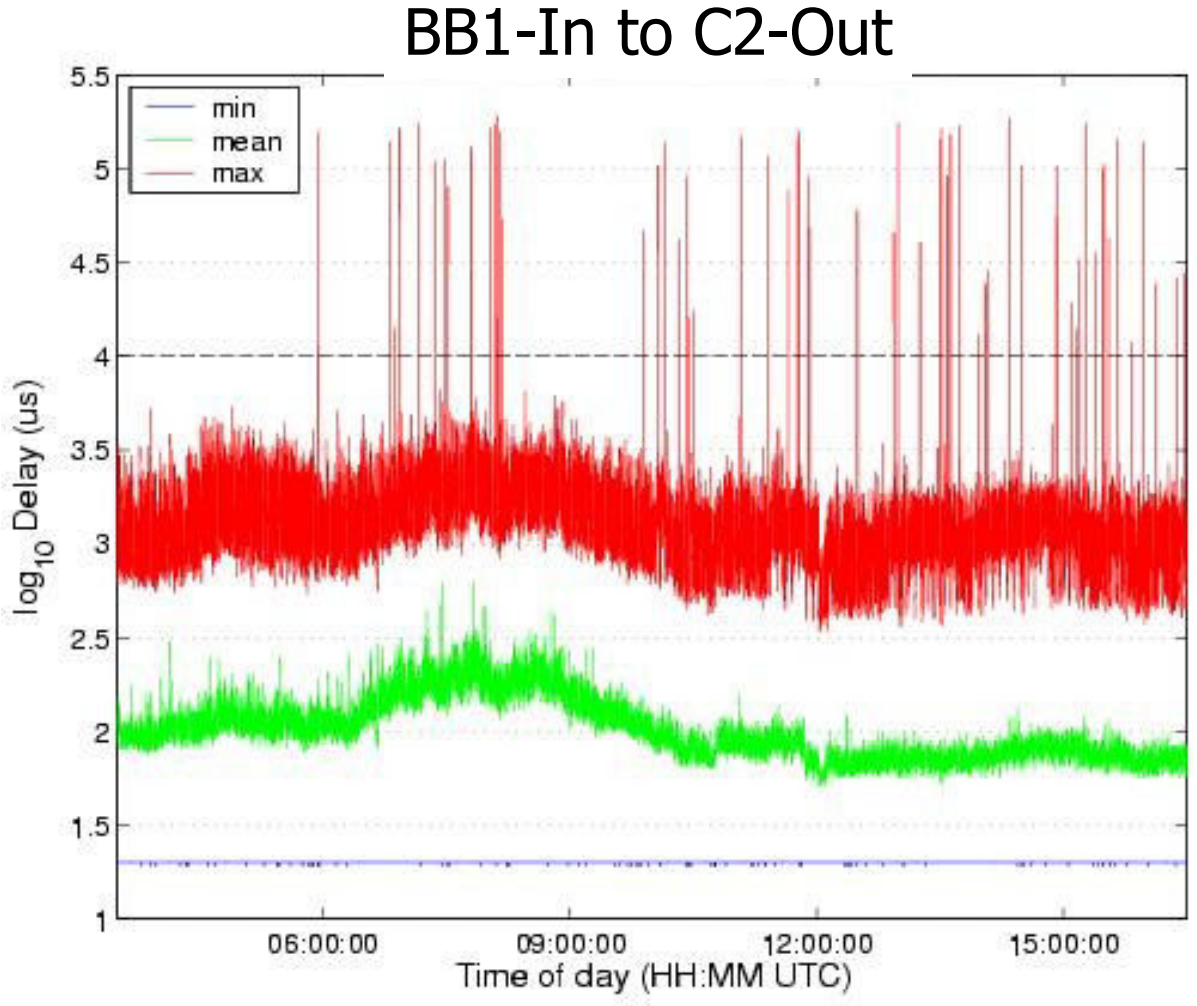
- Store: storage in input linecard's memory ← **Not part of the system**
- Forwarding decision
- Storage in dedicated Virtual Output Queue (VOQ)
- Decomposition into fixed-size cells
- Transmission through switch fabric cell by cell
- Packet reconstruction
- Forward: Output link scheduler

Delays: 1 minute summary

MAX

Mean

MIN

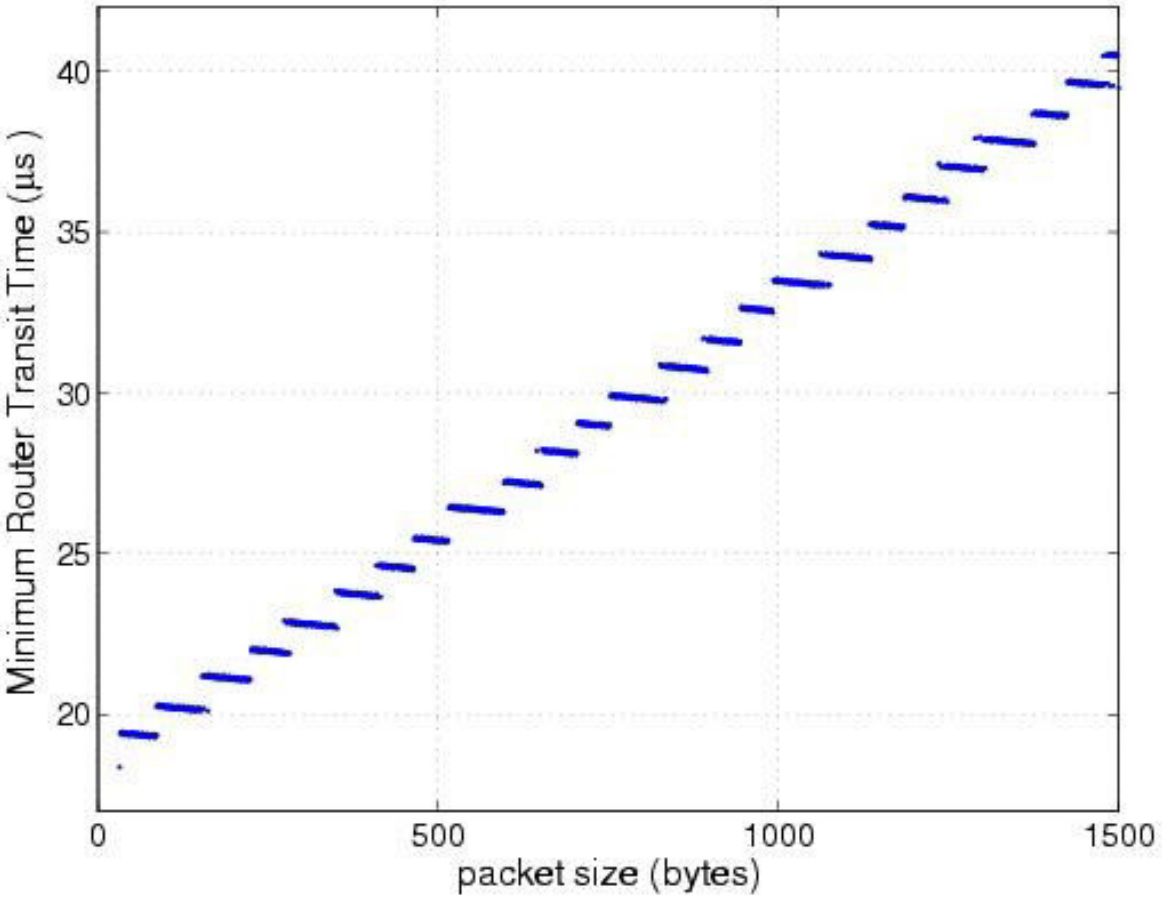


Store & Forward Datapath

- Store: storage in input linecard's memory ← **Not part of the system**
- Forwarding decision
- Storage in dedicated Virtual Output Queue (VOQ)
- Decomposition into fixed-size cells
- Transmission through switch fabric cell by cell
- Packet reconstruction
- Forward: Output link scheduler

$\Delta\lambda_i\Lambda_j(L)$

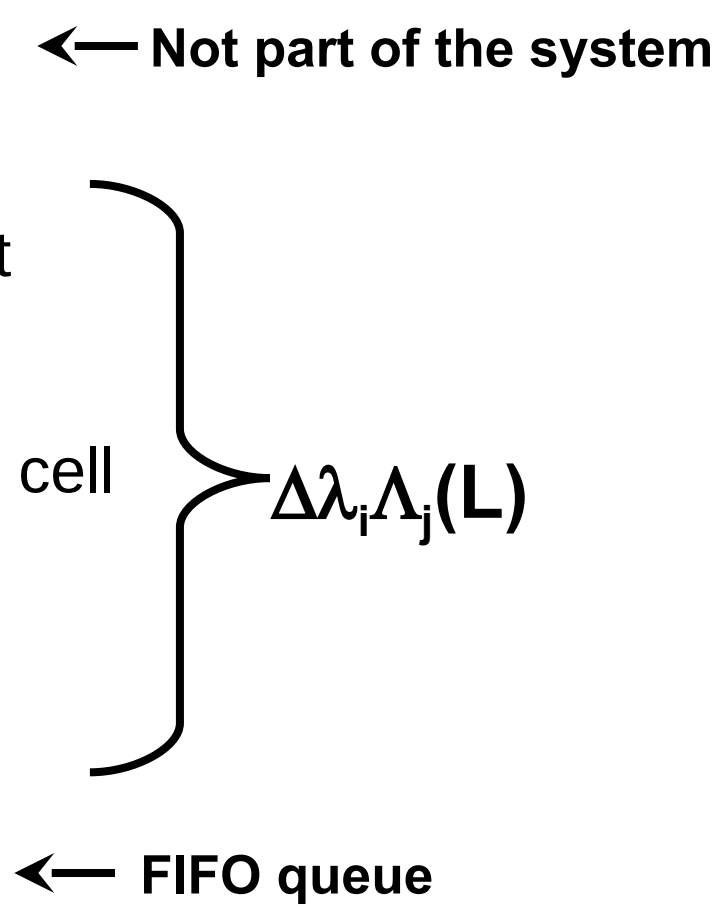
Minimum Transit Time



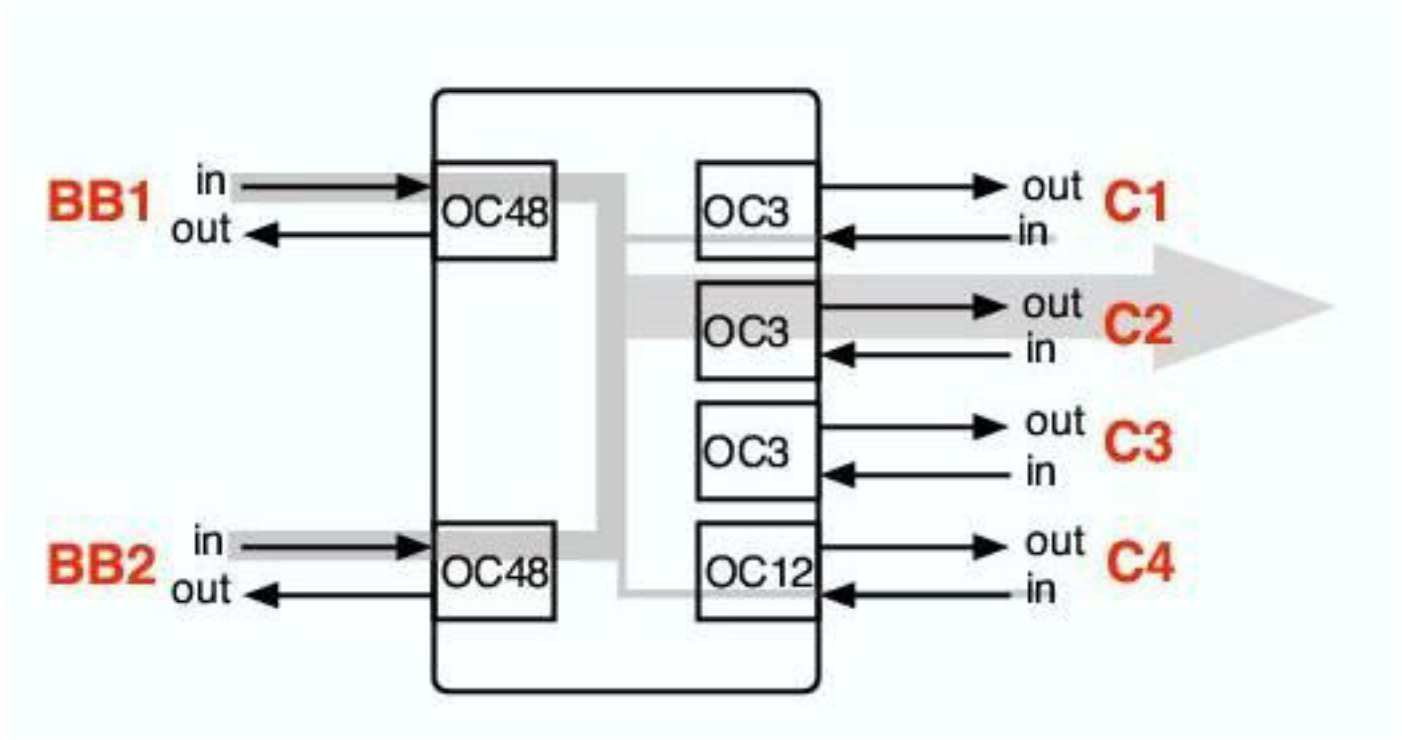
Packet size dependent minimum delay.

Store & Forward Datapath

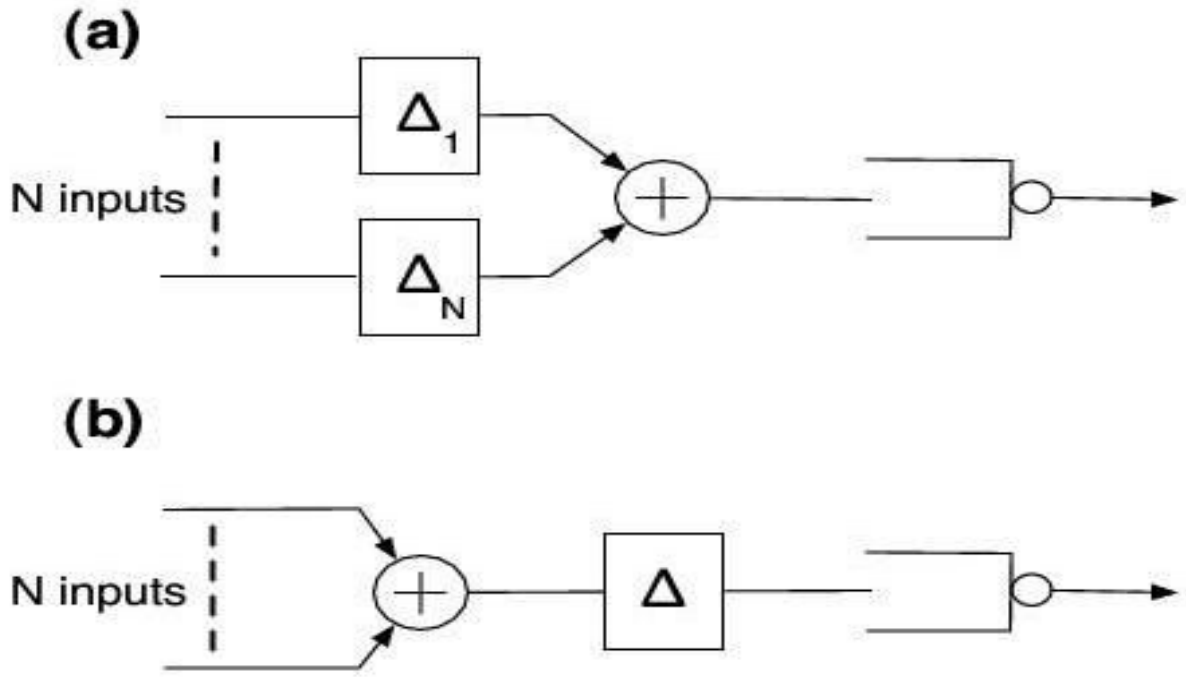
- Store: storage in input linecard's memory
- Forwarding decision
- Storage in dedicated Virtual Output Queue (VOQ)
- Decomposition into fixed-size cells
- Transmission through switch fabric cell by cell
- Packet reconstruction
- Forward: Output link scheduler



Modeling

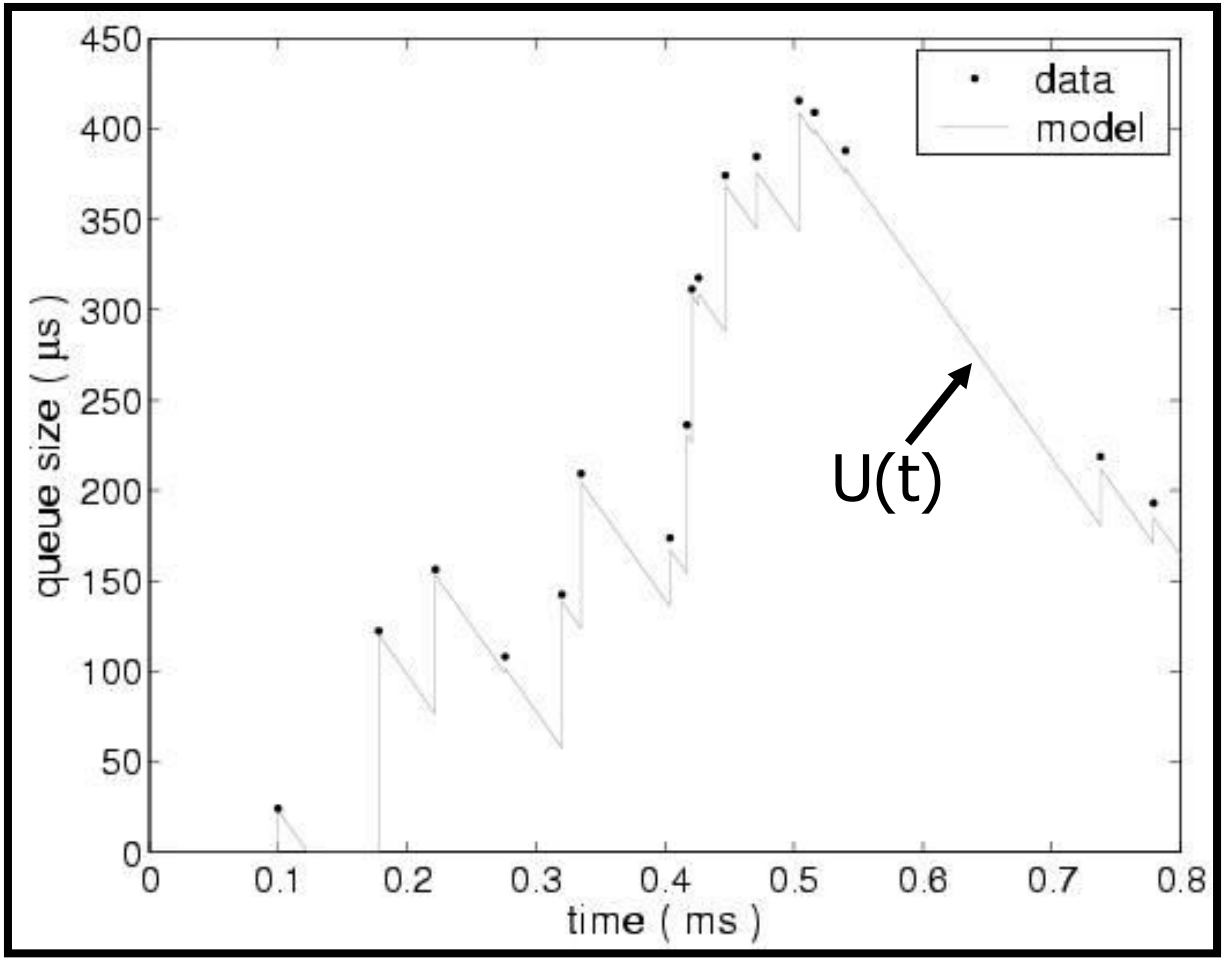

BB1
in
out
OC48
BB2
in
out
OC48
OC3
OC3
OC3
OC12
out
in
C1
out
in
C2
out
in
C3
out
in
C4

Modeling

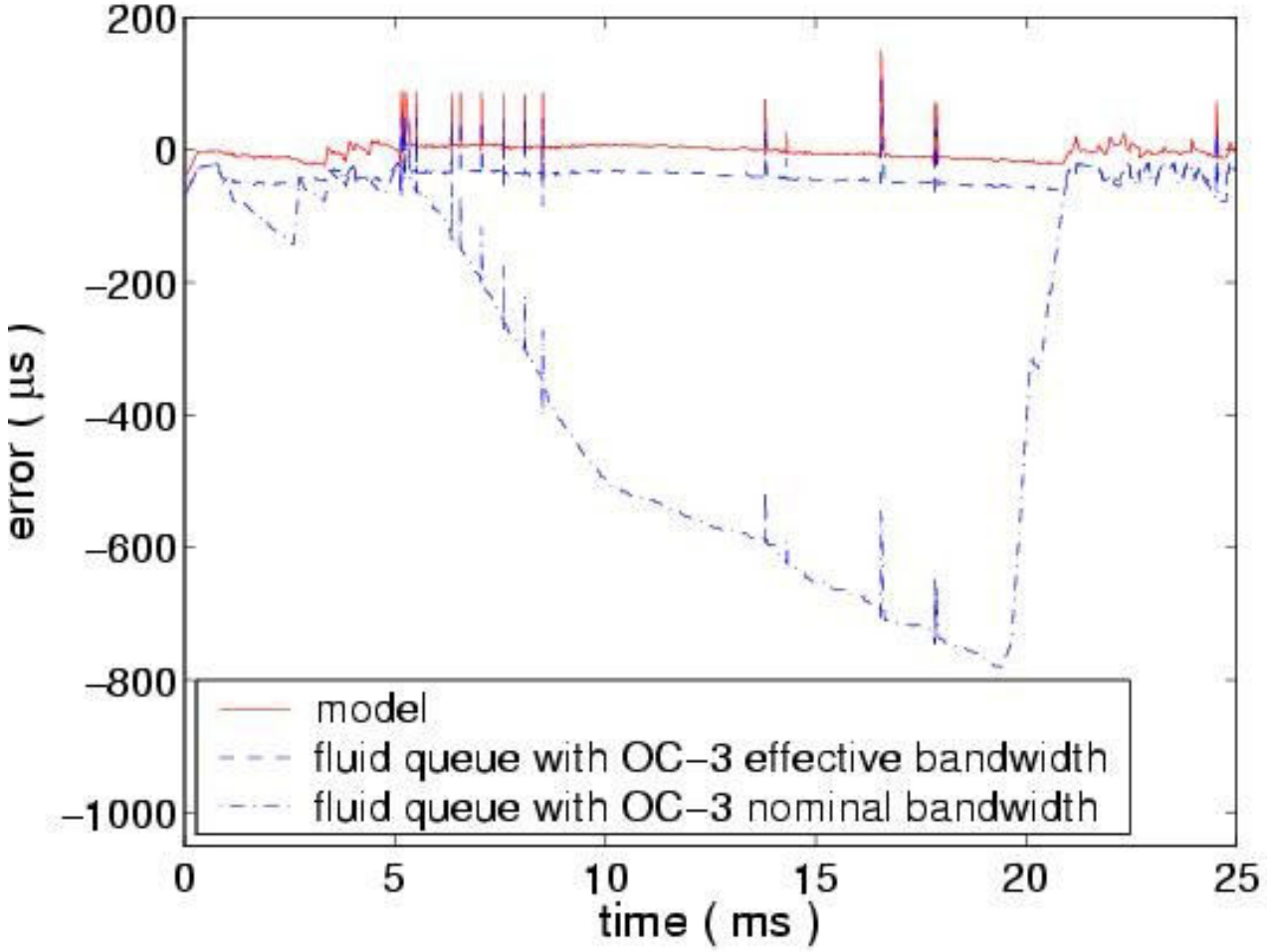


Fluid queue with a delay element at the front

Model Validation



Error as a function of time



Modeling results

- A crude model performs well!
  - As simpler/simpler than an M/M/1 queue
- Use effective link bandwidth
  - account for encapsulation
- Small gap between router performance and queuing theory!
- The model defines Busy Periods: time between the arrival of a packet to the empty system and the time when the system becomes empty again.

Overview

Full Router Monitoring

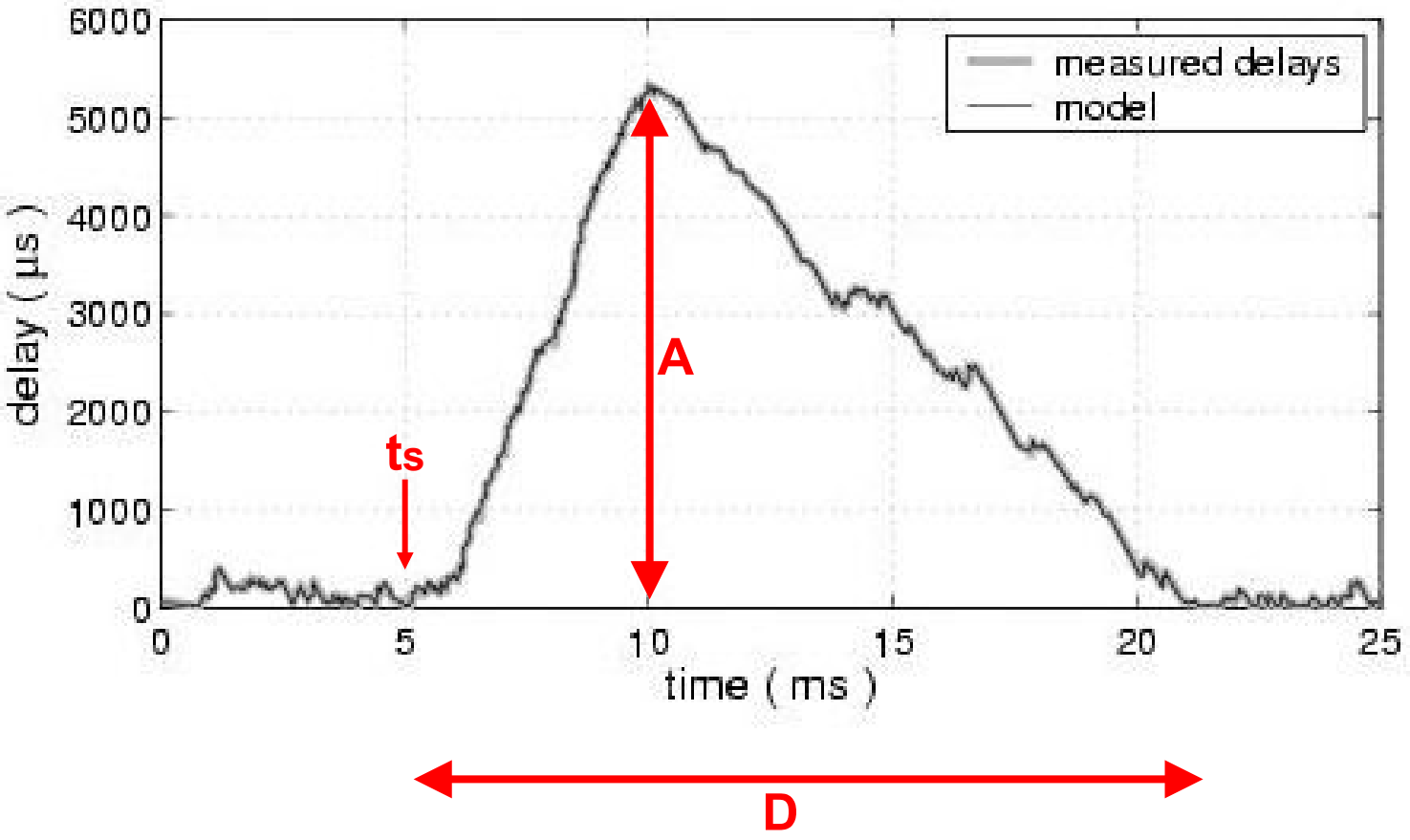
Delay Analysis and Modeling

- Delay Performance: Understanding and Reporting

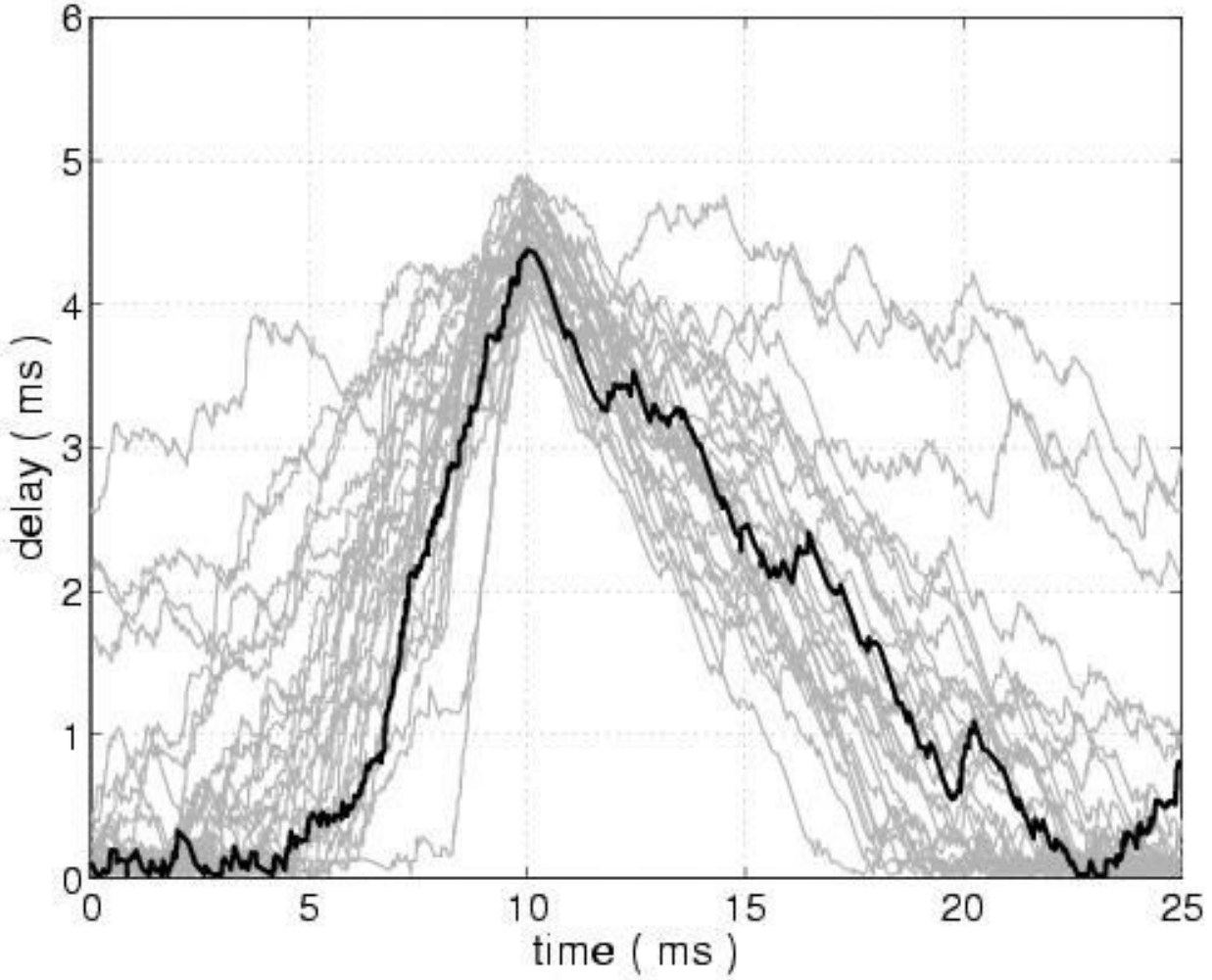
On the Delay Performance

- Model allows for router performance evaluation when arrival patterns are known
- Goal: metrics that
  - Capture operational-router performance
  - Can answer performance questions directly
- Busy Period structures contain *all delay* information
  - BP better than utilization or delay reporting

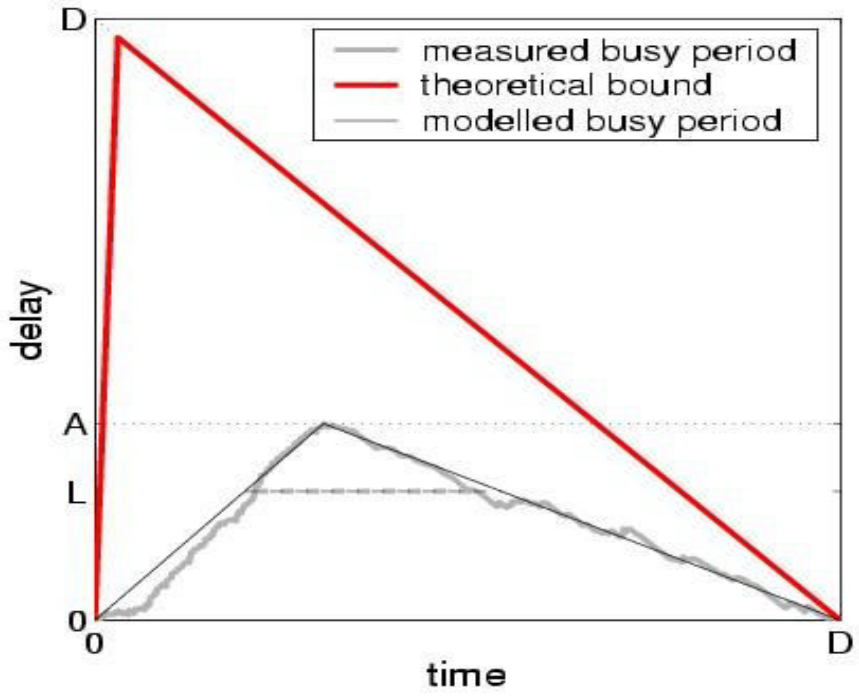
Busy periods metrics



Property of significant BPs



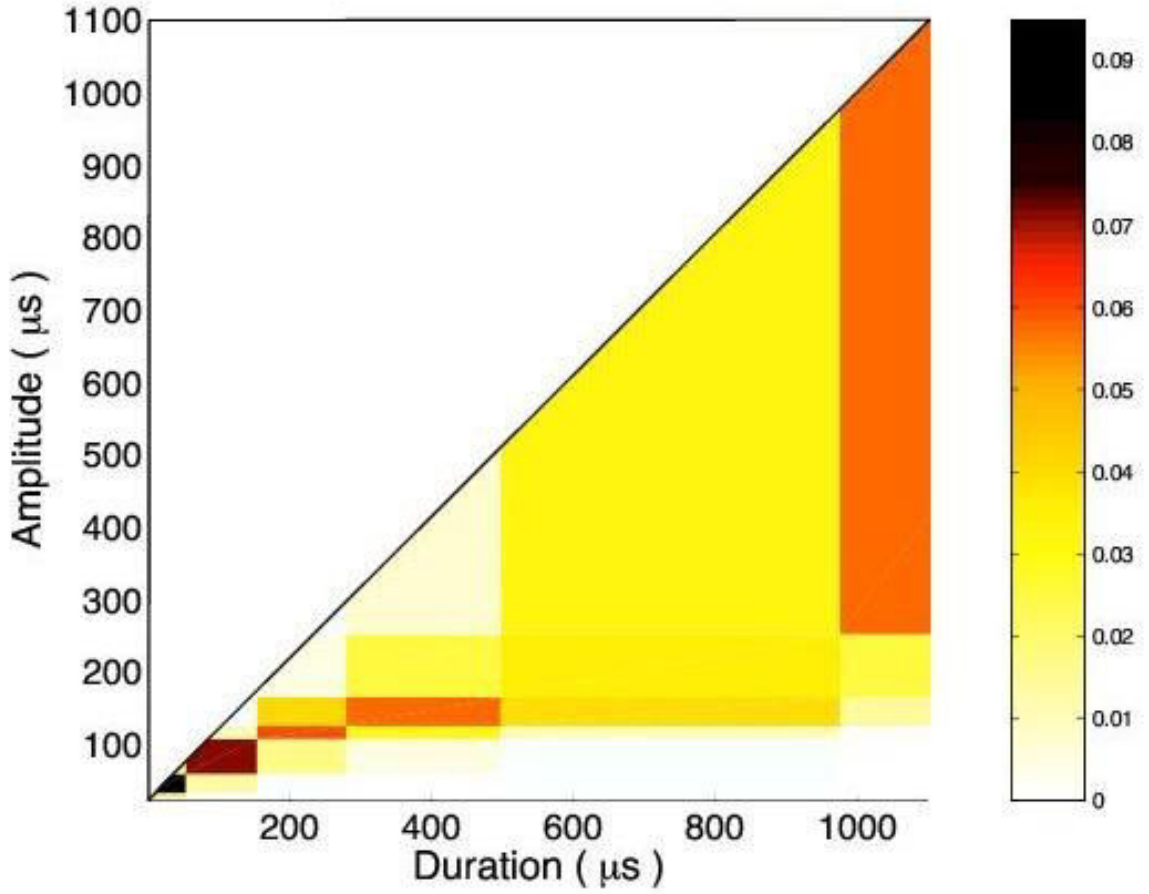
Triangular Model



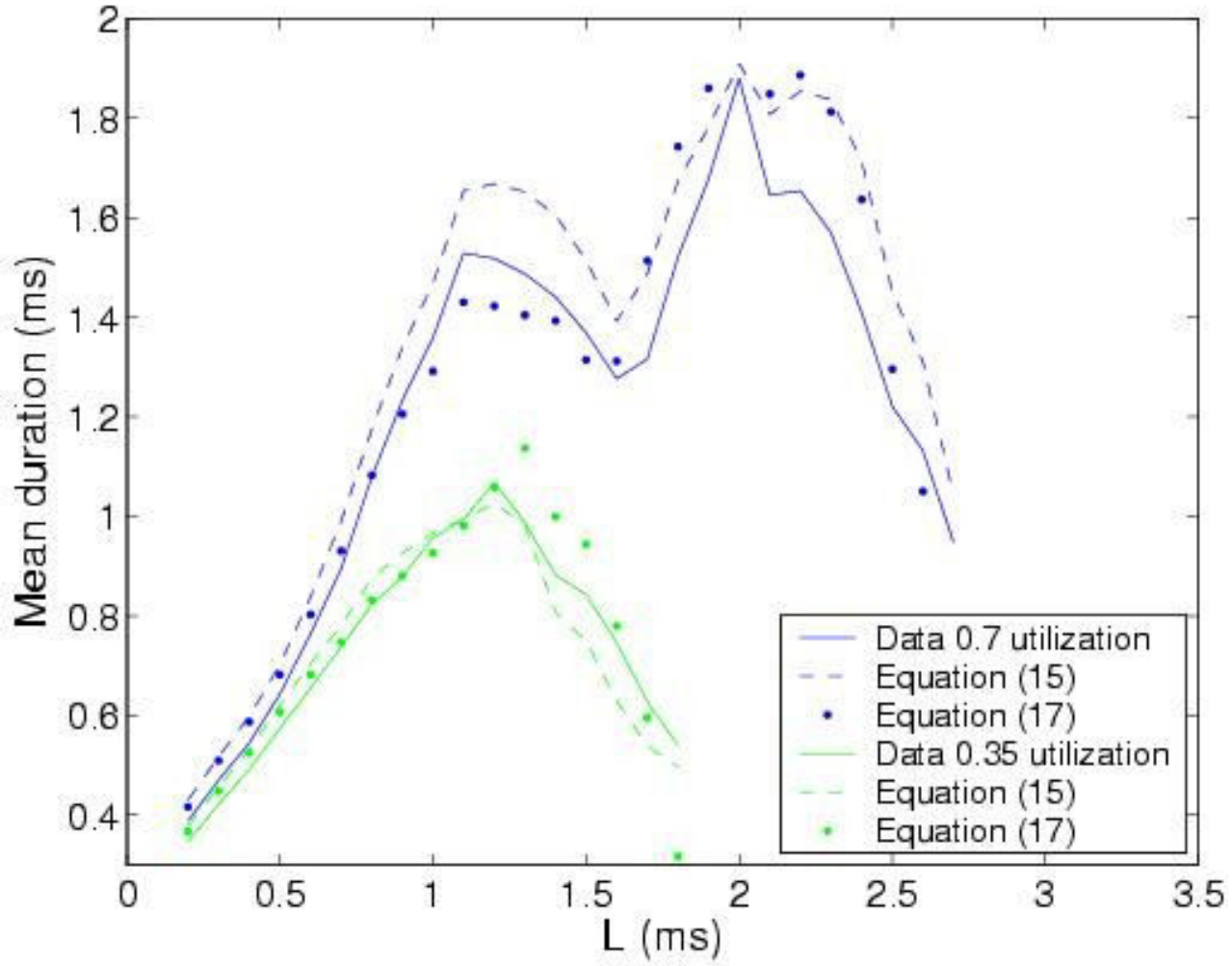
Issues

- Report (A,D) measurements
- There are millions of busy periods even on a lightly utilized router
- Interesting episodes are rare and last for a very small amount of time

Report BP joint distribution



Duration of Congestion Level-L



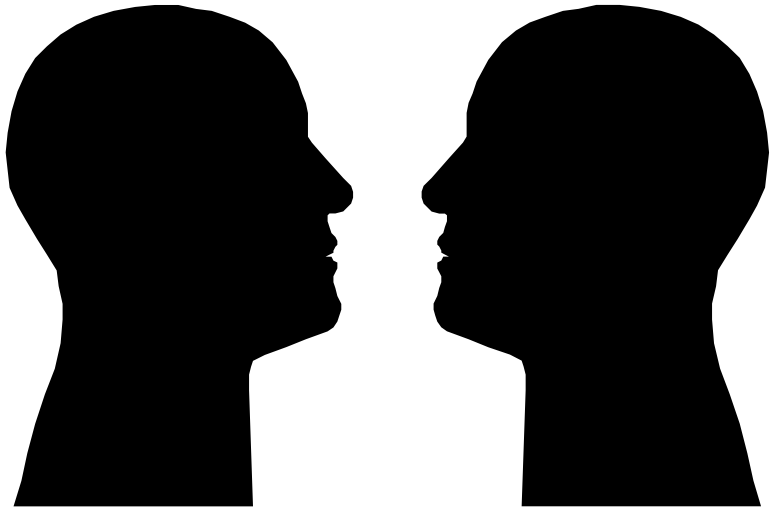
Conclusions

- Results
  - Full router empirical study
  - Delay modeling
  - Reporting performance metrics
- Future work
  - Fine tune reporting scheme
  - Empirical evidence of large deviations theory

Network Traffic Self-Similarity

Slides by Carey Williamson

Department of Computer Science
University of Saskatchewan



Introduction

- A classic measurement study has shown that aggregate Ethernet LAN traffic is self-similar [Leland et al 1993]
- A statistical property that is very different from the traditional Poisson-based models
- This presentation: definition of network traffic self-similarity, Bellcore Ethernet LAN data, implications of self-similarity

Measurement Methodology

- Collected lengthy traces of Ethernet LAN traffic on Ethernet LAN(s) at Bellcore
- High resolution time stamps
- Analyzed statistical properties of the resulting time series data
- Each observation represents the number of packets (or bytes) observed per time interval (e.g., 10 4 8 12 7 2 0 5 17 9 8 8 2...)

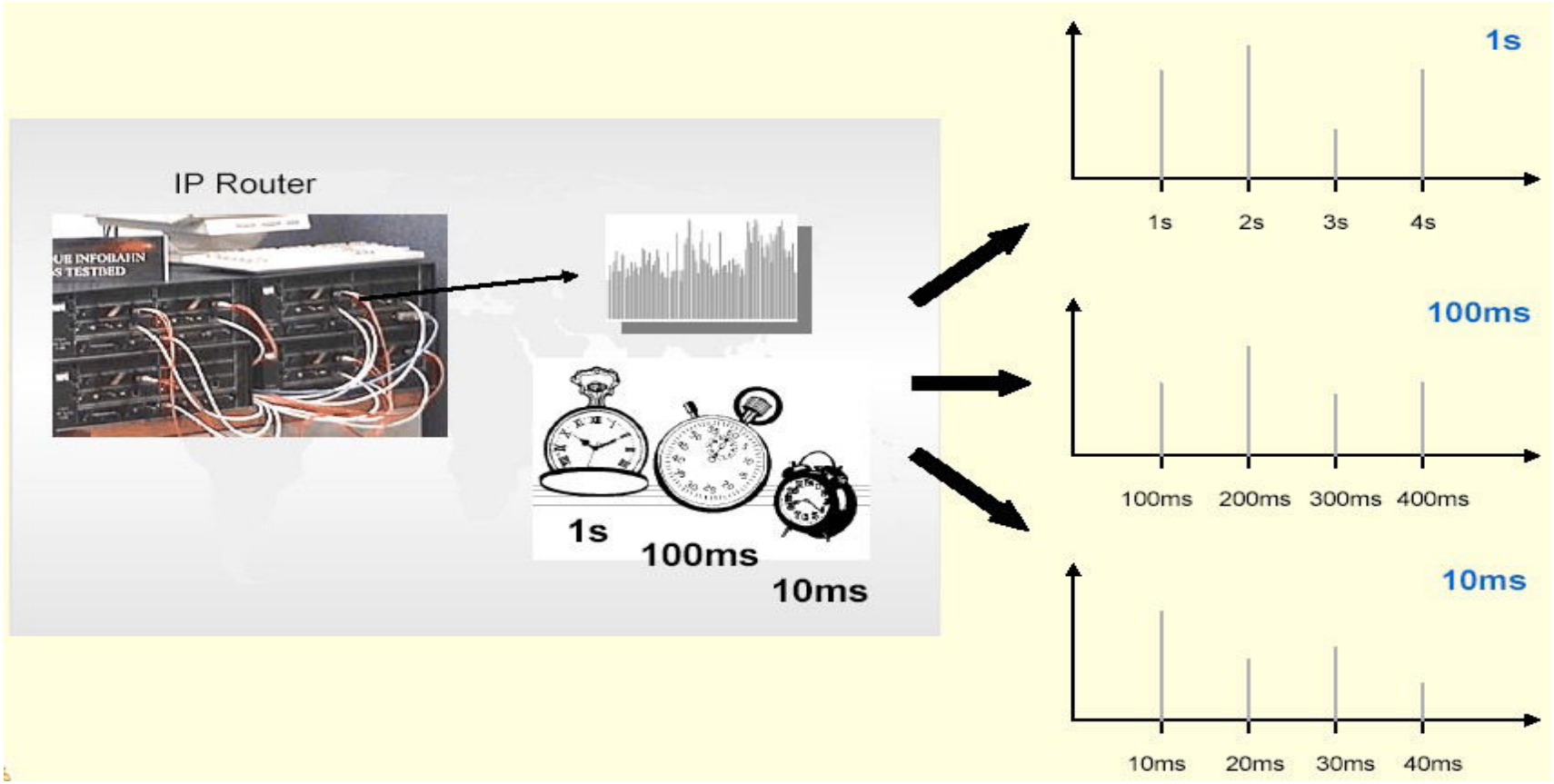
Self-Similarity: The intuition

- If you plot the number of packets observed per time interval as a function of time, then the plot looks ‘‘the same’’ regardless of what interval size you choose
- E.g., 10 msec, 100 msec, 1 sec, 10 sec,...
- Same applies if you plot number of bytes observed per interval of time

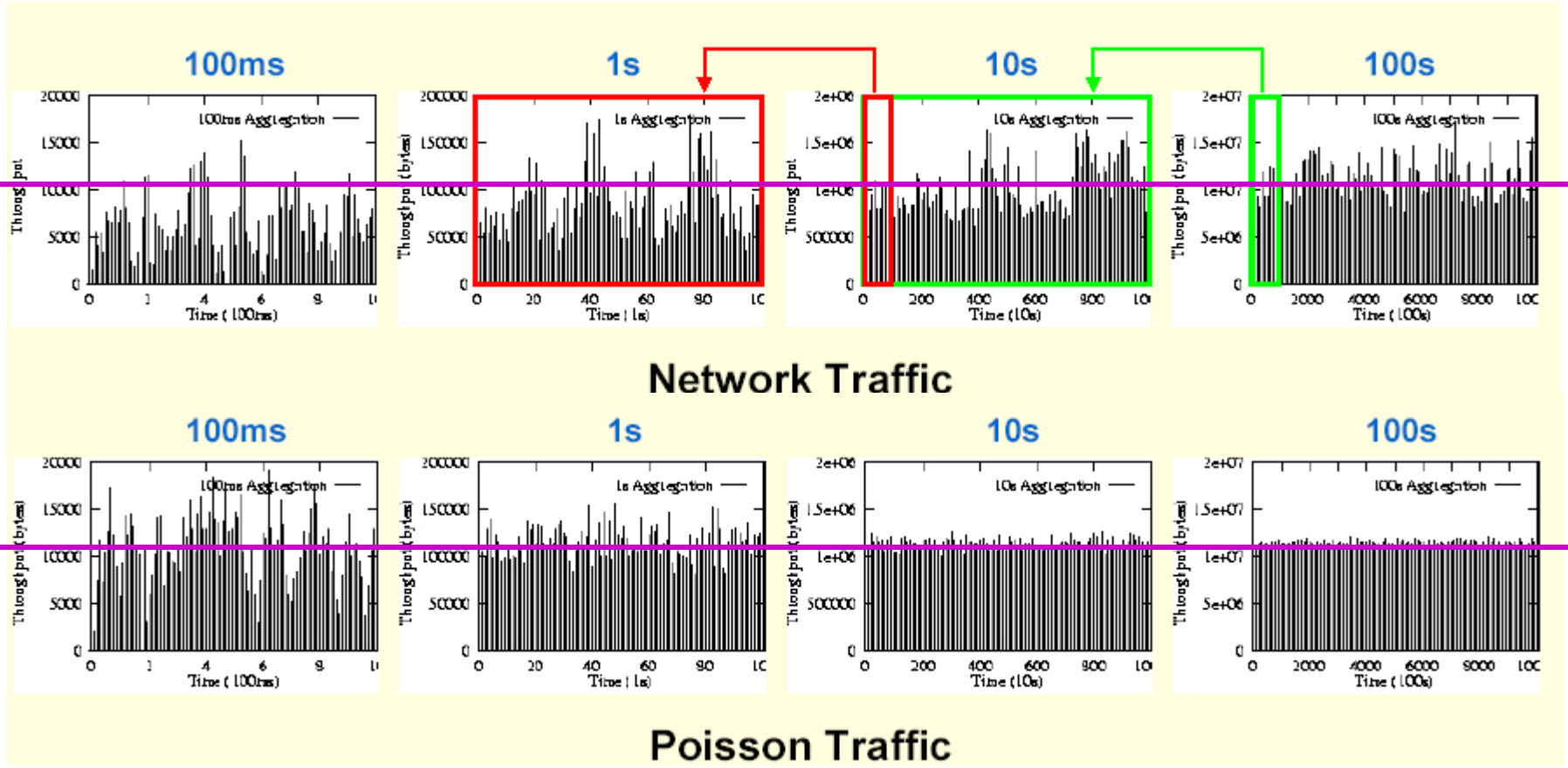
Self-Similarity: The Intuition

- In other words, self-similarity implies a ‘‘fractal-like’’ behavior: no matter what time scale you use to examine the data, you see similar patterns
- Implications:
  - Burstiness exists across many time scales
  - No natural length of a burst
  - Key: Traffic does not necessarily get ‘‘smoother’’ when you aggregate it (unlike Poisson traffic)

Self-Similarity Traffic Intuition (I)



Self-Similarity in Traffic Measurement II

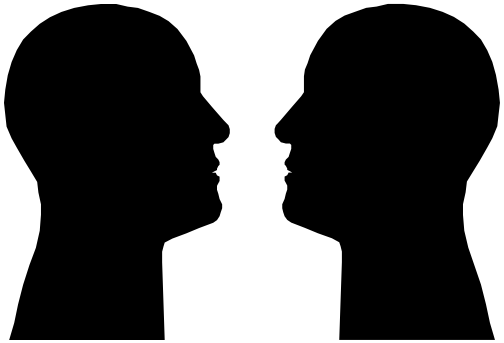


Self-Similarity: The Math

- Self-similarity is a rigorous statistical property
  - (i.e., a lot more to it than just the pretty ‘‘fractal-like’’ pictures)
- Assumes you have time series data with finite mean and variance
  - i.e., covariance stationary stochastic process
- Must be a very long time series
  - infinite is best!
- Can test for presence of self-similarity

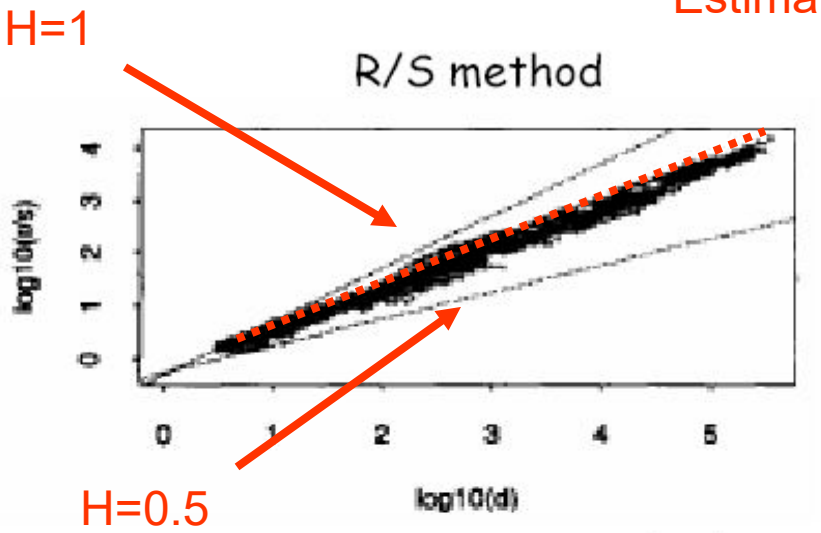
Self-Similarity: The Math

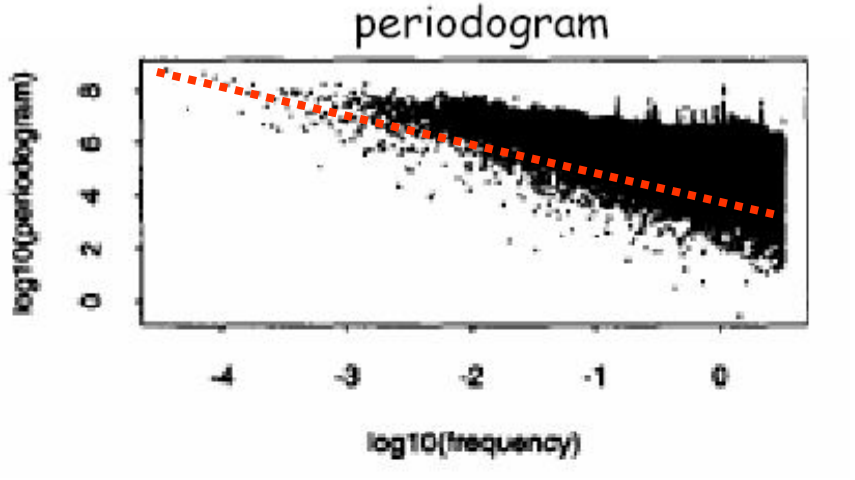
- Self-similarity manifests itself in several equivalent fashions:
- Slowly decaying variance
- Long range dependence
- Non-degenerate autocorrelations
- Hurst effect

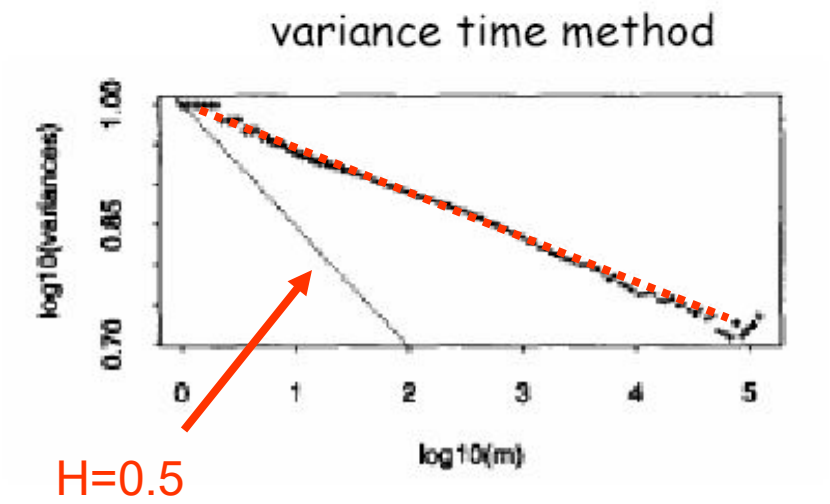


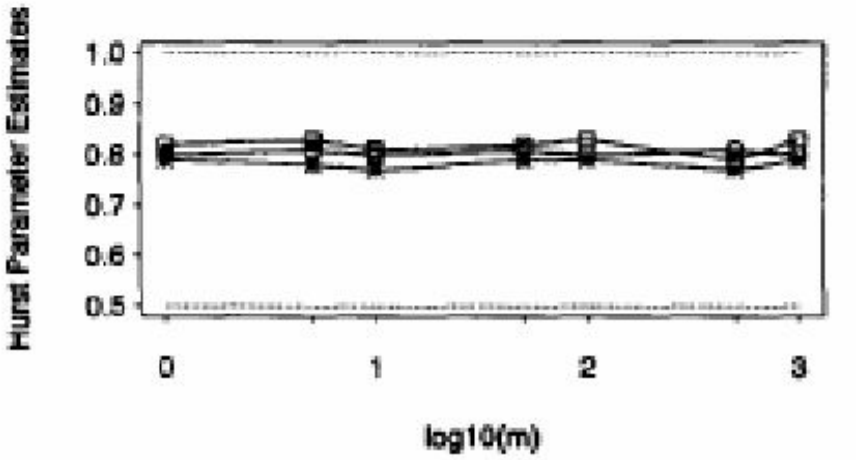
Methods of showing Self-Similarity

Estimate H ≈ 0.8





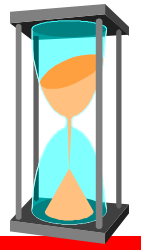




Slowly Decaying Variance

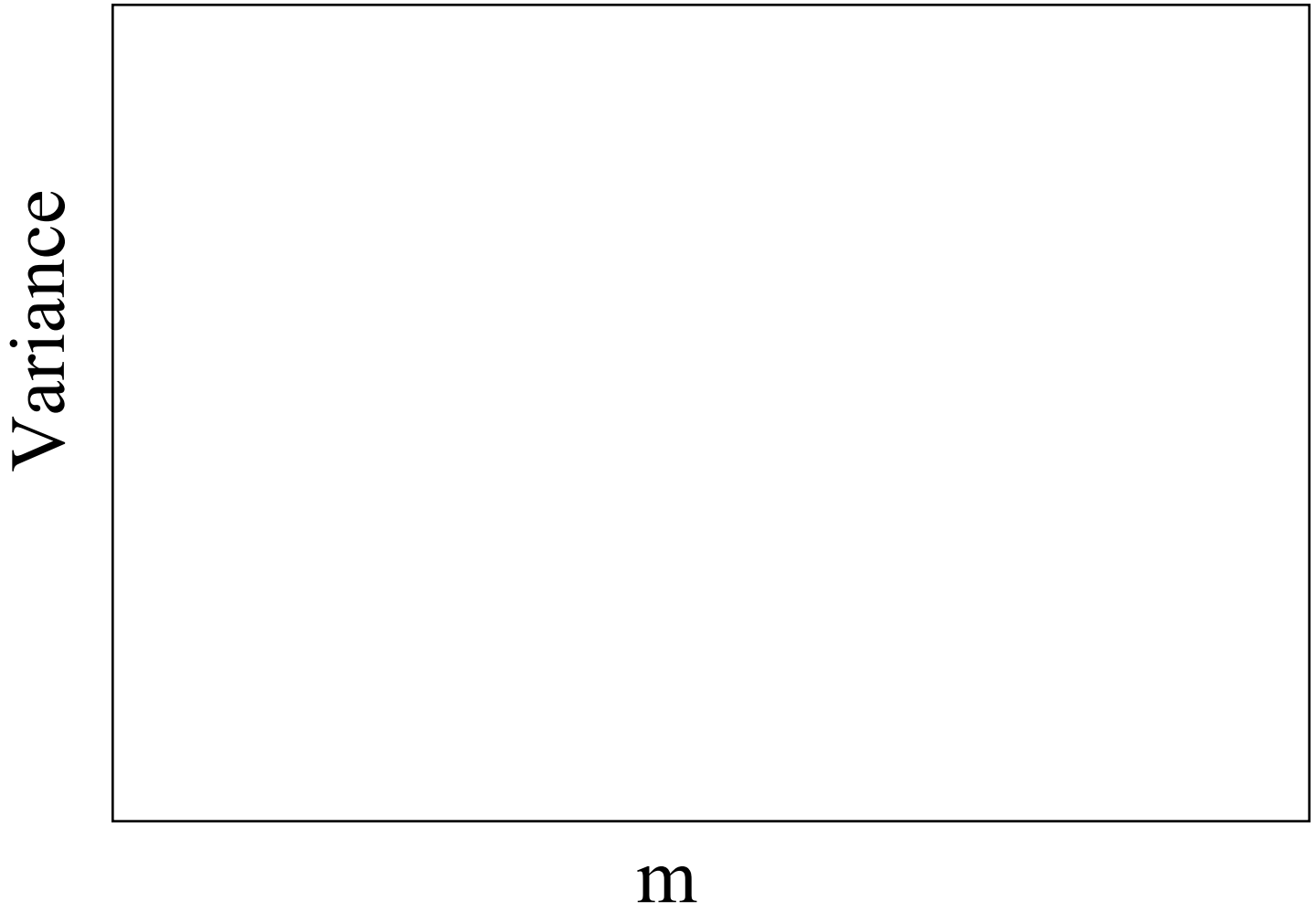
- The variance of the sample decreases more slowly than the reciprocal of the sample size
- For most processes, the variance of a sample diminishes quite rapidly as the sample size is increased, and stabilizes soon
- For self-similar processes, the variance decreases very slowly, even when the sample size grows quite large

Time-Variance Plot

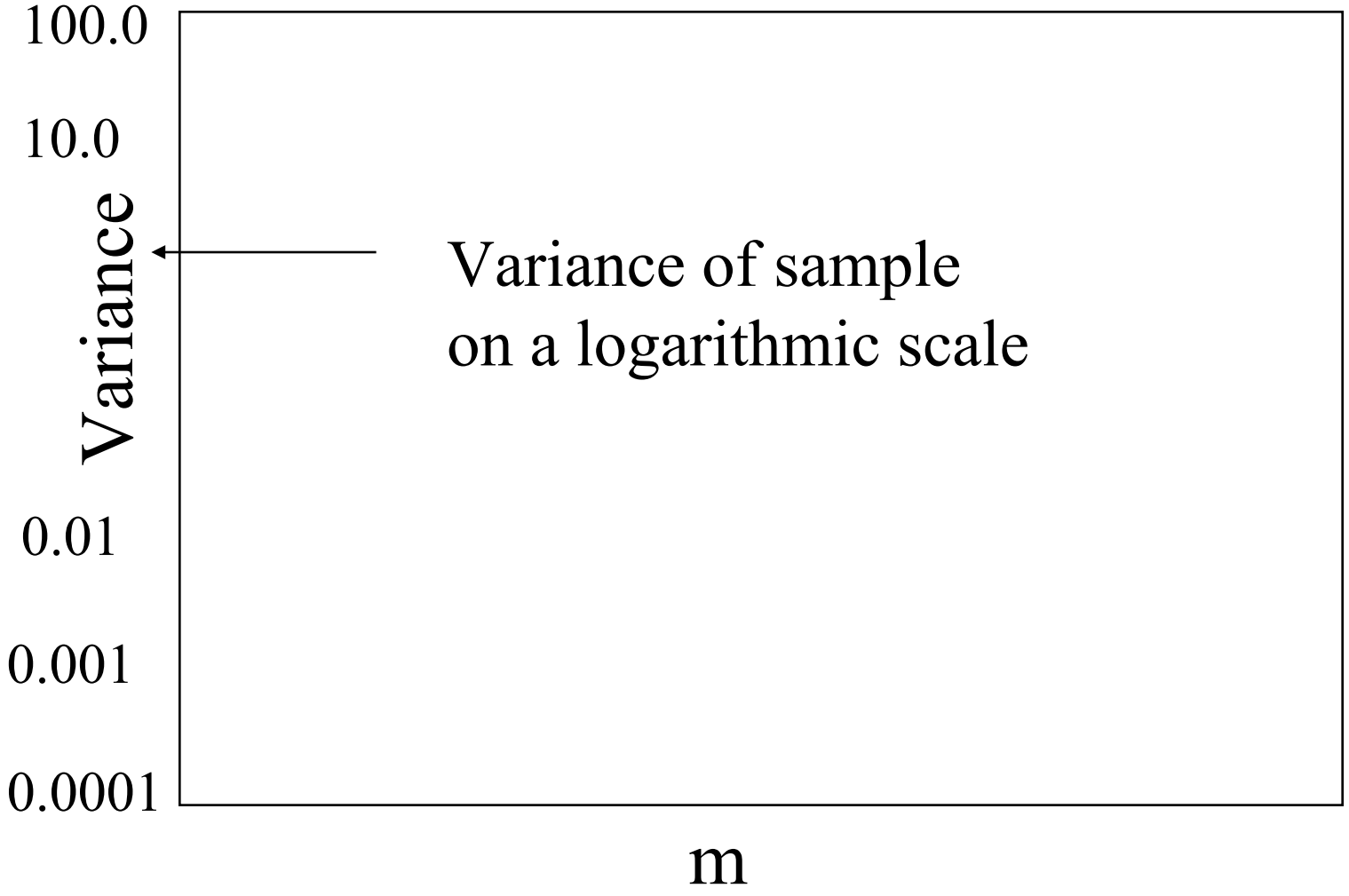


- The ‘‘variance-time plot” is one means to test for the slowly decaying variance property
- Plots the variance of the sample versus the sample size, on a log-log plot
- For most processes, the result is a straight line with slope -1
- For self-similar, the line is much flatter

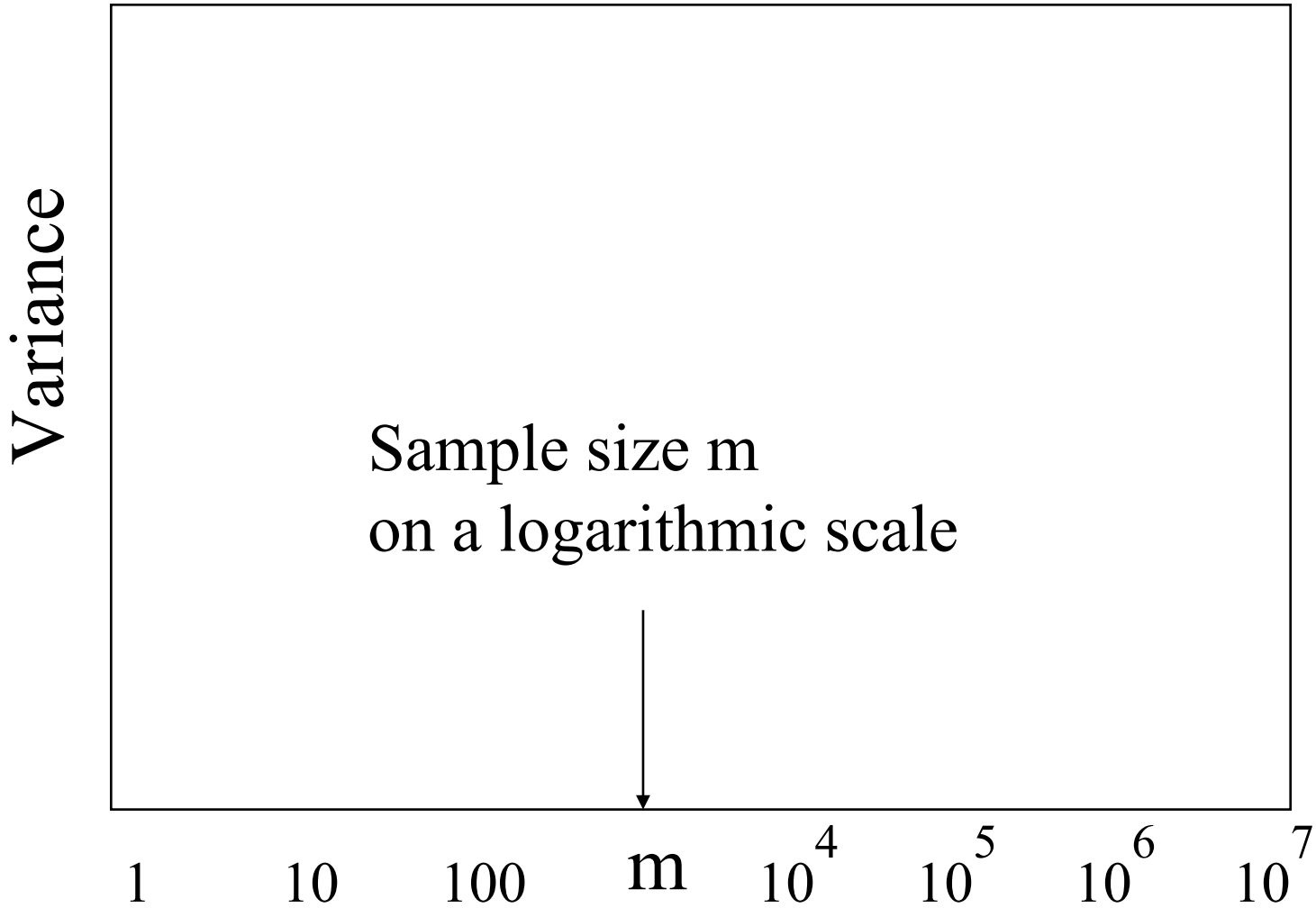
Time Variance Plot



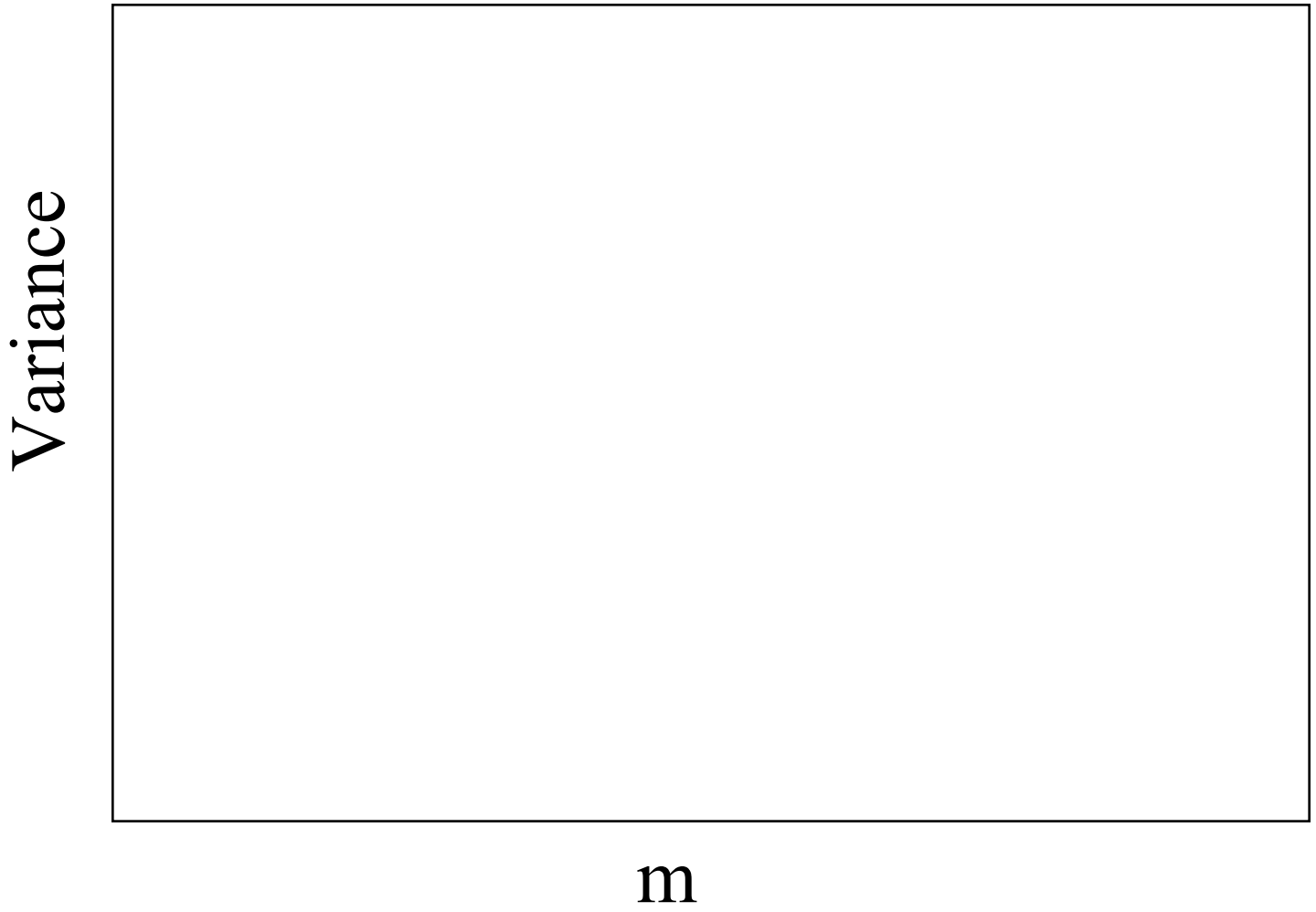
Variance-Time Plot



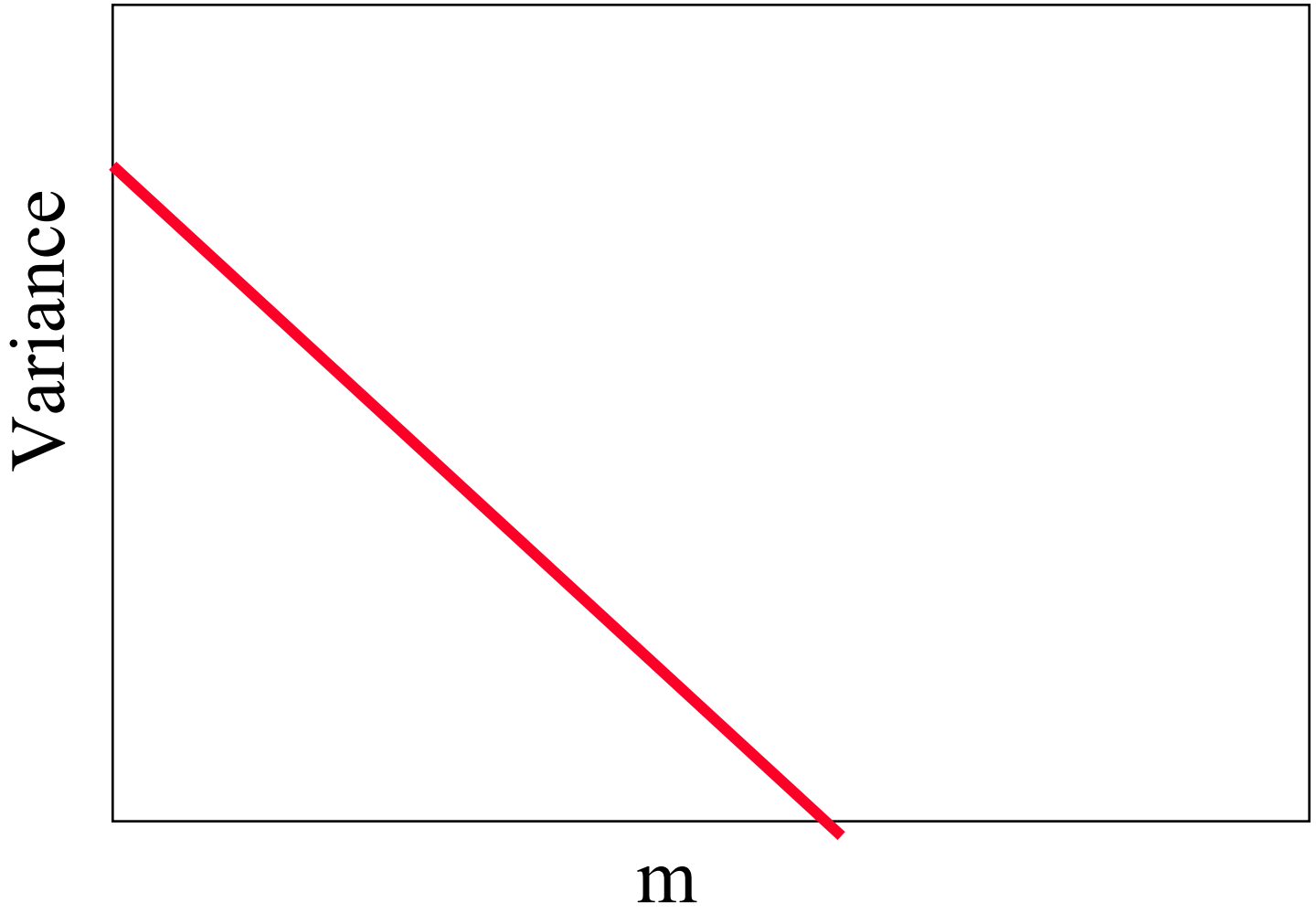
Variance-Time Plot



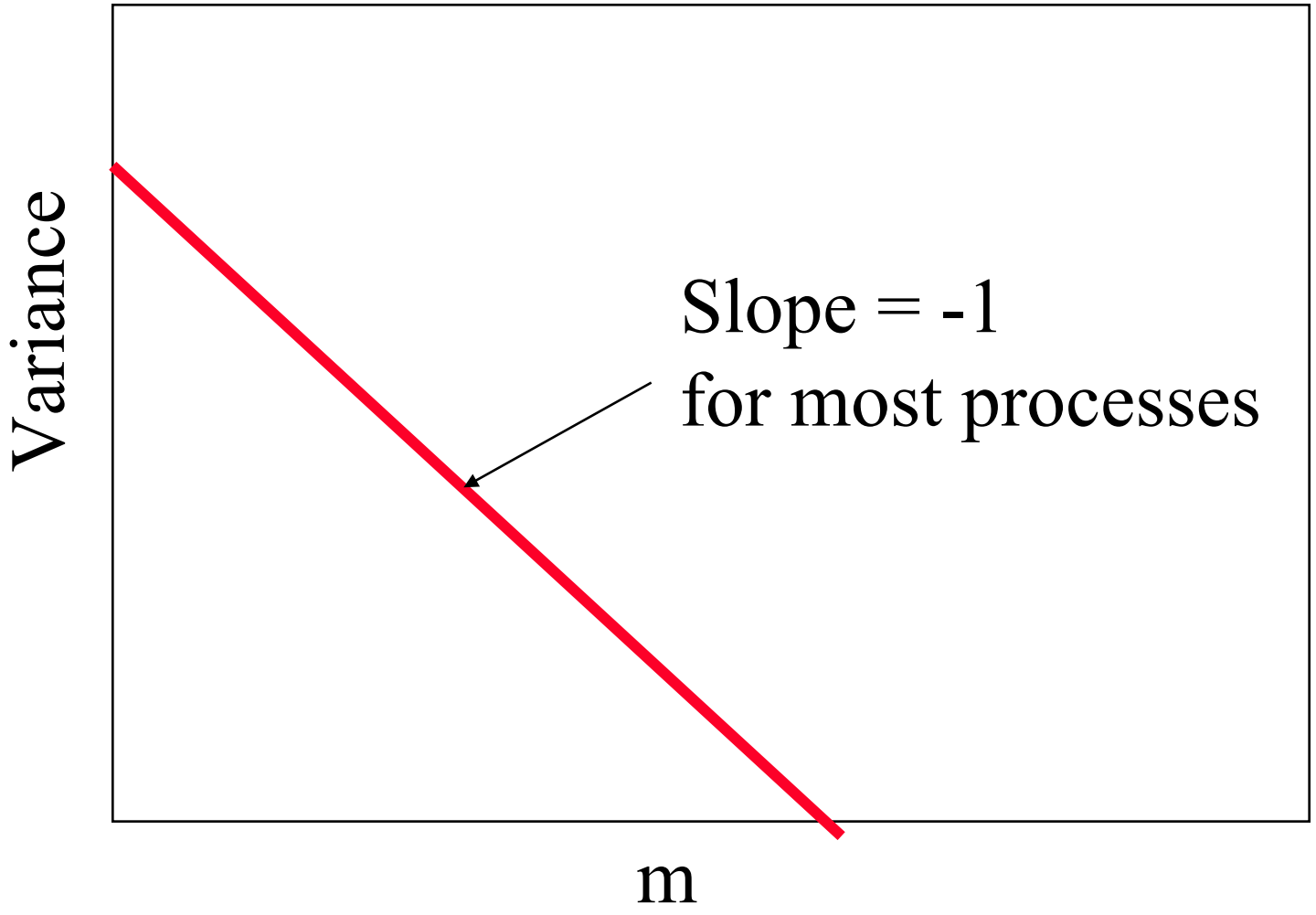
Variance-Time Plot



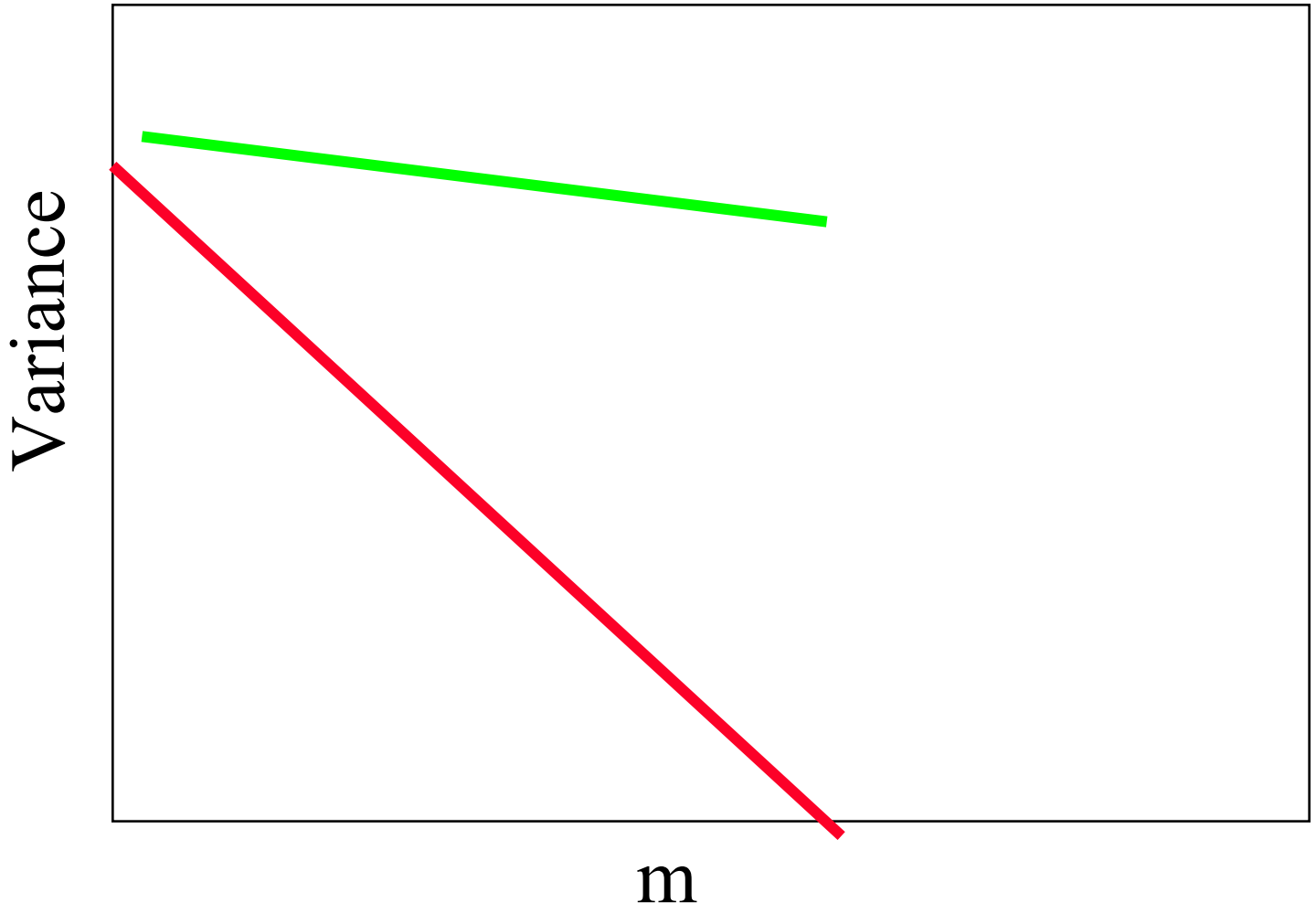
Variance-Time Plot



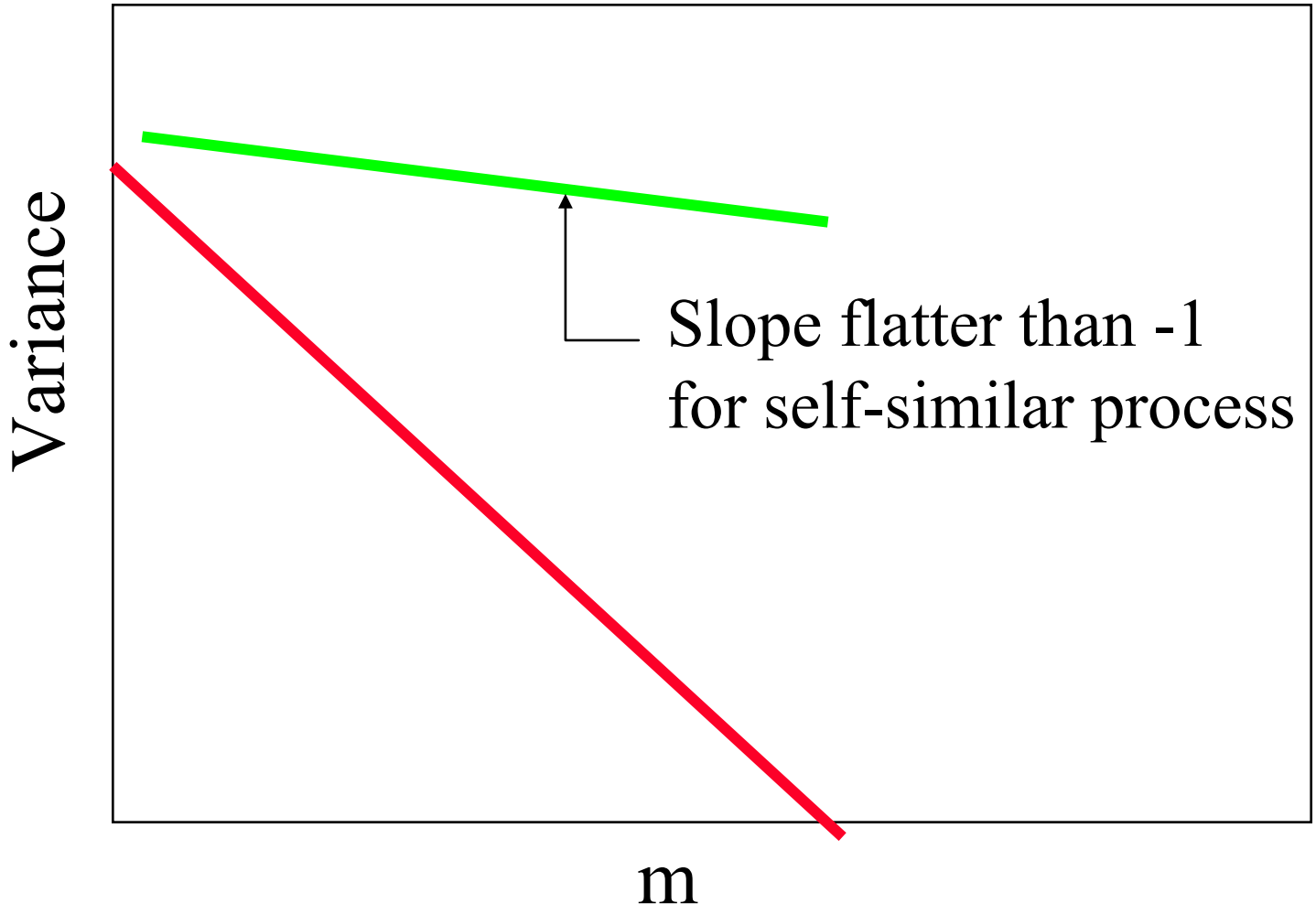
Variance-Time Plot



Variance-Time Plot



Variance-Time Plot



Long Range Dependence

- Correlation is a statistical measure of the relationship, if any, between two random variables
- Positive correlation: both behave similarly
- Negative correlation: behave as opposites
- No correlation: behavior of one is unrelated to behavior of other

Long Range Dependence

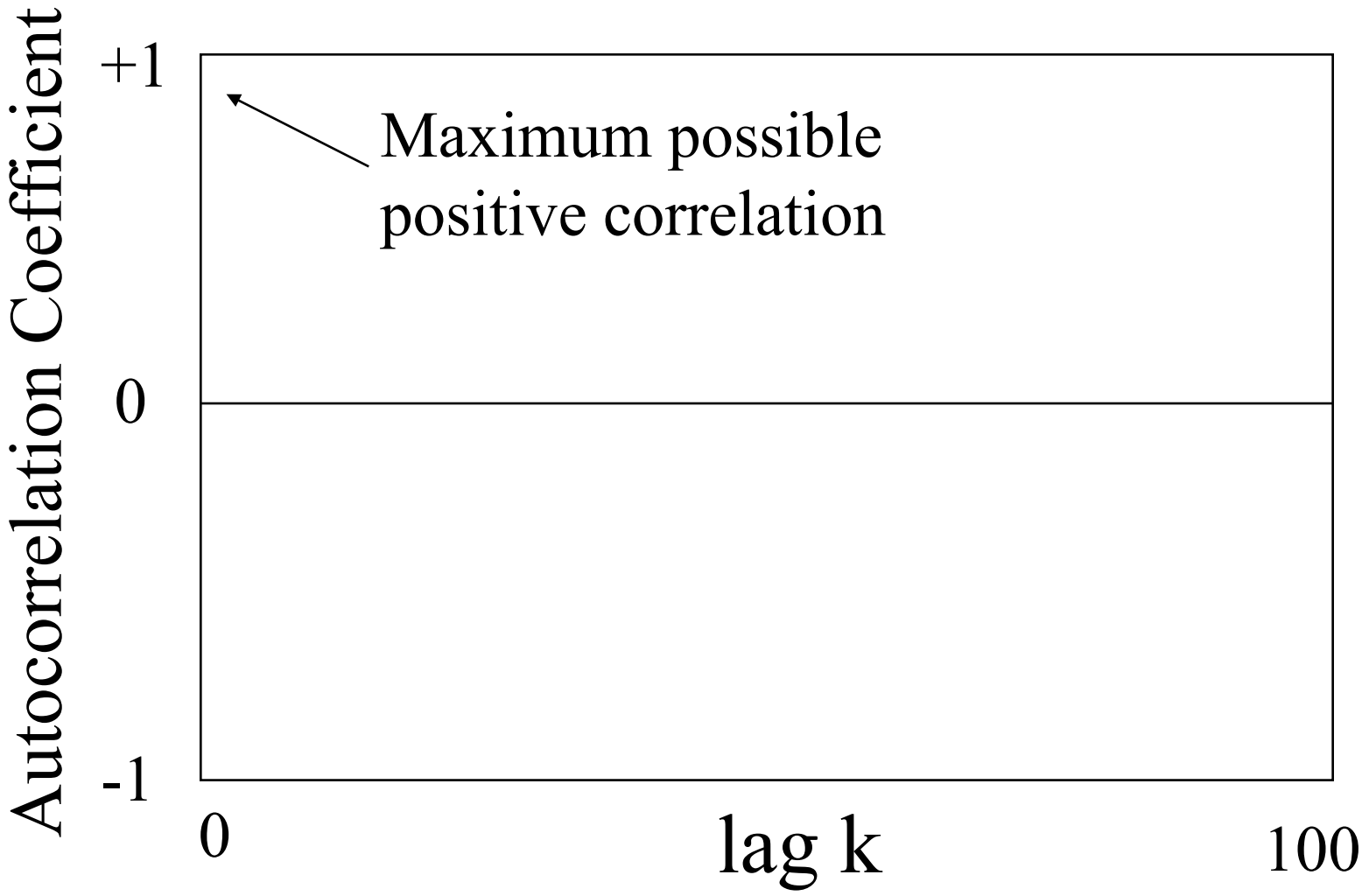
- Autocorrelation is a statistical measure of the relationship, if any, between a random variable and itself, at different time lags
- Positive correlation: big observation usually followed by another big, or small by small
- Negative correlation: big observation usually followed by small, or small by big
- No correlation: observations unrelated

Long Range Dependence

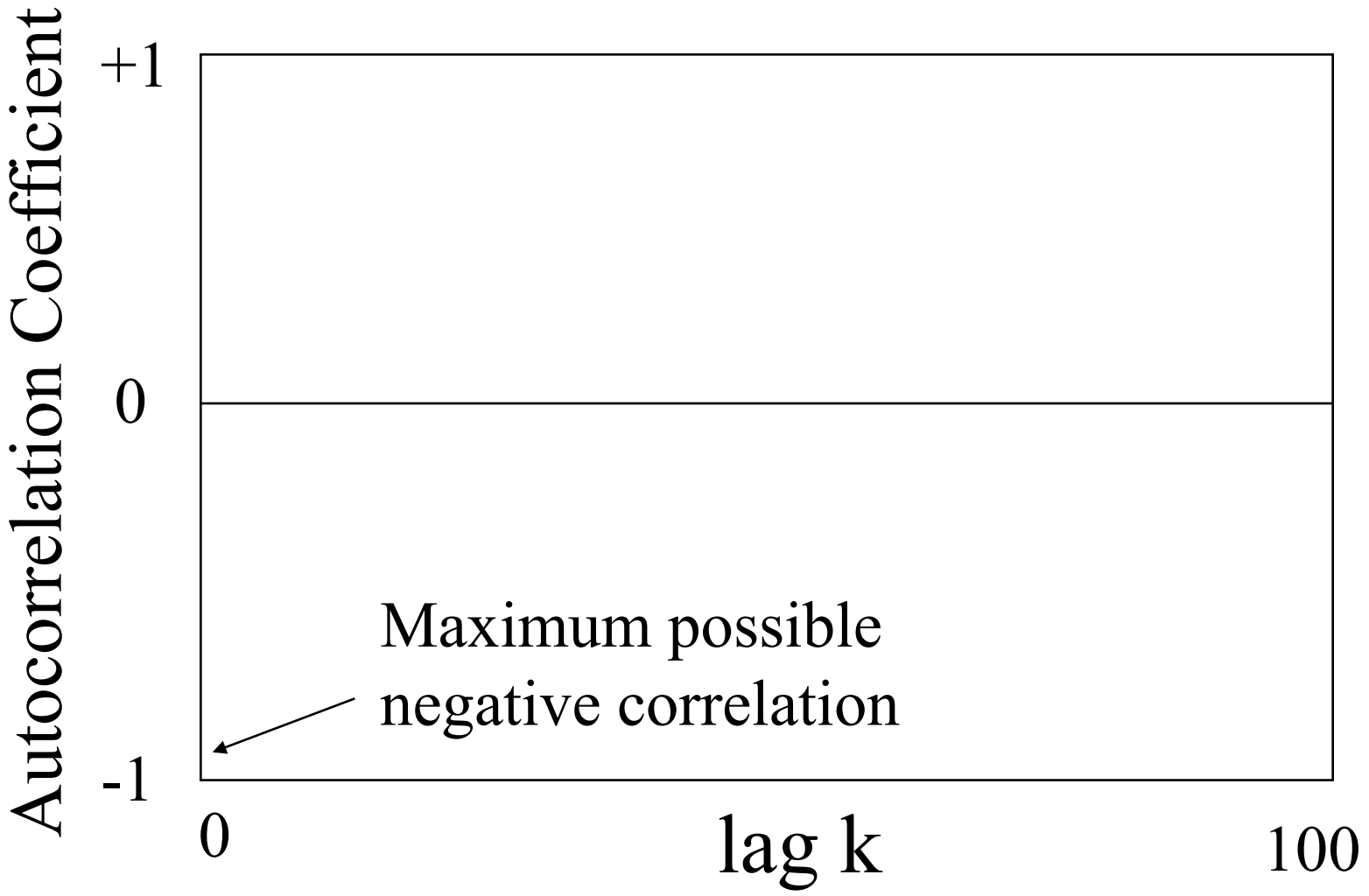
- Autocorrelation coefficient can range between:
  - +1 (very high positive correlation)
  - -1 (very high negative correlation)
- Zero means no correlation
- Autocorrelation function shows the value of the autocorrelation coefficient for different time lags k

Autocorrelation Function

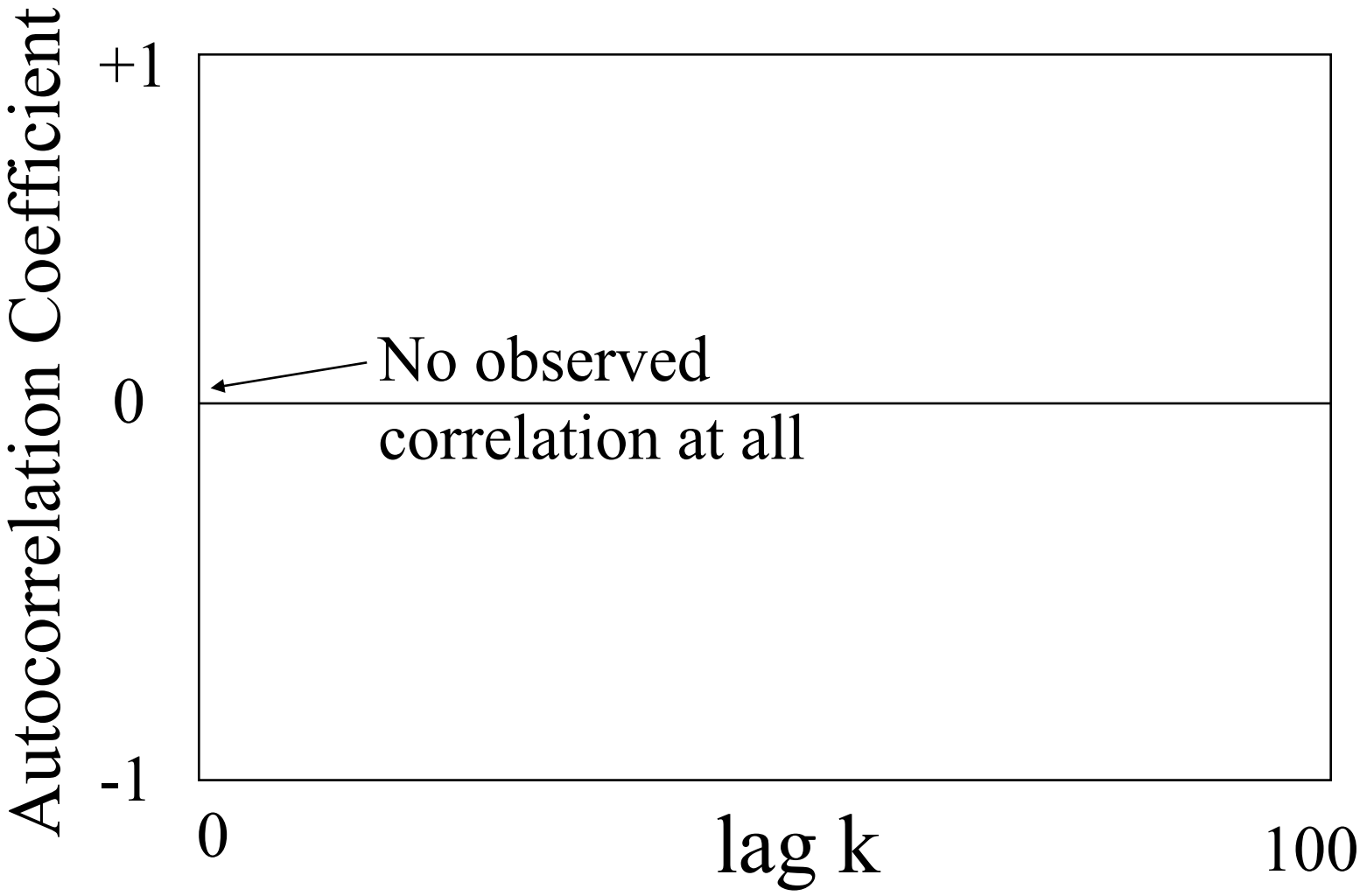
Autocorrelation Function



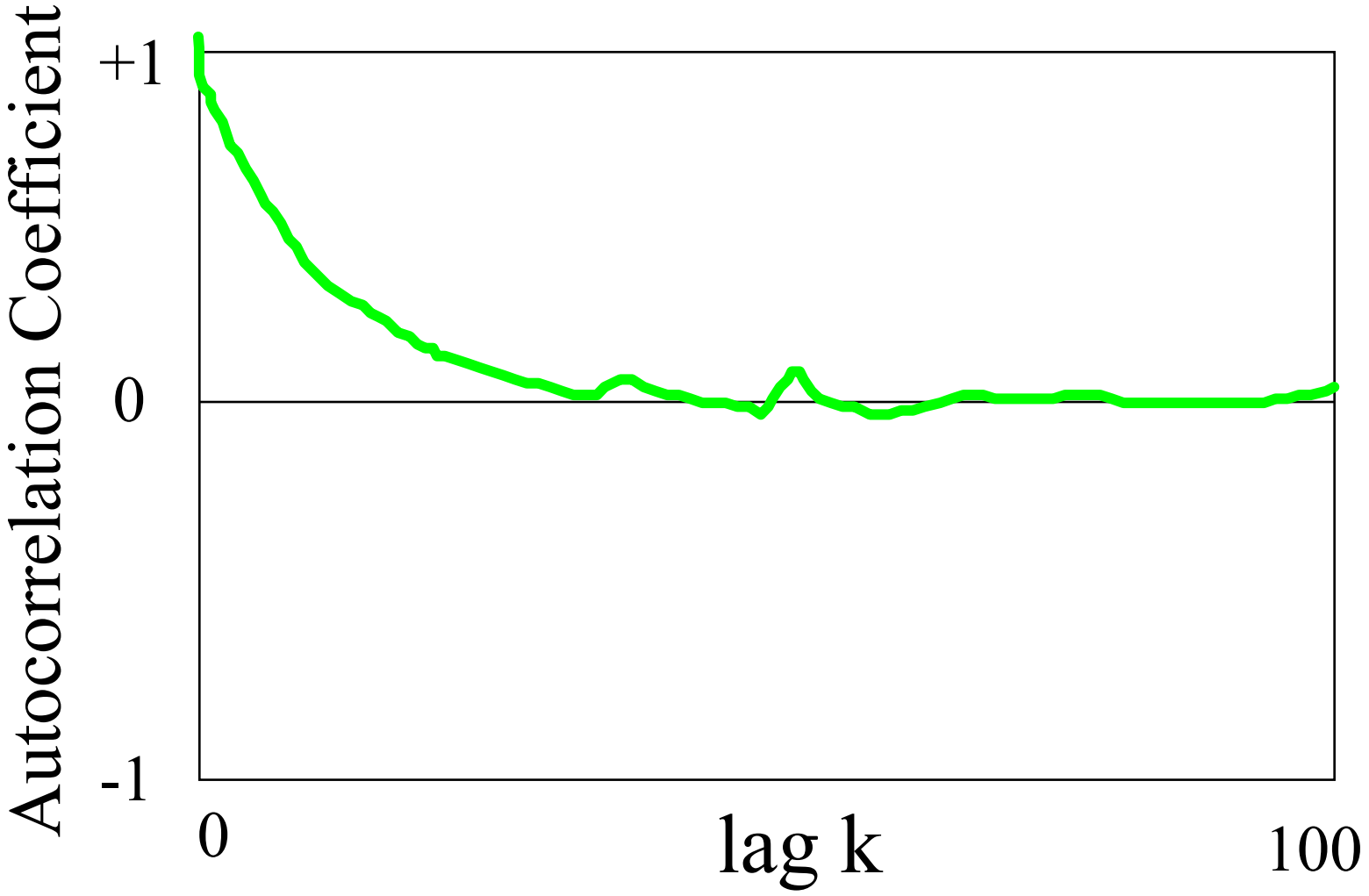
Autocorrelation Function



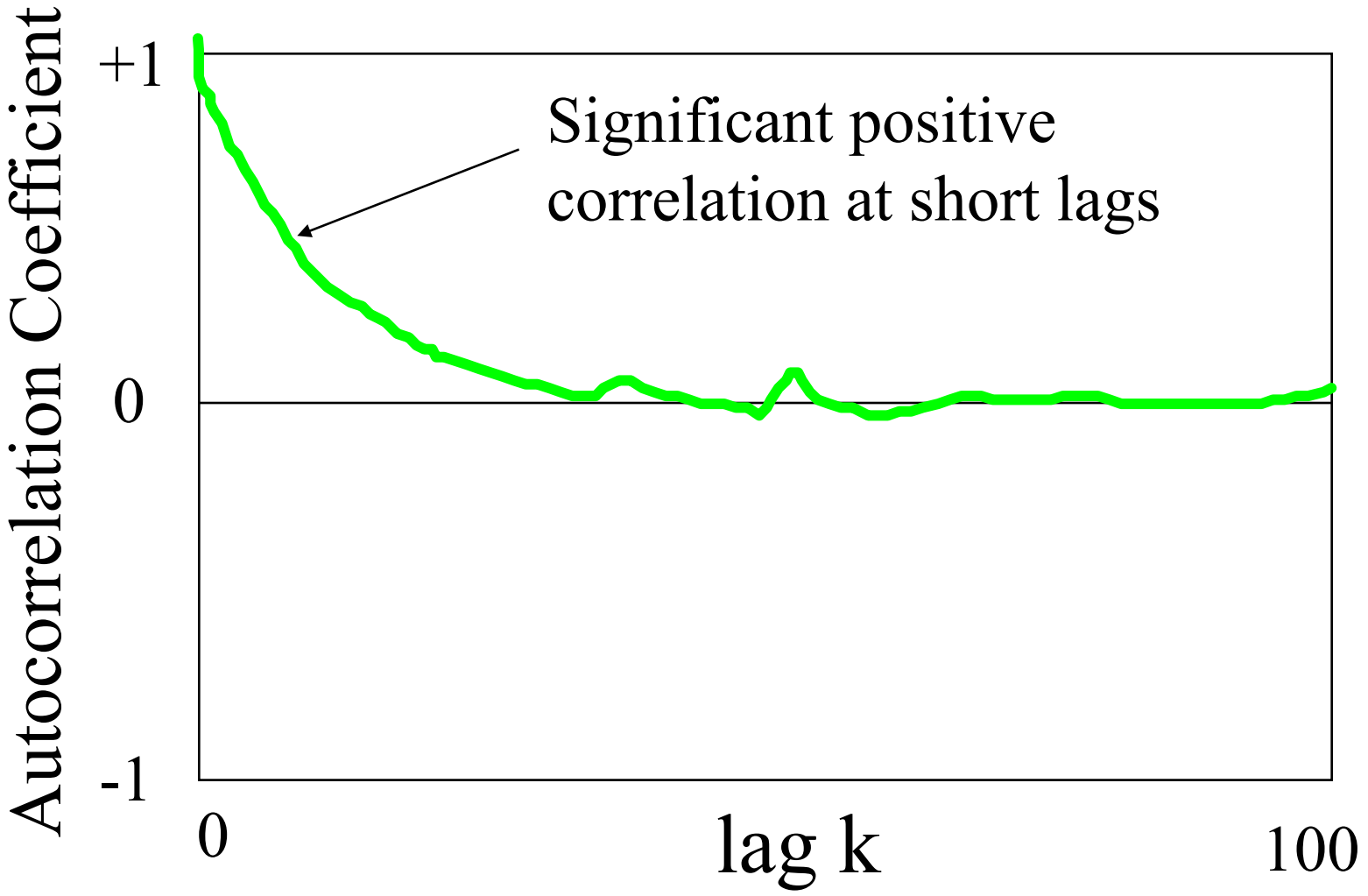
Autocorrelation Function



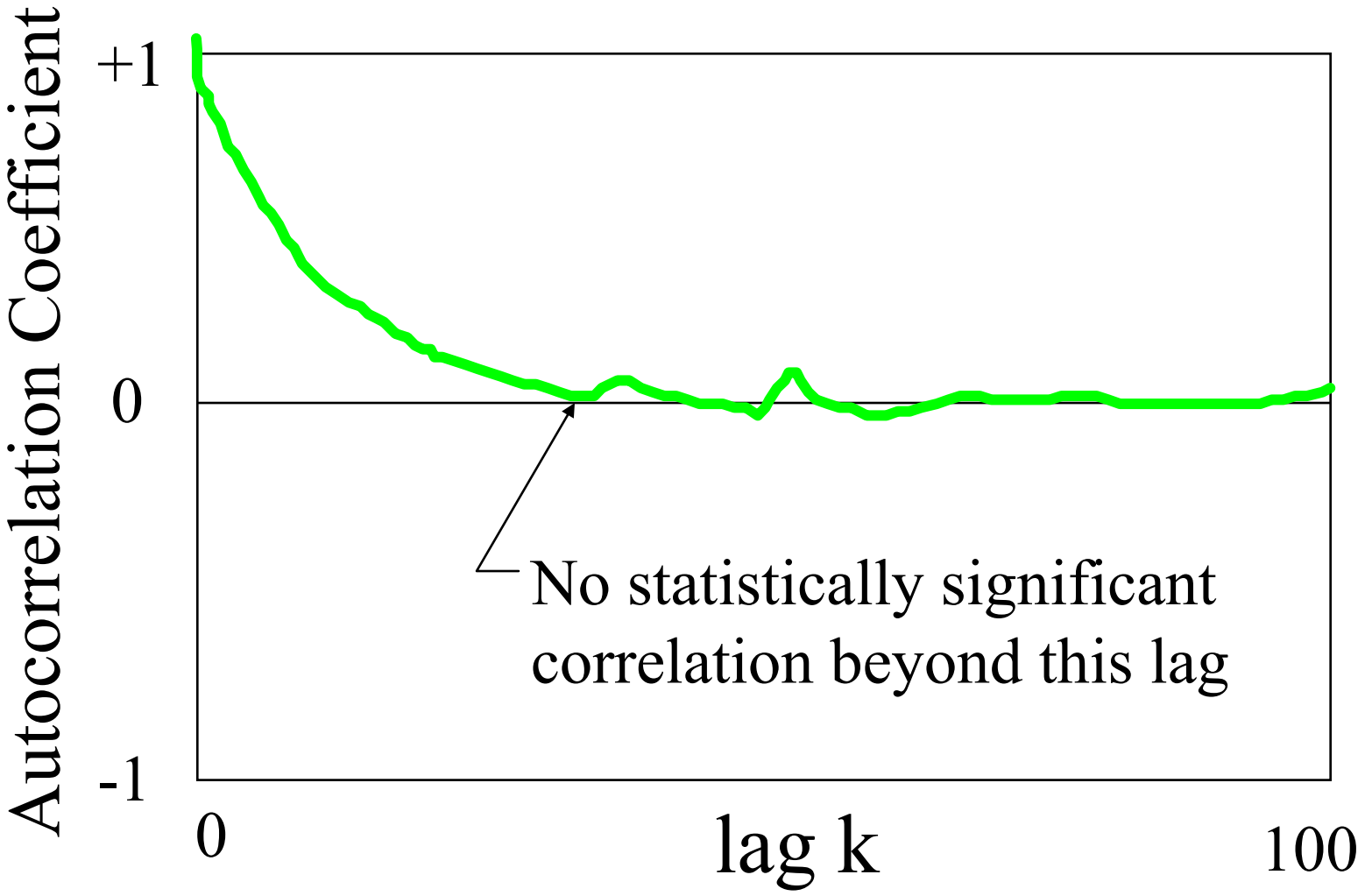
Autocorrelation Function



Autocorrelation Function



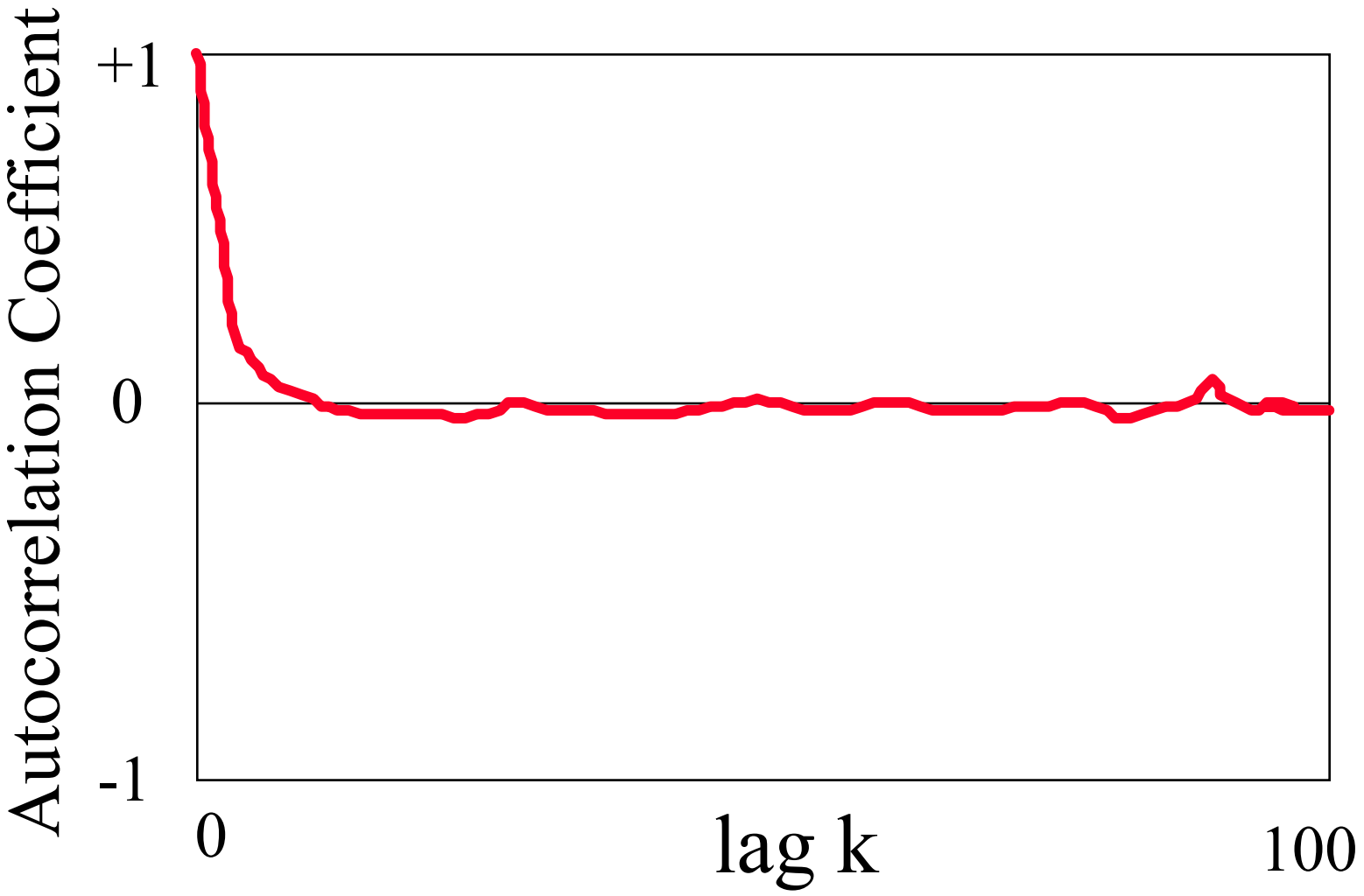
Autocorrelation Function



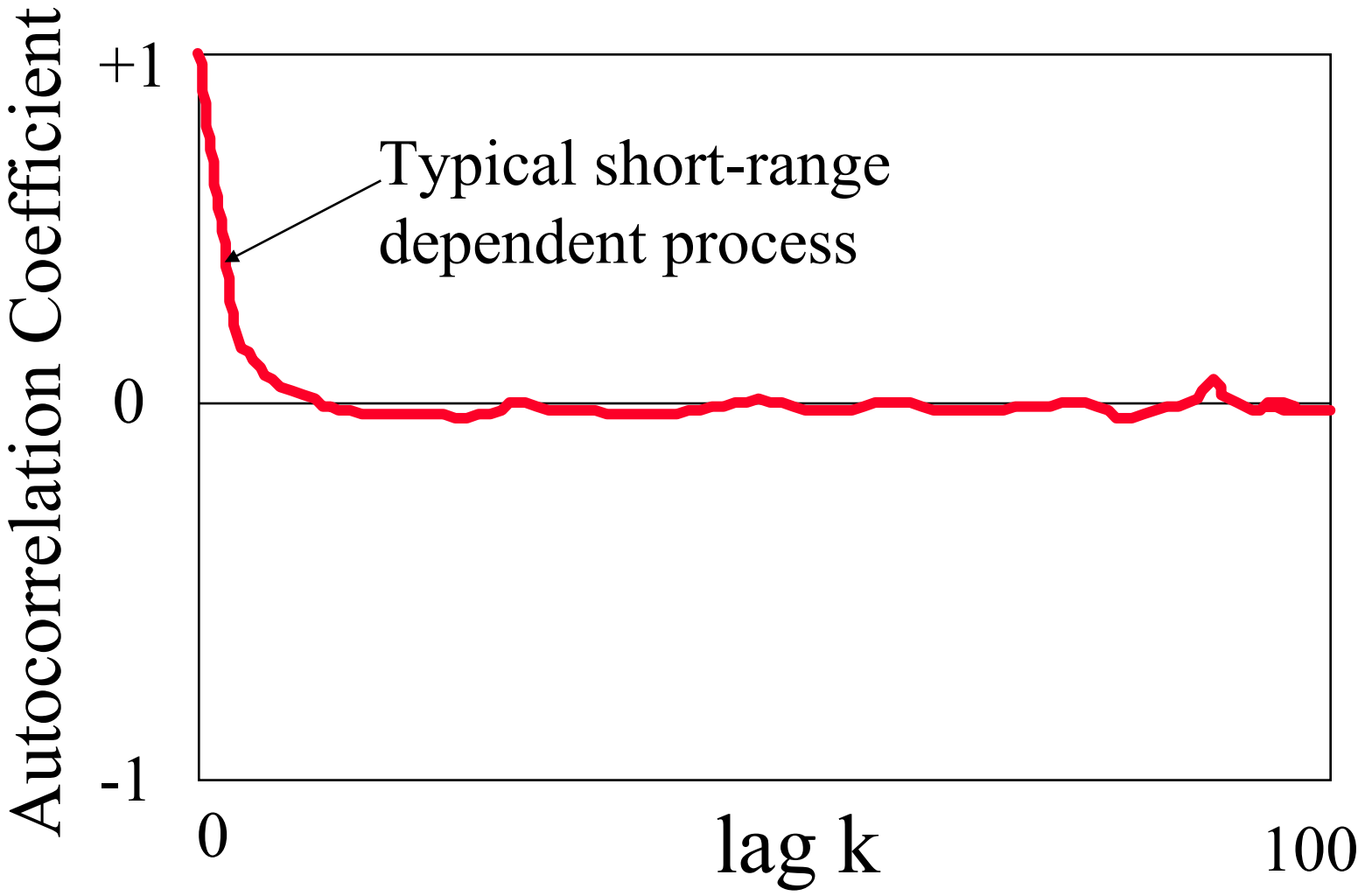
Long Range Dependence

- For most processes (e.g., Poisson, or compound Poisson), the autocorrelation function drops to zero very quickly
  - usually immediately, or exponentially fast
- For self-similar processes, the autocorrelation function drops very slowly
  - i.e., hyperbolically, toward zero, but may never reach zero
- Non-summable autocorrelation function

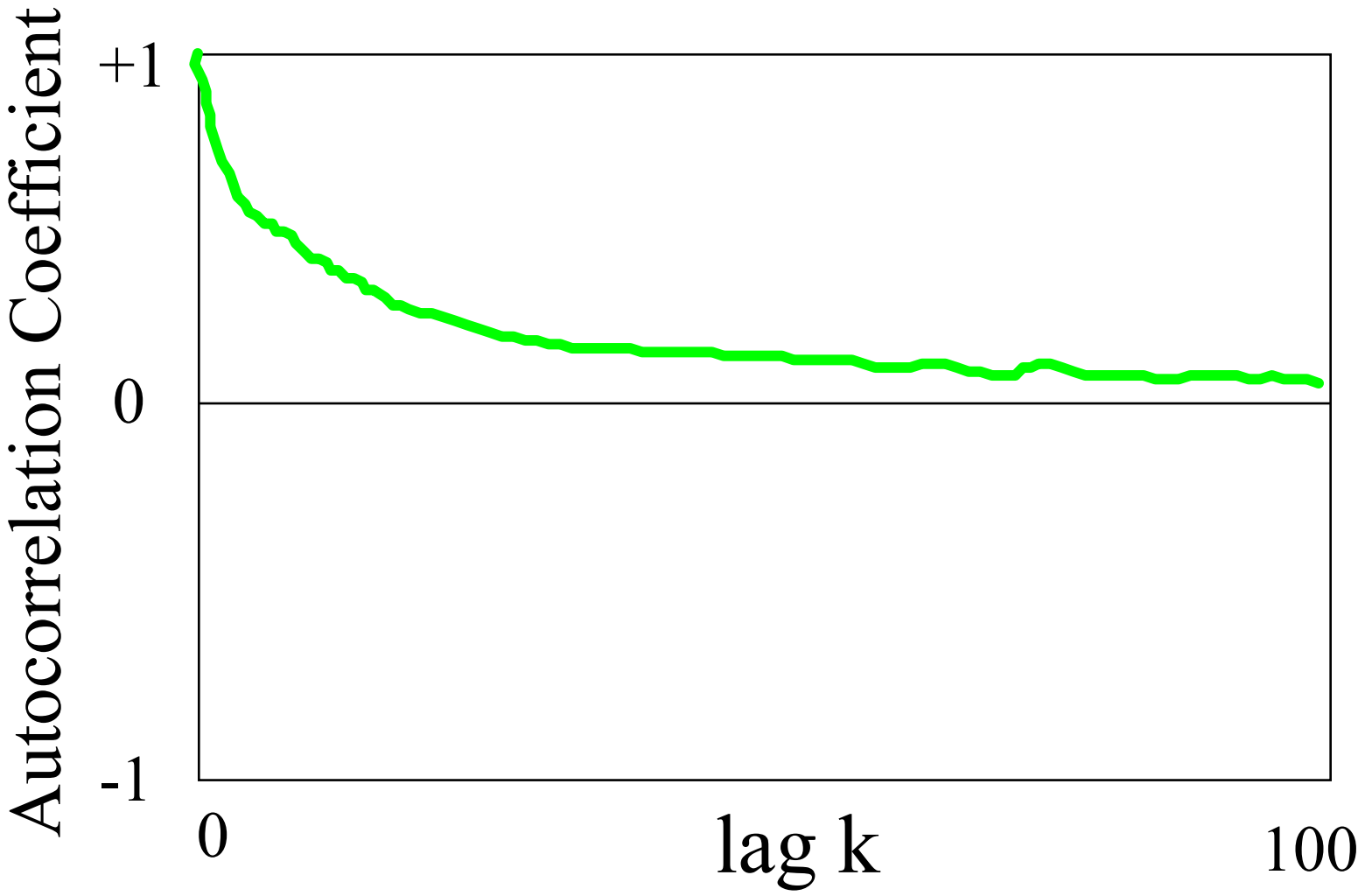
Autocorrelation Function



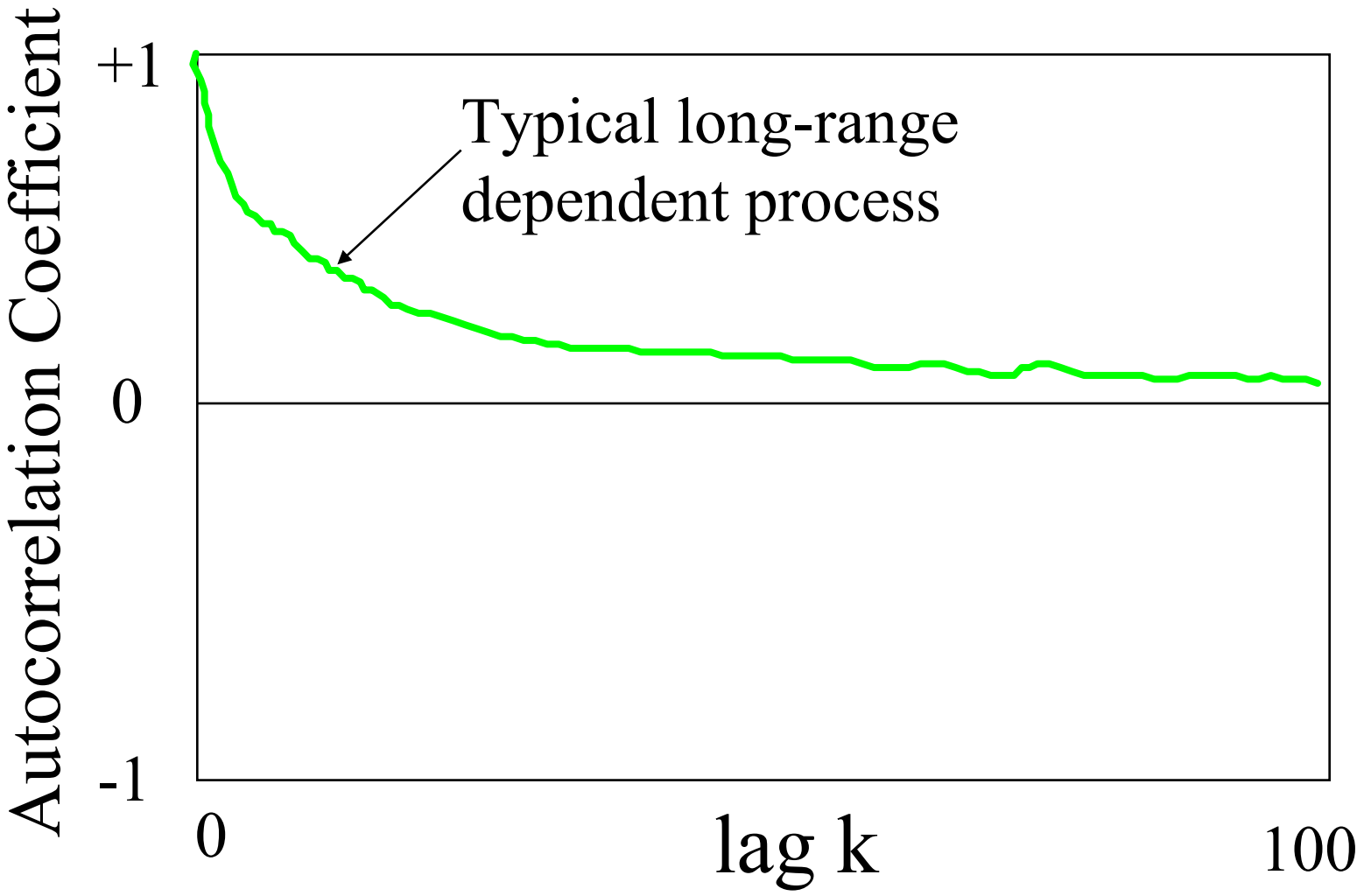
Autocorrelation Function



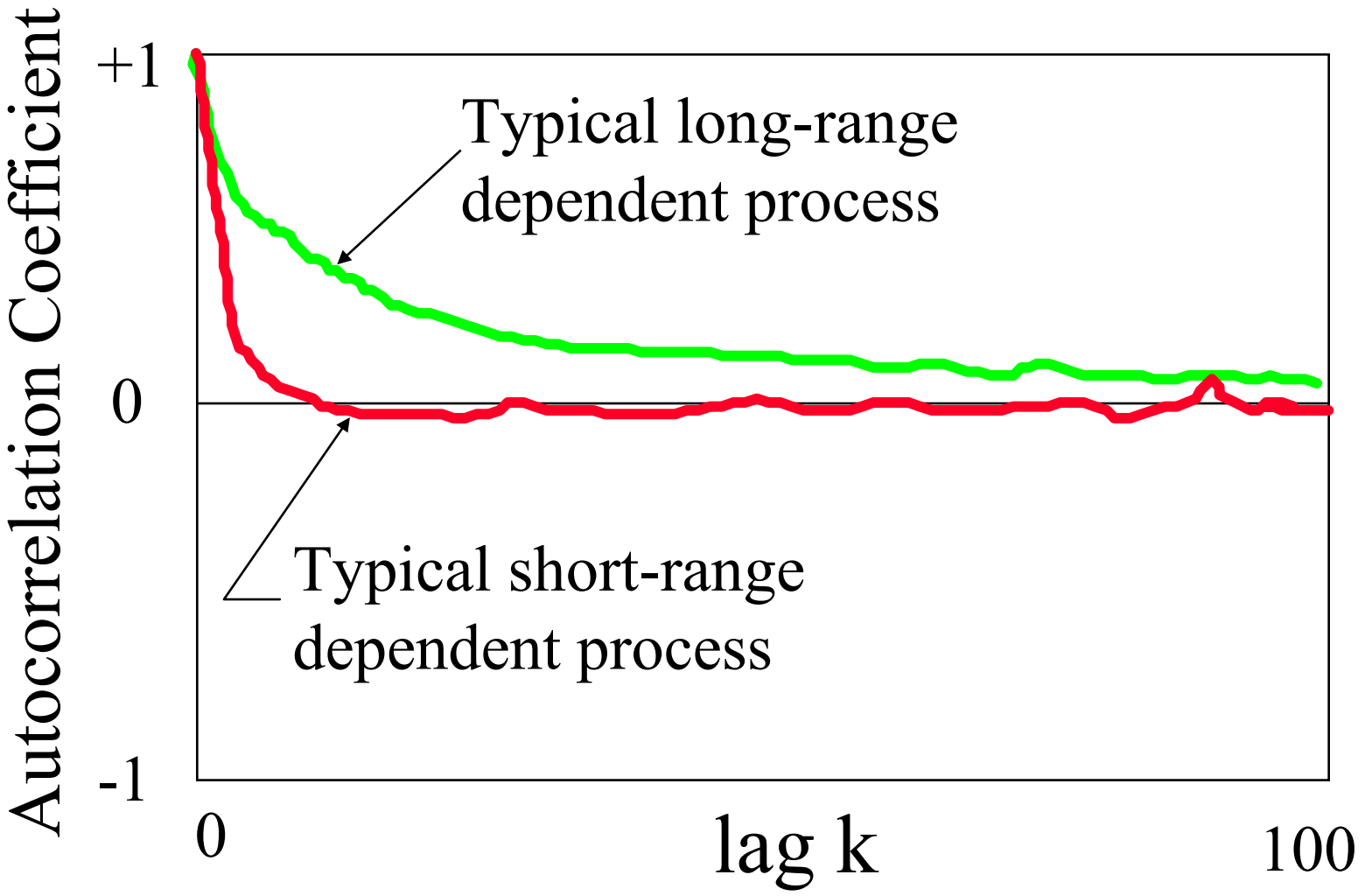
Autocorrelation Function



Autocorrelation Function



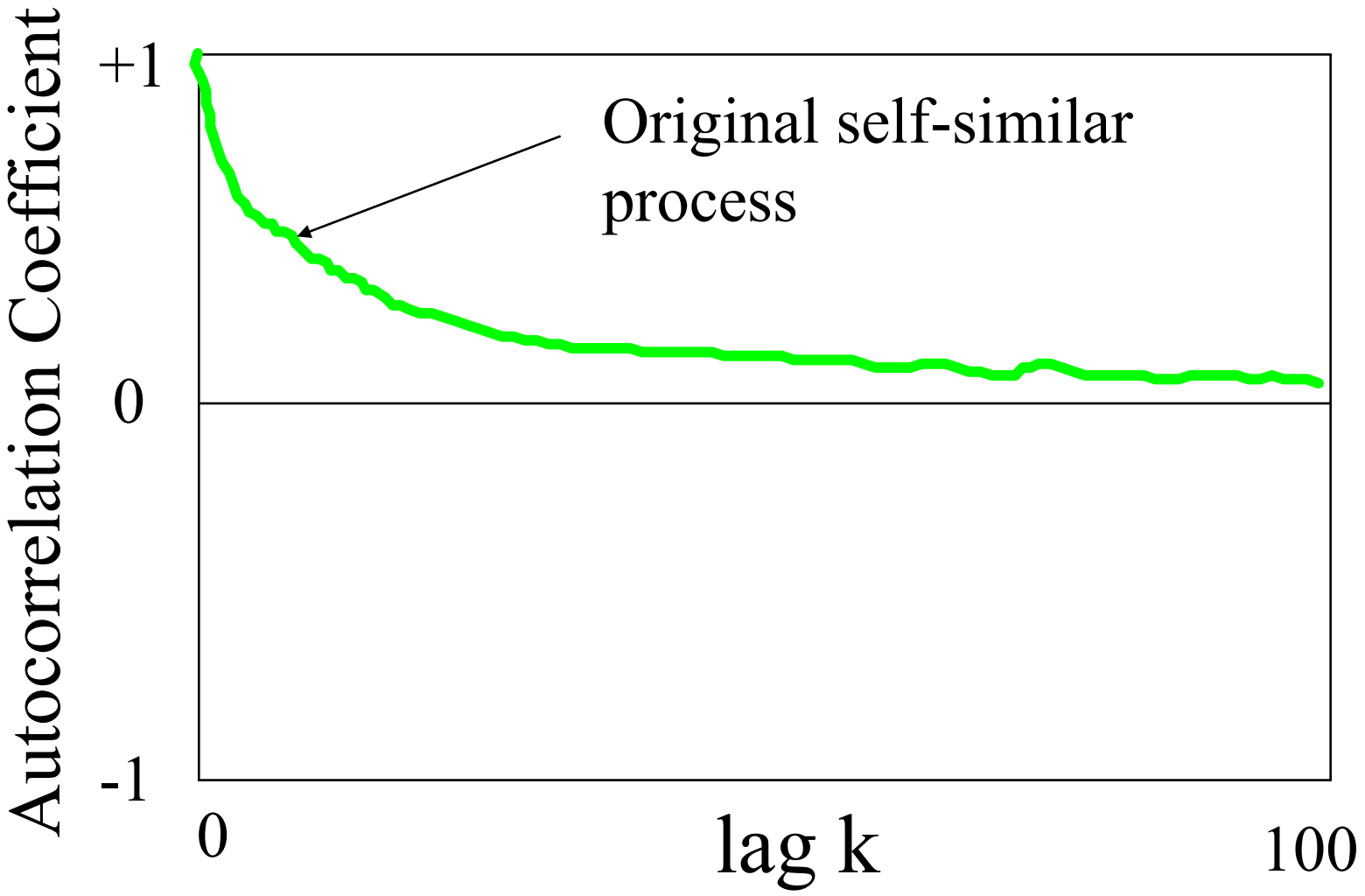
Autocorrelation Function



Non-Degenerate Autocorrelations

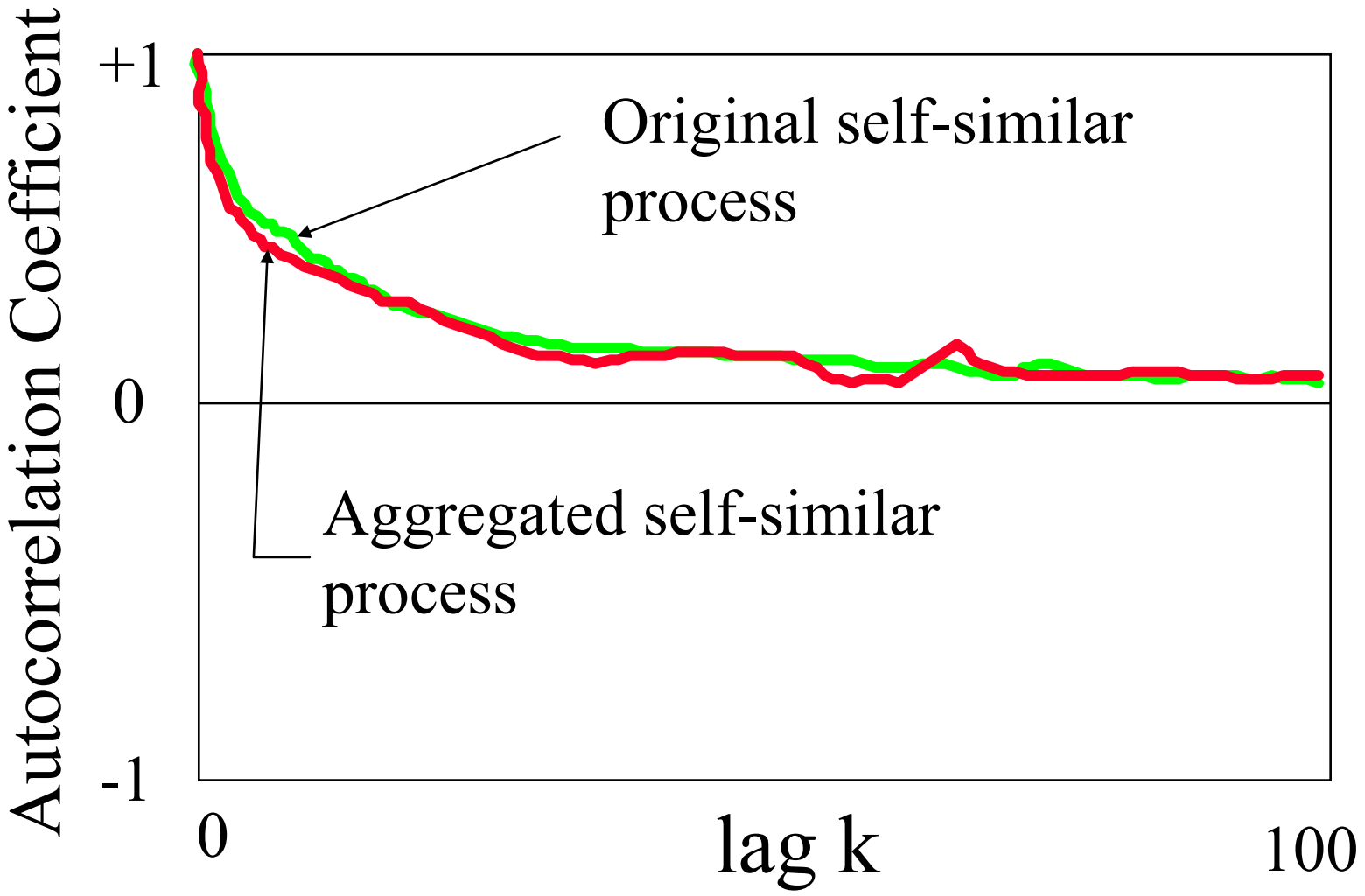
- For self-similar processes, the autocorrelation function for the aggregated process is indistinguishable from that of the original process
- If autocorrelation coefficients match for all lags k, then called exactly self-similar
- If autocorrelation coefficients match only for large lags k, then called asymptotically self-similar

Autocorrelation Function



Autocorrelation Function

Autocorrelation Function



Aggregation

- Aggregation of a time series X(t) means smoothing the time series by averaging the observations over non-overlapping blocks of size m to get a new time series $X_m(t)$

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values

```
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...
```

- Then the aggregated series for m = 2 is:

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

- Then the aggregated series for m = 2 is:

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values:

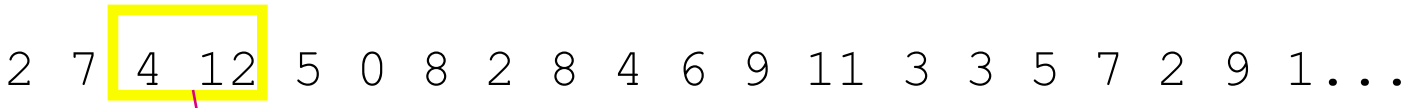
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

- Then the aggregated series for m = 2 is:

4.5

Aggregation example

- Suppose the original time series X(t) contains the following (made up) values:

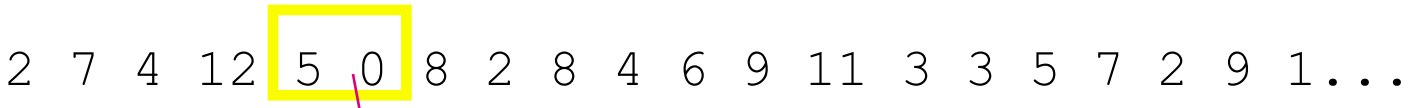


- Then the aggregated series for m = 2 is:

4.5 8.0

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values:

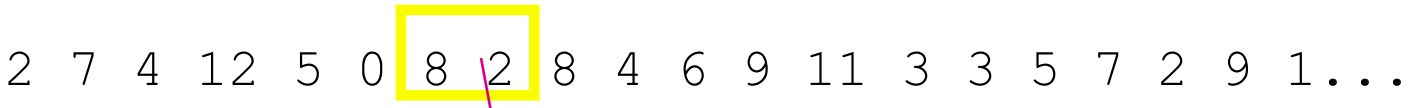


- Then the aggregated series for m = 2 is:

4.5 8.0 2.5

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values:



- Then the aggregated series for m = 2 is:

4.5 8.0 2.5 5.0

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values:

```
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...
```

- Then the aggregated series for m = 2 is:

4.5 8.0 2.5 5.0 6.0 7.5 7.0 4.0 4.5 5.0...

Aggregation Example

- Suppose the original time series X(t) contains the following (made up) values:

```
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...
```

Then the aggregated time series for m = 5 is:

Aggregation: An Example

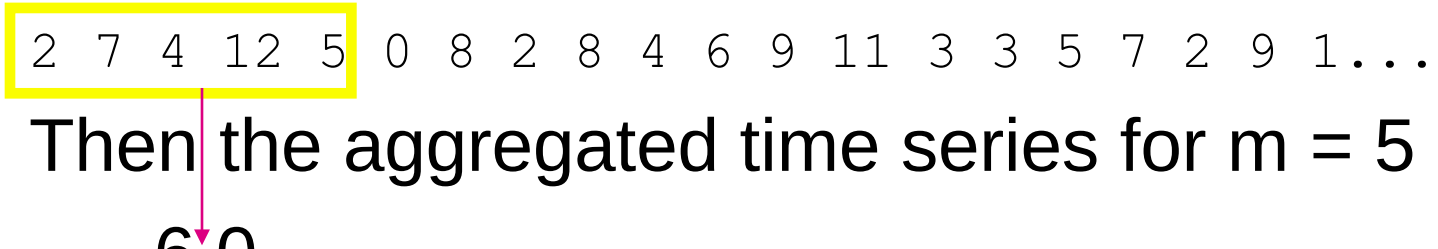
- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

Then the aggregated time series for m = 5 is:

Aggregation: An Example

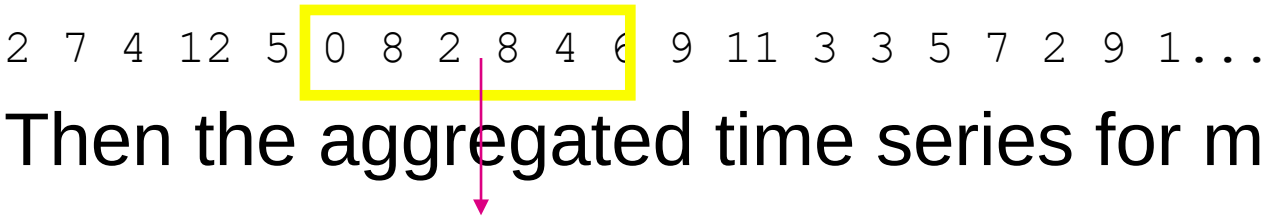
- Suppose the original time series X(t) contains the following (made up) values:



Then the aggregated time series for m = 5 is:

Aggregation: An Example

- Suppose the original time series X(t) contains the following (made up) values:

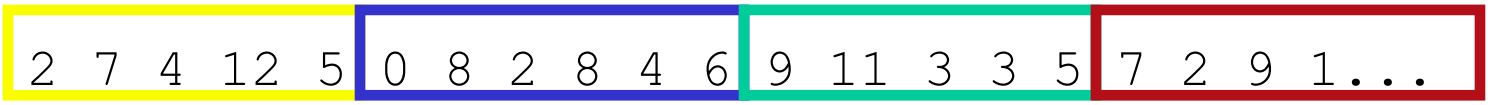


Then the aggregated time series for m = 5 is:

6.0 4.4

Aggregation: An Example

- Suppose the original time series X(t) contains the following (made up) values:



Then the aggregated time series for m = 5 is:

6.0 4.4 6.4 4.8 ...

Aggregation: An Example

- Suppose the original time series X(t) contains the following (made up) values:

```
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...
```

Then the aggregated time series for m = 10 is:

Aggregation: An Example

- Suppose the original time series X(t) contains the following (made up) values:

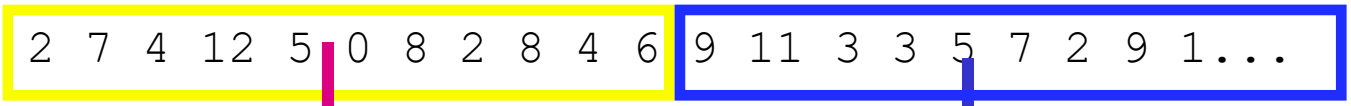
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1...

Then the aggregated time series for m = 10 is:

5.2

Aggregation: An Example

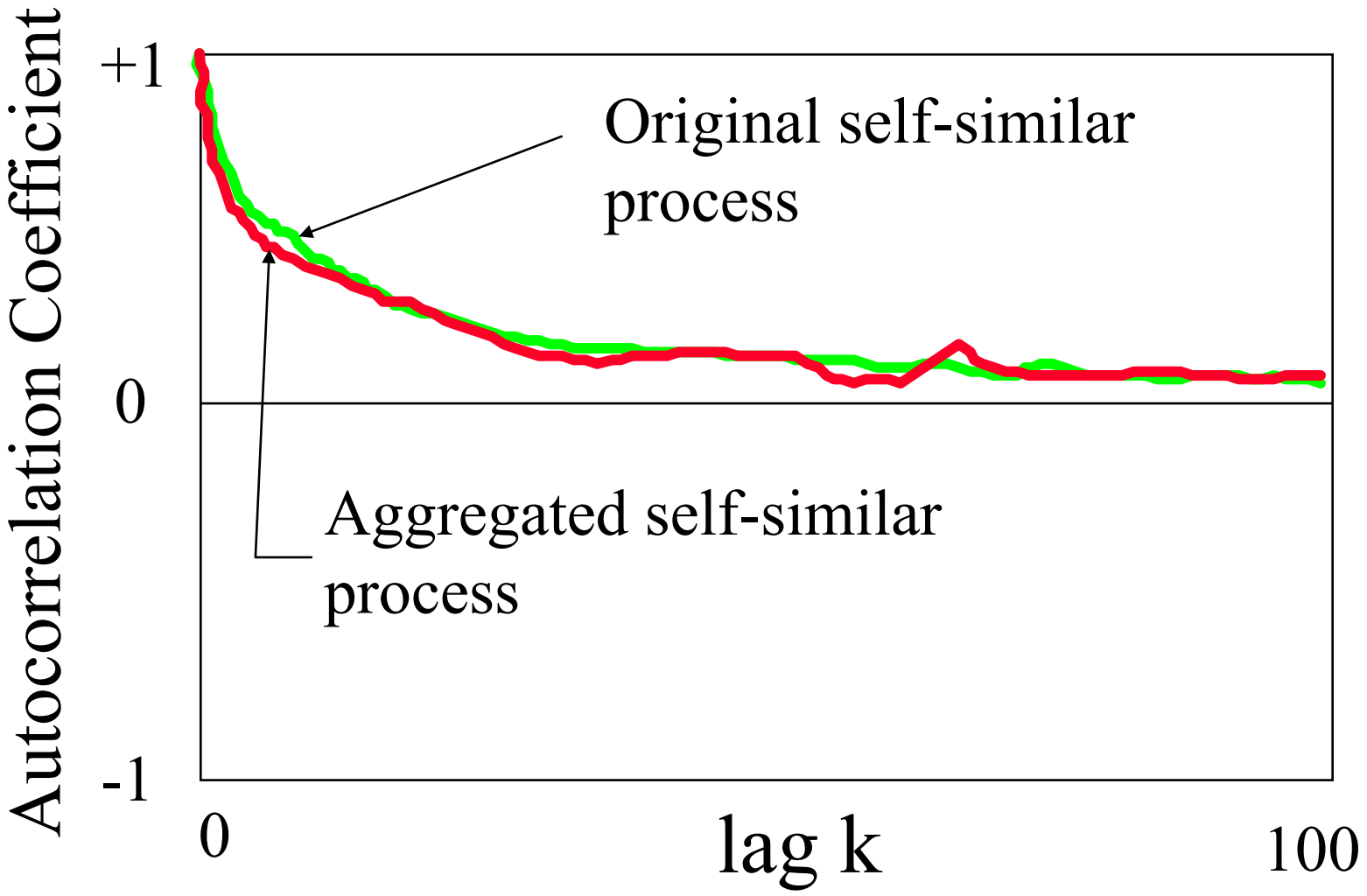
- Suppose the original time series X(t) contains the following (made up) values:



Then the aggregated time series for m = 10 is:

5.2 5.6

Autocorrelation Function



Hurst Effect

- For almost all naturally occurring time series, the rescaled adjusted range statistic (also called the R/S statistic) for sample size n obeys the relationship

$$E[R(n)/S(n)] = c\, n^H$$

where:

$R(n) = \max(0, W_1, \ldots W_n) - \min(0, W_1, \ldots W_n)$

$S^2(n)$ is the sample variance, and

for $k = 1, 2, \ldots n$

Hurst Effect

- For models with only short range dependence, H is almost always 0.5
- For self-similar processes, $0.5 < H < 1.0$
- This discrepancy is called the Hurst Effect, and H is called the Hurst parameter
- Single parameter to characterize self-similar processes

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- There are 20 data points in this example

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- There are 20 data points in this example
- For R/S analysis with n = 1, you get 20 samples, each of size 1:

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- There are 20 data points in this example
- For R/S analysis with n = 1, you get 20 samples, each of size 1:

Block 1: $\overline{X}_{n,1} = 2$, W = 0, R(n) = 0, S(n) = 0

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- There are 20 data points in this example
- For R/S analysis with n = 1, you get 20 samples, each of size 1:

Block 2: X = 7, W = 0, R(n) = 0, S(n) = 0

$$\overline{n} \qquad 1$$

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 2, you get 10 samples, each of size 2:

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 2, you get 10 samples, each of size 2:

Block 1: $\bar{X}_n = 4.5$, $W_1 = -2.5$, $W_2 = 0$,

$R(n) = 0 - (-2.5) = 2.5$, $S(n) = 2.5$,

$R(n)/S(n) = 1.0$

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 2, you get 10 samples, each of size 2:

Block 2: $\overline{X}_n$ = 8.0, W_1 = -4.0, W_2 = 0,

R(n) = 0 - (-4.0) = 4.0, S(n) = 4.0,

R(n)/S(n) = 1.0

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 3, you get 6 samples, each of size 3:

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 3, you get 6 samples, each of size 3:

Block 1: $X_1 = 4.3$, $W_1 = -2.3$, $W_2 = 0.3$, $W_3 = 0$

$R(n) = 0.3 - (-2.3) = 2.6$, $S(n) = 2.05$,

$R(n)/S(n) = 1.30$

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 3, you get 6 samples, each of size 3:

Block 2: $X_n = 5.7$, $W_1 = 6.3$, $W_2 = 5.7$, $W_3 = 0$

$R(n) = 6.3 - (0) = 6.3$, $S(n) = 4.92$,

$R(n)/S(n) = 1.28$

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 5, you get 4 samples, each of size 5:

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 5, you get 4 samples, each of size 4:

Block 1: $X_n = 6.0$, $W_1 = -4.0$, $W_2 = -3.0$,

$W_3 = -5.0$, $W_4 = 1.0$, $W_5 = 0$, $S(n) = 3.41$,

$R(n) = 1.0 - (-5.0) = 6.0$, $R(n)/S(n) = 1.76$

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 5, you get 4 samples, each of size 4:

Block 2: $X_2 = 4.4$, $W_1 = -4.4$, $W_2 = -0.8$,

$W_3 = -3.2$, $W_4 = 0.4$, $W_5 = 0$, $S(n) = 3.2$,

$R(n) = 0.4 - (-4.4) = 4.8$, $R(n)/S(n) = 1.5$

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 10, you get 2 samples, each of size 10:

R/S Statistic: An Example

- Suppose the original time series X(t) contains the following (made up) values:

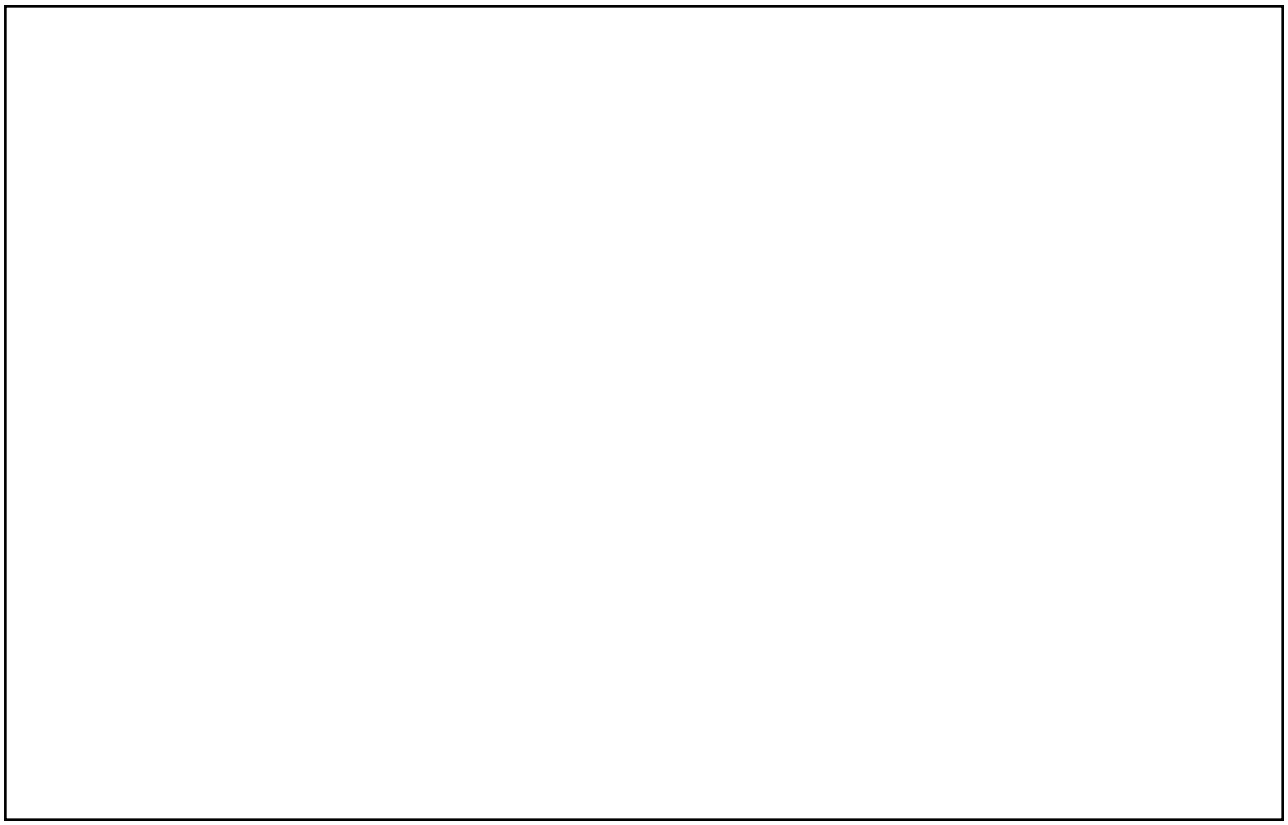
2 7 4 12 5 0 8 2 8 4 6 9 11 3 3 5 7 2 9 1

- For R/S analysis with n = 20, you get 1 sample of size 20:

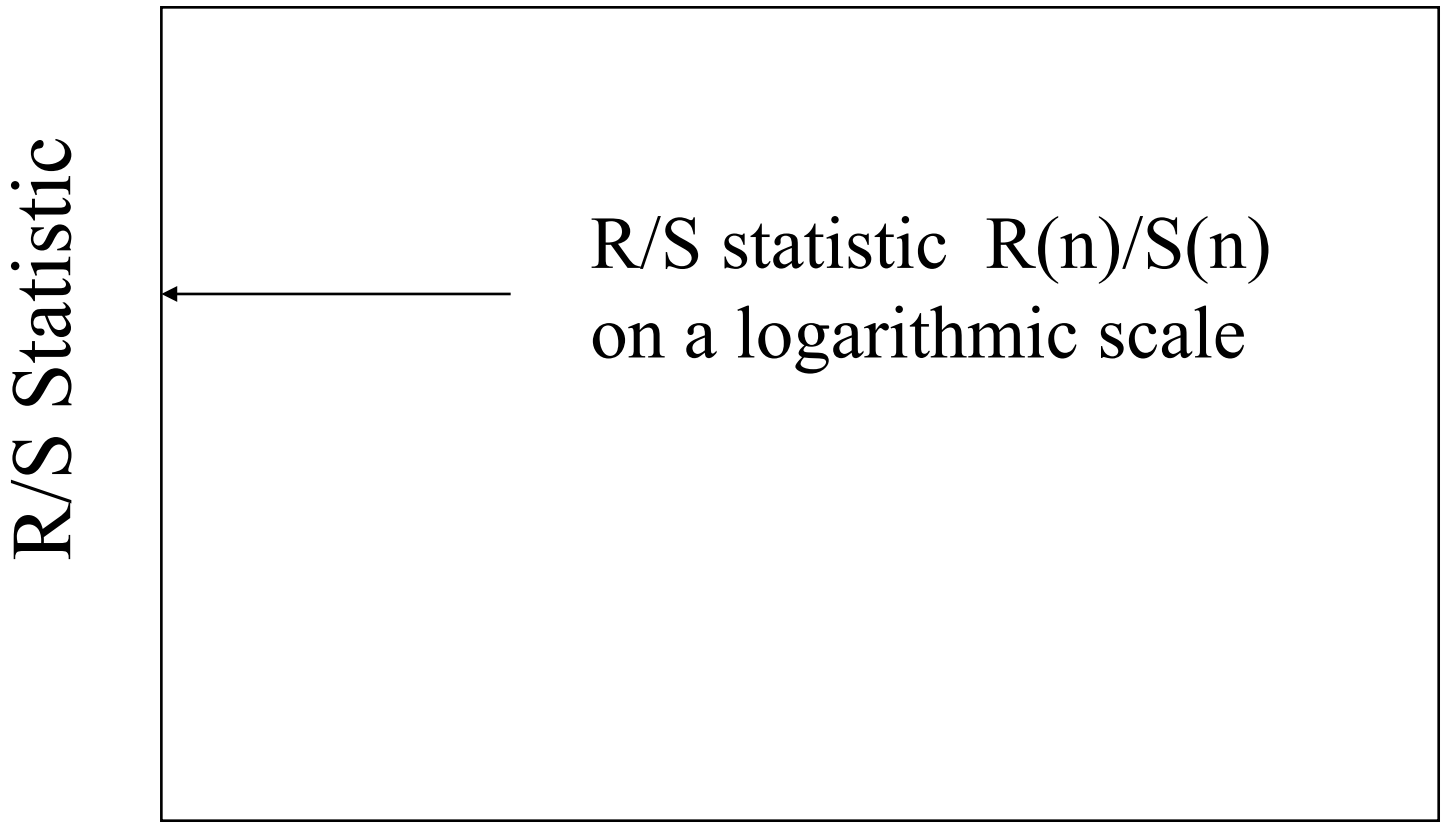
R/S Plot

- Another way of testing for self-similarity, and estimating the Hurst parameter
- Plot the R/S statistic for different values of n, with a log scale on each axis
- If time series is self-similar, the resulting plot will have a straight line shape with a slope H that is greater than 0.5
- Called an R/S plot, or R/S pox diagram

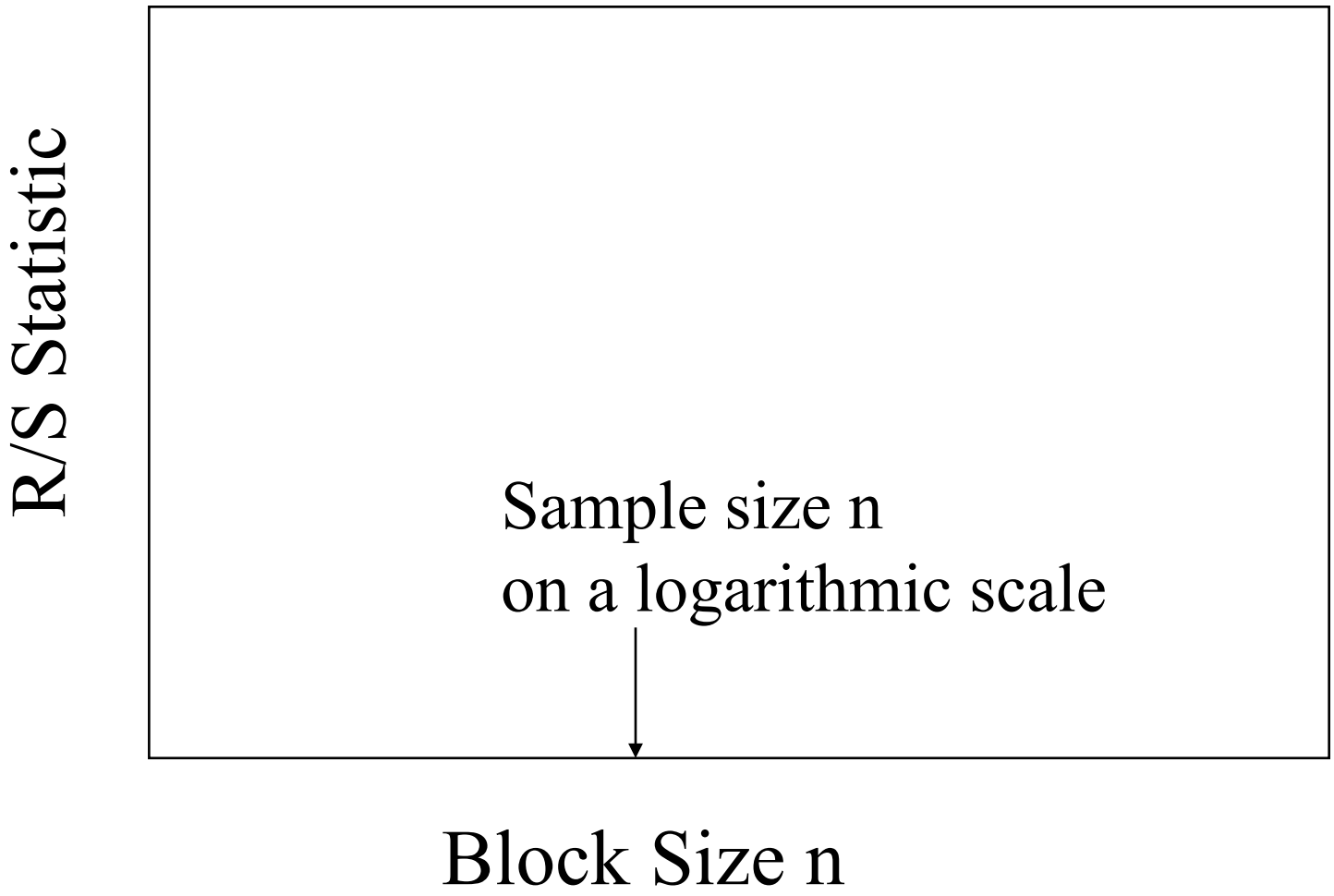
R/S Pox Diagram



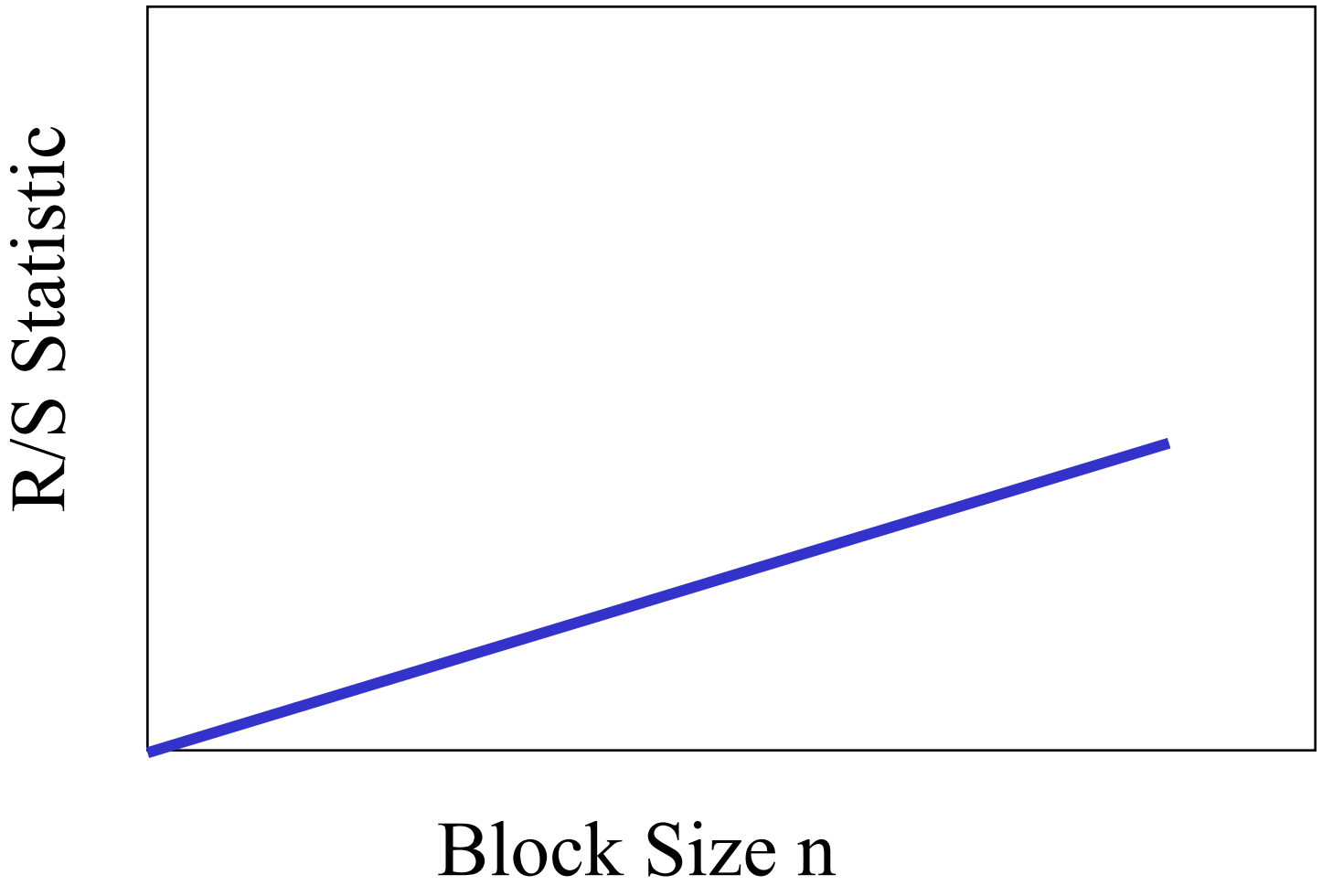
R/S Pox Diagram



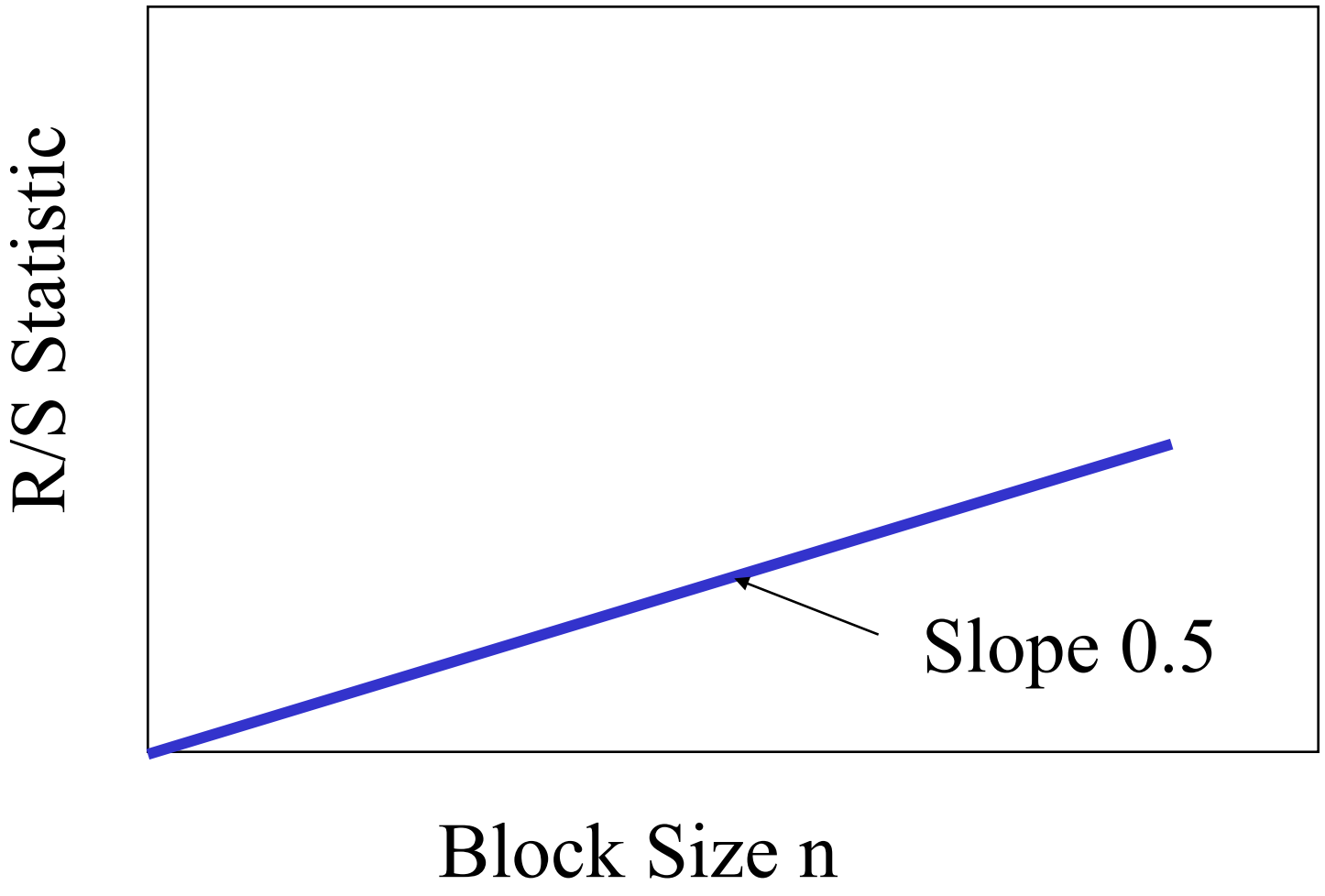
R/S Pox Diagram



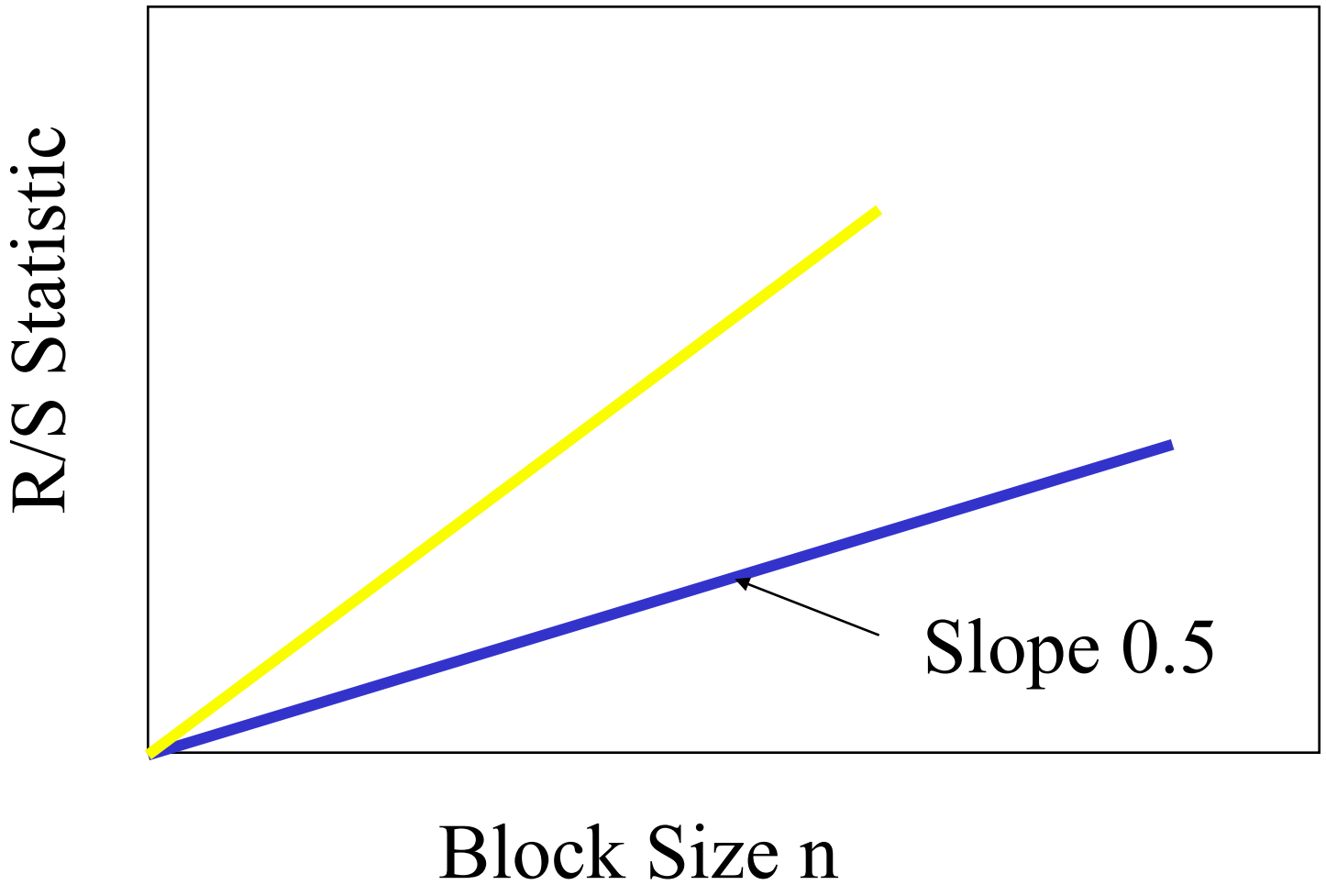
R/S Pox Diagram



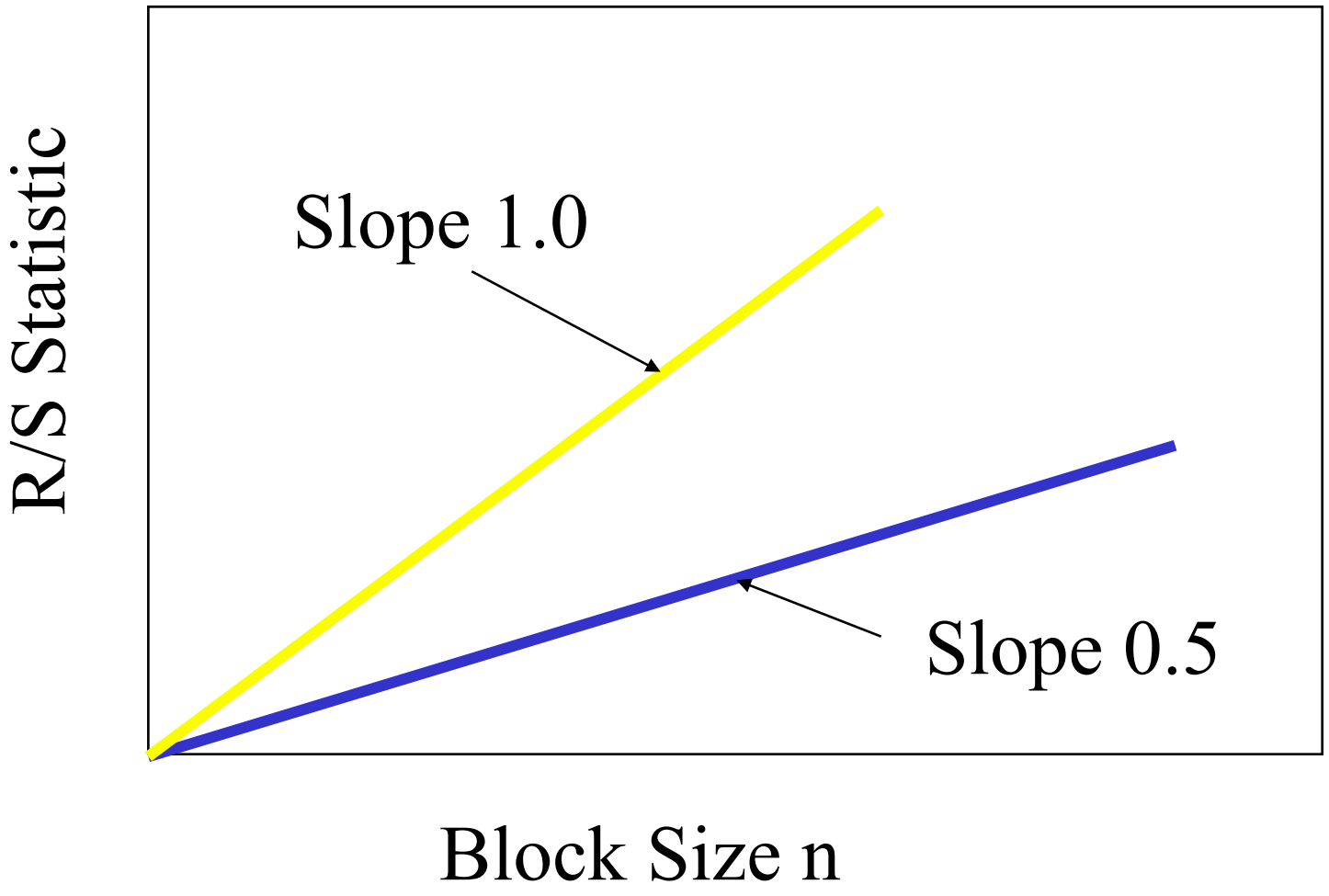
R/S Pox Diagram



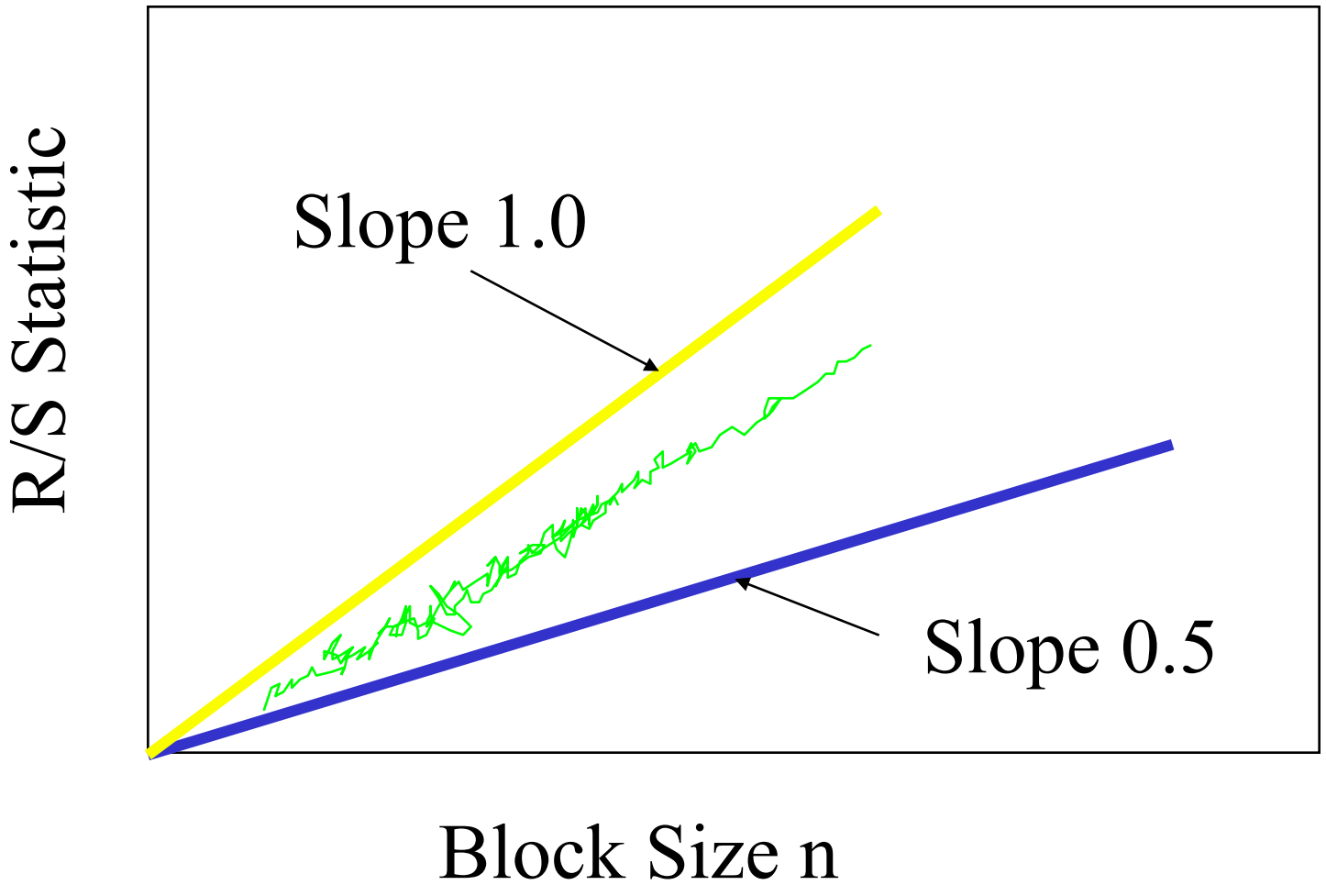
R/S Pox Diagram



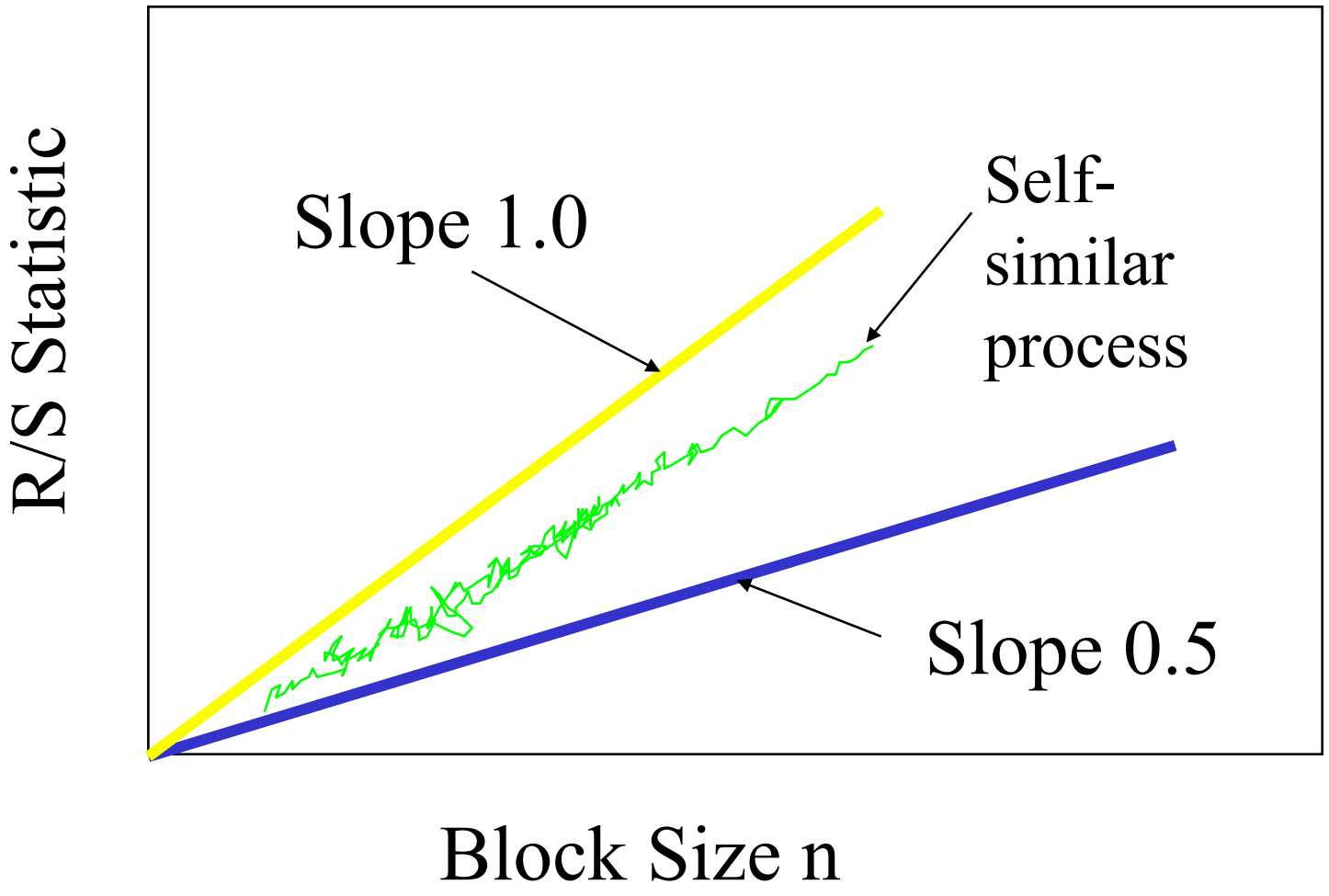
R/S Pox Diagram



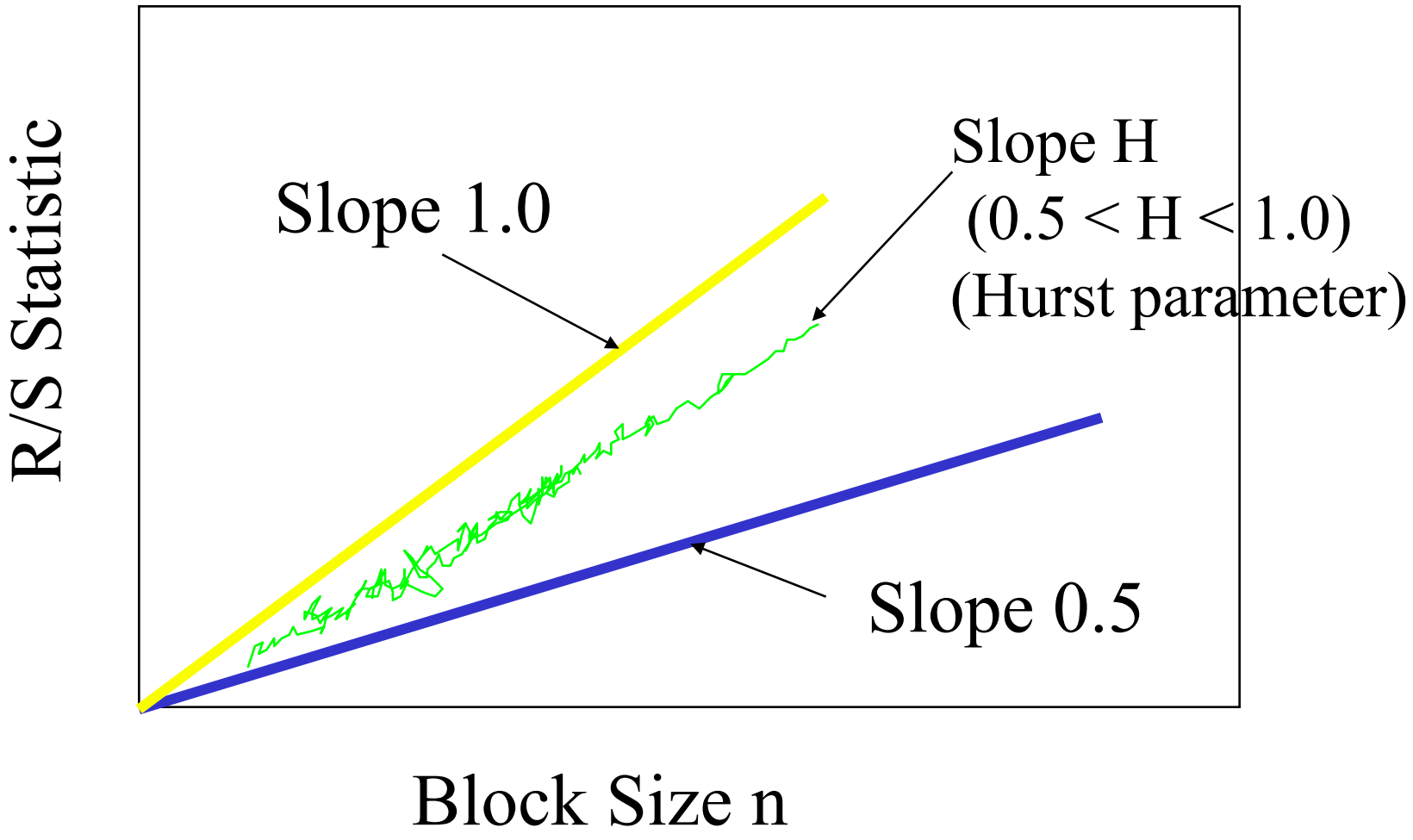
R/S Pox Diagram



R/S Pox Diagram



R/S Pox Diagram



Self-Similarity Summary

- Self-similarity is an important mathematical property that has recently been identified as present in network traffic measurements
- Important property: burstiness across many time scales, traffic does not aggregate well
- There exist several mathematical methods to test for the presence of self-similarity, and to estimate the Hurst parameter H
- There exist models for self-similar traffic

Newer Results

V. Paxson, S. Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling, IEEE/ACM* Transaction on Networking, 1995.

- TCP *session* arrivals are well modeled by a Poisson process
- A number of WAN characteristics were well modeled by *heavy tailed* distributions
- *Packet* arrival process for two typical applications (TELNET, FTP) as well as aggregate traffic is *self-similar*

Another Study

M. Crovella, A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes, IEEE/ACM* Transactions on Networking, 1997

- Analyzed WWW logs collected at clients over a 1.5 month period
  - First WWW client study
  - Instrumented MOSAIC
    - ~600 students
    - ~130K files transferred
    - ~2.7GB data transferred

Self-Similar Aspects of Web traffic

- One difficulty in the analysis was finding stationary, busy periods
  - A number of candidate hours were found
- All four tests for self-similarity were employed
  - $0.7 < H < 0.8$

Explaining Self-Similarity

- Consider a set of processes which are either ON or OFF
  - The distribution of ON and OFF times are heavy tailed
  - The aggregation of these processes leads to a self-similar process
- So, how do we get heavy tailed ON or OFF times?

Impact of File Sizes

- Analysis of client logs showed that ON times were, in fact, heavy tailed
  - Over about 3 orders of magnitude
- This lead to the analysis of underlying file sizes
  - Over about 4 orders of magnitude
  - Similar to FTP traffic
- Files available from UNIX file systems are typically heavy tailed

Heavy Tailed OFF times

- Analysis of OFF times showed that they are also heavy tailed
- Distinction between Active and Passive OFF times
  - Inter vs. Intra click OFF times
- Thus, ON times are more likely to be cause of self-similarity

Major Results from CB97

- Established that WWW traffic was self-similar
- Modeled a number of different WWW characteristics (focus on the tail)
- Provide an explanation for self-similarity of WWW traffic based on underlying file size distribution

Where are we now?

- There is no mechanistic model for Internet traffic
  - Topology?
  - Routing?
- People want to blame the protocols for observed behavior
- Multiresolution analysis may provide a means for better models
- Many people (vendors) chose to ignore self-similarity
  - Does it matter????
  - Critical opportunity for answering this question.

Overview of Simulation

- When do we prefer to develop **simulation model** over an analytic model?
  - When not all the underlying assumptions set for analytic model are valid.
  - When mathematical complexity makes it hard to provide useful results.
  - When “good” solutions (not necessarily optimal) are satisfactory.
- A simulation develops a model to numerically evaluate a system over some time period.
- By estimating characteristics of the system, *the best alternative* from *a set of alternatives under consideration* can be selected.

Overview of Simulation

- *Continuous simulation systems* monitor the system each time a change in its state takes place.
- *Discrete simulation systems* monitor changes in a state of a system at discrete points in time.
- Simulation of most practical problems requires the use of a computer program.

Overview of Simulation

- Approaches to developing a simulation model
  - Using add-ins to Excel such as @Risk or Crystal Ball
  - Using general purpose programming languages such as: FORTRAN, PL/1, Pascal, Basic.
  - Using simulation languages such as GPSS, SIMAN, SLAM.
  - Using a simulator software program.
- Modeling and programming skills, as well as knowledge of statistics are required when implementing the simulation approach.

Monte Carlo Simulation

- Monte Carlo simulation generates random events.
- Random events in a simulation model are needed when the input data includes random variables.
- To reflect the relative frequencies of the random variables, the *random number mapping* method is used.

JEWEL VENDING COMPANY –
an example for the random mapping technique

- Jewel Vending Company (JVC) installs and stocks vending machines.

- Bill, the owner of JVC, considers the installation of a certain product (“Super Sucker” jaw breaker) in a vending machine located at a new supermarket.

JEWEL VENDING COMPANY –
an example of the random mapping technique

- Data
  - The vending machine holds 80 units of the product.
  - The machine should be filled when it becomes half empty.
  - Daily demand distribution is estimated from similar vending machine placements.
    - P(Daily demand = 0 jaw breakers) = 0.10
    - P(Daily demand = 1 jaw breakers) = 0.15
    - P(Daily demand = 2 jaw breakers) = 0.20
    - P(Daily demand = 3 jaw breakers) = 0.30
    - P(Daily demand = 4 jaw breakers) = 0.20
    - P(Daily demand = 5 jaw breakers) = 0.05

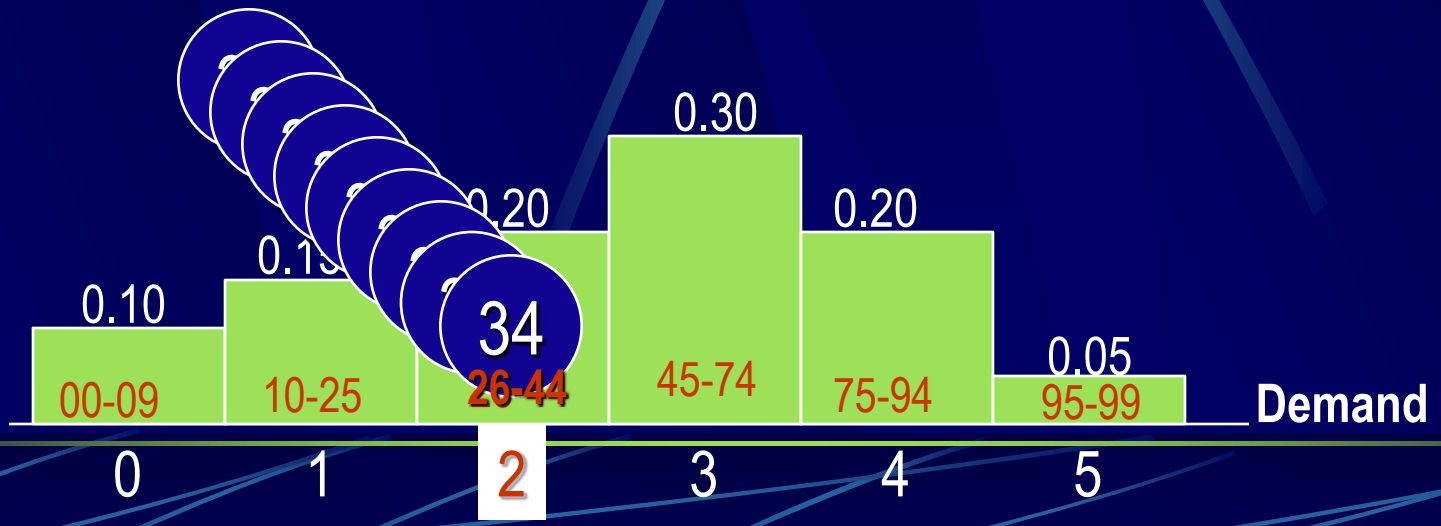
Bill would like to estimate the expected number of days it takes for a filled machine to become half empty.

Random number mapping – The Probability function Approach

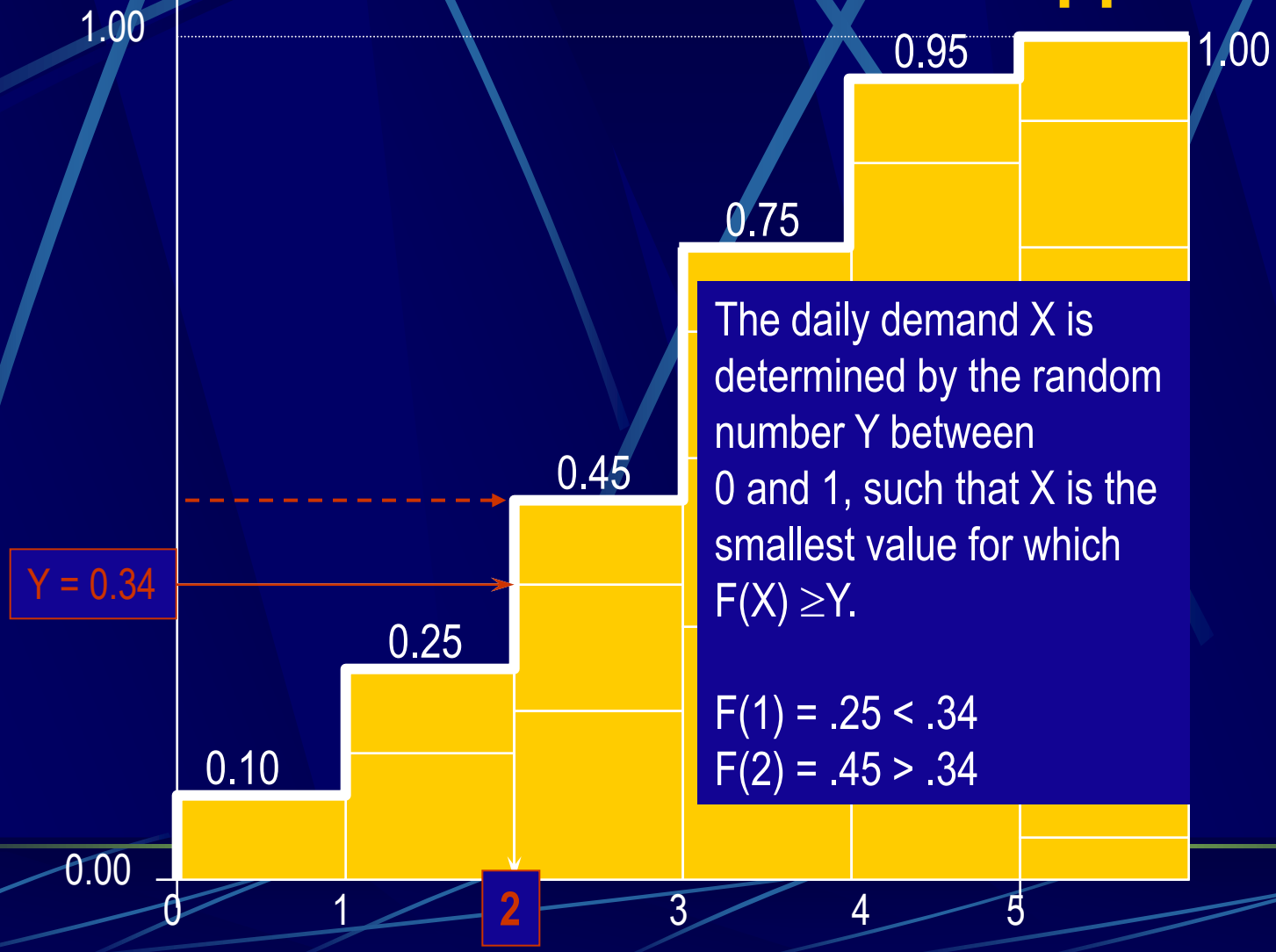
Random number mapping uses the probability function to generate random demand.

A number between 00 and 99 is selected randomly.

The daily demand is determined by the mapping demonstrated below.



Random number mapping – The Cumulative Distribution Approach



The daily demand X is determined by the random number Y between 0 and 1, such that X is the smallest value for which $F(X) \geq Y$.

$F(1) = .25 < .34$

$F(2) = .45 > .34$

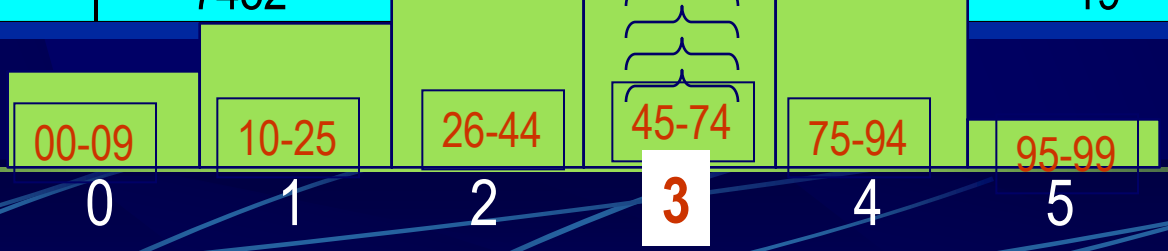
Simulation of the JVC Problem

- A random demand can be generated by hand (for small problems) from a table of pseudo random numbers.
- Using Excel a random number can be generated by
  - The RAND() function
  - The random number generation option (Tools>Data Analysis)

Simulation of the JVC Problem

- An illustration of generating a daily random demand.
- Since we have two digit probabilities, we use the first two digits of each random number.

Day	Random Number	Two First Digits	Demand	Total Demand to Date
1	6506	65	3	3
2	7761	77	4	7
3	6170	61	3	10
4	8800	88	4	14
5	4211	42	2	16
6	7452			19



Simulation of the JVC Problem

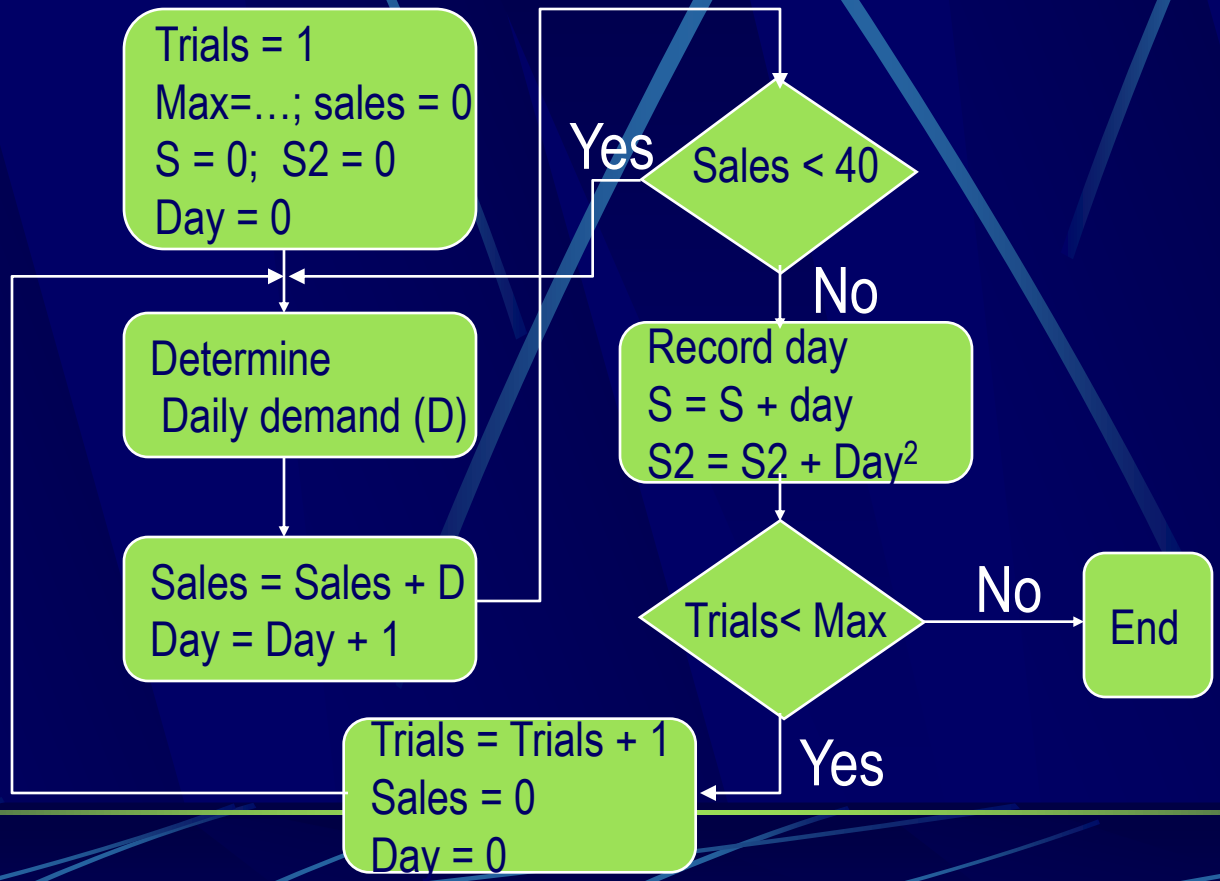
The simulation is repeated and stops once total demand reaches 40 or more.

Day	Random Number	Two First Digits	Demand	Total Demand to Date
1	6506	65	3	3
2				7
3				10
4				14
5				16
6	7452	74	3	19

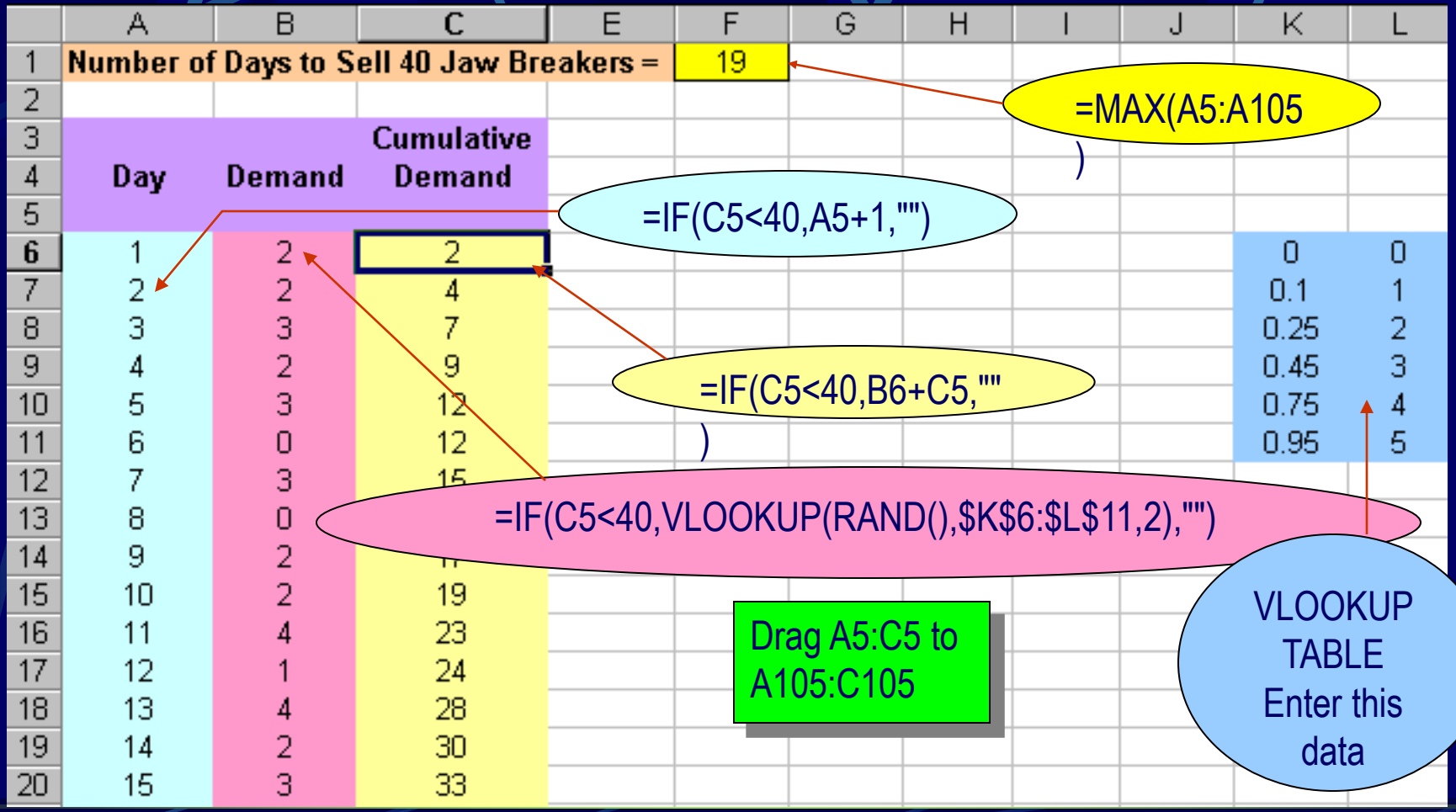
The number of "simulated" days required for the total demand to reach 40 or more is recorded.

JVC – A Flow Chart

Flow charts help guide the simulation program



JVC – Excel Spreadsheet



JVC – Excel Spreadsheet

	A	B	C	D	E	F	G	H
1	**Replication**	**Days**		*Days*				
2	1	18						
3	2	15		Mean	16.6			
4	3	14		Standard Error	0.541603			
5	4	18		Median	17		t	1.107823
6	5	17		Mode	18		p-value	0.296665
7	6	17		Standard Deviation	1.712698			
8					333			
9					627			
10					232			
11								
12					14			
13					19			
14					166			
15				Count	10			

=(E3-16)/E4

=TDIST(ABS(H5),9,2)

- The p-value =.2966… This value is quite high compared to any reasonable significance level.
- **Based on this data there is insufficient evidence to infer that the mean number of days differs from 16.**

Simulation of a Queuing System

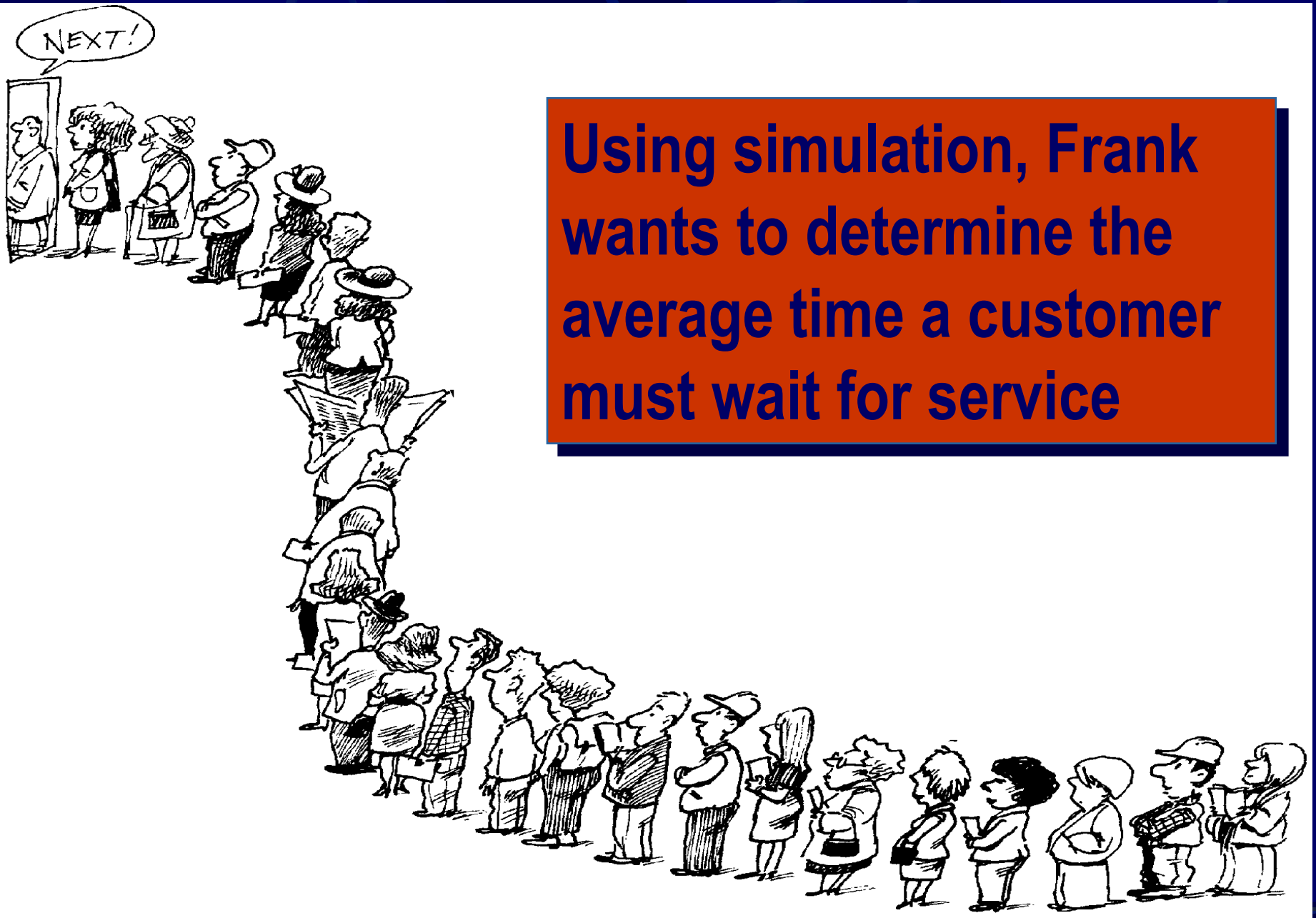
- In queuing systems time itself is a random variable. Therefore, we use the *next event simulation* approach.
- The simulated data are updated each time a new event takes place (not at a fixed time periods.)
- The *process interactive approach* is used in this kind of simulation (all relevant processes related to an item as it moves through the system, are traced and recorded).

Simulation of an M / M / 1 Queue

- Applying the *process interaction approach* we have:
  - New arrival time = Previous arrival time + Random interarrival time.
  - Service finish time = Service start time + Random service time.
  - A customer joins the line if there is a service in progress (its arrival time < current service finish time).
  - A customer gets served when the server becomes idle.
  - Waiting times and number of customers in line and in the system are continuously recorded.

LANFORD SUB SHOP
An example of the M/M/1 queuing simulation

- Lanford Sub Shop sells sandwiches prepared by its only employee, the owner Frank Lanford.
- Frank can serve a customer in 1 minute on the average according to an exponential distribution.
- During lunch time, 11:30 a.m. to 1:30 p.m., an average of 30 customers an hour arrive at the shop according to a Poisson distribution.



Using simulation, Frank wants to determine the average time a customer must wait for service

LANFORD SUB SHOP - Solution

- Input Data

  $\lambda = 30, \ \mu = 60.$
- Data generated by the simulation:
  - C# = The number of the arriving customer.
  - R#1 = The random number used to determine interarrivals.
  - IAT = The interarrival time.
  - AT = The arrival time for the customer.
  - TSB = The time at which service begins for the customer.
  - WT = The waiting time a customer spends in line.
  - R#2 = The random number used to determine the service time.
  - ST = The service time.
  - TSE = The time at which service end for the customer

LANFORD SUB SHOP – Simulation for first 10 Customers

Arrival Time | Time Service Begins | Service Time | Time Service Ends

C#	R#1	IAT	AT	TSB	WT	R#2	ST	TSE
1	0.6506	2.10	2.10	2.10	0	0.7761	1.5	3.6
2	0.6170	1.92	4.02	4.02	0	0.8800	2.12	6.14
3	0.4211	1.09	5.11	6.14	1.03	0.7452	1.37	7.51
4	0.1182	0.25	5.36	7.51	2.15	0.4012	0.51	8.02
5	0.7265	2.59	7.95	9.01	1.06	0.6299	0.99	9.01
6	0.5141	1.99	7.35	9.12	1.77	0.1085	0.11	9.12
7	0.4573	1.73	9.08	10.31	1.23	0.6969	1.19	10.31
8	0.1696	0.37	7.76	10.31	2.55	0.0267	0.03	10.34
9	0.3175	0.76	8.52	10.34	1.82	0.7959	1.59	11.93
10	0.4958	1.37	9.89	11.93	2.04	0.4281	0.56	12.49

Average waiting time =
(0 + 0 +1.03 + ... +2.04) / 10 = 1.59

The interarrival time = - ln(1-0.4211) ⁄ 30 = 0.0182 hours = 1.09 minutes

The explicit inverse method

End of service = 6.14+1.37

Waiting time = 6.14 - 5.11

Arrival time of customer 3 = Arrival time of customer 2 + 1.09 = 4.02 + 1.09

LANFORD SUB SHOP – Simulation for first 1000 Customers

	A	B	C	D	E	F	G
1	Arrival Rate (lambda) per hour =			30			
2	Service Rate (mu) per hour =			60			
3							
4	Average Waiting Time (in minutes) =				0.970982		
5							
6	Cust #	IAT	AT	TSB	WT	ST	TSE
7							
8	1	3.185907	3.185907	3.185907	0	0.161422	3.34733
9	2	0.907386	4.093294	4.093294	0	1.169643	5.262937
10	3	0.448395	4.541689	5.262937	0.721248	0.191208	5.454144
11	4	0.01861	4.560299	5.454144	0.893845	0.860468	6.314613
12	5	1.188251	5.748551	6.314613			
13	6	0.66342	6.411971	7.601137			
14	7	0.906348	7.318318	9.994628			
15	8	1.205453	8.523771	10.00124			
16	9	4.724972	13.24874	13.24874			
17	10	1.362563	14.61131	14.61131			
18	11	1.208411	15.81972	17.64179			
19	12	5.357083	21.1768	21.1768			
20	13	0.79991	21.97671	22.38254			

Cell	Value	Formula
E4	Wq	=AVERAGE(E8:E1007)
Row 8		
A8	Customer Number	=A7+1
B8	Customer Inter-arrival Time	=-LN(1-RAND())/($D$1/60)
C8	Arrival Time	=B8+C7
D8	Time Service Begins	=MAX(C8,G7)
E8	Waiting Time	=D8-C8
F8	Service Time	=-LN(1-RAND())/($D$2/60)
G8	Time Service Ends	=D8+F8