Stochastic Simulations

Autumn Semester 2025

Prof. Fabio Nobile *Assistant: Eliott Van Dieren*

Project 3

submission deadline: 11 January 2026

Variance reduction in Monte Carlo integration via function approximation

1 Introduction and background

The goal of this mini-project is to reduce the variance of a Monte Carlo estimator for $\mu = \mathbb{E}[f]$ by first creating a least-squares polynomial approximation of the integrand f, and then using it as a control variate.

1.1 Monte Carlo with least squares (MCLS)

Let $\Omega = [0,1]^d$, $d \geq 1$ and $f : \Omega \mapsto \mathbb{R}$ integrable with respect to the Lebesgue measure. Suppose we are interested in approximating $I = \int_\Omega f(x)\mathrm{d}x$ using a Monte Carlo quadrature, $I_{MC} \approx \frac{1}{M}\sum_{i=1}^M f(x^{(i)})$, $x^{(i)} \overset{\text{iid}}{\sim} \mathcal{U}(\Omega)$. Recall that the main idea behind control variate techniques is to (i) introduce an auxiliary function $\phi : \Omega \mapsto \mathbb{R}$ that closely resembles f, such that $\mathbb{E}[\phi] = \int_\Omega \phi(x)\mathrm{d}x$ is known, and, based on this auxiliary function ϕ, (ii) compute the following unbiased Monte Carlo estimator

$$I_{\mathrm{CV}} = \frac{1}{M}\sum_{i=1}^{M}\Big(f(x^{(i)}) - c\phi(x^{(i)})\Big) + c\int_\Omega \phi(x)\mathrm{d}x, \quad x^{(i)} \overset{\text{iid}}{\sim} \mathcal{U}(\Omega), \quad c \in \mathbb{R}. \tag{1}$$

Since $\mathrm{Var}[f - c\phi] = \mathrm{Var}[f] + c^2\,\mathrm{Var}[\phi] - 2c\mathrm{Cov}[f,\phi]$, if ϕ is chosen such that f and ϕ are strongly correlated and c is carefully selected, this method can provide a large variance reduction. Consider now the extension of this method to multiple control variates $\phi_0, \dots, \phi_n$. In this case, we get a MC estimator of the form

$$I_{MCV} = \frac{1}{M}\sum_{i=1}^{M}\left(f(x^{(i)}) - \sum_{j=0}^{n} c_j\phi_j(x^{(i)})\right) + \sum_{j=0}^{n} c_j\int_\Omega \phi_j(x)\mathrm{d}x, \quad x^{(i)} \overset{\text{iid}}{\sim} \mathcal{U}(\Omega), \tag{2}$$

where $c_j \in \mathbb{R}$, $j = 0,\dots,n$. The idea now is to choose $\{\phi_j\}_{j=0}^n$ as orthonormal polynomials in $[0,1]^d$ with $\phi_0 := 1$ and $\int_\Omega \phi_j(x)\mathrm{d}x = 0$, $j \geq 1$, e.g. in dimension $d = 1$, we can choose the first $n+1$ Legendre polynomials. Then, we estimate the coefficients c_j by solving a least squares problem using M independent samples $\{(\tilde{x}^{(i)}, f(\tilde{x}^{(i)}))\}_{i=1}^M$, drawn independently of those used in (2):

$$\min_{c \in \mathbb{R}^{n+1}} \sum_{i=1}^{M} \left(f(\tilde{x}^{(i)}) - \sum_{j=0}^{n} c_j \phi_j(\tilde{x}^{(i)}) \right)^2 , \tag{3}$$

which can be written more compactly as

$$\min_{\boldsymbol{c} \in \mathbb{R}^{n+1}} \|\boldsymbol{V}\boldsymbol{c} - \boldsymbol{f}\|_2 , \tag{4}$$

where $\boldsymbol{V} \in \mathbb{R}^{M \times (n+1)}$ is the Vandermonde matrix given by

$$\boldsymbol{V} = \begin{bmatrix} 1 & \phi_1(\tilde{x}^{(1)}) & \phi_2(\tilde{x}^{(1)}) & \dots & \phi_n(\tilde{x}^{(1)}) \\ 1 & \phi_1(\tilde{x}^{(2)}) & \phi_2(\tilde{x}^{(2)}) & & \phi_n(\tilde{x}^{(2)}) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \phi_1(\tilde{x}^{(M)}) & \phi_2(\tilde{x}^{(M)}) & \dots & \phi_n(\tilde{x}^{(M)}) \end{bmatrix} \tag{5}$$

and $\boldsymbol{f} := [f(\tilde{x}^{(1)}), \dots, f(\tilde{x}^{M})]^T$. Notice that Eq. (4) in dimension $d = 1$ corresponds to computing an n^{th}-degree polynomial approximation of f by discrete least squares from the data $\{(\tilde{x}^{(i)}, f(\tilde{x}^{(i)}))\}_{i=1}^{M}$, and the final "Monte Carlo least squares" estimator reads:

$$I_{\text{MCLS}} = \frac{1}{M} \sum_{i=1}^{M} \left(f(x^{(i)}) - \sum_{j=0}^{n} c_j^* \phi_j(x^{(i)}) \right) + c_0^* \tag{6}$$

with c^* a minimizer of (4). Moreover, notice that when $M \to \infty$, we expect the discrete least sqaures polynomial approximation $p(x) = \sum_{j=0}^{n} c_j^* \phi_j(x)$ to converge to the exact L_2 projection $\Pi_n f(x) := \sum_{j=0}^{n} \hat{c}_j \phi_j(x)$ with $\hat{c}_j = \int_\Omega f(x)\phi_j(x)\mathrm{d}x$, (where Π_n is the projection operator), and in particular, c_0^* to converge to $\hat{c}_0 = \mathbb{E}[f]$, so that an alternative estimator is simply

$$I'_{\text{MCLS}} = c_0^*. \tag{7}$$

Note that the cost of computing I_{MCLS} is twice the cost of computing I'_{MCLS} due to the generation of two data sets $\{\tilde{x}^{(i)}, f(\tilde{x}^{(i)})\}_{i=1}^{M}$ and $\{x^{(i)}, f(x^{(i)})\}_{i=1}^{M}$ respectively for the least squares fit and for the final estimator.

2 Goals of the project

1. Show that if $n = 0$, the estimators I_{MCLS} and I'_{MCLS} coincide with the standard Monte Carlo estimator I_{MC}. In addition, show that I_{MCLS} and I'_{MCLS} coincide when the same samples are used to compute the least squares fit in Eq. (4) and the estimator in Eq. (6).

2. Consider the following integral

$$I_1 = \int_0^1 \frac{1}{25x^2 + 1} \mathrm{d}x. \tag{8}$$

   We are interested in determining the error on estimating I using different methods. In particular,

(a) Compute a reference value of I_1 and a crude Monte Carlo estimator for different values of M. Plot the error as a function of M.

(b) Compute the MCLS estimators I_{MCLS} and I'_{MCLS} for the same values of M as in (a) using different values of $n = 1, 2\ldots$, and an expansion in Legendre polynomials[1] up to degree n. Plot the error as a function of M for each value of n chosen. In addition, try also $n = \left\lceil \sqrt{M} \right\rceil$ (where $\lceil x \rceil$ denotes the smallest integer a such that $a > x$) and $n = \left\lceil \frac{M}{2} \right\rceil$. Monitor at the same time the condition number of the Vandermonde matrix $\boldsymbol{V}$ and comment your results.

3. It can be shown [4, 3] that choosing n as a function of the total sample size M can provide a larger rate of convergence. However, one has to be careful when choosing n, and the data $\{(x^{(i)}, f(x^{(i)}))\}_{i=1}^M$, as it is possible to run into issues with conditioning. An approach presented in [1] to resolve these issues with conditioning is to do importance sampling as follows. Define the non-negative probability density function $1/w$ by

$$\frac{1}{w(x)} = \frac{\sum_{j=0}^n \phi_j(x)^2}{n+1}. \tag{9}$$

Having defined $1/w$, one can sample from $1/w$ instead of $\mathcal{U}(\Omega)$. Notice that in this case

$$\int_\Omega \left(f(x) - \sum_{j=0}^n c_j \phi_j(x) \right)^2 \mathrm{d}x = \int_\Omega w(x) \left(f(x) - \sum_{j=0}^n c_j \phi_j(x) \right)^2 \frac{1}{w(x)} \mathrm{d}x, \tag{10}$$

$$\approx \frac{1}{M} \sum_{i=1}^M \left(f(\tilde{x}^{(i)}) - \sum_{j=0}^n c_j \phi_j(\tilde{x}^{(i)}) \right)^2 w(\tilde{x}^{(i)}), \tag{11}$$

where $\{\tilde{x}^{(i)}\}_{i=1}^M \overset{\text{iid}}{\sim} 1/w$, which leads to the weighted discrete least-squares fitting problem

$$\min_{\boldsymbol{c} \in \mathbb{R}^{n+1}} \left\| \sqrt{\boldsymbol{W}} (\boldsymbol{V}\boldsymbol{c} - \boldsymbol{f}) \right\|_2, \tag{12}$$

with $\sqrt{\boldsymbol{W}} = \operatorname{diag}\left(\sqrt{w(\tilde{x}^{(1)})}, \sqrt{w(\tilde{x}^{(2)})}, \ldots, \sqrt{w(\tilde{x}^{(M)})} \right)$. Generalize the estimators I_{MCLS} and I'_{MCLS} to this importance sampling case.

4. Propose a method to sample from the density $1/w$. Then, implement the MCLS estimators with importance sampling and test them in the same integration problem as in point 1 with $n = 1, 2, \ldots$, $n = \left\lceil \sqrt{M} \right\rceil$, $n = \left\lceil \frac{M}{2} \right\rceil$, $n = \lceil M/\log(M) \rceil$.

5. We now implement the above methods for a more complex problem of interest. Consider the Fitzhugh–Nagumo system of ODEs. The Fitzhugh–Nagumo model is a bidimensional simplification of the Hodgkin-Huxley model, which is used extensively in the field

[1] See `https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.polynomial.legendre.Legendre.html`. This module takes the legendre polynomials to be orthonormal in the $[-1, 1]$ domain. You need to change this to $[0, 1]$ using the `domain` option of the module.

of neuroscience to model the phenomenon of spiking neurons. The dynamical equations read as follows:

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v - \frac{v^3}{3} - w + I \\ \epsilon \left(v + a - bw \right) \end{bmatrix}, \quad \begin{bmatrix} v(t=0) \\ w(t=0) \end{bmatrix} = \begin{bmatrix} v^0 \\ w^0 \end{bmatrix}, \quad t \in [0, T],$$

where $[v(t), w(t)]^T$ in $\mathbb{R}^2$ denotes the state variables and a, b, ϵ and I denote system parameters. We set $\epsilon = 0.08$ and $I = 1.0$. We assume that a and b are uniformly distributed random parameters such that $a \sim \mathcal{U}(0.6, 0.8)$ and $b \sim \mathcal{U}(0.7, 0.9)$. We set $v^0 = 0$ and $w^0 = 0$.

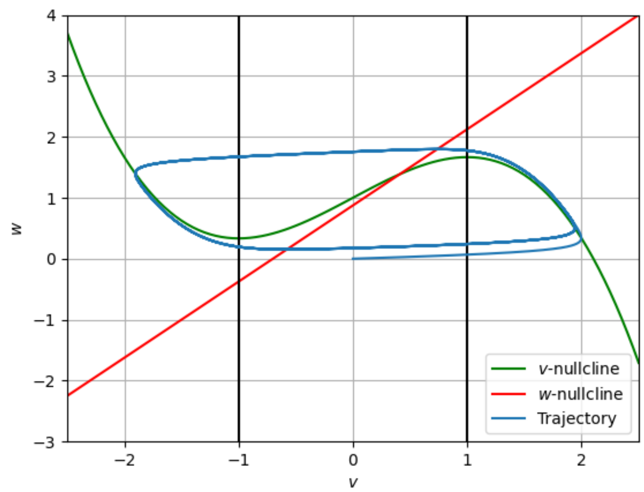

Figure 1: FitzHugh–Nagumo oscillator dynamics

(a) Discretise the interval $[0, T]$ using a uniform grid $t_j = j\Delta t, j \in \{0, 1, ..., N_T\}$, with $\Delta t = T/N_T$ and set $T = 10$ and $N_T = 1000$. Solve the system for a given realization of the parameters a and b.
Remark: One can change the value of T and N_T if the problem proves too computationally expensive for your existing hardware.

(b) We use the notation v_n to denote the approximation of $v(t_n)$ using the forward Euler scheme. Consider the following Quantity of Interest (QoI):

$$\frac{1}{T} \int_0^T v^2(t) dt \approx \sum_{n=0}^{N_T - 1} \left(\frac{(v_n)^2 + (v_{n+1})^2}{2} \right) \Delta t =: Q.$$

Implement the MC and MCLS estimators to compute the following integral:

$$I_2 = \int_{0.6}^{0.8} \int_{0.7}^{0.9} Q da db. \tag{13}$$

Test for different values of M and for $n = 0, 1, 2, 3$.

(c) Propose and implement an error estimator for the above MCLS estimator. Plot this error estimate versus M for different n. Compare this with the "true" error by using a highly accurate estimate of Eq. (13) as a reference value.

References

[1] Albert Cohen and Giovanni Migliorati. Optimal weighted least-squares methods. *The SMAI journal of computational mathematics*, 3:181–203, 2017.

[2] Abdul-Lateef Haji-Ali, Fabio Nobile, Raúl Tempone, and Sören Wolfers. Multilevel weighted least squares polynomial approximation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 54(2):649–677, 2020.

[3] Giovanni Migliorati and Fabio Nobile. Stable high-order randomized cubature formulae in arbitrary dimension. *Journal of Approximation Theory*, 275:105706, 2022.

[4] Yuji Nakatsukasa. Approximate and integrate: Variance reduction in Monte Carlo integration via function approximation. *arXiv preprint arXiv:1806.05492*, 2018.