CHAPTER 5: SIMULATION MODELS

$(\Omega, \mathcal{F})$ and $\{\mathcal{F}_t ; t \in T\} = \mathbb{F}$ a filtration, $\mathcal{F}_t \subset \mathcal{F}$.

Consider the problem of simulating a stochastic process $\{X_t ; t \in T\}$ with the goal of evaluating one (or several) performance functions

$$J = \mathbb{E}(\phi(X_t ; t \in T)),$$

where ϕ is a functional of the whole trajectory $X_t ; t \in T$, that takes values in $\mathbb{R}$.

Remark: Because each trajectory $(X_t(\omega) ; t \in T)$ is uniquely defined for each ω, it follows that $\phi(X_t(\omega) ; t \in T)$ is a well defined random variable on $(\Omega, \mathcal{F})$.

Example: Let X_n be the amount of apples kept in store for sale in the cafeteria. The manager wishes to evaluate the policies of ordering and reducing prices for quick sales. Assume that at the end of a period, any remaining apples must be discarded. Demand $\{d_n(u)\}$ is assumed to be iid, given a price u per apple (the mean depends on price). Ordering costs are of the form $K + p\,y_n$, where y_n is the number of apples ordered at the end of period n.

Here $X_{n+1} = y_n$, and the cost is:

$$C_n = K + X_n p - \max(X_n, d_n(u)) \cdot u$$

A simulation can be done by generating $\{(X_n, d_n(u))\}$ and then the running costs can be calculated using

$$\frac{1}{N} \sum_{n=1}^{N} C_n(X_n, u).$$

- ~~Continuous~~ model for simulation (tick-based)
- Discrete event model (event-based)
- Standard clock model
- Reduced models (Petri-nets, transformations)

(a1). ~~Continuous~~ Tick-based Simulation Model (Tick-simulation)

Origin of model in deterministic context: numerical reproduction of trajectories of a dynamic physical system (computer animation for billiard or pool games). Here the model is described via ODE's (or PDE's more generally):

$$\frac{dx_t}{dt} = v(x, t) \qquad x_t \in \mathbb{R}^d \ ; \ T = (0, T]$$

$x_0 \in \mathbb{R}^d$ is known as the initial position.

$v : \mathbb{R}^d \times T \to \mathbb{R}^d$ is called a vector field or "drift" and in mechanical systems it represents the velocity at each point in space.

Step-by-step or tick-wise animation uses a discretization of time into "ticks" or small units of length $h > 0$:

$$x(i+h) = x(i) + v(x(i), ih)$$

$i = 1, \dots \lfloor T/h \rfloor$, $x(0) = x_0$.

Thm: If v is continuous and bounded, then the piecewise linear interpolation x_t^h of the sequence $\{x(i)\}$ converges, as $h \to 0$, in the sup-norm to the solution of the ODE for every initial condition $x_0 \in \mathbb{R}^d$.

The proof of this result follows from Ascoli-Arzelà Theorem.

Example : Consider the Black-Scholes model of a geometric Brownian motion for the stock price:

$$S_t = S_0 e^{\mu t + \sigma B_t} \quad ; S_0 \in \mathbb{R}$$

where $B(\cdot)$ denotes the standard Brownian motion.

[Def: A stochastic process $\{B(t); t \geq 0\}$ on $(\Omega, \mathcal{F}, \mathbb{P})$ is called a standard Brownian motion if:

(i) $B(0) = 0$ a.s.
(ii) $\{B(t+s) - B(t)\}$ is independent of $\mathcal{F}_t$ $\forall t, s \geq 0$
(iii) For all $t, s \geq 0$ $\{B(t+s) - B(t)\} \sim N(0, s)$ (normal dist.)

]

Using a discretization, the simulation by ticks considers the discrete-time sampling:

$$S(i+1) = S(i) \exp\{\mu h + \sigma h Z_i\}, \; i = 1, \dots \lfloor T/h \rfloor$$

where $\{Z_i\}$ are iid $\mathcal{N}(0,1)$.

A Financial option or derivative on the asset is of the form:

$$\max(0, \phi(\{S_t; t \leq T\}))$$

and can be approximated using the discretization.

A European option: $\phi(\{S_t; t \leq T\}) = S_T - K$

An American option: $\frac{1}{T} \int_0^T S_t \, dt$ // Barrier: $(S_T - K)\mathbb{1}_{(S_t \geq B; t \leq T)}$

A Bermudan option: $\frac{1}{T} \sum^{\lfloor T/h \rfloor} S_{nh}$ // etc.

```
N = INT(T/h), S[0] = S0, C = exp(μh)
FOR i=1,... N
  Generate Z ~ N(0,1)
  S[i] = S[i-1] * C * exp(σhZ)
```

To make it more efficient, work with log-prices to avoid exponentiation calculations:

$$X[i] = X[i-1] + \sigma * h * Z \quad \text{in loop.}$$

then $S[i] = C * \exp(X[i])$ can be calculated after

→ How do we know that this will work?

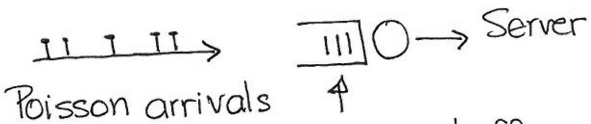
Def : Let $(X(t); t \geq 0)$ be a stochastic process on $(\Omega, \mathcal{F}, \mathbb{P})$ with $X(0) = x_0$ a given random variable. A ~~continuous~~ tick-based simulation model for the process is a family of stochastic processes in discrete time, indexed by $h > 0$: $\{X^h(n); n \in \mathbb{N}\}$ with $X^h(0) = x_0$ $\forall h$, that satisfies:

$$\forall t \in (0, T] \lim_{\substack{n \to \infty: \\ nh = t}} X^h(n) \Rightarrow X(t).$$

The symbol "$\Rightarrow$" means convergence in distribution.

Remark: most performance functions are Lipschitz continuous and convergence of $\{X^h(\cdot)\}$ is sufficient to ensure convergence of the corresponding discrete performance function, but in general, this requires verification.

Example: Queueing model FCFS



$N(t)$: number of arrivals up to time t ($\sim$ Poisson(λt))

$\{\xi_i\}$ iid $\sim G$ are the consecutive service times.

Example of performance or objective functions are : mean queue length, probability that the queue size is larger than a given value, probability that the waiting times of clients is larger than a certain value, mean idle time of server, etc.

How do we build a continuous simulation?

Thm: Let $\{A_n^h\} \sim$ iid Bernouilli random variables with parameter $p = \lambda h$, for $h > 0$ small enough so that $\lambda h < 1$. Let X_n^h be the number of arrivals, $X_n^h \sim Bin(n, \lambda h)$. Then for any $t \in \mathbb{R}$,

$$X_n^h \underset{\substack{h\to 0\\ n = t/h}}{\Longrightarrow} \text{Poisson}(\lambda t).$$

[proof in references, wikipedia, etc ...].

For fixed h (we will drop the subscript notation) we have $n = 1, \ldots \lfloor T/h \rfloor$

$A_n \sim Ber(\lambda h) \in \{0, 1\}$

D_n : number of service completions in $(nh, (n+1)h]$.

$Q(n)$: queue size at start of period.

Consecutive queue lengths satisfy:

$$Q(n+1) = Q(n) + (A_n - D_n).$$

Is this enough information? How can we know what D_n is?

~~At arrival time of i-th client, that is~~

$$i = \min \left(n : \sum_{k=1}^{n} A_k = i\right)$$

we can generate the random variable ξ_i. But this will complicate the code with all concurrent arrivals in queue having to store values. Another possibility:

Augmentation of state space to include memory in service:

Let R_n = residual service time at period n.

Then:

$$R_{n-1} = \begin{cases} R_n - 1 & \text{if } R_n \geq 1 \\ \lfloor \xi_{n+1}/h \rfloor \ \cancel{\text{otherwise}}, & \text{if } R_n = 0 \ \& \ Q_n \geq 1 \end{cases}$$

because if $R_n = 0$, there is a service completion during this period and it is immediately followed by a new customer entering service.

$$Q(n+1) = Q(n) + A_n - \mathbb{1}(R_n = 0 \ \& \ Q(n) = 1)$$

$$R(n+1) = (R(n) - 1)\,\mathbb{1}(R(n) \geq 1) + \lfloor \xi_{n+1}/h \rfloor\, \mathbb{1}(R(n) = 0 \ \& \ Q(n) \geq 1)$$

The above system of equations can be simulated iteratively. Because $\{(A_n, \xi_n)\} \sim$ iid, then it follows that $\{Q(n), R(n)\}$ is Markovian.

Exercises!

Let $X_n^h = \{(Q(n), R(n))\}$ for given $h > 0$.

- Show that $\{X_n^h\}$ is a Markov chain in two dimensions. How many classes? Recurrence?
- Show that as $h \to 0$, ~~the~~ for each $t \in \mathbb{R}$ the random sequence $\{Q(\lfloor t/h \rfloor)\}$ converges in distribution to the original queue process.

Ask students about initializing.

```
% service first
 If R(n) = 0   % completion of service
    if Q(n) > 0
     {
      Q(n+1) = Q(n) - 1
      Generate ξ ~ G
      R(n+1) = ⌊ξ/h⌋
     }
 % arrivals:
     Generate A ~ Ber(λh)
     Q(n+1) = Q(n) + 1
     If Q(n) = 1   % first customer
      {
        Generate ξ ~ G
        R(n+1) = ⌊ξ/h⌋
  % clock-tick advance clock:
    R(n+1) = R(n-1) - 1
```

<u>Important Questions</u>: why are limits valid? How can we estimate approximation errors? How small should h be? All of these questions are studied in the field of Simulation.

A PARENTHESIS FOR MORE PROBABILITY

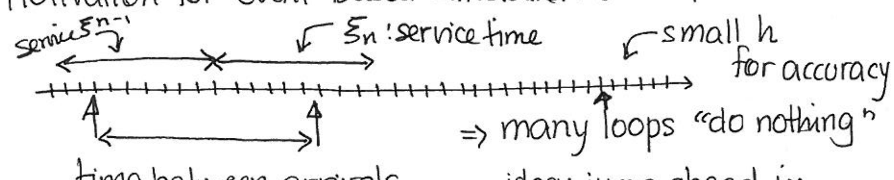
<u>Def</u>: Let $\{X_n; n \in \mathbb{N}\}$ be a sequence of random variables on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and X a random variable. We say that:

- X_n <u>converges almost surely</u> to X if
$$\mathbb{P}(\omega : \lim_{n\to\infty} X_n(\omega) = X(\omega)) = 1. \quad X_n \to X \text{ a.s.}$$
- X_n <u>converges weakly</u> or <u>in distribution</u> to X if
$$\lim_{n\to\infty} \mathbb{P}(X_n \le x) = \mathbb{P}(X \le x) \text{ for every } x \in \mathbb{R}$$
that is a point of continuity of $F(x) = \mathbb{P}(X \le x)$.

<u>Result</u>: Convergence in distribution: $X_n \Rightarrow X$ is equivalent to the condition that for every continuous and bounded function $g : \mathbb{R} \to \mathbb{R}$,
$$\lim_{n\to\infty} \mathbb{E}[g(X_n)] = \mathbb{E}[g(X)].$$

There are a number of important theorems that can be used to establish convergence: Dominated Convergence Theorem, Monotone Convergence Theorem, etc.

Motivation for event-based simulation (notas p. 44-45)



$\Rightarrow$ many loops "do nothing"

<u>idea</u>: jump ahead in one go

(b). Discrete Event Model

Physical state $\mathcal{S}$ (usually but not always countable)

Possible event set (finite number of distinct events) E, $|E| = d$

For each $x \in \mathcal{S}$, $T(x) \subset E$ is the set of possible events @ x.

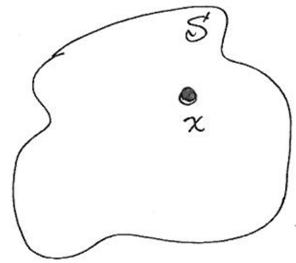
Main dynamics: by "jumps"

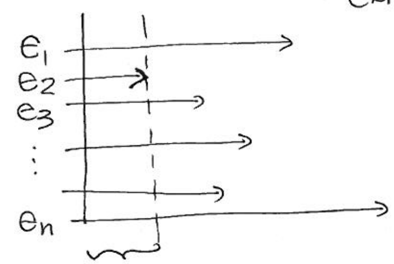
$$\mathbb{P}(X_{\text{"new"}} = j \mid X_{\text{"old"}} = x, \text{event} = e) = p_s(j; x, e)$$

Clock dynamics: also known and Markovian:

$$\mathbb{P}(\text{time for new event} \leq t \mid X_{\text{"new"}} = x) = F_e(t, x)$$

given distribution (known)



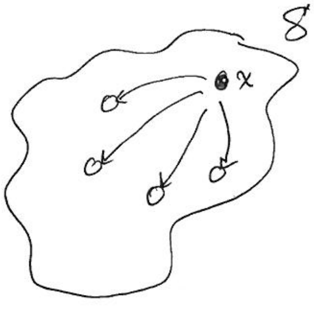


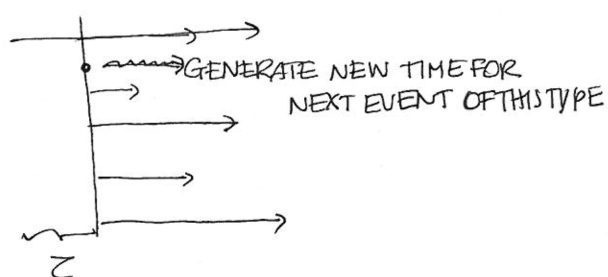
next event is e_2

τ = elapsed time

In τ units of time the state will jump from x to another state, according to the probability

$$\mathbb{P}(X_{\text{"new"}} \in \cdot \mid x, e_2)$$





advance clock and update

Def: A discrete event process Z_t on $(\Omega, \mathcal{F}, \mathbb{P})$ has a physical component X_t and a clock component Y_t.

Let:

$e(x,y) = \text{argmin}(y_i, i \in T(x))$ next event

$\tau(x,y) = \min(y_i, i \in T(x))$ time to next event

$z = (x,y) \subset \mathbb{R}^n \times \mathbb{R}^d$

$$\text{Prob}(X_{t+\tau} \in A \mid Z_t = (x,y)) = p_s(A; x, e(x,y))$$

for any $A \in \mathcal{B}(\mathcal{S})$

If $i \neq e(x,y)$ then

$$Y_{i,t+\tau} = Y_{i,t} - \tau \quad \text{\% update clocks}$$

$$\mathbb{P}(Y_{e(z),t+\tau} \cancel{\leq t} \mid Z_t = (x,y) = z) = F_{e(z)}(t, x)$$

where $p_s(\cdot\,; x, e)$ are well defined probability distributions for any $x \in \mathcal{S}$ and $e \in E$, and for each event $e \in E$, and $x \in X$, $F_e(\cdot, x)$ is a well defined distribution.

Remark: also called "stochastic timed automata"

<u>Result</u>: It is left as an exercise for students to show that the embedded discrete-time process

$$Z_n = \{Z_{\tau_n}\}, \text{ where } \tau_i = \text{time of } i\text{th event},$$

is a Markov Chain on $S \times \mathbb{R}^d$.

Also called "Generalized Semi-Markov Process"

(Y_1, Y_2) list of residual arrival (1) and departure (2) times

(initialize y_1 = Generate new arrival)

$\{T_i\} \sim$ iid dist. F (not necessarily a Poisson arrival)

$\{\xi_i\} \sim$ iid dist G

main loop:
if $Q=0 \Rightarrow e=1$ else
$e = \arg\min(Y_1, Y_2) \in \{1,2\}$;
$\tau = \min(Y_1, Y_2)$

case($e=1$) % arrival

$Q = Q+1$

$T \sim F$, $Y_1 = T$ % new inter-arrival

if $Q=1 \Rightarrow Y_2 = \xi \sim G$

case($e=2$) % service

Q --

If $Q>0 \Rightarrow Y_2 = \xi \sim G$ % new service

what is the simulation model?

Variables: $Q_n, Y_{1,n}, Y_{2,n} = Z_n$

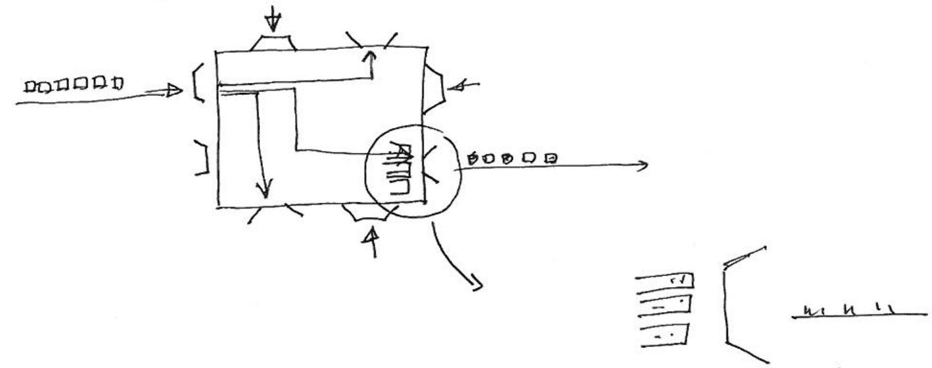
Notice that $\{Z_n\}$ is a Markov chain and consider the natural filtration $\sigma(Z_1, \ldots Z_n) = \mathcal{F}_n$.

Ex: Consider the physical process $\{Q_t;\ t \geq 0\}$ and let $\{\tau_1, \tau_2, \ldots\}$ be the consecutive event or jump times.
Let $\tilde{\mathcal{F}}_t = \sigma(Q_s;\ s \leq t)$. Explain the difference between $\tilde{\mathcal{F}}_{\tau_n}$ and $\mathcal{F}_n$.

Simulation package SSJ highly recommended

(6) ~~Standard Clock~~

Example in telecom router



Exit queues are classified according to priority customers (voice, video, real-time, etc). C classes

How many queues in network?

Each "switch" has CN ~~exits~~ queues (N is the number of exit ports). If there are M identical switches in the network then there are CNM number of queues, MN #servers

How many events in E?

@ each ~~queue~~ server : 1 residual service
C residual arrival times

Needs $MN(C+1)$ clocks in list

List search can be slow, even if heap or other methods are used to accelerate.

Assume that residual times are exponentially distributed & independent.

<u>Result</u>: Let $X \perp\!\!\!\perp Y$ be exponential rv's, on a common space $(\Omega, \mathcal{F}, \mathbb{P})$ with intensities λ_1, λ_2 resp. Then

$$e = \min(X, Y) \overset{d}{=} \exp(\lambda_1, \lambda_2)$$

<u>Proof</u>: $\mathbb{P}(e \le t) = 1 - \mathbb{P}(e > t) = 1 - \mathbb{P}(X > t, Y > t)$

$$= 1 - \mathbb{P}(X > t)\mathbb{P}(Y > t) =$$

$$= 1 - e^{-\lambda_1 t} e^{-\lambda_2 t} = 1 - e^{(\lambda_1 + \lambda_2)t}$$

$\Rightarrow e \overset{d}{=} \exp(\lambda_1 + \lambda_2)$.

<u>Proposition</u>: Let $(Y_1, \dots Y_d)$ d independent rv's with exponential distribution of intensities $(\lambda_1, \dots \lambda_d)$ respectively. Then

$$\tau = \min(Y_1, \dots, Y_d) \overset{d}{=} \exp(\Lambda)$$

$$\Lambda = \sum_{i=1}^{d} \lambda_i .$$

<u>Proposition</u>: Let $Y \overset{d}{=} \exp(\Lambda)$ be the next event time, $\tau = \min(Y_1, \dots Y_d)$. ~~Then that~~ and $e = \operatorname{argmin}(Y_1, \dots Y_d)$ then $\mathbb{P}(e = i \mid Y) = \dfrac{\lambda_i}{\Lambda}$.

The proof is left as an exercise (start with $d=2$).

Given $E, (\lambda_e, e \in E)$

<u>Def</u>: a standard clock model is a stochastic process $\{X_n, \tau_n\}$ such that:

$$\Lambda_n = \sum_{e \in T(X_n)} \lambda_e$$

$\tau_n \overset{d}{=} \exp(\Lambda_n)$ ⤳ time to next event

$\mathbb{P}(e_{n+1} = e) = \dfrac{\lambda_e}{\Lambda_n}$, $\forall e \in T(X_n)$ next event type

$$\mathbb{P}(X_{n+1} \in A \mid X_n) = p_s(A; x, e)$$

There is no search through a list $\Rightarrow$ speed up.

<u>Example</u>: Let $O(m)$ the set of exit ports at each switch m, then we generate τ using

$$\sum_{m=1}^{M} \Big(\Lambda = \underbrace{\sum_{c=1}^{C} \lambda_{c,m}}_{\text{arrivals}} + \sum_{i \in O(m)} \mu_i \, \mathbb{1}_{(Q_i > 0)} \Big)$$

state dependency $X_n = Q_n$.

<u>Remark</u>: generalization to other than exponential

4) <u>Reduced Models</u>

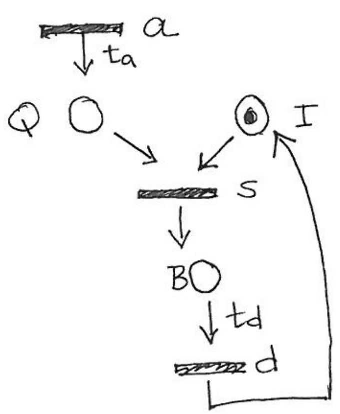
- Process-oriented simulation schemes (Cassandras) and Petri-net representation,
- state aggregation, retrospective simulation models (paper on intelligent subways and bus fleet as examples)

Process-oriented models can also be very useful. Instead of describing the dynamics in terms of the time evolution of the state of the system, these models represent the various operating processes affecting the system, in a similar way that a Gauntt or PERT chart orders the various stages in project management ("tasks").

Petri nets are used to represent the relationships of precedence, recurrence and interactions as follows.

- Nodes indicate completed stages
- Arcs represent precedence and causal relations.

Example of the FCFS queue



transitions

states, called "places"

flow relations, called "arcs"

- the "~~marks~~ tokens" that flow within the Petri net.

a: arrival of clients, always active "timed" ta are the $\{T_i\}$ iid inter-arrival times

s: service initiatio

Q: queue size

d: departure, timed
td : $\{\xi_n\} \sim G$

I: indicator of idle server

B: busy or not (server)

(a) (s) (d) are called "transitions". (See wikipedia)
Initial marking: one token @ I.
To activate it is necessary that there be tokens in the nodes that point to the transition.

Go through logic: "activate" arrivals.

Call $\{A_n\}$ the consecutive firing times or "epochs" for transition (a) [note that they correspond to customer arrival times). Call $\{D_n\}$ the consecutive firing times for transition (d): these are customer departure times. Finally, call $\{W_n\}$ the waiting times in the Q place (the queue) for the n-th token.

If $D_n \leq A_{n+1}$ the net is at same as initial marking so that $W_{n+1} = 0$ (as soon as (a) fires, the tokens enable transition s). Otherwise, the time that the n+1-st token has to wait to ~~fire~~ enable transition s is exactly $W_{n+1} = A_{n+1} - D_n$. That is:

$$W_{n+1} = \max(0, D_n - A_n).$$

On the other hand, following the customer's process, clearly $D_n = A_n + W_n + S_n$, where $\{S_n\}$ are the consecutive service times required to fire transition s. This yields:

$$W_{n+1} = \max(0, (A_{n+1} - A_n) - (W_n + S_n))$$

this expression is also known as Lindley equation.

If $X(n) = W_n + S_n$ denotes total time that customer n spends in the system, then $\{X(n)\}$ is a Markov chain on the space $\mathbb{R}^+$.

```
X[1] ~ G    % service distribution
for n=1 to N do
    T = Generate InterArrival (λ);
    S = Generate Service ~ G;
    If (X[n] < T)    X[n+1] = S;
    else             X[n+1] = S+(X[n] -T);
```

If we are interested in evaluating statistics about the waiting times then this simulation model is much simpler than ~~DES~~ tick- or event- based simulation models.

Example : airport car park.

$\boxed{\text{End of Class}}$

why do we simulate?
- Research question
- Performance functions

Experimental Design
- Methodology and proposed scenarios

Output Analysis

Efficiency of a Simulation