# Simulation Based Inference in Econometrics: Motivation and Methods.

Steven Stern

October 23, 1995

# 1 Introduction

Over the last few years, major advances have occurred in the field of simulation. In particular, McFadden(1989) and Pakes and Pollard(1989) have developed simulation methods to simulate expected values of random functions and have shown how to use those simulators in econometric estimation routines. Also, for example, Geweke(1989), Chib(1993), and McCullough and Rossi(1993 ) have shown how to use simulation methods to solve previously unsolvable Bayesian econometrics problems.

Simulation provides an attractive solution for dealing with problems of the following type: Let $U$ be a random variable with density $f(.)$, and let $h(U)$ be some function of $U$. Then

$$Eh(U) = \int h(u) f(u)\, du. \tag{1.1}$$

Most econometrics problems including all method of moments problems and many maximum likelihood problems require one to evaluate equation (1.1) as part of an estimation strategy for estimating a set of parameters $\theta$. There are many cases where $Eh(U)$ can not be evaluated analytically or even numerically with precision. But we usually can simulate $Eh(U)$ on a computer by drawing $R$ "pseudorandom" variables from $f(.)$, $u^1$, $u^2$,.., $u^R$, and then constructing

$$\hat{E}h(U) = \frac{1}{R}\sum_{r=1}^{R} h(u^r). \tag{1.2}$$

Equation (1.2) provides an unbiased estimator of $Eh(U)$[1] which is frequently enough to

[1] $E\hat{E}h(U) = \frac{1}{R}\sum_{r=1}^{R} Eh(U^r) = \frac{1}{R}\sum_{r=1}^{R} Eh(U) = Eh(U)$ as long as $U^r$ is drawn from $f$.

provide consistent estimates (or estimates with small bias) of $\theta$.

This chapter provides some examples to motivate the problem. The first example is the multinomial probit problem, the second is a problem with unobserved heterogeneity, and the third is a Monte Carlo experiment. Next, the chapter describes a set of simulators that improve upon the most naive simulator in equation (1.2). Improvement is in terms of variance reduction, increased smoothness, and reduced computation cost. Then the most common simulation estimators are described. Finally, it evaluates the performance of the various simulators and estimation methods.

## 1.1 Multinomial Probit

The first example is the multinomial probit problem. Consider a model where $y_j^*$ is the value to a person of choosing choice $j$ for $j = 1, 2, ..., J$ (a person index is suppressed). For example, j might index whether to drive a car, ride in someone else's car, take a bus, or take a train to get to work ($J = 4$); it might index whether to work full-time, part-time, or retire ($J = 3$); or it might index whether an elderly person lives independently, in a nursing home, with a family member, or with paid help ($J = 4$). It is assumed that the person chooses the choice $j$ with the greatest value; $j$ is chosen iff $y_j^* > y_k^*$ for all $k \neq j$. Furthermore, it is assumed that $y_j^*$ is a linear function of a set of observed variables and an error:

$$y_j^* = X_j\beta + u_j, \ j = 1, .., J. \tag{1.3}$$

Let $u = (u_1, u_2, .., u_J)'$ be the vector of errors and assume that the covariance matrix of $u$ is $\Omega$. The errors sometimes represent variation in values due to unobserved variables, and sometimes they represent variation in $\beta$'s across people. Let $y_j = 1$ if choice $j$ is chosen; $y_j = 1$ iff $y_j^* > y_k^*$ for all $k \neq j$.

Usually in data, we observe the covariates $X$ and $y = (y_1, y_2, .., y_J)'$ but not $y^* = (y_1^*, y_2^*, .., y_J^*)'$. In order to estimate $\beta$ and $\Omega$, we need to evaluate the probability of observing $y$ conditional on $X$ or the moments of $y$ conditional on $X$. First, note that, since $y_j$ is binary,

$$\begin{aligned} E(y_j \mid X) &= \Pr[y_j = 1 \mid X] \\ &= \Pr\left[y_j^* > y_k^* \; \forall k \neq j \mid X\right]. \end{aligned} \tag{1.4}$$

If we assume that $u_j \sim iid$ Extreme Value, then the probability in equation (1.4) has the analytical form

$$\Pr[y_j = 1 \mid X] = \exp\{X_j\beta\} / \sum_k \exp\{X_k\beta\}. \tag{1.5}$$

Such a model is called multinomial logit. The problem with multinomial logit is that the independence assumption for the errors is very restrictive. One can read a large literature on the independence of irrelevant alternatives problem caused by the independence of errors assumption. See, for example, Anderson, De Palma, and Thisse(1992).

Alternatively, we could assume that $u \sim N[0, \Omega]$ where $\Omega$ can be written in terms of a small number of parameters. When we assume the error distribution is multivariate normal, the resulting choice probabilities are called multinomial probit. For this case, the parameters to estimate are $\theta = (\beta, \Omega)$.[2] The choice probabilities are

$$\Pr[y_j = 1 \mid X] = \int_{u_1} \cdots \int_{u_J} \Pr[X_j\beta + u_j > X_k\beta + u_k \; \forall k \neq j] \, dF(u \mid \Omega) \tag{1.6}$$

[2] Some restrictions are required for $\Omega$ for identification. See, for example, Bunch(1991).

where $F(u \mid \Omega)$ is the joint normal distribution of $u$ with covariance matrix $\Omega$ (with individual elements $\omega_{jk}$). Let $u^*_{jk} = u_k - u_j$ for all $k \neq j$, and let $u^*_j = \left(u^*_{j1}, u^*_{j2}, .., u^*_{jj-1}, 0, u^*_{jj+1}, .., u^*_{jJ}\right)'$. Then the $J$-dimensional integral in equation (1.6) can be written as a $J-1$-dimensional integral:

$$\Pr[y_j = 1 \mid X] = \int_{u^*_{1j}} \cdots \int_{u^*_{Jj}} \Pr\left[X_j\beta - X_k\beta > u^*_{jk}\ \forall k \neq j\right] dF^*\left(u^*_j \mid \Omega^*_j\right) \qquad (1.7)$$

where $F^*\left(u^*_j \mid \Omega\right)$ is the joint normal distribution of $u^*_j$ : $u^*_j \sim N\left[0, \Omega^*_j\right]$ where $\omega^*_{jkl} = E(u_k - u_j)(u_l - u_j) = \omega_{kl} - \omega_{kj} - \omega_{jl} + \omega_{jj}$ for each element $\omega^*_{jkl}$ of $\Omega^*_j$. Equation (1.7) can be written as

$$\Pr[y_j = 1 \mid X] = \Pr\left[u^*_j < V_j\right] \qquad (1.8)$$

where $V_j$ is a vector with $k$'th element equal to $V_{jk} = X_j\beta - X_k\beta$.

In order to make progress in estimating $\theta$, we need to be able to evaluate equation (1.8) for any $\Omega^*_j$ and any $V_j$. For example, the MLE of $\theta$ maximizes

$$\frac{1}{N}\sum_i y_{ij} \log \Pr\left[u^*_{ij} < V_{ij}\right] \qquad (1.9)$$

where $i$ indexes observations, $i = 1, 2, .., N$. If $J = 3$, then equation (1.8) involves evaluating a bivariate normal probability; most computers have library routines to perform such a calculation. If $J = 4$, then equation (1.8) involves a 3-dimension integral. One can evaluate such an integral using Gaussian quadrature (see Butler and Moffitt, 1982) or the numerical algorithm in Hausman and Wise(1978). But, if $J > 4$, numerical routines will be cumbersome and frequently imprecise.

Simulation provides an alternative method for evaluating equation (1.8). The simplest simulator of equation (1.8) is

$$\frac{1}{R}\sum_{r=1}^{R} 1\left(u_j^{*r} < V_j\right) \tag{1.10}$$

where 1(.) is an indicator function equal to one if the condition inside is true and equal to zero otherwise, and $u_j^{*r}$ is an *iid* draw from $N[0, \Omega_j]$. Essentially, the simulator in equation (1.10) draws a random vector from the correct distribution and then checks whether that random vector satisfies the condition, $u_j^* < V_j$. The simulator in equation (1.10) is called a frequency simulator. It is unbiased and bounded between zero and one. But its derivative with respect to $\theta$ is either undefined or zero because the simulator is a step function; this characteristic makes it difficult to estimate $\theta$ and to compute the covariance matrix of $\hat{\theta}$. Also, especially when $\Pr[y_j = 1 \mid X]$ is small, the frequency simulator has a significant probability of equaling zero; since MLE requires evaluating $\log\Pr[y_j = 1 \mid X]$, this is a significant problem. The simulators discussed in Section 2 suggest ways to simulate $\Pr[y_j = 1 \mid X]$ with small variance, with derivatives, and in computationally efficient ways.

## 1.2 Unobserved Heterogeneity

The second example involves unobserved heterogeneity in a nonlinear model. Let $y_{it}$ be a random count variable; i.e., $y_{it} = 0, 1, 2, \ldots$, with $i = 1, 2, .., N$ and $t = 1, 2, .., T$. Assume that $y_{it} \sim$ Poisson $(\lambda_{it})$:

$$f(y_{it} \mid \lambda_{it}) = \exp\{-\lambda_{it}\} \lambda_{it}^{y_{it}} / y_{it}! \tag{1.11}$$

and that

$$\log \lambda_{it} = X_{it}\beta + u_i + e_{it} \tag{1.12}$$

where $u_i \sim iidG\left(. \mid \alpha_G\right)$, $G\left(. \mid \alpha_G\right)$ is a specified distribution up to a set of parameters $\alpha_G$,

$$e_{it} = \rho e_{it-1} + \epsilon_{it}, \tag{1.13}$$

$\epsilon_{it} \sim iidH\left(. \mid \alpha_H\right)$, and $H\left(. \mid \alpha_H\right)$ is a specified distribution up to a set of parameters $\alpha_H$.[3] For example, $y_{it}$ might be the number of trips person $i$ takes in period $t$, the number of patents firm $i$ produces in year $t$, or the number of industrial accidents firm $i$ has in year $t$. Adding the unobserved heterogeneity $u_i$ and serially correlated error $e_{it}$ allows for richness frequently necessary to explain the data. The goal is to estimate $\theta = (\beta, \rho, \alpha_G, \alpha_H)$. The log likelihood contribution of observation $i$ is

$$L_i = \log \int_{u_i} \int_{\epsilon_{i1}} \cdots \int_{\epsilon_{iT}} \prod_{t=1}^{T} \left[\exp\left\{-\lambda_{it}\right\} \lambda_{it}^{y_{it}} / y_{it}! dH\left(\epsilon_{it} \mid \alpha_H\right)\right] dG\left(u_i \mid \alpha_G\right) \tag{1.14}$$

where $\lambda_i = \left(\lambda_{i1}, \lambda_{i2}, .., \lambda_{iT}\right)'$ depends upon $X_{it}\beta$, $u_i$, and $\epsilon_i = \left(\epsilon_{i1}, \epsilon_{i2}, .., \epsilon_{iT}\right)'$ through equations (1.12) and (1.13). When there is no serial correlation term $e_{it}$, the integral in equation (1.14) can be solved analytically for well chosen $G\left(u_i \mid \alpha_G\right)$.[4] But for general $G\left(. \mid \alpha_G\right)$ and $H\left(. \mid \alpha_H\right)$, the integral can be evaluated neither analytically nor numerically.

Simulating the integral is quite straightforward. Let $\epsilon_i^r$ be an *iid* pseudorandom draw of $\epsilon_i$, $r = 1, 2, .., R$. Similarly, let $u_i^r$ be an *iid* random draw of $u_i, r = 1, 2, .., R$. Then $L_i$ can be simulated by evaluating the integrand for each draw $r$ and taking an average:

$$\hat{L}_i = \log \left\{ \frac{1}{R} \sum_{r=1}^{R} \left[ \prod_{t=1}^{T} \exp\left\{-\lambda_{it}^r\right\} \left(\lambda_{it}^r\right)^{y_{it}} / y_{it}! \right] \right\} \tag{1.15}$$

---

[3] One might want to specify a different distribution for $e_{i0}$ because of an initial conditions problem.

[4] See Hausman, Hall, and Griliches(1984).

where $\lambda_{it}^r$ is evaluated using the pseudorandom draws of $\epsilon_i$ and $u_i$ in equation (1.12). The simulated maximum likelihood estimator of $\theta$ maximizes $\sum_i \hat{L}_i$. Note that even though $\exp\left\{\hat{L}_i\right\}$ is unbiased, $\hat{L}_i$ is biased for finite $R$ (because $\hat{L}_i$ is a nonlinear function of $\exp\left\{\hat{L}_i\right\}$). This will cause $\hat{\theta}$ to be inconsistent unless $R \to \infty$ as $NT \to \infty$. However, Monte Carlo results discussed later show that the asymptotic bias is small as long as "good" simulators are used.

## 1.3 Monte Carlo Experiments

The last example is a Monte Carlo experiment. Let $U$ be a vector of data and $s(U)$ be a proposed statistic that depends upon $U$. The statistic $s(U)$ may be an estimator or a test statistic. In general, the user will want to know the distribution of $s(U)$. But, for many statistics $s(.)$, deriving the small sample properties of $s(U)$ is not possible analytically. Simulation can by used to learn about the small sample properties of $s(U)$. All moments of $s(U)$ can be written in the form $Eh(U)$.[5] Medians and, in fact, the whole distribution of $s(U)$ can be written in the form $Eh(U)$. Monte Carlo experiments are powerful tools to use in evaluating statistical properties of $s(U)$. However care must be taken in conducting such experiments. In particular, one must be careful in generalizing Monte Carlo results to cases not actually simulated; a Monte Carlo experiment really only provides information about the specific case simulated. Also, one must be careful not to attempt simulating objects that do not exist. For example, simulating the expected value of a two stage least squares (2SLS) estimator would provide an answer (because any particular draw of $s(U)$ is finite) but it would be meaningless because 2SLS estimators have no finite moments. See Hendry(1984)

[5] For $Es(U)$, $h(U) = s(U)$, and for $\text{Var}[s(U)]$, $h(U) = [s(U) - Es(U)]^2$.

for more on Monte Carlo experiments.

# 2 Simulators

This section discusses various simulation methods. Throughout, the goal will be to simulate $Eh(U)$ or, in some special cases, $\Pr\{y_j = 1 \mid X\}$. The first requirement of a simulation method is to simulate $U$ from its distribution $F$. In general, if $Z \sim$ Uniform $(0,1)$, then $F^{-1}(Z) \sim F$.[6] For example, the exponential distribution is $F(x) = 1 - \exp\{-\lambda x\}$. Thus, $-\log(1-Z)/\lambda \sim F$. If $F$ is standard normal, then $F^{-1}$ has no closed form, but most computers have a library routine to approximate $F^{-1}$ for the standard normal distribution. Truncated random variables can be simulated in the same way. For example, assume $U \sim N[\mu, \sigma^2]$ but let it be truncated between $a$ and $b$. Then, since

$$F(u) = \left[\Phi\left(\frac{u-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right] / \left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right] \tag{2.1}$$

where $\Phi$ is the standard normal distribution function, $U$ can be simulated by letting $F(u) = Z$ in equation (2.1) and solving equation (2.1) for $u$ as

$$\sigma\Phi^{-1}\left\{Z\left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right] + \Phi\left(\frac{a-\mu}{\sigma}\right)\right\} + \mu. \tag{2.2}$$

This idea can be applied with a small twist to discrete random variables. Assume $U = i$ with probability $p_i$ for $i = 1, 2, .., n$. Let $P_i = \Pr[U \le i] = \sum_{j=1}^{i} p_j$. Let $Z \sim$ Uniform $(0,1)$, and let $U = i$ iff $P_{i-1} < Z \le P_i$ (where $P_0 = 0$). Then $U$ is distributed as desired.

[6] Most computers have a library routine to generate standard uniform random variables. See, for example, Ripley (1987) for a discussion of standard uniform random number generators.

Random variables frequently can be simulated by using a composition formula. For example, since a binomial random variable is the sum of independent Bernoulli random variables, we can simulate a binomial random variable by simulating independent Bernoulli's and then adding them up. A more useful example is simulating multivariate $U \sim N[\mu, \Omega]$. Let $Z \sim N[0, I]$, and let $C$ be any matrix such that $CC' = \Omega$ (e.g., the Cholesky decomposition of $\Omega$). Then it is easy to verify that $CZ+\mu \sim N[\mu, \Omega]$. So we can simulate $U$ by simulating $Z$ and then transforming it.

In some cases, it will be necessary to simulate a random variable conditional on some event where the inverse conditional distribution has no analytical form (or good approximation). There are a number of acceptance- rejection methods available for many such cases. Assume $(\bar{U}, Z)$ have joint distribution $F(u, z)$ and that it is straightforward to draw $(U, Z)$ from its joint distribution. Further, assume we want to draw $U$ conditional on $Z \in S$ where $S$ is a subset of the support of $Z$. The simplest acceptance-rejection simulation method is:

(a) Draw $(U, Z)$ from $F$.

(b) If $Z \notin S$, go to (a).

(c) If $Z \in S$, keep.

There are more sophisticated methods that reduce the expected number of draws of $(U, Z)$ needed (see, for example, Devroye(1986) or Ripley(1987)), but all acceptance-rejection simulation methods suffer from a) the potentially large number of draws needed and b) the lack of differentiability of $Eh(U)$ with respect to parameter vector $\theta$.[7] Thus, for the most

[7] Differentiability is important for estimation.

part, they should be avoided. For the remainder of the chapter, it will be assumed one can simulate $U$.

The most straightforward simulator for $Eh(U)$ is

$$\hat{E}h(u) = \frac{1}{R}\sum_{r=1}^{R} h(u^r) \tag{2.3}$$

where $u^r$, $r = 1, 2, .., R$ are $R$ *iid* pseudorandom draws of $U$. When simulating $\Pr[y_j = 1 \mid X]$, equation (2.3) becomes equation (1.10). If $h$ is continuous and differentiable with respect to $\theta$, then $\hat{E}h(U)$ will be continuous and differentiable. Equation (2.3) is unbiased, and its variance is $\text{Var}[h(U)]/R$. Note that as $R \rightarrow \infty$, the variance of the simulator $\rightarrow$ zero.

## 2.1 Importance Sampling

Several methods allow us to improve the performance of a simulator significantly either in terms of reduced variance, better smoothness properties, and/or better computation time properties. The rest of this section describes the most popular simulation methods. The first method is importance sampling. Consider $Eh(U)$ in equation (1.1) where it is either difficult to draw $U$ from $f$ or where $h$ is not smooth. In some cases, one can rewrite equation (1.1) as

$$Eh(U) = \int \frac{h(u)}{g(u)} g(u)\, du \tag{2.4}$$

where $g(u)$ is a density with the following properties:

a) it is easy to draw $U$ from $g$,

b) $h$ and $g$ have the same support,

c) it is easy to evaluate $h(u)/g(u)$ given $u$, and

d) $h(u)/g(u)$ is bounded and smooth over the support of $U$.

Note that equation (2.4) is $E[h(U)/g(U)]$ where $U \sim g$. Then the importance sampling simulator for $Eh(U)$ is

$$\hat{E}h(u) = \frac{1}{R}\sum_{r=1}^{R}\frac{h(u^r)}{g(u^r)} \tag{2.5}$$

where $u^r$, $r = 1, 2, ..., R_1$ are $R$ *iid* draws from $g$. The purpose of conditions (a) and (c) are to increase computational speed. The purpose of condition (d) is variance bounding and smoothness.

Consider simulating $\Pr[y_j = 1 \mid X]$ for the multinomial probit problem. Equation (1.8) can be written as

$$\int\limits_{u_j^* < V_j} f(u_j^*)\, du_j^* = \int\limits_{u_j^* < V_j} \left[f(u_j^*)/g(u_j^*)\right] g(u_j^*)\, du_j^* \tag{2.6}$$

for some multivariate density $g$ satisfying Conditions (a) through (d). Consider $g$ where the $k$th element of $u_j^*$ is distributed independently truncated normal with upper truncation point $V_{jk}$ and variance $\Omega_{jkk}^*$ for each $k$. The candidate $g$ satisfies Conditions (a), (b), and (c), and $h(u)/g(u)$ is smooth over the support $u_j^* < V_j$. But $h(u)/g(u)$ is not bounded especially when $\Omega^*$ has large off-diagonal terms. Thus, this choice of g may be problematic. In fact, in general it is the boundedness condition that is difficult to satisfy. For the multinomial probit problem, the Geweke-Keane-Hajivassilliou (GHK) and decomposition simulators discussed below both can be thought of as importance sampling simulators that satisfy Conditions (a) through (d).

## 2.2 GHK Simulator

The GHK simulator, developed by Geweke(1991), Hajivassilliou(1990), and Keane(1994) has been found to perform very well in Monte Carlo studies (discussed later) for simulating $\Pr\left[u_j^* < V_j\right]$. The GHK algorithm switches back and forth between computing univariate, truncated normal probabilities, simulating draws from univariate normal distributions, and computing normal distributions conditional on previously drawn truncated normal random variables. Since each step is straightforward and fast, the algorithm can decompose the more difficult problem into a series of feasible steps. The algorithm is as follows:

(a) Set $t = 1$, $\mu = 0$, $\sigma^2 = \Omega_{jtt}^*$, and $\hat{P} = 1$.

(b) Compute $p = \Pr\left(u_{jt}^* < V_{jt}\right)$ analytically, and increment $\hat{P} = \hat{P} * p$.

(c) Draw $u_{jt}^*$ from a truncated normal distribution with mean $\mu$, variance $\sigma^2$ and upper truncation point $V_{jt}$.

(d) If $t < J - 1$, increment $t$ by 1; otherwise goto (g).

(e) Compute (analytically) the distribution of $u_{jt}^*$ conditional on $u_{j1}^*, u_{j2}^*, .., u_{jt-1}^*$. Note that this is normal with an analytically computable mean vector $\mu$ and variance $\sigma^2$.

(f) Goto (b).

(g) $\hat{P}$ is the simulator.

The algorithm relies upon the fact that normal random variables conditional on other normal random variables are still normal. The GHK simulator is strictly bounded between zero and one because each increment to $\hat{P}$ is strictly bounded between zero and one. It is continuous and differentiable in $\theta$ because each increment to $\hat{P}$ is continuous and differentiable. Its variance is smaller than the frequency simulator in equation (1.10) because each draw of $\hat{P}$ is strictly bounded between zero and one while each draw of the frequency simulator is either zero or one.

A minor modification of the algorithm provides draws of normal random variables $u_j^*$ conditional on $u_j^* \leq V_j$. Other minor modifications are useful for related problems.

## 2.3 Decomposition Simulators

Next, two decomposition simulators are described. The Stern(1992) simulator uses the property that the sum of two normal random vectors is also normal. The goal is to simulate $\Pr\left[u_j^* < V_j\right]$. Decompose $u_j^* = Z_1 + Z_2$ where $Z_1 \sim N[0,\lambda], Z_2 \sim N\left[0, \Omega_j^* - \lambda\right], Z_1$ and $Z_2$ are independent, and $\lambda$ is chosen to be a diagonal matrix as large as possible such that $\Omega_j^* - \lambda$ is positive definite.[8] Then equation (1.8) can be written as

$$\begin{aligned} &\int \Pr\left[Z_1 < V_j - z_2\right] g(z_2)\, dz_2 \\ &= \int \prod_k \Phi\left(\tfrac{V_{jk} - z_{2k}}{\lambda_k}\right) g(z_2)\, dz_2 \end{aligned} \tag{2.7}$$

where $g(.)$ is the joint normal density of $Z_2$. Equation (2.7) can be simulated as

$$\frac{1}{R}\sum_{r=1}^{R}\prod_k \Phi\left(\frac{V_{jk} - z_{2k}^r}{\lambda_k}\right) \tag{2.8}$$

where $z_{2k}^r, k = 1, 2, .., J-1$, are pseudorandom draws of $Z_2$. The Stern simulator has all of the properties of the GHK simulator. So which one performs better is an empirical matter left to later discussion.

[8] An easy way to pick $\lambda$ is to set each diagonal element of $\lambda$ equal to the smallest eigenvalue of $\Omega_j^*$ minus a small amount.

Another decomposition simulator, suggested by McFadden(1989), changes the specification of equation (1.3) to

$$y_j^* = X_j\beta + u_j + \tau e_j,\ j = 1,..,J \tag{2.9}$$

where $\tau$ is a small number and $e_j \sim iid$ Extreme Value. In the limit, as $\tau \to 0$, $\Pr[y_j = 1 \mid X]$ converges to a multinomial probit probability. But for any $\tau > 0$,

$$\Pr[y_j = 1 \mid X] = \int \exp\left\{\frac{X_j\beta + u_j}{\tau}\right\} / \sum_k \exp\left\{\frac{X_k\beta + u_k}{\tau}\right\} f(u)\, du \tag{2.10}$$

which is the multinomial logit probability conditional on $u = (u_1, u_2, .., u_J)$ integrated over $f$. Equation (2.10) can be simulated as

$$\frac{1}{R}\sum_{r=1}^{R}\left[\exp\left\{\frac{X_j\beta + u_j^r}{\tau}\right\} / \sum_k \exp\left\{\frac{X_k\beta + u_k^r}{\tau}\right\}\right] \tag{2.11}$$

where $u^r$ are pseudorandom draws of $u$. The idea in McFadden(1989) is to think of equation (2.9) as a kernel-type approximation of equation (1.3) for small $\tau$. However, assuming equation (2.9) is the true structure (where $\tau$ is a parameter that can sometimes be estimated) takes away no flexibility and frequently eases simulation. Multivariate normality is a desirable assumption because of its flexible covariance matrix. But there are very few applications where theory dictates that the error in equation (1.3) should be multivariate normal. Berkovec and Stern(1991) use the McFadden specification as the "true" specification in a structural model of retirement behavior.

## 2.4 Antithetic Acceleration

Antithetic acceleration is a powerful variance reduction method (see Geweke, 1988). In any simulation method, there is some probability that the pseudorandom draws will be unusually large (or small). Antithetic acceleration prevents such events from occurring and thus reduces the variance of the simulator. Consider the general problem of simulating $Eh(U)$ where $U \sim F$. Let $Z \sim$ Uniform $(0,1)$. Then $h\left(F^{-1}(Z)\right)$ is a simulator of $Eh(U)$. But $h\left(F^{-1}(1-Z)\right)$ is also a simulator of $Eh(U)$ (because $1 - Z \sim$ Uniform $(0,1)$ also). The antithetic acceleration simulator of $Eh(U)$ is

$$\hat{E}h(u) = \frac{1}{2R}\sum_{r=1}^{R}\left[h\left(F^{-1}(z^r)\right) + h\left(F^{-1}(1-z^r)\right)\right] \tag{2.12}$$

where $z^r$ is a pseudorandom draw of $Z$. When $F$ is $N\left[0,\sigma^2\right]$, equation (2.12) becomes

$$\hat{E}h(u) = \frac{1}{2R}\sum_{r=1}^{R}\left[h(u^r) + h(-u^r)\right] \tag{2.13}$$

where $u^r$ is a pseudorandom draw of $U$. For any $F$, if $h$ is linear, the variance of $\hat{E}h(U)$ is zero. For monotone $h$, the variance of $\hat{E}h(U)$ with $R$ draws and antithetic acceleration is smaller than the variance of $\hat{E}h(U)$ with $2R$ draws and no antithetic acceleration. If $Eh(U)$ is being simulated to estimate a parameter $\theta$ with $N$ observations and $h$ is monotone, then the increase in $\mathrm{Var}\left(\hat{\theta}\right)$ due to simulation when antithetic acceleration is used is of order $(1/N)$ times the increase in $\mathrm{Var}\left(\hat{\theta}\right)$ due to simulation when antithetic acceleration is not used. The value of this is discussed more in the next section.

There are simulation problems where antithetic acceleration does not help. For example, let $U \sim N\left[0,\sigma^2\right]$, and let $h(U) = U^2$. Then $\mathrm{Var}\left[\hat{E}h(U)\right]$ with antithetic acceleration and $R$ draws is greater than that without antithetic acceleration and $2R$ draws. This is because $h(-U) = h(U)$ which means that equation (2.13) becomes equation (2.3); the variance is twice as great as with no antithetic acceleration and $2R$ draws. In general, deviations from monotone $h$ will diminish the performance of antithetic acceleration. But Hammersly and Handscomb(1964) suggests generalizations of antithetic acceleration that will reduce variance for more general $h$.

# 3 Estimation Methods

The goal of this section is to use the simulators developed in the last section in some estimation problems. Four different estimation methods are discussed: method of simulated moments (MSM), simulated maximum likelihood estimation (SML), method of simulated scores (MSS), and Gibbs sampling. Each method is described, and its theoretical properties are discussed.

## 3.1 Method of Simulated Moments

Many estimation problems involve finding a parameter vector $\theta$ that solves a set of orthogonality conditions

$$EQ'h(y, X \mid \theta) = 0 \tag{3.1}$$

where $Q$ is a set of instruments. Such estimators are called method of moments (MOM) estimators. All least squares methods are special cases of equation (3.1), and many problems usually estimated as MLE can be recast as MOM estimators. For example, Avery, Hansen, and Hotz(1983) suggest how to recast the the multinomial probit problem as a MOM problem where $h(y, X \mid \theta)$ is the vector $y - E(y \mid X)$ in the multinomial probit problem of Section 1 with $j$th element given by equation (1.4).

In many MOM problems, the orthogonality condition can not be evaluated analytically.

For example, in the multinomial probit problem, evaluating $E[y \mid X]$ involves evaluating equation (1.4). MSM replaces $h(y, X \mid \theta)$ with an unbiased simulator $\hat{h}(y, X \mid \theta)$ and then finds the $\theta$ that solves

$$Q'\hat{h}(y, X \mid \theta) = 0. \tag{3.2}$$

The $\theta$ that solves equation (3.2) is the MSM estimator of $\theta, \hat{\theta}$. McFadden(1989) and Pakes and Pollard(1989) show that, as long as $\hat{h}(y, X \mid \theta)$ is an unbiased simulator of $h(y, X \mid \theta)$, deviations between $\hat{h}$ and $h$ will wash out by the Law of Large Numbers because equation (3.2) is linear in $\hat{h}$ and $\text{plim}(\hat{\theta}) = \theta$ as the sample size $N \to \infty$ even for small $R$.[9]

Consider the multinomial probit problem in more detail. As in Section 1, let $y_i$ be the vector of dependent variables for observation $i, i = 1, 2, .., N$, where $y_{ij} = 1$ iff choice $j$ is chosen by $i$. The probability of $i$ choosing $j$ conditional on $X_i$ is given in equation (1.8), and its frequency simulator is given in equation (1.10). The frequency simulator should be replaced by one of the simulators discussed in Section 2, but for now we will use the frequency simulator for ease of presentation. As was discussed earlier, $E[y_{ij} \mid X_i] = \Pr[y_{ij} = 1 \mid X_i]$. Let $P_i$ be a $J$-element vector with $\Pr[y_{ij} = 1 \mid X_i]$ in the $j$th element of $P_i$, and let $\epsilon_i = y_i - P_i$. Then $E[\epsilon_i \mid X_i] = 0$, and

$$E \sum_i Q_i' \epsilon_i = 0 \tag{3.3}$$

for any set of exogenous instruments $Q_i$. Thus, conditional on a chosen $Q = (Q_1, Q_2, .., Q_N)$, the $\theta = (\beta, \Omega)$ that satisfies equation (3.3) is the MOM estimator of $\theta$. Let $\hat{P}_i$ be an unbiased

[9] Extra conditions are found in McFadden(1989) and Pakes and Pollard(1989).

simulator of $P_i$, and let $\hat{\epsilon}_i = y_i - \hat{P}_i$. Then the $\theta$ that solves

$$\sum_i Q_i' \hat{\epsilon}_i = 0 \tag{3.4}$$

is the MSM estimator of $\theta$.

To find a reasonable $Q$, consider the log likelihood contribution for the multinomial probit model:

$$L_i = \sum_j y_{ij} \log P_{ij} \tag{3.5}$$

The score statistics for $\theta$ can be written as

$$\begin{aligned} \partial L_i / \partial \theta &= \sum_j y_{ij} \frac{\partial P_{ij}/\partial \theta}{P_{ij}} \\ &= \sum_j \frac{\partial P_{ij}/\partial \theta}{P_{ij}} \left( y_{ij} - P_{ij} + P_{ij} \right) \\ &= \sum_j \frac{\partial P_{ij}/\partial \theta}{P_{ij}} \left( y_{ij} - P_{ij} \right) + \sum_j \frac{\partial P_{ij}}{\partial \theta} \end{aligned} \tag{3.6}$$

where the last term equals zero because the $\sum_j P_{ij} = 1$. Thus, one can write the score statistics in the form of equation (3.4). With an initial estimate of $\theta$, one can construct $(1/P_{ij})(\partial P_{ij}/\partial \theta)$ for $\theta$ and all $j$ and use it as an instrument matrix $Q_i$ for each $i$. It is likely that the instruments $Q$ will need to be simulated (e.g., if the elements of $Q_i$ are $(1/P_{ij})(\partial P_{ij}/\partial \theta)$). This presents no significant problems as long as the pseudorandom variables used to simulate $Q_i$ are independent of those used in the estimation process (to ensure exogeneity). For any exogenous $Q$, the $\hat{\theta}$ that solves equation (3.4) is a consistent estimate of $\theta$. Thus, once $\theta$ is estimated, $Q$ can be updated using $\hat{\theta}$ and then used to find a new $\hat{\theta}$ that solves equation (3.4).

For any exogenous $Q$, the covariance matrix of $\hat{\theta}$ has two terms: a term due to random variation in the data and a term due to simulation. As long as $\hat{P}_i$ is an exogenous, unbiased simulator of $P_i$, one can write

$$\hat{P}_i = P_i + \xi_i \tag{3.7}$$

where $\xi_i$ is a random variable caused by simulation with zero mean independent of $\epsilon_i$, the deviation between $y_i$ and $P_i$. Thus, the covariance matrix of $\hat{\epsilon}_i$ can be written as $E\epsilon\epsilon' + E\xi\xi'$. If $\hat{P}_i$ is the frequency simulator of $P_i$, then $\xi$ is just an average of $R$ independent pseudorandom variables each with the same covariance matrix as $\epsilon$. Thus, the covariance matrix of $\hat{\epsilon}$ is the covariance matrix of $\epsilon$ times $[1 + R^{-1}]$. The asymptotic covariance matrix of $\theta$ is a linear function of the covariance matrices for $\hat{\epsilon}_i, i = 1, 2, .., N$ (McFadden, 1989, p. 1006). Note that for any $R \geq 1$, $\hat{\theta}$ is consistent; that as $R \rightarrow \infty$, the MSM covariance matrix approaches the MOM covariance matrix (which is efficient when the two-step procedure described above is used); and that the marginal improvement in precision declines rapidly in $R$. If an alternative simulator with smaller variance is used, then the loss of precision due to simulation declines. For example, if antithetic acceleration is used, then the loss in precision becomes of order $(1/N)$ which requires no adjustment to the covariance matrix.

Below is a roadmap for using MSM to estimate multinomial probit parameters:

a) Choose an identifiable parameterization for $\Omega$ and initial values for $\theta = (\beta, \Omega)$. Make sure that the initial guess results in probabilities reasonably far from zero or one.

b) Choose a simulator.

c) Simulate 2NJR[10] standard normal random variables. Store NJR of them in an instruments random number file and NJR in an estimation random number file. These random numbers

[10] Remember that $N$ = sample size, $J$ = number of choices, and $R$ = number of draws.

will be used throughout the estimation process and never changed.

d) Given the initial guess of $\theta$ and the instruments random number file, simulate $Q$. Store the simulated instruments.

e) Given the initial guess of $\theta$, the simulated $Q$, and the estimation random number file, solve equation (3.4) for $\theta$. This is an MSM estimator of $\theta$.

f) Given the initial MSM estimator, reperform steps (d) and (e) once.

Solving equation (3.4) requires using an optimization algorithm to find the $\theta$ that minimizes

$$\sum_i \hat{\epsilon}' Q_i Q_i' \hat{\epsilon}_i \tag{3.8}$$

The derivatives of $P_i$ are well behaved, so derivative based optimization routines should be used. At each guess of $\theta$, the standard normal pseudorandom numbers in the estimation random number file are used to create a new set of $N[0, \Omega]$ random numbers using the method described in Section 2. Thus, even though the standard normal random numbers never change, one is always using random numbers from the correct normal distribution.

Consider the unobserved heterogeneity count problem described in equations (1.11) through (1.13). Let $y_{it}$ be the number of events for $i$ at time $t$. $E[y_{it} \mid \lambda_{it}]$ is $\lambda_{it}$, but the covariance matrix of $y_i$ has no closed form. Let $v_i$ be a vector of residuals with $[T + T(T+1)/2]$ elements. The first $T$ elements of $v_i$ are $y_{it} - E\lambda_{it}$ for $t = 1, 2, .., T$ where the expectation is over $e_{it}$ and $u_i$ in equation (1.12). The last $T(T+1)/2$ elements correspond to "covariance residuals." A representative element would be

$$(y_{it} - E\lambda_{it})(y_{is} - E\lambda_{is}) - C_{its} \tag{3.9}$$

for two periods, $t$ and $s$, where $C_{its}$ is the $\text{Cov}(y_{it}, y_{is})$. The MOM estimator of $\theta = (\beta, \rho, \sigma_G, \sigma_H)$ solves

$$\sum_i Q_i' v_i = 0 \tag{3.10}$$

given a set of instruments $Q$. Since both $E\lambda_{it}$ and $C_{its}$ can not be evaluated analytically,[11] the MOM estimator is not feasible. But $E\lambda_{it}$ and $C_{its}$ can be simulated. Let $\hat{y}_{it}^r$ be a simulated count variable. We can simulate $e_{it}$ and $u_i$ and therefore $\lambda_{it}$. Conditional on the simulated $\lambda_{it}$, we can simulate $y_{it}$ either directly or by using the relationship between Poisson random variables and exponential random variables.

## 3.2 Simulated Maximum Likelihood

A common estimation method with good optimality properties is maximum likelihood (ML) estimation. The basic idea is to maximize the log likelihood of the observed data over the vector of estimated parameters. ML estimators are consistent and efficient for a very large class of problems. Their asymptotic distribution is normal for a slightly smaller class of problems. However there are many likelihood functions that can not be evaluated analytically. In many cases, they can be thought of as expected values of some random function that can be simulated.

Consider again the multinomial probit problem. The log likelihood contribution for observation i is defined in equation (3.5). Note that only one element of $y_i$ is not zero, so only one probability needs to be computed. This is a significant advantage of simulated

[11] Under special assumptions about the distribution of $u_i$ and $e_{it}$ described in Hausman, Hall, and Griliches (1984), the moments have analytical forms.

maximum likelihood (SML) over MSM. Still, to evaluate the log likelihood function, one must be able to evaluate or simulate $P_{ij}$ for the choice chosen. The SML estimator of $\theta$ is the value of $\theta$ that maximizes

$$L = \sum_{i=1}^{N} \sum_{j} y_{ij} \ln \hat{P}_{ij} \tag{3.11}$$

where $\hat{P}_{ij}$ is the simulated value of $P_{ij}$.

A significant problem with SML is that the log likelihood function is not linear in $\hat{P}$. Thus, unlike MSM, the simulation errors, $\hat{P} - P$, will not wash out asymptotically as $N \to \infty$ unless $R \to \infty$ also. Lerman and Manski(1981) suggested using SML with a frequency simulator. They found that $R$ needed to be quite large to deal with this problem. However, Borsch-Supan and Hajivassilliou(1993) show in Monte Carlo studies that if better simulators are used, in particular smooth, smaller variance simulators bounded away from zero and one, then the bias caused by finite $R$ is small for moderate sized $R$. In fact, in their study, SML performs better than MSM.

Consider the unobserved heterogeneity model described in equations (1.11) through (1.13). The log likelihood contribution for observation i is given in equation (1.14). The argument of the log is the expected value of

$$\prod_{t=1}^{T} \left[\exp\{-\lambda_{it}\} \lambda_{it}^{y_{it}} / y_{it}!\right] \tag{3.12}$$

over the distribution of the errors determining $\lambda_{it}$. One can simulate $\lambda_{it}$ for each $i$ and $t$ and therefore the expected value of the term in equation (3.12). Since the simulator of $L_i$ is the log of this term, it is biased, and the bias disappears only as $R \to \infty$. But the simulator of equation (3.12) is smooth, and antithetic acceleration can be used to significantly reduce the variance. Thus the asymptotic bias associated with simulating the log likelihood function should be small.

## 3.3 Method of Simulated Scores

A property of maximum likelihood is that the score statistic, the derivative of the log likelihood function, should have an expected value of zero at the true value of $\theta$. This idea is the motivation behind the method of simulated scores (MSS). The potential advantage of MSS is to use an estimator with the efficiency properties of ML and the consistency properties of MSM. The difficulty in this method is to construct an unbiased simulator of the score statistic. The problems this causes will become clear in the multinomial probit example. The log likelihood contribution of observation i is given in equation (3.5), and its derivative is

$$\begin{aligned} \partial L_i/\partial\theta &= \textstyle\sum_j y_{ij} \frac{\partial P_{ij}/\partial\theta}{P_{ij}} \\ &= \frac{\partial P_{ij}/\partial\theta}{P_{ij}} \end{aligned} \tag{3.13}$$

for the $j$ corresponding to the chosen alternative. The goal is to construct an unbiased simulator for equation (3.13) so that the problem can be turned into a MSM problem. While it is straightforward to construct an unbiased simulator for both the numerator and denominator in equation (3.13), the ratio will not be unbiased as long as the denominator is random.

Consider constructing an unbiased simulator of the ratio. Supressing the $i$ subscript, equation (3.13) can be written as

$$\frac{\partial P_j/\partial\theta}{P_j} = \frac{\partial}{\partial\theta}\int_{A_j} f\left(y^*\right) dy^*/P_j \tag{3.14}$$

where $y^* = (y_1^*, y_2^*, .., y_J^*)$, $f$ is the joint density of $y^*$, and $A_j$ is the subset of the support of $y^*$ where $y_j^* > y_k^*$ for all $k \neq j$. This equals

$$\begin{aligned} \frac{\partial P_j/\partial\theta}{P_j} &= \int\limits_{Aj} \frac{\partial f(y^*)/\partial\theta}{f(y^*)} f(y^*)\, dy^*/P_j \\ &= E\left[\frac{\partial}{\partial\theta} \ln f(y^*) \mid y_j = 1\right] \end{aligned} \tag{3.15}$$

where the expectation is with respect to the joint density of $y^*$. One usually can simulate the expectation in equation (3.15) (e.g., using the GHK simulator) and thus get an unbiased estimator of the ratio.

## 3.4 Gibbs Sampling

The last estimation procedure discussed is quite different than the others in that it is a Bayesian estimator. In general, we have a model specified up to a set of parameters $\theta$, some data $\{(y_i, X_i)\}_{i=1}^N$, and a prior distribution for $\theta$. The goal is to use the data to update the prior distribution to get a posterior distribution for $\theta$. Computing the posterior involves using Bayes rule which usually involves solving a difficult integral, thus making it an intractable problem. The idea in Gibbs sampling is to augment the data with another unobserved variable, let's say $\{y_i^*\}_{i=1}^N$ that has the following properties:

a) the posterior distribution of $y_i^*$ given $(y_i, \theta)$ is easy to simulate from, and

b) the posterior distribution of $\theta$ given $(y_i^*, y_i)$ and the prior distribution of $\theta$ is easy to compute and simulate from.

Assume there is a $\{y_i^*\}_{i=1}^N$ that satisfies these two conditions. Then the Gibbs sampling algorithm draws $\{y_i^*\}_{i=1}^N$ given $\{y_i\}_{i=1}^N$ and $\theta$, then draws $\theta$ given new $\{y_i^*, y_i\}_{i=1}^N$, and repeats this process over and over again. The draws of $\theta$ provide information about the posterior distribution of $\theta$. The algorithm is:

(a) Assume a prior distribution for $\theta$ conditional on $(y_i, \theta)$. Choose $R_0$ such that the first $R_0$ draws will not count and $R_1$ such that the process will stop after $R_1$ draws. Set $r = 0$.

(b) Simulate one draw of $\theta$ from its posterior distribution.

(c) If $r > R_0$, store the draw of $\theta$ as draw $r - R_0$.

(d) If $r > R_1$, goto (g).

(e) Simulate one draw of $\{y_i^*\}_{i=1}^N$ conditional on $\left(\{y_i\}_{i=1}^N, \theta\right)$.

(f) Evaluate analytically the posterior distribution for $\theta$ given $\{(y_i, y_i^*)\}_{i=1}^N$. Increment $r = r + 1$. Goto (b).

(g) Use the $R_1 - R_0$ draws of $\theta$ as a random sample of draws of $\theta$ and compute any sample characteristics desired.

Markov chain theory implies that the Gibbs sampling algorithm described above will produce a distribution of draws of $\theta$ corresponding to the posterior distribution of $\theta$ conditional on $\{(y_i, X_i)\}_{i=1}^N$. See, for example, Casella and George (1992), Gelfand and Smith(1990), Geman and Geman(1984), and Tanner and Wong(1987) for more about Markov chains.

Consider how Gibbs sampling can be applied to the multinomial logit problem. To simplify exposition, assume we know $\Omega$ and only need to estimate $\beta$. Assume $\beta_1 = 0$ as a normalizing factor. For step (a), we need a prior distribution for $\{\beta_j\}_{j=2}^J$. If we pick $R_0$ big enough and the prior diffuse enough, then the choice of prior will become irrelevant. Thus, pick the prior to be diffuse. The diffuse prior makes it easy to compute posterior distributions for $\{\beta_j\}_{j=2}^{J}$. Next, let $y_i^*$ be the latent variable associated with $y_i$:

$$y_{ik}^* = X_i\beta_k + u_{ik},\ k = 1, 2, ..., J,\ i = 1, 2, .., N \tag{3.16}$$

where $u_i \sim N[0, \Omega]$.

For step (b), we need to simulate $\beta$ from its posterior distribution. Since at any iteration of the algorithm, $\beta$ is normal, we can simulate $\beta$ using the method described in Section 2.

For step (e), we need to simulate $\{y_i^*\}_{i=1}^{N}$ conditional on $\left(\{y_i\}_{i=1}^{N}, \beta\right)$. Since the observations are independent, we need only simulate $y_i^*$ conditional on $(y_i, \beta)$ for each $i = 1, 2, .., N$ separately. Let $j$ be the chosen choice. Then

$$X_i\beta_j + u_{ij} > X_i\beta_k + u_{ik}\ \forall k \neq j \tag{3.17}$$

or

$$u_{ijk}^* < X_i(\beta_j - \beta_k)\ \forall k \neq j \tag{3.18}$$

where $u_{ijk}^* = u_{ik} - u_{ij}$. The errors $u_{ijk}^*\ \forall k \neq j$ can be simulated using the GHK algorithm, and the $y_{ik}^*$ can be constructed $\forall k \neq 1$[12] as

$$y_{ik}^* = X_i\beta_k + u_{ijk}^* - u_{ij1}^*. \tag{3.19}$$

Alternatively, we can use an acceptance-rejection simulator.

For step (f), we need to evaluate the posterior distribution of $\beta$ given $\{y_i^*\}_{i=1}^{N}$. Since $u_{ij} \sim N(0, \omega_{jj})$ for each $i$ and $j$, $y_{ij}^* \sim N[X_i\beta_j, \omega_{jj}]$ which means that computing a posterior distribution for $\beta$ involves running an OLS regression of $y^*$ on $X$.

[12] Recall that choice 1 is the base choice.

For step (g), the sample of $R_1 - R_0$ draws of $\beta$ are distributed from the distribution of $\beta$ conditional on the data (including the dependent variables $\{y_i\}_{i=1}^{N}$). A few notes of caution are in order here. First, the draws of $\beta$ are not independent even though any dependence dies out as the number of draws between two draws becomes large. Thus, we must not compute any statistics that depend upon the ordering of the draws. Second, the draws are conditional on $\{y_i\}_{i=1}^{N}$. This is quite different than what we would expect in classical statistical analysis (where we would condition on only the exogenous variables). The effect of this is that the researcher does not know how the estimator would have behaved had a different realization of the data been observed. This is a fundamental difference between classical estimators and Bayesian estimators. There are other reasonable (and perhaps better) choices for implementing the Gibbs sampler to the multinomial probit problem. The real issues involve also estimating $\Omega$. See McCullough and Rossi(1993) or Albert and Chib(1993) for a much more extensive discussion.

The unobserved heterogeneity count problem is also easily adaptable to Gibbs sampling. The data should be augmented with $\{\lambda_{it}\}_{t=1}^{T}{}_{i=1}^{N}$ and its prior should be normal. Steps (b) and (f) are the same as in the multinomial probit problem. Step (e) involves simulating $\lambda_{it}$ conditional on $(y_{it}, \beta)$ which is not as straightforward. The density of $\lambda_{it}$ conditional on $(y_{it}, \beta)$ is

$$f\left(\lambda_{it} \mid y_{it}, \beta\right) = C\left(y\right) e^{-\lambda_{it}} \lambda_{it}^{y_{it}-1} \phi\left(\frac{\log \lambda_{it} - X_{it}\beta}{\sigma_\lambda}\right) \tag{3.20}$$

where $\sigma_\lambda$ is the standard deviation of the composite error in equation (1.12), $\phi$ is the standard normal density function, and $C\left(y\right)$ is a proportionality constant chosen so that equation

(3.20) integrates to one. One can evaluate the integral of equation (3.20) numerically for each value of $y = 0, 1, ..$ for a finite number of points: $\delta$, $2\delta$,.., $K\delta$ for some small $\delta$. Figure 1 draws the approximate distribution curves for $y = 0, 1, .., 5$, $\delta = .01$, and $K = 1000$. Then one can use the discretized distribution as an approximation to draw $\lambda$ from. This is equivalent to drawing a random point on the vertical axis of Figure 1 (e.g., point A), drawing a horizontal line to the curve corresponding to $y$ (e.g., $B$ when $y = 4$) and choosing $\lambda$ to be the horizontal component of the curve at that vertical point (e.g., point $C$). An alternative would be to use the Metropolis-Hastings algorithm described in, for example, Chib and Greenberg(1994).

# 4 Empirical Comparison of Methods

A number of studies have compared the performance of various simulators and estimation methods especially for the multinomial probit problem. This section summarizes the results of four of those studies and presents some new results focusing on questions that are neglected in the other studies.

Borsch-Supan and Hajivassiliou (1993) compare the GHK simulator to the Stern simulator and a frequency simulator. They present convincing evidence that the GHK simulator has a significantly smaller standard deviation than the other two simulators. They further show that the standard deviation of the GHK simulator is small enough so that it can be used in an SML estimation routine providing parameter estimates with small root mean squared errors (RMSE's). Having a good simulator with small standard deviation for SML is important because, unlike MSM, SML does not provide consistent estimates for fixed $R$.

Hajivassiliou, McFadden, and Ruud (1994) compare ten different simulators (including the Stern simulator, a Gibbs sampler, and a kernel smoothed simulator) in terms of the RMSE of the multinomial probit probability and its derivatives. They consider a large class of $V_j$'s and $\Omega_j^*$'s. They find that the GHK simulator performs the best overall. In particular, it performs well relative to the alternatives when $\Omega_j^*$ displays high correlation terms. They provide no results concerning parameter estimates.

Geweke, Keane, and Runkle (1994a) compare MSM using GHK, SML using GHK, Gibbs sampling, and kernel smoothing. In an unrestricted estimation procedure (including covariance parameters), MSM-GHK and Gibbs sampling dominated SML-GHK. Kernel smoothing was dominated by all methods. In various restricted models, the performance of SML-GHK improved. In general, as more restrictions were placed on the model, the performance of MSM-GHK, SML-GHK, and Gibbs sampling converged. But Gibbs sampling seemed to dominate other methods overall.

Geweke, Keane, and Runkle (1994b) compare MSM-GHK, SML-GHK, and Gibbs sampling in the related multinomial multiple period probit model. They find that Gibbs sampling dominates and MSM-GHK is second. Estimated standard errors are good for Gibbs sampling and MSM-GHK but are downward biased for SML-GHK.

None of these methods compare the computational cost of the alternatives. Computational cost is important because the simulators are essentially a method to reduce computation time; if time was not an issue, we could compute the relevant integrals numerically using arbitrarily precise approximation methods or we could simulate them letting $R$ be an arbitrarily large number. If one method takes twice as much time as another for a given $R$, then a fair comparison requires using different $R$ for each method to produce comparable times. Also none of the methods considers the effect of using antithetic acceleration (AA) despite Geweke's strong theoretical results.

Table 1 presents the results of a small Monte Carlo study. Its results should be interpreted as suggestive of where more work needs to be done. The methods that are compared are MSM-GHK, MSM-Stern, SML-GHK, SML-Stern, Gibbs sampling (with acceptance-rejection), and MSM-KS (kernel smoothing). Three different models are used: a) $\Omega$ is diagonal and $N$ (sample size) = 500, b) $\Omega$ is diagonal and $N = 1000$, and c) $\Omega$ corresponds to an $AR(1)$ process with $\rho = .9$ and $N = 1000$. Except for Gibbs sampling, results are reported with and without AA. RMSE results and average times per estimation procedure are reported.

Kernel smoothing methods performed poorly in terms of RMSE of the simulated multinomial probit probabilities. Also, more importantly, its derivatives with respect to parameters were poorly behaved in that if the bandwidth parameter was small, the derivatives were very volatile (and therefore derivative based optimization algorithms for estimation behaved poorly), and if it was large, parameter bias was very large. Thus kernel smoothing method results are not reported. In terms of RMSE, Gibbs sampling estimators behave reasonably well. But the amount of time involved is an order of magnitude greater than for the MSM and SML estimates.[13] Thus, there are only limited results reported for the Gibbs samplers.

The remainder of the discussion focuses on MSM, SML, GHK, Stern, and AA. First, it is clear that SML dominates MSM in these examples. It provides smaller RMSE's and it requires less computation time. GHK dominates Stern in terms of RMSE, but Stern is significantly faster. One might consider using Stern with twice as large $R$. Unreported Monte Carlo experiments suggest that for the examples used here the standard deviation of the multinomial probit probabilities is about twice as large for the Stern simulator as for the GHK simulator when $R = 10$. This would suggest that doubling $R$ for the Stern simulator (relative to the GHK simulator) would make the GHK simulator more efficient by a factor of $\sqrt{2}$. Thus, these results are consistent with Borsch-Supan and Hajivassiliou, suggesting that SML-GHK provides estimates with the smallest RMSE's even after controlling for variation in computation time. Based on results in Borsch-Supan and Hajivassiliou and Hajivassiliou, McFadden, and Ruud, it probably performs even better for pathological cases with highly correlated errors or small multinomial probit probabilities.

The poor performance of AA is striking. AA almost uniformly improves the performance of the Stern simulator. But it behaves poorly for the GHK simulator. However, unreported results show that AA significantly reduces the standard deviation of the simulated multinomial probit probabilities for GHK, Stern, and kernel smoothing. This apparent paradox occurs because of the small sample properties of method of moments (MOM) and maximum

[13] It should be noted that in these Monte Carlo experiments, I am conditioning on the true value of $\Omega$. It might be the case that the Gibbs sampler performs better relative to the other methods when $\Omega$ also is estimated.

likelihood (MLE). In other words, the RMSE of MOM and MLE dominate any extra randomness caused by simulation. This is verified by unreported results showing that when $R$ is increased to 50, SML-GHK and SML-Stern RMSE's converge to each other with or without AA and they are similar to the RMSE's for the case when $R = 5$ with AA or $R = 10$ without AA. The bottom line is that for MSM and SML, the choice of simulation method has a second order effect on RMSE relative to RMSE caused by the underlying estimation method. This further suggests that computation time issues should be given high priority.

Table 1

| Results for Diagonal Covariance Matrix | | | | |
|---|---|---|---|---|
| N = 500 | | | | |
| | w/ Antithetic Acceleration | | wo/ Antithetic Acceleration | |
| Method | Avg RMSE | Avg Time | Avg RMSE | Avg Time |
| MSM-GHK | 0.299 | 3559.0 | 0.257 | 3373.8 |
| MSM-Stern | 0.270 | 1047.0 | 0.288 | 1097.0 |
| SML-GHK | 0.247 | 1571.0 | 0.246 | 1598.8 |
| SML-Stern | 0.254 | 654.9 | 0.252 | 674.7 |
| Gibbs | | | 0.263 | 16119.9 |

| Results for Diagonal Covariance Matrix | | | | |
|---|---|---|---|---|
| N = 1000 | | | | |
| | w/ Antithetic Acceleration | | wo/ Antithetic Acceleration | |
| Method | Avg RMSE | Avg Time | Avg RMSE | Avg Time |
| MSM-GHK | 0.181 | 6470.9 | 0.167 | 6283.1 |
| MSM-Stern | 0.173 | 1911.4 | 0.186 | 2006.0 |
| SML-GHK | 0.158 | 1951.5 | 0.161 | 1889.9 |
| SML-Stern | 0.161 | 802.0 | 0.163 | 853.0 |
| Gibbs | | | 0.170 | 29746.9 |

Table 1 cont'd

Results for Non-Diagonal Covariance Matrix

N = 1000

| | w/ Antithetic Acceleration | | wo/ Antithetic Acceleration | |
|---|---|---|---|---|
| Method | Avg RMSE | Avg Time | Avg RMSE | Avg Time |
| MSM-GHK | 0.267 | 7192.1 | 0.201 | 6782.8 |
| MSM-Stern | 0.358 | 2422.5 | 0.420 | 2565.2 |
| SML-GHK | 0.175 | 2194.8 | 0.192 | 2010.0 |
| SML-Stern | 0.180 | 1114.3 | 0.195 | 1174.0 |

Notes:

There are 200 Monte Carlo draws per experiment.

There are 6 choices and 5 explanatory variables per choice.

For experiments with AA, $R = 5$, and for experiments without AA, $R = 10$.

All experiments are performed on an IBM RS6000 Model 390.

Gibbs sampling results are based on 10000 draws after skipping 2000 draws; i.e., $R_0 = 2000$ and $R_1 = 12000$.

# References

Albert, J. and S. Chib (1993). "Bayesian Analysis of Binary and Polychotomous Data." JASA. 88: 669-679.

Anderson, S.P., A. de Palma, and J.F. Thisse (1992). Discrete Choice Theory of Product Differentation. Cambridge: MIT

Avery, Robert, Lars Hansen, and V. Joseph Hotz (1983). "Multiperiod Probit Models and Orthogonality Condition Estimation." International Economic Review. 24(1): 21-35.

Berkovec, J. and S. Stern (1991). "Job Exit Behavior of Older Men." Econometrica. 59(1): 189-210.

Borsch-Supan, Axel and Vassilis A. Hajivassiliou (1993). "Smooth Unbiased Multivariate Probability Simulators for Maximum Likelihood Estimation of Limited Dependent Variable Models." Journal of Econometrics. 58: 347-368.

Bunch, David (1991). "Estimability in the Multinomial Probit Model." Transportation Research, Part B, Methodological. 25B: 1-12.

Butler, J.S. and Robert Moffitt (1982)."A Computationally Efficient Quadrature Procedure for the One-Factor Multinomial Probit Model." Econometrica. 50: 761-764.

Casella, G. and E. George (1992). "Explaining the Gibbs Sampler." American Statistician. 46: 167-174.

Chib, Siddhartha (1993). "Bayes Regression with Autoregressive Errors: A Gibbs Sampling Approach." Journal of Econometrics. 58(3): 347-368.

Chib, Siddhartha and Edward Greenberg (1994). "Understanding the Metropolis-Hastings Algorithm." Washington University, St. Louis, manuscript.

Devroye, L. (1986). Non-Uniform Random Variate Generation. New York: Springer.

Gelfand, A. and A. Smith (1990). "Sampling-Based Approaches to Calculating Marginal Densities." JASA. 85: 398-409.

Geman, S. and D. Geman (1984). "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images." IEEE Transactions on Pattern Analysis and Machine Intelligence 6. 721-741.

Geweke, John F. (1988). "Antithetic Acceleration of Monte Carlo Integration in Bayesian Inference." Journal of Econometrics. 38: 73-89.

Geweke, John (1989). "Bayesian Inference in Econometric Models Using Monte Carlo Integration." Econometrica. 57(6): 1317-1339.

Geweke, John (1991). "Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints." Computer Science and Statistics: Proceedings of the Twenty-Third Symposium on the Interface. 571-578.

Geweke, J., M. Keane, and D. Runkle (1994a). "Alternative Computational Approaches to Inference in the Multinomial Probit Model." Federal Reserve Bank of Minneapolis, Staff Report 170.

Geweke, J., M. Keane, and D. Runkle (1994b). "Statistical Inference in the Multinomial Multiperiod Probit Model." Federal Reserve Bank of Minneapolis, Staff Report 177.

Hajivassiliou, Vassilis (1990). "Smooth Simulation Estimation of Panel Data LDV Models." Unpublished paper.

Hajivassiliou, V., D. McFadden, and P. Ruud (1994). "Simulation of Multivariate Normal Rectangle Probabilities and their Derivatives: Theoretical and Computational Results." Cowles Foundation Discussion Paper No. 1021R.

Hammersly, J.M. and D.C. Handscomb (1964). Monte Carlo Methods. London: Methuen.

Hausman, Jerry, Bronwyn Hall, and Zvi Griliches (1984). "Econometric Models for Count Data with an Application to the Patents R & D Relationship." Econometrica. 52: 903-938.

Hausman, Jerry A. and David A. Wise (1978). "A Conditional Probit Model for Qualitative Choice: Discrete Decisions Recognizing Interdependence and Heterogenous Preferences." Econometrica. 46(2): 403-426.

Hendry, David F. (1984). "Monte Carlo Experimentation in Econometrics." Handbook of Econometrics, Volume II.

Keane, Michael P. (1994). "A Computationally Practical Simulation Estimator for Panel Data." Econometrica. 62(1): 95-116.

Lerman, Steven and Charles Manski (1981). "On the Use of Simulated Frequencies to Approximate Choice Probabilities." in Structural Analysis of Discrete Data with Econometric Applications. ed. by Charles Manski and Daniel McFadden. Cambridge: MIT Press.

McCulloch, Robert and Peter Rossi (1993). "An Exact Likelihood Analysis of the Multinomial Probit Model." Journal of Econometrics. ?

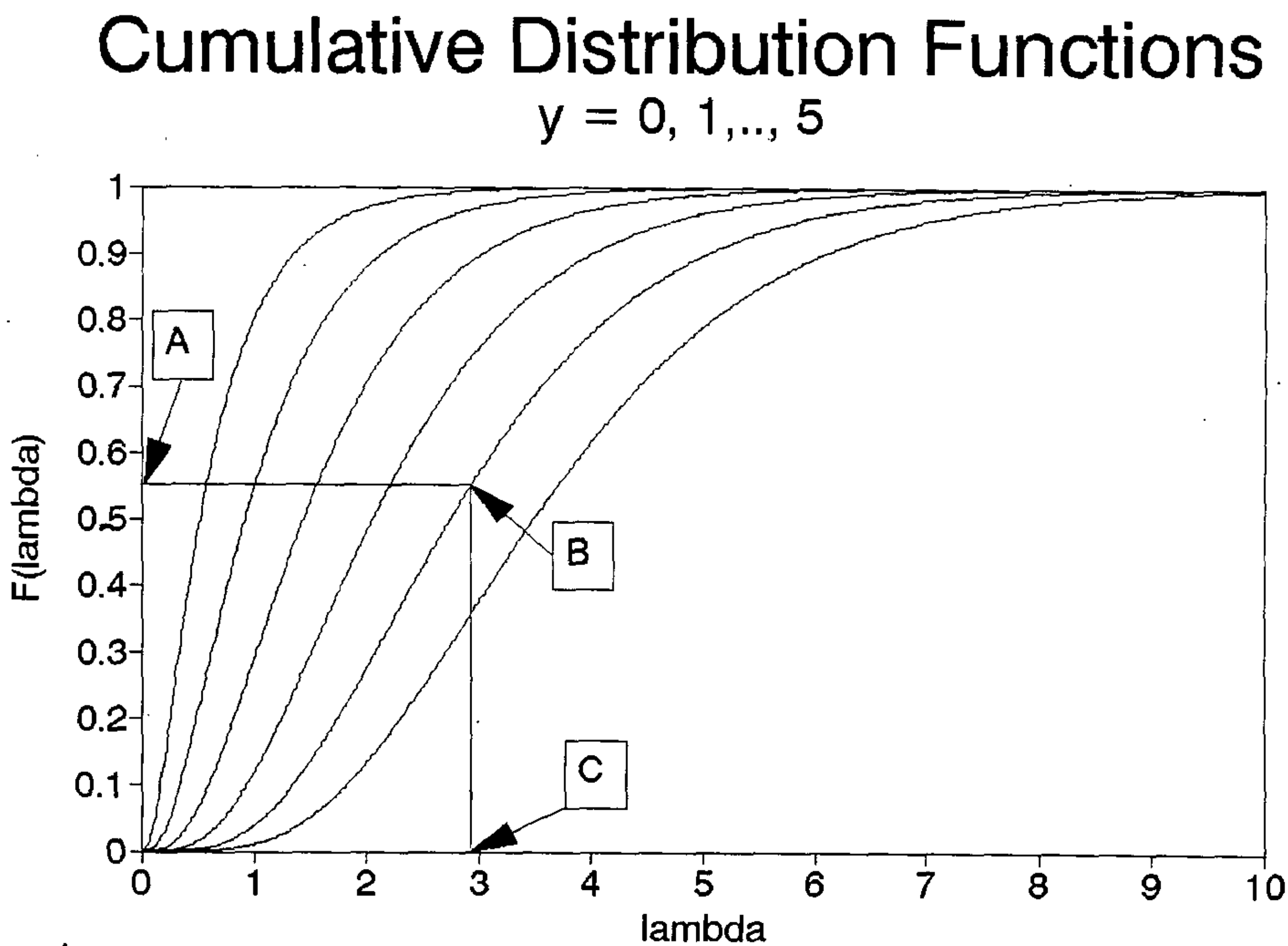
McFadden, Daniel (1989). "A Method of Simulated Moments for Estimation of Discrete Response Models without Numerical Integration." Econometrica. 57(5): 995-1026.

Pakes, Ariel and David Pollard (1989). "Simulation and the Asymptotics of Optimization Estimators." Econometrica. 57(5): 1027-1057.

Ripley, Brian (1987). Stochastic Simulation. New York: John Wiley and Sons.

Stern, Steven (1992). "A Method for Smoothing Simulated Moments of Discrete Probabilities in Multinomial Probit Models." Econometrica. 60(4): 943-952.

Tanner, T. and W. Wong (1987). "The Calculation of Posterior Distributions by Data Augmentation." JASA. 82: 528-549.


Cumulative Distribution Functions
y = 0, 1,.., 5
F(lambda)
lambda
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10
A
B
C