Gaussian Quadrature versus Simulation for the Estimation of Random Parameters: Some Evidence from Stated-Preference Choice Data

William S. Breffle
Stratus Consulting
PO Box 4059
Boulder, CO 80306-4059
bbreffle@stratusconsulting.com

Edward Morey
Department of Economics
University of Colorado
Edward.Morey@colorado.edu
http://www.colorado.edu/Economics/morey/

Donald Waldman
Department of Economics
University of Colorado
Donald.Waldman@colorado.edu

August 30, 2004

Abstract

In environmental economics, numerical simulation using random draws is the method most commonly used to estimate joint probabilities of individual choices in discrete-choice, random-parameters models. This paper compares simulation to another method of estimation, Gaussian quadrature, on the basis of speed and accuracy. The comparison is done using stated preference data consisting of the answers to choice questions for fishing in Green Bay, a large bay on Lake Michigan. Each sampled individual chose between a pair of Green Bay scenarios with different fishing conditions. Quadrature is found to be as accurate as simulation based on random draws, but Gaussian quadrature attains stability in estimated parameters considerably faster.

Acknowledgement 1 *Thanks to Anna Alberini, Trudy Cameron, Stephane Hess, David Layton, Ricardo Scarpa and Ken Train for comments on earlier drafts.*

1 Introduction

Simulation is the most common numerical method used to estimate random parameters models. Using random draws to simulate moments of a distribution dates back to [McFadden, 1989],[1] and has been used to model demand for an array of environmental commodities. Its use is likely to increase as a result of this volume and the recent book by [Train, 2003].[2]

While simulation studies are numerous, little research has been conducted to examine the speed and accuracy of the simulation method, or the importance of addressing simulation noise (variance due to simulation). Brownstone and Train (1999) examine the sensitivity of average probabilities, the log-likelihood function, and parameter gradients to different numbers of draws and different sets of random numbers (that is, different values for the random number generator seed), holding constant the values of the parameters that generate the data as they conduct the tests (that is, a new model is not estimated for every value of the seed). [Breffle & Morey, 2000] determine the minimum number of draws needed to stabilize the parameter estimates, and to minimize the simulation noise in estimated expected consumer surplus.

The purpose of this chapter is to compare simulation to another method of estimation, Gaussian quadrature (see [Butler & Moffit, 1982]; [Waldman, 1985]; [Geweke, 1996]). The comparison is done using stated preference data consisting of the answers to choice questions for fishing in Green Bay. Each sampled individual chose between a pair of Green Bay scenarios with different fishing conditions. Choices were made over multiple pairs with varied characteristics.

2 A choice model with random parameters

Assume individual i answers J pair-wise choice questions. Assume the utility from choosing alternative K in pair j is:

$$U_{ij}^K = \boldsymbol{\beta}_i' \mathbf{x}_{ij}^K + \varepsilon_{ij}^K \tag{1}$$

where the $L \times 1$ vector $\mathbf{x}_{ij}^K$ contains the observed characteristics of alternative K. The stochastic component ε_{ij}^K is assumed to be i.i.d. and normally distributed, generating a probit model. Under the assumption of random heterogeneous preferences,

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \mathbf{u}_i, \text{ where } \mathbf{u} \sim N(\mathbf{0}, \boldsymbol{\Sigma}) \tag{2}$$

where $\mathbf{u}_i$ is a random $L \times 1$ vector representing the differences between the mean marginal utilities of the characteristics and individual i's marginal utilities,

[1] See [Hajivassiliou *et al.* , 1992] [McFadden & Ruud, 1994], and [Stern, 1992].

[2] See, for example, [Train, 1998], [Brownstone & Train, 1999], [Revelt & Train, 1998], [Layton & Brown, 2000], and [Breffle & Morey, 2000]. [Train, 2003] includes additional examples. This volume contains many more.

and $\mathbf{\Sigma}$ is the variance-covariance matrix of $\mathbf{u}$. Assuming that $\mathbf{u}_i$ is constant across choice occasions causes an individual's different choices to be correlated. Under the assumption that $\mathbf{\Sigma}$ is diagonal (that is, the random parameters are uncorrelated with one another), the joint probability of observing the J pairwise choices of individual i, P_i, is:

$$\begin{aligned} P_i &= P(U_{i1}^k > U_{i1}^{3-k}, ..., U_{iJ}^k > U_{iJ}^{3-k}) \\ &= P(\varepsilon_{i1}^{3-k} - \varepsilon_{i1}^k < \boldsymbol{\beta}_i^{'}(\mathbf{x}_{i1}^{3-k} - \mathbf{x}_{i1}^k), ..., \varepsilon_{iJ}^{3-k} - \varepsilon_{iJ}^k < \boldsymbol{\beta}_i^{'}(\mathbf{x}_{iJ}^{3-k} - \mathbf{x}^k) \\ &= \int_{-\infty}^{+\infty} \cdots \int \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \mathbf{u})^{'}(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^k)\right] f(\mathbf{u}) du_1 ... du_L \end{aligned} \tag{3}$$

where $k = 1$ if alternative 1 was chosen and 2 if alternative 2 was chosen; $\Phi(\cdot)$ is the univariate, standard-normal CDF and $f(\cdot)$ is the L-variate, normal density function. That is, $\Phi(\cdot)$ is one dimensional, $f(\mathbf{u})$ is L-dimensional, $\Phi(\cdot)$ is standard normal, and $f(\mathbf{u})$ is normal, but not standard normal.

The order of magnitude of the integral in Equation 3 is the number of random parameters (the number of non-zero diagonal elements in $\mathbf{\Sigma}$). For example, if there are two random parameters and the parameters are ordered such that the two random parameters are in the first and second positions, Equation 3 simplifies to:

$$P_i = \int_{-\infty}^{+\infty}\int \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix}\mathbf{u} \\ \mathbf{0}\end{matrix})^{'}(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^k)\right] f(\mathbf{u}) du_1 du_2 \tag{4}$$

where $\mathbf{u}$ now has only two elements and $\mathbf{0}$ is a vector of dimension $(L-2) \times 1$.

The likelihood function is $\prod_{i=1}^{N} P_i$. The P_i can be approximated using either simulation or Gaussian quadrature.

3 Simulation

Using simulation, the multiple integral, Equation 3, is approximated in steps: first, a random draw, $\mathbf{u}^d$, is taken from $\mathbf{u} \sim N(\mathbf{0}, \mathbf{\Sigma})$, then P_i is computed conditional on that draw, $P_i(\mathbf{u}^d) = \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \mathbf{u}^d)^{'}(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^k)\right]$.[3] These two steps are repeated for D draws. The simulated P_i is the average of the conditional $P_i(\mathbf{u}^d)$'s. Details may be found in Train (1998). The attractive feature of simulation is it's versatility: it can be used to approximate many different types of probability integrals.

[3] Note that the $\mathbf{u}$ draw is conditional on $\mathbf{\Sigma}$. When one simulates probabilities in the context of maximum likelihood estimation, the process is called simulated maximum liklihood. The objective is to estimate the $\boldsymbol{\beta}$ and $\mathbf{\Sigma}$, but at each iteration, $P_i(\mathbf{u}^d)$ is conditional on the current interative values of $\boldsymbol{\beta}$ and $\mathbf{\Sigma}$.

A problem with simulating probabilities is "simulation noise". That is, for a given $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$, the simulated P_i is different each time it is re-simulated with a new set of random draws. A simulated P_i is an approximation that differs every time the approximation formula is reapplied, so P_i has a distribution. If the number of draws is "small" this distribution will have a variance sufficiently large that substantively different parameter estimates will result each time the model is estimated. Simulation noise declines, for a given sample size, as the number of draws increases, becoming trivial if the number of draws is large enough.[4] Put simply, the minimum number of draws required for estimation by simulation is the number necessary for model parameters to be stable.[5] Determining this number is time consuming. For example, to check whether $1,000$ draws is sufficient one must re-estimate the model many times with $1,000$ draws to make sure the parameter estimates are "almost" the same in each simulation. If they are not, one has to repeat this exercise with a larger number of draws.

4 Gaussian-Hermite quadrature

4.0.1 The basics of G-H quadrature

Alternatively, Equation 3 can be approximated using a form of Gaussian quadrature called Hermite polynomial quadrature, or Gaussian-Hermite quadrature. Put simply, any integral of the form $\int_{-\infty}^{+\infty} g(v)dv = \int_{-\infty}^{+\infty} e^{-v^2}h(v)dv$ can be approximated by

$$\int_{-\infty}^{+\infty} e^{-v^2}h(v)dv \approx \sum_{m=1}^{M} w_m h(v_m) \tag{5}$$

where M is the number of evaluation points.[6] Put simply, the function to be integrated, minus the e^{-v^2} term, is first evaluated at a number of carefully selected evaluation points, the v_m, $m = 1, 2, ..., M$; then the value of the function at each evaluation point, $h(v_m)$ is weighted by w_m.[7] The approximation of the integral is the sum of these weighted values. Implementation is simple: tables (such as table 25.10 in [Abramowitz & Stegun, 1964]) report the (v_m, w_m) pairs for different ascending values of M; one just has to use them to calculate the $w_m h(v_m)$ terms. It is called Gaussian quadrature because $\int_{-\infty}^{+\infty} e^{-v^2}dv$ is the

[4] In addition, for a given number of draws in the simulation, simulation noise decreases as the number of answered choice-pairs increases.

[5] Stability is subjective and depends on how the parameter estimates will be used.

[6] The accuracy of the approximation increases with the number of evaluation points. Typically, 10 or less are sufficient for parameter stability, not hundreds or thousands as with simulation.

[7] As explained below $w_m \neq e^{-v^2}$.

Gaussian integral, sometimes called the *probability integral*.[8] Gaussian quadrature requires fewer evaluation points than ordinary quadrature (approximately half).

The $v_1, v_2, ..., v_M$ are the M roots of the Hermite polynomial $H_M(v)$; this is why the technique is called Hermite polynomial quadrature.[9] The numerical weights, the w_m, are more difficult to explain, but implementation does not require one to understand why they are the number they are.[10]

To solidify how Hermite polynomial quadrature works, we use it here to determine the area under a univaraite, normal density function with zero mean, $f_u(0,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-u^2/2\sigma^2}$, which we know is 1. Begin by noting that $\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-.5(u^2/\sigma^2)} du = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi}} e^{-v^2} dv$ where $v = \frac{u}{\sqrt{2}\sigma}$. Then note that $\int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi}} e^{-v^2} dv$ is a simple, special case of $\int_{-\infty}^{+\infty} e^{-v^2} f(v) dv$, where $f(v)$ is simply the constant $\frac{1}{\sqrt{\pi}}$. Therefore $\int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi}} e^{-v^2} dv = \sum_{m=1}^{M} w_m f(v_m) = \frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m$. If $M = 2$, this is $\frac{1}{\sqrt{\pi}}(0.886227 + 0.886227) = 1.000\,000\,1$; if $M = 4$ it is $\frac{1}{\sqrt{\pi}}(0.804914 + 0.804914 + 0.0813128 + 0.0813128) = 0.999\,999\,86$, a closer approximation, but both are close enough.

4.1 Appying G-H quadrature to Equation 3

As noted above, Gaussian quadrature requires that the integral be of the form $\int_{-\infty}^{+\infty} e^{-v^2} h(v) dv$; Equation 3 is of this form. For the case of one random-parameter,

---

[8] Note that $\int_{-\infty}^{+\infty} e^{-v^2} dv = \sqrt{\pi}$.

[9] Hermite polynomials are a set of orthogonal polynomials with domain $(-\infty, +\infty)$. By way of example, $H_0(v) = 1$, $H_1(v) = 2v$, $H_2(v) = 4v^2 - 2$, $H_3(v) = 8v^2 - 12v$, $H_4(v) = 16v^4 - 48v^2 + 2$ and $H_{10}(v) = 1024v^{10} - 23040v^8 + 161280v^6 - 403200v^4 + 302400v^2 - 30240$. Visually, these polynomials become more and more wave-like as M increases. The M roots of $H_M(v)$, the v_m, are those values of v that make the Hermite polynomial zero. For example $H_2(v)$ has two roots: $\pm.5\sqrt{2}$. And, the four roots of $H_4(v)$ are $\pm\frac{1}{2}\sqrt{\sqrt{34}+6}$and $\pm\frac{1}{2}\sqrt{6-\sqrt{34}}$. *Mathematica*, for example, generates Hermite polynomials with the command $HermiteH[M, v]$.

[10] Here are some of the w_m, reported with their corresponding values of v_m

M	v_m	w_m
2	$\pm\frac{1}{2}\sqrt{2} = \pm 0.707107$	$\frac{1}{2}\sqrt{\pi} = 0.886227$
3	0	$\frac{2}{3}\sqrt{\pi} = 1.18164$
	$\pm\frac{1}{2}\sqrt{6} = \pm 1.22474$	$\frac{1}{6}\sqrt{\pi} = 0.295409$
4	$\pm\sqrt{\frac{3-\sqrt{6}}{2}} = \pm 0.524648$	$\frac{\sqrt{\pi}}{4(3-\sqrt{6})} = 0.804914$
	$\pm\sqrt{\frac{3+\sqrt{6}}{2}} = \pm 1.65068$	$\frac{\sqrt{\pi}}{4(3+\sqrt{6})} = 0.0813128$

For each M , the weights sum to $\sqrt{\pi}$.

Equation 3 simplifies to

$$
\begin{aligned}
P_i &= \int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} u \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] \phi(u) du \qquad (6) \\
&= \int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} u \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-u^2/2\sigma^2}\right] du
\end{aligned}
$$

where u is a scalar and $\mathbf{0}$ is a vector of dimension $(L-1)\times 1$. With a change of variable $v = \frac{u}{\sqrt{2}\sigma}$, this becomes

$$
\begin{aligned}
P_i &= \int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} v\sigma\sqrt{2} \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-(v\sigma\sqrt{2})^2/2\sigma^2}\right] d(v\sigma\sqrt{2}) \\
&= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} v\sigma\sqrt{2} \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] \left[e^{-v^2}\right] dv \\
&= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} h(v) \left[e^{-v^2}\right] dv \approx \frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m h(v_m) \qquad (7)
\end{aligned}
$$

where $h(v) = \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} v\sigma\sqrt{2} \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right]$.

If there are two random parameters, Equation 4, and one adds the assumption they're uncorrelated,

$$
\begin{aligned}
P_i &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} u_1 \\ u_2 \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] f_{u_1}(u_1) f_{u_2}(u_2) du_1 du_2 \\
&= \int_{-\infty}^{+\infty} \left\{ \int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} u_1 \\ u_2 \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] f_{u_1}(u_1) du_1 \right\} f_{u_2}(u_2) du_2 \\
&= \frac{1}{\pi} \int_{-\infty}^{+\infty} \left\{ \int_{-\infty}^{+\infty} \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} v_1\sigma\sqrt{2} \\ v_2\sigma\sqrt{2} \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] e^{-v_1^2} dv_1 \right\} e^{-v_2^2} dv_2 \\
&\approx \frac{1}{\pi} \sum_{m_1=1}^{M_2} w_{m_2} \left[\sum_{m_1=1}^{M_1} w_{m_1} h(v_{1m_1}, v_{2m_2}) \right] \qquad (8)
\end{aligned}
$$

where

$$
h(v_{1m_1}, v_{2m_2}) = \prod_{j=1}^{J} \Phi\left[-(\boldsymbol{\beta} + \begin{matrix} v_{1m1}\sigma\sqrt{2} \\ v_{2m_2}\sigma\sqrt{2} \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k})\right] \qquad (9)
$$

In summary, Gaussian-Hermite quadrature is easy to apply if one assumes a small number of uncorrelated normally-distributed random parameters, a common assumption in many applications.

5 An application: Green Bay fishing under different conditions

The sample consists of 647 randomly-sampled anglers who purchased licenses in eight counties near the bay of Green Bay, a large bay on Lake Michigan, and who fished the bay at least once in 1998. Each angler was presented eight pairs of Green Bay alternatives. Anglers made their selections on the basis of the average time to catch a fish for four species ($PERCH$, $TROUT$, $WALLEYE$, and $BASS$), a fish consumption advisory index (FCA), and a boat launch fee (FEE). The FCA index takes one of nine discrete levels, the first being no advisory, and contains combined information on separate advisories for the four species.

The four parameters on the catch rates and the eight parameters on the FCA dummy variables are all assumed to be random and normally distributed. Specifically, for each angler and each pair, it is assumed that the conditional indirect utility function for alternative is K is

$$\begin{aligned} V_i^K &= (\beta_c + u_{ci}) \\ &\quad \times \left[\beta_p PERCH^K + \beta_t TROUT^K + \beta_w WALLEYE^K + \beta_b BASS^K\right] \\ &\quad +(\beta_{FCA} + u_{FCAi}) \left[\beta_{FCA2} FCA^K + ... + \beta_{FCA9} FCA^K\right] \\ &\quad +\beta_y(-FEE^K) \end{aligned} \tag{10}$$

where

$$f(u_c, u_{FCA}) = N(\begin{array}{c} 0 \\ 0 \end{array}, \begin{array}{cc} \sigma_c & 0 \\ 0 & \sigma_{FCA} \end{array}) \tag{11}$$

The parameter β_y is the marginal utility of money. The parameter β_c is the mean catch-parameter; the β_p,β_t,β_w and β_b allow the means and variances of the catch parameters to vary by species; β_{FCA} is the mean FCA parameter; and the $\beta_{FCA2}, ..., \beta_{FCA9}$ allow the means and variances of the FCA parameters to differ by FCA regime. With this specification, the ratio of the mean parameter to the standard deviation is the same for each of the four catch rates, and for each of the eight FCA levels, so only two standard deviation parameters need to be estimated, σ_c and σ_{FCA}. Assuming that the standard deviation varies in proportion to the mean is a common way of dealing with heteroskedasticity and a reasonable way to limit the number of random parameters that need to be estimated. It also causes an individual's four catch parameters to be correlated with one another, and his eight FCA parameters to be correlated with one

another; something one would expect, and this is accomplished in a way that does not require that one assumes separate u_{cp} and u_{ct} that are correlated. Of course, this specification for the random structure is not necessarily appropriate for other applications and is in no way required for quadrature to work. β_p is set to one to identify the catch parameters and β_{FCA2} is set to one identify the FCA parameters.[11]

This model was estimated using both Hermite quadrature and simulation with random draws. Parameter estimates are reported in Table 1. Results from various model runs show that 500 draws in simulation and 9 evaluation points using quadrature are sufficient for parameter estimates to be stable. That is, for quadrature parameter-estimates, when more than 9 evaluation points are used, the individual parameter estimates are never more than 2% different from the estimates obtained with 9 evaluation points. When at least 500 draws are used in simulation, parameter estimates vary by at most 2% across runs. An important finding is that simulation took almost three times longer than quadrature to reach this level of stability.

Results are also reported for 100 draws using simulation, and 3 and 6 evaluation points for quadrature,but one should not make too much of these. Comparing the properties of estimates estimated with too few quadrature points to estimates estimated with too few random draws is a questionable endeavor: one would never present either as one's parameter estimates.

6 Conclusions

In conclusion, this paper provides an example of a choice question probit model with two random parameters. We demonstrate that to obtain a high level of accuracy, quadrature is faster than simulation with random draws. What should be made of this? When it comes to estimating random-parameter models, there is an alternative to the ubiquitous method of simulation with random draws, and it can be faster and more accurate. It's best suited to applications with a small number of uncorrelated random-parameters.

An interesting issue is what happens to the magnitudes of the simulation errors as one increases the number of draws and what happens to the approximation errors as one increases the number of evaluation points. One would hope that simulation error always decreases as the number of draws increases; that is, it would be comforting to those doing simulation to know that if they re-estimate their model with more draws, there will be less simulation error in the parameter estimates. Unfortunately, this is not always the case as we demonstrate below.

Those using quadrature would hope that re-estimating their model with more evaluation points will always decrease the approximation errors in the parameter estimates. This is the case.

[11] Code for both estimation techniques can be found at http://www.colorado.edu/Economics/morey/dataset.html.

To investigate, we define the bias in parameter β as $Bias_\beta = \left|\hat{\beta} - \beta\right|$ where β is our stable estimate of β (obtained with either 1000 draws or 10 evaluation points). To make this measure of bias comparable across parameters (by accounting for the relative flatness of the likelihood-function), we divide each by the standard error, $s.e.$, of the stable β, $\left|\hat{\beta} - \beta\right| / (s.e_\beta)$. Averaging these over all of the parameters provides one possible measure of aggregate bias, denoted $Bias$. By definition, this measure of aggregate bias is, in our example, zero for simulation with 1000 draws and quadrature with 10 evaluation points.

Note that this measure of aggregate bias, $Bias$, will change every time one reruns the simulation program, even if one takes the same number of draws in each run - the draws are random. There is no comparable variation in $Bias$ when the the parameters are estimated with quadrature - $Bias$ does not vary across runs when the number of evaluation points is held constant because the parameter estimates are always the same.

Our measure of aggregate bias is plotted in Figure 1 for the parameter estimates obtained with 100, 300, 500 and 1000 draws. This example plot proves, by example, that $Bias$ does not always decrease as the number of random draws increases (the biggest "error" for these three sets of estimates is with 500 draws). Every time one re-estimates the model with 100, 300, 500 draws Figure 1 will change, and while one would expect it to usually be monotonically decreasing, it does not have to be, as our example demonstrates.[12]

Figure 2 plots $Bias$ for 3, 6 and 9 and 10 evaluation points and the plot is invariant to reruns. Aggregate bias is monotonically decreasing in the number of evaluation points, as it must when the integral evaluated is of the form $\int_{-\infty}^{+\infty} e^{-v^2} h(v) dv$. Note the big drop from 9 to 10 evaluation points.

What should be concluded? Use enough random draws or evaluation points to get stable parameter estimates. Determining whether one's parameter estimates are stable is more difficult with simulation than with quadrature. With quadrature, just keep increasing the number of evaluation points until the estimates with e evaluation points and $e+1$ evaluation points differ by less than some predetermined percentage. With simulation, increase the number of draws until the parameter estimates differ by less than that predetermined percentage, then increase the number of draws some more to make sure the stability was not a random fluke.

[12] Consider another measure of bias, $\left|ave\hat{\beta} - \beta\right|$ where $ave\hat{\beta} = ave[\hat{\beta}^1 + ... + \hat{\beta}^R]$ and $\hat{\beta}^r$ is the esimate of β obtained the r^{th} time the parameters are simulated holding constant the number of draws ([Hess *et al.* , 2004], [Sandor & Train, 2004]). $\left|\hat{\beta} - \beta\right| = \left|ave\hat{\beta} - \beta\right|$ for quadrature but not for simulation; for simulation there is less randomness in $\left|ave\hat{\beta} - \beta\right|$ than in $\left|\hat{\beta} - \beta\right|$. Therefore, $\left|ave\hat{\beta} - \beta\right|$ is likely to be monotonically decreasing in the number of draws, particularly for large R. However, there is no great practical significance to this likelihood unless the researcher plans to run the simulation program multiple times with each number of random draws.

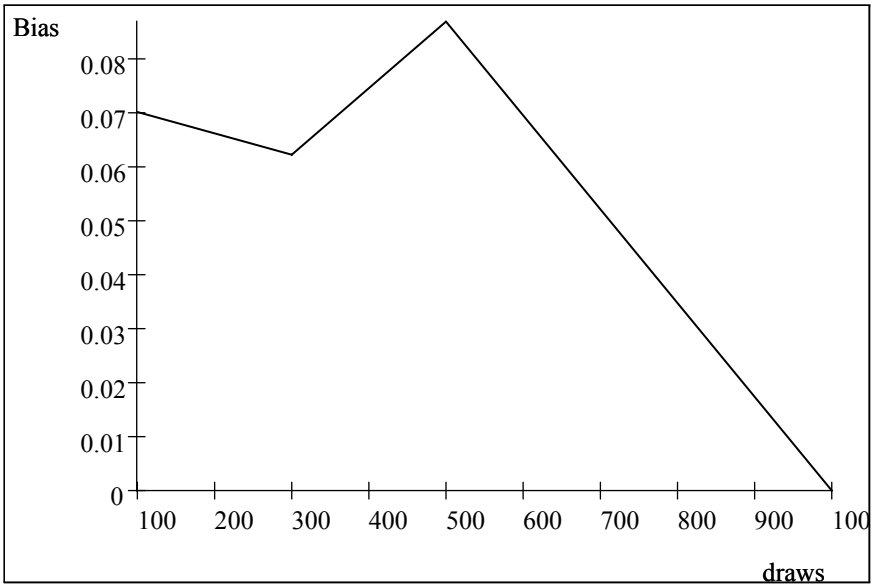

Figure 1: Example simulation bias with 100, 300 and 500 draws

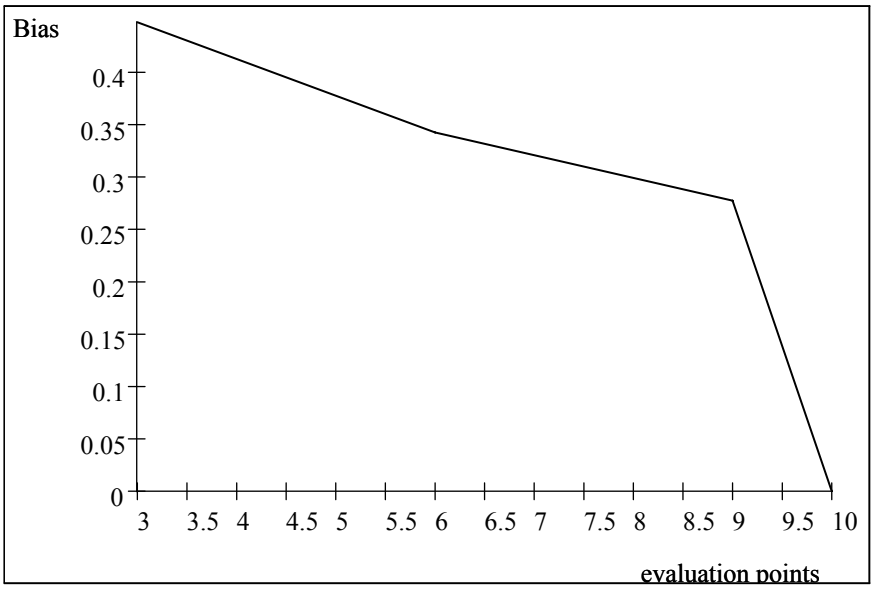

Figure 2: Quadrature bias with 3, 6 and 9 evaluation points

A body of literature is now emerging on simulation with non-random draws; that is, taking systematic draws from the density of interest. Halton draws are one example, and a few applications with environmental implications are available.[13] These methods have the potential to significantly reduce simulation noise and the required number of draws. [Train, 2003] devotes much of his Chapter 9 to a discussion of simulating with systematic draws. After estimating the same model twice, first with 1000 random draws and then with 100 Halton draws, Train summarizes:

> These results show the value of Halton draws. Computer time can be reduced by a factor of ten by using Halton draws instead of random draws, without reducing, and in fact increasing, accuracy. These results need to be viewed with caution, however. The use of Halton draws and other quasi-random numbers in simulation-based estimation is fairly new and not completely understood.

Future research might compare quadrature with simulation using systematic methods of drawing parameter vectors.

In closing, consider a direct mathematical comparison of the simulation formula and quadrature formula for P_i. Consider the case of one random-parameter. With simulation

$$P_i \approx \frac{1}{D} \sum_{d=1}^{D} h(u^d) \tag{12}$$

and for quadrature

$$P_i \approx \frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m h(v_m) \tag{13}$$

where $h(\omega) = \prod_{j=1}^{J} \Phi \left[-(\boldsymbol{\beta} + \begin{matrix} \omega \\ \mathbf{0} \end{matrix})'(\mathbf{x}_{ij}^{3-k} - \mathbf{x}_{ij}^{k}) \right]$. It is evaluated at $\omega_m = v_m \sigma \sqrt{2}$ for quadrature and at $\omega_d = u^d$ with simulation, quadrature carefully selecting the evaluation points,simulation randomly selecting the evaluation points. One should not be surprised that when the draws are carefully selected (e.g. Halton draws), simulation does better than when they are randomly selected. The other big difference is the weights: simulation uses equal weights independent of of how the draws are made; quadrature uses carefully selected weights, each specific to the point at which the function is evaluated.

References

[Abramowitz & Stegun, 1964] Abramowitz, M., & Stegun, I. 1964. *Handbook of Mathematical Functions, with Formulas, Graphs and Math Tables,Applied*

[13] See [Goett *et al.* , 2000], [Hess *et al.* , 2004], and [Sandor & Train, 2004].

Math Series 55. Washington DC: National Bureau of Standards, , U.S. Government Printing Office.

[Breffle & Morey, 2000] Breffle, W.S., & Morey, E.R. 2000. Investigating Preference Heterogeneity in a Repeated Discrete-Choice Recreation Demand Model of Atlantic Salmon Fishing. *Marine Resource Economics*, **15**, 1–20.

[Brownstone & Train, 1999] Brownstone, D., & Train, K. 1999. Forecasting new product penetration with flexible substitution patterns. *Journal of Econometrics*, **89**, 109–129.

[Butler & Moffit, 1982] Butler, J., & Moffit, R. 1982. A computationally efficient quadrature procedure for the one-factor multinomial probit model. *Econometrica*, **50**, 761–764.

[Geweke, 1996] Geweke, K. 1996. Monte Carlo simulation and numerical integration. *Pages 731–800 of:* Kendrick, D., & Rust, J. (eds), *Handbook of Computational Economics*. Elsevier Science, Amsterdam.

[Goett *et al.* , 2000] Goett, A., Train, K., & Hudson, K. 2000. Customer choice among retail energy suppliers: the willingness-to-pay for service attributes. *Energy Journal*, **21**, 1–28.

[Hajivassiliou *et al.* , 1992] Hajivassiliou, V., McFadden, D., & Ruud, P. 1992. *Simulation of multivariate normal rectangle probabilities and their derivatives: theoretical and computational results*. discussion paper 1021,Cowles Foundation, Yale University.

[Hess *et al.* , 2004] Hess, S., Train, K.A, & Polak, J. 2004. On the use of a modified latin hypercube sampling (MLHS) method in the estimation of a mixed logit model for vehicle choice. *Transportation B, forthcoming*.

[Layton & Brown, 2000] Layton, D. F., & Brown, G. 2000. Heterogeneous preferences regarding global climate change. *The Review of Economics and Statistics*, **82**(4), 616–624.

[McFadden, 1989] McFadden, D. 1989. A method of simulated moments for estimation of the multinomial probit without numerical integration. *Econometrica*, **57**, 995–1026.

[McFadden & Ruud, 1994] McFadden, D., & Ruud, P. 1994. Estimation by simulation. *Review of Economics and Statistics*, **76**, 591–608.

[Revelt & Train, 1998] Revelt, D., & Train, K. 1998. Mixed logit with repeated choices: households' choices of appliance efficiency level. *Review of Economics and Statistics*, **80**(4), 647–657.

[Sandor & Train, 2004] Sandor, Z., & Train, K. 2004. Quasi-random simulation of discrete choice models. *Transportation Research Part B*, **38**, 313–327.

[Stern, 1992] Stern, S. 1992. A method for smoothing simulated moments of discrete probabilities in multinomial probit models. *Econometrica*, **60**, 943–952.

[Train, 1998] Train, K. E. 1998. Recreation demand models with taste differences over people. *Land Economics*, **74**(2), 230–239.

[Train, 2003] Train, K.E. 2003. *Discrete Choice Methods with Simulation*. First edn. Cambridge: Cambridge University Press.

[Waldman, 1985] Waldman, D. 1985. Computation in duration models with heterogeneity. *Journal of Econometrics*, **28**, 127–134.

Table 1
Parameter Estimates and Computing Time [1]

Method	Gaussian Hermite quadrature			Simulation	
Evaluation points/ random draws	3	6	9	100	500
Utility parameters					
β_c	-0.637 (-11.638)	-0.649 (-11.487)	-0.645 (-11.607)	-0.639 (-11.439)	-0.648 (-11.607)
β_{FC4}	-0.288 (-3.961)	-0.348 (-4.499)	-0.327 (-4.916)	-0.329 (-4.671)	-0.324 (-4.515)
β_2	1.0 (fixed)	1.0 (fixed)	1.0 (fixed)	1.0 (fixed)	1.0 (fixed)
β_3	0.0470 (6.332)	0.0477 (6.248)	0.0480 (6.384)	0.0488 (6.341)	0.0478 (6.348)
β_4	0.0636 (7.855)	0.0643 (7.842)	0.0647 (7.985)	0.0661 (7.949)	0.0650 (7.989)
β_5	0.0534 (7.220)	0.0554 (7.285)	0.0544 (7.295)	0.0561 (7.346)	0.0544 (7.306)
β_{FC2}	1.0 (fixed)	1.0 (fixed)	1.0 (fixed)	1.0 (fixed)	1.0 (fixed)
β_{FC3}	1.616 (4.957)	1.610 (5.941)	1.618 (6.224)	1.621 (6.010)	1.643 (5.774)
β_{FC4}	2.232 (5.003)	2.129 (6.094)	2.189 (6.519)	2.172 (6.179)	2.215 (5.958)
β_{FC5}	3.080 (4.743)	2.856 (5.713)	2.963 (6.151)	2.916 (5.847)	3.000 (5.608)
β_{FC6}	2.499 (4.602)	2.393 (5.551)	2.463 (5.944)	2.450 (5.654)	2.503 (5.437)
β_{FC7}	3.633 (4.506)	3.421 (5.414)	3.531 (5.857)	3.494 (5.545)	3.578 (5.326)
β_{FC8}	4.980 (4.339)	4.622 (5.160)	4.813 (5.607)	4.741 (5.296)	4.881 (5.098)
β_{FC9}	5.471 (4.324)	5.091 (5.087)	5.300 (5.526)	5.209 (5.226)	5.384 (5.035)
β_T	0.0545 (15.322)	0.0555 (15.206)	0.0555 (15.267)	0.0556 (15.282)	0.0556 (15.282)
Standard deviations [2]					
σ_c	0.423 (-5.397)	0.432 (-5.131)	0.428 (-5.270)	0.417 (-5.334)	0.431 (-5.322)
σ_{FC4}	0.247 (-5.232)	0.342 (-4.478)	0.302 (-5.638)	0.317 (-5.099)	0.296 (-5.238)
Run time (hrs.) on the same computer	0.43	1.14	1.97	1.13	5.56

[1] Asymptotic *t*-statistics are reported in parentheses.
[2] *t*-statistics are for the natural logarithms of the standard deviations. The parameters were exponentiated in estimation to restrict them to be positive.

Figure 1: