SE290 Modeling and Simulation

Reference Sources:

1] Nelson, Barry. *Foundations and methods of stochastic simulation: a first course*. Vol. 187. Springer Science & Business Media, 2013;

2] Banks, Jerry; Carson, John S.; Nelson, Barry L.; Nicol, David M., *Discrete-event system simulation*, 5th ed. ed. Upper Saddle River, NJ : Prentice Hall, 2010; and

3] Asmussen, Søren, and Peter W. Glynn. *Stochastic simulation: Algorithms and analysis*. Vol. 57. Springer Science & Business Media, 2007.

4] Peter E. Kloeden, Eckhard Platen.*Numerical solution of stochastic differential equations*, Stochastic Modelling and Applied Probability, Springer, 1995.

5] Kloeden, Peter Eris, Eckhard Platen, and Henri Schurz. *Numerical solution of SDE through computer experiments.* Springer Science & Business Media, 2012.

Instructor: Soumyendu Raha

Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore 560012
email: raha@serc.iisc.in

Aug-Dec Semester, 2015

Introduction::Single Server Queue

Consider a single-server queue possessing an infinite capacity waiting room and processing customers according to a first-in–first-out (FIFO) queue discipline. Let A_n, D_n, and W_n be the arrival time, departure time, and waiting time (exclusive of service) for the nth customer to enter the queue. The FIFO discipline then clearly implies that

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

Also, it is evident that $D_n = A_n + W_n + V_n$, where V_n is the service time of customer n, and hence

$$W_{n+1} = \max(0, W_n + V_n - T_n)$$

(the Lindley recursion), where $T_n := A_{n+1} - A_n$ is the time between the arrivals of customers n and $n+1$, $n = 0, 1, 2, \cdots$. Suppose that $\{V_n\}, n \geq 0$ and $\{T_n\}, n \geq 0$ are independent sequences of independent and identically distributed (i.i.d.) random variables (r.v.s). Then the single-server queue model is called the $GI/G/1$ queue. Even when $X_n := V_{n-1} - T_{n-1}$ has an explicit known standard distribution, W_n cannot be computed in closed form. The following $2n$-dimensional integration is computationally challenging:

$$\mathsf{P}(W_n > x) = \int_{B_n(x)} \prod_{k=0}^{n-1} \mathsf{P}(V_k \in dv_k)\mathsf{P}(T_k \in dt_k), \text{ where}$$

$$B_n(x) := \{(v_0, t_0), \ldots, (v_{n-1}, t_{n-1}) : \max_{k=0,\ldots,n-1} \sum_{j=k}^{n} (v_j - t_j) > x\}.$$

Single Server Queue (continued)

The distribution of W_n is an example of a transient characteristic, as opposed to steady-state or stationary characteristics, which are defined by taking the limit as $n \to \infty$. For example, an r.v. W_∞ having the limit distribution of W_n as $n \to \infty$ (provided such a limit exists as a probability measure on $\mathbb{R}$) is said to be the steady-state waiting time. Note that $\{W_n\}_{n\in\mathbb{N}}$ is a Markov chain with state space $[0, \infty)$. The theory of Markov chains with a discrete (i.e., finite or countable) state space suggests that under conditions corresponding to positive recurrence and aperiodicity, W_∞ will exist, and that the Markov chain $\{W_n\}_{n\in\mathbb{N}}$ itself will obey the law of large numbers (LLN) when the load is strictly less than the service offered.

$$\frac{1}{N}\sum_{n=0}^{N-1} f(W_n) \to \mathsf{E}f(W_\infty) \text{ almost surely, as } N \to \infty.$$

Further transient characteristics of interest are first passage time quantities such as the time $\inf\{n : W_n > x\}$ until a customer experiences a long delay x, the number $\sigma := \inf\{n > 0 : W_n = 0\}$ of customers served in a busy period (recall $W_0 = 0$), and the total length $V_0 + \ldots + V_{\sigma-1}$ of the busy period. For both transient and steady-state characteristics, it is also of obvious interest to consider other stochastic processes associated with the system, such as the number $Q(t)$ of customers in system at time t (including the one being presently served), and the workload $V(t)$ (time to empty the system provided no new arrivals occur).

$GI/G/1$ **queue and Random Walk:** One of the key properties of the single-server queue is its close connection to random walk theory, which as a nice specific feature allows representations of steady-state distributions as well as transient characteristics in terms of an associated random walk. To make this connection precise, write $X_k = V_{k-1} - T_{k-1}$ and $S_n := X_1 + \ldots + X_n$ (with $S_0 = 0$), and note that if customer 0 enters an empty queue at time 0, then

$$\begin{aligned} W_1 &= \max(X_1, 0) = \max(S_1 - S_0, S_1 - S_1), \\ W_2 &= \max(W_1 + X_2, 0) \\ &= \max(\max(S_1 - S_0, S_1 - S_1) + S_2 - S_1, S_2 - S_2) \\ &= \max(S_2 - S_0, S_2 - S_1, S_2 - S_2) = S_2 - \min(S_0, S_1, S_2) \end{aligned}$$

and in general,

$$W_n = S_n - \min_{k=0,\ldots,n} S_k = \max_{k=0,\ldots,n} [S_n - S_k].$$

Under our basic assumption that $\{V_n\}, n \geq 0$ and $\{T_n\}, n \geq 0$ are independent sequences of iid rv's, $\{S_n\}, n \geq 0$ is a random walk.

Single Server Queue (Continued)

We make the time reversal argument

$$\begin{aligned}
&(S_n - S_n, S_n - S_{n-1}, S_n - S_{n-2}, \ldots, S_n - S_1, S_n - S_0) \\
&= (0, X_n, X_n + X_{n-1}, \ldots, X_n + \ldots + X2, X_n + \ldots + X_1) \\
&\sim (0, X_1, X_1 + X_2, \ldots, X_1 + \ldots + X_{n-1}, X_1 + \ldots + X_n) \\
&= (S_0, S_1, S_2, \ldots, S_{n-1}, S_n),
\end{aligned}$$

where $\sim$ denotes equality in distribution, to obtain

$$W_n = \max_{k=0,\ldots,n} [S_n - S_k] \sim \max_{k=0,\ldots,n} S_k := M_n.$$

As a consequence, $W_n \sim\!\!\rightarrow W_\infty$ as $n \to \infty$, where $W_\infty \sim M := \max_{k \geq 0} S_k$. It follows that if $\rho = \frac{\mathsf{E}V_0}{\mathsf{E}T_0} < 1$ (i.e., the mean arrival rate $1/\mathsf{E}T_0$ is smaller than the service rate $1/\mathsf{E}V_0$), then W_∞ is a proper r.v. (so that the steady state is well defined), and $\mathsf{P}(W_\infty > x) = \mathsf{P}(M > x) = \mathsf{P}(\tau(x) < \infty)$, where $\tau(x) := \min\{n > 0 : S_n > x\}$. Then

$$\mathsf{P}(W_\infty > x) = \int_0^\infty \mathsf{P}(W_\infty \in dy)\mathsf{P}(X_1 > x - y) = \int_0^\infty \mathsf{P}(W_\infty \leq x - y)\mathsf{P}(X_1 \in dy)$$

which is the Wiener-Hopf type analogue of the stationarity equation for a Markov chain and provides a numerical way to solve for the distribution of W_∞.

Single Server Queue (Continued)

The analytically most tractable special case of the $GI/G/1$ queue is the $M/M/1$ queue, where both the interarrival time and the service time distribution are exponential, say with rates (inverse means) λ and μ. Then $\rho = \lambda/\mu$, and the distribution of W_∞ is explicitly available,

$$\mathsf{P}(W_\infty \leq x) = 1 - \rho + \rho(1 - \exp(-\gamma x)), \quad \gamma := \mu - \lambda.$$

The probabilistic meaning of this formula is that the probability $\mathsf{P}(W_\infty = 0)$ that a customer gets served immediately equals $1 - \rho$, whereas a customer experiences delay w.p. ρ, and conditionally upon this the delay has an exponential distribution with rate parameter γ.
This is also the distribution of the steady-state workload $V(\infty)$, and the steady-state queue length $Q(\infty)$ is geometric with success parameter $1 - \rho$, i.e.,
$\mathsf{P}(Q(\infty) = n) = (1 - \rho)\rho^n, n \in \mathbb{N}$. There are also explicit formulas for a number of transient characteristics such as the busy period density and the transition probabilities $p_{ij}(t) = \mathsf{P}(Q(s + t) = j | Q(s) = i)$ of the Markov process $Q(t), t \geq 0$, but the expressions are complicated and involve Bessel functions.
Beyond the $M/M/1$ queue, the easiest special case of the $GI/G/1$ queue is $GI/M/1$ (exponential services), where steady-state quantities have an explicit distribution given that one has solved the transcendental equation $\mathsf{E}\exp(\gamma(V - T)) = 1$.
Also, $M/G/1$ (Poisson arrivals) simplifies considerably.

Consider computing $\mathrm{P}(W_n > x)$ by appealing to a sampling-based method. Suppose we have algorithms that generate two independent sequences of i.i.d. r.v.s $\{V_n\}, n \geq 0$ and $\{T_n\}, n \geq 0$ with the appropriate service-time and inter-arrival-time distributions. Then, by recursively computing the W_k according to the Lindley recursion, we obtain the r.v. W_n. By repeatedly drawing additional V_k and T_k, one could then obtain R i.i.d. copies $W_{1n}, \ldots, W_{Rn}$ of W_n. The probability $z := \mathrm{P}(W_n > x)$ could then be computed via the sample proportion of the W_{rn} that are greater than x, namely via the estimator

$$\hat{z} := \hat{z}_R := \frac{1}{R} \sum_{r=1}^{R} \mathbf{1}\{W_{rn} > x\},$$

where $\mathbf{1}$ is the indicator function. The LLN guarantees that the algorithm converges to $z = \mathrm{P}(W_n > x)$ as the number R of independent replications tends to ∞. This leads to the idea of simulating independent realizations of the stochastic phenomenon under consideration and then of computing an estimate for the probability or expectation of interest via an appropriate estimator obtained from independent samples.

Monte Carlo Method (Continued)

To compute $z = \mathrm{E}Z$, we can develop an algorithm that will generate i.i.d. copies $Z_1, \ldots, Z_R$ of the r.v. Z and then to estimate z via the sample-mean estimator $\hat{z} := \hat{z}_R := \frac{1}{R}\sum_{r=1}^{R} Z_r$. One runs R independent computer experiments replicating the r.v. Z, and then computes z from the sample.

Use of random sampling or a method for computing a probability or expectation is often called the **Monte Carlo method**. When the estimator $\hat{z}$ of $z = \mathrm{E}Z$ is an average of i.i.d. copies of Z we call z a **crude Monte Carlo (CMC)** estimator.

Note that an LLN also holds for many dependent and asymptotically stationary sequences. As a consequence, one can compute characteristics like $\int f d\pi$ of the limiting stationary distribution π by averaging over a long simulation-time horizon. For example, the integral equation of Wiener-Hopf type can be solved numerically in this way.

Issue 1: How do we generate the needed input random variables?
Our discussion for the $GI/G/1$ queue assumes the ability to generate independent sequences V_n and T_n of i.i.d. r.v.s with appropriately defined distributions. The simplest examples are the $M/M/1$ queue, where the V_k, T_k are exponential, but more complicated models can occur.
All such algorithms work by transforming a sequence of uniform i.i.d. r.v.s into the appropriate randomness needed for a given application. Thus, we shall need to discuss the generation of i.i.d. uniform r.v.s and to show how such uniform randomness can be transformed into the required nonuniform randomness needed by the simulation experiment.
The additional difficulty is that we cannot simulate the entire path $S(t), 0 \leq t \leq T$. At best, we can simulate only at a finite number of discrete time points. Furthermore, because $S(t)$ typically evolves continuously through dynamics specified via an Stochastic differential Equation or SDE, the same difficulties can arise here as in the solution of ordinary (deterministic) differential equations (ODEs). In particular, the forward simulation via the process with a finite-difference approximation induces a systematic bias into the simulated process. This is comparable to the error associated with finite-difference schemes for ODEs.

Issue 2: How many computer experiments should we do?
Given that we intend to use the Monte Carlo method as a computational tool, some means of assessing the accuracy of the estimator is needed. Simulation output analysis comprises the body of methods intended to address this issue. One standard way of assessing estimator accuracy is to compute a (normal) confidence interval for the parameter of interest, so a great deal of our discussion relates to computing such confidence intervals. Special difficulties arise in:
(a) the computation of steady-state expectations, as occurs in analyzing the steady-state waiting time r.v. W_∞ of the $GI/G/1$ queue (the limit in distribution of W_n as $n \to \infty$);
(b) quantile estimation.
A necessary condition for approaching item (a) at all is of course that we be able to deal with the issue of computing expectations associated with limiting stationary distributions.

Issue 3: How do we compute expectations associated with limiting stationary distributions?

For the $GI/G/1$ queue, the problem is of course that the relation $W_n \rightsquigarrow W_\infty$ and the possibility of generating an r.v. distributed as W_n does not allow us to generate an r.v. distributed as W_∞. In other words, we are facing an *infinite-horizon problem*, in which an ordinary Monte Carlo experiment cannot provide an answer in finite time. This is also clearly seen from the formular $\mathsf{P}(W_\infty > x) = \mathsf{P}(M > x) = \mathsf{P}(\tau(x) < \infty)$, where $\tau(x) = \min\{n > 0 : S_n > x\}$ and $M = \max_{n=0,1,2,\ldots} S_n$. Indeed, simulating a finite number n of steps of the random walk S_n cannot determine the value of M but only of M_n, and furthermore, one can get an answer only to whether the event $\{\tau(x) \leq n\}$ occurs but not whether $\{\tau(x) < \infty\}$ does.

Of course, one possible approach is to use a single long run of the W_n and appeal to the LLN. Here, the complication is that error assessment will be challenging because the observations are serially correlated.

However, a currently extremely active area of steady-state simulation with a rather different flavor from that of queuing models is **Markov chain Monte Carlo methods (MCMC)**; some key application areas are statistics, image analysis, and statistical physics.

Issue 4: Can we exploit problem structure to speed up the computation?
Unlike the sampling environment within which statisticians work, the model that is generating the sample is completely known to the simulator. This presents the simulator with an opportunity to improve the computational efficiency (i.e., the convergence speed as a function of the computer time expended) by exploiting the structure of the model. In the context of the $GI/G/1$ queue, $\mathsf{E}X_1$ is known, so that for any choice of $\lambda \in \mathbb{R}$, $W_n - \lambda(S_n - n\mathsf{E}X_1)$ is again an estimator having mean $\mathsf{E}W_n$. This is an example of variance reduction (and is the one specifically known as the method of control variates). Use of good variance-reduction methods can significantly enhance the efficiency of a simulation.

Issue 5: How does one efficiently compute probabilities of rare events?
Suppose that we wish to compute the probability that a typical customer in steady state waits more than x prior to receiving service, i.e., $\mathsf{P}(W_\infty > x)$. If x is large, $z := \mathsf{P}(W_\infty > x)$ (or, equivalently, $z = \mathsf{P}(\tau(x) < \infty)$) is the probability of a rare event. For example, if $z = 10^{-6}$, the fraction of customers experiencing waiting times exceeding x is one in a million, so that the simulation of 10 million customers provides the simulator with (on average) only 10 samples of the rare event. This suggests that accurate computation of rare-event probabilities presents significant challenges to the simulator. The topic is probably more relevant for queuing theory and its communications applications.

Issue 6: How do we estimate the sensitivity of a stochastic model to changes in a parameter?

The need to estimate such sensitivities arises in mathematical finance in hedging. More precisely, in a hedging portfolio formed by a bank account and the asset underlying the option one tries to hedge, it holds rather generally that the amount to invest in the asset at time t should equal what is called the the delta, the partial derivative of the expected payout at maturity w.r.t. the current asset price.

In the $GI/G/1$ queue, suppose we are uncertain about the load the system will face. In such circumstances, it may be of interest to compute the sensitivity of system performance to changes in the arrival rate. More precisely, suppose that we consider a parameterized system in which the arrival epochs are given by the sequence $(\lambda^{-1}A_n)$, so that $\frac{\lambda}{\mathsf{E}A_1}$ is the arrival rate. The sensitivity of the system performance relating to long delays is then $\frac{d}{d\lambda}\mathsf{P}(W_\infty > x)$.

Issue 7: How do we use simulation to optimize our choice of decision parameters?

Suppose that in the $GI/G/1$ queue with a high arrival rate (say exceeding λ_0), we intend to route a proportion p of the arriving customers to an alternative facility that has higher cost but is less congested, in order to ensure that customers are served promptly. There is a trade-off between increasing customer satisfaction (increasing p) and decreasing costs (decreasing p). The optimal trade-off can be determined as the solution to an optimization problem in which the objective function is computed via simulation.

Event-Based Simulation

Lindleys equation makes queueing simulation look deceptively easy. However, if there are ten servers instead of one, or if customers are served in some priority order, or if we are interested in the number of customers in the queue rather than the waiting time, then an event-based simulation is typically required.

- The Simulation Clock jumps from event time to event time; called the *next event time advance*.
- The current state of the system, the event logic and the list of future events are needed to advance the system to the next state change.
- Simulation ends when a particular system state occurs or at a fixed time or after a fixed number of event counts.
- The system state over time is a *sample path* output from which performance measures are extracted. This is one realization of a stochastic process.

Event-Based Simulation

- Involves
  - Event scheduling
  - Activity scanning
  - Process interaction
  - List Processing
- Static: System consists of Entities which have Attributes and has a System State
- Temporal Part:
  - Event is System State change
  - Event notice schedules Event
  - FEL Future Event List
  - Activity is a known period of time
  - Delay is an unknown period of time

List is ordered set; **Clock** is the simulated time counter
Model is set of relations between everything
Activity is time interval between causally related events: Duration known at start time Activity starts with an event E1: E1 may be Arrival; E1 may be Service completion; and ends with primary event E2 caused by E1: E2 may be Next Arrival; E2 may be Next service completion.
Delay is time interval between an event E1 and another event E2 which are not directly related by cause and effect; Duration unknown at start time. Delay starts with an event E1: E1 may be Arrival of customer C. Delay ends with event E2: E2 may be Customer leaves. Many events in between if C has to queue up.

Event-Based Simulation

Discrete Event Scheduling is based on one simple idea of Causality: Events cause other events.

Mathematical Viewpoint of Event Scheduling Algorithm: State: $x(t) \in \mathbb{R}^n$; Event Set: $\mathcal{E} = \{E_1, \ldots, E_k\}$. Global Clock: t. Model: $\mathcal{R} = \{R_1, \ldots, R_k\}$, 1 rule per event. A rule R_i creates a state change caused by event E_i at time t and a set of m future event notices $\{E_{k1}(t_1), \ldots, E_{km}(t_m)\}$. Causality demands $t_j > t,\ j = 1, \ldots, m$. The m time intervals are m activities created by event E_i.

Event scheduling/time-advance algorithm: Assume we have an FEL (future event list) consisting of time ordered event notices: FEL $= \{E_1(t_1), ..., E_L(t_L)\}$, $t_k < t_n$ for $k < n$, E_k is a data structure (Object) with at least a time member variable (usually more).

The algorithm's core loop is:

Remove first event E from FEL

  Advance clock to t=E.t

  Apply model rule R to E to create:

    State change

    Set of events $\{E_1(t_1), .., E_m(t_m)\}$ caused by E

Insert $\{E_1(t_1), .., E_m(t_m)\}$ into FEL

Collect data (whatever youre interested in)

Go back to the top of the loop.

Event scheduling/time-advance algorithm

Bootstrapping:
Define initial state at $t = 0$
Clear all counters and measurement vars
Place first event on FEL
Define termination condition
Place special stop-event $E_s(T)$ on FEL to stop at predetermined time T.
Define stop condition to check for at every event.

Example: Server queue with impatient customers
State = (Q,S)
Q = [c1,c2,c3,...] (queue of customers); S = 0 or 1, idle or busy
Customer c new entity, with unique id
Event set = A,F,L,S
A = arrival of customer; F = server finishes; L = Customer leaves queue; S = stop simulation event
4 Rules for the 4 events, RA, RF, RL and RS (t0 is prev. event time)
RA : state response to A
Create customer entity c
If S(t0)=0 {Q unchanged,S=1} else {Q.Enqueue(C)),S=1}

RA : Event notices caused by A(t), Create event notice A(t + getArrivalTime())
If S(t0)=0, Create F(t+getServiceTime())
If S(t0)=1, Create Set L(t+getFedupTime(),c)
RF : state response to F
If Q.len(t0)=0, S=0 else {Q.dequeue, S(t)=1}; (Q.dequeue removes first in line)
RF : Event notices caused by F(t)
If Q.len(t0) greater than 0, Create F(t+getServiceTime())
RL : state response to L
If L.c is in Q, Q.remove(c), Else do nothing (defunct event)
RS: Stop simulation and process results
RF : state response to F: If Q.len(t0)=0, S=0 else {c=Q.dequeue, S(t)=1}
Check FEL for L event for customer c, if so remove from FEL RF : Event notices caused by F(t): If Q.len(t0) is greater than 0, Create F(t+getServiceTime())

Note that L is a more complex event with a customer property besides a time property. The last part optimizes event removal.

Uniform Random Variables

The basic vehicle in the area of (stochastic) simulation is a stream $u_1, u_2, \ldots$ of numbers produced by a computer, which is treated as the outcome of a sequence $U_1, U_2, \ldots$ of i.i.d. r.v.s with a common uniform distribution on $(0, 1)$. Many software libraries contain routines for generating such streams, such as a command of the type u1:=random; u2:=random; ... in the language C++ or u=rand(1,n) in Matlab (creating a row vector containing u1, . . . , un).

The algorithms in practical implementations are all deterministic (typically using recursion) and can therefore at best mimic properties of i.i.d. uniform r.v.s; for this reason, the sequence of outputs is called a sequence of **pseudorandom numbers**. Earlier generations of computers and software had quite a few examples of random number generators with unfortunate properties. However, it is our view that the situation is much improved now and that the development of algorithms for generation of pseudorandom numbers (as well as sequences of quasirandom numbers for similar uses) is now largely a specialists topic: the typical user will do well with existing software (which is fast, certainly much faster than home-made high-level language routines) and will seldom be able to improve it substantially.

Physical Devices

In the early era of computer simulation, a widely used idea was to exploit properties of radioactive decay: the times at which particles are emitted from a radioactive source is a classic example of a Poisson process, as has been verified empirically and which also has strong theoretical support. In particular, since individual atoms split in an unpredictable way, independent of one another, and the number of atoms is astronomical, the Poisson approximation to the binomial distribution is highly accurate. Thus, the interevent times can be viewed as i.i.d. exponentials. They can be transformed to other distributions such as the uniform.

Such physically based generators were slow, generating at best only hundreds of random numbers per second. The lack of replicability can also limit ones ability to apply certain variance-reduction techniques. Of course, the rapid decrease in the cost of high-speed computer memory offers the opportunity to prerecord huge quantities of such physically generated random numbers. A more fundamental objection to the use of such physical devices to generate random numbers is that most of them produce numbers with a systematic (and difficult to quantify) bias.

While deterministic mathematical algorithms also generally exhibit biases, such biases can often be studied mathematically, so that their possible effect is better understood. Still are situations in which physically generated truly random numbers are preferable (at a minimum as seeds) to pseudorandom numbers, e.g., gambling and cryptography.

Deterministic Recursive Algorithms

Instead of using physical devices, pseudorandom numbers $u_1, u_2, \ldots$ are typically produced by deterministic recursive algorithms. A general framework covering virtually all such generators occurring in practice is a quadruple (E, μ, f, g). Here E is the (finite) set of states, the state of the random number generator evolves according to the recursion $s_n = f(s_n - 1)$; and the random number stream produced is $u_n = g(s_n)$, where g maps S into $[0, 1]$. The initialization is determined by μ, which is a probability measure selecting x_1 (the seed). Since E is finite, the range of g is not all of $[0, 1]$ but only the finite subset $g(E)$. In practice, the generator is modified so that the values 0 and 1 cannot occur (say 0 is replaced by ϵ, the smallest nonzero value of g or the smallest positive number representable on the computer). This is to avoid problems in using the sequence; say one needs division or to take logarithms. Finiteness also implies that there is a d such that $x_l + d = x_d$ for some l; the period is the smallest d for which this happens. After l, the algorithm will produce replicates of cycles of length d or smaller. One difficulty with generators having short periods is that the gaps in the sequence may not be evenly distributed.
To reduce the problems associated with the finiteness of $g(E)$ and the period, a necessary (but not sufficient!) condition is obviously that E be large.

Examples of Deterministic Recursive Algorithms

Linear congruential generators (were popular for many years but are now somewhat outdated.): Such algorithms have the form

$$u_n = s_n/M \text{ where } s_{n+1} = (As_n + C) \mod M.$$

The difficulty is in choosing a large M and associated A, C such that the period is large, preferably M when $C = 0$ or $M - 1$ when $C = 0$ (this property is called full period). Number-theoretic considerations provide verifiable conditions under which linear congruential generators are of full period. This has led to certain popular parameter choices for A, C, and M. A dominant one in earlier generations of computers and software was $M = 2311 = 2\ 147\ 483\ 647, A = 75 = 16\ 807, C = 0$. This choice has the nice property that its period is (very) close to the number of machine-representable integers in a 32-bit computer. Another example is $M = 2\ 147\ 483\ 563, A = 40\ 014$.

Generalization of Linear congruential generators.

$$x_n = (A_1 x_{n-1} + \ldots + A_k x_{n-k}) \mod M. \quad s_n = (x_n, x_{n-1}, ..., x_{n-k+1}).$$

When M is (the largest computer representable) prime number, the maximal period M^{k-1}. Long integers are used in the programming implementations.

Examples of Deterministic Recursive Algorithms

Longer Periodicity. Certain numerically demanding simulations require enormous numbers of random numbers. The congruential generators mentioned above have periods only on the order of billions. Much larger periods can be obtained by mixing generators.

1. $x_n := (A_1 x_{n-2} - A_2 x_{n-3}) \mod M_1$,
2. $y_n := (B_1 y_{n-1} - B_2 y_{n-3}) \mod M_2$,
3. $z_n := (x_n - y_n) \mod M_1$,
4. If $z_n > 0$, return $u_n = z_n/(M_1 + 1)$; else return $u_n = M_1/(M_1 + 1)$,

where $M_1 = 4\ 294\ 967\ 087$, $M_2 = 4\ 294\ 944\ 443$, $A_1 = 1\ 403\ 580$, $A_2 = 810\ 728$, $B_1 = 527\ 612$, $B_2 = 1\ 370\ 589$ (the seed is the first three x's and the first three y's).

Modulo 2 Generators. Instead of performing algebra modulo M for some huge integer, one can exploit algebra modulo 2. A general scheme is

$$x_n = Ax_{n-1},\ y_n = Bx_n,\ u_n = y_{n,1}2^{-1} + \ldots + y_{n,l}2^{-l}$$

where $x_n \in \{0,1\}^k$, $y_n \in \{0,1\}^l$, A, B are $k \times k$ and $l \times k$, and all operations are performed modulo 2. Output sequence follows $u_n = \alpha_1 u_{n-1} + \ldots + \alpha_k u_{n-k} \mod 2$ vis-a-vis the characteristic polynomial $\det(zI - A)$. **Tausworthe generators** are special cases: $z_n = a_1 z_{n-1} + \ldots + a_k z_{n-k} \mod 2$, $u_n = z_{nt+1}2^{-1} + \ldots + z_{nt+r}2^{-k}$. For $z_n = z_{n-102} + z_{n-249}$ (A mostly 0), the period is $2^{500} - 1$; and the fast **Mersenne twister** with equidistributional properties has a period of $2^{19937} - 1$. Combinations possible.

Statistical Tests

Statistical tests concentrate on confirming whether the marginal empirical distribution of the U_n is uniform (up to rounding errors) and that observations look independent within a narrow time range. However, it should be noted that (being deterministic) no stream of pseudorandom numbers is truly random, which is always revealed by a suitable statistical test. Tests for deviations from a realization of a sequence of truly random numbers are basically just standard statistical goodness-of-fit tests.

Q-Q plot plots the empirical quantiles of the set $u_1, \ldots, u_n$ relative to the uniform quantiles $q_\alpha = \alpha$ and is useful w.r.t. the first uniformity of the one-dimensional marginals.

Standard Significance Tests. The χ^2-test splits $I = (0, 1)$ into subintervals $I_1, \ldots, I_K$, typically just $(0, 1/K), [1/K, 2/K), \ldots$. For subinterval k, observed number O_k is defined as the number of $u_1, \ldots, u_n$ taking values in I_k and $E_k = n|I_k|$ (Lebesgue measure) as the expected number. The χ^2-statistic is $\sum_{k=1}^{K} (O_k - E_k)^2 / O_k$ and is approximately χ^2-distributed with $f = K - 1$ degrees of freedom provided the E_k are not too small. The Kolmogorov-Smirnov statistic is $\max_x |\hat{F}_n(x) - F(x)|$, where $\hat{F}_n$ is the empirical c.d.f. of $u_1, \ldots, u_n$ and $F(x)$ $(= x$ here) the theoretical c.d.f.; the asymptotic distribution of the statistic is complicated, but the quantiles have been tabulated and are available from statistical software packages.

Statistical Tests

Testing independence. One way is through a goodness-of-fit approach, testing that d-blocks $(u_{k+1}, \ldots, u_{k+d})$ are uniformly distributed in $(0,1)^d$. The χ^2-test applies to this, but a difficulty is that the E_k quickly become too small for the asymptotic results as d grows, and one should note also that the independence assumptions are violated, since neighboring d-blocks are dependent (a way out is to base the test on blocks $k = 0, d, 2d, \ldots$ only).

Another idea is to use a normal transformation $x_k = \Phi^{-1}(u_k)$ where the empirical correlations at lags $k = 1, 2, \ldots$ are possible test statistics.

Ad hoc methods are: the gap test based on the fact that the spacings between the (random) indices k for which $u_k \in J \subset (0,1)$ should behave like i.i.d. geometric r.v.s with success parameter $|J|$; the coupon collectors test, where one records the successive lengths of minimal sequences containing all K values of $\lfloor K u_n \rfloor$ for a suitable integer K and compares to the theoretical distribution; and so on.

Care must be taken w.r.t. the interpretation of multiple tests. For example, if 20 tests are each performed at a 5% rejection level in a traditional statistical setting, the expected number of rejections under the null hypothesis is one. It is therefore a delicate matter to assess when one should make an overall rejection in a multiple-test setting, and dependence further complicates the matter. The situation in testing random number generators is somewhat different, since one would usually not fix a significance level like 5% or 1%, but continue the test until the test comes out with a clear conclusion.

Randomness Criteria

This concept is not at all well defined.
A desirable property is obviously that the empirical distribution of d-blocks (the distribution on $(0,1)^d$ giving mass $1/n$ to each of the n d-blocks in $u_1, ..., u_{n+d-1}$) should converge to the uniform distribution on $(0,1)^d$ as $n \to \infty$, for any $d = 1, 2, \ldots$; this is sometimes referred to as the sequence being ∞-distributed (also the terminology normal number is used). It is also reasonable to require unpredictability. One formulation of this is to require all subsequences of $u_1, u_2, \ldots$ to be ∞-distributed, and one then talks of a vonMises-Church collective (A proper selection rule for sub-sequences which is defined as any recursive function which having read the first N elements of the sequence decides if it wants to select element number $N+1$).
Another possible approach is via specific measures of deviations, in particular discrepancies; often discussed in connection with quasirandom numbers.

Nonuniform Random Variables

Accepting the U_n as i.i.d. uniform, the next step is to use them to produce an r.v. X with a prescribed distribution F, say Poisson, exponential, normal, etc. As for uniform random numbers, the user will typically use standard routines for this purpose. However, occasionally one may come across a nonstandard distribution that is not avaliable in this way, e.g., conditional distributions occuring in Gibbs sampling; and generation of r.v.s from truncated and normalized Lévy measures. In general, it seems reasonable to have some insight into how nonuniform random numbers are produced.
Standard routines are usually designed with considerable attention to speed. In cases in which the user for one reason or another writes his/her own routine for generating random numbers, optimal speed is, however, rarely a concern, and the algorithms that we describe below are not necessarily the fastest ones; the average user may want to use a naive but easily programmed method rather than invoking more sophisticated methods or various library packages. A rule of thumb is that it is much faster to generate a bunch of uniform random numbers, say 5–10, than to evaluate even a single special function such as the logarithm, the exponential, or a trigonometric function.
We use $U, U_1, U_2, \ldots$ without further explanation for independent r.v.s from the uniform$(0, 1)$ distribution.

A simple case is a Bernoulli r.v., $\mathrm{P}(X = 1) = 1 - \mathrm{P}(X = 0) = p$, where one can just let $X := 1(U \leq p)$. This construction generalizes in a straightforward way to distributions with finite support. In particular, if we want to generate X as being uniform on $\{1, \ldots, N\}$, we may take $X = 1$ if $U \leq 1/N,\ X = 2$ if $1/N < U \leq 2/N$, and so on. This is often written as $X = \lceil U/N \rceil$.
If the target distribution is given by $\mathrm{P}(X = x_i) = p_i$, where $\sum_{i=1}^{N} p_i = 1,\ p_i \geq 0$, we let $X = x_1$ if $U \leq p_1$, $X = x_2$ if $p_1 < U \leq p_1 + p_2$ and so on. This is a case of an inversion algorithm. The efficiency depends on how rapidly the search for the subinterval straddling U can be accomplished. The search can be avoided when $]\mathrm{P}(X = i) = k_i/n$ for integers $k_i \geq 0,\ n \geq 1$, at the cost of memory. One stores the integer i in the first k_i memory locations. Upon generating U, one returns the integer in memory location $\lceil nU \rceil$.

For a continuous r.v. with a density $f(x)$, there are two general methods: inversion and acceptancerejection (A–R), and a variety of more *ad hoc* methods that use special properties of the target distribution.

Inversion

For inversion, one defines F^+ as the inverse of the c.d.f. F of X. The simplest case is that in which F is strictly increasing and continuous; then $x = F^+(u) = F^{-1}(u)$ is the unique solution of $F(x) = u,\ 0 < u < 1$. For distributions with non-connected support or jumps, more care is needed, and one then needs the more general definition of the inverse, where we here have chosen the left-continuous version

$$F^+(u) := \inf\{x : F(x) \geq u\} = \min\{x : F(x) \geq u\},\ 0 < u < 1;$$

that the minimum is attained follows since $\{x : F(x) \geq u\}$ is an interval (infinite to the right) that must contain its left endpoint by right-continuity of F.

Proposition. (a) $u \leq F(x) \iff F^+(u) \leq x$; (b) if U is uniform$(0,1)$, then $F^+(U)$ has c.d.f. F; (c) if F is continuous, then $F(X)$ is uniform$(0,1)$.

Proof. Part (a) follows directly from the definition of F^+ ($F(x) \geq u$ for $x \geq F^+(u)$ but not for any $x < F^+(u)$), and (b) follows trivially from (a), which yields $\mathrm{P}(F^+(U)) \leq x) = \mathrm{P}(U \leq F(x)) = F(x)$. Finally, for (c) we have to prove $\mathrm{P}(F(X) \geq u) = 1 - u,\ 0 < u < 1$. However, by (a) and continuity of F, $\mathrm{P}(F(X) \geq u) = \mathrm{P}(X \geq F^+(u)) = \mathrm{P}(X > F^+(u)) = 1 - F(F^+(u))$, so we must prove $F(F^+(u)) = u$. $F(F^+(u)) \geq u$ is clear from (a) with $x = F^+(u)$. Let $y_n < F^+(u)$, $y_n \uparrow F^+(u)$. Then $F(y_n) < u$ by (a), and so by continuity of F, $F(F^+(u)) \leq u$. □

Part (b) allows us to generate X as $F^+(U)$ (the most common case is an F that is continuous and strictly increasing on an interval).

Exponential RVs

Let F be exponential with rate (inverse of mean) δ; the density is $f(x) = \delta \exp(-\delta x)$. Then $F^+(x) = \log(1-x)/\delta$, so inversion means that $X = -\log(1-U)/\delta$ (in practice, one often uses $X = -\log U/\delta$). Packages often avoid the expensive log computation. From exponential r.v.'s, one can build $\Gamma(p, \delta)$ r.v.'s with integer shape parameter p (the density is $\delta^{p-1}/(p-1)! \exp(-\delta x)$) by simply adding p independent copies.

A Poisson r.v. N with rate λ ($\mathrm{P}(X = n) = \exp(-\lambda)\lambda^n/n!$) can be constructed using the relation between the exponential distribution and the Poisson process: generate $X_1, X_2, \ldots$ as i.i.d. exponentials with rate λ one at a time until the random time at which the sum exceeds 1, say $X_1 + \ldots + X_N < 1 < X_1 + \ldots + X_{N+1}$. Then N is the desired Poisson r.v. Using $N = \max\{n \geq 0 : \prod_{i=1}^n U_i > \exp(-\lambda)\}$ avoids evaluating special functions. There are usually more efficient methods available for standard distributions.

Inversion applies to many other examples, but (in addition to being slow) a main limitation is that quite often F^+ is not available in explicit form, for example when $F = \Phi$, the standard normal c.d.f. Sometimes approximations are used. For example, the following symmetrical in u rational polynomial approximation is standard, simple, and accurate for the normal distribution:
$\Phi^{-1}(u) = y + \frac{p_0+p_1 y+p_2 y^2+p_3 y^3+p_4 y^4}{q_0+q_1 y+q_2 y^2+q_3 y^3+q_4 y^4}$, $0.5 < u < 1$, where $y = \sqrt{-2\log(1-u)}$,
$p_0 = -0.322232431088, q_0 = 0.099348462606$; $p_1 = -1, q_1 = 0.588581570495$;
$p_2 = -0.342242088547, q_2 = 0.531103462366$;
$p_3 = -0.0204231210245, q_3 = 0.10353775285$;
$p_4 = -0.0000453642210148, q_4 = 0.0038560700634$.

Remark on Inversion. A particularly convenient feature of inversion is that whenever F^+ is available, one can simulate not only a r.v. distributed as X but also one from certain conditional distributions. In particular, if X has a density and $a < b$, then an r.v. distributed as X given $X \in (a, b)$ has c.d.f. $(F(x) - F(a))/(F(b) - F(a))$ and can therefore be generated as $F^+(F(a)(1-U) + F(b)U)$. Similarly, an r.v. distributed as the overshoot X – a given $X > a$ can be generated as $F^+(UF(a)) - a$, etc.

Simple Acceptance-Rejection. The idea is to start from an r.v. Y with a density $g(x)$ (the proposal), which is easily simulated and has the property $f(x) \leq Cg(x)$, where $f(x)$ (the target) is the density of X and $C < \infty$ is a constant. Given $Y = x$, one accepts Y and lets $X = Y$ w.p. $f(x)/Cg(x)$. Otherwise, a new Y is generated, and one continues until eventual acceptance. Algorithmically:

1. Generate Y from the density $g(x)$ and U as uniform$(0, 1)$.
2. If $U \leq f(Y)/Cg(Y)$, set $X := Y$. Otherwise return to Step 1.

This produces an r.v. with the desired density $f(x)$ since

$$\mathsf{P}(X \in dx) = \mathsf{P}((Y \in dx)|A) = \frac{\mathsf{P}((Y \in dx) \cap A)}{\mathsf{P}A}$$

$$= \frac{g(x)dx f(x)/Cg(x)}{\int_{-\infty}^{\infty} g(y)f(y)/Cg(y)dy} = \frac{f(x)dx}{\int_{-\infty}^{\infty} f(y)dy} = f(x); \; A := \{U \leq f(Y)/Cg(Y)\}$$

Acceptance-Rejection

The efficiency of A-R is determined by the speed of generation from $g()$ and by the acceptance probability

$$\mathrm{P}A = \mathrm{E}(\mathrm{P}(A|Y)) = \mathrm{E}\frac{f(Y)}{Cg(Y)} = \int \frac{f(y)}{Cg(y)} g(y)dy = \frac{1}{C}\int f(y)dy = 1/C.$$

Obviously it is preferable to have C as close to 1 as possible, which in turn means that $g()$ should look as much alike $f()$ as possible.
As a simple example, let $f(x)$ be a bounded density on $(0,1)$, say the Beta density $x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta)$ with $\alpha,\beta > 1$. Then, we may take Y as uniform on $(0,1)$ and $C = \sup_{0<x<1} f(x)$.
Let $f(x) = \sqrt{2/\pi}\exp(-x^2/2)$ be the density of the absolute value $X = |Z|$ of a standard normal r.v. As trial density, we consider the standard exponential density $g(x) = \exp(-x)$. Then $f(x)/g(x) = \sqrt{2/\pi}\exp(-x^2/2)$ is maximized for x $= 1$ and the maximum is $C = \sqrt{2e/\pi}$. The acceptance probability is $1/C \approx 0.76$. The normal r.v. Z itself is easily simulated by assigning X a random sign. It is rare that, as in this example, $f(x)/g(x)$ can easily be maximized analytically and thereby the optimal C determined. However, usually it is not worthwhile to look for the optimal C, and a rather crude bound determined numerically will do.

Acceptance-Rejection Example

Let the target density $f()$ be the $\Gamma(\alpha, \lambda)$ density with $\alpha = 4.3,\ \lambda = 1$. For $g(x)$, we can exploit the fact that $\Gamma(p, \lambda)$ r.v.'s with integer p are easily simulated (as a sum of p independent exponentials) and that the corresponding density $g(x)$ is not too far from $f(x)$ if p is close to α and λ close to 1. The precise choice of p, λ is restricted by boundary conditions: for $f(x) \leq Cg(x)$ to hold with a finite C for all $0 < x < \infty$, $g(x)$ must go more slowly to 0 than $f(x)$ as $x \to 0$ or $\to \infty$. Now the $\Gamma(\alpha, \lambda)$ density is of order $x^{\alpha-1}$ as $x \to 0$ and of order $x^{\alpha-1}\exp(-\lambda x)$ as $x \to \infty$. In chosing between $p = 4$ and $p = 5$, the $x \to 0$ requirement therefore excludes $p = 5$. With $p = 4$ we must then take $\lambda < 1$ to meet the $x \to \infty$ requirement since $x^{4.3-1}\exp(-x)$ decays slower than $x^{4-1}\exp(-\lambda x)$ when $\lambda > 1$. Fixing λ, thereafter one must find C to satisfy $f(x) \leq Cg(x)$ for all x, which can be done empirically from the plot (check graphically that the inequality holds unless x is very close to 0 or ∞, and rely on the boundary asymptotics in this range).

Log-Concave Densities. Let $f(x)$ be a log-concave decreasing density on $[0, \infty)$ with mode at $x = 0$; for r.v. generation, we can w.l.o.g. assume $f(0) = 1$, since otherwise we can just proceed via $\tilde{X} := f(0)X$, which has a density $\tilde{f}(x)$ satisfying the given assumptions. Then

$$f(x) \leq \min(1, \exp(1 - x)).$$

Assume $f(x_0) > \exp(1 - x_0)$ for some x_0 (necessarily > 1). By logconcavity, $\log f(x)$ is above the line connecting $(0, 0)$ and $(x_0, \log f(x_0))$ for $x \in (0, x_0)$, which yields

$$\int_0^\infty f(x)dx > \int_0^{x_0} \exp(x(1 - x_0)/x_0)dx = \frac{x_0}{x_0 - 1}(1 - \exp(1 - x_0)) > 1,$$

a *contradiction*. Interpreted as the density of an r.v. Y with a distribution that is a mixture with equal weights $1/2$ of a uniform$(0, 1)$ distribution and a standard exponential distribution shifted rightward by 1.

Assume still that f is log-concave, not necessarily concentrated on $(0, \infty)$, but now we want to generate an r.v. conditioned to (a, b) where $\infty < a < b < \infty$. Letting $\beta = f'(a)/f(a)$, the density of such an r.v. is bounded by $\frac{f(a)}{F(b)-F(a)} \exp(\beta(x - a)) = Ch(x)$, where density h is proportional to $\exp(\beta x)$ in (a, b) and $C > 1$. We can therefore just sample from h by inversion and obtain the desired conditioned r.v. by rejection. $b < \infty$ may be omitted if $\beta < 0$.

The BoxMuller method for Normal r.v.s. Scaling an exponential r.v. by 2, one obtains a χ^2 with 2 degrees of freedom, which is the distribution of the squared radial part $R^2 := Y_1^2 + Y_2^2$ of independent $N(0,1)$ r.v.s Y_1, Y_2. Since the conditional distribution of Y_1, Y_2 given $R = r$ is uniform on the circle $\{(v_1, v_2) : v_1^2 + v_2^2 = r^2\}$ with radius r, we obtain the following algorithm for generating (pairs of) normal r.v.s from (pairs of) uniforms: $Y_1 := \sqrt{-2\log U_1}\sin 2\pi U_2,\ Y_2 := \sqrt{-2\log U_1}\cos 2\pi U_2$. But this is slow because of special functions; work-arounds are complex.

Marsaglia's Polar Method. Replaces the evaluation of the trigonometric functions above by an A-R step. Observe that if $V_1 = Z\cos\Theta, V_2 = Z\sin\Theta$ is the polar representation of a random pair (V_1, V_2) that is uniform on the unit disk $\{(v_1, v_2) : v_1^2 + v_2^2 \leq 1\}$, then $W = Z^2$ and Θ are independent and uniform on $(0,1)$, respectively $(0, 2\pi)$. Thus $\sqrt{-2\log Z} = \sqrt{-\log W}$ and $(\cos\Theta, \sin\Theta) = (V_1/\sqrt{W}, V_2/\sqrt{W})$ are independent with an exponential distribution, respectively an uniform distribution on the unit circle, so that $Y_1 := \sqrt{(-\log W)/W}V_1,\ Y_2 := \sqrt{(-\log W)/W}V_2$. (V_1, V_2) are produced by A-R (generate (U_1, U_2) and repeat until $U_1^2 + U_2^2 \leq 1$).

Geometric Distribution exists in two variants: supports $\{0, 1, \ldots\}$ and point probabilities $p_n = (1-\rho)\rho^n$: and supports: $\{1, 2, \ldots\}$ and point probabilities $p'_n = (1-\rho)\rho^{n-1}$. For the corresponding r.v.s X, X' one has $X' := X + 1$. Generate Bernoulli(p) r.v.s with $p := 1 - \rho$ until the first 1, and X be the number of preceding 0s. I f $\rho \to 1$, this is slow. Alternatively use a geometric r.v. X with the same distribution as the integer part of an exponential r.v. Y with mean $-1/\log\rho$. Set $X := \lfloor \log U/\log\rho \rfloor$ with $U(0,1)$.

Normal Variance mixtures. Some rv's can be simulated as $X = SY$, where $Y \sim N(0,1)$, $S > 0$.

The Alias Method. Let F be an $(n+1)$-point distribution on $x_1, \ldots, x_{n+1}$ with weight p_k for x_k. Naive r.v. generation consists in letting $X = x_K$, where K satisfies $p_1 + \ldots + p_{K-1} < U < p_1 + \ldots + p_K$ with U uniform. To avoid the search for K, the *alias method* consists in representing F as a mixture of n distributions $G_1, \ldots, G_n$ with equal weights $1/n$, such that G_l is a two-point distribution on $x_{1,l}, x_{2,l} \in \{x_1, \ldots, x_{n+1}\}$ with weights $p_{1,l}, p_{2,l}$ for each. Given such a representation, one can then generate two uniforms U_1, U_2, set $M = \lceil U_1/n \rceil$ and take $X = x_{1,M}$ if $U_2 \leq p_{1,M}$, $X = x_{2,M}$ otherwise. To find G's, we start by choosing i such that $p_i < 1/n$; that such i exists follows since otherwise $\sum_i p_i > 1$. Next, we can choose j such that $p_i + p_j \geq 1/n$, since otherwise one would have $p_j < 1/n$ for $j = i$ and therefore $\sum_i p_i < 1/n + (n-1)/n \leq 1$. We then set $x_{1,1} := x_i,\ x_{2,1} := x_j,\ p_{1,1} = np_i, p_{2,1} := 1 - np_i$. For $k = i, j$ we then have $p_k \leq 1 - p_i - p_j < (n-1)/n$. Therefore, if we define H_1 as the n-point measure on $\{x_1, \ldots, x_{n+1}\} \setminus \{x_i\}$ with point probability $n/(n-1)p_k$ for $x_k \neq x_i,\ x_j$ and $n/(n-1)(p_i + p_j - 1/n)$ for x_j, then H_1 is a probability distribution satisfying $F = G_1/n + (n-1)/nH_1$. Decomposing H_1 in a similar way as $1/(n-1)G_2 + (n-2)/(n-1)H_2$, we have $F = G_1/n + G_2/n + (n-2)/nH_2$, and repeating the procedure a further $n-2$ times yields the desired decomposition. Trade-off between the setup time and the number of X_k is important. Generalizes in a straightforward way to a mixture $F = p_1F_1 + \ldots + p_nF_n$.

Further A-R

The A-R method allows some extension beyond the case in which $f(x)$ is a normalized density. More precisely, one can sometimes deal with the situation in which the target distribution F has density $f(x)/D$ w.r.t. some reference measure μ but D is unknown. One can find an upper bound C on $C := \sup_x f(x)/h(x)$ for some known $h(x)$ and use A-R for simulating r.v.s $Y_1, Y_2, \ldots$ from $h(x)$ and accepting w.p. $f(x)/Ch(x)$. The situations with a known $f(x)$ but unknown D are, however, typically complicated. In fact, they constitute the typical area of problems handled by Markov chain Monte Carlo/MCMC and the problems one encounters there when attempting to apply A-R rather than the traditional MCMC methods are first of all that it may be difficult to find a suitable h. Even if an h with $C < \infty$ and allowing for easy r.v. generation can be found, the bound C may be difficult to evaluate and will typically be far from C, making the rejection rate huge. Note that computing C has the same order of difficulty as maximum likelihood estimation!

Ratio of uniforms. the ratio V_1/V_2 of a random pair with a uniform distribution on the unit disk has the same distribution as that of the ratio of a pair of standard normals which (by Box-Muller) is in turn that of $\tan(2\pi U)$, that is, the Cauchy distribution. The ratio-of-uniforms method replaces the unit disc by a more general region $\Omega := \{(v_1, v_2) : 0 < v_1 \leq \sqrt{cf(v_2/v_1)}\}$, f being the target density; samples (V_1, V_2) from the uniform distribution on Ω and returns $X = V_2/V_1$. Change to $(x = v_2/v_1, y = v_1)$ from (v_1, v_2) and the Jacobian is $v_1^{-1} = y^{-1}$. Then $|\Omega| = \int\int_\Omega dv_1 dv_2 = \int dx \int_0^{\sqrt{cf}} y dy = \int cf/2 dx = c/2 < \infty$, and the density of (X, Y) is the density $|\Omega|^{-1}$ of (V_1, V_2) divided by the Jacobian, so that marginal density of X is $|\Omega|^{-1} \int_0^{\sqrt{cf}} y dy = cf/(2|\Omega|) = f(x)$.
In practice (V_1, V_2) is sampled as uniforms from a rectangle R containing Ω and accepting them if they fall in R. R is generally $(0, (\sup\{cf, x \in \mathbb{R}\})^{1/2}) \times (\pm(\sup\{x^2 cf, x \geq 0\})^{1/2})$ provided the sups are finite. Let $(v_1, v_2) \in \Omega$. Then $0 < v_1 \leq a$ is obvious. $h := cf$. If $v_2 > 0$, then $v_1^2 \leq h^2$; $v_2^2 \leq x^2 h$ with $x = v_2/v_1$. Then $0 \geq v_2 \geq -x^2 h$ follows.
Take $h = \exp(-x^2/2)$ and $R = (0, 1) \times (\pm(2/\exp(1))^{1/2})$. Acceptance condition is $V_1 \leq \sqrt{h(V_2/V_1)}$, that is, $V_2^2 \leq -4V_1^2 \log V_1$, that is, $X^2 \leq -4 \log V_1$.

von Neumann, Exponential. Let X be standard exponential and Y exponential conditioned to $(0,1)$, that is, with density $\exp(-y)/(1-\exp(-1))$ for $0<y<1$. Then X can be generated as $M+Y$, where M is independent of Y and geometric with $\mathsf{P}(N=n)=(1-\exp(-1))\exp(-n)$. To generate Y, consider sequential sampling of i.i.d. uniforms until $N:=\inf\{n>1: U_1>U_2>\ldots>U_{n-1}<U_n\}$. Since $U_1>U_2>\ldots>U_{n-1}>U_n$ corresponds to one of $n!$ orderings, we have $\mathsf{P}(N>n)=1/n!$. Similarly, the extra requirement $U_1\le y$ imposes the conditioning $U_k\le y$ for all $k\le n$, so that $\mathsf{P}(N>n, U_1\le y)=y^n/n!$, $\mathsf{P}(N=n, U_1\le y)=\frac{y^{n-1}}{(n-1)!}-\frac{y^n}{n!}$, $n=2,3,\ldots$. Summing over $n=2,4,\ldots$, yields $\mathsf{P}(N \text{ is even}, U_1\le y)=1-\exp(-y)$. Thus repeating the experiment until an even N comes out, we can take Y as the U_1 in the final (i.e., the first accepted) trial.

Squeezing. The target density $f(x)$ may sometimes be difficult to evaluate due to nonstandard special functions. A-R implemented via the upper bound $f(x)\le Cg(x)$ is then potentially slow because of the repeated evaluations of $f(x)$. The idea of squeezing is to supplement with an easily evaluated lower bound $h(x)\le f(x)$, which makes it possible to reduce the number of evaluations of $f(x)$. Then one first tests $U\le h(Y)$ and then accepts, so that the test $U\le f(Y)/Cg(Y)$ needs to be performed only when $U>h(Y)$.

Transform Based Methods

In some cases, the density $f(x)$ or the c.d.f. $F(x)$ of a distribution $F(dx)$ may not be available in explicit form, whereas the transform $\hat{F}[s] = \int \exp(sx)F(dx)$ is. In others, $f(x)$ or $F(x)$ may be in principle computable, but is much more complicated than $\hat{F}[s]$. One may therefore ask whether one can generate r.v.s from F using $\hat{F}[s]$ only. Note that $\psi(s) := \hat{F}[\mathrm{i}s]$ viewed as function of a real variable s is the characteristic function of F.
Fourier Inversion of $\psi(x)$ is $f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \psi(x)\exp(-\mathrm{i}xs)ds$
Approximate Procedure: 1. Evaluate $f(x_i),\ i = 1, \ldots, n$ from Fourier Inversion.
$x_1 < \ldots < x_n$.
2. Discrete $\tilde{F}$ (approximate of F) as point mass $p_i := f(x_i)/(\sum_{i=1}^n f(x_i))$ at x_i.
3. The desired rv is sampled from $\tilde{F}$.
FFT. The above method can be implemented via FFT. Choose the support of $\tilde{F}$ as $\{x_1, x_2 = x_1 + \epsilon, x_3 = x_1 + 2\epsilon, \ldots, x_n + (n-1)\epsilon\}$. The DFT of the corresponding (a priori unknown) approximate weights $p_1, p_2, \ldots, p_n$ is then $\hat{p} = Fp/n$, where F is the finite Fourier matrix of order n and $p := (p_1, \ldots, p_n)^{\mathsf{T}}$. Element by element,
$\hat{p}_r = \frac{1}{n}\sum_{s=1}^n p_s w^{rs} = \frac{1}{n}\sum_{s=1}^n p_s \exp\left(\frac{2\pi \mathrm{i}r}{n}(1 + (x_s - x_1)/\epsilon)\right) =$
$\frac{1}{n}\exp\left(\frac{2\pi \mathrm{i}r}{n}(1 - x_1/\epsilon)\right)\sum_{s=1}^n \exp(\frac{2\pi \mathrm{i}r}{n}x_s)$, where $w := \exp(2\pi \mathrm{i}/n)$ and we solved $x_s = x_1 + (s-1)\epsilon$ for s. Approximate $\hat{p}_r$ by $\hat{q}_r := \frac{1}{n}\exp(\frac{2\pi \mathrm{i}r}{n}(1 - x_1/\epsilon)\psi(2\pi r/n\epsilon)$. A good choice of p is the inverse transform $|\bar{F}\hat{q}|/(|\bar{F}\hat{q}|\mathbf{1})$, element wise absolute value.
Sample and return $\tilde{X}$ or $\tilde{X} + \epsilon(U - 1/2)$.

For a more sophisticated method, assume that F has a second moment, i.e., that ψ'' is integrable. Integration by parts and Fourier Inversion yield

$$f(x) = \frac{1}{2\pi \mathrm{i} x} \int_{-\infty}^{\infty} \psi'(s) \exp(-\mathrm{i}sx) ds = -\frac{1}{2\pi \mathrm{i} x^2} \int_{-\infty}^{\infty} \psi''(s) \exp(-\mathrm{i}sx) ds$$

making the tail of $f(x)$ at most $O(x^{-2})$. This suggests we use a proposal $g(x)$ proportional to $\min(c, kx^{-2})$ for suitable c, k.
The algorithm computes

$$c := \frac{1}{2\pi} \int_{-\infty}^{\infty} |\psi(s)| ds; \; k := \frac{1}{2\pi} \int_{-\infty}^{\infty} |\psi''(s)| ds.$$

Then set $g(x) := \min(c, kx^{-2})/(4\sqrt{kc})$. Simulation of an r.v. Y from $g()$ is easy, since $g()$ is a mixture of a uniform distribution on $\pm\sqrt{k/c}$ and a shifted Pareto equipped with a random sign, with weight $1/2$ for each. The algorithm then accepts Y w.p. $g(Y)/(4\sqrt{kc})f(Y)$, where $f(Y)$ has to be calculated by Fourier Inversion.

Multivariate Normals A multivariate normal r.v. $X = X_{\mu,\ \Sigma} \in \mathbb{R}^p$ is given by its mean vector μ and its covariance matrix Σ. Since $X_{\mu,\ \Sigma}$ can be generated just as $\mu + X_{0,\Sigma}$, we assume $\mu = 0$ in the following and write $X := X_\Sigma$.
$\Sigma = CC^\mathsf{T}$ has a square-root. X_Σ can be generated as CX_I.
Use $\Sigma^{1/2} := B\Lambda^{1/2}B^\mathsf{T}$, B being unitary matrix; Λ diagonal matrix.
Cholesky Factorization. C is lower or upper triangular. Express $X = CY$ (with C as a lower triangular matrix) a combination of $N(0,1)$ rv's. Observe that
$\Sigma_{pi} = \mathrm{Cov}(X_p, X_i) = \sum_{k=1}^{i} c_{pk}c_{ik}$ which gives p linear equations that are solved as $c_{pj} = (\Sigma_{pj} - \sum_{k=1}^{j-1} c_{pk}c_{jk})/c_{jj}, j < p, c_{pp}^2 = \Sigma_{pp} - \sum_{k=1}^{p-1} c_{pk}^2$.
Symmetric Positive Correlations. This corresponds to $\Sigma_{ii} = \sigma^2,\ \Sigma_{ij} = \rho\sigma^2$ with $\rho \in (0,1]$. Take $X_k = \sigma\rho^{1/2}Z + \sigma(1-\rho)^{1/2}Y_k,\ k = 1,\ldots,p$, where $Z, Y_1, \ldots, Y_p$ are i.i.d. standard normals.
Symmetric Negative Correlations. $\rho < 0$ but $\rho \geq -1/(p-1)$ for Σ being positive semi-definite. $X_k = bY_k - a\sum_{l\in\{1,\ldots,p\}\setminus\{k\}} Y_l,\ k = 1,\ldots,p$ where $Y_\cdot$ are iid standard normal rv's and $(p-1)a^2 + b^2 = 1,\ (p-2)a^2 - 2ab = \rho$. the maximal attainable negative correlation is obtained by setting $X_k := (1-1/n)^{-1/2}\epsilon_k$ where $\bar{Y} := (Y_1 + \ldots + Y_p)/p$ is the sample mean and $\epsilon_k := Y_k - \bar{Y}$ is the residual.

Often, it may be reasonable to assume that $\mathrm{Cov}(X_k, X_l)$ depends only on $l-k$.

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{p-2} & \rho_{p-1} \\ & 1 & \rho_2 & \dots & \rho_{p-3} & \rho_{p-2} \\ & & \ddots & & & \vdots \\ & & & & 1 & \rho_1 \\ & & & & & 1 \end{pmatrix}$$

where $\rho_l = \mathrm{Corr}(X_k, X_{k+l})$. A multivariate normal r.v. with such a covariance matrix is in particular obtained by sampling p consecutive values in a stationary Gaussian process, for which there are many specific candidates around.

Moving Averages. In above, many cases will have a decreasing tendency of the ρ_l. To obtain a particular desired pattern, one may consider an $MA(q)$ (moving average of order q) process where $\rho_l = 0$ for $l > q$, leading to $X_k = \sum_{j=1}^{q} a_j Y_{k+j-1}$, where $Y_\cdot$ are iid standard normals and lead to the equations $1 = \sum_{i=1}^{q} a_i^2$;
$\rho_1 = a_1a_2 + a_2a_3 + \ldots + a_{q-1}a_q$; $\rho_2 = a_1a_3 + a_2a_4 + \ldots + a_{q-2}a_q$; $\ldots$; $\rho_q = a_1a_q$.

Other Parametric Multivariate Distributions

Multivariate t. This is defined as the distribution of $\left(\frac{Y_1}{\sqrt{W}}, \ldots, \frac{Y_p}{\sqrt{W}}\right)$ where $Y_\cdot$s are iid standard normals and W is an independent χ^2_f/f rv (f=degrees of freedom). Generalization is to take $Y \sim N(0, \Sigma)$.

Multivariate Laplace. This is defined as the distribution of $\left(\sqrt{W}Y_1, \ldots, \sqrt{W}Y_p\right)$ where $Y_\cdot$ are i.i.d. standard normals, and W an independent standard exponential r.v..

Dirichlet Distribution. The distribution has parameters $a_1, \ldots, a_p$ and is defined as distribution of $(Y_1/s, \ldots, Y_p/S)$ where $Y_\cdot$ are independent and we have $\Gamma(a_1, 1), \ldots, \Gamma(a_p, 1)$ and $S := Y_1 + \ldots + Y_p$. The kth marginal is Beta(a_k, a), where $a := a_1 + \ldots + a_p$.

Multinomial Distribution. This is the distribution of the counts $(X_1, \ldots, X_k)$ of $N = X_1 + \ldots + X_k$ objects classified into k categories, with probability say p_j for the jth. A naive method is to go through the objects one by one and classify an object by sampling j w.p. p_j . A more efficient method (assuming that fast generation of binomial (M, p) variables is available also for large M) is to generate first X_1 as binomial (N, p_1), next X_2 as binomial $(N - X_1, p_2/(1 - p_1)$, and so on. The algorithm uses sampling from the marginal distribution of X_1 and the conditional distributions of X_i given $X_1, \ldots, X_{i-1}$. This can in principle be used for any multivariate distribution, but the problem is that the conditional distributions seldom have an attractive form.

Marshall-Olkin bivariate exponential distribution is that of (X_1, X_2) where $X_1 := T_1 \wedge T_{12}, X_2 := T_2 \wedge T_{12}$ with $T_.$ independent exponentials with rates $\lambda_.$. X_i fails at time T_i specific to it and a time T_{12} common for both items, such that the actual failure occurs at the first of these potential failure times. The marginals are exponential with rates $\lambda_1 + \lambda_{12}$, respectively $\lambda_2 + \lambda_{12}$.
A completely different type of **multivariate distributions is that of uniform distributions on a region** Ω. The simplest algorithm is to find (if possible) a box $(a_1, b_1) \times \ldots \times (a_p, b_p)$ containing Ω generate consecutive uniforms on the box in the obvious way from p uniform$(0, 1)$ r.v.s, and accept the first in Ω. A case not covered by this is $|\Omega| = 0$ (Lebesgue measure). The particular case in which Ω is the unit sphere $\{x : \|x\|_2^2 = 1\}$ in $\mathbb{R}^p$ is particularly important and easy: X can be generated as $(Y_1/\sqrt{R}, \ldots, Y_p/\sqrt{R})$, where $Y_1, \ldots, Y_p$ are i.i.d. standard normal and $R := \|Y\|_2^2$.

Whereas there is basically one and only one standard way to extend the normal distribution to multidimensions, that is not the case for other standard distributions. In a few cases (such as the exponential) there is a variety of suggestions around, but usually no clear picture of which one is the natural one. However, in most cases it is hard to come up with even just one. Copulas then provide a possible approach, not least in examples in which one has a rather well-defined idea of the marginals but a rather vague one of the dependence structure.

Copulas

A copula is defined as the distribution of a p dimensional random vector U where each U_i has a marginal distribution that is uniform$(0,1)$ but the U_i may be dependent. For a general p-dimensional vector Y with a continuous marginal distribution, say F_i, of Y_i, the copula of Y is defined as the distribution of $(F_1(Y_1) \ \ldots \ F_p(Y_p))^{\mathsf{T}}$. Some simple basic examples are co-monotonicity (the completely positively dependent copula), where $U_1 = \ldots = U_p$; the completely negatively dependent copula for $p = 2$ where $U_1 = 1 - U_2$; Gaussian copulas corresponding to Y being multivariate normal, and multivariate t copulas. In simulation, copulas are useful for generating multivariate dependent r.v.s outside of standard situations such as the multivariate Gaussian. In more general terms, one can describe copulas as a general tool to model dependence of whatever kind and to separate the dependence structure from the marginal distributions. Generate $U \in (0,1)^p$ with the desired copula, and next transform the components of U with the respective inverse c.d.f.s.
If $p = 2$ for simplicity and write $C(u_1, u_2) := P(U_1 \leq u_1, U_2 \leq u_2)$. If C is absolutely continuous on $(0,1)^2$, the corresponding density is $c(u_1, u_2) := \frac{\partial^2}{\partial u_1 \partial u_2} C(u_1, u_2)$. A further relevant concept is the tail copula, which in terms of (U_1, U_2) is defined as the copula of $(1 - U_1, 1 - U_2)$.

Farlie-Gumbel-Morgenstern Copula. $C(u_1, u_2) = u_1 u_2(1 + \epsilon(1 - u_1)(1 - u_2))$ with $\epsilon \in [-1, 1]$. Generate rvs as follows. Draw V_1, V_2 as uniform$(0, 1)$ and set $U_1 := V_1;\ U_2 := 2V_2(a + b)$, where $a := 1 + \epsilon(1 - 2V_1),\ b := (a^2 - 4(1 - 1)V_2)^{1/2}$.

Archimedean Copula Such a copula is specified in terms of a generator, $\phi : [0, 1] \to [0, \infty]$ that is C^2 on $(0, 1)$ with $\phi(1) = 0$ and $\phi'(u) < 0,\ \phi''(u) > 0$ for all $u \in (0, 1)$. The bivariate c.d.f. of the copula is $\mathsf{P}(U_1 \le u_1, U_2 \le u_2) = \phi^{-1}(\phi(u_1) + \phi(u_2))$ if $(\phi(u_1) + \phi(u_2)) \le \phi(0)$, and 0 otherwise.

Clayton Copula. $\phi(u) = u^{-\alpha} - 1$ for some $\alpha > 0$.

Frank Copula. $\phi(u) = -\log \frac{\exp(\alpha u) - 1}{\exp(\alpha) - 1}$ for some $\alpha \in \mathbb{R}$.

Gumbel-Hougaard Copula. $\phi(u) = [-\log u]^{\alpha},\ \alpha \ge 1$.

The generation of an r.v. from an Archimedean copula can be performed by drawing V_1, V_2 as uniform$(0, 1)$ and letting $U_1 := \phi^{-1}(U_1 \phi(w)),\ U_2 := \phi^{-1}((1 - U_1)\phi(w))$ where $w := K^{-1}(U_2),\ K(t) := t - \phi(t)/\phi'(t)$.

Frailty Copulas. Create dependence via an unobservable r.v. Z. Taking U_1, U_2 to be conditionally independent given Z with $\mathsf{P}(U_1 \le u_1 | Z = z) = u_1^z,\ \mathsf{P}(U_2 \le u_2 | Z = z) = u_2^z$, the bivariate c.d.f. of the copula becomes $\mathsf{P}(U_1 \le u_1; U_2 \le u_2) = \mathsf{E}(u_1 u_2)^Z = \hat{Z}(\log u_1 + \log u_2)$, where $\hat{Z} = \mathsf{E} \exp(zZ)$ is the m.g.f of Z. The Clayton copula is an example, and a frailty copula is Archimedean. One can design specific copulas.

A common way to illustrate the dependence structure of a bivariate copula is via a scatter-plot. The singular component of the MarshallOlkin copula of course corresponds to the case $X_1 = X_2$, which arises if $T_{12} < T_1 \wedge T_2$.

Tail Dependence

Qualifying or quantifying dependence. Take bivariate case of a random vector $(X_1, X_2) \in \mathbb{R}^2$ with joint c.d.f. $F(x_1, x_2) := \mathsf{P}(X_1 \leq x_1, X_2 \leq x_2)$ with marginals $F_1(x_1) := F(x_1, \infty) = \mathsf{P}(X_1 \leq x_1)$ and similarly for F_2. For maximal/minimal positive dependence, upper and lower *Fréchet-Hoeffding bounds* are useful: Look for F's s.t. $[F_1(x_1) + F_2(x_2) - 1]^+ \leq F(x_1, x_2) \leq F_1(x_1) \wedge F_2(x_2)$ are attained. Define a set $S \subseteq \mathbb{R}^2$ to be increasing if $(x_1, x_2), (y_1, y_2) \in S$ and $x_1 < y_1$ implies $x_2 \leq y_2$, and decreasing if $(x_1, x_2), (y_1, y_2) \in S$ and $x_1 < y_1$ implies $x_2 \geq y_2$.

Proposition. *A random vector $(X_1, X_2) \in \mathbb{R}^2$ attains the upper Fréchet-Hoeffding bound iff its support is an increasing set. Similarly, the lower Fréchet-Hoeffding bound is attained iff the support is a decreasing set.*

E.g., An increasing (decreasing) set is the graph of a nondecreasing (nonincreasing) function $f : \mathbb{R} \to \mathbb{R}$. Taking $f(x) := \pm x$, we obtain:

Corollary. *In the case of identical marginals, $F_1 = F_2$, the upper Fréchet-Hoeffding bound is attained for the co-monotonic copula $X_1 = X_2$ a.s., and the lower Fréchet-Hoeffding bound is attained for the counter-monotonic copula $X_1 = F^+(U), X_2 = F^+(1-U)$ with U uniform$(0, 1)$.*

Positive quadrant dependence is $F(x_1, x_2) \geq F_1(x_1)F_2(x_2)\ \forall\ x_1, x_2$, that is $\mathsf{P}(X_1 > x_1, X_2 > x_2) \geq \mathsf{P}(X_1 > x_1)\mathsf{P}(X_2 > x_2)$. The related (upper) tail dependence coefficient $\lambda := \lim_{t \uparrow 1} \mathsf{P}(F_1(X_1) > t | F_2(X_2) > t) = 2 - \lim_{t \uparrow 1} \frac{1 - C(t,t)}{1-t}$. $\lambda = 0$ given by independence cupola; co-monotonic gives $\lambda = 1$. If (X_1, X_2) have $N(0, 1)$ marginals and correlation ρ, then $\lambda = 0$ but $\mathsf{E}(X_1 | X_2 = x) = \rho x \to \infty$ as $x \to \infty$.

Markov chains and recursions. The simplest process with dependence is a (time-homogeneous) Markov chain $\{X_n\}_{n\in\mathbb{N}}$ with a finite or countable state space E, and one can generate a sample path by simulation based on the transition probabilities $p_{ij} = \mathsf{P}(X_{n+1} = j | X_n = i)$. Generate X_{n+1} from $X_n = i$ by generating an E-valued r.v. from the distribution $p_i := (p_{ij})_{j\in E}$ with point probabilities p_{ij} by one of the parametric or nonparametric approaches.

Most often the time evolution of $\{X_n\}$ has a form other than transition probabilities, e.g., a recursion not restricted to a discrete E. The Lindley Recursion for waiting times of the $GI/G/1$ is an example. Another example is an auto-regressive process $X_{n+1} = aX_n + \epsilon_n$ with ϵ_n iid or ARMA. ARMA(p, q) is not itself Markov but is representable as a function of higher-dimension Markov Chain. A Generalized Autoregressive Conditionally Heteroscedastic (GARCH) process is given as $X_n = \sigma_n Z_n,\ \sigma_n^2 = \alpha_0 + \alpha_1 X_{n-1}^2 + \beta\sigma_{n-1}^2$ with $Z_.$ iid. If $Z_. \sim N(0, 1)$ it models mean-zero X_n with variance σ_n^2 that fluctuates in a non-iid way (e.g., stochastic volatility models). The Markov Chain is $\{\sigma_n^2, X_n\}$ with state space $(0, \infty) \times \mathbb{R}$.

For autoregressive, ARMA, and GARCH processes and many others, the expression for the transition kernel is complicated; simulation via the recursion becomes the approach.

Any Markov chain satisfying weak regularity conditions can be represented via a recursion $X_{n+1} = \phi(X_n, \epsilon_n)$ for some deterministic ϕ and innovation $\{\epsilon_n\}$. ϵ_n can be even taken as i.i.d. uniform$(0, 1)$. E.g., if the state space is $\mathbb{R}$, then $F_x(y) := \mathsf{P}_x(X_1 \le y)$ and $\phi(x, u) := F^+x(u)$.

Inhomogeneous Poisson Processes

The epochs σ_n of a standard Poisson process $\{N(t)\}$ with rate β are easily generated since the interarrival times $T_n = \sigma_n - \sigma_{n-1}$ can be generated as i.i.d. exponential(β). In many situations, a given point process $\{N^*(t)\}$ is Poisson, but with a time varying rate $\beta(t) \leq \beta < \infty$. $\{N^*(t)\}$ can then be constructed by thinning $\{N(t)\}$ with retention probability $\beta(t)/\beta$ at time t. An epoch σ_n of $\{N(t)\}$ is accepted as an epoch σ_m^* of $\{N^*(t)\}$ w.p. $\beta(\sigma_n)/\beta$.

1. Set $n := 0;\ n^* := 0; \sigma := 0;\ \sigma^* := 0$.
2. Generate T as exponential(β). Set $\sigma := \sigma + T;\ n := n + 1$.
3. Generate $U \sim \text{uniform}(0, 1)$; if $U \leq \beta(t)/\beta$, set $n^* := n + 1;\ \sigma^* := \sigma$.
4. Return to step 2.

Correct $\beta(t)$ for $\{N(t)\}$:

$$\begin{aligned} \sigma(t)dt &:= \mathsf{P}(\sigma_m^* \in [t, t + dt] \text{ for some } m = 0, 1, \ldots) \\ &= \mathsf{P}(\sigma_n \in [t, t + dt] \text{ for some } n = 0, 1, \ldots)\beta(t)/\beta \\ &= \beta dt \beta(t)/\beta = \beta(t)dt. \end{aligned}$$

For nonbounded $\beta()$, choose different dominating β's on different intervals.
When $B := \int_0^t \beta(s)ds,\ B^{-1}$ is avaliable, $T_1 = B^{-1}(X_1)$ is the first interarrival time; X_1 is standard exponential; $T_2 = B^{-1}(X_1 + X_2)$ and
$\mathsf{P}(T_1 > t) = \mathsf{P}(X_1 > B(t)) = \exp(-B(t))$.

Let $\{J(t)\}_{t\geq 0}$ be a Markov process with a finite state space E and intensity matrix $\Lambda = (\lambda_{ij})_{i,j\in E}$. One can simulate $\{J(t)\}$ at transition epochs by noting that the holding time of i is exponential with rate $\lambda_i := -\lambda_{ii}$, and that the next state j is chosen w.p. λ_{ij}/λ_i.

1. Set $t := 0,\ J := i_0$.
2. Set $i := J$; generate T as exponential with rate λ_i and K with $\mathsf{P}(K = j) = \lambda_{ij}/\lambda_i$, $j \neq i$.
3. Set $t := t + T, J := K$ and return to 2.

Uniformization of Markov processes. The uniformization algorithm creates events at a uniform rate η instead of λ_i depending on $i = J(t)$.

1. Set $t := 0, J := i_0$.
2. Set $i := J$; generate T as exponential with rate η and K with $\mathsf{P}(K = j) = \lambda_{ij}/\eta,\ j \neq i$, and $\mathsf{P}(K = i) = \lambda_i/\eta$.
3. Set $t := t + T,\ J := K$ and return to 2.

The algorithm makes the event times a homogeneous Poisson process with rate η, and the successive values of J a Markov chain with transition matrix $I + \Lambda/\eta$. The method applies also to the countable case, provided $\sup_{i\in E} \lambda_i < \infty$ although with unclear efficiency.

Markov-modulated Poisson processes.

Consider a Markov-modulated Poisson process with arrival rate β_i when $J(t) = i$ (here $\{J(t)\}$ is Markov with transition rates λ_{ij}). The intensity of an event (a transition $i \to j$ or a Poisson arrival) is $\lambda_i + \beta_i$ when $J(t) = i$. Thus, choosing $\eta \geq \max_{i \in E}(\lambda_i + \beta_i)$ and letting Δ be some point $\notin E$ (marking an arrival), we may generate the arrival epochs σ as follows:

1. Set $t := 0$, $J := i_0, \sigma := 0$.
2. Set $i := J$; generate T as exponential with rate η and K with

$$\mathsf{P}(K = j) = \begin{cases} \beta_i/\eta \text{ if } j = \Delta \\ \lambda_{ij}/\eta \text{ if } j \in E,\ j \neq i \\ (\eta - \lambda_i - \beta_i)/\eta \text{ if } j = i. \end{cases}$$

Set $t := t + T$.

3. If $K = \Delta$, set $\sigma := t$; otherwise, set $J := K$ and return to 2.

Some More Random Objects

Random Permutations. For a given integer N, generate a random permutation $(\sigma_1, \ldots, \sigma_N)$ of $(1, \ldots, N)$ such that all $N!$ permutations are equally likely. Represent a permutation by a set $(a_1, \ldots a_N)$ of integers with $a_i \in \{1, \ldots, N - i + 1)$ such that σ_1 is element a_1 in the list $L_1 = (1, \ldots, N)$ and for $i > 1$, σ_i is element a_i in list L_i obtained from L_{i-1} by deleting its element σ_{i-1}. L_N is singleton and $a_N = 1$. To make every permutation equally likely, $a_1, \ldots, a_N$ should be chosen as the outcomes of independent r.v.s $A_1, \ldots, A_N$ with A_i being uniform on $\{1, \ldots, N - i + 1\}$. The A_i can be drawn one at a time, or starting from a uniform r.v. B_0 on $\{1, \ldots, n!\}$. Then $A_1 := \lceil B_0/(n-1)! \rceil$; $B_1 := B_0 \mod (n-1)!$ are independent, with A_1 having the desired distribution and B_1 being uniform on $\{1, \ldots, (n-1)!\}$. Thus A_2 can be taken as $\lceil B_1/(n-1)! \rceil$, A_3 as $\lceil B_2/(n-1)! \rceil$, where $B_2 = B_0 \mod (n-2)!$ and so on. As N grows, overflow problems quickly appear due to $N!$.

Order Statistics. Given that inversion is feasible, if $U_{(1)} < \ldots < U_{(n)}$ are the order statistics from the uniform$(0, 1)$ distribution, $(F^+U_{(1)}), \ldots, F^+(U_{(n)}))$ have the desired distribution. A $O(n)$ algorithm exploits the connection to the Poisson process and standard properties of its relative spacings: simulate $Z_\cdot$ as standard exponentials and let $U_{(i)} := S_i/S_{n+1}$ where $S_k = \sum_{i=1}^{k} Z_i$. Also used is $\mathsf{P}(U_{(k)} \leq y_k | U_{(k+1)} = x) = y^k/x^k$ implying the order in $U_\cdot$ is generated by recursion $U_{(k)} = U_k^1 / U_{(k+1)}$ with $U^\cdot$ being uniforms. The uniform case can also be reduced to the exponential case by $z \to u = 1 - \exp(-x)$. A-R schemes are also employed. Records are defined via the random times $1 = n_1 < \ldots < n_K$ at which $X_{n_k} > \max_{i < n_k} X_i$.

Point Processes. Let $\Omega \subset \mathbb{R}^2$ have finite area $|\Omega|$. A Poisson process on Ω with intensity λ can then be simulated by embedding Ω in a rectangle R, generatiing N as Poisson(λ/p), where $p := |\Omega|/|R|$, distributing N points uniformly within R, rejecting the ones not in Ω w.p $1-p$ and retaining the rest.The rectangle may be replaced by a disk when the shape of Ω makes this more natural.

Discrete Event Systems and GSMPs

Discrete-event systems are in close correspondence with the class of generalized semi-Markov processes. A GSMP generalizes the concept of a discrete continuous-time Markov process as well as that of a semi-Markov process. Specify the set S of (physical) states and the set E of possible events that can trigger state transitions; the events active in $s \in S$ are denoted by $E(s) \subseteq E$. For the single-server queue, $S = N = \mathbb{N}$ and $E = \{0, 1\}$, where 0 corresponds to an arrival event and 1 to a departure event. $E(0) = \{0\},\ E(s) = E = \{0, 1\}$ for $s \geq 1$. When an event $e \in E$ is scheduled, a clock is set. While $s \in S$, the clock corresponding to e runs down at the deterministic rate $r_{s,e}$. When a clock hits zero, the event e corresponding to that clock occurs, $s \in S$ transitions to a new (randomly chosen) state s' w.p. $p(s'; s, e)$. When the new state s' is entered, new events typically need to be rescheduled by $N(s'; s, e)$ (the new clocks); a clock reading for $e' \in N(s'; s, e)$ is then set independently according to the distribution $F(; s', e', s, e)$. On the other hand, the remaining old clocks active in s', namely $O(s'; s, e^*) := E(s') \setminus N(s'; s, e)$, continue running down in s'.

For the $GI/G/1$ queue, $p(s+1;s,0)=1$ and $p(s;s+1,1)=1$ for $s \geq 0$; all other $p(s';s,e)$ values are zero. The distributions F depend only on e'; F corresponds to the interarrival distribution if $e'=0$ and the service time distribution if $e'=1$. For this example, $r_{s,e}=1$ for $s \in S,\ e \in E$. If $S(t)$ is the state at time t, then $\{S(t)\}, t \geq 0$ is called a generalized semi-Markov process. If all the distributions F are exponential, then $\{S(t)\}$ is a continuous-time Markov process. If $E(s)$ is a singleton for each $s \in S$, then $\{S(t)\}$ is a semi-Markov process.

Consider a processor-sharing queue in which all customers are served simultaneously at rate $1/s$ when $s \geq 1$ customers are present. We can again take $S=N$ and use the convention that when $s>1$, the customers are ordered in their order of arrival. We let $E=N$, such that $e=0$ is an arrival event and $e=i \geq 1$ the event that customer i departs. The clock speeds are $r_{s,0}=1$ for all s and $r_{s,i}=1/s$ for $s \geq 1$ and $i=1,\ldots,s$, where $E(s)=\{0,1,\ldots,s\}$. At an event of type $e=0$, the state goes from s to $s+1$, clock 0 is reset according to the interarrival time, clocks $1,\ldots,s$ remain the same, and clock $s+1$ is initialized with a service time r.v. At an event of type $e=i \geq 1$, the state goes from s to $s-1$, clocks $0,\ldots,i-1$ remain the same, and clocks $i+1,\ldots,s$ are shifted to positions $i,\ldots,s-1$.

GSMPs

Because of the multiple clocks active in each state of a GSMP and the memory the old clocks carry with them into the next state, GSMPs are typically analytically intractable. But they can be simulated. $E(s)$ can be large (in the dozens or hundreds) and needs efficient data structures for representing clock readings and for triggering events as simply as possible. A common choice is to use an ordered linked list for this purpose, but more exotic data structures (such as trees) are also used.
Analytically, by appending the state vector $C(t)$ of the clocks active in $S(t)$ to our state description, we arrive at a Markov process. Because the dynamics of this Markov process are deterministic between physical state transitions, one can alternatively use the embedded jump chain to study the GSMP.
A closely related class of stochastic models that has essentially identical modeling power to that of GSMPs is that of stochastic Petri nets. Petri net representations of discrete-event systems have the advantage of offering graphical representations of the model that are often (much) more compact than comparable graph representations derived from the GSMP description, particularly when the model needs to resolve concurrency issues.

Output analysis

Normal Confidence Intervals. Assessment of the accuracy requires first studying the rate of convergence. Assuming $\sigma^2 := \text{Var } Z < \infty$, the central limit theorem (CLT) states that $\sqrt{R}(\hat{z} - z) \rightsquigarrow N(0, \sigma^2)$ as $R \to \infty$. That is, $\hat{z} \approx\sim z + \sigma V/\sqrt{R},\ V \sim N(0,1)$ when R is large. Implications are:

1. The convergence rate is of order $R^{1/2}$. Slow convergence rate; accuracy estimates needed
2. The error, for large R, is asymptotically normally distributed. Confidence inteervals needed.
3. The error, for large R, depends on the underlying problems probability distribution through a single scalar measure, namely the standard deviation σ which characterizes the problem's difficulty.

If z_α denotes the α-quantile of the normal distribution $\Phi(z_\alpha) = \alpha$, then the CLT implies that the asymptotic probability of the event $\{z_{\alpha/2}\sigma/\sqrt{R} < \hat{z} - z < z_{1-\alpha}\sigma/\sqrt{R}\}$ is $1 - \alpha/2 - \alpha/2 = 1 - \alpha$. Let $I_\alpha := (\hat{z} - z_{1-\alpha/2}\sigma/\sqrt{R}, \hat{z} - z_{\alpha/2}\sigma/\sqrt{R})$. Then $\mathrm{P}(z \in I_\alpha) \to 1 - \alpha$. I_α is the asymptotic $1 - \alpha$ confidence interval for z and contains true value w.p. $\approx 1 - \alpha$.

σ^2 is unknown and must be estimated. Sample variance can be an estimator: $s^2 := \frac{1}{R-1}\sum_{r=1}^{R}(Z_r - \hat{z})^2 = \frac{\sum_{r=1}^{R} Z_r^2 - R\hat{z}^2}{R-1}$. Replace σ by s in I_α; as $s^2 \to \sigma^2$ in probability, $I_\alpha := \hat{z} \pm z_{1-\alpha}s/\sqrt{R}$. This is the form in which results are reported.

Canonical Rate. Being of the same dimension as the expectation z, the standard deviation σ is the natural measure of precision rather than the variance. The rate $R^{-1/2}$ is intrinsic for virtually any Monte Carlo experiment as the optimal rate which can be achieved. In fact, there are quite a few situations in which one gets slower rates such as $O(R^{-(1/2-\epsilon)})$ with $\epsilon > 0$, e.g., in gradient estimation via finite differences, or variance estimation via time series methods. In view of these remarks, one often refers to $O(R^{-1/2})$ as the canonical Monte Carlo convergence rate. There are, however, a few isolated examples (but just a few!) in which $O(R^{-1/2})$ can in fact be improved to supercanonical (faster) rates.

Output analysis via normal confidence intervals can be and is in fact performed beyond the setting of i.i.d. replications. Let z be the unknown number to be estimated by simulation and consider a sampling-based scheme with sample size R in which $\hat{z} := \hat{z}_R$ is an estimator obeying a CLT of the form $R^{1/2}(\hat{z} - z) \rightsquigarrow N(0, \sigma^2)$. If an estimator $\hat{\sigma}^2 := \hat{\sigma}_R^2$ is available such that $\hat{\sigma}_R^2 \to \sigma^2$, an asymptotic $1 - \alpha$ confidence interval is $\hat{z} \pm z_{1-\alpha}\hat{\sigma}/\sqrt{R}$. Ergodic Markov chain $\{X_n\}_{n\in\mathbb{N}}$ is an example. If f is a function on the state space, then under mild conditions $\hat{z} := (f(X_0) + \ldots + f(X_{R-1}))/R$ has a limit, viz., $z := \mathsf{E}_\pi f(X_0)$ w.r.t. the stationary distribution π. $\hat{z}$ obeys a CLT of the form as above. However, estimation of the variance is more difficult than for the i.i.d. case. In this case, both dependence and nonstationarity may be noted.

Two-Stage and Sequential Procedures

In many computational setting of trial runs followed by production runs (two-stage), 2 accuracy criteria are commonly used:
absolute accuracy: compute the quantity z to an accuracy ϵ;
relative accuracy: compute the quantity z to an accuracy $\epsilon|z|$.
For absolute precision ϵ, choose R s.t. confidence interval half-width $z_{1-\alpha/2}\sigma/\sqrt{R}$ is approximately equal to ϵ: $R \approx \frac{z^2_{1-\alpha/2}\sigma^2}{\epsilon^2}$. For relative precision ϵ, the appropriate number of simulations is $R \approx \frac{z^2_{1-\alpha/2}\sigma^2}{\epsilon^2 z^2}$. Rules for selecting R cannot be implemented directly, since they rely on problem-dependent unknown parameters σ^2 and z^2. The number R of *Production Runs* is given by one of the above formula with σ, z substituted by $\hat{\sigma}_{\text{trial}}, \hat{z}_{\text{trial}}$. The final confidence interval for z may be based on only the production runs or on both the production runs and the trial runs, taking into account the random nature of R and its dependence on the trial runs correlates all the observations in a nontrivial way. When a specific level of accuracy is required, an alternative to using a two-stage procedure is to implement a sequential procedure. A sequential procedure essentially just monitors the confidence interval until the precision criterion is met. A sequential version of the absolute precision algorithm will end up by generating variates $Z_1, \ldots, Z_N$, where the (random) total sample size N is given by $N := \inf\{n > 1 : z_{1-\alpha/2}s_n \leq \epsilon\sqrt{n}\}$. The sample s.d. s_n is recomputed after each additional sampling.

The main difficulty with a sequential procedure is that the standard deviation estimator s_n is highly unreliable at small values of n. If s_n is abnormally small, as might occur in estimating a probability $z = \mathrm{P}A$ via a sample proportion when no occurrences of A have yet been observed by time n (for then $s_n = 0$), this may cause the sequential procedure to shut off at too small a sample size, leading to a poor final estimator for z.
This problem of early termination is the most difficult challenge presented to users of sequential procedures. A second approach to addressing this issue is to modify N as $N' := \inf\{n \geq n_0 : z_{1-\alpha/2} s_n \leq \epsilon\sqrt{n}\}$, so that at least n_0 (roughly 50) simulations are run before initialization of the sequential check of the precision criterion. In this way, one hopes that the standard deviation estimator has stabilized at something close to the correct value. The final $1 - \epsilon$ confidence interval is then given by $z_{N'} \pm \epsilon$ which is asymptotically valid as ϵ decreases to zero. Two implementation details are worth mentioning. The first is that computational effort must be expended on sequentially updating the estimators of σ and z; this can be mitigated by checking the precision criterion every $k > 1$ samples only (rather than after every sample is taken). Secondly, in sequentially updating s_n^2, one wishes to use an algorithm that is both numerically stable and can be updated in constant time (independent of n).

Computing Smooth Functions of Expectations

Although the problem of estimating $z = \mathsf{E}Z$ is a very general one, it does not subsume all computational problems of interest. In particular, suppose that $z = (z_1 \ z_2 \ \ldots \ z_d)^{\mathsf{T}}$ is a d-dimensional vector, in which each component z_i can be expressed as $\mathsf{E}Z(i)$ for some r.v. $Z(i)$. Some applications demand that one compute $f(z)$, where $f : \mathbb{R}^d \to \mathbb{R}$ is an explicitly known and smooth function. This is known as the smooth function estimation problem and occurs in traditional statistics as well as in the analysis of simulation output. The function $f(z) = 1/z$ occurs in Buffon's needle experiment. For another example with this f, let $Z_\cdot$ be i.i.d. failure times of some system. Then $f(z) = 1/z = 1/\mathsf{E}Z$ is the long-run failure intensity.

Suppose that we want to compute the standard deviation σ of an r.v. W. Then $\sigma = f(z)$, where $f(z) = (z_1 - z_2^2)^{1/2}$ and $Z(1) = W_2$, $Z(2) = W$.

Suppose that we want to compute the correlation ρ between two r.v.s X, Y. Then $\rho = f(z)$, where $Z = (X, Y, X^2, Y^2, XY)$, $f(z) = \frac{z_5 - z_1 z_2}{(z_3 - z_1^2)^{1/2}(z_4 - z_2^2)^{1/2}}$.

A perpetuity is an r.v. of the form $Y = B_0 + A_1 B_1 + A_1 A_2 B_2 + A_1 A_2 A_3 B_3 + \ldots$ where all rv's are independent, the B_n are iid with finite mean and the A_n are i.i.d. and positive with $\mathsf{E} \log A_n < 0$. $Y^* := B_1 + A_2 B_2 + A_2 A_3 B_3 + \ldots$ has the same distribution as Y so that $\mathsf{E} = \mathsf{E}(B_0 + A_1 Y^*) = \mathsf{E}B + \mathsf{E}A.\mathsf{E}Y = \frac{\mathsf{E}B}{1 - \mathsf{E}A}$ which as the form $f(z_1, z_2)$.

Discounted rewards of a finite Markov chain $\{X_n\}$, where A_n, B_n have distribution F_i, G_i when $X_n = i$ and one wants $\mathsf{E}_x Y$ for some x, requires simulation.

Set $Z(1) := \sum_{n=0}^{\tau-1} A_1 \ldots A_n B_n$, $Z(2) := A_1 \ldots A_\tau$ so that $\mathsf{E}_x Y = \mathsf{E}_x Z(1) + \mathsf{E}_x Z(2) \mathsf{E}_x Y$.

The obvious estimator of $f(z)$ is $f(\hat{z})$, where $\hat{z} := (Z_1 + \ldots + Z_R)/R$ and the Z_r are i.i.d. replicates of the random vector $Z := (Z(1), \ldots, Z(d))^{\mathsf{T}}$. If f is continuous at z, consistency (i.e., $f(\hat{z}) a.s. \to f(z)$) is guaranteed due to the consistency of $\hat{z}$. The more subtle issue is the rate of convergence of $f(\hat{z})$ and the construction of associated confidence intervals for $f(z)$. A CLT for $f(\hat{z})$ is derived via an appropriate Taylor expansion (Delta Method). Under suitable smoothness assumptions,
$f(\hat{z}) - f(z) = \nabla f(z)(\hat{z} - z) + o(\|\hat{z} - z\|) = (V_1 + \ldots + V_R)/R + o(\|\hat{z} - z\|)$ and
$V_r := \nabla f(z)(Z_r - z)$. Note $\mathsf{E}V_i = 0$. If $\Sigma := \mathsf{Var}\ Z_1$ with the covariances $(i, j$th entry) well defined and finite; then $\|\hat{z} - z\| = O(R^{-1/2})$ by the CLT and
$\sigma^2 := \mathsf{Var}\ V_1 = \nabla f \Sigma (\nabla f)^{\mathsf{T}} < \infty$. Multiplying by $R^{1/2}$, we obtain
$R^{1/2}(f(\hat{z}) - f(z)) \rightsquigarrow N(0, \sigma^2)$. Similar to 2-stage and sequential processes. However differences are: Firstly, the estimator $f(\hat{z})$ is generally biased as an estimator of f(z), in the sense that $\mathsf{E}f(\hat{z}) = f(z)$. More precisely, note that if f is twice differentiable at z, then $f(\hat{z}) - f(z) = \nabla f(z)(\hat{z} - z) + \frac{1}{2}(\hat{z} - z)^{\mathsf{T}} H(z)(\hat{z} - z) + o(\|\hat{z} - z\|^2)$ where $H(.)$ is the Hessian. Taking expectations, we get
$\mathsf{E}f(\hat{z}) - f(z) = \frac{1}{2R} \sum_{i,j=1}^{d} \mathsf{Cov}(Z(i), Z(j)) H_{i,j}(z) + o(1/R)$ as $R \to \infty$. Under additional regularity conditions, a further Taylor expansion shows that the $o(1/R)$ term is actually of the form $\beta/R^2 + o(1/R^2)$ for some constant β.
A possible means of reducing bias would be to use the modified estimator
$f(\hat{z}) - \frac{1}{2R} \hat{\mathsf{Cov}}(Z(i), Z(j)) H_{i,j}(\hat{z})$ where $\hat{\mathsf{Cov}}_{i,j}$ is the sample covariance between $Z(i)$ and $Z(j)$ based on $Z_1, ..., Z_R$. Because the difference is $O(1/R)$, the last estimate satisfies precisely the same CLT as does $f(\hat{z})$. Consequently, the lower bias of does not improve the asymptotic rate of convergence. Bias corrections are often implemented as a rule of thumb hoping to better estimate small-sample properties and not universally.

To compute the estimator requires calculating $(d+1)d/2$ different Hessian entries, as well as estimating $(d+1)d/2$ different sample covariances.
The second key difference in the CLT relative to $\sqrt{R}(\hat{z}-z) \to N(0,\sigma^2)$ is that estimating the variance is harder. Note that we cannot observe the V_r, since the definition involves the unknown z. Of course, if R is large, $\nabla f(\hat{z})(Z_r - \hat{z})$ should be close to $\nabla f(z)(Z_r - z)$, suggesting that we use the variance estimator $\hat{\sigma}^2 := \frac{1}{R-1}\sum_{r=1}^{R}(\nabla f(\hat{z})(Z_r-\hat{z}))^2$. The challenge in computing $\hat{\sigma}^2$ is the need to explicitly compute the gradient of f. For complex functions or high d, one may wish to avoid this calculation. In any case, given a suitable estimator $\hat{\sigma}^2$ with $\hat{\sigma}^2 \to \sigma^2$ as $R \to \infty$, the generation of an asymptotic $1-\alpha$ confidence interval for $f(z)$ is straightforward with $\hat{z}$ replaced by $f(\hat{z})$. Centering the interval on the last estimator instead of $f(\hat{z})$ would also produce an asymptotically valid procedure.

Computing Roots of Equations Defined by Expectations

Problem is to find a root θ^* to an equation of the form $f(z, \theta) = 0$ where f is explicitly known and $z := \mathsf{E}Z \in \mathbb{R}^d$, $\theta \in \mathbb{R}$. For certain functions f, $\theta^* = \xi(z)$, where $f(z, \xi(z)) = 0$. If ξ can be computed explicitly and is smooth, this is a root finding problem. When not, iterative techniques are needed to find $\hat{\theta}$ as an empirical root of $f(\hat{z}, \hat{\theta}) = 0$ where $\hat{z}$ is the average of R replicates $Z_.$ of Z.
Given that $\hat{z} \to z$ a.s. as $R \to \infty$, $\hat{\theta} \to \theta^*$ a.s. under mild additional regularity conditions on f. To derive asymptotically valid confidence intervals, develop a CLT for $\hat{\theta}$. It is evident that
$0 = f(\hat{z}, \hat{\theta}) - f(z, \theta^*) = f(\hat{z}, \hat{\theta}) - f(z, \hat{\theta}) + f(z, \hat{\theta}) - f(z, \theta^*)$. The delta method establishes that if f is sufficiently smooth,
$\nabla_z(z, \theta^*)(\hat{z} - z) + f_\theta(z, \theta^*)(\hat{\theta} - \theta^*) + o(\|\hat{z} - z\|) = 0$, where f_θ isthe partial derivative w.r.t. θ. This leads to the CLT $R^{1/2}(\hat{\theta} - \theta^*) \rightsquigarrow N(0, \sigma^2)$ as $R \to \infty$, $f_\theta \neq 0$, where $\sigma^2 = \frac{\mathsf{Var}(\nabla_z f(z,\theta^*)Z)}{f_\theta(z,\theta^*)}$. A sample estimate of σ^2 is therefore
$\hat{\sigma}^2 = \frac{\frac{1}{R-1}\sum_{r=1}^{R}[\nabla_z f(\hat{z},\hat{\theta})(Z_r - z)]^2}{f_\theta(\hat{z},\hat{\theta})^2}$ from which the asymptotic confidence interval is computed with $\hat{z}$ replaced by $\hat{\theta}$ in rge confidence interval formula. A still more demanding root-finding problem arises when the objective is to find the root θ^* of an equation of the form $z(\theta) = 0$, where $z(\theta) = \mathsf{E}Z(\theta)$ for each θ.

Stochastic Counterpart Method. Suppose that we wish to numerically minimize a convex function $w(\theta)$ over $\theta \in \mathbb{R}$, where $w(\theta) = \mathrm{E}W(\theta)$. This requires computing the root θ to $z(\theta) = 0$, where $z(\theta) = \mathrm{E}W'(\theta)$. Here, we have assumed that $W(\theta)$ is smooth in θ and that the derivative interchange is valid.
The natural estimator θ^* is of course the root to $\hat{z}(\hat{\theta})$, where $\hat{z}(\theta) := (Z_1(\theta) + \ldots + Z_R(\theta))/R$. Obviously, this assumes that an efficient numerical procedure for computing the root of the random function $\hat{z}(.)$ is used. Under suitable regularity conditions, $\hat{\theta} \to \ a.s.\ \theta^*$ as $R \to \infty$.
Suppose $Z(\theta)$ is smooth in θ and that the derivative interchange with the expectation is valid. Because θ^* and $\hat{\theta}$ are the roots of their respective equations, $\hat{z}(\hat{\theta}) - \hat{z}(\theta^*) = z(\theta^*) - \hat{z}(\theta^*)$. But $\hat{z}(\hat{\theta}) - \hat{z}(\theta^*) = \hat{z}'(\xi)(\hat{\theta} - \theta^*)$, where ξ lies in between $\hat{\theta}$ and θ^*. If $z'(\theta^*) \neq 0$, then applying the CLT to the r.h.s. shows that $R^{1/2}(\hat{\theta} - \theta^*) \rightsquigarrow N(0, \sigma^2),\ \sigma^2 = \frac{\mathrm{Var}\ Z(\theta^*)}{z'(\theta^*)^2}$. Here the natural estimator for σ^2 is $\hat{\sigma}^2 = \frac{1}{\hat{z}'(\hat{\theta})^2(R-1)} \sum_{r=1}^{R} Z_r(\hat{\theta})^2$ from which the asymptotic confidence interval is computed.
Assume that $W(.) \in C^2$, with the interchange valid. Then $\hat{\sigma}^2$ takes the form $\frac{1}{R-1} \sum_{r=1}^{R} W_r'(\hat{\theta})^2 / (\frac{1}{R} \sum_{r=1}^{R} W_r''(\hat{\theta}))^2$. The problem of computing a root to $z(\theta) = 0$ becomes more challenging when $Z(\theta)$ is not smooth in θ.