Chapter 5

Variance Reduction Techniques

The goal of Variance Reduction techniques is to increase the precision of the MC estimates that can be obtained for a given computational effort, by exploiting known information about the simulation model. The more that is known about the behavior of the system, the greater the amount of variance reduction that can be achieved. The main variance reduction techniques discussed in this chapter are: *(i)* antithetic random variables; *(ii)* Importance sampling; *(iii)* control variates; *(iv)* stratified sampling, and *(v)* Latin hypercube sampling.

Let Z be a random variable, output of a stochastic model. We assume that Z is a function, $Z = \Psi(\mathbf{X})$ of a random vector $\mathbf{X} = (X_1, \dots, X_d)$ with PDF $f : \mathbb{R}^d \to \mathbb{R}_+$ and consider the goal of computing the expectation of Z:

$$\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}.$$

The *crude* Monte Carlo approach to approximate μ consists in generating N i.i.d. replicas $Z^{(1)}, \dots, Z^{(N)}$, with $Z^{(i)} = \Psi(\mathbf{X}^{(i)})$, $\mathbf{X}^{(i)} \overset{i.i.d.}{\sim} f$ and computing

$$\hat{\mu}_{CMC} = \frac{1}{N} \sum_{i=1}^{N} Z^{(i)}.$$

By CLT we have that, as $N \to \infty$,

$$|\mu - \hat{\mu}_{CMC}| \leq z_{1-\frac{\alpha}{2}} \frac{\sqrt{\mathrm{Var}[Z]}}{\sqrt{N}} \qquad \text{with probability } 1-\alpha \text{ asymptotically.}$$

The techniques of *variance reduction* aim at improving the performance of a Monte Carlo approximation by reducing the constant $\sqrt{\mathrm{Var}[Z]}$, whence the name *variance reduction*. The idea is rather simple: instead of applying the mean estimator $\hat{\mu} = \hat{\mu}(Z)$ to the variable Z, one applies it to a cleverly modified version $\tilde{Z}$ which satisfies

$$\mathbb{E}\left[\tilde{Z}\right] = \mathbb{E}[Z] = \mu \qquad \text{and} \qquad \mathrm{Var}\left[\tilde{Z}\right] \ll \mathrm{Var}[Z].$$

Hence, a Monte Carlo approximation with variance reduction will look like

$$\hat{\mu}_{VR} = \frac{1}{N} \sum_{i=1}^{N} \tilde{Z}^{(i)}, \qquad \text{with } \tilde{Z}^{(i)} \text{ i.i.d., } \mathbb{E}\left[\tilde{Z}\right] = \mu.$$

we can only work on Var(Z) since we cannot improve the $\sqrt{N}$ at the denominator (so, without working on Var(Z) the only way to have tighter bounds is to increase N)

5.1 Antithetic Variables

Suppose N even. Instead of generating N i.i.d. replicas, the idea of antithetic sampling is to generate $N/2$ i.i.d. pairs of negatively correlated random variables

$$(Z^{(1)}, Z^{(2)}),\ (Z^{(3)}, Z^{(4)}),\ \dots, (Z^{(N-1)}, Z^{(N)}) \quad \text{i.i.d.}$$

the pairs are iid, inside the pair the variables are correlated (negatively)

where all $Z^{(i)}$ have the same distribution as Z but

$$Cov(Z^{(2i-1)}, Z^{(2i)}) < 0, \quad i = 1, \dots, N/2.$$

If we now we consider the estimator

$$\hat{\mu}_{AV} = \frac{1}{N}\sum_{i=1}^{N} \tilde{Z}^{(i)} = \frac{1}{N/2}\sum_{i=1}^{N/2} \frac{Z^{(2i-1)} + Z^{(2i)}}{2}$$

it follows immediately that

1. $\mathbb{E}[\hat{\mu}_{AV}] = \mathbb{E}[Z] = \mu$; since $\forall i\ \mathbb{E}[Z^{(i)}] = \mu$

2. $\mathrm{Var}[\hat{\mu}_{AV}] \overset{1}{=} \frac{4}{N^2}\sum_{i=1}^{N/2} \mathrm{Var}\left[\frac{Z^{(2i-1)} + Z^{(2i)}}{2}\right] \overset{1}{=} \frac{1}{2N}\mathrm{Var}\left[Z^{(1)} + Z^{(2)}\right]$

$$\overset{2}{=} \frac{1}{2N}\left(\mathrm{Var}\left[Z^{(1)}\right] + \mathrm{Var}\left[Z^{(2)}\right] + 2Cov(Z^{(1)}, Z^{(2)})\right) \overset{3}{=} \frac{\mathrm{Var}[Z] + Cov(Z^{(1)}, Z^{(2)})}{N}.$$

1. each couple is iid
2. $Cov(X,Y) = \frac{1}{2}(Var(X+Y) + Var(X) + Var(Y))$
3. $Var(Z^{(1)}) = Var(Z^{(2)}) = Var(Z)$ (since all $Z^{(i)}$ have the same distribution as Z)

Since, by assumption, $Cov(Z^{(1)}, Z^{(2)}) < 0$, we have achieved variance reduction, that is,

$$\mathrm{Var}[\hat{\mu}_{AV}] < \frac{\mathrm{Var}[Z]}{N} = \mathrm{Var}[\hat{\mu}_{CMC}] \qquad \text{at the same cost of } CMC.$$

The question is now how to generate pairs of *negatively correlated* variables $(Z^{(2i-1)}, Z^{(2i)})$. Often, Z is generated as $Z = \Psi(X_1, \dots, X_d)$ starting from a sequence of i.i.d. random variables $X_1, \dots, X_d$, typically uniformly distributed. The following result holds:

Proposition 5.1.1. *Assume that the PDF of X_i is symmetric about its mean $\mathbb{E}[X]$, and that $\Psi : \mathbb{R}^d \to \mathbb{R}$ is a monotone function of each of its arguments. Then*

$$\Psi(\mathbf{X}) \text{ and } \Psi(2\mathbb{E}[\mathbf{X}] - \mathbf{X}) \text{ are negatively correlated.}$$

The previous proposition follows from the

Chebyshev Covariance Inequality
If X is a real-valued random variable with PDF $f : \mathbb{R} \to \mathbb{R}_+$ and $g, h : \mathbb{R} \to \mathbb{R}$ are nondecreasing functions, such that $\mathbb{E}[g(X)], \mathbb{E}[h(X)], \mathbb{E}[g(X)h(X)] < +\infty$, then

$$Cov(g(X), h(X)) \geq 0.$$

this is also true if X is a vector $\vec{X}$ (we have an extension in multidimension)

Proof. Let

$$\tilde{g}(X) = g(X) - \mathbb{E}[g(X)], \qquad \tilde{h}(X) = h(X) - \mathbb{E}[h(X)].$$

Observe first that

$$\frac{1}{2}\iint (\tilde{g}(x) - \tilde{g}(y))(\tilde{h}(x) - \tilde{h}(y)) f(x) f(y) dx dy = \int \tilde{g}(x)\tilde{h}(x) f(x) dx - \underbrace{\int \tilde{g}(x) f(x) dx}_{=0} \underbrace{\int \tilde{h}(y) f(y) dy}_{=0}.$$

Hence

$$\begin{aligned} Cov(g(X), h(X)) &= \frac{1}{2}\iint (g(x) - g(y))(h(x) - h(y)) f(x) f(y) dx dy \\ &= \frac{1}{2}\int_{x \geq y} (g(x) - g(y))(h(x) - h(y)) dF(x) dF(y) \\ &\quad + \frac{1}{2}\int_{y \geq x} (g(y) - g(x))(h(y) - h(x)) dF(x) dF(y) \geq 0 \end{aligned}$$

thanks to the monotonicity of g and h. □

Remark 5.1.1. *The previous inequality generalizes to the multivariate case: let* $\mathbf{X} \in \mathbb{R}^d$ *with PDF* $f : \mathbb{R}^d \to \mathbb{R}_+$ *and* g, h *nondecreasing functions in each argument. Then* $Cov(g(\mathbf{X}), h(\mathbf{X})) \geq 0$.

In the context of antithetic variables, observe that if Ψ is e.g. nondecreasing in each argument, then $\Psi(2\mathbb{E}[\mathbf{X}] - \cdot)$ is increasing so that $Cov(\Psi(\cdot), -\Psi(2\mathbb{E}[\mathbf{X}] - \cdot)) \geq 0$ – analogously if Ψ is nonincreasing.

Remark 5.1.2. *Note that the theorem specifies sufficient conditions for a variance reduction, but not necessary conditions. This means that it is still possible to obtain a variance reduction even if the conditions of the theorem are not satisfied. For example, if* $\Psi(\cdot)$ *is "mostly " monotonic, then a variance reduction, and possibly a substantial one, might be still be obtained.*

Under the assumptions of the previous proposition, a Monte Carlo approximation of $\mu = \mathbb{E}[Z]$ with antithetic variables can be constructed by the following algorithm:

Algorithm 12 Antithetic Variables

1: Generate $N/2$ i.i.d. replicas $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(N/2)}$ of $\mathbf{X}$
2: Compute

$$\hat{\mu}_{AV} = \frac{1}{N/2} \sum_{i=1}^{N/2} \frac{\Psi(\mathbf{X}^{(i)}) + \Psi(2\mathbb{E}[\mathbf{X}^{(i)}] - \mathbf{X}^{(i)})}{2}.$$

– we need to know $\mathbb{E}[\vec{X}]$ but that is something that we know (we don't know $\mathbb{E}[Z]$)

Then, $\hat{\mu}_{AV}$ is unbiased with $\mathrm{Var}[\hat{\mu}_{AV}] < \mathrm{Var}[\hat{\mu}_{CMC}]$.

Example 5.1.1. *Let* $Z \sim Exp(\lambda)$. *Then* $Z = -\frac{1}{\lambda} \log X = \Psi(X)$ *with* $X \sim \mathcal{U}(0,1)$ *and* Ψ *monotone. It then follows that* $\Psi(x)$ *and* $\Psi(1 - X)$ *are negatively correlated and a Monte Carlo estimator with antithetic variables for the computation of*

$$\mu = \frac{1}{\lambda} = \mathbb{E}[Z]$$

is

$$\hat{\mu}_{AV} = \frac{1}{N/2} \sum_{i=1}^{N/2} \frac{\Psi X^{(i)}) + \Psi(1 - X^{(i)})}{2}, \qquad \text{with} \quad X^{(i)} \overset{i.i.d.}{\sim} \mathcal{U}(0,1).$$

Remark 5.1.3. *If we want to compute* $\mathbb{E}[g(X)]$ *where* $X \sim \mathcal{U}(0,1)$ *and* g *is nondecreasing, we can use the estimator*

$$\hat{\mu}_{AV} = \frac{1}{N} \sum_{i=1}^{N/2} (g(X^{(i)}) + g(1 - X^{(i)})), \qquad X^{(i)} \overset{i.i.d.}{\sim} \mathcal{U}(0,1)$$

whose variance is $\mathrm{Var}[\hat{\mu}_{AV}] = \mathrm{Var}[\hat{\mu}_{CMC}] + \frac{1}{N} Cov(g(X), g(1 - X))$.

The generalization to dimensions $d > 1$ *is also straightforward: if* Ψ *is nondecreasing in each argument, then for a set* $\mathbf{X} = (X_1, \ldots, X - d)$ *of i.i.d.* $\mathcal{U}(0,1)$ *random variables we have that* $Cov(\Psi(\mathbf{X}), \Psi(1 - \mathbf{X}) < 0$ *where* $Cov(\Psi(\mathbf{X}), \Psi(1 - \mathbf{X}) = Cov(\Psi(X_1, \ldots, X_d), \Psi(1 - X_1, \ldots, 1 - X_d))$.

We remark that when using MC estimators averages of quantities $\Psi(\mathbf{X}^{(i)})$, then the randomness in the algorithm leads to some error cancellation. In antithetic sampling we try to get even more cancellation. An antithetic sample is one that somehow gives the opposite value of $\Psi(\mathbf{x})$, being low when $\Psi(\mathbf{x})$ is high and vice versa.

Ordinarily, we get an opposite Ψ by sampling at a point $\mathbf{x}$ that is somehow opposite to $\mathbf{x}$. In the case $\mathbf{X} \sim f$, we ask that f has a *symmetric* density – here, symmetry is meant as *reflection through the center point* $\mathbf{c}$ of the domain of $\mathbf{X}$, that is, $f(\tilde{\mathbf{x}}) = f(\mathbf{x})$ including the constraint that $\mathbf{x} \in D$ if and only if $\tilde{\mathbf{x}} \in D$.

For basic examples, when f is $N(\mathbf{0}, \Sigma)$, then $\tilde{\mathbf{x}} = -\mathbf{x}$, and when f is $\mathcal{U}(0,1)^d$ we have $\tilde{\mathbf{x}} = 1 - \mathbf{x}$ componentwise. Therefore, the rationale for antithetic sampling is that each value of $\mathbf{x}$ is balanced by its opposite $\tilde{\mathbf{x}}$ satisfying[1] $(\mathbf{x} + \tilde{\mathbf{x}})/2 = \mathbf{c}$. Whether this balance is helpful depends on f.

- **Example 5.1.2.** *We can estimate the integral* $I = \int_0^\infty \ln(1+x^2)e^{-x}dx$ *using MC with antithetic variables as follows. First, exploit the change of variable* $t = 1 - e^{-x}$*; indeed,*

$$I = \int_0^\infty \ln(1+x^2)e^{-x}dx = \int_0^1 (1-t)\log(1+(\log(1-t))^2)dt.$$

since $t \sim \mathcal{U}([0,1])$ then also $1-t \sim \mathcal{U}([0,1])$, hence we can interpret the integral as: $\int_0^1 \underbrace{(1-t)}_{f(t)} \underbrace{\log(1+\log^2(1-t))^2}_{\Psi(t)} dt$ and so: $\int_0^1 f(t)\Psi(t)\,dt = \mathbb{E}[\Psi(X)]$, we generate N $X^{(i)} \sim f = \mathcal{U}$ and then we approximate as $\mathbb{E}[\Psi(X)] \approx \frac{1}{N}\sum_{i=1}^N \Psi(X)$ using crude MC (or we can use the antithetic variables MC)

Observe that if $X \in \mathcal{U}(0,1)$*, then also* $1 - X \in \mathcal{U}(0,1)$*. In* `Matlab`*:*

```
N = 1000; g = @(t)log(1+log(1-t).^2);

% crude MC
T = rand(1,N); X = g(T);
disp([mean(X) std(X) std(X)/sqrt(N)])
  0.6582    0.7399    0.0234

% MC with antithetic variables:
T = rand(1,N/2); X = ( g(T) + g(1-T) )/2;
disp([mean(X) std(X) std(X)/sqrt(N/2)])
   0.6927    0.3148    0.0141
```

the standard deviation with antithetic variables is much smaller. So, we reached variance reduction. However, remember that the **assumptions** of Ψ being non increasing or non decreasing is **heavy**

Hands-On Problem 5.1 focuses on the MC method with antithetic variables.

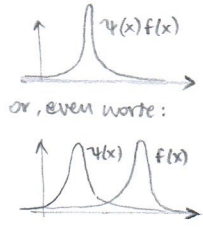
5.2 Importance Sampling

Especially useful for estimation of rare-event probabilities. The importance sampling is a clever trick to perform a sort of change of variables in the integration, introducing a new probability distribution function that weights in a different way events that regarding the original pdf (f) can be with very small measure

One of the most important variance reduction techniques is importance sampling. This technique is especially useful for the estimation of rare-event probabilities.

Let $\mathbf{X} \in \mathbb{R}^d$ be a random vector with PDF $f : \mathbb{R}^d \to \mathbb{R}_+$ and $Z = \Psi(\mathbf{X})$ with $\Psi : \mathbb{R}^d \to \mathbb{R}$. Then, computing the expected value of Z turns into computing the multidimensional integral

$$\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}.$$



The standard Monte Carlo method may not be efficient if the integrand function has large values concentrated in small regions of the domain. The importance sampling technique consists of introducing an auxiliary PDF $g : \mathbb{R}^d \to \mathbb{R}_+$ such that $g(\mathbf{x}) = 0$ only if $\Psi(\mathbf{x})f(\mathbf{x}) = 0$ and $\int_{\mathbb{R}^d} g(\mathbf{x})d\mathbf{x} = 1$. Then, the integral can be rewritten as

$$\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \left(\frac{\Psi(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}\right) g(\mathbf{x})d\mathbf{x} = \mathbb{E}_g\left[\frac{\Psi(\mathbf{X})f(\mathbf{X})}{g(\mathbf{X})}\right] = \mathbb{E}_f[\Psi(\mathbf{X})]$$

where $\mathbb{E}_g[\cdot]$ denotes expectation under the measure $g(\mathbf{x})d\mathbf{x}$. Our original goal is then to find $\mathbb{E}_f[\Psi(\mathbf{X})]$ and, by making a multiplicative adjustment to Ψ, we compensate for sampling from g instead of f. Here f is also called the nominal distribution, while g is the *importance distribution*[2] and must enjoy the property of making more likely those regions in which $\Psi(\mathbf{x})f(\mathbf{x})$ is large.

In other words, importance sampling is helpful is cases where $\Psi(\mathbf{x})$ is largest for values of $\mathbf{x}$ that are unlikely in the f probability distribution: then, we seek a g distribution that puts more weight on the most important (for the expected value of Ψ) $\mathbf{x}$ values.

[1] Indeed, if we reflect $\mathbf{x}$ through $\mathbf{c}$ we get the point $\tilde{\mathbf{x}}$ with $\tilde{\mathbf{x}} - \mathbf{c} = -(\mathbf{x} - \mathbf{c})$, that is $\tilde{\mathbf{x}} = 2\mathbf{c} - \mathbf{x}$.

[2] The distribution g does not have to be positive everywhere; it is enough that $g(\mathbf{x}) > 0$ whenever $\Psi(\mathbf{x}))f(\mathbf{x}) \neq 0$.

It follows that in a Monte Carlo approach, instead of generating i.i.d. replicas of $\mathbf{X}$ to estimate $\mu = \mathbb{E}[\Psi(\mathbf{X})]$, we could generate i.i.d. replicas of $\tilde{\mathbf{X}}$ having PDF g, and estimate

$$\mu = \mathbb{E}_g\left[\frac{\Psi(\tilde{\mathbf{X}})f(\tilde{\mathbf{X}})}{g(\tilde{\mathbf{X}})}\right].$$

This technique is known as *importance sampling* and $w(\mathbf{x}) = f(\mathbf{x})/g(\mathbf{x})$ is called *likelihood ratio*. We thus obtain the following algorithm.

Algorithm 13 Importance Sampling

1: Generate N i.i.d. replicas $\tilde{\mathbf{X}}^{(1)}, \ldots, \tilde{\mathbf{X}}^{(N)} \sim g$

2: Compute

$$\hat{\mu}_{IS} = \frac{1}{N}\sum_{i=1}^{N}\frac{\Psi(\tilde{\mathbf{X}}^{(i)})f(\tilde{\mathbf{X}}^{(i)})}{g(\tilde{\mathbf{X}}^{(i)})}$$

3: Estimate

$$\hat{\sigma}^2_{IS} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{\Psi(\tilde{\mathbf{X}}^{(i)})f(\tilde{\mathbf{X}}^{(i)})}{g(\tilde{\mathbf{X}}^{(i)})} - \hat{\mu}_{IS}\right)^2$$

4: Output $\hat{\mu}_{IS}$ and a (asymptotic) $1-\alpha$ confidence interval

$$\hat{I}_\alpha = \left[\hat{\mu}_{IS} - z_{1-\frac{\alpha}{2}}\frac{\hat{\sigma}_{IS}}{\sqrt{N}}, \hat{\mu}_{IS} + z_{1-\frac{\alpha}{2}}\frac{\hat{\sigma}_{IS}}{\sqrt{N}}\right]$$

Note that to use $\hat{\mu}_{IS}$ we must be able to compute $\Psi f/g$. Assuming that we can compute Ψ, this estimate requires that we can compute $f(\mathbf{x})/g(\mathbf{x})$ at any $\mathbf{x}$ we might sample.

It is indeed straightforward to prove the following

Proposition 5.2.1. *The importance sampling estimator of* $\mu = \mathbb{E}_f[\Psi(\mathbf{X})]$,

$$\hat{\mu}_{IS} = \frac{1}{N}\sum_{i=1}^{N}\frac{\Psi(\tilde{\mathbf{X}}^{(i)})f(\tilde{\mathbf{X}}^{(i)})}{g(\tilde{\mathbf{X}}^{(i)})}, \qquad \tilde{X}^{(i)} \overset{i.i.d.}{\sim} g, \tag{5.1}$$

is unbiased – that is, $\mathbb{E}_g[\hat{\mu}_{IS}] = \mu$ *– and has variance*

$$\mathrm{Var}\,[\hat{\mu}_{IS}] = \frac{1}{N}\left(\int\frac{(\Psi(\mathbf{x})f(\mathbf{x}))^2}{g(\mathbf{x})}d\mathbf{x} - \mu^2\right) = \frac{1}{N}\left(\mathbb{E}_g\left[\left(\frac{\Psi(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}\right)^2\right] - \mu^2\right). \tag{5.2}$$

Proof. It is indeed quite simple to show that

$$\mathbb{E}_g\left[\frac{\Psi(\mathbf{X})f(\mathbf{X})}{g(\mathbf{X})}\right] = \int\frac{\Psi(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x} = \int\Psi(\mathbf{x})f(\mathbf{x})d\mathbf{x} = \mu.$$

and that

$$\mathrm{Var}\,[\hat{\mu}_{IS}] = \frac{1}{N}\left(\int\left(\frac{\Psi(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}\right)^2 g(\mathbf{x})d\mathbf{x} - \mu^2\right)$$

so that, by simple rearrangement, we find that

$$\mathrm{Var}\,[\hat{\mu}_{IS}] = \frac{1}{N}\left(\int\frac{(\Psi(\mathbf{x})f(\mathbf{x}))^2}{g(\mathbf{x})}d\mathbf{x} - \mu^2\right).$$

□

$$\mathrm{Var}(\hat{\mu}_{IS}) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{n}\frac{\Psi(\vec{x}^{(i)})f(\vec{x}^{(i)})}{g(\vec{x}^{(i)})}\right) = \frac{1}{N^2}\cdot N\,\mathrm{Var}\left(\frac{\Psi(\vec{x}^{(i)})f(\vec{x}^{(i)})}{g(\vec{x}^{(i)})}\right)$$

$$= \frac{1}{N}\left[\int\left(\frac{\Psi(\vec{x})f(\vec{x})}{g(\vec{x})}\right)^2 g(\vec{x})\,d\vec{x} - \mu^2\right]$$

Example 5.2.1. *We want to use importance sampling when approximating the integral*

$$I = \int_0^1 e^{x^2} dx$$

employing Monte Carlo, and considering $g(x) = e^x/(e-1)$. *First of all, since we need to sample from this distribution, e.g. by using the inversion method, we need to recover the CDF of* g. *Recall that the CDF is the integral function of the PDF, that is,* $G'(x) = g(x)$. *Hence,*

$$g(x) = \frac{e^x}{e-1}, \quad x \in (0,1) \qquad \Rightarrow \qquad G(x) = \frac{1}{e-1}\int_0^x e^t dt = \frac{e^x - 1}{e-1}$$

Then, using $X = G^{-1}(U) = \log(1 + (e-1)U)$ *with* $U \sim \mathcal{U}(0,1)$, *we have that*

$$I = (e-1)\int_0^1 e^{x^2-x}\frac{e^x}{e-1}dx \approx \frac{e-1}{N}\sum_{i=1}^{N} e^{(X^{(i)})^2 - X^{(i)}} \quad \text{with } X^{(i)} \overset{i.i.d.}{\sim} g.$$

In Matlab :

```
N = 10000;

% crude MC
U = rand(1,N); Y = exp(U.^2);
disp( [mean(Y) std(Y)/sqrt(N)])
    1.4618    0.0048

% importance sampling MC
e = exp(1); X = log(1+(e-1)*U);
T = (e-1)*exp(X.*(X-1));
disp( [mean(T) std(T)/sqrt(N)])
    1.4624    0.0011
```

term that enters in the confidence interval. The ultimate goal is to reduce the error bound of the estimate of $\mathbb{E}[Z]$, and this error bound is related to $\sqrt{Var(Y)}/\sqrt{N}$

sampling from g (inverse method)
we create the sample vector : $(e-1)e^{(x)^2 - x}$

Remark 5.2.1. *Importance sampling and acceptance-rejection sampling are quite similar ideas: both of them distort a sample from one distribution in order to sample from another, however importance sampling is usually easier to implement. Either one can be more efficient.*

The main difficulty in importance sampling is how to choose the importance sampling distribution. Indeed, let us consider the common case where $\Psi(\mathbf{x}) > 0$ for any $\mathbf{x}$. If the numerator of the integrand appearing in (5.2) goes to zero slower than $g(\mathbf{x})$ does, $\mathrm{Var}[\hat{\mu}_{IS}] \to \infty$. A poor choice of g may then seriously compromise the estimate and the confidence intervals. The following sections provide some guidance toward choosing a good importance sampling distribution.

Hands-On Problem 5.2 focuses on the MC method with importance sampling.

Optimal choice of the importance sampling distribution

the selection of a dominating PDF $g(\vec{x})$ has a huge impact on the accuracy of the estimation. In practice, g must fix the zones where Ψ has large values but f doesn't put importance (= weights).

Heuristically, the density $g(\mathbf{x})$ that minimizes the variance $\mathrm{Var}[\hat{\mu}_{IS}]$ is proportional to $|\Psi(\mathbf{x})|f(\mathbf{x})$. Indeed, it is possible to show that the optimal choice of g is the one that minimizes the integral $\int \frac{(\Psi(\mathbf{x})f(\mathbf{x}))^2}{g(\mathbf{x})}d\mathbf{x}$, under the condition that $g \geq 0$ and $\int g(\mathbf{x})d\mathbf{x} = 1$. The solution may be found using the calculus of variations. Introducing the Lagrangian function

$$\mathcal{L}(g,\lambda) = \int \frac{(\Psi f)^2}{g}d\mathbf{x} + \lambda\left(\int g d\mathbf{x} - 1\right)$$

lagrangian method applied to functions (the result is not a point that minimizes, but a function)

and taking variations, imposing that

$$\frac{\partial \mathcal{L}}{\partial g}(\delta g) = -\int \left(\frac{\Psi^2 f^2}{g^2} - \lambda\right)\delta g = 0 \quad \forall \delta g \qquad \Rightarrow \quad g^2 = \frac{\Psi^2 f^2}{\lambda}$$

directional derivative in the direction δg

λ is a number, but $\int g = 1$: $\lambda = \left(\int |\Psi(\vec{x})|\, f(\vec{x})\, d\vec{x}\right)$

so that the optimal g is given by

$$g^*(\mathbf{x}) = \frac{|\Psi(\mathbf{x})| f(\mathbf{x})}{\int |\Psi(\mathbf{x})| f(\mathbf{x}) d\mathbf{x}} = \frac{|\Psi(\mathbf{x})| f(\mathbf{x})}{\mathbb{E}_f[|\Psi(\mathbf{X})|]}.$$

With such optimal g^*, we obtain that

$$\mathrm{Var}\,[\hat{\mu}_{IS}] = \mathbb{E}_f[|\Psi(\mathbf{X})|]^2 - \mathbb{E}_f[\Psi(\mathbf{X})]^2.$$

Clearly, working with g^* is not practical as the normalizing constant $\mathbb{E}[|\Psi|]$ is, in general, as difficult to compute as the original quantity $\mu = \mathbb{E}[\Psi]$ (and we need it explicitly to correct the integral). In particular, if $\Psi(\mathbf{x}) \geq 0$ or $\Psi(\mathbf{x}) \leq 0$, the normalizing constant $\mathbb{E}_f[|\Psi(\mathbf{X})|] = \mu$ is precisely the quantity to compute and $\mathrm{Var}\,[\hat{\mu}_{IS}] = 0$.

Nevertheless, a good importance sampling density g should be "close" to the minimum variance density g^* – that is, g should resemble as much as possible to $|\Psi| f$ while still being easy to simulate and known with an explicit expression (with known constants).

● **Remark 5.2.2.** *One of the main considerations for choosing a good importance sampling PDF is that the estimator* (5.1) *should have finite variance. This is equivalent to the requirement that*

$$\mathbb{E}_g\left[\Psi^2(\mathbf{X})\frac{f^2(\mathbf{X})}{g^2(\mathbf{X})}\right] = \mathbb{E}_f\left[\Psi^2(\mathbf{X})\frac{f(\mathbf{X})}{g(\mathbf{X})}\right] < \infty.$$

This suggests that g should not have lighter tails than f, and that, preferably, the likelihood ratio, f/g, should be bounded.

we want g to have more weight on the tails

Hands-On 6 focuses on the estimation of probability of rare events by importance sampling.

Weighted importance sampling

in certain cases we might know $g(\cdot)$ up to some constant (the one that makes $g(\cdot)$ a distribution). We still can use the importance sampling method.

In certain cases, the PDF f and/or the dominating PDF g are known only up to a constant – however, $\mathbf{X} \sim g$ can still be generated e.g. by Acceptance-Rejection. Assume that $f = C_f \tilde{f}$ and $g = C_g \tilde{g}$, with

$$C_f = \left(\int \tilde{f}\right)^{-1}, \qquad C_g = \left(\int \tilde{g}\right)^{-1}.$$

A modified (self-normalized) version of the importance sampling estimator, which does not require the explicit knowledge of the normalizing constants C_f, C_g, however requiring the stronger condition $g(\mathbf{x}) = 0 \Rightarrow f(\mathbf{x}) = 0$, is:

$$\hat{\mu}^w_{IS} = \frac{\sum_{i=1}^N \Psi(\mathbf{X}^{(i)}) w(\mathbf{X}^{(i)})}{\sum_{i=1}^N w(\mathbf{X}^{(i)})} \qquad \text{with} \quad w(\mathbf{X}) = \frac{\tilde{f}(\mathbf{X})}{\tilde{g}(\mathbf{X})} \text{ and } \mathbf{X}^{(i)} \overset{i.i.d.}{\sim} g$$

and is often referred to as *weighted importance sampling*. Indeed, calling

$$\tilde{w}_i = \frac{w(\mathbf{X}^{(i)})}{\sum_{i=1}^N w(\mathbf{X}^{(i)})},$$

the estimator $\hat{\mu}^w_{IS}$ can be written as a weighted average

$$\hat{\mu}^w_{IS} = \sum_{i=1}^N \tilde{w}_i \Psi(\mathbf{X}^{(i)}).$$

To see that $\hat{\mu}_{IS}^{w}$ is a consistent estimator, observe that

$$\frac{1}{N}\sum_{i=1}^{N} w(\mathbf{X}^{(i)}) \overset{a.s.}{\to} \int \frac{\tilde{f}(\mathbf{x})}{\tilde{g}(\mathbf{x})} g(\mathbf{x}) d\mathbf{x} = C_g/C_f$$

by the Strong Law of Large Numbers (SLLN), and

$$\frac{1}{N}\sum_{i=1}^{N} \Psi(\mathbf{X}^{(i)}) w(\mathbf{X}^{(i)}) \overset{a.s.}{\to} \int \Psi(\mathbf{x}) \frac{\tilde{f}(\mathbf{x})}{\tilde{g}(\mathbf{x})} g(\mathbf{x}) d\mathbf{x} = \frac{C_g}{C_f}\mu$$

again by the SLLN. This estimator is biased, although the bias is usually small.

5.3 Control Variates

Control variates exploit information about the errors in estimates of known quantities to reduce the error of an estimate of an unknown quantity.

We consider again the goal of computing the expectation $\mu = \mathbb{E}[Z]$ of a random variable Z, output of a stochastic model. The idea of *control variates* is to find a variable Y that has a *known mean* and is strongly correlated with the variable Z. Then, the modified variable

$$\tilde{Z}_\alpha = Z + \alpha(Y - \mathbb{E}[Y]), \qquad \alpha \in \mathbb{R}$$

satisfies

$$\mathbb{E}\left[\tilde{Z}_\alpha\right] = \mathbb{E}[Z] = \mu$$

and

$$\mathrm{Var}\left[\tilde{Z}_\alpha\right] = \mathrm{Var}[Z] + \alpha^2 \mathrm{Var}[Y] + 2\alpha Cov(Z,Y).$$

The latter expression is a quadratic function of α and is minimized for

$$\alpha_{opt} = -\frac{Cov(Z,Y)}{\mathrm{Var}[Y]}.$$

also α_{opt} is a random object, hence it needs to be estimated and it comes with a variance

With such optimal choice, one has

$$\mathrm{Var}\left[\tilde{Z}_{\alpha_{opt}}\right] = \mathrm{Var}[Z] - \frac{Cov(Z,Y)^2}{\mathrm{Var}[Y]} = \mathrm{Var}[Z]\,(1 - Corr(Z,Y)^2)$$

which is always smaller than $\mathrm{Var}[Z]$. The amount of variance reduction increases as Z and Y are more correlated

$$Corr(Z,Y)^2 = \frac{Cov(Z,Y)^2}{\mathrm{Var}[Z]\,\mathrm{Var}[Y]} \to 1.$$

It is clear that the ideal control variate is $Y = \gamma Z$, $\gamma \in \mathbb{R}$ for which $Var[\tilde{Z}_{\alpha_{opt}}] = 0$. However, $\mathbb{E}[Y] = \gamma \mathbb{E}[Z]$ is not known in this case, and such a control variate is not a viable option. The control variate Y should be a reasonable approximation of Z, of which, however, we can compute exactly its expectation, or more generally, a random variable highly informative on Z (hence highly correlated with Z). In practice, the optimal α is not known, but can be estimated from a pilot run. Recall that $\mathbb{E}[Y]$ is known.

since $\mathbb{E}[Z]$ is what we want to estimate

The estimator $\hat{\mu}_{CV}$ is unbiased with variance (proof omitted)

$$\mathrm{Var}[\hat{\mu}_{CV}] = \mathbb{E}\left[(\hat{\mu}_{CV} - \mu)^2\right] = \frac{1}{N}\left(\mathrm{Var}\left[\tilde{Z}_{\alpha_{opt}}\right] + \mathrm{Var}[\hat{\alpha}_{opt}]\,\sigma_Y^2\right)$$

where

$$\mathrm{Var}[\hat{\alpha}_{opt}] = O\left(1/\bar{N}\right)$$

Algorithm 14 Control variates

1: Generate $\bar{N}$ i.i.d. replicas $(Z^{(i)}, Y^{(i)})$, $i = 1, \dots, \bar{N}$ of (Z, Y)
2: Estimate

$$\hat{\alpha}_{opt} = -\frac{\hat{\sigma}^2_{ZY}}{\sigma^2_Y} \text{ if } \sigma^2_Y \text{ known, or } \hat{\alpha}_{opt} = -\frac{\hat{\sigma}^2_{ZY}}{\hat{\sigma}^2_Y}$$

$$\text{with } \hat{\sigma}^2_{ZY} = \frac{1}{\bar{N}-1}\sum_{i=1}^{\bar{N}}(Z^{(i)} - \hat{\mu}_Z)(Y^{(i)} - \mathbb{E}[Y]), \qquad \hat{\mu}_Z = \frac{1}{\bar{N}}\sum_{i=1}^{\bar{N}} Z^{(i)};$$

3: Run crude Monte Carlo on $\tilde{Z}_{\alpha_{opt}} = Z + \hat{\alpha}_{opt}(Y - \mathbb{E}[Y])$, that is
4: – generate N i.i.d. replicas $(Z^{(i)}, Y^{(i)})$, $i = 1, \dots, N$ of (Z, Y);
5: – compute

$$\hat{\mu}_{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(Z^{(i)} + \hat{\alpha}_{opt}(Y^{(i)} - \mathbb{E}[Y])\right).$$

and $\text{Var}\left[\tilde{Z}_{\alpha_{opt}}\right]$ can be estimated by the estimator (unbiased if σ^2_Y is known)

$$\hat{\sigma}^2(\tilde{Z}_{\alpha_{opt}}) = \hat{\sigma}^2_Z - \frac{\hat{\sigma}^2_{ZY}}{\sigma^2_Y}.$$

Moreover,

$$\sqrt{N}\frac{\hat{\mu}_{CV} - \mu}{\hat{\sigma}^2(\tilde{Z}_{\alpha_{opt}})} \rightharpoonup N(0,1) \qquad \text{as } N, \bar{N} \to \infty,$$

from which asymptotic confidence intervals can be obtained.

Alternative to the previous algorithm. which uses a pilot run to estimate α_{opt}, one may consider a *one-shot* strategy:

Algorithm 15 Control variates (one shot)

1: Generate N i.i.d. replicas $(Z^{(i)}, Y^{(i)})$, $i = 1, \dots, N$ of (Z, Y)
2: Estimate

$$\hat{\alpha}_{opt} = -\frac{\hat{\sigma}^2_{ZY}}{\sigma^2_Y} \text{ if } \sigma^2_Y \text{ known, or } \hat{\alpha}_{opt} = -\frac{\hat{\sigma}^2_{ZY}}{\hat{\sigma}^2_Y}$$

$$\text{with } \hat{\sigma}^2_{ZY} = \frac{1}{\bar{N}-1}\sum_{i=1}^{\bar{N}}(Z^{(i)} - \hat{\mu}_Z)(Y^{(i)} - \mathbb{E}[Y]), \qquad \hat{\mu}_Z = \frac{1}{\bar{N}}\sum_{i=1}^{\bar{N}} Z^{(i)};$$

3: Estimate

$$\hat{\mu}_{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(Z^{(i)} + \hat{\alpha}_{opt}(Y^{(i)} - \mathbb{E}[Y])\right).$$

This estimator is biased, in general; however, a CLT result still holds and

$$\sqrt{N}\frac{\hat{\mu}_{CV} - \mu}{\hat{\sigma}^2(\tilde{Z}_{\alpha_{opt}})} \rightharpoonup N(0,1) \qquad \text{as } N \to \infty,$$

from which asymptotic confidence intervals can be obtained.

in the previous case we pay the unbiasedness computationally (since we have to generate $\bar{N}$ more). In this case we are breaking the ll of $\hat{\alpha}_{opt}$ and the other objects, hence: $\mathbb{E}[\hat{\mu}_{CV}] \neq \mu$

● **Example 5.3.1.** *We want to use a control variate when approximating the (trivial :-)) integral*

$$\mu = \int_0^1 e^x dx = \mathbb{E}[Z], \qquad \textit{with } Z = e^U, \ U \sim \mathcal{U}(0,1).$$

using Monte Carlo, considering $Y = U \sim U(0,1)$. *Then, we have* $\mathbb{E}[Y] = 1/2$ *and* $\mathrm{Var}[Y] = 1/12$. *Moreover,*

$$\mathrm{Var}[Z] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2 = \int_0^1 e^{2x}dx - \left(\int_0^1 e^x dx\right)^2 = \frac{1}{2}(e^2-1) - (e-1)^2 \approx 0.242;$$

recalling that $Cov(Z,Y) = \mathbb{E}[ZY] - \mathbb{E}[Z]\mathbb{E}[Y]$,

$$Cov(Z,Y) = Cov(e^U, U) = \int_0^1 xe^x dx - \int_0^1 x dx \int_0^1 e^x dx = 1 - (e-1)/2 \approx 0.1409.$$

Finally,

$$\mathrm{Var}\left[\tilde{Z}_{\alpha_{opt}}\right] = \mathrm{Var}[Z] - \frac{Cov(Z,Y)^2}{\mathrm{Var}[Y]} = 0.242 - \frac{0.141086^2}{1/12} \approx 0.0039.$$

we can do it only because we know how to solve analytically: $\int_0^1 e^x dx = \mathbb{E}[Z]$, $\int_0^1 e^{2x} dx = \mathbb{E}[Z^2]$. this isn't usually the case.

In Matlab*:*

crude MC

```
N = 1000; U = rand(1,N); X = exp(U);
disp([mean(X) std(X) std(X)/sqrt(N)])
   1.7478    0.5062    0.0160
```

control variates (one-shot)

```
Y = U; muY = 1/2;
Xb = mean(X); Yb = mean(Y);
cs = -sum( (X-Xb).*(Y-Yb) )/sum( (Y-Yb).^2 );
Z = X + cs*( Y - muY );
disp([mean(Z) std(Z) std(Z)/sqrt(N)])
   1.7212    0.0638    0.0020
```

$= \hat{\alpha}_{opt} = -\frac{\hat{\sigma}^2_{ZY}}{\hat{\sigma}^2_Y}$ (where $Z = X$)

5.3.1 Multiple control variates

The previous technique can be generalized to the case in which multiple control variates $Y_1, \dots, Y_p$ are used. We define the modified variable

$$\tilde{Z}_\alpha = Z + \sum_{j=1}^p \alpha_j (Y_j - \mathbb{E}[Y_j]) = Z + \boldsymbol{\alpha} \cdot (\mathbf{Y} - \mathbb{E}[\mathbf{Y}])$$

with

$$\mathbf{Y} = (Y_1, \dots, Y_p) \qquad \text{and} \qquad \boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p).$$

Then

$$\mathrm{Var}\left[\tilde{Z}_\alpha\right] = \mathbb{E}[Z - \mu + \boldsymbol{\alpha} \cdot (\mathbf{Y} - \mathbb{E}[\mathbf{Y}])] = \mathrm{Var}[Z] + 2Cov(Z, \mathbf{Y}) \cdot \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top Cov(\mathbf{Y}, \mathbf{Y})\boldsymbol{\alpha}$$

where

$$Cov(Z, \mathbf{Y}) = \{Cov(Z, Y_i)\}_{i=1}^p \in \mathbb{R}^p, \qquad Cov(\mathbf{Y}, \mathbf{Y}) = \{Cov(Y_i, Y_j)\}_{i,j=1}^p \in \mathbb{R}^{p \times p}.$$

Again, $\mathrm{Var}\left[\tilde{Z}_\alpha\right]$ is a quadratic function in $\boldsymbol{\alpha}$ and is minimized by

$$\boldsymbol{\alpha}_{opt} = -Cov(\mathbf{Y}, \mathbf{Y})^{-1} Cov(Z, \mathbf{Y}).$$

As we will see, it is also possible to apply the Control Variate concept in a hierarchical fashion: each successive estimator of a difference improves the estimate of its predecessor. This technique is known as multi-level Monte Carlo integration.

Algorithm 16 Multiple control variates (one shot)

1: Generate N i.i.d. replicas $(Z^{(i)}, Y_1^{(i)}, \ldots, Y_p^{(i)})$, $i = 1, \ldots, N$ of $(Z, \mathbf{Y})$
2: Estimate

$$(\hat{\sigma}^2_{Z\mathbf{Y}})_j = \frac{1}{N-1}\sum_{i=1}^{N}(Z^{(i)} - \hat{\mu}_Z)(Y_j^{(i)} - \mathbb{E}[Y_j]), \quad (\hat{\sigma}^2_{\mathbf{YY}})_{jk} = \frac{1}{N}\sum_{i=1}^{N}(Y_j^{(i)} - \mathbb{E}[Y_j])(Y_k^{(i)} - \mathbb{E}[Y_k])$$

3: Compute

$$\hat{\boldsymbol{\alpha}}_{opt} = -(\hat{\sigma}^2_{\mathbf{YY}})^{-1}\hat{\sigma}^2_{Z\mathbf{Y}};$$

4: Estimate

$$\hat{\mu}_{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(Z^{(i)} + \hat{\boldsymbol{\alpha}}_{opt}\cdot(\mathbf{Y}^{(i)} - \mathbb{E}[\mathbf{Y}])\right).$$

5.4 Stratification (or stratified sampling)

this is a sort of composed Monte Carlo. However, it's all fun and games until the dimension grows. Then, stratified sampling becomes unfeasible. → curse of dimensionality

Consider again the goal of computing $\mu = \mathbb{E}[Z]$ where Z is the output of a stochastic model. We assume here that $Z = \Psi(X_1, \ldots, X_d) = \Psi(\mathbf{X})$ where $\mathbf{X} \in \mathbb{R}^d$ is a random vector with PDF $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}_+$ so that

$$\mu = \int_\Omega \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}.$$

The idea of *stratification* is to divide the sample space Ω into S nonoverlapping regions $\Omega_1, \ldots, \Omega_s$ called *strata* such that — we introduce a partition of the domain and we sample on each one separately (domain decomposit.)

$$P(X \in \Omega_j) = \int_\Omega \mathbf{1}_{\Omega_j}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = p_j$$

probability of falling into each strata

is known and

$$\sum_{j=1}^{s} p_j = 1.$$

Assume now that we can generate X conditioned on $X \in \Omega_j$. Let X_j having density $X_j = X|_{\Omega_j}$

$$f_j(\mathbf{x}) = \frac{1}{p_j} f(\mathbf{x}) \mathbf{1}_{\mathbf{X}\in\Omega_j} \qquad \text{(conditional density of } \mathbf{X} \text{ given } \mathbf{X} \in \Omega_j).$$

Clearly,

$$\mu = \mathbb{E}[Z] = \sum_{j=1}^{s} \mathbb{E}[Z|\mathbf{X} \in \Omega_j]\, P(\mathbf{X} \in \Omega_j) = \sum_{j=1}^{s} p_j \mathbb{E}[\Psi(X_j)].$$

Then, the idea is to sample independently each $Z_j = \Psi(X_j)$ – indeed, stratified sampling is a variance reduction technique closely related to the composition method.

Properties of $\hat{\mu}_{str}$:

- $\hat{\mu}_{str}$ is unbiased. Indeed,

$$\mathbb{E}[\hat{\mu}_{str}] = \sum_{j=1}^{s} p_j \mathbb{E}[\hat{\mu}_j] = \sum_{j=1}^{s} p_j \mathbb{E}[Z_j] = \mathbb{E}[Z].$$

- The variance of $\hat{\mu}_{str}$ is given by

$$\mathrm{Var}[\hat{\mu}_{str}] = \sum_{j=1}^{s} p_j^2 \mathrm{Var}[\hat{\mu}_j] = \sum_{j=1}^{s} p_j^2 \frac{\mathrm{Var}[Z_j]}{N_j}.$$

Algorithm 17 Stratification

1: **for** $j = 1, \dots, s$ **do**
2: Generate N_j i.i.d. replicas $Z_j^{(i)}$, $i = 1, \dots, N_j$ of Z_j
3: Compute

on each stratum we have a crude MC

$$\hat{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} Z_j^{(i)}$$ — mean of each zone j

4: **end for**
5: Compute

$$\hat{\mu}_{str} = \sum_{j=1}^{s} p_j \hat{\mu}_j$$ — mean of the means of each zone (we average the means w.r.t. the probability to fall in each zone (p_j))

- Let $N = \sum_{j=1}^{s} N_j$ and choose $N_j = \varphi_j(N)$ such that $\lim_{N\to\infty} N/N_j < +\infty$. Then

$$\lim_{N\to\infty} N \operatorname{Var}[\hat{\mu}_{str}] < +\infty$$

and it is possible to prove that

$$\frac{\hat{\mu}_{str} - \hat{\mu}}{\sqrt{\operatorname{Var}[\hat{\mu}_{str}]}} \rightharpoonup N(0,1) \qquad \text{as } N \to \infty.$$

- Moreover, $\operatorname{Var}[\hat{\mu}_{str}]$ can be estimated by

$$\hat{\sigma}^2 = \sum_{j=1}^{s} p_j^2 \frac{\hat{\sigma}_j^2}{N_j}, \qquad \hat{\sigma}_j^2 = \frac{1}{N_j - 1} \sum_{i=1}^{N_j} (Z_j^{(i)} - \hat{\mu}_j)^2$$

and a $(1-\alpha)$ asymptotic confidence interval is given by

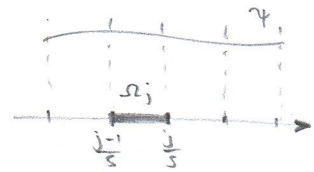
$$I_\alpha = \left[\hat{\mu}_{str} - z_{1-\alpha/2}\hat{\sigma}, \hat{\mu}_{str} + z_{1-\alpha/2}\hat{\sigma}\right].$$

- **Example 5.4.1.** *Let $X \sim \mathcal{U}(0,1)$ and $Z = \Psi(X)$. Then*

$$\mu = \mathbb{E}[Z] = \int_0^1 \Psi(x)dx$$

which we could stratify taking

$$\Omega_j = \left(\frac{j-1}{s}, \frac{j}{s}\right), \qquad j = 1, \dots, s.$$



here we consider all the probabilities equal

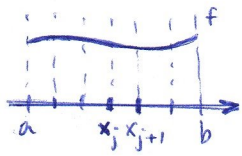
Then

$$\mu = \mathbb{E}[Z] = \sum_{j=1}^{s} \int_{\frac{j-1}{s}}^{\frac{j}{s}} \Psi(x)dx = \sum_{j=1}^{s} \frac{1}{s} \mathbb{E}[Z_j]$$ $Z_j = Z|_{\Omega_j}$

with $Z_j = \Psi(X_j)$ and $X_j \sim \mathcal{U}\left(\frac{j-1}{s}, \frac{j}{s}\right)$. Then

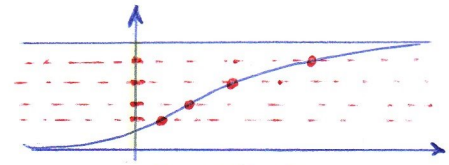
$$\hat{\mu}_{str} = \sum_{j=1}^{s} \frac{1}{s} \cdot \frac{1}{N_j} \sum_{i=1}^{N_j} \Psi(X_j^{(i)}) \qquad \text{with} \quad X_j^{(i)} \overset{i.i.d.}{\sim} \mathcal{U}\left(\frac{j-1}{s}, \frac{j}{s}\right).$$

In the case of $X \sim \mathcal{U}(0,1)$ we thus end up with a strategy that is indeed very close to composite quadrature formulas from numerical analysis.



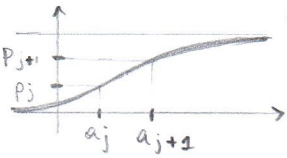
$$\int_a^b f(x)\,dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)\,dx$$

What if we want to stratify something **not** uniform? Taking any CDF we can work on the image and stratify the image. Then we apply the inverse method.



- stratify $\mathcal{U}([0,1])$: —
- apply the inverse transform method

with this we report the strata on the original domain, e.g. *



- **Example 5.4.2.** (Stratifying nonuniform distributions). *Let F a CDF on $\mathbb{R}$ and let F^{-1} denote its inverse as defined in Section 2.1. Given probabilities $p_1, \ldots, p_s$, define $a_0 = -\infty$,*

$$a_1 = F^{-1}(p_1), \quad a_2 = F^{-1}(p_1 + p_2), \quad \ldots, \quad a_s = F^{-1}(p_1 + p_2 + \ldots + p_s) = F^{-1}(1).$$

Define strata $\Omega_1 = (a_0, a_1]$, $\Omega_2 = (a_1, a_2], \ldots, \Omega_s = (a_{s-1}, a_s]$ (or $\Omega_s = (a_{s-1}, a_s)$ if $a_s = \infty$). By construction, each stratum Ω_i has probability p_i under F; indeed, if X has distribution F, then

$$P(X \in \Omega_i) = F(a_i) - F(a_{i-1}) = p_i.$$

Thus, defining strata for F with specified probabilities is straightforward, provided one can find the quantiles a_i. Figure 5.1 (left) displays ten equiprobable ($p_i = 1/s$) strata for the standard normal distribution. To use the sets $\Omega_1, \ldots, \Omega_s$ for stratified sampling, we need to be able to generate samples of X conditional on $X \in \Omega_i$ – this is indeed an application of the inverse transform method *of Section 2.1! If $U \sim \mathcal{U}(0,1)$, then*

$$V = p_{i-1} + U(p_i - p_{i-1}) \sim \mathcal{U}(p_{i-1}, p_i)$$

and then $F^{-1}(V)$ has the distribution of X conditional on $X \in \Omega_i$.

Figure 5.1 (center-right) illustrates the difference between stratified and random sampling from the standard normal distribution. The center panel is a histogram of 500 observations, five from each of 100 equiprobable strata; the right panel is a histogram of 500 independent draws from the normal distribution. Stratification clearly produces a better approximation to the underlying distribution.

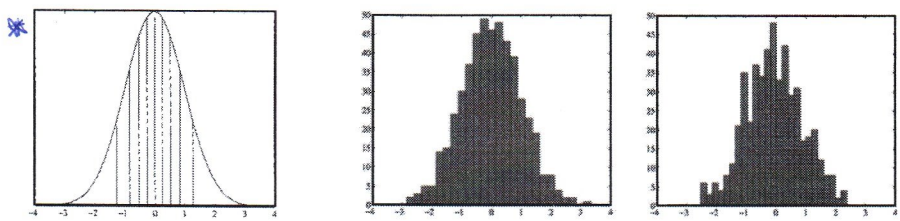


Figure 5.1: Stratifying a standard normal distribution. Center: stratified sampling; Right: random sampling.

In conclusion, stratification ensures that each of the strata contains a fixed number of evaluations. It remains the question on how to choose N_j in each stratum and quantify the amount of variance reduction.

5.4.1 Proportional allocation

If N is the total number of points, proportional allocation simply chooses

$$N_j = Np_j.$$

With this choice, we have

$$\operatorname{Var}[\hat{\mu}_{str}] = \sum_{j=1}^{s} p_j^2 \frac{\operatorname{Var}[Z_j]}{N_j} = \frac{1}{N} \sum_{j=1}^{s} p_j \operatorname{Var}[Z_j] = \frac{1}{N} \mathbb{E}\left[\operatorname{Var}[Z \mid \mathbf{X} \in \Omega_j]\right].$$

Setting discrete random variables $J \in \{1, \ldots, s\}$,

$$J = j \quad \Leftrightarrow \quad \{\mathbf{X} \in \Omega_j\}$$

introduced only for notation

we have $Z_j = Z \mid J = j$ and

$$\begin{aligned}\mathrm{Var}\left[\hat{\mu}_{str}\right] &= \frac{1}{N}\sum_{j=1}^{s} p_j \,\mathrm{Var}\left[Z \mid J=j\right] = \frac{1}{N}\mathbb{E}\left[\mathrm{Var}\left[Z \mid J\right]\right] \\ &= \frac{1}{N}\left(\mathrm{Var}\left[Z\right] - \mathrm{Var}\left[\mathbb{E}\left[Z \mid X\right]\right]\right) \leq \frac{\mathrm{Var}\left[Z\right]}{N} = \mathrm{Var}\left[\hat{\mu}_{CMC}\right].\end{aligned}$$

having exploited the

Variance decomposition formula.
If Z and X are random variables on the same probability space, and the variance of X is finite, then

$$\mathrm{Var}\left[Z\right] = \mathbb{E}\left[\mathrm{Var}\left[Z \mid X\right]\right] + \mathrm{Var}\left[\mathbb{E}\left[Z \mid X\right]\right].$$

Hence, $\mathrm{Var}\left[\mathbb{E}\left[Z \mid X\right]\right] \leq \mathrm{Var}\left[Z\right]$.

Hence, proportional allocation always leads to variance reduction. The amount of variance reduction is given by

$$\gamma = \frac{\mathbb{E}\left[\mathrm{Var}\left[Z \mid J\right]\right]}{\mathrm{Var}\left[Z\right]}.$$

Figure 5.2 shows a random sample from the unit square along with 3 alternative stratified samplings. The unit square $[0,1)^2$ is very easily partitioned into box shaped strata like those shown, and it is also easy to sample uniformly in such strata.

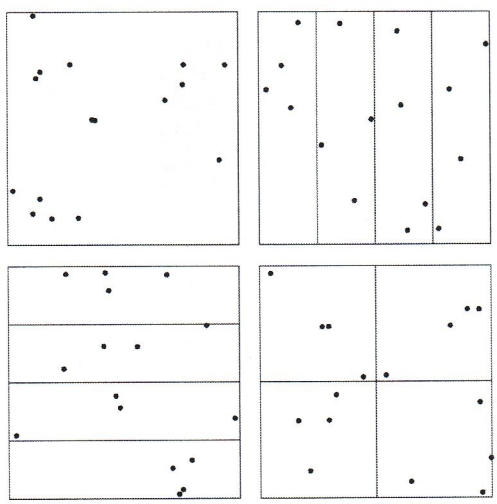


Figure 5.2: The upper left figure is a simple random sample of 16 points in $[0,1)^2$. The other figures show stratified samples with 4 points from each of 4 strata.

Remark 5.4.1. *Another variance reduction technique, based on the variance decomposition formula, goes under the name of* conditional Monte Carlo. *If the goal is to estimate $\mu = \mathbb{E}\left[Z\right]$, and we can find a random variable (or vector), $Y \sim g$, such that the conditional expectation $\mathbb{E}\left[Z \mid Y = y\right]$ can be computed analytically, by the* tower property

$$\mu = \mathbb{E}\left[Z\right] = \mathbb{E}\left[\mathbb{E}\left[Z \mid Y\right]\right],$$

so that $\mathbb{E}[Z \mid Y]$ *is an unbiased estimator of* μ. *Moreover, the variance of* $\mathbb{E}[Z \mid Y]$ *is always smaller than or equal to the variance of* Z. *The conditional Monte Carlo idea is sometimes referred to as* Rao-Blackwellization. *The algorithm is straightforward:*

Algorithm 18 Conditional Monte Carlo

1: Generate N i.i.d. replicas $Y^{(1)}, \dots, Y^{(N)} \sim g$;
2: Calculate $\mathbb{E}\left[Z \mid Y^{(j)}\right]$, $j = 1, \dots, N$, analytically;
3: Compute

$$\hat{\mu}_C = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[Z \mid Y^{(j)}\right].$$

5.4.2 Optimal allocation

Instead of doing a proportional allocation, one may try to find the best choice of N_j that minimizes $\mathrm{Var}[\hat{\mu}_{str}]$:

$$\{N_j^*\} = \arg \min_{(N_1,\dots,N_s)} \sum_{j=1}^{s} p_j^2 \frac{\mathrm{Var}[Z_j]}{N_j} \qquad \text{s.t.} \quad \sum_{j=1}^{s} N_j = N.$$

constrained optimization problem in dimension s

the budget is N: we want to draw N samples

Introducing a Lagrangian function

$$\mathcal{L}(\mathbf{N}, \lambda) = \sum_{j=1}^{s} p_j^2 \frac{\mathrm{Var}[Z_j]}{N_j} + \lambda \left(\sum_{j=1}^{s} N_j - N \right)$$

← constrained optimization: cost function + constraint which is enforced by a lagrange multiplier

we have

$$\frac{\partial \mathcal{L}}{\partial N_j} = -p_j^2 \frac{\mathrm{Var}[Z_j]}{N_j^2} + \lambda = 0 \qquad \Rightarrow \qquad N_j = p_j \frac{\sqrt{\mathrm{Var}[Z_j]}}{\sqrt{\lambda}}$$

and (taking into account the constraint $\sum_{j=1}^{s} N_j = N$)

$$\sqrt{\lambda} = \frac{1}{N} \sum_{j=1}^{s} p_j \sqrt{\mathrm{Var}[Z_j]}$$

which leads to the optimal choice

$$N_j^* = \frac{N\, p_j \sigma_j}{\sum_{k=1}^{s} p_k \sigma_k}, \qquad \sigma_j = \sqrt{\mathrm{Var}[Z_j]}$$

and optimal variance

$$\mathrm{Var}[\hat{\mu}_{str}^*] = \frac{1}{N} \left(\sum_{j=1}^{s} p_j \sigma_j \right)^2.$$

Indeed,

$$\mathrm{Var}[\hat{\mu}_{str}^*] = \sum_{j=1}^{s} p_j^2 \frac{\sigma_j^2}{N_j^*} = \sum_{j=1}^{s} p_j^2 \frac{\sigma_j^2}{\frac{N p_j \sigma_j}{\sum_{k=1}^{s} p_k \sigma_k}} = \frac{1}{N} \left(\sum_{j=1}^{s} p_j \sigma_j \right) \left(\sum_{k=1}^{s} p_k \sigma_k \right) = \frac{1}{N} \left(\sum_{j=1}^{s} p_j \sigma_j \right)^2.$$

Since this variance is smaller than that with proportional allocation, stratification with optimal allocation will always lead to variance reduction. In practice, the σ_j are not known and can be estimated from a pilot run.

Remark 5.4.2. *As expected,* $\mathrm{Var}[\hat{\mu}_{str}^*]$ *is bounded above by the variance of the proportional allocation scheme, as*

$$\frac{1}{N} \left(\sum_{j=1}^{s} p_j \sigma_j \right)^2 \leq \sum_{j=1}^{s} p_j \frac{\sigma_j^2}{N_j}$$

by Jensen's inequality. Note that, in general, $p_i \sigma_i N$ *will not be an integer. However, rounding these values will not make a significant different if* N *is sufficiently large.*

Algorithm 19 Stratification with optimal allocation

1: **for** $j = 1, \dots, s$ **do**
2: Generate $\bar{N}_j$ i.i.d. replicas $Z_j^{(i)}$, $i = 1, \dots, \bar{N}_j$ of Z_j
3: Estimate

$$\hat{\sigma}_j^2 = \frac{1}{\bar{N}_j - 1} \sum_{i=1}^{\bar{N}_j} (Z_j^{(i)} - \hat{\mu}_j)^2$$

4: **end for**
5: Choose

$$N = \left(\frac{z_{1-\alpha/2} \sum_{j=1}^{s} p_j \hat{\sigma}_j}{tol} \right)^2 \qquad \text{to ensure that } |I_\alpha| < 2tol$$

6: **for** $j = 1, \dots, s$ **do**
7: Generate $N_j^* = \frac{N p_j \hat{\sigma}_j}{\sum_{k=1}^{s} p_k \hat{\sigma}_k}$ i.i.d. replicas $Z_j^{(i)}$ of Z_j
8: Compute

$$\hat{\mu}_j = \frac{1}{N_j^*} \sum_{i=1}^{N_j^*} Z_j^{(i)}$$

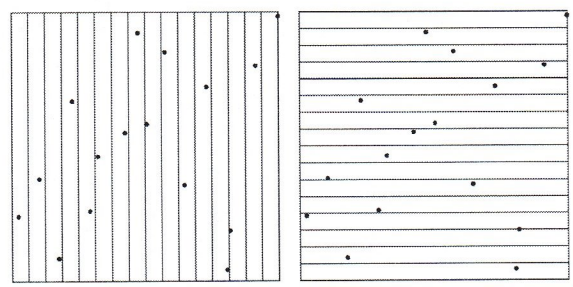
9: **end for**
10: Compute

$$\hat{\mu}_{str}^* = \sum_{j=1}^{s} p_j \hat{\mu}_j$$

5.5 Latin Hypercube Sampling

Here, however, we lose independence! But it avoids the curse of dimensionality

Suppose we can afford to sample 16 points in $[0,1)^2$. Sampling one point from each of 16 vertical strata would be a good strategy if the function f depended primarily on the horizontal coordinate. Conversely if the vertical coordinate is the more important one, then it would be better to take one point from each of 16 horizontal strata. It is possible to stratify both ways with the same sample, in what is known as *Latin hypercube sampling*. Figure 5.3 shows a set of 16 points in the square, that are simultaneously stratified in each of 16 horizontal and vertical strata.



those samples are stratified to both components. We don't have a point on each of the 16×16 strata (that we would have if we would have stratified on both components). We have a point on each of the 16 strata for both X_1 and X_2.

Figure 5.3: The left plot shows 16 points, one in each of 16 vertical strata. The right plot shows the same 16 points. There is one in each of 16 horizontal strata. These points form what is called a Latin hypercube sample.

Consider a random variable $Z = \Psi(X_1, \dots, X_d)$ with $X_j \in \mathbb{R}$, $X_j \sim f_j$. One might want to stratify each variable X_j in s strata. However, this would lead to s^d strata which become unaffordable for large d.

A way to overcome this problem is offered by the Latin Hypercube Sampling (LHS). For simplicity of exposition, let us assume that $(X_1, \dots, X_d) \sim \mathcal{U}([0,1]^d)$.

The idea of LHS is to stratify each component X_j but not the whole sampling domain in $\Omega = [0,1]^d$. In particular, N (correlated) points $\mathbf{X}^{(i)}$, $i = 1, \dots, N$ are drawn in $[0,1]^d$ in such a way that each component is stratified with N strata and one point per stratum.

A latin hypercube sampling can be generated by the following

Algorithm 20 Latin Hypercube Sampling

1: Generate N i.i.d. points $\mathbf{U}^{(i)} \overset{i.i.d.}{\sim} \mathcal{U}((0,1)^d)$
2: Generate d independent permutations $\bar{u}_j$, $j = 1, \dots, d$ of $\{1, \dots, N\}$ and let $\mathbf{V}^{(i)} = (\bar{u}_1(i), \bar{u}_2(i), \dots, \bar{u}_d(i))$.
3: Set

$$\mathbf{X}^{(i)} = \frac{\mathbf{V}^{(i)} - 1 + \mathbf{U}^{(i)}}{N}. \qquad \in \mathbb{R}^d$$

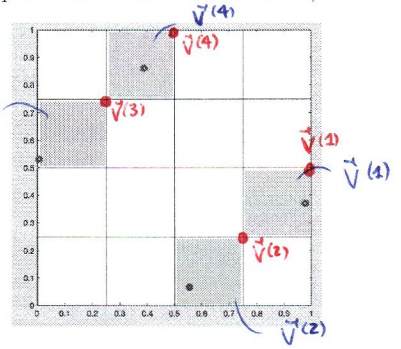
Then, the LHS estimator of $\mu = \mathbb{E}[Z]$ is simply

$$\hat{\mu}_{LHS} = \frac{1}{N} \sum_{i=1}^{N} \Psi(\mathbf{X}^{(i)}). \tag{5.3}$$

- **Remark 5.5.1.** Which is the rationale behind? *Each of the N i.i.d. points $\mathbf{U}^{(i)}$ is put in a subdomain, that is found according to the d components of $\mathbf{V}^{(i)}$. Take for simplicity $d = 2$, and partition the box $[0, N] \times [0, N]$ uniformly into $N \times N$ squared subdomains. For instance,*

$$\bar{u}_1 = \begin{pmatrix} 4 \\ 3 \\ 1 \\ 2 \end{pmatrix}, \quad \bar{u}_2 = \begin{pmatrix} 2 \\ 1 \\ 3 \\ 4 \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} \mathbf{V}^{(1)} &= (4,2) \\ \mathbf{V}^{(2)} &= (3,1) \\ \mathbf{V}^{(3)} &= (1,3) \\ \mathbf{V}^{(4)} &= (2,4) \end{aligned}$$

random permutations of numbers from 1 to N — each one selects a box



• points
• corresponding region

These couples identify the subdomains reported in the Figure above. Then, taking subtracting $\mathbf{V}^{(i)} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ *allows to select the* bottom left corner *of each subdomain; adding* $\mathbf{U}^{(i)}$ *gives the points highlighted in each subdomain. Finally, dividing by N rescales everything to $(0,1)^2$.*

$$\vec{X}^{(i)} = \frac{\vec{V}^{(i)} - 1 + \vec{U}^{(i)}}{N}$$

we select the top-right corner of each domain, we go to the bottom-left corner doing "-1", we add the uniform and then we rescale everything

Taking a permutation of N elements for each dimension ensures that there do not exist two samples on the same dimension and that all dimensions are correctly represented.

Proposition 5.5.1. *Let $\{\mathbf{X}^{(i)}, i = 1, \dots, N\}$ a LHS. Then:*

- $\mathbf{X}^{(i)} \sim \mathcal{U}((0,1)^d)$ *(not independent, though);* ✗
- $\mathbb{E}[\hat{\mu}_{LHS}] = \mathbb{E}[\Psi(\mathbf{X})]$. (still unbiased ✓)

Proof. $\mathbf{X}^{(i)} = \frac{\mathbf{V}^{(i)} - 1 + \mathbf{U}^{(i)}}{N}$ has independent components, hence it is sufficient to show that $\mathbf{X}_j^{(i)} \sim \mathcal{U}(0,1)$, $j = 1, \dots, d$. Now $\mathbf{V}_j^{(i)}$ is the i-th component of a random permutation of $\{1, \dots, N\}$,

hence

$$P(\mathbf{V}_j^{(i)} = k) = \frac{1}{N} \qquad \forall k = 1, \dots, N.$$

Moreover, the conditional distribution of $\mathbf{X}_j^{(i)}$ given $\mathbf{V}_j^{(i)} = k$ is

$$F_{\mathbf{X}_j^{(i)}|\mathbf{V}_j^{(i)}=k} = P(\mathbf{X}_j^{(i)} \le x \mid \mathbf{V}_j^{(i)} = k) = \begin{cases} 0 & x < \frac{k-1}{N} \\ NX - k + 1 & x \in \left[\frac{k-1}{N}, \frac{k}{N}\right] \\ 1 & x > \frac{k}{N}, \end{cases}$$

that is,

$$\mathbf{X}_j^{(i)} \le x \mid \mathbf{V}_j^{(i)} = k \sim \mathcal{U}\left(\frac{k-1}{N}, \frac{k}{N}\right).$$

Hence,

$$P(\mathbf{X}_j^{(i)} \le x) = \sum_{k=1}^{N} \frac{1}{N} P(\mathbf{X}_j^{(i)} \le x \mid \mathbf{V}_j^{(i)} = k) = x.$$

It then follows immediately that $\mathbb{E}[\hat{\mu}_{LHS}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \Psi(\mathbf{X}^{(i)})\right] = \mathbb{E}[\Psi(\mathbf{X})]$. □

Concerning the variance of the estimator $\hat{\mu}_{LHS}$, we mention the following two results.

Proposition 5.5.2. (Owen, 1997). *Let $Z = \Psi(\mathbf{X})$, $\mathbf{X} \sim \mathcal{U}([0,1]^d)$ with $\mu = \mathbb{E}[Z] < +\infty$ and $\sigma^2 = \mathrm{Var}[Z] < +\infty$. Then the LHS estimator $\hat{\mu}_{LHS}$ based on N points (5.3) satisfies*

$$\mathrm{Var}[\hat{\mu}_{LHS}] \le \frac{\sigma^2}{N-1}.$$

This result shows that asymptotically $\mathrm{Var}[\hat{\mu}_{LHS}]$ is not worse than $\mathrm{Var}[\hat{\mu}_{CMC}] = \sigma^2/N$. Moreover, LHS is very effective if the function $\Psi(\mathbf{X})$ has an additive structure, that is,

$$\Psi(\mathbf{X}) = \mu + \sum_{i=1}^{d} \Psi_i(X_i)$$

← one contribution on each component of $\vec{X}$

as the estimator $\hat{\mu}_{LHS}$ corresponds to a stratified estimator with N strata on each function Ψ_i. For a general $\Psi : \mathbb{R}^d \to \mathbb{R}$ let

$$\hat{\Psi}_j(x_j) = \int_{[0,1]^{d-1}} (\Psi(\mathbf{x}) - \mathbb{E}[\Psi(\mathbf{x})]) dx_1 \dots dx_{j-1} dx_{j+1} \dots dx_d$$

and

$$\Psi^{add}(\mathbf{X}) = \mathbb{E}[\Psi(\mathbf{X})] + \sum_{i=1}^{d} \hat{\Psi}_j(X_j).$$

if we define $\hat{\Psi}$ like that then we can somehow rewrite $\Psi(\vec{X})$ as an additive decomposition function, that we call $\Psi^{add}(\vec{X})$.

Then, it can be shown that

Proposition 5.5.3. (Owen, 1987). *For $Z = \Psi(\mathbf{X})$, $\mathbf{X} \sim \mathcal{U}([0,1]^d)$ with $\mu = \mathbb{E}[Z] < +\infty$ and $\sigma^2 = \mathrm{Var}[Z] < +\infty$, and $\hat{\mu}_{LHS}$ a LHS estimator for μ based on N points. Then, it holds*

$$\mathrm{Var}[\hat{\mu}_{LHS}] = \frac{\mathrm{Var}[\Psi - \Psi^{add}]}{N} + O\left(\frac{1}{N}\right).$$

in a LHS estimator (general) we get a variance reduction that is proportional to the gap of our model and how it is likely to be expressed in a sort of an additive way. (Here Ψ is **not** necessarily additive)

This result highlights the variance reduction achieved by the LHS estimator, compared to CMC. In practice, to estimate the variance of $\hat{\mu}_{LHS}$, one generates few independent replicas of $\hat{\mu}_{LHS}$.

In the case of additive structure we have that:
(case in which $\psi(\vec{x})$ has an additive structure)

and $\psi_0 \in \mathbb{R}$

$$\psi(\vec{x}) = \psi_0 + \sum_{j=1}^{d} \psi_j(x_j)$$

(one contribution on each component of $\vec{x}$)

where $\int_0^1 \psi_j(x)\,dx = 0 \quad \forall j$ ✓

(since we are taking points in $[0,1]^d$)

In general, if $\psi(\cdot)$ does **not** have an additive structure, we can try to write $\psi(\cdot)$ in an additive form but we'll obviously get something as remainder (residual)

$$\psi(\vec{x}) = \underbrace{\psi_0 + \sum_{j=1}^{d} \psi_j(x_j)}_{:= \psi^{add}(\vec{x})} + \underbrace{r(\vec{x})}_{=0 \text{ if } \psi(\vec{x}) \text{ has an additive structure}}$$

$$= \psi^{add}(\vec{x}) + r(\vec{x})$$

We re-name things as:

$$\psi^{add}(\vec{x}) = \mu + \sum_{j=1}^{d} \psi_j(x_j)$$

where:

$$\begin{cases} \mu = \int \psi(\vec{x})\,d\vec{x} \\ \psi_j(x_j) = \int_{[0,1]^{d-1}} (\psi(\vec{x}) - \mu)\; dx_1, \dots, dx_{j-1}, dx_{j+1}, \dots, dx_d \end{cases}$$

~~for these settings~~

$\psi^{add}(\vec{x})$ so defined is the best additive approximation of $\psi(\vec{x})$. Any other (additive) expansion of $\psi(\vec{x})$ will have a remainder bigger than the one we get with $\psi^{add}(\vec{x})$.

⟹ **Prop.** If $\psi: [0,1]^d \to \mathbb{R}$ s.t. $\int_{[0,1]^d} (\psi(\vec{x}))^2\,d\vec{x} < \infty$ and we consider an HLS design $(\vec{x}^{(1)}, \dots, \vec{x}^{(N)}, \hat{\mu}_{LHS} = \frac{1}{N}\sum_{j=1}^{N} \psi(\vec{x}^{(j)}))$

then:

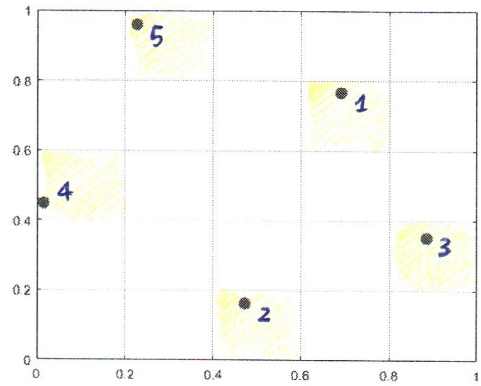
$$Var(\hat{\mu}_{HLS}) = \frac{Var(\psi(\vec{x}) - \psi^{add}(\vec{x}))}{N} + o\left(\frac{1}{N}\right)$$

little o, otherwise there is no variance reduction with respect to Monte Carlo ($1/N$)

```
X = lhsdesign(5,2)
```

```
X = 5×2
    0.6907    0.7660
    0.4727    0.1643
    0.8858    0.3513
    0.0146    0.4496
    0.2272    0.9602
```

```
plot(X(:,1), X(:,2), '.', 'MarkerSize', 30)
grid on
xticks(0:0.2:1)
yticks(0:0.2:1)
axis([0 1 0 1])
```



every column of X contains one random number in $[0;0.2]$, $[0.2;0.4]$, $[0.4;0.6]$, $[0.6;0.8]$, $[0.8;1]$

intervals = # points = $n = 5$

space dimension = $p = 2$

Algorithm 21 LHS estimator

1: Generate K independent LHS samples $\{\mathbf{X}^{(i,j)}, i = 1,\dots,N\}$, $j = 1,\dots,K$ of size N;
2: Compute

$$\hat{\hat{\mu}}_{LHS} = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{N}\sum_{i=1}^{N}\Psi(\mathbf{X}^{(i,j)}) = \frac{1}{KN}\sum_{i,j}\Psi(\mathbf{X}^{(i,j)})$$

3: Compute

$$\hat{\sigma}^2_{LHS} = \frac{1}{K-1}\sum_{j=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N}\Psi(\mathbf{X}^{(i,j)}) - \hat{\mu}_{LHS}\right)^2$$

4: Output $\hat{\hat{\mu}}_{LHS}$ and a confidence interval $\hat{\hat{\mu}}_{LHS} \pm z_{1-\frac{\alpha}{2}}\frac{\hat{\sigma}_{LHS}}{\sqrt{K}}$.

we generate K samples and then we perform MC on each sample (here we have ⫫ only among the K samples, we don't have ⫫ also among the N samples inside every one of the K. That's the reason why, to get a confidence interval we use $\sqrt{K}$ and not $\sqrt{N}$)

- **Remark 5.5.2.** *The* `Matlab` *command*

```
X = lhsdesign(n,p)
```

returns a Latin hypercube sample matrix of size n-by-p. For each column of X, the n values are randomly distributed with one from each interval $(0, 1/n), (1/n, 2/n), \dots, (1-1/n, 1)$, and randomly permuted.

Remark 5.5.3. *The following* `Matlab` *code realizes LHS sampling (in d dimensions) of independent standard Normal distributions providing N samples:*

```
data = rand(N,d);
for i = 1 : d
   index = randperm(N);
   prob = (index'-data(:,i)) / N
   data(:,i) = sqrt(2)*erfinv(2*prob-1);
end
```

here we're using the inverse method to get gaussian (if it was: data(:,i) = prob, then we would have p ⫫ U([0,1]) samples)

Remark 5.5.4. *In the examples we have seen so far, the Monte Carlo method (and its variants) has only been used to estimate expectations, and typically the first moment of the distribution is the focus of interest.*

However, the sample $\Psi(\mathbf{X}^{(1)}), \dots, \Psi(\mathbf{X}^{(N)})$ used to construct the estimator for an expectation of the form $\mathbb{E}[\Psi(\mathbf{X})]$ can be used to extract more information on the distribution of $\Psi(\mathbf{X})$ in addition to its mean.

In particular, the CDF of $\Psi(\mathbf{X})$ can be approximated by the empirical CDF

$$\hat{F}_N(z) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}_{Z^{(i)}\le z},$$

where $Z^{(i)} = \Psi(\mathbf{X}^{(i)})$, $i = 1,\dots,N$. The empirical CDF $\hat{F}_N$ is discontinuous, but continuous variants can be obtained by using interpolation. *Note that, for each z, $\hat{F}_N(z)$ is an unbiased estimator of $F(z) = P(\Psi(\mathbf{X}) \le z)$. Hence, by the strong law of large numbers, $\hat{F}_N$ converges in distribution to the CDF of $Z = \Psi(\mathbf{X})$ as $N \to \infty$. Once we have an approximation for the CDF $F(\cdot)$ of the variable $Z = \Psi(\mathbf{X})$ of interest, we can also get estimates for quantiles.*

Variance reduction ($\mu = \mathbb{E}[\psi(\vec{X})]$ to be estimated)

MC: $\hat{\mu}_{MC} = \frac{1}{N} \sum_{j=1}^{N} \psi(\vec{X}^{(j)})$: $|\hat{\mu}_{MC} - \mu| \overset{\text{asympt. } (N \to \infty)}{\leq} z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}$

we try to work on this with:
(Chapter 5.)
- antithetic variables
- importance sampling
- control variates
- stratified sampling
- Latin hypercube sampling

(Chapter 6.)
Is it possible to obtain something better than $O(N^{-1/2})$?
Monte Carlo is **not** able to do that.
(still, MC is $\perp\!\!\!\perp$ of the dimension d (since $O(N^{-1/2}) \perp\!\!\!\perp d$), so it is something good)
$\Longrightarrow$ QUASI MONTE CARLO methods
however, we need to give up on randomness.