Some Practical Issues in Maximum Simulated Likelihood*

Vassilis A. Hajivassiliou
London School of Economics

October 1999

Abstract

In this Chapter[1], I explore ways of recapturing the efficiency property for estimators that rely on simulation. In particular, I show that this can be achieved by exploiting two-step maximum simulated likelihood (MSL) estimation methods that are familiar from classical applications. I also construct a diagnostic test for adequacy of number of simulations employed to guarantee negligible bias for the MSL and provide some evidence on the computational requirements of the Geweke-Hajivassiliou-Keane (GHK) simulator as a function of (a) the dimension of the problem and (b) the number of simulations employed in a vectorized context. I outline how one can derive a similar approach for checking the adequacy of the number of Gibbs resamplings in simulation estimation methods that employ this technique.

This chapter also shows how to suitably introduce simulation into classical hypothesis testing methods and provide test statistics (simulated Wald, Lagrange Multiplier, and Likelihood Ratio Tests) that are free of influential simulation noise.

Finally, I explain how simulation-variance-reduction techniques, notably antithetics, can improve substantially the practical performance of the GHK simulator and present extensive Monte-Carlo evidence confirming this.

Address correspondence to:

E-mail: vassilis@lse.ac.uk
Address: Department of Economics
London School of Economics
Houghton Street
London, WC2A 2AE
England

Keywords: Simulation Estimation, Maximum Simulated Likelihood, Limited Dependent Variable Models, Antithetic Acceleration.

JEL Classification: 210

*I am grateful to Paul Ruud for long discussions that led to some of the ideas in this paper. I would also like to thank Melvyn Weeks for useful comments and suggestions.

[1]To appear in "Simulation-Based Inference in Econometrics: Methods and Applications," Roberto Mariano, Melvyn Weeks and Til Schuermann (eds.), Cambridge University Press.

Some Practical Issues in Maximum Simulated Likelihood

Vassilis A. Hajivassiliou
London School of Economics

October 1999

1 Introduction

Estimation of econometric models is often hampered by computational complexity. The likelihood and moment functions that characterize an estimator cannot be computed with sufficient speed and accuracy to make the iterative computational search for the estimator feasible. Recent research has developed a marriage of simulation and estimation methods to overcome these computational obstacles. Generally, such methods sacrifice the efficiency of classical estimators for consistency, with simulation noise causing the efficiency loss. In order to make simulation an attractive technique, the number of replications of the simulations must be restricted to small values. Otherwise the repeated computation of the functions required for iterative solution of the estimators remain unmanageable.

After presenting an overview of simulation-based estimation of limited dependent variable models (LDV) in section 2, this chapter explores methods for recapturing the efficiency property for estimators that rely on simulation in section 3. The techniques exploit estimation methods that do not require iterative computation, which are already familiar from classical applications. The leading example is linearized maximum likelihood estimation (LMLE) discussed in subsection 3.1, which computes an asymptotically efficient estimator from an initial $\sqrt{N}$-consistent estimator. Because simulation methods for estimation do offer such initial estimators, one can apply the LMLE technique. In addition, because the LMLE does not require iteration, one can apply simulation with relatively high numbers of replications to reduce the simulation noise to potentially negligible levels. In section 3.2 I discuss the optimal determination of the number of simulations to employ in practice for this type of estimators.

In section 4 I first explain some practical computational advantages of maximum simulated likelihood (MSL) over the method of simulated moments (MSM) and then construct in 4.2 a diagnostic test for adequacy of number of simulations employed to guarantee negligible bias for the MSL. In 4.3 I provide some evidence on the computational speed of the GHK simulator as a function of (a) the dimension of the problem and (b) the number of simulations employed in a vectorized context. I outline how one can derive a similar approach for checking the adequacy of the number of Gibbs

resamplings in simulation estimation methods that employ this technique.

Given the computation of estimators, classical hypothesis test statistics typically do not involve iterative computation either. Therefore, I also show in section 5 how to suitably introduce simulation into classical hypothesis testing methods and provide test statistics (simulated Wald, Lagrange Multiplier, and Likelihood Ratio Tests in 5.1–5.3) that are free of influential simulation noise. This provides improvements in power comparable to the improvements in efficiency of estimators. Examples are given in subsection 5.4 for hypotheses of major interest in LDV models.

Finally, I explain in section 6 how simulation-variance-reduction techniques, most notably antithetics, can improve even further the practical performance of the GHK simulator. Section 7 concludes.

2 Estimation in LDV Models

2.1 The Computational Complexity of LDV Models

Consider the problem of maximum likelihood estimation given the N observations on the vector of random variables y drawn from a population with cumulative distribution function (c.d.f.) $F(\theta, Y) = \Pr\{y \leq Y\}$.[2] Let the corresponding density function with respect to Lebesgue measure be $f(\theta, y)$. The density f is a parametric function and the parameter vector θ is unknown, finite-dimensional, and $\theta \in \Theta$, where Θ is a compact subset of $\mathbf{R}^K$. Estimation of θ by maximum likelihood (ML) involves the maximization of the log-likelihood function $\ell_N(\theta) \equiv \sum_{n=1}^{N} \log f(\theta; y_n)$ over Θ. Often, finding the root of a system of normal equations $\nabla_\theta \ell_N(\theta) = 0$ is equivalent. In the limited dependent variable models that I consider here, F will be a mixture of discrete and continuous distributions, so that f may consist of nonzero probabilities for discrete values of y and continuous probability densities for intervals of y. These functions are generally difficult to compute because they involve multivariate integrals that do not have closed forms, accurate approximations, or rapid numerical solutions. As a result, estimation of θ by classical methods is effectively infeasible.

In general, and particularly in LDV models, one can represent the data generating process for y as an 'incomplete data' or 'partial observability' process in which the observed data vector y is an indirect observation on a latent vector y^*. In such case, y^* cannot be recovered from the *censored* random variable y. Let Y^* be a random variable from a population with c.d.f. $F(Y^*)$ and support $\mathbf{A}$. Let $\mathbf{B}$ be the support

[2]The discussion in this section follows closely the exposition in **?**. See that study for a deeper and more extensive theoretical analysis of the problem.

of the random variable $Y = \tau(Y^*)$ where $\tau : \mathbf{A} \to \mathbf{B}$ is not invertible. Then Y is a *censored* random variable.

In LDV models, τ is often called the 'observation rule;' and though it may not be monotonic, τ is generally piece-wise continuous. An important characteristic of censored sampling is that no observations are missing. Observations on y^* are merely abbreviated or summarized, hence the descriptive term 'censored.' Let $\mathbf{A} \subseteq \mathbf{R}^M$ and $\mathbf{B} \subseteq \mathbf{R}^J$.

The latent c.d.f. $F(\theta; Y^*)$ for y^* is related to the observed c.d.f. for y by the integral equation

$$F(\theta; Y) = \int_{\{y^* | \tau(y^*) \leq Y\}} dF(\theta; y^*). \tag{1}$$

The p.d.f. for y is the function that integrates to $F(\theta; Y)$. In this paper, integration refers to the Lebesgue-Stieltjes integral and the p.d.f. is a generalized derivative of the c.d.f. This means that the p.d.f. has discrete and continuous components. Everywhere in the support of Y where F is differentiable, the p.d.f. can be obtained by ordinary differentiation:

$$f(\theta; Y) = \frac{\partial^J F(\theta; Y)}{\partial Y_1 \dots \partial Y_J}. \tag{2}$$

In the LDV models I consider, F generally has a small number of discontinuities in some dimensions of Y so that F is not differentiable everywhere. At a point of discontinuity Y^d, I can obtain the generalized p.d.f. by partitioning Y into the elements in which F is differentiable, $\{Y_1, \dots, Y_{J'}\}$ say, and the remaining elements $\{Y_{J'+1}, \dots, Y_J\}$ in which the discontinuity occurs. The p.d.f. then has the form

$$\begin{aligned} f(\theta; Y) &= \tfrac{\partial^{J'}}{\partial Y_1 \dots \partial Y_{J'}} \cdot [F(\theta; Y) - F(\theta; Y - 0)] \\ &= f(\theta; Y_1, \dots, Y_{J'}) \cdot \Pr\{Y_j = Y_j^d; j > J' | \theta; Y_1, \dots, Y_{J'}\}, \end{aligned} \tag{3}$$

where the discrete jump $F(\theta; Y) - F(\theta; Y - 0)$ reflects the nontrivial probability of the event $\{Y_j = Y_j^d; j > J'\}$.[3]

It is these probabilities, the discrete components of the p.d.f., that pose computational obstacles to classical estimation. One must carry out multivariate integration and differentiation in (1)–(3) to obtain the likelihood for the observed data — see the following example for a clear illustration of this problem. Because accurate numerical

[3]The height of the discontinuity is denoted by

$$F(\theta; Y) - F(\theta; Y - 0) \equiv \lim_{\epsilon \downarrow 0} F(\theta; Y) - F(\theta; Y - \epsilon).$$

approximations are unavailable, this integration is often handled by such general purpose numerical methods as quadrature. But the speed and accuracy of quadrature is inadequate to make the computation of the MLE practical except in special cases.

2.2 Score Functions

For models with censoring, the score for θ can be written in two ways which I will use to motivate two approaches to approximation of the score by simulation:

$$s(\theta; y) \equiv \nabla_\theta \ln f(\theta; y) = \frac{\nabla_\theta f(\theta; y)}{f(\theta; y)} \tag{4}$$

$$= E\left[\nabla_\theta \ln f(\theta; y^*) \middle| y\right] \tag{5}$$

where ∇_θ is an operator that represents partial differentiation with respect to the elements of θ. The ratio expression in (4) is simply the derivative of the log-likelihood and simulation can be applied to the numerator and denominator separately. The second expression (4), the conditional expectation of the score of the latent log-likelihood, can be simulated as a single expectation if $\nabla_\theta \ln f(\theta; y^*)$ is tractable. **?**, **?**, and **?** have noted alternative ways of writing score functions for the purpose of estimation by simulation.

2.3 Simulation-Based Estimation of LDV Models

I begin with the application of simulation to approximating the log-likelihood function. Next, I consider the simulation of moment functions. Because of the simulation biases that naturally arise in the log-likelihood approach, the unbiased simulation of moment functions and the method of moments is an alternative approach. Finally, I discuss simulation of the score function. Solving the normal equations of ML estimation is a special case of the method of moments and simulating the score function offers the potential for efficient estimation.

Throughout this section, I will assume that we are working with models for which the maximum likelihood estimator is well-behaved. In particular, I suppose that the usual regularity conditions are met, ensuring that the ML estimator is the most efficient consistent, uniformly asymptotically normal (CUAN) estimator.

2.3.1 Simulation of the Log-Likelihood Function

One of the earliest applications of simulation to estimation was the general computation of multivariate integrals in such likelihoods as that of the multinomial probit by Monte Carlo integration. Crude Monte Carlo simulation can approximate the probabilities of

the multinomial probit to any desired degree of accuracy, so that the corresponding *maximum simulated likelihood* (MSL) estimator can approximate the ML estimator.

Definition 1 (Maximum Simulated Likelihood) *Let the log-likelihood function for the unknown parameter vector θ given the sample of observations $(y_n, n = 1, \ldots, N)$ be*

$$\ell_N(\theta) \equiv \sum_{n=1}^{N} [\log f(\theta; y_n)]$$

and let $\tilde{f}(\theta; y, \omega)$ be an unbiased simulator so that $f(\theta; y) = \mathrm{E}_\omega[\tilde{f}(\theta; y, \omega)|y]$ where ω is a simulated vector of R random variates. The maximum simulated likelihood estimator is

$$\hat{\theta}_{MSL} \equiv \arg\max_\theta \tilde{\ell}_N(\theta)$$

where

$$\tilde{\ell}_N(\theta) \equiv \sum_{n=1}^{N} \log \tilde{f}(\theta; y_n, \omega_n)$$

for some given simulation sequence $\{\omega_n\}$.

It is important to note that MSL estimator is conditional on the sequence of simulators $\{\omega_n\}$. For both computational stability and asymptotic distribution theory, it is important that the simulations do not change with the parameter values. See **?** and **?** for an explanation of this point.

Note that unbiased simulation of the likelihood function is neither necessary nor sufficient for consistent MSL estimation. Because the estimator is a nonlinear function (through optimization) of the simulator, the MSL estimator will generally be a biased simulation of the MLE even when the criterion function of estimation is simulated without bias because

$$\mathrm{E}\left[\tilde{\ell}(\theta)\right] = \ell(\theta) \not\Rightarrow \mathrm{E}\left[\arg\max_\theta \tilde{\ell}(\theta)\right] = \arg\max_\theta \ell(\theta).$$

Note also that while unbiased simulation of the likelihood function is often straightforward, unbiased simulation of the *log*-likelihood is generally infeasible. The logarithmic transformation of the intractable function introduces a nonlinearity that cannot be overcome simply. However, to obtain an estimator with the same *probability limit* as the MLE, a sufficient characteristic of a simulator for the log-likelihood is that its sample average converge to the same limit as the sample average log-likelihood. Only by reducing the error of a simulator for the log-likelihood function to zero at a sufficiently rapid rate with sample size can one expect to obtain a consistent estimator.

For LDV models with censoring, the generic likelihood simulator $\tilde{f}(\theta; y_n, \omega_n)$ is the average of R replications of one of the simulation methods described elsewhere:

$$\tilde{f}(\theta; y_n, \omega_n) \equiv \frac{1}{R} \sum_{r=1}^{R} \tilde{f}(\theta; y_n, \omega_{nr}).$$

The simulation error will generally be $O_P(1/R)$. Thus, a common approach to approximating the log-likelihood function with sufficient accuracy is increasing the number of replications per observation R with the sample size N. This statistical approach is in contrast to a strictly numerical approach of setting R high enough to achieve a specified numerical accuracy independent of sample size.

2.3.2 Simulation of Moment Functions

The simulation of the log-likelihood is an appealing approach to applying simulation to estimation, but this approach must overcome the inherent simulation bias that forces one to increase R with the sample size. Instead of simulating the log-likelihood function, one can simulate moment functions. When they are linear in the simulations, moment functions can be simulated easily without bias. The direct consequence is that the simulation bias in the limiting distribution of an estimator is also zero, making the need to increase the number of simulations per observation with sample size unnecessary. This was a key insight of **?** and **?**.

Method of moments (MOM) estimators have a simple structure. Such estimators are generally constructed from 'residuals' that are differences between observed random variables y and their conditional expectations. These expectations are known functions of the conditioning variables x and the unknown parameter vector θ to be estimated, let $\mathrm{E}(y \mid x, \theta) \equiv \mu(\theta; x)$. Moment equations are built up by multiplying the residuals by various weights or instrumental variable functions, z_n, and specifying the estimator as the parameter values which equate the sample average of these products with zero: The MOM estimator $\hat{\theta}_{MOM}$ is defined by

$$\frac{1}{N} \sum_{n=1}^{N} z_n(X, \hat{\theta}_{MOM}) \left[y_n - \mu(\hat{\theta}_{MOM}; x_n, \omega_n) \right] = 0. \tag{6}$$

Simulation has an affinity with the MOM. Substituting an unbiased, finite-variance simulator for the conditional expectation $\mu(\theta; x_n)$ does not alter the essential convergence properties of these sample moment equations. I therefore consider the class of estimators generated by the *method of simulated moments* (MSM).

Definition 2 (Method of Simulated Moments) *Let* $\tilde{\mu}(\theta; x, \omega) = 1/R \sum_{r=1}^{R} \tilde{\mu}(\theta; x, \omega_r)$ *be an unbiased simulator so that* $\mu(\theta; x) = E[\tilde{\mu}(\theta; x, \omega) \mid x]$ *where* ω *is a simulated random variable. The method of simulated moments estimator is*

$$\hat{\theta}_{MSM} \equiv \arg\min \|\tilde{s}_N(\theta)\|$$

where

$$\tilde{s}_N(\theta) \equiv 1/N \sum_{n=1}^{N} w_n(\theta) \left[y_n - \tilde{\mu}(\theta; x_n, \omega_n)\right] \tag{7}$$

for some sequence $\{\omega_n\}$.

2.3.3 Simulation of the Score Function

Interest in the efficiency of estimators naturally leads to attempts to construct an efficient MSM estimator. The obvious way to do this is to simulate the score function as a set of simulated moment equations. Within the LDV framework however, unbiased simulation of the score with a finite number of operations is not possible with simple censored simulators; the efficient weights are nonlinear functions of the objects that require simulation. Nevertheless, it may be possible with the aid of simulation to construct good approximations that offer improvements in efficiency over simpler MSM estimators.

There is an alternative approach based on truncated simulation. It was shown in **?** that every score function can be expressed as the expectation of the score of a latent data generating process taken conditional on the observed data. In the particular case of normal LDV models, this conditional expectation is taken over a truncated multivariate normal distribution and the latent score is the score of an multivariate normal distribution. Simulations from the truncated normal distribution can replace the expectation operator to obtain unbiased simulators of the score function.

I define the *method of simulated scores* as follows.[4]

Definition 3 (Method of Simulated Scores) *Let the log-likelihood function for the unknown parameter vector* θ *given the sample of observations* $(y_n, n = 1, \ldots, N)$ *be* $\ell_N(\theta) \equiv \sum_{n=1}^{N} \log f(\theta; y_n)$. *Let* $\tilde{\mu}(\theta; y_n, \omega_n) = 1/R \sum_{r=1}^{R} \tilde{\mu}(\theta; y_n, \omega_{nr})$ *be an asymptotically (in* R*) unbiased simulator of the score function* $s(\theta; y) \equiv \nabla \ln f(\theta; y)$ *where* ω *is a simulated random variable. The* method of simulated scores *estimator is* $\hat{\theta}_{MSS} \equiv \arg\min_{\theta \in \Theta} \|\tilde{s}_N(\theta)\|$ *where* $\tilde{s}_N(\theta) \equiv 1/N \sum_{n=1}^{N} \tilde{\mu}(\theta; y_n, \omega_n)$ *for some sequence* $\{\omega_n\}$.

[4]The term was coined by **?**.

Truncated Simulation of the Score The truncated simulation methods provide unbiased simulators of the LDV score (4). Such simulation would be ideal, because R can be held fixed, thus leading to fast estimation procedures. The problem is that these truncated simulation methods pose new problems for the MSS estimators that use them.

The Accept/Reject (A/R) method provides simulations that are discontinuous in the parameters. A/R simulation delivers the first element in a simulated sequence that falls into a region which depends on the parameters under estimation. As a result, changes in the parameter values cause discrete changes in which element in the sequence is accepted. See **?** and **?** for treatments of the special asymptotic distribution theory for such simulation estimators. Briefly described, this distribution theory requires a degree of smoothness in the estimator with respect to the parameters that permits such discontinuities but allows familiar linear approximations in the limit. See **?** for an illustrative application.

The Gibbs resampling simulation method can also be used here. This method is continuous in the parameters provided that one uses a continuous univariate truncated normal simulation scheme. But this simulation method also has a drawback: Strictly applied, each simulation requires an infinite number of resampling rounds. In practice, Gibbs resampling is truncated and applied as an approximation. The limited Monte Carlo evidence that I have seen suggests that such approximation is reliable.

3 Statistically Efficient Simulation-Based Estimation

In this section, I discuss various approaches of obtaining statistically fully efficient simulation estimators that are computationally feasible. Often, simulation estimation methods sacrifice the efficiency of classical estimators for consistency, with simulation noise causing the efficiency loss. I show how one can recapturing the efficiency property for estimators that rely on simulation by exploiting two-step estimation methods that are familiar in classical applications. A critical issue is the selection of a value of the number of replications to attain a negligible level of asymptotic efficiency loss due to simulation.

3.1 Two-Step Estimators

In LMLE, the score and the information need only need to be evaluated once (or very few times). This is in contrast to obtaining the simulation estimators themselves

through some iterative scheme, which requires the repeated evaluation of simulated functions. Whenever functions to be simulated only need to be evaluated once, then a large number of replications, R, can be used in the simulation. This implies, of course, that the additional noise contributed by simulation can be negligible (assuming that, as is typically the case, the simulators are consistent, as R grows without bound, for the true expressions). As a result, all the standard asymptotic properties of estimators and test statistics *not based on simulation* still hold.

Hence, an efficient estimation procedure can be outlined as follows: In step 1, I obtain a consistent but inefficient estimator θ_0, using MSM for example. In step 2, a LMLE step is carried out, using the optimal score expressions corresponding to full-information maximum likelihood, using a very high number of operations in approximating these score expressions. "Operations" in this context means number of simulations if the MSL version of the scores is used, and Gibbs resamplings if the MSS version is used. This is computationally appealing, since the second-step (intractable) optimal scores need to be approximated only once.

Let us explain this approach in greater detail. When the MLE $\hat{\theta}_N$ is the root of the score equations

$$\mathrm{E}_N\left[s(\hat{\theta}_N, y, x)\right] = 0,$$

$\hat{\theta}_N$ is consistent, asymptotically normal, and statistically efficient, under standard regularity conditions. Given an initial $\sqrt{N}$-consistent estimator $\bar{\theta}_N$, the LMLE $\check{\theta}_N$

$$\check{\theta}_N \equiv \bar{\theta}_N + \bar{H}_N^{-1}\,\mathrm{E}_N\left[s(\bar{\theta}_N, y, x)\right]$$

is an asymptotically equivalent estimator, where $\bar{H}_N$ is a consistent estimator of the information matrix $\mathrm{E}\left\{\mathrm{Var}\left[s(\theta, y, x) \mid x\right]\right\}$. Two estimators are popular. When

$$\bar{H}_N = \mathrm{Var}_N\left[s(\bar{\theta}_N, y, x)\right],$$

the LMLE is often called the *Gauss-Newton two-step estimator*. When

$$\bar{H}_N = \mathrm{E}_N\left[\nabla_\theta s(\bar{\theta}_N, y, x)\right],$$

the LMLE is called the *Newton-Raphson two-step estimator*. Subject to standard regularity conditions (see, for example, **?**) both two-step estimators are asymptotically equivalent to $\hat{\theta}_N$ in the sense that plim $\sqrt{N}(\check{\theta}_N - \hat{\theta}_N) = 0$. Hence, these two estimators share the consistency and efficiency properties of $\hat{\theta}_N$, even though they are considerably more tractable computationally given that they are based on the inefficient but simple-to-calculate preliminary estimator $\bar{\theta}_N$ and evaluate only once the (intractable) scores of $\hat{\theta}_N$.

This result is readily transportable to estimation by simulation. Let the tractable but inefficient estimator be MSM, and the intractable but efficient one be MSS. Since the score expressions need only be evaluated once at the MSM estimate, one can afford to base this calculation on a huge number of replications, R. Since this implies that the simulated score expressions only add negligible simulation noise since R is very large, no modification of the standard asymptotic theory for estimation without simulation is necessary.

3.2 Choosing the Number of Simulations R

I now discuss a method for choosing the level of R. This is essentially a sampling design question, where one is asking how many replications are needed to get a certain level of statistical precision. First, estimate the simulation noise contribution to the variance. Second, determine R to reduce this to $\alpha\%$ of the sampling variance of the classical estimator. Note that a similar analysis can be developed for choosing an acceptable value of Gibbs resampling rounds for calculating the MSS/Gibbs estimator.

Let the covariance matrix of an MSM estimator be written

$$\text{Var}(\hat{\theta}) = \Omega_C + \frac{1}{R}\Omega_S$$

where Ω_C represents the covariance matrix for the classical estimator that the MSM estimator approximates, and Ω_S represents the covariance matrix contributed by simulation noise. In the special case where the simulation process and the data generating process are the same (except for parameter values), $\Omega_C = \Omega_S$ and one can choose R easily to reduce the contribution of simulation to a fraction considered negligible, say one percent.

In general, $\Omega_C \neq \Omega_S$ and the problem of choosing R is less transparent. I suggest a method based on bounding the contribution of simulation in least favorable circumstances. It is convenient to consider variances of all linear combinations of the parameters to be estimated and ensure that the variance of the most variable linear combination contains no more than a tolerable fraction of simulation variance, say ϵ. Formally, one can choose R so that

$$\frac{1}{R}\max_a \frac{a'\Omega_S a}{\text{Var}(a'\hat{\theta})} \leq \epsilon,\ \epsilon > 0.$$

To make this analytically convenient, note that

$$a'\left(\Omega_C + \frac{1}{R}\Omega_S\right)a > a'\Omega_C a$$

so that

$$\frac{a'\Omega_S a}{\mathrm{Var}(a'\hat{\theta})} \leq \frac{a'\Omega_S a}{a'\Omega_C a}$$

and a conservative solution can be obtained from finding R such that

$$\frac{1}{R}\max_a \frac{a'\Omega_S a}{a'\Omega_C a} \leq \epsilon,\ \epsilon > 0.$$

The maximization problem has a (presumably) well-known solution. The first order conditions state that

$$\begin{aligned} a'\Omega_C a \cdot \Omega_S a - a'\Omega_S a \cdot \Omega_C a = &\quad 0 \\ \Leftrightarrow \left(\Omega_C^{-1}\Omega_S - \frac{a'\Omega_S a}{a'\Omega_C a} I\right) a = &\quad 0 \end{aligned}$$

so that a must be proportional to an eigenvector of $\Omega_C^{-1}\Omega_S$. Restricting attention to eigenvectors,

$$a'\Omega_C^{-1}\Omega_S a = \frac{a'\Omega_S a}{a'\Omega_C a}$$

so that the possible values for the objective function are the eigenvalues of $\Omega_C^{-1}\Omega_S$ and the largest value of the objective function is the largest such eigenvalue, call it λ^*. I conclude that one should set $R = \lambda^*/\epsilon$.

4 Improving the Performance of MSL

Let us begin with a preliminary computatational issue that, though very simple and potentially extremely important, it does not appear to be widely recognized in practice. I am referring to the fact that the tails of the multivariate normal density function die out very rapidly indeed. For example, the standard normal c.d.f. $\Phi(q)$ is less than about $1.e-14$ for q approximately less than -12. What this means is that in practice the calculation of multivariate normal rectangle probabilities will cause underflows (implying severe problems in evaluating their logarithms) unless the arguments of these functions remain not too far from the center of the distribution. In the canonical model in our context where the probabilities of interest are of the form

$$prob(a < Z < b), \quad Z \sim N(X\beta, \Omega)$$

it will typically be very useful to first standardize the X's (except for the intercept) to 0 sample mean and 1 sample variance, calling the standardized regressors X^*. Then

the analysis is carried out with X^* as the regressors, thus alleviating some of the computational problems with the trial arguments of the $prob(\cdot)$ expressions getting too far out in the tails. Given the simple linearity of the transformation of the standardization operation, re-mapping from the estimates corresponding to the X^* regressors to those corresponding to the orginal X is obvious.

With this practical suggestion in mind, let us proceed to some computational characteristics of the MSL and MSM estimators. The fact, stated in the previous section, that consistency and asymptotic normality of the MSL estimator requires that R grow without bound faster than $\sqrt{N}$ begs several questions. First, one needs to ask why use MSL that is biased for finite R as opposed to MSM that is CUAN for any (finite) R. The answer is given in subsection 3.1. Subsection 3.2 develops a diagnostic test for simulation bias, thus enabling one to adopt the following two-step strategy: (1) compute an MSL estimator which is biased, but computationally attractive and (2) test for magnitude of bias and re-estimate if necessary.

4.1 Computational Attractiveness of MSL over MSM

Researchers have noted that the MSM can be numerically unstable (**?**, **?**), whereas MSL is relatively straightforward (**?**, **?**). The MSM estimation criterion function is constructed from a set of moment equations. In this sense, the MSM estimation criterion function is an artificial construct: the distance function. The MSL criterion function has the properties of a log-likelihood function, giving MSL an inherently more tractable criterion function. The computational differences between MSM and MSL are analogous to the differences in identification between the classical MOM and MLE. For the former, identification is a more difficult exercise.

I can give some concreteness to the nature of some of the problems encountered in estimation with simulation by examining the simple binary probit as an example. In this model, I can write the log-likelihood function as

$$\ell(\beta) = y \log \Phi(x'\beta) + (1-y) \log \left[1 - \Phi(x'\beta)\right].$$

The MLE avoids regions of the parameter space in which the fitted probability values are near zero, because the contribution of such terms to the log-likelihood function approaches negative infinity. The score function for binary probit is often written as

$$\begin{aligned} s(\theta; y, x) &= x \frac{\phi(x'\beta)}{\Phi(x'\beta)\left[1 - \Phi(x'\beta)\right]} \left[y - \Phi(x'\beta)\right] \\ &= x \begin{cases} \frac{\phi(x'\beta)}{\Phi(x'\beta)} & \text{if } y = 1 \\ \frac{-\phi(x'\beta)}{\left[1-\Phi(x'\beta)\right]} & \text{if } y = 0 \end{cases} \end{aligned}$$

and the information matrix is

$$J(\theta; x) = \frac{\phi(x'\beta)^2}{\Phi(x'\beta)\left[1 - \Phi(x'\beta)\right]}\, x\, x'.$$

These expressions have familiar MOM interpretations. The denominator $\Phi(x'\beta)\left[1 - \Phi(x'\beta)\right]$ reflects the heteroskedasticity in the residual $y - \Phi(x'\beta)$ and the numerator $\phi(x'\beta)$ captures the nonlinearity of the regression function $\Phi(x'\beta)$. In the balance, the MLE is largely driven to avoid poor in-sample predictions of the sample outcomes in y.

The noise in the instrumental variables employed by MSM estimators can easily obscure this efficient weighting. One typically constructs the MSM estimator by replacing all of the analytical p.d.f. and c.d.f. terms with unbiased simulations. In this way, one attempts to approximate the efficient score. For example, a particular simulated moment function could be

$$g(\theta; y, x, \omega) = x \frac{f(x'\beta, \omega_1)}{F(x'\beta, \omega_1)\left[1 - F(x'\beta, \omega_1)\right]}\left[y - F(x'\beta, \omega_2)\right] \tag{8}$$

where the ω's are simulations and f and F are chosen so that

$$\begin{aligned} \mathrm{E}_\omega\left[F(z, \omega)\right] &= \Phi(z) \\ \mathrm{E}_\omega\left[f(z, \omega)\right] &= \phi(z) \end{aligned}$$

Care must be taken to ensure that ω_1 and ω_2 are independently distributed, otherwise the MSM estimator is inconsistent because the residual $y - F(x'\beta, \omega_2)$ will be correlated with the simulated weighting term. An MSM estimator then seeks to minimize a distance function:

$$\hat{\beta}_{MSM} = \arg\min_\beta \left\| \mathrm{E}_N\left[g(\theta; y, x, \omega)\right] \right\|$$

The operator E_N denotes the empirical expectation over the sample observations.

But the MSM restriction on the ω's has an important side-effect. It vitiates the intimate relationship between the regression function and the weighting function so that poor in-sample predictions are no longer as costly as in the log-likelihood function. In practice, one frequently finds an MSM algorithm searching in regions of the parameter space where the weights are diminished rather than the residuals. The result of this failure is that in small samples, the MSM estimator can be very poorly behaved, wandering into unlikely regions of the parameter space.

The noise in the instrumental variables employed by MSL estimators does not have this effect. In the MSL, one solves the quasi-maximum-likelihood problem

$$\hat{\beta}_{MSL} = \arg\max_\beta \mathrm{E}_N\, y \log F(x'\beta, \omega_1) + (1 - y) \log\left[1 - F(x'\beta, \omega_1)\right]$$

so that (8) is altered by equating $\omega_1 = \omega_2$ and $f(z,\omega) = dF(z,\omega)/dz$ yielding the alternative moment function

$$\begin{aligned} g(\theta; y, x, \omega) &= x \frac{f(x'\beta, \omega_1)}{F(x'\beta, \omega_1)\left[1 - F(x'\beta, \omega_1)\right]} \left[y - F(x'\beta, \omega_1)\right] \\ &= x \left\{ \begin{array}{ll} \frac{f(x'\beta,\omega_1)}{F(x'\beta,\omega_1)} & \text{if } y = 1 \\ \frac{-f(x'\beta,\omega_1)}{[1-F(x'\beta,\omega_1)]} & \text{if } y = 0 \end{array} \right\}. \end{aligned}$$

Like the MLE, the MSL estimator will avoid regions of the parameter space that yield poor in-sample predictions. Even for crude simulators, this yields optimization problems that do not lead numerical algorithms to the edges of the parameter space.

It is well understood that this stability comes at the cost of inconsistency in the MSL estimator. But experience shows that the magnitude of the inconsistency is frequently small. With a modest R, an estimator that is *practically* consistent can be constructed by MSL. Given the computational attractiveness of the MSL estimator, I suggest a diagnostic test for whether the magnitude of the inconsistency is important in inference.

4.2 A Diagnostic Test for Simulation Bias

Given the MSL estimator $\hat{\theta}$, where

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \ell_N(\theta; y, x, \omega) \Longleftrightarrow g(\hat{\theta}; y, x, \omega) = 0,$$

we know that bias arises from the condition

$$\mathrm{E}\left[g(\theta_0; y, x, \omega)\right] \neq 0.$$

As R grows, this expectation approaches zero and the inconsistency disappears. Under the hypothesis that the MSL estimator is consistent,

$$\begin{aligned} \mathrm{E}\left[g\left(\theta; y, x, \omega\right) \mid \omega,\ \theta = \theta_0\right] &= 0 \\ \mathrm{Var}\left[g\left(\theta; y, x, \omega\right) \mid \omega,\ \theta = \theta_0\right] &= \mathrm{E}\left\{g\left[\theta_0; y, x, \omega\right] g\left[\theta_0; y, x, \omega\right]' \mid \omega\right\}. \end{aligned}$$

The basis for a test of consistency is to check the necessary condition that

$$\mathrm{E}\left[g\left(\theta; y, x, \omega\right) \mid \omega,\ \theta = \hat{\theta}\right] = 0.$$

For any R and $\hat{\theta}$, we can easily compute the expectation and variance of the MSL score function. Let $y(\theta)$ denote a simulation of the data generating process for y at

θ conditional on x, where this additional simulation is independent of ω, and let E_S denote the empirical expectation of functions of $y(\theta)$ over S replications of $y(\theta)$. Then

$$m \equiv \mathrm{E}_S \left\{ g \left[\hat{\theta}; y(\hat{\theta}), x, \omega \right] \right\}$$

and

$$V \equiv \mathrm{Var}_S \left\{ g \left[\hat{\theta}; y(\hat{\theta}), x, \omega \right] \right\}$$

are unbiased simulators of $\mathrm{E} \left\{ g \left[\theta; y, x, \omega \right] \mid x,\ \omega,\ \theta = \hat{\theta} \right\}$ and $\mathrm{Var} \left\{ g \left[\theta; y, x, \omega \right] \mid x,\ \omega,\ \theta = \hat{\theta} \right\}$. Under the hypothesis that the MSL estimator is consistent,

$$\left[\mathrm{E}_N \left(V \right) \right]^{-\frac{1}{2}} \sqrt{SN}\, \mathrm{E}_N \left(m \right) \xrightarrow{d} \mathcal{N} \left(0, I \right)$$

as $N \to \infty$. Thus a simple specification test, similar in spirit to the **?** specification test, can be constructed from the Wald statistic:

$$w = SN \cdot \left[\mathrm{E}_N (m) \right]' \left[\mathrm{E}_N (V) \right]^{-1} \left[\mathrm{E}_N (m) \right]$$

evaluating the moments at the particular θ of interest, $\hat{\theta}$. Under the null hypothesis of MSL consistency, w has an asymptotic distribution that is central chi-square with K degrees of freedom, where K is the dimension of θ.

We interpret an insignificant statistic as evidence that the variation in the estimator is large relative to the simulation bias. Therefore, if the precision of the MSL estimator is satisfactory, there is negligible inconsistency in the MSL estimator itself. The statistic w measures the difference between the MSL estimator and a one-step estimator towards bias-correction as advanced by **?**. In cases where the test statistic w is considered to be non-negligible, the researcher can inspect a local estimate of the bias in the MSL estimator. If $\mathrm{E}(m) \neq 0$, then the first-order approximation of the MSM estimator that solves

$$\mathrm{E}_N \left(g \left[\theta; y, x, \omega \right] - \mathrm{E}_S \left\{ g \left[\theta; y(\theta), x, \omega \right] \right\} \right) = 0$$

is

$$\hat{\delta} = J\, m$$

where

$$J \equiv \left[\nabla_\theta\, \mathrm{E}_N \left(g \left[\hat{\theta}; y, x, \omega \right] - \mathrm{E}_S \left\{ g \left[\hat{\theta}; y(\hat{\theta}), x, \omega \right] \right\} \right) \right]^{-1}.$$

The precision in the estimated bias $\hat{\delta}$ can be evaluated using the estimated covariance matrix $J\, \mathrm{E}_N V\ J'$. Although testing whether $\hat{\delta}$ is significantly different from zero is equivalent to w, looking at the outcome in the parameter space rather than the moment space may be more meaningful.

Keep in mind that this analysis is conditional on ω, treating the simulated score as an exact function. Another way to look at this approach is to think of $\tilde{\theta}$ as a potentially misspecified MLE, as in **?**. This diagnostic has the same spirit as White's information test, except one is conducting the test based on first, rather than second, moments. This is possible because the entire data generating process is specified in the MSL setting, permitting us to draw from that DGP given θ.

If the MSL estimator fails to pass this test, the statistic can also be used to compute the level of R that will yield an acceptable estimator. The researcher can experiment with increasing R until the w statistic is acceptably small. Having found such a level, he can return to reapply the MSL at this higher value.

4.3 Investigating the Computational Speed of the GHK Simulator

The leading simulator for multivariate normal rectangle probabilities of the form encountered in ML estimation of LDV models is the Geweke-Hajivassiliou-Keane approach. See **?** for extensive Monte-Carlo evidence that this simulator is to be preferred over all other known simulators for this problem. To outline this method, define $q(u,a,b) \equiv \Phi^{-1}(\Phi(a)\cdot(1-u)+\Phi(b)\cdot u)$, where $0<u<1$ and $-\infty \leq a < b \leq \infty$. Then q is a mapping that takes a uniform $(0,1)$ random variate into a truncated standard normal random variate on the interval $[a,b]$.

Proposition 1 *Consider the multivariate normal $M\times 1$ random vector $Y \sim N(X\beta,\Omega)$ with Ω positive definite, the linear transformation $Z = FY \sim N(FX\beta,\Sigma)$, with F non-singular and $\Sigma = F\Omega F'$, and the event $\mathbf{B} \equiv \{a^* \leq Z = FY \leq b^*\}$, with $-\infty \leq a^* < b^* \leq +\infty$. Define $P \equiv \int_B n(z;FX\beta,\Sigma)dz$, $a \equiv a^* - FX\beta$, $b \equiv b^* - FX\beta$, and let L denote the lower-triangular Cholesky factor of Σ. Let $(u_1,\cdots,u_M)$ be a vector of independent uniform $(0,1)$ random variates. Define recursively for $j = 1,\cdots,M$:*

$$e_j = q\left(u_j,(a_j - L_{j1}e_1 - \cdots - L_{j,j-1}e_{j-1})/L_{jj},(b_j - L_{j1}e_1 - \cdots - L_{j,j-1}e_{j-1})/L_{jj}\right), \tag{9}$$

$$Q_j \equiv \Phi\left((b_j - L_{j1}e_1 - \cdots - L_{j,j-1}e_{j-1})/L_{jj}\right) - \Phi\left((a_i - L_{j1}e_1 - \cdots - L_{j,j-1}e_{j-1})/L_{jj}\right). \tag{10}$$

Define $e \equiv (e_1,\cdots,e_M)'$, $\tilde{Y} \equiv X\beta + F^{-1}Le$, and $Q(e) \equiv Q_1 \cdot ... \cdot Q_M$. Then $\tilde{Y}$ is a random vector on $\mathbf{B}$, and the ratio of the densities of $\tilde{Y}$ and Y at $y = X\beta + F^{-1}Le$, where e is any vector satisfying $a \leq Le \leq b$, is $P/Q(e)$.

Proof: **?**, **?**.

These studies also show that combining Proposition 1 about the GHK simulator together with importance-sampling arguments, one can show that GHK is a smooth,

unbiased, and consistent simulator for the likelihood contributions P_i and their derivatives $P_{\theta i}$, and a smooth, asymptotically unbiased, and consistent simulator for the logarithmic derivatives of the $P(\cdot)$ expressions.

It is instructive to give here a complete implementation of the GHK simulator in the GAUSS computer matrix language.[5],[6],[7]

```
proc 1 = ghk(m,mu,w,wi,c,a,b,r,u);
local j,ii,ta,tb,tt,wgt,v,p;
j = 1;
ii = 1;
ta = cdfn((a[1,1]-mu[1,1])/(c[1,1]+1.e-100))*ones(1,r);
tb = cdfn((b[1,1]-mu[1,1])/(c[1,1]+1.e-100))*ones(1,r);
tt = cdfinvn(u[1,.].*ta+(1-u[1,.]).*tb);
wgt = tb-ta;
do while j < m;
  j = j+1;
  ta = cdfn(((a[j,1]-mu[j,1])*ones(1,r)-c[j,ii]*tt)/(c[j,j]+1.e-100));
  tb = cdfn(((b[j,1]-mu[j,1])*ones(1,r)-c[j,ii]*tt)/(c[j,j]+1.e-100));
  tt = tt|cdfinvn(u[j,.].*ta+(1-u[j,.]).*tb);
  ii = ii|j;
  wgt = wgt.*(tb-ta);
endo;
v = c*tt;
tt = (ones(m,1)*wgt).*v;
p = sumc(wgt')/r;
retp(p);
endp;
```

As can be seen from the timing experiments reported in Table 1 and Figures 1 and 2, the computational time of the GHK simulator is almost linear in the dimension of the multivariate vector Z given the number of simulations employed, as well as approximately linear in the number of simulations given the dimension. This is an extremely convenient feature of this method, making it applicable even for problems of very high

[5]The inputs to the routine are: m=dimension of multivariate normal vector Z; mu=EZ; w=V(Z); wi=w^{-1}; c=cholesky factor of w; the restriction region is defined by $a < Z < b$; r=number of replications; u=a $m \times r$ matrix of i.i.d. uniform [0,1] variates.

[6]To guard against possible division by 0, a very small positive number $(1.e-100)$ is added to denominators.

[7]$cdfn$=standard normal c.d.f. function and $cdfinvn$=inverse of the standard normal c.d.f. function.

dimensionality without making the implied computational burden intractable.[8]

5 Simulation-Based Diagnostic Tests

In this section I discuss the use of the simulation estimation principles introduced above to devise classical hypothesis testing methods that are free of influential simulation noise. These tests also rely on the fact that whenever intractable expressions do not need to be calculated repeatedly, one can afford to employ a huge number of simulations, thus eliminating the impact of additional noise introduced by the simulations. I illustrate the ideas by developing several new diagnostic tests for popular econometric models.

Consider the classical problem of testing a nested hypothesis (possibly non-linear) against a sequence of local alternatives:

$$\begin{array}{ll} H_0: & g(\theta^*) = 0 \\ H_1: & g(\theta^*) = \frac{\delta}{\sqrt{N}} \end{array} \tag{11}$$

where $g(\cdot)$ is a function $\Re^p \rightarrow \Re^r$ defining the r restrictions on the unknown p-dimensional parameter vector θ^*. Under H_1, the unconstrained maximum likelihood estimator

$$\hat{\theta}_N \equiv \arg\max_{\theta} \ell_N(\theta) \tag{12}$$

is asymptotically efficient, while under H_0 the constrained MLE

$$\bar{\theta}_N \equiv \arg\max_{\theta} \ell_N(\theta) \text{ s.t. } g(\theta) = 0. \tag{13}$$

is efficient. I use the definitions:

$$J = \mathrm{E}\left[\ell_\theta(x|\theta^*) \cdot \ell_\theta(x|\theta^*)'\right] = -\mathrm{E}\left[\ell_{\theta\theta}(x|\theta^*)\right] \tag{14}$$

and

$$\begin{aligned} \hat{J}_{1N} &= \sum_{i=1}^{N} \ell_\theta(x_i|\hat{\theta}_N) \cdot \ell_\theta(x_i|\hat{\theta}_N)', \quad & \bar{J}_{1N} &= \sum_{i=1}^{N} \ell_\theta(x_i|\bar{\theta}_N) \cdot \ell_\theta(x_i|\bar{\theta}_N)', \\ \hat{J}_{2N} &= -\sum_{i=1}^{N} \ell_{\theta\theta}(x_i|\hat{\theta}_N), \quad & \bar{J}_{2N} &= -\sum_{i=1}^{N} \ell_{\theta\theta}(x_i|\bar{\theta}_N), \end{aligned} \tag{15}$$

[8] Some slight non-linearities observed are primarily caused by the fact that due to limitations with random-access-memory workspace at high $M \times R$ values, virtual memory is employed, involving disk-reading and writing. As is well-known, virtual disk-based memory is orders of magnitude slower than real RAM.

where $\ell_\theta(\cdot) \equiv \frac{\partial \ell(\cdot)}{\partial \theta}$ and $\ell_{\theta\theta}(\cdot) \equiv \frac{\partial^2 \ell(\cdot)}{\partial \theta \partial \theta'}$. Note that by the local nature of the deviations from H_0, all four estimators for the true J are consistent under local H_1 as well, because even though $\sqrt{N}(g(\hat{\theta}_N) - g(\theta^*))$ converges in distribution to a normal random vector with a nonzero mean under H_1, the unnormalized by $\sqrt{N}$ quantity $g(\hat{\theta}_N) - g(\theta^*)$ converges in probability to 0. (The same holds for $\bar{\theta}_N$).

It should be noted that linear and non-linear hypotheses of interest typically involve restrictions on the coefficients of explanatory variables, as well as restrictions on the elements of the variance-covariance matrix of the latent variable vector. For example, in the context of discrete choice models, special hierarchical structures correspond to certain correlations among the elements of the unobservable utilities being zero. I will give explicit examples of such hypotheses in subsection 5.4 below.

5.1 Wald Tests

As I explained in the Introduction, simulation estimators that are asymptotically equivalent to the unconstrained maximum likelihood estimator $\hat{\theta}_N$ can be obtained by several methods, notably Two-Step methods beginning from consistent but inefficient estimates (e.g., MSM or MSL tested for bias). Once such an estimator is available, one can define the familiar Wald test statistic for 11 as:

$$W_N \equiv g(\hat{\theta}_N)' \left[\hat{g}'_\theta \cdot \hat{J}_N^{-1} \cdot \hat{g}_\theta \right]^{-1} g(\hat{\theta}_N) \tag{16}$$

As is well-known, this statistic converges to a $\chi^2(r)$ distribution under either H_0 and to a non-central $\chi^2(r)$ with non-centrality parameter $\lambda = \delta' g_\theta(\theta^*)' \cdot J^{-1} \cdot g_\theta(\theta^*)\delta$ under local H_1. Since the calculation of W_N will be done only at $\hat{\theta}_N$, a very large number of simulations can be used in evaluating the quantities $g(\cdot)$, $g_\theta(\cdot)$, and $J(\cdot)$, thus introducing only negligible simulation noise.

It is useful to point out that for tests of exclusion or other linear restrictions, the simulated Wald test approach simply corresponds to obtaining the unrestricted coefficients with *moderate* numbers of simulations, and then constructing standard t and χ^2 tests based on estimates of the variance-covariance matrix of the regression coefficients calculated with a *huge* number of simulations.

5.2 Lagrange Multiplier Tests

Such tests are based on the CUAN estimator that is efficient under the null hypothesis (i.e., equivalent to the restricted estimator $\bar{\theta}_N$), and requires the evaluation of the scores corresponding to the estimator efficient under H_1 (i.e., equivalent to the unrestricted

estimator $\hat{\theta}_N$). The imposition of the restrictions will frequently mean that efficient estimation under H_0 will be tractable without the use of simulation, while the efficient scores will require simulation. The LM statistic is of the form:

$$LM_N \equiv \ell_{N\theta}(\bar{\theta}_N)' \bar{J}_N^{-1} \ell_{N\theta}(\bar{\theta}_N) \tag{17}$$

Since the scores will only be evaluated once, the standard asymptotic theory of Lagrange Multiplier (LM) tests remains applicable by basing the score calculations on a very large number of simulations.

5.3 Likelihood Ratio Tests

The familiar LR statistic

$$LR_N \equiv 2 \cdot [\ell_N(\hat{\theta}_N) - \ell_N(\bar{\theta}_N)], \tag{18}$$

which is asymptotically equivalent to W_N and LM_N under both H_0 and local H_1, is computationally more burdensome than either test since it requires one to obtain efficient estimators under both H_0 and H_1. The same basic principle is still applicable, however, namely that the calculation of the two components of LR_N can be based on a very large number of simulations, thus obviating the need for the development of special asymptotic results to allow for simulation noise. This fact makes the application of LR_N tests very appealing, because they do not require the derivation of possibly complicated expressions like $\ell_{N\theta(\cdot)}$, $g_\theta(\cdot)$, or $J(\cdot)$, but only the evaluation of the log-likelihood function at two points. Analogously to the concluding remarks in the simulated Wald section, the simulated LR test for simple restrictions corresponds to obtaining the restricted and unrestricted parameter estimates based on moderate R, and then employing a huge R for the two restricted and unrestricted likelihood function evaluations that are needed for the LR statistic.

5.4 Testing Hypotheses in Discrete Choice Models

As examples, I discuss here two discrete choice models to illustrate concretely the main issues involved with simulation-based testing.

5.4.1 The Multinomial Probit Model

The MNP model is defined as:

$$y_i = \arg\max_k \{y_{i1}^*, \cdots, y_{ik}^* \cdots, y_{iJ}^*\} \tag{19}$$

where $y_i^* \sim N(X_i\beta, \Omega_i)$ denotes the $J \times 1$ vector of latent utilities of the J alternatives.

The following cases have been identified in the literature as being particularly useful for modelling discrete choice settings in practice. Let $\omega_{\ell m}$ be the (ℓ, m)th element of the variance-covariance matrix Ω_i.

Version	Description	Restrictions on Ω
P1	General MNP	none
P2	MNP with nested hierarchical structure	off-diagonal blocks of 0
P3	Independent but heteroskedastic MNP	$\omega_{\ell m} = 0$, for all $\ell \neq m$
P4	Independent and homoskedastic MNP	$\omega_{\ell m} = 0$, for all $\ell \neq m$, $\omega_{\ell\ell} = \text{c}$ for all ℓ

All three classical testing approaches discussed above are applicable in these cases. Of special usefulness are the Wald and LR tests because all hypotheses under test involve estimating the model with and without equality restrictions imposed, and letting the final round of calculations be based on a very large number of simulations.

In terms of computation, P4 and P3 are extremely straightforward since they correspond to likelihood contributions that are products of $J-1$ univariate normal c.d.f.'s $(g(z))$ integrated over the normal p.d.f. of the normalizing utility. Since this integral is of the form $\int_{-\infty}^{+\infty} g(z)\exp(-z^2/2)dz$, Hermite Gaussian quadrature can be used. Alternatively, though not necessary, one could always use simulation estimation. This would provide a good test of the accuracy of simulation-based versus quadrature-based estimation. Models P2 and P1 require simulation-based estimation.

5.4.2 The Multinomial Ordered Probit Model

As with the MNP model, individual i chooses alternative k that offers the highest utility y_{ik}^*. The analyst, however, observes the full ranking of the J alternatives in terms of the utility they yield, i.e., the analyst observes the J-dimensional vector of indices

$$y_i \equiv (k_1, \cdots, k_J)'$$

such that

$$y_{ik_1}^* \leq y_{ik_2}^* \leq \cdots \leq y_{ik_J}^*. \tag{20}$$

If such information is available, the resulting estimators will be, in general, much better behaved than the ones from the MNP model, since, as **??** shows, the informational content of the MNP model can be quite low in view of the severity of the MNP filtering.

6 Antithetics

Antithetics is one of the leading "variance-reduction" simulation techniques and can be explained as follows: Suppose we want to approximate a function $g(u)$ by using the average over $2R$ simulations:

$$S_{na,2R} = \frac{1}{2R} \sum_{r=1}^{2R} g(u_r)$$

where u_r are i.i.d. draws from the appropriate distribution. As a result, simulator $S_{na,2R}$ with no-antithetics and $2R$ simulations has variance $\frac{1}{4R^2} 2R \cdot V(g(u)) = \frac{1}{2R} V(g(u))$. To introduce antithetics, we define the estimator based on only R i.i.d. u_r draws defined by:

$$S_{a,R} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{2} \left[g(u_r) + g(-u_r) \right]$$

i.e., for each u_r we average $g(\cdot)$ both at u_r and at $-u_r$. This simulator has variance $\frac{1}{4R^2} \sum_{r=1}^{R} \left[2 \cdot V(g(u)) + 2cov(g(u_r), g(-u_r)) \right] = \frac{1}{2R} \left(V(g(u) + cov(g(u), g(-u))) \right)$. Depending on whether or not $g(\cdot)$ is monotonic, the covariance term will be negative, implying that the variance of the simulator that employs antithetics with R simulations will be lower than the basic simulator with $2R$ independent simulations. In addition, $S_{a,R}$ may offer also computational advantages over the $S_{na,2R}$ simulator, if the relative savings in drawing R fewer u_r's (R vs. $2R$) are substantial relative to performing the same number of $g(\cdot)$ evaluations in the two cases (R $g(u)$'s plus R $g(-u_r)$'s vs. $2R$ $g(u)$'s).

For more extensive discussion of the use of antithetics in further improving the GHK simulator, see **?**. Tables 2a–2c below summarize the results from the same set of 84 Monte-Carlo experiments analyzed in **?**. That study did not investigate antithetic variance-reduction and concluded that the best method overall was the GHK simulator, followed by the parametric cylinder function simulator (PCF) under certain conditions. In the experiments here, the simulators considered were GHK and PCF as well as GHKA, which is the GHK method with antithetics built into it. The improvements in terms of mean-squared-error performance controlled for computational requirements in terms of CPU time are quite uniform and substantial. Given how simple it is to program the antithetic modification into the GHK method, the results here suggest that GHKA should supplant the basic GHK as the simulator of choice for routine applications.

7 Conclusions

In this Chapter, I explored ways of recapturing the efficiency property for estimators that rely on simulation. In particular, I showed how this can be achieved by exploiting two-step maximum simulated likelihood (MSL) estimation methods that are familiar in classical applications. I also constructed a diagnostic test for adequacy of number of simulations employed to guarantee negligible bias for the MSL and provided some evidence on the computational requirements of the Geweke-Hajivassiliou-Keane (GHK) simulator as a function of (a) the dimension of the problem and (b) the number of simulations employed in a vectorized context.

This chapter also showeed how to suitably introduce simulation into classical hypothesis testing methods and provide test statistics (simulated Wald, Lagrange Multiplier, and Likelihood Ratio Tests) that are free of influential simulation noise.

Finally, I explained how simulation-variance-reduction techniques can improve substantially the practical performance of the GHK simulator and presented extensive Monte-Carlo evidence confirming this.

Computational Speed of GHK: Effects of Dimensionality and Number of Simulations

Table 1

	M=4[a]	M=8	M=16	M=32	M=64
R= 20[b]	1[c]	3	2	3	5
R= 50	2	4	3	4	11
R= 100	3	5	4	8	11
R= 250	4	6	8	16	33
R= 500	5	7	17	27	76
R= 1000	8	16	33	72	176
R= 2500	16	38	82	181	444
R= 5000	38	77	165	429	1022

a $M \equiv$dimension of $Z \sim N(0, \Omega)$.

b Number of replications in GHK simulator.

c Time in $1/100th$ seconds.

d $\Omega = Toeplitz$ matrix with $\rho = 0.5$.

Note: Restricted region considered is the orthant $Z_i > 0, \forall i$.

Figure 1a
Time vs. Dimension M, Given Replications R

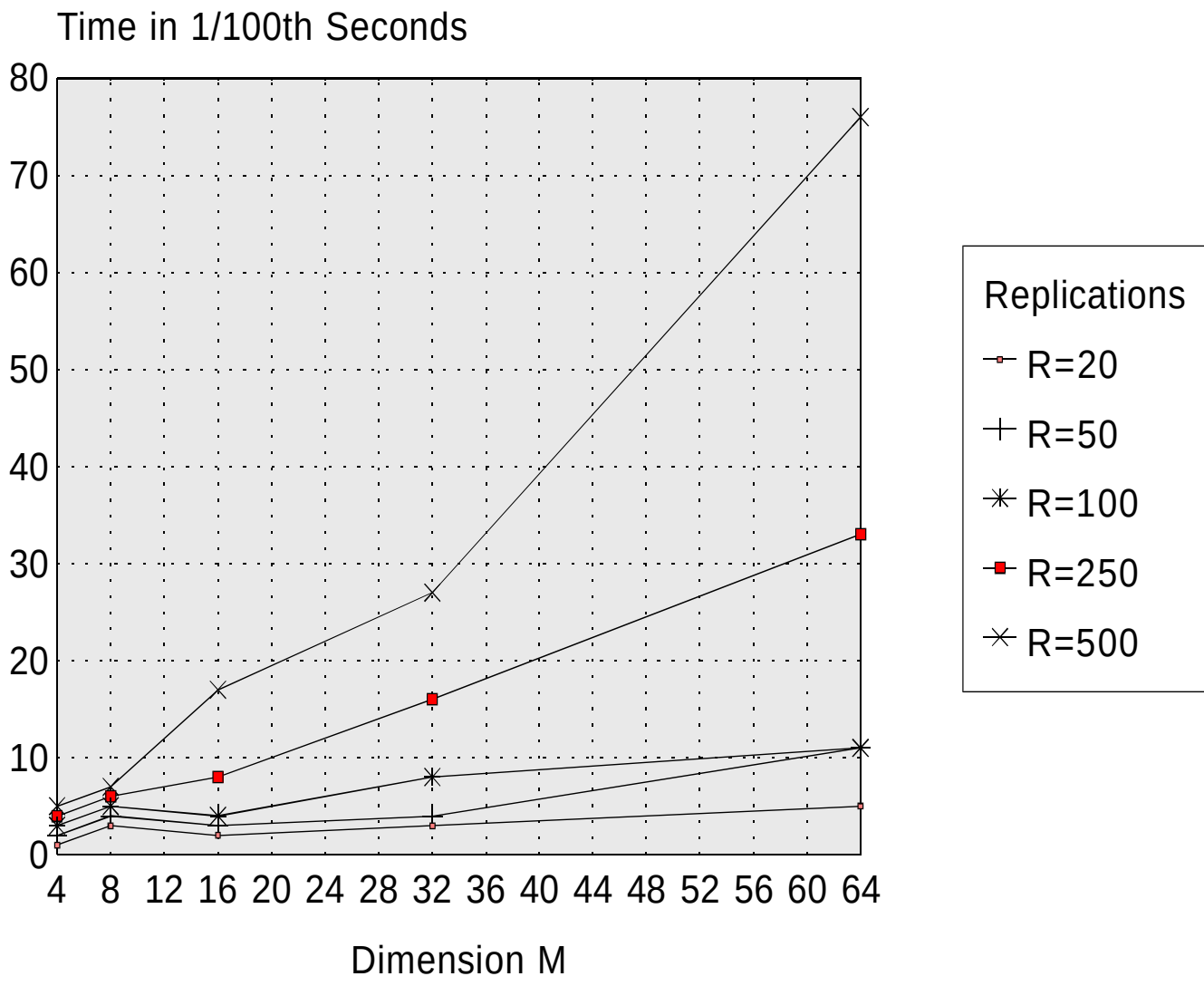

Figure 1b
Time vs. Dimension M, Given Replications R

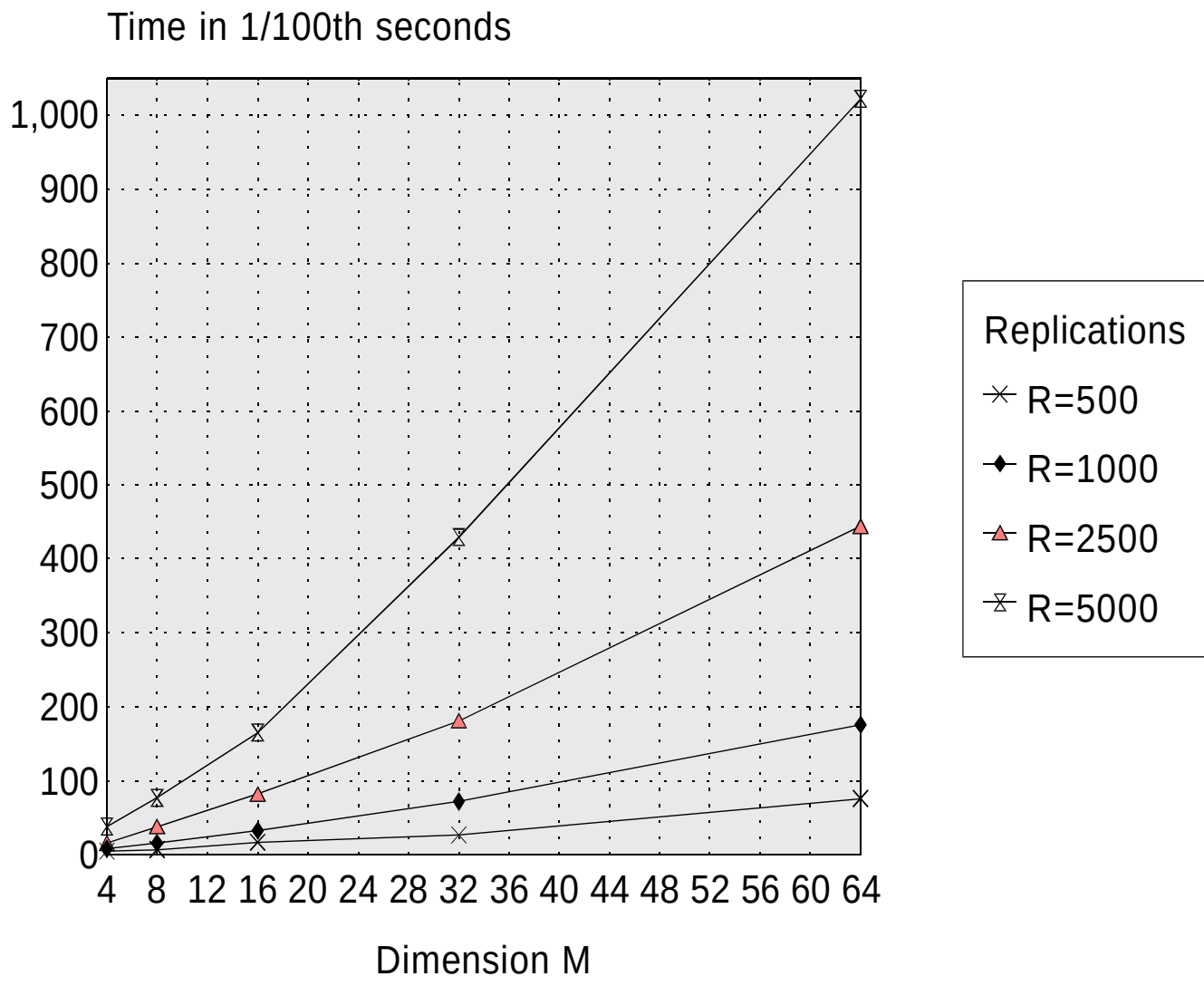

Figure 2
Time vs. Replications R, Given Dimension M

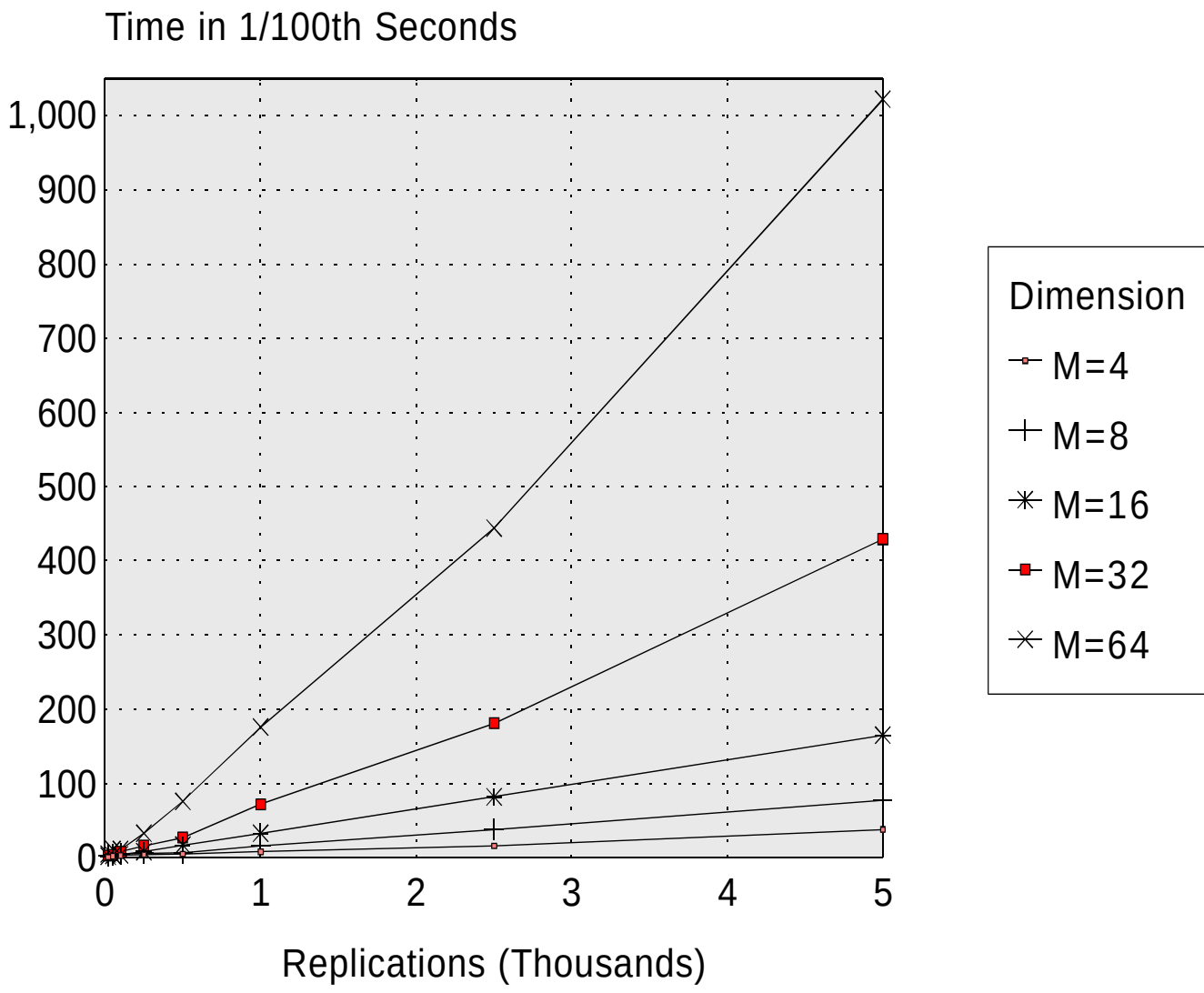

Table 2a
Relative MSE Efficiency: (GHK vs. GHKA)

Quantity Simulated	Simulator	Relative MSE Efficiency[a]
$P(Z \in \mathbf{B})$[b]	GHKA	0.8783
$P(Z \in \mathbf{B})$[b]	GHK	0.7283
$\partial P/\partial \mu_1$	GHKA	0.9904
$\partial P/\partial \mu_1$	GHK	0.4532
$\partial P/\partial \mu_2$	GHKA	0.9900
$\partial P/\partial \mu_2$	GHK	0.4160
$\partial P/\partial \Omega_{11}$	GHKA	0.9395
$\partial P/\partial \Omega_{11}$	GHK	0.6857
$\partial P/\partial \Omega_{12}$	GHKA	0.9692
$\partial P/\partial \Omega_{12}$	GHK	0.5653
$\partial P/\partial \Omega_{22}$	GHKA	0.9508
$\partial P/\partial \Omega_{22}$	GHK	0.6209
Avg. for all Linear Derivs.	GHKA	0.9680
Avg. for all Linear Derivs.	GHK	0.5482
$\partial \ln P/\partial \mu_1$	GHKA	0.9889
$\partial \ln P/\partial \mu_1$	GHK	0.4714
$\partial \ln P/\partial \mu_2$	GHKA	0.9895
$\partial \ln P/\partial \mu_2$	GHK	0.4455
$\partial \ln P/\partial \Omega_{11}$	GHKA	0.9378
$\partial \ln P/\partial \Omega_{11}$	GHK	0.6983
$\partial \ln P/\partial \Omega_{12}$	GHKA	0.9705
$\partial \ln P/\partial \Omega_{12}$	GHK	0.5828
$\partial \ln P/\partial \Omega_{22}$	GHKA	0.9475
$\partial \ln P/\partial \Omega_{22}$	GHK	0.6509
Avg. for all Logarithmic Derivs.	GHKA	0.9668
Avg. for all Logarithmic Derivs.	GHK	0.5698

a Relative Mean-Squared Error efficiency of simulator S averaged over 84 experiments $\equiv \frac{1}{84}\sum_{i=1}^{84} \frac{\text{Lowest MSE for a specific experiment}}{\text{MSE of simulator } s \text{ for experiment } i}$.

b Number of simulations for each simulator selected so as to require equal amounts of CPU time.

Note: Bivariate random vector $Z \sim N(\mu, \Omega)$. Fourteen rectangular regions $\mathbf{B}$ and six correlation structures for Ω analyzed as described in **?**.

Table 2b
Relative MSE Efficiency: (PCF vs. GHKA)

Quantity Simulated	Simulator	Relative MSE Efficiency[a]
$P(Z \in \mathbf{B})$[b]	PCF	0.2685
$P(Z \in \mathbf{B})$[b]	GHKA	0.9621
$\partial P/\partial \mu_1$	PCF	0.3622
$\partial P/\partial \mu_1$	GHKA	0.8693
$\partial P/\partial \mu_2$	PCF	0.2993
$\partial P/\partial \mu_2$	GHKA	0.9303
$\partial P/\partial \Omega_{11}$	PCF	0.6477
$\partial P/\partial \Omega_{11}$	GHKA	0.6947
$\partial P/\partial \Omega_{12}$	PCF	0.5730
$\partial P/\partial \Omega_{12}$	GHKA	0.7961
$\partial P/\partial \Omega_{22}$	PCF	0.4784
$\partial P/\partial \Omega_{22}$	GHKA	0.8595
Avg. for all Linear Derivs.	PCF	0.4721
Avg. for all Linear Derivs.	GHKA	0.8300
$\partial \ln P/\partial \mu_1$	PCF	0.4474
$\partial \ln P/\partial \mu_1$	GHKA	0.8687
$\partial \ln P/\partial \mu_2$	PCF	0.4481
$\partial \ln P/\partial \mu_2$	GHKA	0.8732
$\partial \ln P/\partial \Omega_{11}$	PCF	0.6959
$\partial \ln P/\partial \Omega_{11}$	GHKA	0.6948
$\partial \ln P/\partial \Omega_{12}$	PCF	0.6426
$\partial \ln P/\partial \Omega_{12}$	GHKA	0.7813
$\partial \ln P/\partial \Omega_{22}$	PCF	0.6737
$\partial \ln P/\partial \Omega_{22}$	GHKA	0.7312
Avg. for all Logarithmic Derivs.	PCF	0.5815
Avg. for all Logarithmic Derivs.	GHKA	0.7898

a Relative Mean-Squared Error efficiency of simulator S averaged over 84 experiments $\equiv \frac{1}{84}\sum_{i=1}^{84} \frac{\text{Lowest MSE for a specific experiment}}{\text{MSE of simulator } s \text{ for experiment } i}$.

b Number of simulations for each simulator selected so as to require equal amounts of CPU time.

Note: Bivariate random vector $Z \sim N(\mu, \Omega)$. Fourteen rectangular regions $\mathbf{B}$ and six correlation structures for Ω analyzed as described in **?**.

Table 2c
Relative MSE Efficiency: (GHK vs. PCF vs. GHKA)

Quantity Simulated	Simulator	Relative MSE Efficiency[a]
$P(Z \in \mathbf{B})^{\mathbf{b}}$	PCF	0.2499
$P(Z \in \mathbf{B})^{\mathbf{b}}$	GHK	0.7738
$P(Z \in \mathbf{B})^{\mathbf{b}}$	GHKA	0.8418
$\partial P/\partial \mu_1$	PCF	0.3614
$\partial P/\partial \mu_1$	GHK	0.4299
$\partial P/\partial \mu_1$	GHKA	0.8685
$\partial P/\partial \mu_2$	PCF	0.2981
$\partial P/\partial \mu_2$	GHK	0.4523
$\partial P/\partial \mu_2$	GHKA	0.9288
$\partial P/\partial \Omega_{11}$	PCF	0.6470
$\partial P/\partial \Omega_{11}$	GHK	0.5201
$\partial P/\partial \Omega_{11}$	GHKA	0.6942
$\partial P/\partial \Omega_{12}$	PCF	0.5730
$\partial P/\partial \Omega_{12}$	GHK	0.4872
$\partial P/\partial \Omega_{12}$	GHKA	0.7961
$\partial P/\partial \Omega_{22}$	PCF	0.4748
$\partial P/\partial \Omega_{22}$	GHK	0.6063
$\partial P/\partial \Omega_{22}$	GHKA	0.8441
Avg. for all Linear Derivs.	PCF	0.4709
Avg. for all Linear Derivs.	GHK	0.4992
Avg. for all Linear Derivs.	GHKA	0.8263

(continued)

Table 2c (continued)
Relative MSE Efficiency: (GHK vs. PCF vs. GHKA)

Simulator	Quantity Simulated	Relative MSE Efficiency[a]
$\partial \ln P / \partial \mu_1$	PCF	0.4443
$\partial \ln P / \partial \mu_1$	GHK	0.4447
$\partial \ln P / \partial \mu_1$	GHKA	0.8635
$\partial \ln P / \partial \mu_2$	PCF	0.6254
$\partial \ln P / \partial \mu_2$	GHK	0.4283
$\partial \ln P / \partial \mu_2$	GHKA	0.8690
$\partial \ln P / \partial \Omega_{11}$	PCF	0.6926
$\partial \ln P / \partial \Omega_{11}$	GHK	0.5320
$\partial \ln P / \partial \Omega_{11}$	GHKA	0.6894
$\partial \ln P / \partial \Omega_{12}$	PCF	0.6396
$\partial \ln P / \partial \Omega_{12}$	GHK	0.4972
$\partial \ln P / \partial \Omega_{12}$	GHKA	0.7763
$\partial \ln P / \partial \Omega_{22}$	PCF	0.6712
$\partial \ln P / \partial \Omega_{22}$	GHK	0.5226
$\partial \ln P / \partial \Omega_{22}$	GHKA	0.7268
Avg. for all Log Derivs.	PCF	0.6146
Avg. for all Log Derivs.	GHK	0.4850
Avg. for all Log Derivs.	GHKA	0.7850

a Relative Mean-Squared Error efficiency of simulator S averaged over 84 experiments $\equiv \frac{1}{84}\sum_{i=1}^{84} \frac{\text{Lowest MSE for a specific experiment}}{\text{MSE of simulator } s \text{ for experiment } i}$.

b Number of simulations for each simulator selected so as to require equal amounts of CPU time.

c Bivariate random vector $Z \sim N(\mu, \Omega)$. Fourteen rectangular regions **B** and six correlation structures for Ω analyzed as described in **?**.