Accelerated Monte Carlo Integration :

An Application to Dynamic Latent Variable Models

by

Jean-Francois Richard and Wei Zhang

University of Pittsburgh

November 1995

Paper prepared for the Conference on Simulation-Based Methods in Econometrics, Federal Reserve Bank of Minneapolis, Nov 17-18, 1995.
Financial support from NSF Grant SES- is acknowledged

(Note : Accelerated integration ≠ accelerated typing. A typed version of the paper will be available by the end of November.)

Contact address : Dpt of Economics, Forbes Quad 4D12
University of Pittsburgh, Pittsburgh, PA 15260
e-mail : fantin+@pitt.edu

1. Introduction

Economists are well aware of the fact that the behavior of economic agents often is critically conditioned by latent (unobservable) variables. It is, therefore, hardly surprising that latent variable models have received increased attention over the recent years (made possible by impressive advances in computing power). A few references are

Estimation of latent variable models requires their elimination by marginalization i.e., in the case of continuous variables, by integration of the joint sampling density of the observables and unobservables with respect to the latter. Analytical solutions for such integrals generally are not available (with the important exception of linear Gaussian models for whose evaluation there exist analytical recurrence relationships).

In general one has to rely upon numerical integration. The problem is further complicated by the fact that latent variables often are inherently dynamic to the effect that their elimination requires interdependent (high-dimensional) numerical integration.

One of the most important numerical development in recent years has been the increasing usage of Monte Carlo (MC) simulation techniques as a numerical method for evaluating large-dimensional analytically intractable integrals. See Kloek and van Dijk (1978) or, for more recent developments,

Geweke (1989, 1994) and Richard (1995). The numerical accuracy of MC methods critically depends upon the choice of the auxiliary sampler (the 'importance function') which is used for the random selection of the points at which the integrand has to be evaluated. Unfortunately, 'natural' importance samplers (by which we mean samplers that are direct byproducts of the model specification) often are so 'inefficient' that they are inherently incapable of producing accurate estimates of likelihood functions for DLV models. See e.g. the comments in McFadden (1989) or Pakes and Pollard (1989) to that effect.

Conventional 'acceleration' techniques, as discussed e.g. in Hendry (1984) or Geweke (1994) can help but generally do not solve the problem as they leave the initial sampler unaffected. Danielsson and Richard (1993), hereafter DR, proposed an algorithm for the construction of efficient samplers in high-dimensional problems but their technique is restricted to Gaussian samplers and requires iterations. Richard and Zhang (1995), hereafter RZ, proposed a more general non iterative algorithm which is applicable to a broad class of (sequential) samplers and only requires solving auxiliary weighted least squares problems. The object of the present paper is to discuss the application of their technique to DLV models.

The paper is organized as follows: Section 2 discusses simulated likelihood functions; Accelerated importance sampling is introduced in section 3; its implementation is discussed in section 4, first in the context of a simple stochastic volatility model (section 4.1) and then at a more general level (section 4.2); Numerical and statistical accuracy are discussed in [illegible] Bayesian [illegible] [illegible] section 6 and section 7 concludes.

2. Simulated Likelihood Function

Let $y_t \in \mathbb{R}^n$ denote a vector of random variables observable at time t and $\lambda_t \in \mathbb{R}^p$ a vector of unobservable or latent variables. A sample of size T is available in the form of a matrix $Y' = (y_1 \dots y_T)$. The corresponding matrix $\Lambda' = (\lambda_1 \dots \lambda_T)$ Initial conditions are represented by the matrices Y_0 and Λ_0 respectively. For the ease of exposition we shall assume that Y_0 has been observed.[1]

Let $Y_t' = (Y_0' \; y_1 \dots y_t)$ and $\Lambda_t' = (\Lambda_0' \; \lambda_1 \dots \lambda_t)$. DLV models are typically expressed in the form of a sequence of conditional density functions[2] $\phi(\cdot)$ for (y_t, λ_t) given $(Y_{t-1}, \Lambda_{t-1}, \theta)$, where θ denote a vector of unknown parameters. Let $\phi(y_0, \lambda_0 | \theta)$ denote an assumed density function for $y_0 = \mathrm{vec}\, Y_0$ and $\lambda_0 = \mathrm{vec}\, \Lambda_0$.
The likelihood function associated with the $(T+\ell)n$

---

[2] For the ease of notation, we shall only consider the continuous random variables. Our analysis trivially extends to discrete or mixed random variables.

---

[1] If Y_0 were unknown, it would have to be treated in the same way as Λ_0, i.e. it would have to be integrated out along with the λ's.

matrix Y_T is given by

$$L(\theta; Y_T) = \int \phi(Y_T, \lambda_T | \theta) \, d\lambda_T \qquad (1)$$

where $\phi(Y_T, \lambda_T | \theta)$ denotes the joint sampling density of Y_T and λ_T and is given by

$$\phi(Y_T, \lambda_T | \theta) = \prod_{t=0}^{T} \phi(y_t, \lambda_t | Y_{t-1}, \lambda_{t-1}, \theta) \qquad (2)$$

Note that ϕ is used as a generic notation for all density functions (sampling) associated with the model under consideration. Assumptions relative to the dynamic structure of the model are generally formulated in terms of either one of the following two additional factorizations[3]

$$\phi(y_t, \lambda_t | Y_{t-1}, \lambda_{t-1}, \theta) = \begin{cases} \phi(y_t | Y_{t-1}, \lambda_t, \theta) \cdot \phi(\lambda_t | Y_{t-1}, \lambda_{t-1}, \theta) & (3) \\ \phi(y_t | Y_{t-1}, \lambda_{t-1}, \theta) \cdot \phi(\lambda_t | Y_t, \lambda_{t-1}, \theta) & (4) \end{cases}$$

In general it is not possible to derive from equations (3) and/or (4) operational expressions for the distribution of

---

[3] A cursory look through the literature suggests that factorization (3) prevails. In particular, it applies to cases where a 'state space' representation of the latent process is paired with a (stochastic) 'measurement equation'. See e.g. Harvey (1990) for details.

$\lambda_t | \lambda_{t-1}, Y_T, \theta$ (except for the obvious and rather uninteresting case when λ_t does not 'cause' y_t in the sense of Granger (1969), i.e. where y_t is independent of λ_{t-1}, conditionally on Y_{t-1}). This is merely why the integral in (1) has to be numerically evaluated.

In order to evaluate (1) by Monte Carlo (MC) we first have to construct an <u>auxiliary</u>[4] sampler for the λ's, say

$$p_0(\Lambda_T | Y_T, \theta) = \prod_{t=0}^{T} p_t^0(\lambda_t | \lambda_{t-1}, Y_T, \theta) \qquad (5)$$

Based upon the factorization in (3) and (4) two 'natural' choices for p_t^0 are $\phi(\lambda_t | Y_{t-1}, \lambda_{t-1}, \theta)$ or $\phi(\lambda_t | Y_t, \lambda_{t-1}, \theta)$ though, as illustrated by the example discussed in section 4.1 below, other (possibly more 'efficient') choices may be available. The 'remainder function' g_0 associated with the sampler p_0 is defined as $g_0 = \phi / p_0$ and is partitioned conformably with ϕ and p_0

$$g_0(\Lambda_T; Y_T, \theta) = \prod_{t=0}^{T} g_t^0(\lambda_t; Y_T, \theta) \qquad (6)$$

where

$$g_t^0(\lambda_t; Y_T, \theta) = \frac{\phi(y_t, \lambda_t | Y_{t-1}, \lambda_{t-1}, \theta)}{p_t^0(\lambda_t | \lambda_{t-1}, Y_T, \theta)} \qquad (7)$$

---

[4] p_0 is 'auxiliary' in the sense that it differs from $\phi(\Lambda_T | Y_T, \theta)$, the <u>actual</u> sampling distribution of $\Lambda_T | Y_T, \theta$.

It follows that $L(\theta, Y_T)$, as defined in equation (1), can be rewritten as

$$L(\theta; Y_T) = \int g_0(\lambda_T; Y_T, \theta) \cdot p_0(\lambda_T | Y_T, \theta) \, d\lambda_T \qquad (8)$$

An 'initial' MC estimate of $L(\theta; Y_T)$ (for a preassigned value of θ) is given by

$$\bar{L}_S^0(\theta; Y_T) = \frac{1}{S} \sum_{i=1}^{S} g_0(\tilde{\lambda}_{Ti}; Y_T, \theta) \qquad (9)$$

where $\{\tilde{\lambda}_{Ti}^0; i: 1 \to S\}$ denotes a set of S independent random draws from the initial sampler p_0. In practice,

the λ_{ti}'s are drawn conformably with the sequential factorization of p_0, as given in equation (5), to the effect that $\tilde{\lambda}_{ti}$ denotes a draw from the conditional density $p_E^0(\lambda_t | Y_{t-1}, \tilde{\lambda}_{t-1,i}, \theta)$. The MC sampling variance of L_S^0 is given by

$$\text{Var}\left[L_S^0(\theta; Y_T)\right] = \frac{1}{S} \text{Var}_{p_0}\left[g_0(\lambda_T; Y_T, \theta)\right] \qquad (10)$$

It is now well documented that for a broad range (and by modification thereof) of applications the MC sampling variance of g_0 is so large that accurate MC estimation of $L(\theta; Y_T)$ is utterly impractical. See e.g. McFadden (1989) or Pakes and Pollard (1989). This explains the growing popularity of alternative methods of estimation and, in particular, of the Method of Simulated Moments (MSM) which, relative to inference procedures based on simulated likelihood, is computationally simpler but potentially statistically inefficient.

The object of the present paper is to demonstrate that it is possible to construct accurate MC estimates of the likelihood function itself, at the cost of replacing the numerically inefficient initial MC sampler p_0 by a more efficient one, obtained by application of the generic acceleration principle proposed by RZ.

3. Accelerated Importance Sampling

In this section we discuss how to efficiently evaluate the integral in (8) for given values of Y_T and θ[5]. In order to simplify notation, let $\delta = (\theta; Y_T)$. Equation (5) is rewritten as

$$L(\delta) = \int \phi(\Lambda_T; \delta)\, d\Lambda_T \tag{11}$$

with

$$\phi(\Lambda_T; \delta) = g_0(\Lambda_T; \delta) \cdot p_0(\Lambda_T | \delta) \tag{12}$$

In our experience, p_0 often is so dramatically inefficient that conventional acceleration techniques, as discussed e.g. in Hendry (1984) or Geweke (1994), are incapable of delivering sufficient efficiency gain. The only remedy consists of the replacement of p_0 by a more efficient sampler p_*.

For obvious practical reasons, the search for p_* is restricted to a preassigned class of samplers, the choice of

---

[5] Such calculations will have to be repeated for different values of θ as provided, for example, by an ML optimization algorithm and, possibly also, for different values of Y_T, in the context of an MC simulation of the finite sample (statistical) properties of the relevant estimator of θ.

which will be discussed in section 4 below. Therefore, let M denote a class of MC samplers indexed by an auxiliary parameter vector $\alpha \in A$.

$$M = \{ m(\Lambda_T | \alpha) \; ; \; \alpha \in A \} \tag{13}$$

In practice we shall partition in conformably with p_*. However, for expository purposes, we prefer the proposed acceleration principle at a 'global' level first and discuss its sequential application next.

3.1 General principle

For any arbitrary value $\alpha \in A$, we can rewrite the integral in (11) as

$$L(\delta) = \int \frac{\phi(\Lambda_T, \delta)}{m(\Lambda_T | \alpha)} \cdot m(\Lambda_T | \alpha) \, d\Lambda_T \tag{14}$$

$$= \int g_0(\Lambda_T ; \delta) \cdot \omega(\Lambda_T ; \alpha, \delta) \cdot m(\Lambda_T | \alpha) \, d\Lambda_T \tag{15}$$

where

$$\omega(\Lambda_T ; \alpha, \delta) = \frac{p_0(\Lambda_T | \delta)}{m(\Lambda_T | \alpha)} \tag{16}$$

The corresponding MC estimate of $L(\delta)$ is given by

$$\hat{L}_S(\delta ; \alpha) = \frac{1}{S} \sum_{i=1}^{S} g_0(\tilde{\Lambda}_{Ti} ; \delta) \cdot \omega(\tilde{\Lambda}_{Ti} ; \alpha, \delta) \tag{17}$$

where $\{\tilde{\Lambda}_{Ti}; i: 1-\}$ denotes a set of S independent random draws from m. As shown in RZ, its MC sampling variance is given by

$$V_S(\delta;\alpha) = \frac{1}{S}\cdot\left[\int \frac{\phi^2(\Lambda_T;\delta)}{m(\Lambda_T|\alpha)}\, d\Lambda_T - L^2(\delta)\right] \tag{18}$$

$$= \frac{1}{S}\cdot L(\delta)\cdot \int h\left[d(\Lambda_T;\delta,\alpha)\right]\cdot \phi(\Lambda_T;\delta)\, d\Lambda_T \tag{19}$$

where

$$d(\Lambda_T;\delta,\alpha) = \ln\left[\frac{\phi(\Lambda_T;\delta)}{L(\delta)\cdot m(\Lambda_T|\alpha)}\right] \tag{20}$$

$$h(d) = e^{d} + e^{-d} - 2 \tag{21}$$

If, in particular, there existed $\alpha_0 \in A$ such that

$$m(\Lambda_T|\alpha_0) \equiv \frac{\phi(\Lambda_T;\delta)}{L(\delta)} \tag{22}$$

then $V_S(\delta;\alpha_0) \equiv 0$. More generally we should aim at finding values of α such that $m(\Lambda_T|\alpha)$ closely 'mimics' $\phi(\Lambda_T;\delta)$ in Λ_T. However, there does not appear to exist easy ways of directly minimizing $V_S(\delta;\alpha)$ with respect to α. We propose instead to replace $h(d)$ in equation ( ) by d^2 on the grounds that its Taylor Series expansion around zero is given by

$$h(d) = 2\cdot\sum_{i=1}^{\infty}\frac{d^{2i}}{(2i)!} \tag{23}$$

and that we expect d to remain close to zero for 'efficient' choices of α. A more formal argument to that

effect can be found in RZ.

Hence, we propose to solve the following minimization problem

$$\alpha_*(\delta) = \underset{\alpha \in A}{\arg\min} \left[Q(\delta; \alpha) \right] \tag{24}$$

where

$$Q(\delta;\alpha) = \int \left[\ln \phi(\lambda_T;\delta) - \ln L(\delta) - \ln m(\lambda_T|\alpha) \right]^2 \cdot \phi(\lambda_T;\delta)\, d\lambda_T \tag{25}$$

In practice we can use the initial sampler p_0 to construct the following MC estimate of $Q(\delta;\alpha)$

$$\hat{Q}_N(\delta;\alpha) = \frac{1}{N} \sum_{i=1}^{N} \left[\ln \phi(\tilde{\lambda}^0_{Ti};\delta) - \ln L(\delta) - \ln m(\tilde{\lambda}^0_{Ti}|\alpha) \right]^2 \cdot g_0(\tilde{\lambda}^0_{Ti};\delta) \tag{26}$$

where $\{\tilde{\lambda}^0_{Ti};\ i: 1 \to N\}$ denotes a set of N independent random draws from p_0. Whence our proposed 'efficient' importance sampler is given by

$$p_*(\lambda_T|\delta) = m(\lambda_T \mid \hat{\alpha}_N(\delta)) \tag{27}$$

where

$$\hat{\alpha}_N(\delta) = \underset{\alpha \in A}{\arg\min} \left[\hat{Q}_N(\delta;\alpha) \right] \tag{28}$$

In other words $\hat{\alpha}_N(\delta)$ is obtained by application of a Weighted (Non linear) least Squares (W(NL)LS) procedure applied to an auxiliary data set constructed with random draws from p_0. The constant $\ln L(\delta)$ in

equation (26) is to be treated as an unconstrained intercept since p_0 cannot produce an accurate estimate of $\ln L(\delta)$ itself[6]. Our final (efficient) MC estimate of $L(\delta)$ is given by

$$\bar{L}_S^*(\delta) = \frac{1}{S}\sum_{i=1}^{S} g_0(\tilde{\Lambda}_{Ti}^*;\delta)\cdot\omega(\tilde{\Lambda}_{Ti}^*;\delta) \qquad (29)$$

where

$$\omega(\tilde{\Lambda}_{Ti}^*;\delta) = \frac{p_0(\tilde{\Lambda}_{Ti}^*;\delta)}{p_*(\tilde{\Lambda}_{Ti}^*;\delta)} \qquad (30)$$

and $\{\tilde{\Lambda}_{Ti}^*; i: 1\to S\}$ denotes a set of S independent random draws from p_*.

3.2 Sequential acceleration

In the context of DLV models all densities are given in sequential form as in equation (1). Let, therefore partition $m(\Lambda_T|\alpha)$ conformably

$$m(\Lambda_T|\alpha) = \prod_{t=0}^{T} m_t(\lambda_t|\Lambda_{t-1},\alpha_t) \qquad (31)$$

---

[6] We could construct a two-step estimate of $L(\delta)$ as follows: The first step consists of the procedure we just described with unconstrained intercept; The second step reruns the WNLLS procedure replacing $L(\delta)$ in equation (26) by its (accurate) first step MC estimate. As discussed in RZ, the efficiency gain produced by this second step appear to be insignificant

with $\alpha' = (\alpha'_0 \dots \alpha'_T)$. The corresponding partitionings for the functions ϕ, p_0 and g_0 are found in equations (1), (5) and (7) respectively. For the ease of exposition we shall keep using the shorthand notation introduced in section 3.1. The correspondance between the two sets of notation is found in the following equations

$$\phi_t(\lambda_t ; \delta) = \phi(y_t, \lambda_t \mid y_{t-1}, \lambda_{t-1}, \theta) \tag{32}$$

$$g_t^\circ(\lambda_t ; \delta) = g_t^\circ(y_t \mid y_{t-1}, \lambda_t, \theta) \tag{33}$$

$$p_t^\circ(\lambda_t \mid \lambda_{t-1}, \delta) = p_t^\circ(\lambda_t \mid y_{t-1}, \lambda_{t-1}, \theta) \tag{34}$$

In order to provide heuristic support for the sequential version of our acceleration technique, we first provide the sequential counterpart of the condition for an 'ideal' MC sampler, as given by formula (22).

Theorem 1. Condition (22) holds if and only if there exist $\{\alpha_t^\circ ; t = 0 \to T\}$ such that

$$m_t(\lambda_t \mid \lambda_{t-1}, \alpha_t^\circ) = \frac{k_t(\lambda_t ; \delta)}{\chi_t(\lambda_{t-1}, \delta)} \tag{35}$$

where the k_t's and χ_t's are given by the backward recursion :

$$\chi_{T+1}(\Lambda_T ; \delta) \equiv 1 \tag{36}$$

$$k_t(\Lambda_t ; \delta) = \phi_t(\Lambda_t ; \delta) \cdot \chi_{t+1}(\Lambda_t ; \delta) \tag{37}$$

$$\chi_t(\Lambda_{t-1} ; \delta) = \int k_t(\Lambda_t ; \delta)\, d\lambda_t \tag{38}$$

whence

$$\chi_0(\cdot ; \delta) \equiv L(\delta) \tag{39}$$

Proof : (i) Sufficiency : Under conditions (36) to (38) we have

$$\phi(\Lambda_T ; \delta) = \prod_{t=0}^{T} \left[\frac{k_t(\Lambda_t ; \delta)}{\chi_{t+1}(\Lambda_t , \delta)} \right]$$

$$= \chi_0(\cdot ; \delta) \cdot \prod_{t=0}^{T} \left[\frac{k_t(\Lambda_t ; \delta)}{\chi_t(\Lambda_{t-1} ; \delta)} \right]$$

$$= L(\delta) \cdot m(\Lambda_T | \alpha_0)$$

with $\alpha_0 = \{ \alpha_t^0 ; t : 0 \to T \}$.

(ii) Necessity obtains by backward induction. See RZ for details.

□

In words which are familiar to Bayesians k_t, as defined in (37), is a 'density kernel' for m_t and χ_t is its integrating constant (w.r.t. λ_t only). The

following notation will be used

$$m_t(\lambda_t | \Lambda_{t-1}, \delta) \propto k_t(\lambda_t; \delta) \qquad (40)$$

In practice we do not expect formula (35) to hold. Nevertheless, it unambiguously suggests that it would be suboptimal to require that m_t individually 'mimics' ϕ_t for $t: T \to 0$. The point is that the integrating constant of ϕ_t with respect to λ_t depends on Λ_{t-1}, while that of m_t doesn't (being equal to one by definition). It follows that the sequential construction of an efficient sampler inherently requires backward transfer of appropriate integrating constants, in line with formula (35).

Note that it would be impractical to transfer back integrating constants for the ϕ_t's, as these are not analytically available. In contrast, the χ_t's will have analytical expression for a broad range of sequential samplers. Hence, theorem 1 provides the key insight for the construction of a sequential sampler based upon an operational [transfer backward] rule for integrating constants.

Let M_t denote a parametric class of sequential samplers

$$M_t = \{ m_t(\lambda_t | \Lambda_{t-1}; \alpha_t) \; ; \; \alpha_t \in A_t \} \qquad (41)$$

Let K_t denote the corresponding set of density kernels

$$K_t = \{ k_t(\lambda_t ; \alpha_t) \; ; \; \alpha_t \in A_t \} \tag{42}$$

The correspondance between M_t and K_t is characterized by the identity

$$m_t(\lambda_t | \lambda_{t-1} ; \alpha_t) = \frac{k_t(\lambda_t ; \alpha_t)}{\chi_t(\lambda_{t-1} ; \alpha_t)} \tag{43}$$

where

$$\chi_t(\lambda_{t-1} ; \alpha_t) = \int k_t(\lambda_t | \lambda_{t-1} ; \alpha_t) \, d\lambda_t \tag{44}$$

Note that in formula (43) we can multiply k_t and χ_t by an arbitrary function $\tau(\lambda_{t-1})$ without changing m_t (which, as we just discussed, is a key component of sequential 'marching' between ϕ and m). It follows that equation (43) defines an equivalence relationship on K_t, where equivalence classes regroup all density kernels associated with a common density m_t in M_t.

In line with formula (37), the proposed sequential version of our acceleration principle consists of the (backward) search for a sequence of α_t's such that

$$\frac{k_t(\lambda_t ; \alpha_t)}{\chi_{t+1}(\lambda_t ; \alpha_{t+1})} \quad \text{'mimics'} \quad \phi_t(\lambda_t ; \delta) \tag{45}$$

for t $T \searrow 0$, where the χ_t's are given by (44). This condition is tantamount to requiring that k_t / χ_{t+1}

could serve as an 'efficient' sampler for ϕ_t in the auxiliary integral[7]

$$I_t(\delta) = \int \phi_t(\lambda_t;\delta) \cdot \prod_{t=0}^{t-1} p_t^\circ(\lambda_t|\lambda_{t-1},\delta)\, d\lambda_t \qquad (46)$$

Application of the general principle introduced in section 3.1 leads to the following sequence of (low-dimensional) optimization problems

$$\hat{\alpha}_{t,N}(\delta) = \underset{\alpha_t \in A_t}{\text{Arg Min}} \left[\hat{Q}_{t,N}(\delta;\alpha_t)\right] \qquad (47)$$

where

$$\hat{Q}_{t,N}(\delta;\alpha_t) = \frac{1}{N}\sum_{i=1}^{N}\left[\ln \phi_t(\tilde{\lambda}_{ti}^\circ;\delta) + \ln \chi_{t+1}(\tilde{\lambda}_{ti}^\circ;\hat{\alpha}_{t+1,N}(\delta)) - \ln k_t(\tilde{\lambda}_{ti}^\circ;\alpha_t)\right]^2 g_t^\circ(\tilde{\lambda}_{ti}^\circ;\delta) \qquad (48)$$

Note that, following equation (46), the $\tilde{\lambda}_{ti}^\circ$'s all are subsets of a common set of $\{\tilde{\lambda}_{Ti}^\circ;\ i:1\to N\}$ independently drawn from the initial sampler p_0. The corresponding

---

[7] Note that we don't expect to have an analytical expression for the integrating constant of k_t/χ_{t+1}. This is irrelevant since we do not propose to evaluate $I_t(\delta)$, (actually) where only use is that of providing a formal validation for the algorithm which follows.

sequence of 'efficient' samplers is given by

$$\hat{p}_t(\lambda_t | \lambda_{t-1}, \delta) = \frac{k_t(\lambda_t ; \hat{\alpha}_{t,N}(\delta))}{\chi_t(\lambda_t ; \hat{\alpha}_{t,N}(\delta))} \qquad (49)$$

where χ_t is the (analytical) integration constant of the density kernel k_t, as defined in equation (44).

The 'sequential' version of our acceleration principle cannot be expected to be as efficient as its 'general' counterpart. On the other hand, it consists of a sequence of T+1 low dimensional WNLLS optimization problems in the individual α_t's while the general algorithm requires joint optimization in $\alpha' = (\alpha_0' \ldots \alpha_T')$ and is, therefore, expected to be prohibitively expensive to run for large T's.

Our current experience, which is based upon such applications as that detailed in sections 4.1 and 5 below, unequivocally suggests that our sequential algorithm can produce highly accurate MC estimates of likelihood functions (and their by-products) for DLV models at reasonable computing costs even for very large T's (1,500 to 2,000+).

4. Implementation

In order to motivate additional technical assumptions that lead to a particularly performant version of our sequential algorithm in the context of (a broad range of) DLV models, we

First discuss its application to a simple dynamic stochastic volatility model.

4.1 A dynamic stochastic volatility model

Let y_t denote a (univariate) observable return variable and $\lambda_t > 0$ a latent (stochastic) 'volatility' variable. The bivariate model we propose to analyse consists of a static measurement process

$$y_t | \lambda_t \sim N(0, \lambda_t) \tag{50}$$

and of a dynamic latent process for λ_t. As for the latter, a broad variety of specifications can be found in the literature. Early contributions, where λ is interpreted as a 'mixing' variable, often assumed independence in order to preserve numerical tractability[8]. See, for example, Clark (1973), Epps and Epps (1976), or Tauchen and Pitts (1983). Dynamic equations for λ_t predominantly appear in the form of G(ARCH) processes whereby λ_t is assumed to be a <u>non-stochastic</u> function of past λ's and/or σ^2's (in which case the evaluation of $L(\theta; Y_T)$ requires no integration at all).

---

[8] When the λ_t's are assumed to be independent over time, the integral in (1) reduces to a product of univariate integrals for which there exists a broad range of efficient evaluation procedures.

A few references are Engle (1982), Bollerslev (1986), Nelson (1991), or Friedman and Laibson (1989). Dynamic stochastic equations for λ_t have been estimated by the method of moments. See, for example, Taylor (1986), Melino and Turnbull (1990), or Duffie and Singleton (1988). Danielsson and Richard (1993) apply an Accelerated Gaussian Importance Sampler (AGIS), which in many ways is a precursor of the more general procedure proposed here, to a simple bivariate model for daily S&P 500 data ($T = 2,022$).

In the present paper we shall assume that $\lambda_t | \lambda_{t-1}$ has an Inverted-Gamma distribution. There are several reasons for introducing that assumption: (i) First and foremost, (Inverted) gamma distributions are widely used in statistics ~~(if not in econometrics)~~ to model non negative random variables[9]; see, for example, Cox and Oakes (1984) or Kalbfleisch and Prentice (1980); (ii) it serves to illustrate the applicability of the proposed algorithm to non-Normal distributions; and (iii) it produces a particularly simple version of our algorithm.

Let, therefore,

---

[9] Bayesians, in particular, would immediately recognize that an Inverted Gamma distribution for λ_t constitutes a 'natural' complement to equation (50). Such distributions are widely used as 'Natural Conjugate' priors for unknown variances in regression models. See for example Zellner (1971)

Properties of the IG distribution and selection of a distribution for λ_0 are discussed in Appendix.

$$\lambda_t | \lambda_{t-1}, \theta \sim IG(\gamma + \tau\lambda_{t-1}, \nu) \tag{51}$$

where $\theta' = (\gamma, \tau, \nu)$ and $IG(s, \nu)$ represents an Inverted Gamma distribution with density function[10]

$$h_{ig}(\lambda | s, \nu) = \left[\left(\frac{s}{2}\right)^{\frac{1}{2}\nu} / \Gamma\left(\frac{1}{2}\nu\right)\right] \cdot \lambda^{-\frac{1}{2}(\nu+2)} \cdot \exp -\frac{1}{2}\frac{s}{\lambda} \tag{52}$$

for $\lambda > 0$, $s > 0$ and $\nu > 0$. For the case of exposition, all constants which have no bearing on the actual computations – such as those which are included between brackets in equation (52) – are deleted from notation in the sequel of our discussion.

The function ϕ_t associated with equations (50) and (51) is given by

$$\phi_t(\lambda_t; \delta) \propto (\gamma + \tau\lambda_{t-1})^{\frac{1}{2}\nu} \cdot \lambda_t^{-\frac{1}{2}(\nu+3)} \cdot \exp\left[-\frac{1}{2\lambda_t}(\gamma + y_t^2 + \tau\lambda_{t-1})\right] \tag{53}$$

Our first task is that of factorizing ϕ_t into a (sequential) initial sampler p_t° and a remainder function g_t°. A 'natural' choice for p_t° would be the latent process itself, as given in equation (51). In the present case, however, examination of ϕ_t immediately suggests using for p_t° an IG distribution with parameters $(\gamma + y_t^2 + \delta\lambda_{t-1}, \nu+1)$ whence

---

[10] The implied restriction that $\gamma + \tau\lambda_{t-1} > 0$ is 'non binding' for all calculations which are reported in section 4.

$$p_t^\circ(\lambda_t|\lambda_{t-1},\delta) \propto (\gamma+y_t^2+\tau\lambda_{t-1})^{\frac{1}{2}(\nu+1)} \cdot \lambda_t^{-\frac{1}{2}(\nu+3)} \cdot \exp\left[-\frac{1}{2\lambda_t}(\gamma+y_t^2+\tau\lambda_{t-1})\right] \quad (54)$$

$$g_t^\circ(\lambda_t;\delta) \propto (\gamma+\tau\lambda_{t-1})^{\frac{1}{2}\nu} \cdot (\gamma+y_t^2+\tau\lambda_{t-1})^{-\frac{1}{2}(\nu+1)} \quad (55)$$

It also appears that K_t, the set of density kernels from which an 'efficient' kernel will be selected, should itself consist of IG density kernels (We shall argue in section 4.2 below that it generally helps to select for K_t a set of density kernels which comprises the latent posterior itself). We do not choose a specific parametric representation of K_t at this stage of the discussion as an operational characterization thereof naturally emerges from our subsequent analysis.

(c) Period T : Note that ϕ_T already is in the form of an IG density kernel with parameters $(\gamma+y_T^2+\tau\lambda_{T-1}, \nu+1)$. Therefore, no optimization is required and an efficient choice for k_T is given by

$$k_T(\lambda_T;\delta) \equiv \phi_T(\lambda_T,\delta) \quad (56)$$

It follows immediately that, except for irrelevant constants, χ_T is given by

$$\chi_T(\lambda_{T-1};\delta) \propto g_T(\lambda_{T-1};\delta) \quad (57)$$

where g_T has been defined in equation (55).

(ii) Period t ($t: T-1 \searrow 0$) : It will appear by recursion that $\chi_{t+1}(\lambda_t; \delta)$ only depends on λ_t (and δ). Furthermore as mentioned earlier, ϕ_t itself already is in the form of an IG density kernel. Finally, products of IG density kernels are themselves IG density kernels. These considerations immediately suggest parametrizing k_t in the following way

$$k_t(\lambda_t; \alpha_t) \equiv \phi_t(\lambda_t; \delta) \cdot \kappa_t(\lambda_t; \alpha_t) \tag{58}$$

where"

$$\kappa_t(\lambda_t; \alpha_t) = \lambda_t^{-\frac{1}{2}a_t} \cdot \exp\left[-\tfrac{1}{2}\left(b_t + \frac{c_t}{\lambda_t}\right)\right] \tag{59}$$

with $\alpha_t' = (a_t, b_t, c_t)$. The specific characterization of k_t offers the advantage that, as we substitute equation (58) into equation (48), ϕ_t immediately cancels out! We are then left with a particularly simple expression for the objective function $\hat{Q}_{t,N}$ which is given by

$$\hat{Q}_{t,N}(\delta; \alpha_t) = \frac{1}{N}\sum_{i=1}^{N}\left[\ln \chi_{t+1}(\tilde{\lambda}^{o}_{ti}; \hat{\alpha}^{*}_{t+1,N}(\delta)) - \ln \kappa_t(\tilde{\lambda}^{o}_{ti}; \alpha_t)\right] \cdot g_t(\tilde{\lambda}^{o}_{t-1,i}; \delta) \tag{60}$$

" The motivation for introducing b_t in equation (59) originates from the fact that, as discussed earlier, we have to include an unconstrained intercept in the WNLLS regression associated with equation (48).

where g_t and κ_t have been defined in equation (55) and (59) respectively. As for χ_t it is obtained by integration of k_t, as given in (58), with respect to λ_t

$$\chi_t(\lambda_{t-1}; \alpha_t) \propto (\gamma + \tau\lambda_{t-1})^{\frac{1}{2}\nu} \cdot (\gamma + c_t + y_t^2 + \tau\lambda_{t-1})^{-\frac{1}{2}(\nu + a_{t+1})} \qquad (61)$$

In summary, the two simple steps which are required in order to produce an efficient sequential sampler are:

(i) We use the initial sampler $\{p_t^\circ\}$, as defined in equation (54) to produce a set of N i.i.d draws $\tilde{\lambda}_{t,i}^\circ$; $i: 1 \to N\}$;

(ii) We run the following T+1 weighted <u>linear</u> LS regression problems for $t: T \searrow 0$

- dependent variable : $\ln \chi_{t+1}(\tilde{\lambda}_{t,i}^\circ, \hat{\alpha}_{t+1,N}^*(\delta))$
- regressors : cste, $\frac{1}{\tilde{\lambda}_{t,i}^\circ}$, $\ln \tilde{\lambda}_{t,i}^\circ$
- weights : $g_t(\tilde{\lambda}_{t,i}^\circ, \delta)$

with $i: 1 \to N$.

Note that these calculations have to be rerun for all relevant values of θ as produced, for example, by an ML optimization routine. As discussed in RZ - see also McFadden (1989), or Pakes and Pollard (1989) - all calculations are run under a set of Common Random Numbers (CRN's) in order to 'smooth' the simulated likelihood function. In other words, our $\{\tilde{\lambda}_{t,i}^\circ\}$ are all obtained from a common set of $(T+1).N$ uniform

draws by 'inversion' of the distribution function of our IG samples [12]

Numerical results will be presented in section 5 below. The example we just discussed is particularly simple. In general we do not expect ϕ_t to belong to K_t. Nevertheless, a weaker form of equation (58) applies to a broad class of DLV models and serves as the basis of the algorithm which is discussed in section 4.2 below.

In order to introduce this more general case, consider how the computations we just described would be affected if we chose the latent process itself, as given in (51), as our initial sampler instead of p_t in equation (54). Equations (54) and (55) would be replaced by

$$\tilde{p}_t^o(\lambda_t | \lambda_{t-1}, \delta) \propto (\gamma + \tau\lambda_{t-1})^{\frac{1}{2}\nu} \cdot \lambda_t^{-\frac{1}{2}(\nu+2)} \exp\left[-\frac{1}{2\lambda_t}(\gamma + \tau\lambda_{t-1})\right] \quad (62$$

$$\tilde{g}_t^o(\lambda_t; \delta) \propto \lambda_t^{-\frac{1}{2}} \exp\left(-\frac{y_t^2}{2\lambda_t}\right) \quad (63$$

If for some reason (related, for example, to the availabilit

---

[12] As discussed, for example, in Devroye (1986), 'inversion' constitutes a (relatively) inefficient technique for the generation of IG random variates. In our experience this inefficiency is more than compensated by the gain in the smoothness of the simulated likelihood function and, furthermore, can be greatly reduced by judicious usage of interpolation applied to a precomputed table of auxiliary fractiles.

of a preprogramed algorithm) we were to recognize that $\tilde{p}_t^*$ is in the form of an IG kernel, while ignoring that so is $\tilde{g}_t^*$, we might replace k_t in equation (58) by

$$\tilde{k}_t(\lambda_t; \tilde{x}_t) \equiv \tilde{p}_t^*(\lambda_t; \delta) \cdot \kappa_t(\lambda_t, \tilde{x}_t) \tag{64}$$

where κ_t has been defined in (59). Under these changes ϕ_t no longer cancels out in equation (48), but $\tilde{p}_t^*$ does. Therefore, we would now use a weighted linear LS with dependent variable in $[X_{t+1} \cdot \tilde{g}_t^*]$, same regressors as before and weights $\tilde{g}_t^*$. Since, however, $\tilde{g}_t^*$ actually is in the form of an IG kernel, it will be the case that ~~except for MC small sample variations induced by the change of initial sampler and weight function~~, the optimal values $\hat{a}_t$ in the initial regression and $\tilde{a}_t$ in the modified one will be related in the following way

$$\tilde{a}_t = \hat{a}_{t+1} \qquad \tilde{b}_t = \hat{b}_t \qquad \tilde{c}_t = \hat{c}_t + y_t^2 \tag{65}$$

Actual MC estimates would differ in finite samples due to the change of initial sampler and weight function. In other words, except for MC small sample perturbations, the optimal kernel k_t and its integrating constant χ_t inherently are invariant relative to the change from p_t to $\tilde{p}_t$. In practice, p_t turns out to be more efficient than $\hat{p}_t$ in delivering a close approximation to the optimal sampler under (very) small N's.

4.2 An operational algorithm

The example we just discussed indicates that the simplicity of the proposed algorithm depends upon the choice of K_t, the class of density kernels from which an efficient sampler is to be selected. The following two conditions may a useful role in our design:

(i) We should aim at selecting K_t's which contain kernels that are good functional approximations to the ϕ_t's themselves. More specifically, it ought to be the case that, for relevant values of δ, there exist $\alpha_t^i(\delta)$'s in A_t such that

$$\phi_t(\lambda_t, \delta) = k_t(\lambda_t; \alpha_t^i(\delta)) \cdot r_t(\lambda_t; \delta) \qquad (66)$$

where r_t are as 'simple' functions of λ_t as possible (If, in particular $\phi_t \in K_t$, as in the example we discussed above, then $r_t \equiv 1$) More generally, there exist a broad class models which can be factorized according to equation (4) and for which the 'measurement' process which transforms the latent λ_t^i's into observable y_t's takes (very) simple forms. Under such circumstances we should select K_t's which contain the latent process itself and define k_t and r_t accordingly as

$$k_t(\lambda_t; \alpha_t^i(\delta)) = \phi(\lambda_t | Y_{t-1}, \lambda_{t-1}, \theta) \qquad (67)$$

$$r_t(\lambda_t; \delta) = \phi(y_t | Y_{t-1}, \lambda_t, \theta) \tag{68}$$

(2) K_t should be 'closed under multiplication' in the sense of DeGroot (1970, section 9.3).

Definition 1. A class $K = \{k(\cdot; \alpha); \alpha \in A\}$ of density kernels is closed under multiplication if and only if, for any two α_1 and α_2 in K, there exists an α_3 in K such that

$$k(\cdot; \alpha_3) \propto k(\cdot; \alpha_1) \cdot k(\cdot; \alpha_2) \tag{69}$$

We shall use the following notation to represent the operator which maps (α_1, α_2) into α_3 via transformation (69)

$$\alpha_3 = \alpha_1 * \alpha_2 \tag{70}$$

Definition 1 plays a central role in Bayesian statistics as the cornerstone of the concept of 'natural conjugate' prior density. It is satisfied for a broad range of distributions from the exponential family[13]

---

[13] As shown e.g. by DeGroot (1970, section 9.3), if the family of density kernels $\{f_n(\cdot|\omega); \omega \in \Omega\}$ has a sufficient statistic T_n of fixed dimension, i.e. if there exists a function v_n such that

$$f_n(x_1 \ldots x_n|\omega) \propto v_n[T_n(x_1 \ldots x_n); \omega]$$

and if v_n is integrable w.r. to t, then there exists a density function $g(\cdot|t, n)$ on Ω such that

$$g(\omega|t, n) \propto v_n(t; \omega)$$

If K_t is closed under multiplication and if equation (66) holds, we can factorize the approximating kernel k_t in equation (45) as

$$k_t(\lambda_t; \alpha_t^*(\delta)) = k_t(\lambda_t; \alpha_t'(\delta)) \cdot k_t(\lambda_t; \alpha_t^\circ) \tag{71}$$

It follows that $k_t(\cdot; \alpha_t'(\delta))$ cancels out in the objective function (48). Our 'efficient' chain for α_t^* is then given by

$$\hat{\alpha}_{t,N}^*(\delta) = \alpha_t'(\delta) * \hat{\alpha}_{t,N}^\circ(\delta) \tag{72}$$

$$\hat{\alpha}_{t,N}^\circ(\delta) = \underset{\alpha_t^\circ \in A_t}{\operatorname{Arg\,Min}} \left[\hat{Q}_{t,N}(\delta; \alpha_t^\circ) \right] \tag{73}$$

$$\hat{Q}_{t,N}(\delta; \alpha_t^\circ) = \frac{1}{N} \sum_{i=1}^{N} \left[\ln r_t(\tilde{\lambda}_{ti}^\circ; \delta) + \ln \chi_{t+1}(\tilde{\lambda}_{ti}^\circ; \hat{\alpha}_{t+1,N}^*(\delta)) \right. \\ \left. - \ln k_t(\tilde{\lambda}_{ti}^\circ; \alpha_t^\circ) \right]^2 g_t^\circ(\tilde{\lambda}_{ti}^\circ; \delta) \tag{74}$$

where the $\tilde{\lambda}_{ti}^\circ$'s are subsets of a common set $\{\tilde{\lambda}_{ti}^\circ; i = 1 \to N\}$ of i.i.d. draws from p_0 and χ_{t+1} denotes the integrating constant of k_t as given by equation (44).

As illustrated by the examples discussed in section 4.1 and by the numerical results which are provided in section 5 below, the algorithm described by equations (72) to (74) is easy to program, fast and very efficient in particular for DLV models characterized by a simple measurement process

The algorithm we just described differs from that proposed by DA in two key respects: (1) it is not restricted to Gaussian samplers, and; (2) the objective function used by DA differs from equation (74) in that it does not include the factors χ_{t+1} and g_t°.

As discussed in greater details in RZ, the omission of χ_{t+1} from equation (74) is inconsequential for Gaussian samplers as χ_{t+1} itself is in the form of a Gaussian kernel for λ_t. In general, however, it will not be the case that $\chi_{t+1} \in K_t$, which precisely is why DA's algorithm cannot be generalized as such to a broader class of samplers.

The omission of g_t° implies that DA algorithm is not as efficient as ours in its search for an optimized sampler (for the very same reason as OLS estimates are inefficient relative to GLS estimates in the presence of heteroskedasticity) which is why it requires several rounds of computation. In contrast our algorithm produces an efficient MC sampler in a single round of computation (though a second round might occasionally prove useful when the initial sampler p_0 is particularly inefficient and N is very small).

5 Numerical and statistical standard deviations

There are a broad range of classical and Bayesian techniques which require (numerical) evaluation of the

likelihood function. In the present paper we restrict our attention to Maximum Likelihood (ML) estimation. A simulated ML estimator is one which maximizes the (log) likelihood function, as defined in equation (29). We shall draw a clear distinction between the actual (unfeasible) ML estimator $\hat{\theta}(Y_T)$ and its simulated counterpart $\hat{\theta}_S(Y_T)$, respectively defined as

$$\hat{\theta}(Y_T) = \underset{\theta \in \Theta}{\text{Arg Max}}\ L(\theta; Y_T) \tag{75}$$

$$\hat{\theta}_S(Y_T) = \underset{\theta \in \Theta}{\text{Arg Max}}\ \bar{L}_S^*(\theta; Y_T) \tag{76}$$

where $\bar{L}_S^*$ has been defined in equation (29).

It is common practice in the literature to treat $\hat{\theta}_S(Y_T)$ as an estimate of θ itself. As discussed in Richard (1995), such practice confuses the issue of assessing the statistical properties of $\hat{\theta}(Y_T)$ as an estimator of θ with that of evaluating the numerical accuracy of $\hat{\theta}_S(Y_T)$ as an estimate of $\hat{\theta}(Y_T)$.

Therefore, in the application which follows (as well as in future applications of our techniques) we propose to compute two distinct standard deviations for simulated ML estimators: (1) an MC standard deviation, whereby $\hat{\theta}_S(Y_T)$ is treated as a function of the auxiliary MC samples and Y_T is kept fixed and; (2) an estimate of the statistical standard deviation of $\hat{\theta}(Y_T)$ obtained by treating $\hat{\theta}_S(Y_T)$ as a function of Y_T under a fixed set of common random

numbers.

Asymptotic formulae for these two sets of standard deviations can be found in RZ (1995). However, their evaluation requires a fair amount of (additional) programming work. We find it easier (and often more relevant) to compute finite sample standard deviations based upon additional auxiliary MC simulations.

MC standard deviations are obtained by keeping Y_T fixed and rerunning our entire algorithm (including the search for an efficient sampler) under different sets of random draws for Λ_T accounting, thereby, for the overall MC uncertainty induced by our procedure.

"statistical standard deviations" (MC estimates of the SE) of $\hat{\theta}(Y_T)$ are obtained by generating auxiliary Y_T samples from the model itself and each time rerunning the entire algorithm under a common set of random numbers for the Λ's.[14]

As we shall illustrate below, such auxiliary simulations

---

[14] Specifically we compute a different 'efficient' sampler for each auxiliary Y_T. The corresponding sets of $\tilde{\Lambda}_{tc}$ are all obtained by inversion of a common set of random numbers. We find this procedure to be preferable (by far) to one whereby a single set of $\tilde{\Lambda}_{tc}$'s would be used for all Y_T's. Our procedure secures maximal numerical efficiency at all stages of the auxiliary MC simulations (As we shall illustrate below, statistical standard deviations far exceeds MC standard deviations).

are far less computer intensive than one might expect, in light of the high efficiency of our algorithm. More importantly, they only require minor adjustements to the baseline algorithm.

6. Numerical illustration

In order to evaluate the performance of our algorithm we use it to compute simulated ML estimates for the parameters of the stochastic volatility model defined by equations (50) and (51), using a fairly large data set ($T = 1,447$) taken from Duffie and Singleton (1989) and consisting of IBM daily stock price changes for the period from 1/9/82 to 8/31/87

All calculations are based upon 10 MC replications only ($N = S = 10$) which, as we shall see below, suffices to produce accurate MC estimates under 'efficient' sampling at least.

Each 'run of computation' which is reported in tables 1 and 2 consists of the search for a simulated ML estimate and includes, therefore, as many likelihood evaluations as necessitated for convergence. Each run is based upon a single $10 \times 1,447$ matrix $\tilde{U}$ of uniform random draws from which inverted gamma CRN's are obtained by 'inversion'. Calculation of MC standard deviations requires using a different $\tilde{U}$ for each run.

In table 1, we report ML estimates obtained by maximization of the simulated likelihood given by equation (9), together with MC standard deviations (statistical standard deviations were not computed as the reported results are useless anyway). These results confirm the frequent impracticability of 'natural' importance sampling evaluation of likelihood functions, as documented in the literature - see e.g McFadden (1989). Moreover, the regression coefficient of λ_t on λ_{t-1}, which equals $\tau.(\nu-2)^{-1}$, is dramatically as well as significantly downward biased (as seen by comparison with the more accurate results in table 2 or with results found in the literature for similar models). Each run of computation in table 1 requires of the order of 2 hours of CPU on a UNIX DEC 5000/240 workstation, resulting for a large part from slow convergence (or partial lack thereof) caused by inaccurate numerical estimates of the likelihood function.

In table 2, we report ML estimates obtained from an 'efficient' sampler obtained by application of the algorithm described in section 4.1 (The results are essentially the same whether we draw the 'initial' sample $\{\tilde{\lambda}^0_{ti}\}$ from p^0_t or from $\tilde{p}^0_t$, as defined in equations (54) and (61), respectively) Note that, in spite of the small number of replications ($N=S=10$), the simulated ML estimates are numerically quite accurate (particularly those of ν, τ and $\tau.(\nu-2)^{-1}$ which are the key coefficients of our model). We could easily achieve greater accuracy by increasing S We did not do so since statistical

TABLE II A

Maximum Likelihood Estimates with RZ-acceleration
Monte Carlo Replications N=10

	gamma	tau	nu	loglik.
1	0.000722	42.4865	48.6306	4241.01
2	0.000767	42.3839	48.8696	4241.62
3	0.000920	42.2840	49.7925	4241.31
4	0.000954	42.3479	49.9800	4239.89
5	0.000868	42.3597	49.4187	4240.58
6	0.000806	42.3741	48.9732	4242.32
7	0.000819	42.3691	49.4016	4246.90
8	0.000990	41.1223	48.9770	4242.32
9	0.000962	42.4506	49.9309	4244.24
10	0.000931	42.4504	49.9848	4238.76
11	0.000887	42.3376	49.9124	4243.60
12	0.000722	42.1088	47.9635	4241.40
13	0.000810	42.4976	49.3935	4244.10
14	0.001066	42.3289	50.6304	4240.39
15	0.000548	42.4080	47.4255	4242.11
16	0.001138	42.2803	50.9828	4239.55
17	0.000565	42.4864	47.6824	4245.13
18	0.000736	41.9340	48.7493	4241.10
19	0.000685	42.0434	48.3735	4239.57
20	0.000814	42.1450	49.4697	4240.58
mean	0.000835	42.2599	49.2271	4241.82
MC s.d.	0.000148	0.30058	0.90628	2.01971

St. s d

TABLE IA

Maximum Likelihood Estimates without Acceleration
Monte Carlo Replications N=10

	gamma	tau	nu	loglik.
1	0.011410	-0.0001	66.8655	4204.56
2	0.007142	0.1658	42.2482	4204.24
3	0.007056	0.7178	42.2537	4205.89
4	0.004975	3.8307	33.5841	4207.78
5	0.007033	1.7653	44.0172	4207.08
6	0.005206	3.3973	34.0815	4200.24
7	0.011482	-0.0001	67.0000	4207.93
8	0.005621	-0.0001	33.5000	4204.77
9	0.010521	5.5777	67.0000	4208.30
10	0.007225	-0.0001	42.5622	4204.69
mean	0.007767	1.5454	47.3112	4205.55
MC s.d.	0.002353	1.92621	13.40333	2.30019

TABLE II B
The ML Estimates with RZ-acceleration
MC Replications N=10

	gamma/(nu-2)	tau/(nu-2)	loglik.
1	0.0000155	0.91113	4241.01
2	0.0000164	0.90429	4241.62
3	0.0000192	0.88474	4241.31
4	0.0000199	0.88262	4239.89
5	0.0000183	0.89331	4240.58
6	0.0000172	0.90209	4242.32
7	0.0000173	0.89383	4246.90
8	0.0000211	0.87537	4242.32
9	0.0000201	0.88566	4244.24
10	0.0000194	0.88466	4238.76
11	0.0000185	0.88365	4243.60
12	0.0000157	0.91614	4241.40
13	0.0000171	0.89670	4244.10
14	0.0000219	0.87042	4240.39
15	0.0000121	0.93357	4242.11
16	0.0000232	0.86317	4239.55
17	0.0000124	0.93004	4245.13
18	0.0000157	0.89700	4241.10
19	0.0000148	0.90663	4239.57
20	0.0000171	0.88783	4240.58
mean	0.0000176	0.89514	4241.82
MC s.d.	0.0000028	0.01781	2.02045

St. s d

TABLE IB
The ML Estimates without Acceleration
MC Replications N=10

	gamma/(nu-2)	tau/(nu-2)	loglik.
1	0.0001759	0.00000	4204.56
2	0.0001774	0.00412	4204.24
3	0.0001753	0.01783	4205.89
4	0.0001575	0.12129	4207.78
5	0.0001674	0.04201	4207.08
6	0.0001623	0.10590	4200.24
7	0.0001766	0.00000	4207.93
8	0.0001784	0.00000	4204.77
9	0.0001619	0.08581	4208.30
10	0.0001781	0.00000	4204.69
mean	0.0001711	0.03769	4205.55
MC s.d.	0.0000076	0.04602	2.29993

$$E(\lambda_t | \lambda_{t-1}) = \frac{\gamma}{\nu-2} + \frac{\tau}{\nu-2}\lambda_{t-1}$$

standard deviations already are 4 to 5 times as large as the standard deviations.

In addition to the fact that the results in table 2 are far more accurate than those in table 1, their evaluation only requires of the order of 23' of CPU time for each run, as opposed to 120' for unaccelerated runs. This five-fold reduction in computing time, which obtains in spite of the search time for an efficient sampler, is explained by the fact that the ML algorithm converges (much) faster as the individual likelihood estimates get increasingly accurate. It also implies that, within a given computing budget, the increase in numerical accuracy produced by the use of an efficient MC sampler is even greater than reported in tables 1 and 2.

Similar results are reported in DR (1993) and RZ (1995, 1996) which fully support the high efficiency of our new algorithm.

7. Conclusion

The results presented in this paper confirm fully similar findings in DR (1993) and RZ (1995). They unequivocally indicate that, contrary to current wisdom, accurate MC estimates of the likelihood functions of DLV models can be obtained with very small numbers of replications (as few as 10) provided one uses efficient samplers, such as the ones produced by our algorithm.

The methods we propose offers three key advantages.

(1) Programming cost is minimum, as it essentially consists of the implementation of a sequence of simple weighted least squares problems which are applied to artificial data generated from an 'initial' sampler (typically the latent process itself)

(2) Relatedly, the algorithm is highly generic. Its base structure does not depend on specific classes of samplers (beyond the requirement that the class under consideration be closed under multiplication which, as we discussed, contributes to the simplicity of our method). Changes in the statistical formulation of the model under scrutiny can easily be accommodated as they only require minor programming adjustments (For example, in RZ(1995), we reestimate the stochastic volatility model presented in section 4.1 under different distributional assumptions at the cost of minor modifications of our FORTRAN code);

(3) The additional computing cost required for the evaluation of an efficient sampler is small and, moreover, is more than compensated by the large efficiency gains it produces. In the context of simulated ML estimation, the net result is a substantial reduction in overal computing time (for a preassigned number of replications).

In short, our new acceleration procedure paves the way for routine application of a broad range of (classical and Bayesian) likelihood-based inference techniques to DLV models

even when sample size is (very) large. It does not render alternative estimation techniques such as the Method of Simulated Moments obsolete but it does imply that the choice of an inference procedure no longer is dominated by computational considerations and can be based instead on more fundamental statistical ones (robustness, statistical efficiency, ...).

References

Forthcoming.