Regression-based Monte Carlo Integration

CORENTIN SALAÜN, Max-Planck-Institut für Informatik, Germany
ADRIEN GRUSON, McGill University & École de Technologie Supérieure, Canada
BINH-SON HUA, VinAI Research, Vietnam
TOSHIYA HACHISUKA, University of Waterloo, Canada
GURPRIT SINGH, Max-Planck-Institut für Informatik, Germany

Monte Carlo integration is typically interpreted as an estimator of the expected value using stochastic samples. There exists an alternative interpretation in calculus where Monte Carlo integration can be seen as estimating a constant function—from the stochastic evaluations of the integrand—that integrates to the original integral. The integral mean value theorem states that this constant function should be the mean (or expectation) of the integrand. Since both interpretations result in the same estimator, little attention has been devoted to the calculus-oriented interpretation. We show that the calculus-oriented interpretation actually implies the possibility of using a more complex function than a constant to construct a more efficient estimator for Monte Carlo integration. We build a new estimator based on this interpretation and relate our estimator to control variates with least-squares regression on the stochastic samples of the integrand. Unlike prior work, our resulting estimator is provably better than or equal to the conventional Monte Carlo estimator. To demonstrate the strength of our approach, we introduce a practical estimator that can act as a simple drop-in replacement for conventional Monte Carlo integration. We experimentally validate our framework on various light transport integrals. The code is available at https://github.com/iribis/regressionmc.

Fig. 1. Given an integrand (a), we first sample (b) \( f(x) \) as in Monte Carlo (MC) integration. (c-d) Traditional MC estimator can be interpreted as fitting a constant model function to the sample values, with the integral of this constant function equaling to \( F \). (e) We, instead, propose to use a non-constant model function such as a polynomial, which is then fitted to the sampled values. (f) The resulting estimator is based on control variates; we add the analytical integral of the model function to MC integration of the difference between the original integrand and the model function. The bottom row shows renderings and the corresponding error images to demonstrate the impact of our regression approach against the traditional MC integration. The insets on the right compare our method with different orders (\( O_x \)) of polynomials. Our method has significant error reduction at equal time.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CCS Concepts: • Computing methodologies → Rendering.

Additional Key Words and Phrases: Monte Carlo integration, Regression, Control Variates, Light transport simulation

ACM Reference Format:
Corentin Salaün, Adrien Gruson, Binh-Son Hua, Toshiya Hachisuka, and Gurprit Singh. 2022. Regression-based Monte Carlo Integration. ACM Trans. Graph. 41, 4, Article 79 (July 2022), 14 pages. https://doi.org/10.1145/3528223.3530095
1 INTRODUCTION

Monte Carlo (MC) integration is a basic tool for many numerical integration problems. In textbooks, MC integration is typically interpreted as a statistical process where it samples the integrand and estimates the expected value by taking the average of these samples. While less common, there is another interpretation of MC integration based on the mean value theorem in calculus. The mean value theorem for definite integrals tells us that there exists a constant that integrates exactly to the same solution as the given definite integral (see Fig. 1). This constant, by definition, is equal to the mean of the integrand, which can be again estimated by taking the average of random samples, and thus both interpretations result in the same estimator. This alternative interpretation, however, raises interesting questions: Can we use a non-constant function to construct a MC estimator? If so, what happens? These questions do not appear in the statistics-oriented interpretation as the expected value is fundamentally tied to the idea of samples’ average.

We propose a new MC estimator that takes the same input as MC integration (see Fig. 1b) with no extra assumptions. Instead of merely taking the average to find a constant function (Fig. 1c), we propose to fit a non-constant model function (Fig. 1e) to the samples via least-squares regression and utilize its analytical integral to define an estimator (Fig. 1f). We show that our approach can be formulated as the method of control variates, where control variates are obtained via least-squares regression.

While the use of regression in control variates is not new, we show, for the first time how regression, control variates, and MC integration are closely coupled. MC integration is a special case under our formulation where a constant function is the solution of regression. Our approach is provably shown to be better than or equivalent to MC integration; in the worst case, it will revert back to conventional MC integration as long as a constant function is included in regression. This theoretical guarantee has not been explored in prior work on regression-based control variates. To summarize our contributions:

- We introduce a novel formulation of MC integration which connects least-squares regression, control variates, and conventional MC integration.
- We prove that our regression-based formulation leads to a class of estimators that are better than or equal to conventional MC integration in terms of variance.
- We develop a practical algorithm based on our formulation and polynomials, which can work as drop-in improvement over conventional MC integration.
- We demonstrate the effectiveness of our algorithms over light transport integral estimations by performing experiments over multiple dimensions (2D to 15D).

2 RELATED WORK

2.1 Numerical integration

Integration is a classical subject in numerical analysis [Davis 2013; Trefethen 2012]. The calculus-oriented approach has lead to deterministic quadrature techniques such as a Riemann sum. The quadrature techniques sample the integrand at predetermined points and approximate the given integrand using an analytically integrable function (e.g., quadrangles, trapezoids, and certain basis functions). The calculus-oriented approach, in general, suffers from the exponential growth of the number of samples for higher dimensional functions. A statistics-oriented approach, MC integration, performs a similar summation but instead evaluates the integrand at stochastically sampled locations. This stochastic nature of sample generation extends well to higher dimensions [Niederreiter 1992; Owen 2013] without the exponential growth of the number of samples, which has led to its wide adoption in the rendering community. We take a step back from this trend and show how a calculus-oriented formulation of MC integration can lead to more efficient numerical integration methods.

2.2 Control variates

Control variates (CVs) [Glynn and Szechtman 2002; Loh 1995] are variance reduction techniques which utilize MC integration of the difference between a given integrand \( f \) and another function \( g \) to perform numerical integration. This difference estimate is added to the analytical integral of the function \( g \) to estimate the integral of \( f \). The improvement over conventional MC integration is dependent on how well \( g \) approximates \( f \), and the improvement is not guaranteed in general.

Owen and Zhou [2000] discussed a practical approach to perform importance sampling with control variates for variance reduction. They showed that adaptive control variates using a mixture density cannot be much worse (i.e., can only be slightly worse) than conventional MC using the best mixture component. Our approach, on the other hand, is provably never worse than conventional MC using the same PDF (shown in Section 4.3 and Equation (25)). Fan et al. [2006] used the method of Owen and Zhou in rendering with an additional approximation.

Multi-level Monte Carlo [Heinrich 2001; Keller 2001] shows how low resolution renderings can be used as CVs for high-resolution synthesis. Rousselle et al. [2016] shows a connection between CVs and the reconstruction step in gradient-domain rendering [Hua et al. 2019]. Recently, Kondapaneni et al. [2019] showed a connection between weights in multiple importance sampling [Veach and Guibas 1995] and CVs. We show connections among control variates, least-squares regression, and MC integration to have a provable variance reduction. We will discuss more related work [Crespo et al. 2021; Hickernell et al. 2005; Nakatsukasa 2018; Owen and Zhou 2000; Pajot et al. 2014; Rubinstein and Marcus 1985] later once we introduced our approach.

Construction of control variates in rendering. Constructing control variates for complex integrands in light transport simulation is a challenging task. In rendering, the simplest form of control variate uses the constant ambient term to approximate incoming illumination [Lafortune and Willems 1994]. This, however, does not work well in complex lighting conditions. Clarberg and Akenine-Möller [2008] used visibility correlations to design control variates for direct illumination. Vévoda et al. [2018] used bayesian regression to construct a light clustering. For further improvement, a control variate function is derived from the respective estimated contribution of each cluster at any shading point. Kutz et al. [2017] proposed a method of control variate extension for heterogenous participating media rendering. It is based on the decomposition of the medium into a control and a residual components. These two parts can be sampled separately. The control variate has also been used by Belcour et al. [2018] to extend their method of integration of spherical harmonic expansions over polygonal domains to non-analytic integrands. Neural control variates [Müller et al. 2020] fit a neural network to define control variates. They also proposed to simultaneously train another network to importance sample the difference function. The resulting estimator shows remarkable improvements.
but with a significant computational overhead and changes to the sampling process that have to be accelerated by utilizing a high-performance multi-GPU workstation. Recently, Subr [2021] propose a neural network Q-NET that uses a neural proxy as a control variate. Our control variates are constructed from the samples within each pixel using least squares regression.

The expected improvement from these control variates is limited by construction due to the lack of complexity captured by these control variates. It is thus reasonable to assume that control variates usually do not approximate the integrand well in light transport simulation. Recognizing this limitation, our approach is designed to be provably better than MC integration as long as least-squares regression can find any non-constant solution. A major deviation from prior work is that we show how regression is closely connected to control variates and MC integration. Many prior work on control variates instead focus on introducing a sophisticated model function for control variates. This conceptual difference allows us to realize that we can use a significantly simpler model function, such as polynomials, and still provably outperforms MC integration. We demonstrate that our estimators are already practical without any special optimization for regression (e.g., use of high performance hardware like GPUs) and without any change to the sampling process.

2.3 Regression in rendering
Regression has been applied for adaptive sampling and reconstruction in rendering [Zwicker et al. 2015]. Moon et al. [2014] proposed to locally approximate the unknown image function by locally regressing a low-order polynomial on auxiliary buffers (normals, depth, textures). To alleviate the computational overhead of the method, Moon et al. [2015] used a iterative linear model to simultaneously reconstruct multiple pixels. These methods control the filtering bandwidths locally to increase numerical accuracy. Bitterli et al. [2016] analyze various zero-order and first-order regression models and use auxiliary data just to fit the data and considers only the pixel color to compute the regression weights. Our framework is largely agnostic to how regression is implemented, and we demonstrate multiple implementations as examples. The regression part of our framework can thus utilize findings in those previous.

3 TWO INTERPRETATIONS OF MC INTEGRATION
Let us motivate our main question by recapitulating the two different interpretations of MC integration. Suppose that we want to find a definite integral \( F \) of a non-negative function \( f(x) \) over the domain \( \Omega \) where

\[
F = \int_{\Omega} f(x) dx. \tag{1}
\]

In many practical problems, it is difficult to find the analytical solution to the above. In light transport simulation, the function \( f(x) \) is defined as the measurement contribution function for a light transport path \( x \) [Veach 1997]. The definite integral in this case is the solution to light transport simulation, which is generally unavailable in an analytical form.

Rather than trying to find an analytical solution, MC integration uses a numerical estimator \( \langle F \rangle \) where

\[
\langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}. \tag{2}
\]

The variables \( x_i \) are \( N \) independent random samples distributed according to the probability density function \( p(x) \) which satisfies \( p(x) \neq 0 \) whenever \( f(x) \neq 0 \) in the domain \( \Omega \). This estimator \( \langle F \rangle \) is said to be unbiased in the sense that its expected value is exactly equal to the integral \( F \):

\[
E\{\langle F \rangle\} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} = \frac{1}{N} \sum_{i=1}^{N} \int_{\Omega} \frac{f(x)}{p(x)} dx = F. \tag{3}
\]

Change of variables. The same estimator can also be redefined in a unit hypercube \( U \) of uniformly distributed random numbers \( u_i \) that are used for generating samples \( x_i \). Suppose that we have a mapping \( x = \Phi(u) \) where the Jacobian \( |dx/du| = 1/p(x) \). By change of variables, we can rewrite the integral into

\[
F = \sum_{i=1}^{N} \frac{1}{N} \int_{U} \frac{f(\Phi(u))|\partial \Phi(u)/\partial u|}{p(\Phi(u))} du = \int_{U} \hat{f}(u) du, \tag{4}
\]

where \( \hat{f}(u) = \frac{f(\Phi(u))}{p(\Phi(u))} \), and rewrite the estimator \( \langle F \rangle \) into

\[
\langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i). \tag{5}
\]

In other words, by properly redefining the integrand as above, one can always consider having uniformly distributed samples \( u_i - p(u) = 1 \) in this unit hypercube \( U \), instead of samples \( x_i - p(x) \) in the domain \( \Omega \) which are generally not uniformly distributed. This definition is known as as primary sample space [Kelemen et al. 2002] in rendering. We use this formulation in the rest of the paper. One can generally convert Equation 1 to Equation 4 as long as change of variables from \( x \) to \( u \) is possible.

3.1 Statistic-oriented interpretation
The process of MC integration can be seen as an estimation of the expected value of \( \hat{f}(u) \) since we have

\[
F = \int_{U} \hat{f}(u) p(u) du = E \{ \hat{f}(u) \} = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i) = \langle F \rangle \tag{6}
\]

where \( u \) is distributed according to \( p(u) = 1 \). Under this interpretation, one can also derive an expression of the expected squared errors of \( \langle F \rangle \) as

\[
\text{Var}(\langle F \rangle) = \frac{1}{N} \text{Var}[\hat{f}(u)]. \tag{7}
\]

where we used the unbiasedness of the estimator \( E\{\langle F \rangle\} = F \). It is a well known expression that shows that the expected squared error of MC integration is proportional to the variance of \( \hat{f}(u) \) and inversely proportional to the number of samples \( N \).

3.2 Calculus-oriented interpretation
One can also interpret MC integration as a two-step process; replacement of the integrand \( \hat{f} \) by a model function \( \hat{g} \) and succeeding analytical integration of \( \hat{g} \). Let us consider a constant function \( \hat{g}(u) = c \) as the model function. To keep its integral \( G \) equal to the original integral \( F \), we would like to find a value for \( c \) such that

\[
F = \int_{U} \hat{f}(u) du = \int_{U} \hat{g}(u) du = G. \tag{8}
\]

The integral mean value theorem in calculus shows that there exist such a constant \( c \) for a given definite integral. Since the function \( \hat{g}(u) \) can be analytically integrated, we have

\[
F = G = \int_{U} \hat{g}(u) du = c |U| \implies c = \frac{1}{|U|} \int_{U} \hat{g}(u) du \tag{9}
\]
where $|U| = 1$ is the volume of the unit hypercube $U$. Therefore, the constant $c$ can be derived as the average of $\hat{f}(u)$ since

$$c = \frac{1}{|U|} \int_{U} \hat{f}(u)du = \frac{1}{|U|} \int_{U} \hat{F}(u)du$$  \hspace{1cm} (10)

where the last expression is the definition of the average of $\hat{f}(u)$ over the unit hypercube $U$. Note that there is no approximation introduced until this point. Since $c$ is defined as the average of $f(u)$, we can estimate $c$ by taking the average of samples:

$$c = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i) = \langle F \rangle.$$  \hspace{1cm} (11)

This interpretation thus leads to the same estimator as the statistics-oriented interpretation since

$$F = \int_{U} \hat{g}(u)du = \int_{U} cdud = \int_{U} \left( \frac{\langle F \rangle}{|U|} \right) du = \langle F \rangle.$$  \hspace{1cm} (12)

Under this interpretation, MC integration can be seen as the process of first estimating a function $\hat{g}(u) = c$, and then analytically integrating (the approximation of) the function $\hat{g}(u)$.

### 3.3 Problem statement

One might see that the calculus-oriented interpretation is rather redundant at first, since the end result is the same. However, this interpretation actually points out the existence of $\hat{g}$ in the definition of the estimator, and also the fact that MC integration is derived by using a constant function $\hat{g} = c$. This existence of $\hat{g}$ is not apparent in the statistics-oriented interpretation.

Our main question is whether making $\hat{g}$ non-constant is possible and whether it is beneficial to consider such $\hat{g}$ to begin with. We show that such a $\hat{g}$ can be defined based on control variates and least-squares regression of $\hat{g}$ to $\hat{f}$. Our formulation shows that conventional MC integration is a special case where regression happens to return a constant function for $\hat{g}$. We also show that considering a more complex $\hat{g}$ is provably better than or equal to simply having a constant $\hat{g}$ in terms of its expected error.

### 4 REGRESSION-BASED MC INTEGRATION

We introduce a novel formulation of MC integration, regression-based MC integration, which allows us to consider any analytically integrable functions for $\hat{g}$, in place of the constant function. Fig. 1 illustrates our approach for a 1D function.

We start by considering an arbitrary model function $\hat{g}(u)$ which can be analytically integrated. Unlike the calculus-oriented interpretation, our function $\hat{g}(u)$ can be non-constant and the integral $G$ does not need to be equal to the original integral $F$. We can use such $\hat{g}(u)$ by adding the difference from the original integral as

$$F = G + (F - G)$$  \hspace{1cm} (13)

where $(F - G)$ is estimated by MC integration;

$$F - G = \frac{1}{N} \sum_{i=1}^{N} (\hat{f}(u_i) - \hat{g}(u_i)) = \langle D \rangle.$$  \hspace{1cm} (14)

The final estimator is defined as a sum of two terms where

$$F \approx \langle F \rangle^* = G + \langle D \rangle.$$  \hspace{1cm} (15)

The first term $G$ is analytically evaluated for this given $\hat{g}(u)$ in a deterministic manner. This estimator is not new and is known as the method of control variates. The function $\hat{g}(u)$ is called the control variate which is typically given by a user. In our method, we find $\hat{g}(u)$ by (least-squares) regression.

Control variates are unbiased regardless of $\hat{g}(u)$. Since $G$ is deterministic, the expected squared error of $(F)^*$ is given by the difference estimator

$$E \left[ \left( (F)^* - F \right)^2 \right] = \text{Var} \left[ \langle D \rangle \right] = \frac{1}{N} \text{Var} \left[ \hat{f}(u_i) - \hat{g}(u_i) \right].$$  \hspace{1cm} (16)

Control variates thus lead to a smaller expected square error compared to conventional MC integration only when $\text{Var} \left[ \hat{f}(u_i) - \hat{g}(u_i) \right] < \text{Var} \left[ \hat{f}(u_i) \right]$. Error reduction is thus not guaranteed and it is widely recognized so [Hickernell et al. 2005].

Our key contribution is to show that least-squares regression not only helps us automatically finding $\hat{g}(u)$ but also provably reduces the expected squared error when a constant function is included in regression.

As we discuss later, a constant function has been usually ignored in (adaptive) control variates. Our work is the first to show that this combination of the least-square regression and inclusion of a constant function results in the provable improvement with a proper reduction to MC integration.

Fig. 2 showcases our method for simple 1D integration problems. The labels $O1, O3,$ and $O5$ in our method mean different orders of polynomials used for $\hat{g}$ as we explain later. The convergence plots show that, after a certain number of samples, our method is no worse than MC integration regardless of the order of polynomials. It is also worth noting that regression might not well approximate the integrand (e.g., step and complex cases). The method of control variates alone with such a bad approximation could increase the error. Even in such cases, our method still outperforms MC integration once regression converges after some numbers of samples. When the integrand can be modeled exactly by $\hat{g}$, as shown in the last row where the integrand is a 5th-order polynomial, our method dramatically reduces numerical error compared to MC integration. We now explain how our estimator $(F)^*$ achieves this reduction of errors by showing its connections to control variates and MC integration.

#### 4.1 Least-squares regression of $\hat{g}(u)$

We model $\hat{g}(u, \theta)$ as a parametric function with the parameters $\theta = (\theta_0, \cdots, \theta_M)$. We define the residual for $\theta$ as

$$R(\theta) = \int_{U} \left( \hat{f}(u) - \hat{g}(u, \theta) \right)^2 du.$$  \hspace{1cm} (17)

Let us then consider regression of $\hat{g}(u)$ to $\hat{f}(u)$ such that the residual $R(\theta)$ is minimized. Since analytical solution to the integral in $R(\theta)$ is usually not available, we first consider using $N$ random samples $u_i$ to approximate $R(\theta)$

$$R(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left( \hat{f}(u_i) - \hat{g}(u_i, \theta) \right)^2 = \langle R \rangle(\theta),$$  \hspace{1cm} (18)

which is equivalent to MC integration of the function $(\hat{f}(u) - \hat{g}(u, \theta))^2$ using the estimator $\langle R \rangle(\theta)$. This unbiased approximation thus converges to $R(\theta)$ as we increase the number of samples. This expression of $\langle R \rangle(\theta)$ can also be seen as the squared $L_2$ norm of the differences between $\hat{f}(u_i)$ and $\hat{g}(u_i, \theta)$ on the samples $u_i$. The solution to this regression can be obtained via least-squares regression of $\hat{g}(u, \theta)$ to $N$ pairs of $(u_i, \hat{f}(u_i))$. 

ACM Trans. Graph., Vol. 41, No. 4, Article 79. Publication date: July 2022.
4.2 Connection to control variates

The original definition of control variates considers a given function \( h(u) \) and a scalar parameter \( \alpha \) such that

\[
F = \int_U \tilde{f}(u)\,du = \alpha H + \int_U (\tilde{f}(u) - \alpha h(u))\,du
\]  

(19)

where \( H = \int_U h(u)\,du \). Given this definition, it is known that

\[
\alpha = \text{Cov} \left[ \tilde{f}, h \right] / \text{Var} [h]
\]  

(20)

minimizes the variance of \( \langle F \rangle^* \). This approach can be seen as a special case of our formulation where \( \theta = \alpha \) and \( \tilde{g}(u, \theta) = \alpha h(u) \).

Let us analyze the error of \( \langle F \rangle^* \) when \( \tilde{g}(u, \theta) \) is given by least-squares regression. Since the expected squared error of \( \langle F \rangle^* \) is equal to the variance of \( \tilde{f} - \tilde{g} \) (Equation (16)), we have

\[
\text{Var} \left[ \tilde{f}(u) - \tilde{g}(u, \theta) \right] = E \left[ (\tilde{f}(u) - \tilde{g}(u, \theta))^2 \right] - E \left[ \tilde{f}(u) - \tilde{g}(u, \theta) \right]^2
\]

\[
= \int_U (\tilde{f}(u) - \tilde{g}(u, \theta))^2 \, du - (F - G)^2 = R(\theta) - (F - G)^2.
\]  

(21)

where we used the fact that \( \rho(u) = 1 \) in the primary sample space and substituted \( R(\theta) \). The expected squared error of the estimator is thus bounded as

\[
E \left[ (\langle F \rangle^* - F)^2 \right] = \frac{1}{N} \left( R(\theta) - (F - G)^2 \right) \leq \frac{1}{N} R(\theta).
\]  

(22)

We can see that, by defining the control variate \( \tilde{g}(u) \) via least-squares regression, one can also minimize (the bound of) the expected squared error of the estimator. In practice, we cannot directly minimize \( R(\theta) \), but we minimize its estimate \( \langle R(\theta) \rangle \), so this relationship does not exactly hold for regression with a finite number of samples as one can also see in Fig. 2. Our experiments, however, show that minimizing \( \langle R(\theta) \rangle \) works well in practice when \( N \) is not too small.

Note also that control variates directly minimize the variance by setting \( \alpha \), while we minimize only its bound by least-squares regression.

While it is tempting to consider more general regression to directly minimize the variance in our method, it is the use of least-squares regression which allowed us to reveal clear connections among least-squares regression, control variates, and MC integration as we see in the followings.

4.3 Reduction to Monte Carlo integration

It is now trivial to show that our approach reduces to MC integration when regression happened to return a constant function \( \tilde{g}(u) = c \) as its best fit. Substituting \( \tilde{g}(u) = c \) to Equation (15),

\[
\langle F \rangle^* = c|U| + \frac{1}{N} \sum_{i=1}^{N} (\tilde{f}(u_i) - c) = \frac{1}{N} \sum_{i=1}^{N} \tilde{f}(u_i) = \langle F \rangle.
\]  

(23)

It shows that, when \( \tilde{g}(u) = c \) is the solution to regression, our approach reduces to MC integration.
In this case, since we have \( F = G \), the upper-bound in Equation (22) becomes exact and

\[
E \left[ \left( (F^*)^2 - (F)^2 \right) \right] = E \left[ \left( (F) - F \right)^2 \right] = R(c) \tag{24}
\]

where \( R(c) \) is the residual for \( \hat{g}(u) = c \).

Based on this reduction to Monte Carlo integration, we can show for the first time that the expected squared error of this approach is provably better than or equal to MC integration. Deceivingly simple, but the important difference from prior work is that we include a constant function in least-squares regression by adding \( c \) to \( \hat{g}(u) \) and we minimize the residual \( R(\theta) \). This allows us to have a “safe-guard” against performing worse than MC integration, when combined with least-squares regression. Since \( R(\theta) \) is minimized for a model function including a constant, \( R(\theta) \leq R(c) \).

It thus follows that

\[
E \left[ \left( (F^*)^2 - (F)^2 \right) \right] \leq \frac{1}{N} E \left[ \left( (F) - F \right)^2 \right],
\]

This allows us to have a “safe-guard” against performing worse than MC integration. We explain two different approaches for regression. While regression is often used as an approach to find a control variate, none of the prior work shows how including a constant function in regression allows us to unify control variates and MC integration. For example, a thorough discussion on control variates [Hickernell et al. 2005, Section 4] immediately dismisses a constant control variate as redundant since it will not reduce variance at all.

Recognizing a problem of control variates that it could perform worse than MC integration, Pajot et al. [2014] proposed switching to conventional MC when the estimated variance of adaptive control variate is determined to be larger. Our approach does not need this explicit switching, as it is automatically done as a result of least-squares with theoretical guarantee. Although our estimator looks similar to a general mixture estimator as Owen and Zhou [2000], their estimator does not revert back to MC integration. The difference becomes apparent when we fit a mixture density including a constant function. When regression returns a constant as the solution, the method of Owen and Zhou also changes the sampling density to a constant. On the other hand, our method reduces to MC integration while keeping the sampling density as the given mixture density. The two estimators are thus fundamentally different.

Some existing methods [Rousselle et al. 2016; Rubinstein and Marcus 1985] use multiple different control variate functions to take advantage of their respective strengths. Rubinstein and Marcus [1985], in particular, shows that the variance reduction depends on the optimal number of control variates used in the estimator. Our work is orthogonal to this work.

Algorithm 1: Direct matrix estimator.

\[
f \leftarrow \{\}; u \leftarrow \{\}; \Phi_0 \leftarrow 0^{M \times M}; b_0 \leftarrow 0^{M \times 1};
\]

\textbf{for} \( i \leftarrow 1 \) to \( N \) \textbf{do}

\( u_i \leftarrow \text{random}(); \)

\( f_i \leftarrow \hat{f}(u_i); \)

\( (\Phi_{i+1}, b_{i+1}) \leftarrow \text{update_system}(\Phi_i, b_i, u_i, f_i); \)

\textbf{end}

\( \theta_{\min} \leftarrow \text{solve_linear_system}(\Phi_{N+1}, b_{N+1}); \)

\( \text{return } G(\theta_{\min}) + \frac{1}{N} \sum_{i=1}^{N} f_i - \hat{g}(u_i, \theta_{\min}); \)

5 POLYNOMIAL-BASED ESTIMATORS

While our formulation supports arbitrary integrable functions for \( \hat{g} \), we elaborate a practical implementation with a polynomial function for \( \hat{g} \) to demonstrate the strength of our method. In general, a polynomial function can be written as

\[
\hat{g}(u) = \sum_{q=0}^{M} c_q \phi_q(u)
\]

with monomials \( \phi \) defining a polynomial. We can analytically integrate \( \hat{g} \) as the weighted sum of the integrals of monomials as

\[
\int_{U} \hat{g}(u)du = \sum_{q=0}^{M} c_q \int_{U} \phi_q(u)du.
\]

The advantage of polynomials is that it is simple and easily generalizable for different dimensions with different orders of polynomials. Note that we do not claim that polynomials are optimal as control variates. The integrand \( f \) is, in general, far more complex than a polynomial, and it will not be well approximated by a polynomial in many cases. Using polynomials in the conventional control variates might be seen as a bad approach because of this reason.

What is important here is that a polynomial is still more complex than a constant and includes a constant. Our formulation thus provably guarantees improvement over MC integration in this case. Therefore, even with a lower-order polynomial, it is expected that our method either outperforms or at least performs equally to MC integration. We explain two different approaches for regression.

5.1 Direct matrix estimator

It is well known that regression with a polynomial can be formulated as a solution to a linear system. The size of this linear system is proportional to the number of monomials \( M \) and independent from the number of samples \( N \), making it suitable for MC integration.

We can define a linear system by taking partial derivatives with respect to coefficients and setting all to zero

\[
\frac{\partial R(\theta)}{\partial c_q} = 0 \quad \forall q \in \{0, 1, \ldots, M\}.
\]

Since \( R(\theta) \) is the squared \( L_2 \) norm of the differences between \( f(u_i) \) and \( \hat{g}(u_i, \theta) \), the solution to this system is guaranteed to give a global minimum of \( R(\theta) \). In a matrix form, this linear
system can be represented as
\[
\begin{bmatrix}
\Phi(0,0) & \cdots & \Phi(0,M) \\
\Phi(1,0) & \cdots & \Phi(1,M) \\
\vdots & & \vdots \\
\Phi(M,0) & \cdots & \Phi(M,M)
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
\vdots \\
c_M
\end{bmatrix}
= \begin{bmatrix}
\sum_{i=1}^{N} \hat{f}(u_i) \phi_0(u_i) \\
\sum_{i=1}^{N} \hat{f}(u_i) \phi_1(u_i) \\
\vdots \\
\sum_{i=1}^{N} \hat{f}(u_i) \phi_M(u_i)
\end{bmatrix}
\] (30)

where the function \(\Phi(p,q)\) represents
\[
\Phi(p,q) = \sum_{i=1}^{N} \phi_p(u_i) \phi_q(u_i)
\] (31)

which is precomputed for given bases \(\phi\). Updating the full matrix requires \(M^2 + M\) floating point multiplication and summation if we ignore the cost of evaluating the monomials.

Since we need to evaluate the resulting \(\hat{g}(u)\) at each of \(u_i\) after regression, we chose to store all the \(N\) samples first, solve the linear system, and then evaluate \(\hat{g}(u_i)\) afterward. An iterative approach which does not require storing all the samples is possible under the matrix formulation [Nakatsukasa 2018], but we leave it as future work and use the simpler alternative to store all the samples.

Note also that the size of the matrix stays the same regardless of the number of samples \(N\). We use complete orthogonal decomposition as a solver as it is fast for small matrices and does not require that the matrix is invertible (which may happen when \(N\) is small). Algorithm 1 shows the details of this matrix-based approach.

5.2 Gradient descent estimator

Since our formulation does not assume a specific approach for regression, we can use an alternative approach based on gradient descent. Rather than setting partial derivatives to zero to define a linear system, we use them as a gradient to iteratively update \(\theta_k\) by starting from its initial guess \(\theta_1\) and calculating
\[
\theta_{k+1} = \theta_k - \gamma \left( \frac{\partial \hat{r}(u,\theta)}{\partial \theta_k} \right)
\] (32)

where \(\gamma\) is a parameter which we set to 0.01. Using stochastic gradient descent, we can modify the above equation by observing that \(u_i\) is a random sample used for Monte Carlo estimator \((R)(\theta)\) and use only one sample for \(k = 1, \cdots, N\) as
\[
\theta_{k+1} = \theta_k - \gamma \left( \frac{\partial \hat{r}(u_k,\theta)}{\partial \theta_k} \right).
\] (33)

where we defined \(\hat{r}(u,\theta) = (\hat{f}(u) - \hat{g}(u,\theta))^2\) for simplicity. This process can be repeated multiple times over the same set of samples to potentially reach a better solution for regression.

This gradient descent approach is quite general and can be used for more complex integrable functions than polynomials such as neural networks [Müller et al. 2020]. In the case of polynomials, the computation of gradients is straightforward and only requires one evaluation of \(\hat{g}(u)\) for each sample \(u_i\) and the evaluations of the monomials. We thus found that the gradient descent approach is generally less computationally costly than the matrix-based approach. While more sophisticated algorithms may perform better, we have found that the above simple algorithm works well in our case. Algorithm 2 shows the details.

Discussion. The gradient descent approach is general and updating the coefficient is less costly compared to solving a matrix. The matrix approach can be used only when the residual is quadratic to the parameters, as in the case for polynomials. In more general cases where the residual is not a quadratic function of the parameters, one can still use the gradient descent approach. We advise to use the matrix for polynomials because it directly solves for the optimal solution.

At a low sample count, however, we found that the matrix approach can sometimes be numerically unstable as the matrix could be ill-conditioned. We have observed this issue in particular
Algorithm 2: Gradient descent estimator.

Parameter: \( lr \): learning rate, \( T \): number of iterations

\[
\begin{align*}
&f \leftarrow \{ \}; \ u \leftarrow \{ \}; \\
&\text{for } i \leftarrow 1 \text{ to } N \text{ do} \\
&\quad f_i \leftarrow f(u_i); \\
&\quad \theta_i \leftarrow 0^{M \times 1}; \\
&\text{for } i \leftarrow 1 \text{ to } 7 \text{ do} \\
&\quad \theta_i \leftarrow \text{gradient_descent}(\theta_i, lr, u_i, f_i); \\
&\text{return } G(\theta_{\text{min}} + \sum_{i=1}^{N} f_i - g(u_i, \theta_{\text{min}}));
\end{align*}
\]

To handle a vector (i.e., RGB) integrand in rendering, we first estimate the RGB estimate \( F_{rgb} \) by averaging RGB samples just as \( F_{rgb} = \frac{1}{N} \sum_{i=1}^{N} f_{rgb}(u_i) \) where \( f_{rgb}(u_i) \) is the integrand that returns an RGB sample. We then perform regression on the luminance value of the samples, resulting in the luminance value \( G \) as the analytical integral. The reconstruction of the RGB value based on our estimator is written as \( (F_{rgb}) = ((F_{rgb})/u_i(F_{rgb}))(F) \) where \( y \) is the luminance function and \( (F) \) estimate the luminance value of \( F_{rgb} \) based on our method. In other words, we correct the noisy luminance value of \( F_{rgb} \) by our estimate.

6.1 Low-dimensional comparisons

We show in Fig. 1 and Fig. 3 an equal-time comparison between the conventional MC integration and our approach using polynomials of order 1 to 3 on three different scenes. The Dining Room scene (Fig. 1) is composed of a single triangle light source for direct illumination, making it a 2-dimensional integration problem. Our method shows significant variance reduction compared to MC integration as predicted. Even in the penumbra region where the reduction of error is small, our method is no worse than MC integration. The Killeroos scene (top row, Fig. 3) is composed of a single spherical light and motion blur, which is a 3-dimensional integration problem. Despite the fact that motion blur introduces more discontinuities in the integrand—which makes it difficult to approximate by polynomials of any order—our method can still reduce the variance in most of the regions. This result demonstrates that it is possible to achieve variance reduction with model functions that only roughly approximate the underlying integrand.

The Bathroom scene (bottom row, Fig. 3) has multiple mesh light sources picked randomly using a random number, making it also a 3-dimensional integration problem. Moreover, each light in this scene is a mesh light with several triangles making discontinuities in the 2D area light parameterization. For this scene, the improvement in our method is limited but no worse than MC integration.

Fig. 4 shows the error convergence plots for the scenes described above. As predicted by our theory, our method consistently outperforms MC integration once the solution to regression is almost converged. At a fewer sample count than 10 spp, our method with higher order polynomials (order two and three) may perform worse than Monte Carlo integration due to the error in regression. This is because the matrix tends to be ill-conditioned due to noise in the sampling. Even though a linear model function (O1) is only slightly more complex than a constant model function (MC), our method with O1 significantly outperforms MC integration. We also observe that the maximum possible reduction in variance is limited by the complexity of the model function. There is thus a trade-off between the maximum reduction of variance and its stability at a few sample count, when one chooses a model function. The order or number of basis also seems to impact the overhead.

6.2 Combination with multiple importance sampling

In Fig. 5, we show that our method can be used in conjunction with MIS [Veach and Guibas 1995]. From different sampling strategies available during Multiple importance sampling (MIS), we simply perform the regression on weighted samples from each sampling strategy.
Fig. 5. Equal sample comparison of two MIS techniques (BRDF+light sampling) is performed using balance heuristics [Veach 1997] and optimal weights [Kondapaneni et al. 2019]. The combination with our method uses an order-2 polynomial basis. (a-d) Error maps and the relMSE values marked at the bottom shows improvement brought by our method.

Fig. 5 shows the results using 64 samples per pixel of our method with two different importance sampling techniques: BRDF and light sampling. We compare the balance heuristic [Veach 1997] and the optimal weights [Kondapaneni et al. 2019]. Our method improves the quality of the result for both weighting schemes compared to Monte Carlo counterpart. We also found that the combination of our method with optimal weight [Kondapaneni et al. 2019] keeps the strength of these two techniques, leading to the best variance reduction. The error reduction brought by the optimal weights allows a closer regression to the integrand leading to a greater error reduction. This result indicates that our method can take advantage of orthogonal variance reduction methods that simplify the shape of the integrand in primary sample space.

6.3 High-dimensional comparisons

Our method directly extends to higher dimensions.

We start the numerical analysis with simple analytic integrands. In Fig. 6, we show the convergence curves over 1D, 5D and 15D space for two integrands: (a) sum of sinusoids and (b) a multi-dimensional exponential function. In both cases, our method shows significant improvements in variance reduction. Both O1 and O2 polynomials gives significant improvements. However, at low sample count (up to about 100 samples) in 15D, O2 polynomial performs poorly. This is because with increasing polynomial order, in higher dimensions, more samples are needed to better approximate the control variate.

We apply our method to a path tracing rendering. Fig. 7 shows the results of our method for a duck-cornell scene rendered in path tracing with 3 bounces. This corresponds to a 15D problem in our algorithm: a 3D random number for each next-event estimation and a 2D random number for the next-direction sampling at each bounce. Improvements are visible in both directly- and indirectly-illuminated regions. However, the improvements in the indirectly-lit regions are less due to the strong variations of the function. These variations come from the light making several bounces, which makes the control variate function less accurate than in the case of direct illumination. One can of course consider having a more complex model function, but it will put an additional computational overhead on our method.

Fig. 8 shows the improvement due to our method on illumination coming from different number of bounces. Our method provides the largest improvement on direct illumination (the top row), and the improvement generally decreases with more bounces (the middle and the bottom rows). As we discussed above, it is not directly due to the increased number of dimensions for higher number of bounces. We speculate that it is rather specific to the fact that the integrand in the primary sample space becomes more and more complex for higher number of bounces. Similar observations have been made in prior work that attempt to approximate the integrand in the primary sample space [Crespo et al. 2021; Guo et al. 2018; Zheng and Zwicker 2019], so we believe that it is not specific to our regression approach. We
We also show on columns (d)-(f) the error images of MC and ACM Trans. Graph., Vol. 41, No. 4, Article 79. Publication date: July 2022.

In this case, the number of discontinuities inside the primary sample space by recursively splitting the space based on an error heuristic built upon nested quadrature rules and regions’ volume. We leave the possibility of extending our approach over a block of pixels as future work.

Fig. 9 shows an equal-sample comparison between Crespo et al. and our method. We found that their importance sampling strategy can increase variance, which can be explained by closely investigating their adaptive subdivision scheme. In the first row of Fig. 9, Crespo et al.’s control variate function misses important integrand features. They used this approximation for importance sampling as well which may increase variance.

Fig. 11 demonstrates the impact of importance sampling in Crespo et al.’s method. We found that their importance sampling strategy can increase variance, which can be explained by closely investigating their adaptive subdivision scheme. In the first row of Fig. 9, Crespo et al.’s control variate function misses important integrand features. They used this approximation for importance sampling as well which may increase variance.

In summary, due to the lack of robustness during the quadrature construction caused by their fixed sample locations and error heuristics, Crespo et al.’s method does not consistently outperform the classical MC overall. In the case of inaccurate subdivisions of the primary sample space, their method might result in inefficient importance sampling, leading to higher errors than MC. By contrast, our method is provably never worse than the MC estimator despite that ours could result in rather coarse subdivisions of the primary sample space, their method might result in inefficient importance sampling, leading to higher errors than MC. By contrast, our method is provably never worse than the MC estimator despite that ours could result in rather coarse approximations of the function, limiting the error reduction in some regions. As future work, Crespo et al.’s adaptive model for $\hat{g}$ can potentially be useful in our formulation, but developing a practical least-squares regression algorithm for piecewise polynomials remains challenging.

6.5 Analysis

Choice of basis functions. We perform an experiment using different basis functions in the regression. The results are shown in Fig. 13. We compare the performance of order 2 polynomial, step functions, Gaussian mixture and Sine functions. All these basis have been chosen to contain the same number of parameters for a fair comparison. The quality of the results obtained slightly varies according to the bases used and the scenes chosen. Other results for additional bases and scenes is available in the supplemental materials.

In general, while it is challenging to define a function basis robust for all situations, our experiments show that polynomial basis works relatively well in many cases. As long as a constant function is included in least-squares regression, our method would be no worse than MC integration, no matter, what basis functions are chosen.

Impact of discontinuities on polynomial basis. Polynomial bases have difficulty in representing functions with many discontinuities. In Fig. 14, we study the evolution of the integration error for a fixed sample budget of 1024 samples when integrating the contribution of multiple light sources over a diffuse plane (Fig. 14.a).

In this case, the number of discontinuities inside the 3D integrand increases with the number of lights due to the random
Regression-based Monte Carlo Integration

Control variate function

Error images: Equal Time Rendering

RelMSE

Crespo et al. [2021]

Ours O2

(d) (e) (f)

Fig. 9. Equal-time comparison between our method and Crespo et al. [2021] where each pixel is considered independently. Columns (a)-(c) show the control variate function for a pixel centered at each inset. Columns (d)-(f) show the error images for each inset. Following Crespo et al., we ensure 30% of the samples are used for building the control variate. For both the dragon and the chopper-titan scene, our method show relatively less error ((d)-(f)). Crespo et al. is efficient in some cases. For dragon scene, column (e), Crespo et al. gives worse error than the classical MC. The relMSE value of our method is lower than Crespo et al.. For better visualization, please see the html viewer in the supplemental.

Fig. 10. Equal-sample comparison of our method using an O2 polynomial function basis with Crespo et al. [2021] method.

Computation time. We empirically found that the computation overhead of our method becomes significant for higher-order polynomial regression using the direct matrix approach. For example, our method with low-order polynomials has negligible overhead over MC integration (Fig. 3) but a polynomial of order 5 can cause 68% overhead in terms of running time. Similarly, higher-dimensional integration also requires significant overhead over MC integration. However, we focus in demonstrating

decision on the light selection (Fig. 14.b). We study the impact of using O1, O2 and O3 polynomials with our method against the conventional Monte Carlo estimator. The plot in Fig. 14(c)

Fig. 11. Importance sampling (IS) performed in Crespo et al.’s approach can increase the variance wrt classical MC due to bad IS strategy.
The results are demonstrated with 32spp and 256spp on a bathroom scene taken from PBRT. We use three different polynomials. The gradient descent approach scales well thanks to its lower computational complexity ($O(M^2)$) where $M$ is the number of monomials.

Matrix vs. Gradient descent solver. Fig. 15 shows the comparison between the gradient descent and the direct-matrix solver. The results are demonstrated with 32spp and 256spp on a bathroom scene taken from PBRT. We use three different polynomials of order 2, 5 and 7. We compare the direct matrix approach based on complete orthogonal decomposition and stochastic gradient descent estimator (SGD) to solve the polynomial regression. At a low sample count, SGD performs better than the direct matrix approach as the matrix tends to be ill-conditioned due to noise in sampling. At a high sample count, the direct matrix approach outperforms SGD due to its ability to find a global minimum. For a high-order polynomial (order 7), the direct matrix approach becomes computationally expensive whereas SGD runs at at least 2× faster speed while giving similar relMSE.

7 DISCUSSION

Incremental estimators. In our implementation, we assume that the $N$ sample estimator $(F)^+$ is constructed by first finding the solution to regression $\hat{g}$ using all the $N$ samples, and then evaluating $(F)^+$ with the same $N$ samples. Therefore, when $N$ is changed to $N+1$, we need to solve regression again with $N+1$ samples first and then re-evaluate $(F)^+$ with the new $\hat{g}$ using $N+1$ samples. In some applications, such as progressive rendering, being able to incrementally update both $(F)^+$ and $\hat{g}$ as we add a new sample is desirable. Thanks to the use of control variates, our formulation allows such incremental update as follows.

Let us denote $\langle F \rangle_N^+$ and $\hat{g}_N$ as the estimated value and the solution to regression with $N$ samples. Using an online regression algorithm, one can obtain $\hat{g}_{N+1}$ based on $\hat{g}_N$ by adding a new sample $u_{N+1}$. We define the incremental estimate $\langle F \rangle_{N+1}^+$ as

$$\langle F \rangle_{N+1}^+ = \frac{1}{N+1} \left( \hat{g}_{N+1} + \langle F \rangle_N^+ + \left( G_N + \hat{f}(u_{N+1}) - \hat{g}_N(u_{N+1}) \right) \right)$$

where $G_{N+1}$ is the integral of $\hat{g}_{N+1}$. This estimator remains unbiased since

$$E \left[ \langle F \rangle_{N+1}^+ \right] = E \left[ \frac{1}{N+1} \left( \hat{g}_{N+1} + \langle F \rangle_N^+ + \left( G_N + \hat{f}(u_{N+1}) - \hat{g}_N(u_{N+1}) \right) \right) \right]$$

$$= \frac{1}{N+1} \left( NF + E \left[ \left( G_N + \hat{f}(u_{N+1}) - \hat{g}_N(u_{N+1}) \right) \right] \right)$$

$$= \frac{1}{N+1} \left( NF + F \right) = F.$$ (35)

The expected squared error of incremental estimators is likely worse than that of non-incremental counterparts; the $N$ sample incremental estimator now combines the results of regression for all the $N-1$, $N-2$, ..., 1 samples which might be numerically unstable at the beginning. The incremental estimators are, however, often computationally less expensive for the same number of samples than the non-incremental counterparts. We leave the exact analysis of these differences as future work, but we show a preliminary result in Fig. 16 where the incremental estimate
Our regression-based Monte Carlo integration bears some similarities to Monte Carlo Least-Squares (MCLS) [Nakatsukasa 2018]. Nakatsukasa reformulated Monte Carlo integration as least-squares regression followed by the analytical integration of the resulting function. MCLS also performs no worse than Monte Carlo integration as long as a constant function is included in regression. There are two major differences between MCLS and our approach. Firstly, MCLS considers a model function \( g \) which integrates exactly to \( F \) (i.e., \( F = G \)).

Such a family of model functions is far more restrictive than our model function which is allowed to integrate to any value. Secondly, Nakatsukasa claims that MCLS is derived under an entirely different framework than control variates, thus its connection to control variates is unclear. In contrast, we show how control variates is closely related to the combination of Monte Carlo integration and least-squares regression.

When we use a model function \( \hat{g} \) such that \( G = F \) (i.e., keeps the integral value), then our approach in fact reduces to MCLS. MCLS is thus can be seen as a special case of our formulation.

**Differences from MCLS.** Our regression-based Monte Carlo integration with gradient descent estimator. The numerical error of the incremental estimate is slightly higher than the non-incremental version, but it still outperforms MC integration.

**Russian roulette.** In our current implementation, we manually set all path lengths to be the same. This is because, for regression it is necessary to fix the dimensionality of the search space. Russian roulette, however may result in paths of different length/dimensionality.

One simple way to combine Russian roulette with our approach would be to perform regression up to a certain prescribed dimension and only invoke Russian roulette for paths that have dimensionality higher than the prescribed one. The high-dimensional sample components can then be projected back into the lower-dimensional space.

### 8 LIMITATIONS AND FUTURE WORK

While our method often just works as a drop-in improvement over Monte Carlo integration in many cases, there are several limitations and future work.

Regression in our method incurs additional computation over MC integration. Regression may or may not be worth the additional cost depending on the amount of error reduction in the end. In general, regression becomes computationally more costly for a more complex model function, but a complex function typically leads to better error reduction. There is a trade-off among the reduction of error, the complexity of a model function, and the computational overhead of the regression process. Finding an optimal balance among those can potentially be achieved in an adaptive manner. Our formulation does not specify how regression should be solved in practice, so there is a large room of design choices in its implementation.

Since our formulation is based on the primary sample space, applying our method to a higher number of bounces in light transport simulation is challenging. The dimensionality of the primary sample space is directly proportional to the number of bounces, and regression in a high dimensional is not numerically stable. It is being recognized that light transport simulation has a much lower dimensional structure in the original path space, as was utilized by path guiding methods [Müller et al. 2017; Vorba et al. 2014]. It would be interesting to study how our regression-based formulation would benefit from such a low dimensional structure of light transport.

Depending on how the regression is implemented, our method can have a very small bias. When the same set of samples are used for both regression of the control variate and estimation of the difference, our estimator can be biased. Owen and Zhou [2000] and Hickernell et al. [2005] have analyzed the amount of bias introduced when estimating the \( \alpha \) parameter and evaluating...
the difference estimator using the same set of samples in control variates. In brief, the bias is negligible compared to the variance of the estimator and reduces at the order of \(O(N^{-1})\) for \(N\) samples as opposed to \(O(N^{-1/2})\) of the error of Monte Carlo integration. This makes our estimator consistent and the bias reduces faster than variance. One can also always remove this bias by separating a set of samples into two halves, performing regression only on the other half, and taking the average of the two difference estimators, in a manner akin to cross-validation. We, however, have not used this approach in the results in this paper since we have observed that bias is negligible. Our analysis is not influenced by this bias since this bias happens only when the same set of samples are used in its implementation of regression, while our analysis does not assume any specific implementation of regression.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their comments that helped shape the paper. We thank the following for their comments and ideas that helped shape the paper. We thank the following for their comments and ideas that helped shape the paper. We thank the following for their comments and ideas that helped shape the paper.

REFERENCES

Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. 2018. Integrating clipped spherical harmonics expansions. ACM Transactions on Graphics (TOG) 37, 2 (2018), 1–12.

Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A. Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. 2016. Nonlinearly Weighted First-Order Regression for Denoising Monte Carlo Renderings. 35, 4 (June 2016), 107–117. https://doi.org/10.4122/cr.2016.01.107

Petrík Clarberg and Tomas Akenine-Möller. 2008. Exploiting Visibility Correlation in Direct Illumination. 27, 4 (2008), 1125–1136. https://doi.org/10/d2gd6k

Fred J. Hickernell, Christiane Lemieux, and Art B. Owen. 2005. Control Variates using Piecewise-Polynomial Approximations. ACM Transactions on Graphics (TOG) 35, 4 (Oct. 2006), 169:1–169:12. https://doi.org/10.1145/1133516.1133517

Yuki Nakatsukasa. 2018. Approximate and integrate: Variance reduction in Monte Carlo integration via function approximation. arXiv:math.NA/1806.05492

ACM Trans. Graph., Vol. 41, No. 4, Article 79. Publication date: July 2022.