Examples

(*notes: textbook* - *Chapter 2*)

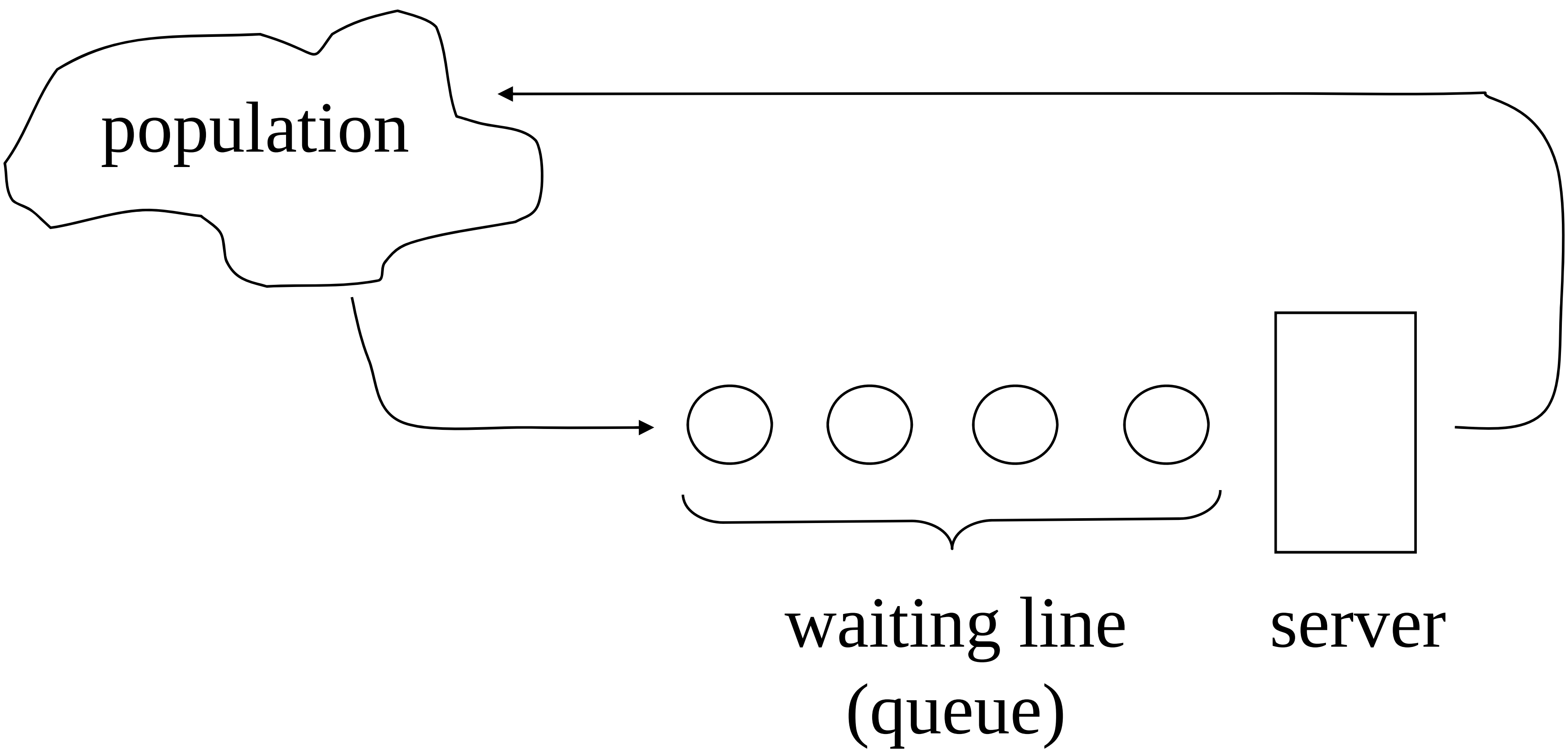
Basic Procedure
for hand calculation

- Determine the input characteristics
- Use this information to construct a "simulation table" typically in the format:

Repetition #	Inputs $x_1, x_2, . . ., x_n$	Response y
1		
2		
3		
.		
.		
.		
t		

- For each repetition, generate values for the inputs and calculate the response

Example 1
single server queue



Example 1: Assumptions

- The ***population*** is infinite
  - the arrival rate is unaffected by population size
- ***Arrivals*** are one at a time (in random fashion) in accord with some (fixed) probability distribution
- ***Service times*** are of some (random) length in accord with some (fixed) probability distribution
- System ***capacity*** is unlimited
- ***Service*** is in order of arrival (FIFO - First-In, First-Out)

System Components for Example 1

- **Entity** (item of interest)
- **Attribute** (characteristic of an entity)
- **Activity** (process causing change, given by a period of specified length)
- **Event** (discrete occurrence that may change system state)
- **State** (collection of variables needed to describe the system at any point in time that an event occurs in order to handle the event)

Entities and Attributes

- **Entity** (item of interest)
  - customer
  - server
- **Attribute** (characteristic of an entity)
  - service needed (customer)
  - mean service time for a service (server)

Activities and Events

- **Activity** (process causing change, given by a period of specified length)
  - preparing service materials (customer)
  - serving customer (server)
- **Event** (discrete occurrence that may change system state)
  - arrival of a customer (endogenous event)
  - initiation of service (endogenous event)
  - completion of service (endogenous event)

Simulation State

- **State** (collection of variables needed to describe the system at any point in time that an event occurs in order to handle the event)
  - server state (busy or idle)
  - customer queue (first and last, empty or not empty)
- Remarks:
  - on an arrival event, the server state is needed to determine whether to queue the customer or have an initiation of service event.
  - on a completion of service event, the queue state is needed to determine whether to have an initiation of service event or go idle

Food for thought: In providing outcomes information for average customer queue times, if a customer arrives and the server is idle, do we log 0 time on queue? i.e., do we reflect the customer as having been in line for 0 time?

Event Handling

Departure event → another customer waiting?
- yes → remove the customer from queue → serve the customer
- no → begin server idle time

Arrival event → server busy?
- yes → customer enters queue for service
- no → serve the customer

System State

NOTE: if an arrival and a departure occur simultaneously and the arrival is considered first, then the customer is put on queue with 0 wait time! This means some care must be exercised in how such "ties" are broken in the simulation implementation.

- At the point an arrival event occurs

server \ queue	not empty	empty
BUSY	enter queue	enter queue
IDLE	NOT POSSIBLE	enter service

- At the point a departure event occurs

server \ queue	not empty	empty
BUSY	remain BUSY	NOT POSSIBLE
IDLE	NOT POSSIBLE	remain IDLE

Trial Run for Example 1
Customer Arrivals

- Assume *uniformly distributed* **interarrival** times (times between consecutive customers) with the values 1 through 6
  - Can be generated by rolling a 6-sided die
  - To be uniformly distributed, each possible interarrival time (1,2,3,4,5,6) must be equally likely, and so has the same probability (1/6)
  - The expected value of the interarrival time is

$$E = \left(\sum_{1}^{6} \frac{1}{6} \times i \right) = \left(\sum_{1}^{6} i \right) \times \frac{1}{6} = \frac{21}{6} = 3.5$$

Trial Run for Example 1
Customer Service Times

- Assume uniformly distributed among 1, 2, 3, 4
  - The expected value of the service time is 10/4 = 2.5
  - For model **stability**, the expected service time must be less than the expected interarrival time
    - As now configured, the trial run is stable

Trial Run for Example 1
Customer Attributes

- Suppose 6 customers are introduced to the system, with randomly obtained interarrival times and service times (acquired randomly, say by rolling a die) as follows:

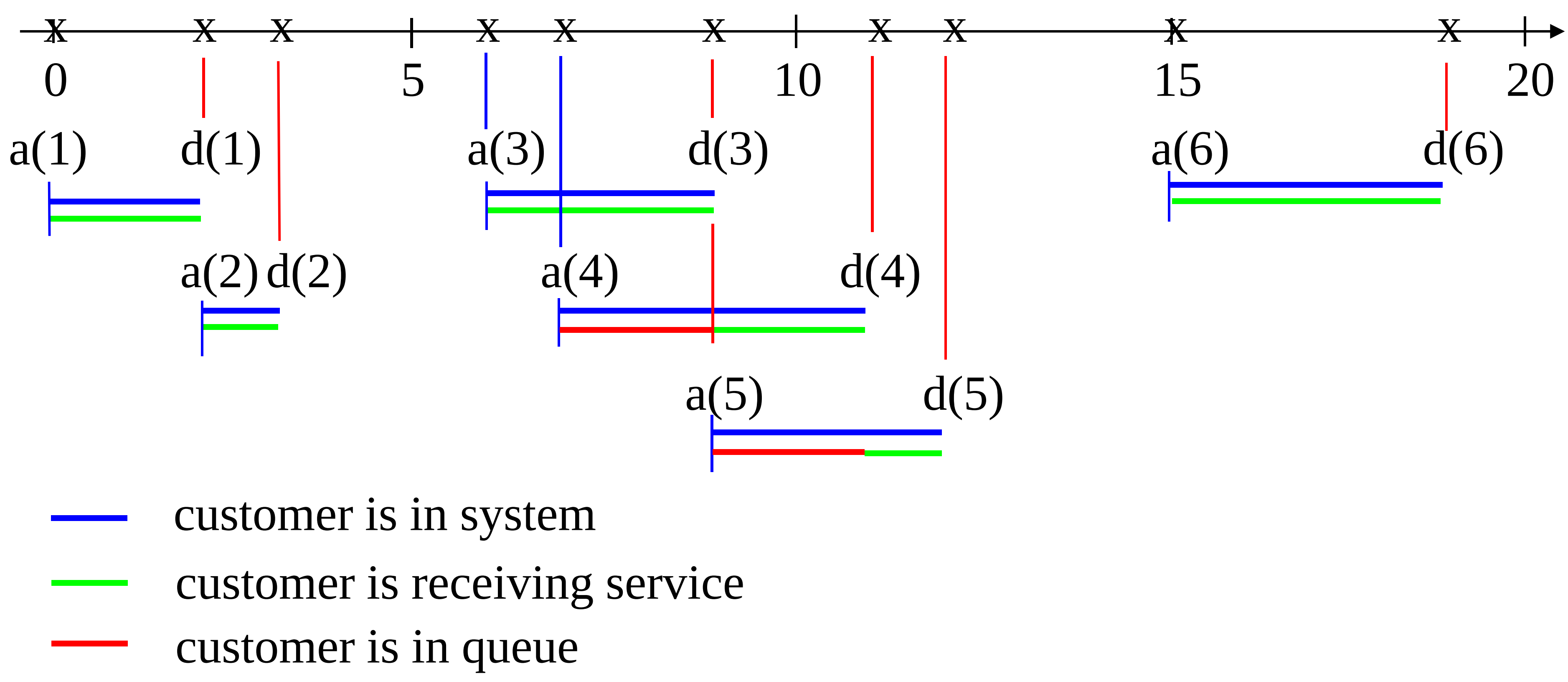
customer	interarrival time	service time
1	-	2
2	2	1
3	4	3
4	1	2
5	2	1
6	6	4

Trial Run for Example 1

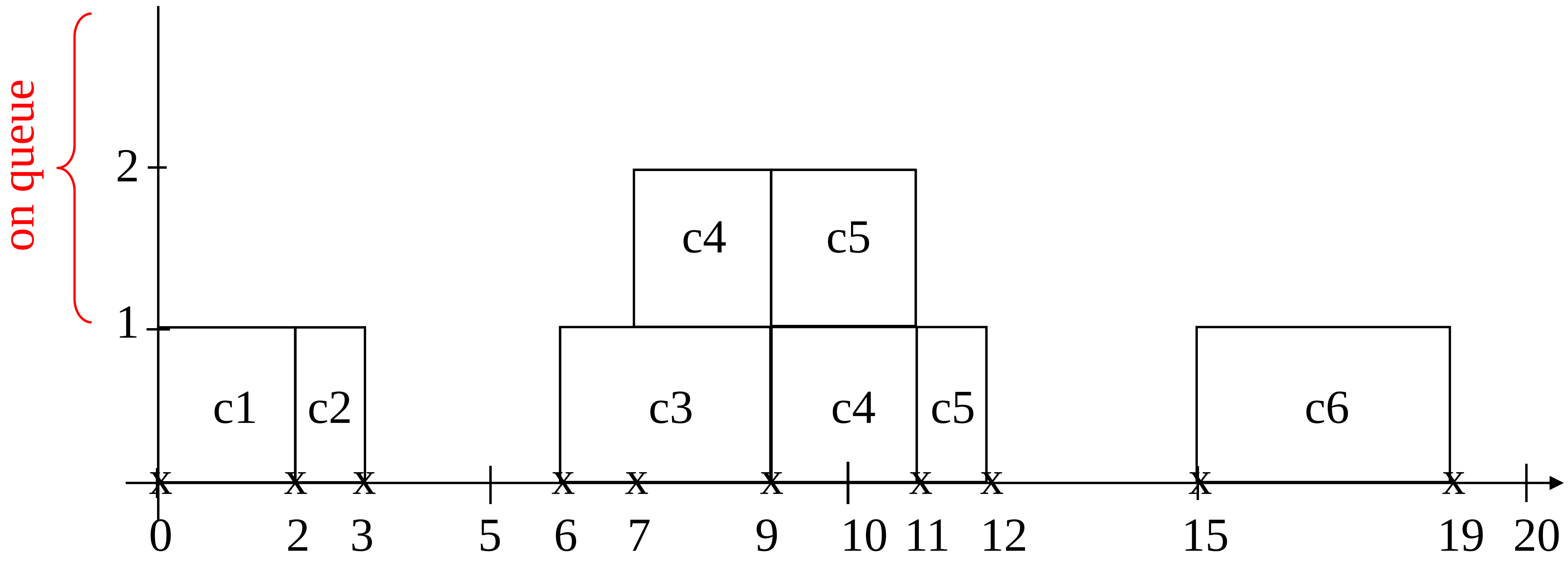
Simulation Table

customer	arrival (clock)	service time	service begins (clock)	service ends (clock)	*iat*
1	0	2	0	2	-
2	2	1	2	3	*2*
3	6	3	6	9	*4*
4	7	2	9	11	*1*
5	9	1	11	12	*2*
6	15	4	15	19	*6*

Trial Run for Example 1
Timeline of Simulation



Trial Run for Example 1
Graphical Representation



Trial Run for Example 1
Simulation Statistics

- *Average waiting time* (for those who waited):

$$\frac{\text{time in queue (aggregate)}}{\text{\# of customers who waited}} = \frac{2+2}{2} = 2$$

- *Probability a customer had to wait*:

$$\frac{\text{\# who waited}}{\text{\# of customers}} = \frac{2}{6} = 0.33$$

- *Percentage of time the server was idle*:

$$\frac{\text{idle time}}{\text{total time}} = \frac{6}{19} = 32\%$$

- *Average time in system* (time in queue + service time):

$$\frac{\sum \text{customer times}}{\text{\# of customers}} = \frac{2+1+3+4+3+4}{6} = \frac{17}{6} = 2.8$$

Trial Run for Example 1
Input Characteristics (*can be predicted*)

- Average service time:

$$\frac{\text{total service time}}{\text{\# of customers}} = \frac{13}{6} = 2.2 \quad (2.5 \text{ is the expected value})$$

- Average time between arrivals:

$$\frac{\sum \text{interarrival times}}{\text{\# of customers-1}} = \frac{15}{5} = 3.0 \quad (3.5 \text{ is expected})$$

Extending Example 1

- Suppose the system in Example 1 is extended by the addition of another server (with a different service distribution)
  - The customer service time may depend on which server gets the customer.
  - An added *rule* is needed for the case when *both* servers are idle
  - There are also two new events (initiation of service for the new server and completion of service for the new server)
- By identifying the servers as A and B, the *effect* of the "rule" can be tested/inferred by using simulation runs to examine how busy each server is

"Real" System Scenario That Fits the Context of Example 1

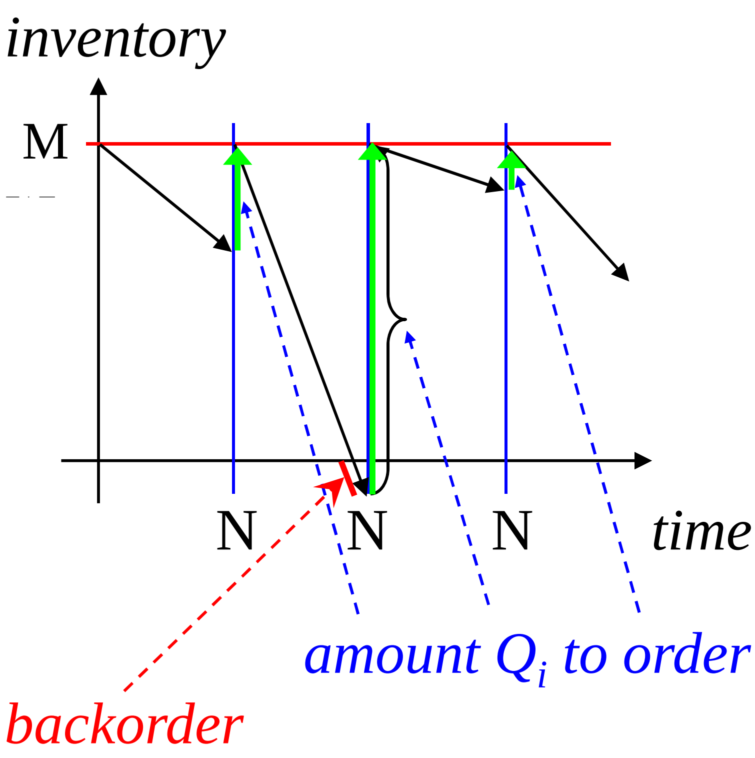
- Scenario
  - A small grocery store has only one checkout counter. Customers arrive at this counter at random from 1 to 8 minutes apart. Each possible value of interarrival time has the same probability of occurrence. Service time may vary from 1 to 6 minutes according to the distribution:

Service time	Probability
1	.1
2	.2
3	.3
4	.25
5	.1
6	.05

- It is easy to obtain the expected value when a table like the one above is given. In this case, it is $1\times0.1 + 2\times0.2 + 3\times0.3 + 4\times0.25 + 5\times0.1 + 6\times0.05 = 3.2$
- This is an easy statistic to gather in implementing a model for this system, providing a "reality check" regarding basic model integrity

Other Examples
Inventory System

- Scenario
  - Every N units of time, inventory is checked, at which time an order is made to bring inventory up to level M



Assumptions:

Lead time is 0 between making and receiving an order. Demand is uniform over each N unit review interval (linear decrease).

Inventory Model Considerations

- There are costs associated with
  - Excess inventory
  - More frequent review (making N smaller)
  - Running short on inventory
- The model can measure performance (cost/profit) with M and N adjusted as inputs
- At the end of each period *i*, an order quantity

  $$Q_i = \mathrm{M} - inventory$$

  is made
- Simplifying Assumptions:
  - Lead time is 0 (time between making and receiving an order)
  - Demand is uniform over a review period (linear decrease)

Other Examples
Vendor System

- Scenario
  - A newspaper seller buys papers for 43¢ and sells them for 75¢. Unsold papers go for scrap at 7¢. Papers are purchased in bundles of 10. There are 3 types of newsdays: "good", "fair", and "poor" with probabilities 0.35, 0.45, and 0.2. The demand distributes as follows:

Demand	good day	fair day	poor day
40	.03	.10	.44
50	.05	.18	.22
60	.15	.40	.16
70	.20	.20	.12
80	.35	.08	.06
90	.15	.04	.00
100	.07	.00	.00
	1.00	1.00	1.00

may want to interpolate

Vendor Model Considerations

- Question: What is the optimal number of papers to purchase?

  potential profit =

  revenue - cost of newspapers - $\underbrace{\text{lost profit from excess demand}}_{\text{debatable: 32¢ per copy}}$ + salvage

- Procedure
  - Use as input the number of papers purchased
  - run the simulation over an extended period (say a month) to estimate the profit
  - Increase the purchase until a profit decrease occurs.

Other Examples

Reliability Problem

- Scenario

A large milling machine has 3 different bearings that fail according to the table

bearing life	probability
1000	0.10
⋮	⋮
1900	0.05
	1.00

When a bearing fails, the mill stops and a repair procedure is invoked to install a new bearing. Down time costs approximately \$15 per minute. On site repair costs \$59 per hour. It takes 20 minutes to change 1 bearing, 30 minutes for 2, and 40 minutes for all 3. Bearings cost \$56 each. Is it cost effective to replace all 3 bearings whenever a bearing fails? Assume the delay time prior to arrival of the repair service is given by:

delay (minutes)	probability
5	0.6
10	0.3
15	0.1

Reliability Model Considerations

- Model each bearing: since replacement protocols are what is being tested, a simulation decision must be made regarding the "bearing pool"
  - Does it matter when we repeat the simulation with alternate protocols whether or not we use the same pool?
  - Or do we count on the law of averages to cover us? (probably a safe assumption in this case).

UNF

Reliability Model
Simulation Procedure

- Simulate long enough for at least 12 repair events, say 20,000 hours
- Model
  - The current repair tactic (bearings treated separately)
  - The proposed repair tactic (or other variations)
  - Compare the simulation outcomes.

Reliability Model

Simulation Setup

- Bearing treated separately - assume no two bearings ever fail simultaneously

Bearing 1 life	Bearing 1 accum life	Bearing 1 repair delay	Bearing 2 life	Bearing 2 accum life	Bearing 2 repair delay	Bearing 3 life	Bearing 3 accum life	Bearing 3 repair delay

- Bearings as a group

Bearing life 1	Bearing life 2	Bearing life 3	hours to failure (min of bearing life)	repair delay

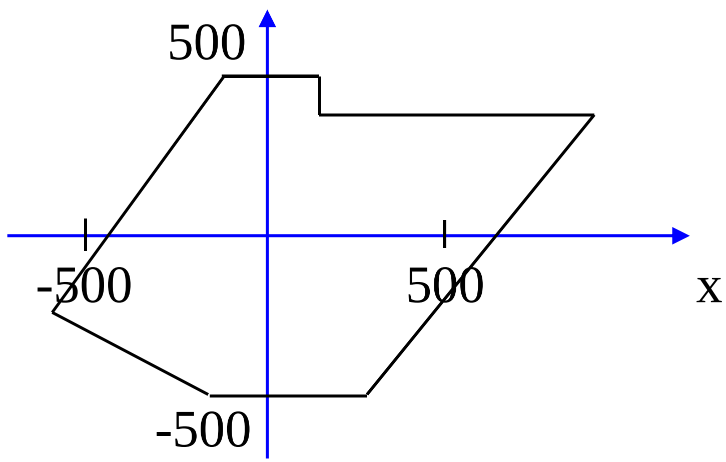
step through 20,000 hours event to event

Other Examples

Bomb Dispersal

- Scenario

A squadron of bombers is attempting to destroy an ammunition depot shaped as follows:



The aiming point is (0,0), but a bomb which falls anywhere within the target is scored a hit. Assume that the bombs are normally distributed in the x direction with a standard deviation of 600 and are normally distributed in the y direction with a standard deviation of 300 (mean of the distribution is 0 in each case). Ten bombers are in a squadron, and each drops 1 bomb. How many of a squadron's bombs may we expect to hit the target in a bombing run (may also be handled analytically).

Bomb Dispersal Model

Simulation Setup

Bomber	random normal x	random normal y	result (hit or miss)
1			
2			
.			
.			
.			
10			