Econometric Theory, **17**, 2001, 933–961. Printed in the United States of America.

INTERPOLATION, QUADRATURE, AND STOCHASTIC INTEGRATION

LUNG-FEI LEE
Ohio State University

This paper considers features in numerical and stochastic integration approaches for the evaluation of analytically intractable integrals. It provides a unification of these two approaches. Some important features in quadrature formulations, namely, interpolation and region partition, can provide a valuable device for the design of a stochastic simulator. An interpolating function can be used as a valuable control variate for variance reduction in simulation. We illustrate possible variance reduction by some numerical cases with Gaussian quadrature. The resulting simulator may also be regarded as a monitor of the approximation error of a quadrature.

1. INTRODUCTION

The evaluation of an integral can be an important component in the estimation of some econometric models. Numerical integration problems have a long history in numerical analysis (see, e.g., Hildebrand, 1974). Recent advances in computing technology have rapidly promoted the development of simulation methods for stochastic integration. Stochastic integration techniques have provided computationally feasible approaches for high-dimensional integration problems. However, for a single or low-dimensional integration problem, it is a common view that numerical integration methods may be easier to implement and can be numerically efficient for some models.[1]

In the existing literature, numerical integration and stochastic integration are treated as separate and unrelated approaches. Numerical integration methods are deterministic as they involve neither random elements nor simulation. A numerical integration is based on numerical approximation, and its accuracy will depend on how close the approximation is. Many well-known numerical integration (quadrature) methods are derived by replacing the integrand with an interpolating polynomial, and the unknown integral is approximated by the integral of the interpolating polynomial. For numerical integration methods, accuracy of an approximated integral may depend on an integrand's smoothness and the possibility of its being adequately approximated by a polynomial. The

This research was supported by a Research Grant Council of Hong Kong under research grant HKUST595/96H to the Hong Kong University of Science and Technology. I appreciate valuable comments from an anonymous referee for improving the presentation of this paper. Address correspondence to: Lung-fei Lee, Department of Economics, Ohio State University, 410 Arps Hall, 1945 N. High Street, Columbus, OH 43210-1172, USA; e-mail: lflee@econ.ohio-state.edu.

© 2001 Cambridge University Press 0266-4666/01 $9.50

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

error of a numerical integral cannot, in general, be evaluated exactly. The best one can do is to provide error bounds on the approximation error (see, e.g., Hildebrand, 1974; Geweke, 1996). The cornerstone of a stochastic integral is the law of large numbers. A stochastic integral is usually an unbiased simulator of the unknown integral, in the sense that the error of approximation is stochastic and has a zero mean.[2] A stochastic integral can be an importance sampler (IS).[3] In the IS framework, various sampling densities can be used. A desirable IS is supposed to be easily implemented, and its associated simulator may have small simulation variance (Geweke, 1989).

As both the quadrature and stochastic integration approaches share the common goal of evaluating an integral, some features of one approach may be valuable to the other. In this paper, we investigate the possible use of numerical integration techniques to improve the design of an IS simulator. Coincidentally, stochastic simulation techniques can supplement a numerical integration approach in the evaluation of approximation error. For concise exposition, we present our ideas for the evaluation of a single dimensional integral. Those ideas might also be valuable for multidimensional integral problems. For numerical integration approaches, we will focus our attention on the Gaussian quadrature and Newton–Cotes formulas. In econometrics, trapezoidal integration, which is a member of the Newton–Cotes formulas, has been used in Heckman and Willis (1975), and the Gaussian quadrature formula has been popularized by Butler and Moffitt (1982).

2. GAUSSIAN QUADRATURE AND STOCHASTIC SIMULATION

Consider the evaluation of a unidimensional integral

$$I = \int_a^b f(x)dx, \tag{2.1}$$

where the limits a and b can be finite or infinite.

2.1. Gaussian Quadrature: A Review

In the Gaussian quadrature approach, a basis function $w(x)$ depending on the range of integration will first be chosen. Let $r(x) = f(x)/w(x)$. The integral (2.1) can then be written as

$$I = \int_a^b r(x)w(x)dx. \tag{2.1$'$}$$

The Gaussian quadrature proceeds as follows. Given the chosen $w(x)$, there exist points $x_i \in [a, b]$, $i = 1, \ldots, m$ (for an m-point quadrature) and corresponding weights ω_i associated with x_i such that I in (2.1)$'$ can be approximated by

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

$$I_m = \sum_{i=1}^{m} \omega_i r(x_i). \tag{2.2}$$

A popular basis function is the Gauss–Legendre rule when $[a, b]$ is a finite interval, the generalized Laguerre rule when $[a, b]$ is an interval with either a or b being infinite, and the Hermite rule when both a and b are infinite. Numerical tables on Gaussian points and weights are available from Stroud and Secrest (1966) and Abramowitz and Stegun (1964). Press, Teukolsky, Vetterling, and Flannery (1992) provide computing codes for generating quadrature points and weights for a given basis function, which can even be a nonclassical one.

The Gaussian quadrature formulation is based on the theory of orthogonal polynomials. Popular basis functions and their associated orthogonal polynomials can be found in standard numerical analysis books (see, e.g., Hildebrand, 1974). The Gaussian quadrature formula with m points provides an exact integral I when $r(x)$ in (2.1)′ happens to be a polynomial of degree $2m - 1$. If $r(x)$ is $2m$-times differentiable, the error E_m of an m-point quadrature approximation to I can be expressed as

$$E_m = I - I_m = c_m r^{(2m)}(\xi) \tag{2.3}$$

for some $\xi \in [a, b]$, where $r^{(2m)}(x)$ denotes the $2m$-order derivative of $r(x)$ and $\{c_m\}$ is a sequence of constants such that $\lim_{m\to\infty} c_m = 0$. For example, if $w(x) = 1$ for $a = -1$ and $b = 1$, $c_m = 2^{2m+1}(m!)^4/\{(2m + 1)!(2m!)^3\}$ (see Geweke, 1996, p. 735). The sophistication of the Gaussian quadrature is in its strategical selection of locations at which the function $r(x)$ is to be evaluated.

For our purpose to unify quadrature and stochastic integration, it is helpful to review underlying interpolating polynomials used for approximating $r(x)$ in a Gaussian quadrature. Let x_i, $i = 1, \ldots, m$, be m chosen locations and let $(x_i, r(x_i))$ be the corresponding points on the graph of $r(x)$. These m points determine a unique interpolating polynomial $P_{m-1}(x)$ of maximum degree $(m - 1)$. This polynomial in its Lagrangian form is

$$P_{m-1}(x) = \sum_{i=1}^{m} r(x_i) \frac{(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_m)}{(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_m)}. \tag{2.4}$$

The $P_{m-1}(x)$ coincides with $r(x)$ at x_i for $i = 1, \ldots, m$. Given a basis function $w(x)$, a system of orthogonal polynomials (with the first one being a unit constant) will be determined. The zeros of an orthogonal polynomial are real and distinct (Hildebrand, 1974). In a Gaussian quadrature, the locations x_i, $i = 1, \ldots, m$, are selected to be the zeros (Gaussian points) of the orthogonal polynomial of degree m. Because the zeros x_i, $i = 1, \ldots, m$, are distinct, there exists a unique set of weights ω_i, $i = 1, \ldots, m$, such that

$$\int_a^b P_{m-1}(x) w(x) dx = \sum_{i=1}^{m} \omega_i P_{m-1}(x_i) = \sum_{i=1}^{m} \omega_i r(x_i). \tag{2.5}$$

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

These weights can be solved by the method of undetermined coefficients from the following linear system:

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_m \\ \vdots & \vdots & & \vdots \\ x_1^{m-1} & x_2^{m-1} & \dots & x_m^{m-1} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_m \end{pmatrix} = \begin{pmatrix} \int_a^b w(x)dx \\ \int_a^b xw(x)dx \\ \vdots \\ \int_a^b x^{m-1}w(x)dx \end{pmatrix}.$$

As x_i, $i = 1, \dots, m$, are the zeros of the orthogonal polynomial of degree m, it can be shown that when $r(x)$ is a polynomial of maximum degree $(2m - 1)$, the Gaussian quadrature formula provides the exact integral I. If $r(x)$ in (2.1)′ is not a polynomial, there will be a nonvanishing error E_m of the m-point Gaussian quadrature approximation to I, i.e.,

$$\int_a^b r(x)w(x)dx = \sum_{i=1}^{m} \omega_i r(x_i) + E_m. \tag{2.6}$$

This error term has an unknown value because I is unknown. In practice, error criteria are usually specified as the absolute or relative difference in the computed approximation to I using an n-point and an m-point quadrature (Kahaner, Moler, and Nash, 1989, Ch. 5). An m-point quadrature may be selected when its difference with another quadrature with a larger number of Gaussian points satisfies prescribed convergence or error criteria. Such error criteria will depend on $E_m - E_n$ for $n > m$ instead of E_m. The relative error $E_m - E_n = I_n - I_m$ is readily available, but the individual absolute error $|E_m|$ is not. It will be valuable if the absolute error can somehow be monitored.

2.2. A Natural Simulator

Instead of quadrature, an alternative approach to provide an estimate of I is to use a sample average based on stochastic simulation. The resulting estimate is a stochastic integral of I. With an expression such as the one in (2.1)′, a simulator can naturally be derived. Because the basis function $w(x)$ in (2.1)′ is a positive function on (a, b), it can be turned into a density function $w^*(x)$ with support (a, b) as

$$w^*(x) = w(x) \Big/ \int_a^b w(z)dz. \tag{2.7}$$

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

Let $d = \int_a^b w(z)dz$. Then (2.1)′ can be rewritten as

$$I = d \int r(x)w^*(x)dx. \tag{2.8}$$

The term $w^*(x)$ can be used as a density for random draws.[4] With R random draws $x^{(r)}$, $r = 1,\ldots,R$, from $w^*(x)$, the simulator

$$\hat{I}_R = \frac{d}{R}\sum_{r=1}^{R} r(x^{(r)}) \tag{2.9}$$

is an unbiased estimator of I. The accuracy of a simulator can be accessed by its Monte Carlo (MC) variance. The MC variance of $\hat{I}_R$ in (2.9) is

$$\text{var}(\hat{I}_R) = \frac{1}{R}\int_a^b [d\cdot r(x) - I]^2 w^*(x)dx. \tag{2.10}$$

2.3. A Hybrid Approach

The quadrature I_m in (2.2) and the simulator $\hat{I}_R$ in (2.9) are not comparable. Each has its own advantage and disadvantage. The I_m can provide the exact integral I when $r(x)$ is a polynomial of maximum degree $(2m - 1)$. On the other hand, the simulator $\hat{I}_R$ would have a nonzero MC variance even if $r(x)$ were a polynomial. If $r(x)$ is not a polynomial, a quadrature approximation will have a nonzero error. A larger number of Gaussian points will not necessarily reduce a quadrature's approximation error as high order approximation is not necessarily the same as high accuracy unless the integrand is very smooth (Press et al., 1992, p. 140). On the contrary, the MC variance of the simulator $\hat{I}_R$ can always be reduced by increasing the number of random draws R at the expense of increasing computing time. Instead of regarding the two approaches as competitors, it would be of interest to derive a hybrid procedure that possesses the advantages of both.

The approximation error E_m of an m-point quadrature in (2.6) can be rewritten as

$$\begin{aligned} E_m &= \int_a^b r(x)w(x)dx - \sum_{i=1}^{m}\omega_i r(x_i) \\ &= \int_a^b r(x)w(x)dx - \int_a^b P_{m-1}(x)w(x)dx \\ &= \int_a^b d[r(x) - P_{m-1}(x)]w^*(x)dx \end{aligned} \tag{2.11}$$

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

by using (2.5), where $P_{m-1}(x)$ is in (2.4). The term E_m can be estimated by an unbiased simulator $\hat{E}_m^{(R)}$ where

$$\hat{E}_m^{(R)} = \frac{1}{R}\sum_{r=1}^{R} d[r(x^{(r)}) - P_{m-1}(x^{(r)})] \tag{2.12}$$

and $x^{(r)}$, $r = 1,\ldots,R$, are random draws from $w^*(x)$. This simulator provides a possible measure of the approximation error E_m of the quadrature formula I_m using $P_{m-1}(x)$ as the interpolating polynomial for $r(x)$. In turn, (2.12) provides an unbiased simulator

$$\hat{I}_h^{(R)} = I_m + \hat{E}_m^{(R)} \tag{2.13}$$

of I. The simulator $\hat{I}_h^{(R)}$ can be compared with $\hat{I}_R$ in (2.9). The MC variance of $\hat{I}_R$ is $\mathrm{Var}(\hat{I}_R) = (1/R)V$ where

$$\begin{aligned} V &= \int_a^b [d\cdot r(x) - I]^2 w^*(x)dx \\ &= \int_a^b [d\cdot r(x) - (I_m + E_m)]^2 w^*(x)dx \\ &= \int_a^b [d\cdot r(x) - I_m]^2 w^*(x)dx - E_m^2 \end{aligned} \tag{2.14}$$

is the corresponding MC variance of a single draw. The last equality in (2.14) holds because $\int_a^b (d\cdot r(x) - I_m)w^*(x)dx = E_m$ from (2.11). As I_m is deterministic, the MC variance of $\hat{I}_h^{(R)}$ is $\mathrm{Var}(\hat{I}_h^{(R)}) = \mathrm{Var}(\hat{E}_m^{(R)}) = (1/R)V_m$ where

$$\begin{aligned} V_m &= \int_a^b \{d[r(x) - P_{m-1}(x)] - E_m\}^2 w^*(x)dx \\ &= S_m - E_m^2, \end{aligned} \tag{2.15}$$

where

$$\begin{aligned} S_m &= \int_a^b \{d[r(x) - P_{m-1}(x)]\}^2 w^*(x)dx \\ &= \int_a^b \{[d\cdot r(x) - I_m] - [d\cdot P_{m-1}(x) - I_m]\}^2 w^*(x)dx \\ &= \int_a^b [d\cdot r(x) - I_m]^2 w^*(x)dx + \int_a^b [d\cdot P_{m-1}(x) - I_m]^2 w^*(x)dx - 2C_m \end{aligned} \tag{2.16}$$

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

and

$$C_m = \int_a^b [d\cdot r(x) - I_m][d\cdot P_{m-1}(x) - I_m]w^*(x)dx$$
$$= \int_a^b [d\cdot r(x) - I][d\cdot P_{m-1}(x) - I_m]w^*(x)dx$$

is the covariance of $d\cdot r(x)$ and $d\cdot P_{m-1}(x)$ for x with density $w^*(x)$ because $\int_a^b P_{m-1}(x)w(x)dx = I_m$. It follows from (2.14)–(2.16) that if C_m is larger than half of the variance of $d\cdot P_{m-1}(x)$, i.e.,

$$C_m > \frac{1}{2}\int_a^b [d\cdot P_{m-1}(x) - I_m]^2 w^*(x)dx, \tag{2.17}$$

V_m of (2.15) will be smaller than V of (2.14). That is, under the condition (2.17), the simulator $\hat{I}_h^{(R)}$ can have a smaller MC variance than that of $\hat{I}_R$. Condition (2.17) is likely to be satisfied if $P_{m-1}(x)$ is an adequate interpolating polynomial of $r(x)$ so that C_m is approximately equal to the variance of $d\cdot P_{m-1}(x)$ under $w^*(x)$. Condition (2.17) is similar to the one emphasized in Hendry (1984) for the use of control variates (for reduction of intraexperimental variation in MC experimentation). We will have more to offer on the use of control variates for variance reduction in subsequent paragraphs.

2.4. Interpolation, Quadrature, and Control Variates

For simulation estimation of an integral, the method of control variates may reduce the MC variance of an unbiased simulator (see Hammersley and Handscomb, 1964; Lavenberg and Welch, 1981; Ripley, 1987; Davidson and MacKinnon, 1992; Geweke, 1996).[5] Consider the simulator $\hat{I}_R$ in (2.9) where $x^{(r)}$, $r = 1,\ldots,R$, are random draws from $w^*(x)$. Suppose that there exists a vector-valued function h such that H, where $H = \int_a^b h(x)w^*(x)dx$, is a known vector. Then $H_R = (1/R)\sum_{r=1}^R h(x^{(r)})$ is an unbiased simulator of H. Then a simulator of the form

$$\hat{I}_c^{(R)}(\beta) = \hat{I}_R - \beta'(H_R - H) \tag{2.18}$$

for any constant vector β is also an unbiased simulator of I because $H_R - H$ has a zero mean. As H is known, the computation of $\hat{I}_c^{(R)}(\beta)$ is feasible for any β. The MC variance of $\hat{I}_c^{(R)}(\beta)$ is

$$\mathrm{var}(\hat{I}_c^{(R)}(\beta)) = \frac{1}{R}\int [d\cdot r(x) - \beta'(h(x) - H) - I]^2 w^*(x)dx$$
$$= \frac{1}{R}\left\{\int [d\cdot r(x) - \beta'(h(x) - H)]^2 w^*(x)dx - I^2\right\}.$$

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

The optimum simulator with the smallest MC variance within the class of simulators in (2.18) is obtained at

$$\beta^* = [\mathrm{Var}(H_R)]^{-1}\,\mathrm{Cov}(H_R, \hat{I}_R)$$
$$= \left[\int (h(x) - H)(h(x) - H)' w^*(x)dx\right]^{-1} \int (h(x) - H)(d \cdot r(x)) w^*(x)dx, \quad \textbf{(2.19)}$$

as the MC variance minimization with β is a least-squares problem. The MC variance of $\hat{I}_c^{(R)}(\beta^*)$ is

$$\mathrm{Var}(\hat{I}_c^{(R)}(\beta^*)) = \mathrm{Var}(\hat{I}_R) - \mathrm{Cov}(H_R, \hat{I}_R)'[\mathrm{Var}(H_R)]^{-1}\,\mathrm{Cov}(H_R, \hat{I}_R), \quad \textbf{(2.20)}$$

which is smaller than the MC variance of $\hat{I}_R$ whenever $\mathrm{Cov}(H_R, \hat{I}_R) \neq 0$. The control variates H_R are valuable if they are highly correlated with $\hat{I}_R$.

In practice, β^* is not known but can be estimated by simulation (Ross, 1990, pp. 120–125). The term β^* can either be estimated by simulation with sample split or estimated simultaneously with I with common random draws. These methods can be accomplished by the method of least squares. With R number of random draws, define $y_R = d \cdot (r(x^{(1)}), \ldots, r(x^{(R)}))'$, $l_R = (1, \ldots, 1)'$, and $Z_R = (h(x^{(1)}) - H, \ldots, h(x^{(R)}) - H)'$. The estimation procedure that can be used is simply a linear regression method by regressing y_R on the constant vector l_R and the control variate matrix Z_R. With common random draws, the control-variate simulator of I can be the least-square estimate of the intercept term, i.e.,

$$\hat{I}_c^{(R)} = (l_R'[J_R - Z_R(Z_R'Z_R)^{-1}Z_R']l_R)^{-1} l_R'[J_R - Z_R(Z_R'Z_R)^{-1}Z_R']y_R, \quad \textbf{(2.21)}$$

where J_R is the R-dimensional identity matrix (Davidson and MacKinnon, 1992). Let $\hat{\beta}^*$ be the least-square estimate of the coefficient vector of Z_R. With sample split, additional random draws are taken independently from those used in the estimation of β^*. Those additional independent draws are used for the construction of $\hat{I}_R$ and H_R in (2.18).[6] A feasible control simulator is derived with $\hat{\beta}^*$ replacing β in (2.18) as

$$\hat{I}_c^{(R)}(\hat{\beta}^*) = \hat{I}_R - \hat{\beta}^{*\prime}(H_R - H). \quad \textbf{(2.22)}$$

An estimated $\hat{\beta}^*$ introduces extra stochastic variance in the final simulator. Ripley (1987) provides results on the variances of both estimates with and without sample split under the assumption that all the random variables (including control variates) are normally distributed (see also Lavenberg and Welch, 1981). The conclusion is that, as long as the number of control variates used, and hence the number of unknown coefficients in β, is not large relative to the number of random draws for its estimation, the gain in variance reduction, in principle, can be realized with an estimated $\hat{\beta}^*$. Davidson and MacKinnon (1992) provide a valuable insight for the general case without assuming control variates

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

being normally distributed. They point out that, with common random draws, in addition to the regression estimate of the constant term providing a control simulator, the standard error from the regression output provides a consistent estimate of the MC variance of the simulator (for details, see Lavenberg and Welch, 1981; Davidson and MacKinnon, 1992).

The method of control variates is useful only if control variates that are highly correlated with $\hat{I}_R$ can be found. For our problem, the interpolating polynomial $P_{m-1}(x)$ in (2.4) suggests a relevant control variate for possible improvement upon the simulator $\hat{I}_R$ in (2.9). The variate that can be used may be $h(x) = d \cdot P_{m-1}(x)$. The mean of this variate is $\int_a^b d \cdot P_{m-1}(x) w^*(x)\, dx = \int_a^b P_{m-1}(x) w(x) dx = I_m$ from (2.5), which is known. As $P_{m-1}(x)$ is an interpolating polynomial for $r(x)$, $P_{m-1}(x)$ and $r(x)$ with x being a random variable can be correlated. Therefore, the interpolating polynomial can be a useful control variate.[7] Our subsequent MC results will provide numerical evidence to illustrate this point.

With $h(x) = d \cdot P_{m-1}(x)$ as a control variate, the control simulator in (2.18) is related to the hybrid simulator in (2.13). The control simulator in (2.18) can be rewritten as

$$\hat{I}_c^{(R)}(\beta) = \frac{1}{R} \sum_{r=1}^{R} [d \cdot r(x^{(r)}) - \beta (d \cdot P_{m-1}(x^{(r)}) - I_m)]. \tag{2.23}$$

On the other hand, the hybrid simulator in (2.13) can be rewritten as

$$\hat{I}_h^{(R)} = \frac{1}{R} \sum_{r=1}^{R} [d \cdot r(x^{(r)}) - (d \cdot P_{m-1}(x^{(r)}) - I_m)]. \tag{2.24}$$

The hybrid simulator is a member of the class of control variate simulators with β restricted to be one.

3. NEWTON–COTES NUMERICAL INTEGRATION AND STOCHASTIC SIMULATION

When both a and b are finite, Newton–Cotes quadrature formulas are popular methods of numerical integration of $I = \int_a^b f(x) dx$ in (2.1). The well-known trapezoid rule, Simpson's $\frac{1}{3}$ rule, and Newton's $\frac{3}{8}$ rule belong to the Newton–Cotes (closed) class of numerical integration methods. Suppose that $[a, b]$ is divided into n subintervals of equal length. Let x_i, $i = 0, \ldots, n$, be the evenly spaced values of $[a, b]$ with subinterval length $h = x_{i+1} - x_i$ where $x_0 = a$ and $x_n = b$. The Newton–Cotes quadrature formula Q_m is derived by approximating $f(x)$ by interpolating polynomials of degree m over each set of m subintervals of $[a, b]$ and computing the integral of the interpolating polynomial over its interval of definition.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

The Newton–Cotes quadrature formula Q_m has the form $Q_m = \sum_{i=0}^{n} A_{mi} f(x_i)$ for some A's. The trapezoid rule Q_1 is derived by approximating $f(x)$ by the linear interpolating polynomial $P_{1i}(x)$ in each subinterval (x_i, x_{i+1}) where

$$P_{1i}(x) = f(x_i) + \frac{(f(x_{i+1}) - f(x_i))}{h}(x - x_i), \qquad x \in (x_i, x_{i+1}). \tag{3.1}$$

This interpolating polynomial gives an integral q_{1i} over (x_i, x_{i+1}) where $q_{1i} = \int_{x_i}^{x_{i+1}} P_{1i}(x)dx = (h/2)[f(x_i) + f(x_{i+1})]$. The q_{1i} provides an approximation to $I_{1i} = \int_{x_i}^{x_{i+1}} f(x)dx$. Because $f(x)$ in (2.1) is approximated by $P_{1i}(x)$ in (x_i, x_{i+1}) for each $i = 0,1,\ldots,n-1$, $I = \sum_{i=0}^{n-1} I_{1i}$ is approximated by

$$Q_1 = \sum_{i=0}^{n-1} q_{1i} = \frac{h}{2}[f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)], \tag{3.2}$$

which is the form of the trapezoid rule.

The Simpson's $\frac{1}{3}$ rule requires that n be even. It is derived by approximating $f(x)$ by a second-degree interpolating polynomial $P_{2i}(x)$ over each of the subinterval pairs (x_{2i}, x_{2i+2}), i.e., over (x_0, x_2), (x_2, x_4), (x_4, x_6), etc., where

$$P_{2i}(x) = f(x_{2i}) + \frac{\Delta f_{2i}}{h}(x - x_{2i}) + \frac{\Delta^2 f_{2i}}{2h^2}(x - x_{2i})(x - x_{2i+1}), \tag{3.3}$$

with $\Delta f_{2i} = f(x_{2i+1}) - f(x_{2i})$ and $\Delta^2 f_{2i} = f(x_{2i+2}) - 2f(x_{2i+1}) + f(x_{2i})$, is a second-degree interpolating polynomial expressed in the Newtonian forward form. Over the subinterval pair (x_{2i}, x_{2i+2}), the integral $I_{2i} = \int_{x_{2i}}^{x_{2i+2}} f(x)dx$ can be approximated by

$$q_{2i} = \int_{x_{2i}}^{x_{2i+2}} P_{2i}(x)dx = \frac{h}{3}[f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})]. \tag{3.4}$$

The I is then approximated by

$$Q_2 = \sum_{i=0}^{n/2-1} q_{2i} = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 4f(x_{n-1}) + f(x_n)], \tag{3.5}$$

which is Simpson's $\frac{1}{3}$ rule. Higher-order Newton–Cotes quadrature formulas can similarly be derived with high order interpolating polynomials (see, e.g., Press et al., 1992). The error of approximation of I by the Newton–Cotes formula Q_m is $E_m = I - Q_m$.

The interpolating polynomial in each interval (x_i, x_{i+1}) of the Newton–Cotes formula suggests valuable control variates for simulation and the construction of stochastic integrals. Consider, e.g., Simpson's $\frac{1}{3}$ rule with $P_{2i}(x)$ in

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

(3.3). This $P_{2i}(x)$ in (3.3) passes through the points $(x_{2i}, f(x_{2i}))$, $(x_{2i+1}, f(x_{2i+1}))$, and $(x_{2i+2}, f(x_{2i+2}))$. The error of the approximation of I_{2i} by q_{2i} in (3.4) is $E_{2i} = I_{2i} - q_{2i} = I_{2i} - \int_{x_{2i}}^{x_{2i+2}} P_{2i}(x)dx$. Hence

$$E_{2i} = \int_{x_{2i}}^{x_{2i+2}} [\, f(x) - P_{2i}(x)]\, dx. \tag{3.6}$$

The overall approximation error of I by Q_2 will be

$$E_2 = \sum_{i=0}^{n/2-1} E_{2i} = \int_a^b [\, f(x) - P_2(x)]dx, \tag{3.7}$$

where $P_2(x) = \sum_{i=0}^{n/2-1} P_{2i}(x)1_{[x_{2i}, x_{2i+2}]}(x)$ and $1_S(x)$ denotes the set indicator of a set S. The error E_2 can be evaluated by simulation. The density $1^*_{[a,b]}(x) = 1_{[a,b]}(x)/(b-a)$ is uniform with support $[a,b]$. As

$$E_2 = (b-a)\int_a^b [\, f(x) - P_2(x)]1^*_{[a,b]}(x)dx, \tag{3.7'}$$

random variables $x^{(r)}$ can be drawn from $1^*_{[a,b]}(x)$ and E_2 can be estimated by

$$\hat{E}_2^{(R)} = \frac{(b-a)}{R}\sum_{r=1}^{R} [\, f(x^{(r)}) - P_2(x^{(r)})]. \tag{3.8}$$

With (3.5) and (3.7), $Q_2 + \hat{E}_2^{(R)}$ provides a (hybrid) unbiased simulator for I. The method of control variates similar to that described in the previous section can be applied to derive simulators with smaller MC variances.

In addition to the method of control variates, better simulators can be further developed. As the overall interval has been partitioned into subintervals, simulation can be conducted separately for functions at each subinterval. At $[x_{2i}, x_{2i+2}]$, E_{2i} in (3.6) can be rewritten as

$$E_{2i} = (x_{2i+2} - x_{2i})\int_{x_{2i}}^{x_{2i+2}} [\, f(x) - P_{2i}(x)]1^*_{[x_{2i}, x_{2i+2}]}(x)dx. \tag{3.9}$$

Random variables $x_i^{(r)}$, $r = 1, \ldots, R$, can be drawn from $1^*_{[x_{2i}, x_{2i+2}]}(x)$, and E_{2i} can be estimated by the simulator

$$\tilde{E}_{2i}^{(R)} = \frac{(x_{2i+2} - x_{2i})}{R}\sum_{i=1}^{R} [\, f(x_i^{(r)}) - P_{2i}(x_i^{(r)})]. \tag{3.10}$$

An overall unbiased simulator for E_2 in (3.7) is $\tilde{E}_2^{(R)} = \sum_{i=1}^{n/2-1} \tilde{E}_{2i}^{(R)}$. For the simulation of I_{2i} by the method of control variates, the control variate can be

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

$h_i(x) = (x_{2i+2} - x_{2i})P_{2i}(x) - q_{2i}$. The efficiency of simulators developed by partition will be shown in the next section.

4. PARTITION AND MC VARIANCE REDUCTION

Consider an integral

$$\mu = \int_{\Omega} h(x)p(x)dx,$$

where Ω is a subset of a finite-dimensional Euclidean space and $p(x)$ is a density function with support Ω. The original integration problem may be $\mu = \int_{\Omega} f(x)dx$ written in the IS framework. With $p(x)$ as an IS density, random variates $x^{(r)}$, $r = 1,\ldots,R$, can be generated with density $p(x)$ and μ can be estimated by an unbiased simulator as

$$\hat{\mu}^{(R)} = \frac{1}{R}\sum_{r=1}^{R} h(x^{(r)}). \tag{4.1}$$

A partition will divide Ω into a finite, exhaustive, and mutually exclusive collection of subsets C_k, $k = 1,\ldots,K$, such that $\cup_{k=1}^{K} C_k = \Omega$ and $C_j \cap C_k = 0$ when $j \neq k$. With a partition, $\int_{\Omega} h(x)p(x)dx = \sum_{k=1}^{K} \mu_k$ where

$$\mu_k = \int_{C_k} h(x)p(x)dx. \tag{4.2}$$

Based on a partition $\{C_k\}_{k=1}^{K}$ of Ω, an alternative simulator of μ can be derived by separate simulation estimation for each μ_i, $i = 1,\ldots,K$. On C_k, define a truncated density

$$p_{C_k}(x) = p(x)/P(C_k) \tag{4.3}$$

with support C_k. In this setting, we assume that $P(C_k)$'s can be computed and random variables can be drawn from p_{C_k}'s. The μ_k in (4.2) can be rewritten as

$$\mu_k = P(C_k)\int_{C_k} h(x)p_{C_k}(x)dx, \tag{4.2'}$$

which can be simulated by

$$\hat{\mu}_k^{(R_k)} = \frac{P(C_k)}{R_k}\sum_{r=1}^{R_k} h(x_k^{(r)}), \tag{4.4}$$

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

where $x_k^{(r)}$, $r = 1, \ldots, R_k$, are R_k random draws from $p_{C_k}(x)$. Instead of (4.1), an alternative simulator of μ is

$$\hat{\mu}^{(R_1,\ldots,R_K)} = \sum_{k=1}^{K} \hat{\mu}_k^{(R_k)}. \tag{4.5}$$

It will be shown that $\hat{\mu}^{(R_1,\ldots,R_K)}$ can have smaller MC variance relative to $\hat{\mu}^{(R)}$ for some selection of $R_1, \ldots, R_K$ with a total amount of simulation R, i.e., $\sum_{k=1}^{K} R_k = R$.[8]

The MC variance of $\hat{\mu}_k^{(R_k)}$ in (4.4) is

$$V_k^{(R_k)} = \frac{1}{R_k}\left[P^2(C_k)\int_{C_k} h^2(x)p_{C_k}(x)dx - \mu_k^2\right]. \tag{4.6}$$

It follows that the MC variance of $\hat{\mu}^{(R_1,\ldots,R_K)}$ in (4.5) is

$$V_K^{(R_1,\ldots,R_K)} = \sum_{k=1}^{K} V_k^{(R_k)} = \sum_{k=1}^{K} \frac{1}{R_k}\left[P^2(C_k)\int_{C_k} h^2(x)p_{C_k}(x)dx - \mu_k^2\right]. \tag{4.7}$$

For the simulator $\hat{\mu}^{(R)}$ in (4.1), its MC variance is

$$V^{(R)} = \frac{1}{R}\int_{\Omega} [h(x) - \mu]^2 p(x)dx,$$

which can be rewritten into

$$\begin{aligned} V^{(R)} &= \frac{1}{R}\left\{\int_{\Omega}\left[\sum_{k=1}^{K} 1_{C_k}(x)h(x)\right]^2 p(x)dx - \mu^2\right\} \\ &= \frac{1}{R}\left[\sum_{k=1}^{K}\int_{C_k} h^2(x)p(x)dx - \left(\sum_{k=1}^{K}\mu_k\right)^2\right]. \end{aligned} \tag{4.8}$$

For comparison, it is sensible to require the total number of random draws used in (4.1) and (4.5) to be the same. Thus, consider the case that $\sum_{k=1}^{K} R_k = R$. Note that (4.7) can be rewritten into

$$\begin{aligned} V_K^{(R_1,\ldots,R_K)} &= \frac{1}{R}\sum_{k=1}^{K}\left[\frac{R}{R_k}P^2(C_k)\int_{C_k} h^2(x)p_{C_k}(x)dx - \frac{R}{R_k}\mu_k^2\right] \\ &= \frac{1}{R}\sum_{k=1}^{K}\left[\frac{R}{R_k}P(C_k)\int_{C_k} h^2(x)p(x)dx - \frac{R}{R_k}\mu_k^2\right] \end{aligned} \tag{4.9}$$

by using (4.3).

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

There is freedom in the design of $R_1,\ldots,R_K$ under the condition $\sum_{k=1}^{K} R_k = R$ for a simulator in (4.5). The optimum design can be obtained by minimizing the total variance in (4.7) with $R_1,\ldots,R_K$ subject to the constraint $\sum_{k=1}^{K} R_k = R$. The optimum solution has $R_k^* = RV_k^{1/2}/[\sum_{j=1}^{K} V_j^{1/2}]$ where $V_k = V_k^{(1)}$ is the MC variance in (4.6) with a single draw. With the optimum R_k^*'s, the MC variance of $\hat{\mu}^{(R_1^*,\ldots,R_K^*)}$ is $V_K^{(R_1^*,\ldots,R_K^*)} = (1/R)[\sum_{k=1}^{K} V_k^{1/2}]^2$. Unfortunately, the optimum design might not be readily available because it involves unknown moments $\int_{C_k} h^2(x)p(x)dx$. Instead of the optimum design, a practical one may be a balanced design in that R_k's are chosen such that

$$\frac{R}{R_j} P(C_j) = \frac{R}{R_k} P(C_k), \quad \text{for all } j \text{ and } k; \quad \text{and} \quad \sum_{k=1}^{K} R_k = R. \tag{4.10}$$

The solution of a balanced design in (4.10) for k is simply

$$R_k = R \cdot P(C_k). \tag{4.11}$$

The balanced design is feasible as $P(C_k)$, $k = 1,\ldots,K$, are known for a partition. For special cases where $P(C_k)$'s are equal for all $k = 1,\ldots,K$, R_k in (4.11) is simply R/K, i.e., an equal amount of draws for each subset of the partition. For the Newton–Cotes (closed) quadrature, this is indeed the case. With a balanced design in (4.11), the corresponding MC variance in (4.9) is

$$V_{K,b}^{(R)} = \frac{1}{R} \sum_{k=1}^{K} \left[\int_{C_k} h^2(x)p(x)dx - \frac{R}{R_k} \mu_k^2 \right]. \tag{4.12}$$

Comparing it with the one in (4.8), one has

$$\begin{aligned} V^{(R)} - V_{K,b}^{(R)} &= \frac{1}{R} \left[\sum_{k=1}^{K} \frac{R}{R_k} \mu_k^2 - \left(\sum_{k=1}^{K} \mu_k \right)^2 \right] \\ &= \frac{1}{R} \left\{ \sum_{k=1}^{K} q_k \left(\frac{\mu_k}{q_k} \right)^2 - \left[\sum_{k=1}^{K} q_k \left(\frac{\mu_k}{q_k} \right) \right]^2 \right\}, \end{aligned} \tag{4.13}$$

where $q_k = R_k/R$. The difference in (4.13) is nonnegative by the identity $\sum_{k=1}^{K} x_k^2 p_k - (\sum_{k=1}^{K} x_k p_k)^2 = \sum_{k=1}^{K} (x_k - \sum_{j=1}^{K} x_j p_j)^2 p_k$, where p_k's are probability weights, i.e., $p_k \geq 0$ for all $k = 1,\ldots,K$ and $\sum_{k=1}^{K} p_k = 1$. Therefore, we conclude that, at least with the balanced design, the MC variance of a simulator with partition in (4.5) can be smaller than that of a corresponding simulator in (4.1) without the partition.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

5. SOME NUMERICAL ILLUSTRATIONS

5.1. A Single Integral

To illustrate the various methods and potential gain in using interpolating functions as control variates for variance reduction, consider the following unknown integral:

$$\int_{-\infty}^{\infty} \prod_{i=1}^{T} \Phi\left[D_i\left(\frac{x_i\beta + \rho^{1/2}w}{(1-\rho)^{1/2}}\right)\right]\phi(w)dw, \tag{5.1}$$

where Φ and ϕ denote, respectively, the standard normal probability and density functions. This integral is motivated by an integration problem of a random-component probit model of Butler and Moffitt (1982) and Borjas and Sueyoshi (1994). A random-component dichotomous choice model is specified as

$$y_{ij}^* = x_{ij}\beta + \epsilon_{ij}, \quad j = 1,\ldots,G; \qquad i = 1,\ldots,T_j, \tag{5.2}$$

where x_{ij} consists of exogenous variables, and the disturbance has an error-component structure

$$\epsilon_{ij} = u_j + v_{ij}, \tag{5.3}$$

where u_j and v_{ij} are mutually independent, u_j are independent and identically distributed (i.i.d.) for all j, and v_{ij} are i.i.d. for all i and j. The y_{ij}^* are latent variables, but their signs D_{ij} are observable as $D_{ij} = 1$ if $y_{ij}^* > 0$, $D_{ij} = -1$ otherwise. For a probit model, both u and v are assumed to be normally distributed. A typical normalization for the probit model specifies a zero mean and a unit variance on the total disturbance ϵ. With the variance normalization, if u_j is $N(0,\rho)$, then v_{ij} is $N(0,1-\rho)$. For a cross section of time series in Butler and Moffitt (1982), G refers to the number of cross-section units and T_j refers to the number of time periods of the unit j. For the study of group effects in Borjas and Sueyoshi (1994), G is the number of groups and T_j refers to the number of individuals in the group j. The probability in (5.1) is the choice probability for an observed path $(D_1,\ldots,D_T)$ of a single cross-section unit or observed choices of a group (the subscript j of a group in x_{ij}, D_{ij}, and T_j has been suppressed for simplicity).

For illustration, choices $(D_1,\ldots,D_T)$ were generated by the process in (5.2) and (5.3) for a single group where the parameters $x_i\beta$ and ρ were set at some specific values. The integration problem in (5.1) is recognized to be a difficult numerical problem in Borjas and Sueyoshi (1994) when the number of individuals T in a group is large. This is so because the unknown value of this integral can be quite small (see Borjas and Sueyoshi, 1994). Lee (1996) recognizes that the Gaussian quadrature formulation might not provide an accurate estimate of the unknown integral under such a circumstance.[9] We experimented with several different values for $x_i\beta$, ρ, and T. The corresponding integrals were eval-

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

uated by Gaussian quadrature with various Gauss–Hermite points. These results can be compared with simulators with or without control variates. All the computation was done on a Sun-SPARCstation 20 model 720 workstation, and the Fortran codes were written in Fortran 77 language. The Gauss–Hermite points and weights were generated by the subroutine "gauher" from the Numerical Recipes by Press et al. (1992). For results of the Gaussian quadrature reported in tables, the m points refers to the number of Gaussian quadrature points used for I_m in (2.2). Value refers to the Gaussian quadrature evaluation of the unknown integral. Time refers to the average CPU time in seconds per replication for Fortran program execution. For stochastic integration, the NS refers to the sampling simulator (2.9) with random variables drawn from a standard normal density. The C-S refers to the simulator (2.13). The difference of this simulator and I_m, i.e., $\hat{E}_m^{(R)}$ in (2.12), provides an unbiased estimate of the corresponding quadrature error. The C-C refers to the simulator in (2.12) with $d \cdot P_{m-1}(x) - I_m$ as the control variate and common random draws used for estimating β^*. The C-I refers to the same control-variate method but with sample split (as in (2.22)). For stochastic integration, the numbers of random draws used in our study are set to $R = 30$, 60, and 120.[10] The numbers of Gaussian quadrature points used for interpolation are fixed at 4, 8, and 16 through the experiments. We report the empirical mean (Mean) and empirical standard deviation (SD) based on 500 replications for each case of stochastic integration.

Consider first the case with $\rho = 0.3$ and $x_i\beta = 0$ for all i with $T = 10$, 50, 100, or 300. Table 1 reports the Gaussian quadrature. For the integral with $T = 10$, the values of quadratures stabilize with 8 or more quadrature points.[11] The value of the integral is around 5.06×10^{-4}. The integral becomes more complicated with larger T. For $T = 50$, the integrals stabilize with 30 or more quadrature points, and the value of the integral is about 5.2×10^{-15}. For $T = 100$, the values stabilize around 4.5×10^{-29} with 60 or more quadrature points. For the large $T = 300$ case, one can see some variations even for quadratures with very large numbers of points. The quadrature value for $T = 300$ seems to be around 2.1×10^{-81}. The gauher subroutine for generating the Gaussian points and weights reports excessive iterations when 180 points or more are tried. The time cost increases with T and the number of Gaussian points. They are reported here for reference. As one may see later, for large T, because a large number of Gaussian points is needed, the time cost of the Gaussian quadrature might even be larger than those of some simulators. Tables 2 and 3 report results of various simulators, respectively, for the integrals of $T = 10$, 50, 100, and 300. The main features of the various simulators across the two tables can be summarized.

1. All the simulators of NS, C-S, C-C, and C-I are unbiased even with $R = 30$.
2. With the same number of random draws R, the simulators C-S, C-C, and C-I have smaller SD's than those of NS in all cases with a few exceptions. The exceptional cases are those of the C-S simulator using the interpolating polynomial based on

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

Table 1. Gaussian quadrature ($x\beta = 0$ and $\rho = 0.3$)

$T = 10$			$T = 50$			$T = 100$			$T = 300$		
	$\times 10^{-4}$			$\times 10^{-15}$			$\times 10^{-29}$			$\times 10^{-81}$	
m points	Value	Time	m points	Value	Time	m points	Value	Time	m points	Value	Time
4	4.197	0.0011	4	11.036	0.0019	4	12.251	0.0028	4	10.291	0.0084
8	4.968	0.0015	8	7.511	0.0030	8	8.544	0.0049	8	0.733	0.0125
16	5.066	0.0026	16	4.641	0.0057	16	3.114	0.0096	16	0.046	0.0252
20	5.067	0.0034	20	4.713	0.0072	20	2.751	0.0122	25	1.608	0.0393
25	5.067	0.0046	25	5.455	0.0093	25	5.774	0.0155	35	0.576	0.0559
35	5.067	0.0073	30	5.142	0.0117	30	3.888	0.0189	45	1.716	0.0738
			35	5.220	0.0141	35	4.594	0.0225	55	2.863	0.0920
			40	5.226	0.0168	40	4.633	0.0263	65	2.916	0.1109
			50	5.221	0.0223	50	4.682	0.0347	75	2.402	0.1307
						60	4.569	0.0433	85	1.967	0.1509
						70	4.499	0.0530	95	1.839	0.1717
						85	4.499	0.0686	110	2.314	0.2044
						95	4.496	0.0798	120	2.161	0.2265
									130	2.077	0.2498
									150	2.099	0.3014
									170	2.176	0.3512
									180	—	—

TABLE 2. Stochastic integration ($T = 10$ and $T = 50$; $x\beta = 0$ and $\rho = 0.3$)

$T = 10$		$R = 30$			$R = 60$			$R = 120$		
Method	m points	$\times 10^{-4}$ Mean	$\times 10^{-4}$ SD	Time	$\times 10^{-4}$ Mean	$\times 10^{-4}$ SD	Time	$\times 10^{-4}$ Mean	$\times 10^{-4}$ SD	Time
NS		5.074	(0.619)	0.0016	5.066	(0.456)	0.0031	5.065	(0.315)	0.0062
C-S	4	5.068	(0.466)	0.0032	5.065	(0.340)	0.0053	5.068	(0.235)	0.0094
C-C		4.915	(0.465)	0.0032	4.992	(0.308)	0.0053	5.034	(0.210)	0.0103
C-I		5.072	(0.464)	0.0053	5.061	(0.301)	0.0094	5.076	(0.221)	0.0177
C-S	8	5.069	(0.225)	0.0051	5.069	(0.161)	0.0086	5.066	(0.115)	0.0158
C-C		5.029	(0.226)	0.0051	5.051	(0.152)	0.0087	5.058	(0.107)	0.0157
C-I		5.071	(0.221)	0.0087	5.060	(0.148)	0.0158	5.070	(0.107)	0.0299
C-S	16	5.067	(0.057)	0.0120	5.067	(0.038)	0.0225	5.066	(0.028)	0.0401
C-C		5.060	(0.060)	0.0123	5.063	(0.039)	0.0214	5.064	(0.028)	0.0402
C-I		5.066	(0.061)	0.0214	5.066	(0.041)	0.0407	5.066	(0.028)	0.0787
$T = 50$										
Method	m point	$\times 10^{-15}$ Mean	$\times 10^{-15}$ SD	Time	$\times 10^{-15}$ Mean	$\times 10^{-15}$ SD	Time	$\times 10^{-15}$ Mean	$\times 10^{-15}$ SD	Time
NS		5.399	(1.466)	0.0070	5.356	(1.036)	0.0140	5.299	(0.731)	0.0280
C-S	4	5.147	(1.597)	0.0095	5.140	(1.135)	0.0169	5.206	(0.813)	0.0320
C-C		5.189	(1.066)	0.0095	5.214	(0.758)	0.0170	5.232	(0.545)	0.0320
C-I		5.236	(1.093)	0.0170	5.274	(0.813)	0.0320	5.234	(0.565)	0.0620
C-S	8	5.218	(1.066)	0.0124	5.216	(0.764)	0.0211	5.235	(0.535)	0.0391
C-C		5.160	(0.922)	0.0121	5.206	(0.650)	0.0211	5.231	(0.461)	0.0392
C-I		5.253	(0.941)	0.0215	5.280	(0.651)	0.0391	5.223	(0.476)	0.0751
C-S	16	5.271	(0.786)	0.0205	5.242	(0.560)	0.0354	5.274	(0.406)	0.0650
C-C		5.154	(0.764)	0.0274	5.172	(0.521)	0.0354	5.215	(0.385)	0.0650
C-I		5.194	(0.796)	0.0354	5.253	(0.536)	0.0650	5.223	(0.410)	0.1243

four quadrature points with $T = 50$, 100, and 300. The corresponding 4-point quadratures in Table 1 for the $T = 50$, 100, and 300 have been poor.

3. The simulators C-S, C-C, and C-I have relatively larger SD's when only a 4-point interpolation is used in their formulations as compared with those using 8- or 16-point interpolations. For $T = 10$ and 50, the use of the interpolating polynomial based on 16 quadrature points provides the smallest SD's. For $T = 100$, C-S using a 16-point interpolation also has a smaller SD than that using an 8-point interpolation. Except for the latter, there is no significant advantage in using a 16-point interpolation compared to using an 8-point interpolation for $T = 100$ and 300.
4. With the same R, C-C tends to have slightly smaller SD's than C-I has; in turn, C-I tends to have smaller SD's than C-S has. These patterns are more apparent for larger T cases. Among the simulators, the overall best one is the C-C.
5. Among the simulators, with the same R, the NS has the smallest time cost. The NS is then followed by the simulators of C-S and C-C, for which the time costs

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 3. Stochastic integration ($T = 100$ and $T = 300$; $x\beta = 0$ and $\rho = 0.3$)

$T = 100$		$R = 30$			$R = 60$			$R = 120$		
Method	m points	$\times 10^{-29}$ Mean	$\times 10^{-29}$ SD	Time	$\times 10^{-29}$ Mean	$\times 10^{-29}$ SD	Time	$\times 10^{-29}$ Mean	$\times 10^{-29}$ SD	Time
NS		4.692	(1.553)	0.0139	4.632	(1.086)	0.0276	4.578	(0.771)	0.0552
C-S	4	4.412	(1.977)	0.0155	4.392	(1.395)	0.0303	4.475	(1.009)	0.0583
C-C		4.504	(1.254)	0.0155	4.506	(0.882)	0.0298	4.519	(0.638)	0.0583
C-I		4.531	(1.271)	0.0298	4.569	(0.930)	0.0584	4.502	(0.671)	0.1156
C-S	8	4.485	(1.425)	0.0173	4.474	(1.013)	0.0333	4.505	(0.708)	0.0646
C-C		4.447	(1.131)	0.0173	4.484	(0.791)	0.0331	4.510	(0.561)	0.0647
C-I		4.546	(1.151)	0.0331	4.571	(0.793)	0.0647	4.492	(0.583)	0.1278
C-S	16	4.597	(1.152)	0.0243	4.544	(0.808)	0.0459	4.538	(0.588)	0.0893
C-C		4.499	(1.135)	0.0246	4.485	(0.779)	0.0460	4.510	(0.573)	0.0891
C-I		4.491	(1.179)	0.0459	4.564	(0.798)	0.0890	4.491	(0.614)	0.1755
$T = 300$										
Method	m points	$\times 10^{-81}$ Mean	$\times 10^{-81}$ SD	Time	$\times 10^{-81}$ Mean	$\times 10^{-81}$ SD	Time	$\times 10^{-81}$ Mean	$\times 10^{-81}$ SD	Time
NS		2.261	(1.060)	0.0410	2.212	(0.769)	0.0820	2.206	(0.536)	0.1639
C-S	4	2.026	(1.839)	0.0427	2.010	(1.279)	0.0842	2.120	(0.930)	0.1671
C-C		2.145	(0.938)	0.0428	2.135	(0.675)	0.0845	2.171	(0.484)	0.1674
C-I		2.128	(0.978)	0.0843	2.206	(0.686)	0.1676	2.170	(0.499)	0.3333
C-S	8	2.243	(0.990)	0.0446	2.198	(0.717)	0.0980	2.200	(0.505)	0.1737
C-C		2.145	(0.908)	0.0446	2.138	(0.645)	0.0875	2.176	(0.481)	0.1734
C-I		2.134	(0.941)	0.0875	2.211	(0.654)	0.1734	2.165	(0.484)	0.3453
C-S	16	2.260	(1.053)	0.0514	2.211	(0.764)	0.1004	2.206	(0.533)	0.1997
C-C		2.280	(0.958)	0.0515	2.213	(0.668)	0.1140	2.205	(0.458)	0.1978
C-I		2.153	(0.973)	0.1002	2.206	(0.661)	0.1977	2.173	(0.482)	0.3935

are almost the same. The time cost of C-I is about twice that of C-C. The difference in time cost between NS on one side and C-S and C-C on the other is the largest for $T = 10$. For $T = 100$ and 300, the corresponding differences are small. This feature can be explained. The simulators with control variates involve additional evaluation of the interpolating polynomial and a simple regression as compared with the construction of an NS simulator. However, these additional computations are rather elementary as compared with the evaluation of the integrand in (5.1), which involves the evaluation of normal probabilities. The dominant cost on evaluating the integrand for our example also shows up as R increases. As R increases, the computing time of an NS simulator increases in proportion to R. On the other hand, the computing time of the C-S and C-C simulators increases less than a proportion. In some cases, with their time costs being similar or equal, the C-C simulator can have smaller SD's than NS has.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

6. A comparison of the Gaussian quadrature approximations with the control-variate simulators reveals that the quadrature provides computationally efficient and accurate approximation for the integral with small T ($T = 10$). For large T ($T = 50$, 100, and 300), the stochastic simulators can be more robust even with a small number of quadrature points (8 or 16) for interpolation.

To investigate the possible sensitivity of the various simulators on statistical performance, additional MC experiments are performed by varying the values of ρ and $x_i\beta$. Tables 4–6 report results for the case with $\rho = 0.7$ and $x_i\beta = 0$ for all i. This case implies that the variance of the error component u is much larger than the variance of v in (5.3). Comparing the Gaussian quadratures in Table 4 with those in Table 1 shows that the quadratures for the case with $\rho = 0.7$ are less accurate than those of $\rho = 0.3$ for $T = 50$ or more. For $T = 100$ and 300, the quadratures do not stabilize with very large numbers of quadrature points. The various simulators for each T have similar mean values. The simulators C-S, C-C, and C-I have smaller SD's than those of the NS. The efficiency gains for those simulators are larger for small T than those of larger T's.

TABLE 4. Gaussian quadrature ($x\beta = 0$ and $\rho = 0.7$)

$T = 10$		$T = 50$		$T = 100$		$T = 300$	
$\times 10^{-1}$		$\times 10^{-8}$		$\times 10^{-13}$		$\times 10^{-38}$	
m points	Value	m points	Value	m points	Value	m points	Value
4	1.605	4	22.122	4	7.307	4	0.607
8	1.575	8	1.726	8	0.031	8	0.000
16	1.734	16	5.522	16	4.815	16	0.027
20	1.740	20	11.134	20	12.985	25	0.001
25	1.731	25	5.136	25	1.405	35	0.921
35	1.734	35	8.579	30	9.990	45	5.676
		40	9.011	35	9.101	55	3.017
		50	8.340	40	3.476	65	0.570
		60	9.296	50	5.569	75	0.327
		70	9.913	60	8.914	85	1.417
				70	9.289	95	3.130
				85	7.707	110	0.819
				95	8.295	120	1.702
						130	2.738
						150	3.102
						170	1.863
						180	—

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 5. Stochastic integration ($T = 10$ and $T = 50$; $x\beta = 0$ and $\rho = 0.7$)

$T = 10$		$R = 30$		$R = 60$		$R = 120$	
Method	m points	$\times 10^{-1}$ Mean	$\times 10^{-1}$ SD	$\times 10^{-1}$ Mean	$\times 10^{-1}$ SD	$\times 10^{-1}$ Mean	$\times 10^{-1}$ SD
NS		1.757	(0.568)	1.754	(0.411)	1.742	(0.273)
C-S	4	1.736	(0.252)	1.737	(0.179)	1.733	(0.120)
C-C		1.747	(0.215)	1.747	(0.152)	1.739	(0.106)
C-I		1.728	(0.238)	1.731	(0.163)	1.728	(0.117)
C-S	8	1.735	(0.130)	1.733	(0.093)	1.734	(0.061)
C-C		1.764	(0.142)	1.746	(0.099)	1.738	(0.062)
C-I		1.726	(0.152)	1.735	(0.091)	1.734	(0.062)
C-S	16	1.734	(0.054)	1.736	(0.037)	1.735	(0.026)
C-C		1.731	(0.054)	1.735	(0.036)	1.734	(0.025)
C-I		1.737	(0.054)	1.734	(0.034)	1.735	(0.025)
$T = 50$		$\times 10^{-8}$	$\times 10^{-8}$	$\times 10^{-8}$	$\times 10^{-8}$	$\times 10^{-8}$	$\times 10^{-8}$
Method	m points	Mean	SD	Mean	SD	Mean	SD
NS		9.837	(4.116)	9.790	(3.007)	9.800	(2.070)
C-S	4	9.333	(4.284)	9.358	(2.996)	9.614	(2.070)
C-C		9.283	(3.520)	9.419	(2.544)	9.620	(1.744)
C-I		9.386	(3.633)	9.719	(2.512)	9.724	(1.757)
C-S	8	9.796	(3.944)	9.758	(2.874)	9.785	(1.985)
C-C		9.472	(3.514)	9.532	(2.538)	9.690	(1.813)
C-I		9.424	(3.670)	9.758	(2.450)	9.696	(1.826)
C-S	16	9.757	(3.161)	9.744	(2.326)	9.783	(1.551)
C-C		9.824	(3.234)	9.774	(2.315)	9.791	(1.538)
C-I		9.479	(3.405)	9.752	(2.330)	9.728	(1.534)

The efficiency gains are rather small for this case ($\rho = 0.7$) with $T = 100$ or 300. These may reflect the poor interpolations underlying the Gaussian quadratures for those cases.

Additional experiments by varying the value of $x\beta$ to $x_i\beta = 1$ or $x_i\beta = -1$ for all i (while $\rho = 0.3$) are performed for $T = 50$. Tables 7 and 8 report the corresponding results. From Table 7, we see that the Gaussian quadratures with 8 points or more are quite accurate for the case $x\beta = -1$. For the case $x\beta = 1$, the quadratures stabilize with 20 or more points. The better accuracy of corresponding interpolating polynomials reflects in the substantially better performances of the simulators C-S, C-C, and C-I than that of the NS in Table 8.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 6. Stochastic integration ($T = 100$ and $T = 300$; $x\beta = 0$ and $\rho = 0.7$)

$T = 100$		$R = 30$		$R = 60$		$R = 120$	
Method	m points	$\times 10^{-13}$ Mean	$\times 10^{-13}$ SD	$\times 10^{-13}$ Mean	$\times 10^{-13}$ SD	$\times 10^{-13}$ Mean	$\times 10^{-13}$ SD
NS		7.512	(3.883)	7.532	(2.761)	7.569	(1.935)
C-S	4	7.346	(3.570)	7.389	(2.501)	7.507	(1.727)
C-C		7.072	(3.479)	7.235	(2.450)	7.421	(1.708)
C-I		7.173	(3.574)	7.478	(2.444)	7.547	(1.730)
C-S	8	7.512	(3.881)	7.532	(2.759)	7.568	(1.933)
C-C		7.277	(3.510)	7.350	(2.480)	7.490	(1.780)
C-I		7.203	(3.627)	7.515	(2.416)	7.521	(1.809)
C-S	16	7.443	(3.148)	7.492	(2.217)	7.555	(1.529)
C-C		7.396	(3.209)	7.483	(2.230)	7.547	(1.530)
C-I		7.215	(3.347)	7.506	(2.268)	7.551	(1.501)
$T = 300$		$\times 10^{-38}$	$\times 10^{-38}$	$\times 10^{-38}$	$\times 10^{-38}$	$\times 10^{-38}$	$\times 10^{-38}$
Method	m points	Mean	SD	Mean	SD	Mean	SD
NS		2.023	(1.667)	2.036	(1.252)	2.114	(0.890)
C-S	4	1.997	(1.651)	2.023	(1.233)	2.109	(0.873)
C-C		1.888	(1.587)	1.952	(1.188)	2.068	(0.848)
C-I		1.913	(1.623)	2.061	(1.187)	2.131	(0.796)
C-S	8	2.023	(1.667)	2.036	(1.252)	2.114	(0.890)
C-C		1.941	(1.580)	1.981	(1.169)	2.078	(0.865)
C-I		1.937	(1.598)	2.066	(1.179)	2.107	(0.839)
C-S	16	2.021	(1.664)	2.035	(1.250)	2.113	(0.889)
C-C		2.022	(1.620)	2.052	(1.203)	2.139	(0.783)
C-I		1.961	(1.609)	2.060	(1.190)	2.154	(0.735)

5.2. A Double Integral

Gaussian quadrature can be applied to double integration problems. For a general double integral of the form $I = \int_a^b \int_a^b r(x,y)\omega(x)\omega(y)dxdy$ with a weight function $\omega(x)$, an m-point Gaussian quadrature based on a product rule will be $I_m = \sum_{i=1}^m \sum_{j=1}^m \omega_i \omega_j r(x_i, x_j)$ where x_i, $i = 1, \ldots, m$, are the m quadrature points and ω's are the corresponding weights. With m quadrature points, the underlying interpolating polynomial $P_{m-1}(x,y)$ can be

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 7. Gaussian quadrature ($\rho = 0.3$ and $T = 50$)

$x\beta = -1$		$x\beta = 1$	
$\times 10^{-3}$		$\times 10^{-13}$	
m points	Value	m points	Value
4	2.182	4	4.904
8	1.833	8	3.177
16	1.880	16	2.136
20	1.884	20	2.226
25	1.886	25	2.449
30	1.885	30	2.395
35	1.885	35	2.388
40	1.885	40	2.403
		50	2.398
		60	2.397
		70	2.397

$$P_{m-1}(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{m} r(x_i,x_j)\,\frac{(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_m)}{(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_m)}$$

$$\times\,\frac{(y-x_1)\ldots(y-x_{j-1})(y-x_{j+1})\ldots(y-x_m)}{(x_j-x_1)\ldots(x_j-x_{j-1})(x_j-x_{j+1})\ldots(x_j-x_m)}. \tag{5.4}$$

This interpolating polynomial is used as a control variate for our simulators.

To provide some possible numerical evidence of the usefulness of interpolating polynomials as control variates for simulators involving a double integral, we generalize the case in (5.1) to a double integration problem by assuming that the β in (5.1) is a normal random variable $N(\mu,\sigma^2)$ with a mean μ and variance σ^2. This can be motivated by a random coefficient, discrete panel data model as in (5.2) and (5.3): $y_{ij}^{*} = x_{ij}\beta_j + u_j + v_{ij}$ with both the intercepts u_j and the slope coefficient β_j being randomly distributed across units $j = 1,\ldots,G$. Thus, the integral under consideration is

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\prod_{i=1}^{T}\Phi\left[D_i\left(\frac{x_i\beta+\rho^{1/2}w}{(1-\rho)^{1/2}}\right)\right]\phi(w)\cdot\frac{1}{\sigma}\,\phi\left(\frac{\beta-\mu}{\sigma}\right)dwd\beta. \tag{5.5}$$

For our MC studies, the parameter ρ is set at 0.3 and x_i is generated by a standard normal generator. We have experimented with distributions of β with

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 8. Stochastic integration ($T = 50$; $\rho = 0.3$)

$x\beta = -1$		$R = 30$		$R = 60$		$R = 120$	
Method	m points	$\times 10^{-3}$ Mean	$\times 10^{-3}$ SD	$\times 10^{-3}$ Mean	$\times 10^{-3}$ SD	$\times 10^{-3}$ Mean	$\times 10^{-3}$ SD
NS		1.922	(0.475)	1.916	(0.338)	1.897	(0.234)
C-S	4	1.877	(0.265)	1.877	(0.190)	1.880	(0.131)
C-C		1.843	(0.270)	1.862	(0.193)	1.872	(0.132)
C-I		1.878	(0.261)	1.884	(0.181)	1.889	(0.139)
C-S	8	1.885	(0.151)	1.884	(0.108)	1.885	(0.074)
C-C		1.878	(0.145)	1.878	(0.101)	1.883	(0.068)
C-I		1.875	(0.145)	1.887	(0.095)	1.885	(0.069)
C-S	16	1.886	(0.045)	1.887	(0.031)	1.887	(0.021)
C-C		1.888	(0.044)	1.888	(0.030)	1.887	(0.020)
C-I		1.887	(0.044)	1.886	(0.030)	1.886	(0.020)
$x\beta = 1$		$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$
Method	m points	Mean	SD	Mean	SD	Mean	SD
NS		2.476	(0.652)	2.458	(0.463)	2.434	(0.326)
C-S	4	2.364	(0.693)	2.362	(0.493)	2.392	(0.352)
C-C		2.379	(0.462)	2.393	(0.330)	2.402	(0.237)
C-I		2.402	(0.475)	2.421	(0.355)	2.406	(0.245)
C-S	8	2.399	(0.445)	2.399	(0.320)	2.406	(0.225)
C-C		2.371	(0.401)	2.392	(0.284)	2.404	(0.202)
C-I		2.410	(0.411)	2.424	(0.284)	2.400	(0.209)
C-S	16	2.419	(0.325)	2.408	(0.233)	2.411	(0.168)
C-C		2.364	(0.307)	2.375	(0.209)	2.396	(0.154)
C-I		2.385	(0.318)	2.410	(0.214)	2.401	(0.162)

a unit variance and various mean values, namely, $N(0,1)$, $N(1,1)$, and $N(-1,1)$. The T's considered are $T = 10$ or 50. The number of repetitions is again 500 for each case. The MC results are reported in Tables 9–11. Table 9 indicates that for $T = 50$, the Gaussian quadratures for $\beta \sim N(1,1)$ stabilize with 16 points but, for the case with $\beta \sim N(-1,1)$, 25 points or more are needed. For the case with $\beta \sim N(0,1)$, 35 or more points will be needed for $T = 50$ even though 8 points are doing fine with $T = 10$. For stochastic integration, the results in Tables 10 and 11 show that the hybrid simulator C-S can definitely have smaller SD's than those of the NS simulator when a 16-point interpolat-

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 9. Gaussian quadrature: Double integral ($\rho = 0.3$)

$\beta \sim N(0,1)$				$\beta \sim N(1,1)$		$\beta \sim N(-1,1)$	
$T = 10$		$T = 50$		$T = 50$		$T = 50$	
$\times 10^{-3}$		$\times 10^{-14}$		$\times 10^{-7}$		$\times 10^{-13}$	
m points	Value	*m* points	Value	*m* points	Value	*m* points	Value
4	3.557	4	24.968	4	3.366	4	2.129
8	2.635	8	15.694	8	1.785	8	2.494
16	2.533	16	4.736	16	1.613	16	1.654
20	2.547	20	3.806	20	1.623	20	1.567
25	2.562	25	6.893	25	1.634	25	1.176
35	2.558	30	4.287	30	1.630	30	1.284
		35	5.894			35	1.335
		40	5.084			40	1.220
		50	5.377			50	1.247
		60	5.408			60	1.272

ing polynomial is used. In all cases, the simulators C-C and C-I have smaller SD's than those of the NS simulator. Overall, the C-C simulator is slightly better than the C-I simulator. The overall evidence is thus similar to that of the single integral case.

6. CONCLUSIONS

In this paper, we consider features in numerical and stochastic integration approaches for the evaluation of an analytically intractable integral. Instead of regarding them as competitors, our analysis reveals that they can supplement each other and, hence, provide a coupling of these two approaches. Our analysis reveals that interpolation and control variates provide the essential link.

Many familiar and popular quadrature formulations of a numerical integral are based on interpolation and partition of a region of an integral. Newton–Cotes quadrature and Gaussian quadrature are based on interpolation by polynomials. Other forms such as trigonometric interpolation and smooth splines may also be useful. The stochastic integration approach is justified with the law of large numbers. Whereas numerical integration methods are used mainly for low-dimensional integration problems, the stochastic method can have broad applications even for high-dimensional integrals. The stochastic integration

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 10. Stochastic integration: Double integral ($\beta \sim N(0,1)$ and $\rho = 0.3$)

$T = 10$		$R = 30$		$R = 60$		$R = 120$	
Method	m-points	$\times 10^{-3}$ Mean	$\times 10^{-3}$ SD	$\times 10^{-3}$ Mean	$\times 10^{-3}$ SD	$\times 10^{-3}$ Mean	$\times 10^{-3}$ SD
NS		2.573	(0.766)	2.538	(0.524)	2.543	(0.371)
C-S	4	2.575	(0.587)	2.560	(0.409)	2.562	(0.291)
C-C		2.510	(0.465)	2.520	(0.313)	2.538	(0.227)
C-I		2.570	(0.486)	2.543	(0.335)	2.546	(0.242)
C-S	8	2.563	(0.328)	2.546	(0.231)	2.557	(0.160)
C-C		2.521	(0.334)	2.529	(0.236)	2.549	(0.164)
C-I		2.562	(0.338)	2.561	(0.232)	2.547	(0.170)
C-S	16	2.557	(0.141)	2.555	(0.095)	2.557	(0.070)
C-C		2.551	(0.142)	2.555	(0.097)	2.557	(0.071)
C-I		2.566	(0.138)	2.555	(0.098)	2.556	(0.072)
$T = 50$		$\times 10^{-14}$	$\times 10^{-14}$	$\times 10^{-14}$	$\times 10^{-14}$	$\times 10^{-14}$	$\times 10^{-14}$
Method	m points	Mean	SD	Mean	SD	Mean	SD
NS		5.395	(3.504)	5.385	(2.573)	5.355	(1.908)
C-S	4	5.263	(7.830)	5.512	(5.590)	5.433	(3.849)
C-C		5.053	(3.039)	5.276	(2.326)	5.274	(1.631)
C-I		5.520	(3.454)	5.278	(2.284)	5.269	(1.601)
C-S	8	5.040	(5.731)	5.207	(3.896)	5.314	(2.694)
C-C		4.782	(2.755)	5.106	(2.051)	5.209	(1.429)
C-I		5.398	(3.148)	5.312	(2.070)	5.242	(1.463)
C-S	16	5.293	(2.662)	5.293	(1.960)	5.332	(1.353)
C-C		4.922	(2.757)	5.126	(2.010)	5.235	(1.388)
C-I		5.370	(2.991)	5.333	(1.956)	5.242	(1.400)

method can provide an unbiased estimate of the unknown integral, but its accuracy may depend on the design of appropriate simulators. We point out that important features in quadrature formulations, namely, interpolation (and in this regard, function approximation also) and region partition, can provide valuable devices for the design of stochastic simulators. The main message of this paper is that an interpolating function can be used as a valuable control variate for variance reduction in simulation. We illustrate possible variance reduction by some numerical cases with Gaussian quadrature. The resulting simulator may also be regarded as a monitor of the approximation error of a quadrature.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

TABLE 11. Stochastic integration: Double integral ($\rho = 0.3$ and $T = 50$)

$\beta \sim N(1,1)$		$R = 30$		$R = 60$		$R = 120$	
Method	m points	$\times 10^{-7}$ Mean	$\times 10^{-7}$ SD	$\times 10^{-7}$ Mean	$\times 10^{-7}$ SD	$\times 10^{-7}$ Mean	$\times 10^{-7}$ SD
NS		1.662	(0.780)	1.632	(0.530)	1.636	(0.397)
C-S	4	1.639	(0.885)	1.646	(0.627)	1.645	(0.430)
C-C		1.557	(0.584)	1.597	(0.396)	1.614	(0.282)
C-I		1.629	(0.610)	1.636	(0.423)	1.636	(0.304)
C-S	8	1.622	(0.463)	1.618	(0.332)	1.634	(0.233)
C-C		1.541	(0.492)	1.585	(0.336)	1.613	(0.231)
C-I		1.617	(0.504)	1.643	(0.352)	1.628	(0.249)
C-S	16	1.634	(0.222)	1.634	(0.158)	1.634	(0.115)
C-C		1.597	(0.274)	1.627	(0.169)	1.631	(0.117)
C-I		1.652	(0.244)	1.630	(0.170)	1.629	(0.121)
$\beta \sim N(-1,1)$		$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$	$\times 10^{-13}$
Method	m points	Mean	SD	Mean	SD	Mean	SD
NS		1.306	(0.792)	1.320	(0.587)	1.306	(0.424)
C-S	4	1.294	(0.829)	1.331	(0.577)	1.313	(0.410)
C-C		1.258	(0.728)	1.298	(0.519)	1.294	(0.370)
C-I		1.359	(0.760)	1.279	(0.522)	1.307	(0.367)
C-S	8	1.250	(0.927)	1.293	(0.618)	1.299	(0.440)
C-C		1.197	(0.650)	1.264	(0.462)	1.280	(0.335)
C-I		1.354	(0.692)	1.279	(0.498)	1.300	(0.345)
C-S	16	1.318	(0.545)	1.320	(0.379)	1.300	(0.263)
C-C		1.205	(0.539)	1.267	(0.370)	1.276	(0.253)
C-I		1.336	(0.549)	1.290	(0.384)	1.294	(0.253)

NOTES

1. This view is not necessarily true for some cases as demonstrated later in Section 5.

2. Unbiasedness is not necessarily a requirement, but it is a desirable property of a simulator.

3. In a stochastic integral, there is naturally a density involved as the measure of the integral. However, if it is difficult to simulate random variables from the natural density or when the variance of the resulting simulator is large, one may prefer to select an alternative density that has the same (or large) support of the original density. Random variables can be simulated from the alternative density, and, with the original integrand modified accordingly, the resulting sampler is known as an IS simulator.

4. We note that the choice of $w(x)$ (and hence $w^*(x)$) needs not to be dictated by its being a classical basis function for a familiar quadrature. Given (2.1), an investigator may have selected an

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

IS by some sophisticated procedure with the hope that the resulting IS simulator can have little MC variance. With such an IS density, our suggestions described subsequently can still be applied. If the selected IS density does not correspond to a classical basis function, Gaussian points and weights may still be generated by a subroutine available in the Numerical Recipe package (Press et al., 1992).

5. A random variable Z is called a control variable for Y if the expectation of Z is known and if Z is correlated with Y. The method of antithetic variates for MC-variance reduction can be regarded as a special case of the method of control variates.

6. In general, with sample split, the number of independent draws for $\hat{I}_R$ and H_R does not need to equal the number of draws for the estimation of β^*.

7. Because Gaussian points x_i, $i = 1,\ldots,m$, are zeros of the orthogonal polynomial of degree m, this implies that $\int_a^b L_m(x)Q_{m-1}(x)w(x)dx = 0$ where $L_m(x) = \prod_{i=1}^m (x - x_i)$ and $Q_{m-1}(x)$ is any polynomial of maximum degree $(m - 1)$. It follows that $\int_a^b L_m(x)p_m(x)w^*(x)dx = 0$ for $p_m(x) = (1, x,\ldots, x^{m-1})$. As the mean of $L_m(x)p_m(x)$ is known (zero), in principle, elements of the vector $L_m(x)p_m(x)$ can also be used as control variables. However, as there is no good reason to expect that elements of $L_m(x)p_m(x)$ would be highly correlated with $r(x)$ in addition to the interpolation polynomial, they are not recommended to be included as additional control variates. The inclusion of such variables would also increase substantially the number of control variates. Lavenberg and Welch (1981) emphasize the importance of keeping the number of control variables relatively small so that the gain in variance reduction can be realized with an estimated β^*. We have experimented in some preliminary MC studies with including $L_m(x)p_m(x)$ as additional control variates. Our results confirm the view of Lavenberg and Welch (1981).

8. The latter condition is intended to equalize the cost of simulation draws for the two approaches.

9. This issue motivates our search for other integration methods. This paper is a consequence of that pursuit.

10. For the method of control variates with sample split, the total number of random draws will effectively be double.

11. Butler and Moffitt (1982) suggest the use of 4-point quadrature in their empirical study with panel data of time lengths no more than 11 periods.

REFERENCES

Abramowitz, M. & I. Stegun (1964) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* National Bureau of Standards Applied Mathematics Series 55. Washington, DC: U.S. Government Printing Office.

Borjas, G.J. & G.T. Sueyoshi (1994) A two-stage estimator for probit models with structural group effects. *Journal of Econometrics* 64, 165–182.

Butler, J.S. & R. Moffitt (1982) A computationally efficient quadrature procedure for the one-factor multinomial probit model. *Econometrica* 50, 761–764.

Davidson, R. & J.G. MacKinnon (1992) Regression-based methods for using control variates in Monte Carlo experiments. *Journal of Econometrics* 54, 203–222.

Geweke, J. (1989) Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57, 1317–1340.

Geweke, J. (1996) Monte Carlo simulation and numerical integration. In H.M. Amman, D.A. Kendrick, & J. Rust (eds.), *Handbook of Computational Economics*, vol. 1. Amsterdam: Elsevier Science B.V.

Hammersley, J.M. & D.C. Handscomb (1964) *Monte Carlo Methods*. London: Methuen.

Heckman, J.J. & R.J. Willis (1975) Estimation of a stochastic model of reproduction: An econometric approach. In N. Terleckyj (ed.), *Household Production and Consumption*. New York: Cambridge University Press.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press

Hendry, D.F. (1984) Monte Carlo experimentation in econometrics. In Z. Griliches & M.D. Intriligator (eds.), *Handbook of Econometrics*, vol. II. Amsterdam: North-Holland.
Hildebrand, F.B. (1974) *Introduction to Numerical Analysis*, 2nd ed. New York: Dover.
Kahaner, D., C. Moler, & S. Nash (1989) *Numerical Methods and Software*. Englewood Cliffs, New Jersey: Prentice-Hall.
Lavenberg, S.S. & P.D. Welch (1981) A perspective on the use of control variables to increase the efficiency of Monte Carlo simulations. *Management Science* 27, 322–335.
Lee, L.F. (2000) A numerically stable quadrature procedure for the one-factor random-component discrete choice model. *Journal of Econometrics* 95, 117–129.
Press, W.H., S.A. Teukolsky, W.T. Vetterling, & B.P. Flannery (1992) *Numerical Recipes*, 2nd ed. New York: Cambridge University Press.
Ripley, B.D. (1987) Stochastic Simulation. New York: Wiley.
Ross, S.M. (1990) *A Course in Simulation*. New York: Macmillan.
Stroud, A. & D. Secrest (1966) *Gaussian Quadrature Formulas*. Englewood Cliffs, New Jersey: Prentice-Hall.

https://doi.org/10.1017/S0266466601175043 Published online by Cambridge University Press