

MA3227 Numerical Analysis II

Lecture 23: Simulation of Random Variables

Simon Etter



2019/2020

Simulation of Random Variables

Introduction

In Lecture 22, we introduced random variables $X \sim \mathcal{X}$ as functions $X : \Omega \rightarrow \Xi$ defined on some unspecified probability space (Ω, P) such that

$$P(X \in A) = P(X^{-1}(A)) = \mathcal{X}(A) \quad \text{for all } A \subset \Xi. \quad (1)$$

In order to work with such random variables, we hence need two things:

- ▶ A probability space (Ω, P) .
- ▶ A function $X(\omega)$ such that (1) is satisfied.

On a computer, the probability space (Ω, P) is almost always given by $\Omega = [0, 1]^n$ and $P(A) = \text{volume}(A)$. The reason for this is that uniformly distributed $\omega_k \in [0, 1]$ can be easily generated by generating a string of bits (e.g. 10110) where each bit is equally likely to be 0 or 1, and then mapping these strings onto equally-spaced points in $[0, 1]$. This process is known as *random number generation*, and I will provide a bit more detail on the next slide.

Once we have a sequence of uniformly distributed numbers $\omega_k \in [0, 1]$, the next task is find a function $X((\omega_k)) \rightarrow \Xi$ such that (1) is satisfied. This is called *simulation* or *sampling* of random variables, and the main aim of this lecture is to introduce several techniques for doing so.

Simulation of Random Variables

Def: Random number generator (RNG)

Any algorithm / piece of hardware which produces a sequence $u_k \in [0, 1]$ which looks as if the u_k were independent samples from a random variable $U \sim \text{Uniform}[0, 1]$.

Remark

For the most part, the u_k produced by RNGs will play the role of the ω_k on the previous slide. Nevertheless, it is common practice to write u_k instead of ω_k to emphasise that the u_k are uniformly distributed in $[0, 1]$.

Discussion of random number generators

There is no rigorous definition of “ u_k which look as if they were independent samples of $U \sim \text{Uniform}[0, 1]$ ”. Instead, there are long lists of tests which you can use to measure how close your RNG is to producing “true” samples of $U \sim \text{Uniform}[0, 1]$.
See e.g. https://en.wikipedia.org/wiki/Diehard_tests.

Simulation of Random Variables

Discussion of random number generators (continued)

RNGs come in two varieties:

- ▶ “True” RNGs use noise in your hardware to produce truly unpredictable u_k .
- ▶ Pseudo RNGs (pRNGs) take in a seed s and return a sequence u_k which is fully deterministic but which looks like a sequence of independent samples of $U \sim \text{Uniform}[0, 1]$.

True RNG are important for applications like cryptography, but for Monte Carlo purposes they have two important drawbacks:

- ▶ They are much slower than pRNGs, see `rng_benchmark()`.
- ▶ They by definition make it impossible to reproduce results, which is a nuisance when you want to debug your code.

For these reasons, we will exclusively consider pRNGs in this module.

Designing a fast and high-quality pRNG is highly non-trivial. Luckily, you will almost surely never have to do this yourself since most programming languages come with a pre-installed pRNG.

Simulation of Random Variables

pRNGs in Julia (and most other programming languages)

The pRNG functionality in Julia is provided by the `rand()` function. This function implicitly defines a sequence u_k and keeps an index k pointing to the current element. Each call to `push()` returns the current u_k and then increments $k \leftarrow k + 1$.

The state of the pRNG can be reset using `Random.seed!()`.

Example

```
julia> Random.seed!(42);
```

```
julia> rand()  
0.5331830160438613
```

```
julia> rand()  
0.4540291355871424
```

```
julia> Random.seed!(42);  
julia> rand()  
0.5331830160438613
```

Note that the argument to `Random.seed!()` is not the index k .

```
julia> Random.seed!(43); rand()  
0.18097523182192754    (not 0.5331830160438613)
```

Simulation of Random Variables

Discussion

The above concludes our discussion of random number generation.

We now move on to simulation of random variables, i.e. the problem of finding $X : [0, 1]^n \rightarrow \Xi$ such that

$$P(X \in A) = P(X^{-1}(A)) = \mathcal{X}(A) \quad \text{for all } A \subset \Xi.$$

Let us begin on with a simple example.

Simulation of Random Variables

Example

Task: Given $U \sim \text{Uniform}[0, 1]$, construct $X(U) \sim \text{Uniform}[a, b]$.
 $\text{Uniform}[a, b]$ is the uniform distribution on the interval $[a, b]$, i.e. we want $P(X \in [c, d]) = \frac{d-c}{b-a}$ for all c, d such that $a \leq c \leq d \leq b$.

Solution: A simple solution is

$$X(U) = a + (b - a) U \quad \Longleftrightarrow \quad X^{-1}(x) = \frac{x-a}{b-a}$$

since then

$$\begin{aligned} P(X(U) \in [c, d]) &= P(U \in X^{-1}[c, d]) \\ &= P\left(U \in \left[\frac{c-a}{b-a}, \frac{d-a}{b-a}\right]\right) \\ &= \frac{d-a}{b-a} - \frac{c-a}{b-a} \\ &= \frac{d-c}{b-a}. \end{aligned}$$

Alternatively, we could set $X(U) = b + (a - b) U$, or we could construct $X(U)$ by piecing together several linear functions whose ranges partition $[a, b]$, etc.

Simulation of Random Variables

Discussion

Constructing a random variable X with a desired target distribution \mathcal{F} was easy in the above example because $\mathcal{F} = \text{Uniform}[a, b]$ was just a linear transformation of initial distribution $\text{Uniform}[0, 1]$.

In general, finding $X(U)$ such that $U \sim \text{Uniform}[0, 1] \implies X \sim \mathcal{F}$ can be quite difficult. The following slides present two strategies for constructing random variables which work for fairly general distributions \mathcal{F} but which may not be very efficient.

Remember that there are many ways how we can construct $X(U)$ such that $X \sim \mathcal{F}$. Of course, it is well possible that some $X(U)$ are easier to evaluate than others.

Simulation of Random Variables

Thm: Transformation sampling

Let \mathcal{F} be a distribution on \mathbb{R} with cumulative distribution function $F(x)$, and assume $U \sim \text{Uniform}[0, 1]$. Then,

$$X = F^{-1}(U) \sim \mathcal{F}.$$

Proof. We have

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x),$$

where in the second step I used the monotonicity of $F(x)$ and in the third step I used $U \sim \text{Uniform}[0, 1]$.

Example

Consider the distribution \mathcal{F} with density function $f(x) = 2x$ on $[0, 1]$ and cumulative distribution function

$$F(x) = \int_0^x 2x' dx' = x^2 \quad \text{for } x \in [0, 1].$$

If $U \sim \text{Uniform}[0, 1]$, then $X = \sqrt{U} \sim \mathcal{F}$.

See `transformation_sampling()`.

Simulation of Random Variables

Pros and cons of transformation sampling

- ▶ Pro: Easy to implement and fast if $F^{-1}(u)$ can be easily computed.
- ▶ Con: Only works for random variables $\Omega \rightarrow \mathbb{R}$.
- ▶ Con: $F^{-1}(u)$ may not be easy to compute.

To illustrate the last point, consider the normal distribution $\mathcal{N}(\mu, \sigma^2)$. There is no known direct formula for the CDF

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right);$$

hence the only way to compute $F^{-1}(u)$ is to evaluate the above integral using quadrature and then apply a root finder to determine $x \in \mathbb{R}$ such that $F(x) = u$.

Simulation of Random Variables

Thm: Rejection sampling

Let \mathcal{F}, \mathcal{G} be two distributions on \mathbb{R}^n with probability density functions $f(x)$ and $g(x)$, respectively, and let

$$U_k \sim \text{Uniform}[0, 1] \quad \text{and} \quad G_k \sim \mathcal{G} \quad \text{be independent.}$$

Assume there exists $M > 0$ such that

$$f(x) \leq M g(x) \quad \text{for all } x \in \mathbb{R}^n.$$

Consider the random variable F defined through the following algorithm.

Algorithm 1 Rejection sampling

```
1: for  $k = 1, 2, \dots$  do
2:   if  $U_k(\omega) \leq \frac{f(G_k(\omega))}{M g(G_k(\omega))}$  then
3:     Return  $F(\omega) = G_k(\omega)$ 
4:   end if
5: end for
```

We then have $F \sim \mathcal{F}$.

Simulation of Random Variables

Proof (not examinable). Let us introduce the abbreviation

$$C_k(\omega) = \begin{cases} 1 & \text{if } U_k(\omega) \leq \frac{f(G_k(\omega))}{M g(G_k(\omega))}, \\ 0 & \text{otherwise.} \end{cases}$$

Using the law of total probability, we obtain

(https://en.wikipedia.org/wiki/Law_of_total_probability)

$$\begin{aligned} P(C_k = 0) &= \int_{\mathbb{R}^n} g(x) P(C_k = 0 \mid G_k = x) dx \\ &= \int_{\mathbb{R}^n} g(x) \left(1 - \frac{f(x)}{M g(x)}\right) dx \\ &= \int_{\mathbb{R}^n} g(x) dx - \frac{1}{M} \int_{\mathbb{R}^n} f(x) dx \\ &= 1 - \frac{1}{M}. \end{aligned}$$

Simulation of Random Variables

Proof (not examinable, continued).

Using that

- ▶ $P(A \cup B) = P(A) + P(B)$ if A, B are disjoint, and
- ▶ $p(x, y) = p_X(x) p_Y(y)$ if $p(X, Y), p_X(x), p_Y(y)$ are the PDFs of two independent random variables X, Y , respectively,

we obtain

$$\begin{aligned}P(F \in A) &= P(G_1 \in A, C_1 = 1) + P(G_2 \in A, C_1 = 0, C_2 = 1) + \dots \\&= \int_A g(x) \frac{f(x)}{M g(x)} dx + \left(1 - \frac{1}{M}\right) \int_A g(x) \frac{f(x)}{M g(x)} dx + \dots \\&= \frac{1}{M} \int_A f(x) dx \left(\sum_{k=0}^{\infty} \left(1 - \frac{1}{M}\right)^k \right) \\&= \frac{1}{M} \int_A f(x) dx \frac{1}{1 - \left(1 - \frac{1}{M}\right)} \\&= \int_A f(x) dx,\end{aligned}$$

i.e. $F \sim \mathcal{F}$ as claimed.

Simulation of Random Variables

Example

Consider again the distribution \mathcal{F} with density function $f(x) = 2x$ on $[0, 1]$, and set $\mathcal{G} = \text{Uniform}[0, 1]$ with density function $g(x) = 1$.

We have $f(x) \leq 2g(x)$, i.e. $M = 2$ in the notation of the rejection sampling theorem. We can hence generate a sample f according to \mathcal{F} by generating samples g_k according to \mathcal{G} and u_k according to $\text{Uniform}[0, 1]$, and setting $f = g_k$ where k is the smallest integer such that

$$u_k \leq \frac{f(g_k)}{M g(g_k)} = \frac{2g_k}{2 \times 1} = g_k.$$

See `rejection_sampling()` for numerical demonstration.

Moreover, we have seen in the proof of the rejection sampling theorem that the probability for accepting a proposal g_k is $P(C_k = 1) = \frac{1}{M}$.

Hence,

$$\mathbb{E}[\text{number of tries until accepted}] = \frac{1}{P(C_k=1)} = M$$

This is the expectation value of a geometrically distributed random variable with success probability $P(C_k = 1) = \frac{1}{M}$.

See https://en.wikipedia.org/wiki/Geometric_distribution.

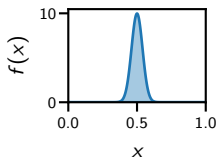
Simulation of Random Variables

Pros and cons of rejection sampling

- ▶ Pro: Works for fairly general distributions \mathcal{F} . All we need is a sampleable proposal distribution \mathcal{G} such that $M = \sup_x \frac{f(x)}{g(x)} < \infty$.
- ▶ Con: Can be very inefficient: we have seen that $M = \sup_x \frac{f(x)}{g(x)}$ is the expected number of samples of \mathcal{G} required to generate a single sample of \mathcal{F} ; thus if M is very large, we generate and discard many samples of \mathcal{G} to generate a single sample of \mathcal{F} .

Example

Consider the distribution



If we choose $\mathcal{G} = \text{Uniform}[0, 1]$, we need $M = 10$ samples of \mathcal{G} to generate a single sample of \mathcal{F} .

Simulation of Random Variables

Discussion

The goal of the transformation and rejection sampling is to generate samples of some random variable F which we cannot sample otherwise. However, the aim of Monte Carlo algorithms is to compute $\mathbb{E}[F]$, and sampling F is only a means towards this end. The following result shows that it is possible to compute $\mathbb{E}[F]$ even if we can only sample some other random variable G .

Thm: Importance sampling

Let \mathcal{F}, \mathcal{G} be distributions with probability densities $f(x)$ and $g(x)$, respectively, and let $F \sim \mathcal{F}$, $G \sim \mathcal{G}$. Then,

$$\mathbb{E}[F] = \mathbb{E}\left[G \frac{f(G)}{g(G)}\right].$$

Proof.

$$\mathbb{E}[F] = \int_{\Omega} x f(x) dx = \int_{\Omega} x \frac{f(x)}{g(x)} g(x) dx = \mathbb{E}\left[G \frac{f(G)}{g(G)}\right].$$

Simulation of Random Variables

Example

Consider again the distribution \mathcal{F} with density function $f(x) = 2x$ on $[0, 1]$, and set $\mathcal{G} = \text{Uniform}[0, 1]$ with density function $g(x) = 1$.

Assuming $F \sim \mathcal{F}$ and $G \sim \mathcal{G}$, we then have

$$\mathbb{E}[F] = \mathbb{E}\left[G \frac{f(G)}{g(G)}\right] = \mathbb{E}\left[G \frac{2G}{1}\right] = \mathbb{E}[2G^2].$$

This is easily confirmed analytically,

$$\mathbb{E}[F] = \int_0^1 x \, 2x \, dx = \frac{3}{2} = \int_0^1 2x^2 \, dx = \mathbb{E}[2G],$$

and demonstrated numerically in `importance_sampling()`.

Simulation of Random Variables

Discussion

Recall from Lecture 22 that

$$\mathbb{E} \left[\tilde{\mathbb{E}}_N[F] - \mathbb{E}[F] \right] = \sqrt{\frac{1}{N} \text{Var}[F]}.$$

After applying the importance sampling trick, we hence obtain

$$\mathbb{E} \left[\tilde{\mathbb{E}}_N \left[G \frac{f(G)}{g(G)} \right] - \mathbb{E}[F] \right] = \sqrt{\frac{1}{N} \text{Var} \left[G \frac{f(G)}{g(G)} \right]},$$

which shows that the error becomes larger if we choose \mathcal{G} such that

$$\text{Var} \left[G \frac{f(G)}{g(G)} \right] > \text{Var}[F].$$

Surprisingly, it is sometimes also possible to reduce the variance using the importance sampling trick as demonstrated in the example on the next slide.

Simulation of Random Variables

Example

Consider the random variables $F \sim \text{Uniform}[0, 1]$ and $G \sim \mathcal{G}$ where \mathcal{G} has probability density $g(x) = 2x$ on $[0, 1]$.

We then have

$$\text{Var}[F] = \mathbb{E}[F^2] - \mathbb{E}[F]^2 = \int_0^1 x^2 dx - \left(\int_0^1 x dx \right)^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$$

but

$$\text{Var}\left[G \frac{f(G)}{g(G)}\right] = \text{Var}\left[G \frac{1}{2G}\right] = \text{Var}\left[\frac{1}{2}\right] = 0,$$

that is

$$\mathbb{E}\left[\tilde{\mathbb{E}}_N[F] - \mathbb{E}[F]\right] = \sqrt{\frac{1}{12N}} \quad \text{but} \quad \mathbb{E}\left[\tilde{\mathbb{E}}_N\left[G \frac{f(G)}{g(G)}\right] - \mathbb{E}[F]\right] = 0.$$

Simulation of Random Variables

Summary

- ▶ Pseudo random number generator (pRNG): sequence $u_k \in [0, 1]$ such that u_k “looks like samples of $U \sim \text{Uniform}[0, 1]$ ”.
- ▶ Transformation sampling: $X = F^{-1}(U)$ with $U \sim \text{Uniform}[0, 1]$ is distributed according to the CDF $F(x)$.
- ▶ Rejection sampling: propose samples according to a proposal distribution \mathcal{G} and then reject with probability $\frac{f(x)}{M g(x)}$ to produce samples according to \mathcal{F} .
- ▶ Importance sampling: $\mathbb{E}[F] = \mathbb{E}[G \frac{f(G)}{g(G)}]$.