

# EmmPack 1.01: C/C++ code for use with Ox for estimation of univariate stochastic volatility models with the efficient method of moments

Pieter J. van der Sluis\*

Department of Actuarial Science and Econometrics

University of Amsterdam

and

Tinbergen Institute

February 12, 1998

## Abstract

Econometric estimation using simulation techniques, such as the efficient method of moments, may be time consuming. The use of ordinary matrix programming languages such as Gauss, Matlab, Ox or S-plus will very often cause extra delay. For the Efficient Method of Moments implemented to estimate stochastic volatility models this will surely be the case. Therefore the author made a C/C++ library containing the bulk of the procedures needed in the implementation of the efficient method of moments technique for a broad range of univariate stochastic volatility models. As a side effect of the Efficient Method of Moments, EGARCH models with a variety of nonnormal distributions can be estimated with this package. Implementations have been made for the Intel Pentium platform under Windows and for the IBM RS/6000 platform under AIX. The library is dynamically linked to Ox under Windows and statically under AIX. The speed improvements are considerable compared with pure Ox code. The paper serves as a manual for this library. It describes the efficient method of moments for this specific case of stochastic volatility models. It describes the program. Some examples are given from other work of the author. Technicalities are given in the appendices

*Keywords:* Computing; Efficient Method of Moments; EGARCH; Financial Time-series; SemiNonParametric Methods; Statistical Software; Stochastic Volatility models

*JEL-classification:* C13-15; C22; C87

---

\*Address for correspondence: Pieter J. van der Sluis, University of Amsterdam, Department of Actuarial Science and Econometrics, Roetersstraat 11, NL-1018 WB Amsterdam, The Netherlands, phone +31 20 525 4257, fax +31 20 525 4349, email [Sluis@fee.uva.nl](mailto:Sluis@fee.uva.nl), homepage [http://www.fee.uva.nl/vak\\_groep/ake/vdsluis.html](http://www.fee.uva.nl/vak_groep/ake/vdsluis.html). The author thanks Peter Boswijk, Jurgen Doornik, Ruud Koning, Neil Shephard and an anonymous referee for comments. This paper was presented at The Third International Conference on Computing in Economics and Finance, Stanford University, Stanford (CA.), June 30 - July 2, 1997. This manuscript is prepared for *Studies in Nonlinear Dynamics and Econometrics* 2.3.

# 1 Introduction

Even with the current state of computing power the estimation of *stochastic volatility models* (see Ghysels *et al.* (1996) and section 2 below) using simulation techniques such as the *efficient method of moments* (EMM, see Gallant and Tauchen (1996) and section 3 below) may still be very time-consuming. This is amplified by the fact that many econometricians use for their applied work *matrix oriented programming languages*, such as GAUSS, MATLAB, S-PLUS or more recently Ox (Doornik (1996)). See e.g. Cribari-Neto (1996) for a comparison of GAUSS, Ox and S-PLUS. As long as the estimation involves mainly matrix manipulations without any substantial *for-next* or *do-while* loops, which is the case for linear models such as *regression*, AR and VAR models, there is no loss of computing power. There is also no loss in computing power in case the developers of the matrix language have provided the researcher with a *hard coded* built-in procedure for doing a certain time consuming job. However, there is a big loss of computing time as soon as the solution to a certain problem involves *loops*, such as the Kalman filter, where a system of *Riccati equations* is evaluated recursively, or the EMM estimation technique that we are dealing with in this paper, where the score of a specific complex *auxiliary model* is evaluated under the *structural model*, in order to calibrate the parameters of the structural model. With EMM with a well specified auxiliary model, one can obtain *maximum likelihood efficiency*<sup>1</sup> of the parameters for models where direct maximum likelihood is not feasible.

One faces a substantial loss of computational speed for two reasons: (i) A matrix oriented language is not good at doing loops. (ii) Kalman filter techniques or EMM are not yet standard procedures of these matrix oriented programming languages. One solution may be to hard code the *whole* program in C, Fortran, Modula 2 or Pascal. This has the drawback that the programmer has the continuous feeling that he or she is reinventing the wheel. The econometrician has to do many things that do not belong to his or her core competence, therefore from an economic perspective the marginal benefits of less computing costs may not be on equal footing with the marginal losses in human capital<sup>2</sup>. One possible solution to this is to *link* hard coded procedures that contain the loops, to the matrix programming language, combining the facilities of the matrix programming language with the speed of a (relatively) low level programming language. Under Ox this link can be implemented *dynamically* using dynamic link libraries (DLL's) on Windows based platforms (Windows 3.xx, Windows 95 and Windows NT) or *statically* on mainframe computers under AIX, UNIX, SUN and many others<sup>3</sup>. These libraries basically extend the matrix language with some procedures. For Ox this has been done for the Kalman filter by Koopman *et al.* (1997, SffPack) and for several Bayesian procedures to estimate stochastic volatility models by Shephard (1996a, SvPack), among others. Here time consuming procedures were hard coded in C/C++ and linked to the matrix programming language Ox. Exactly this is done here for the EMM estimation technique for a wide class of stochastic volatility models using dynamic link libraries for Windows and statically on the

---

<sup>1</sup>It should be noted that very often maximum likelihood delivers not only first-order asymptotic efficiency but also second-order asymptotic efficiency. In this paper we will limit ourselves to first-order asymptotic efficiency and the regular case: an estimator that has maximum likelihood efficiency means that this estimator has the same variance-covariance matrix of the asymptotic normal distribution of the maximum likelihood estimator. For EMM there are no results yet regarding higher-order efficiency.

<sup>2</sup>An example where C/C++ code is provided in this context is Danielsson (1996a) where C/C++ source code for the estimation of a basic stochastic volatility model is given.

<sup>3</sup>For an extensive list of different platforms for which Ox is available, see section 4. Since the list is still growing, for an up to date list see

<http://www.nuff.ox.ac.uk/Users/Doornik>

RS/6000 under AIX<sup>4</sup>.

The program is called `EmmPack`. Unlike its name it does not only provide code to estimate stochastic volatility models via EMM. As a consequence of the EMM methodology one can also use the package to estimate by maximum likelihood the auxiliary *SemiNonParametric* models, for which, in the case of stochastic volatility models, we have specified an EGARCH *leading term*. This means that this program can also be used to quickly estimate EGARCH models with disturbance terms that follow a variety of non-normal distributions.

The plan of the paper will be as follows. In section 2 a cursory introduction to stochastic volatility models is provided. In section 3 a cursory introduction to EMM is provided. These two sections are merely provided to introduce some notation and nomenclature. The reader who is unfamiliar with any of these subjects, is advised to consult the above mentioned references and the references mentioned in these sections. Section 4 presents some of the key features of `Ox`. Here we also explain how to get the program installed and working. Section 5 provides some empirical examples taken from van der Sluis (1997a and 1997b). Section 6 concludes. In the appendices details about the actual implementation are provided.

## 2 Stochastic Volatility Model

Stochastic volatility models have become quite popular in the econometrics and finance literature. Key references on estimation of stochastic volatility models include: Harvey *et al.* (1994), Harvey and Shephard (1996), Fridman and Harris (1997) and Sandmann and Koopman (1997) for Kalman filter techniques<sup>5</sup>. Jacquier *et al.* (1994), Schotman and Mahieu (1994), Kim *et al.* (1996) on Bayesian methods<sup>6</sup>, Danielsson (1994) and Danielsson and Richard (1993) on Simulated Maximum Likelihood methods<sup>7</sup>. Finally, Gallant and Tauchen (1996) and Gallant *et al.* (1994) are the main references for EMM methods<sup>8</sup>. These have been the most successful techniques. We also mention Andersen and Sørensen (1996) for GMM techniques and Monfardini (1996) for indirect inference technique à la Gouriéroux *et al.* (1993). For foundations of stochastic volatility models see Clark (1973), Tauchen and Pitts (1983), Taylor (1986) and Hull and White (1987). Review articles have been provided by Ghysels *et al.* (1996) and Shephard (1996b).<sup>9</sup>

A stochastic volatility model in its basic discrete time format reads

$$\begin{aligned}y_t &= \sigma_t \epsilon_t \\ \ln \sigma_t^2 &= \omega + \gamma \ln \sigma_{t-1}^2 + \sigma_\eta \eta_t \\ \epsilon_t, \eta_t &\sim IN(0, 1), t = 1, \dots, n\end{aligned}\tag{1}$$

This model has served as the benchmark and starting point of the bulk of the econometric literature on stochastic volatility models. Note that this model is in discrete form, `EmmPack` is designed to deal with discrete models. However one may modify some of the `ox`-code in `emm.ox` and make it also applicable to continuous-time models. The reason why model (1) cannot be estimated by standard maximum likelihood lies in the fact that

---

<sup>4</sup>This code is available from the author on request. For other platforms contact the author.

<sup>5</sup>On Kalman filter techniques in this context we also mention Ruiz (1994).

<sup>6</sup>On Bayesian methods in this context we also mention Shephard (1996a)'s `SvPack`.

<sup>7</sup>See also Danielsson (1996a,b), Richard and Zhang(1995a,b) for more on SML methods in this context. Danielsson (1996a) actually contains the source code.

<sup>8</sup>Other references on EMM in this context are Andersen and Lund (1996), Andersen and Lund (1997), Gallant and Long (1997) and Van der Sluis (1997a,b,c) and Andersen *et al.* (1997)

<sup>9</sup>See also Taylor (1994), Andersen (1994) and Andersen (1992).

the  $\sigma_t$  are latent or unobserved variables which have to be integrated out of the likelihood. This is typically intractable both numerical and analytical since for every observation there is such a  $\sigma_t$ . Standard Kalman filter techniques cannot be applied since the latent process is non-Gaussian and the resulting state-space form does not have a conjugate filter.

Many variations on (1) are possible. In van der Sluis (1997a and b) a rather broad class of models has been proposed namely

$$\begin{aligned}
 y_t &= \sigma_t \epsilon_t \\
 \ln \sigma_t^2 &= \omega + \sum_{i=1}^p \gamma_i L^i \ln \sigma_t^2 + \sigma_\eta (1 + \sum_{j=1}^q \zeta_j L^j) \eta_t \\
 \begin{bmatrix} \epsilon_t \\ \eta_{t+1} \end{bmatrix} &\sim NIID(0, \begin{bmatrix} 1 & \lambda \\ \lambda & 1 \end{bmatrix}), -1 \leq \lambda \leq 1, t = 1, \dots, n
 \end{aligned} \tag{2}$$

In the program this model is referred to as the **ASARMAV**( $p, q$ ) model<sup>10</sup>. The asymmetric component arising from the correlation parameter  $\lambda$  has also been considered in Harvey and Shephard (1996). The inclusion of MA parameters is a novelty<sup>11</sup>. Exactly model (2) is the model the author claims that can be estimated by the package. Also estimation of several other models with **EmmPack** is discussed in van der Sluis (1997c). In these models the  $\epsilon_t$  follow a wide variety of non-Gaussian models

The following model have been supported by the package: **SARMAV**(1, 0), **SARMAV**(2, 0), **SARMAV**(3, 0), **SARMAV**(1, 1), **SARMAV**(1, 2), **ASARMAV**(1,0), **ASARMAV**(2, 0), **ASARMAV**(3, 0), **ASARMAV**(1, 1) and **ASARMAV**(1, 2). The user can built his own procedures following the structure given in section 4.

In a next version of this program conditional mean terms like  $y_t = \mu_t + \sigma_t \epsilon_t$  can be dealt with. This may be interesting for modeling the term structure of interest rates, see Andersen and Lund (1996,1997). For now the user should prewhiten his data so in order to take care of a time dependent mean. Multi-variate extensions of this model (2) are currently investigated in van der Sluis (1997d) and in Jiang and van der Sluis (1997), where the paper van der Sluis (1997d) will be accompanied by the program **MemPack**, which is a multivariate extension of **EmmPack**.

### 3 Efficient Method of Moments

Gallant and Tauchen (1996)<sup>12</sup> solve the efficiency problems moment-based techniques generally have by proposing the *efficient method of moments* (EMM) technique. The structural model is estimated by using an *auxiliary model*. The connection between the auxiliary model and the structural model is achieved by means of the scores of the auxiliary model (*score calibration*), where strict guidelines are given for the choice of the auxiliary model such that maximum likelihood efficiency is attained. For the program we are only concerned with case 2 in Gallant and Tauchen (1996). In short the EMM method goes as follows: the sequence of densities for the structural model will be denoted

$$\{p_1(x_1 | \theta), \{p_t(y_t | x_t, \theta)\}_{t=1}^\infty\}$$

<sup>10</sup> Asymmetric Stochastic AutoRegressive Moving Average Volatility

<sup>11</sup> It is a novelty in the sense that this model can actually be estimated. In Harvey *et al.* (1994) the ARMA specification was already suggested.

<sup>12</sup> See also Tauchen (1997)

The sequence of densities for the auxiliary process will be denoted as

$$\{f_1(w_1 | \beta), \{f_t(y_t | w_t, \beta)\}_{t=1}^{\infty}\}$$

where  $x_t$  and  $w_t$  are observable endogenous variables. In particular the  $x_t$  will be a vector of lagged  $y_t$  and the  $w_t$  will also be a vector of lagged  $y_t$ . The lag-length may differ therefore a different symbol is chosen. We impose assumptions 1 and 2 in Gallant and Long (1997) on the structural model, these are technical assumptions that imply standard properties of quasi maximum likelihood estimators and properties of estimators based on *Hermite expansions* which will be explained below. Important is that the structural model is stationary and ergodic. The reader is referred to the original papers for details. Define

$$m(\theta, \beta) := \int \int \frac{\partial}{\partial \beta} \ln f(y | w, \beta) p(y | x, \theta) dy p(x | \theta) dx$$

the expected score of the auxiliary model under the dynamic model. The expectation is written in integral form to anticipate on the fact this integral is approximated by standard Monte Carlo techniques:

$$m_N(\theta, \beta) := \frac{1}{N} \sum_{\tau=1}^N \frac{\partial}{\partial \beta} \ln f(y_{\tau}(\theta) | w_{\tau}(\theta), \beta)$$

where  $y_{\tau}(\theta)$  are drawings from the structural model. Let  $n$  denote the sample size, the EMM estimator is defined as

$$\hat{\theta}_n(\mathcal{I}_n) := \arg \min_{\theta \in \Theta} m'_N(\theta, \hat{\beta}_n)(\mathcal{I}_n)^{-1} m_N(\theta, \hat{\beta}_n)$$

where  $\mathcal{I}_n$  is a weighting matrix and  $\hat{\beta}_n$  denotes an estimator for the parameter of the auxiliary model. The optimal weighting matrix here is obviously

$$\mathcal{I}_0 = \lim_{n \rightarrow \infty} V_0 \left[ \frac{1}{\sqrt{n}} \sum_{t=1}^n \left\{ \frac{\partial}{\partial \beta} \ln f_t(y_t | w_t, \beta^*) \right\} \right]$$

where  $\beta^*$  is a (pseudo) true value. The small sample pendant is

$$\mathcal{I}_n = V_0 \left[ \frac{1}{\sqrt{n}} \sum_{t=1}^n \left\{ \frac{\partial}{\partial \beta} \ln f_t(\tilde{y}_t | \tilde{w}_t, \hat{\beta}_n) \right\} \right]$$

In the program  $\mathcal{I}_n$  is estimated using the outer product gradient.

The main result is consistency and asymptotic normality of the EMM estimator of the structural parameters  $\hat{\theta}_n$  follows:

$$\sqrt{n}(\hat{\theta}_n(\mathcal{I}_0) - \theta_0) \xrightarrow{d} N(0, [\mathcal{M}'_0(\mathcal{I}_0)^{-1} \mathcal{M}_0]^{-1})$$

where  $\mathcal{M}_0 := \frac{\partial}{\partial \theta} m(\theta_0, \beta^*)$ .

In order to obtain *maximum likelihood efficiency* it is required that the auxiliary model in some sense embeds the structural model. The semi-nonparametric (SNP) density of Gallant and Nycká (1987)<sup>13</sup> may be a good choice, see Gallant and Tauchen (1996) and Gallant and Long (1997). The auxiliary model is built as follows. The process  $y_t(\theta_0)$  is the process under investigation,  $\mu_t(\beta^*) := E_{t-1}[y_t(\theta_0)]$ , is the conditional mean of the auxiliary model,  $\sigma_t^2(\beta^*) := \text{Var}_{t-1}[y_t(\theta_0) - \mu_t(\beta^*)]$  is the conditional variance and  $z_t(\beta^*) := [y_t(\theta_0) - \mu_t(\beta^*)] \cdot [\sigma_t^2(\beta^*)]^{-1/2}$  is the standardized process. The SNP density now takes the following form

$$f(y_t; \beta) = \frac{1}{\sqrt{\sigma_t^2}} \frac{[P_K(z_t, x_t)]^2 \phi(z_t)}{\int [P_K(u, x_t)]^2 \phi(u) du} \quad (3)$$

<sup>13</sup>Building on earlier work of Phillips (1983). See also Fenton & Gallant (1996a, b) for recent results on SNP densities.

where  $\phi$  denotes the standard normal density,  $x := (y_{t-1}, \dots, y_{t-L})$  and the polynomials

$$P_K(z, x_t) := \sum_{i=0}^{K_z} a_i(x_t) z^i := \sum_{i=0}^{K_z} \left[ \sum_{j=0}^{K_x} a_{ij} x_t^j \right] z^i$$

A specific form for the polynomials is taken, namely orthogonal Hermite polynomials (see Gallant *et al.* (1991) and Andersen and Lund (1997)). Relevant formulae for the derivatives can be found in Abramowitz and Stegun (1972) and Fenton and Gallant (1996a) and in appendix A below. The model  $\sigma_t^2(\beta)$  and  $\mu_t(\beta)$  is chosen as a *leading term* in the Hermite expansion to relieve the expansion of some of its task, improving its small sample properties. This version of `EmmPack` contains no models for  $\mu_t(\beta)$  however one may modify the `Ox` source code in order to include processes for the mean. The easiest way is to filter out a time dependent mean and to give this prewhitened data as input to the program. For  $\sigma_t^2(\beta)$  an EGARCH( $p, q$ ) model was used see appendix A for the relevant formulas.

In this paper we will take  $p := \dim(\theta)$ ,  $q := \dim(\beta)$ . The number of moment conditions  $q$  may be determined using several criteria. For EMM, it is necessary that  $q$  increases with  $n$ . Note in this respect the conceptual difference with GMM. It will automatically happen that  $q$  increases with  $n$  using any of the model specification criteria such as the Akaike Information Criterion (AIC, Akaike (1973)), the Schwarz Criterion (BIC, Schwarz (1978)) or the Hannan-Quinn Criterion (HQC, Hannan and Quinn (1979) and Quinn (1980)). The theory of model selection in the context of SNP models is not very well developed yet. Results in Eastwood (1991) may lead to believe AIC is optimal in this case. However, as for multivariate ARMA models, the AIC may overfit the model to noise in the data so we may be better off by following the BIC or HQC. The same findings were reported in Andersen and Lund (1997). In their paper Gallant and Tauchen (1996) rely on the BIC in their applications. Recent Monte Carlo results in Andersen *et al.* (1997) and van der Sluis (1997c) show that for pure (E)GARCH (so  $K_x = K_z = 0$ ) models “E”MM provides rather efficient estimates for the Gaussian SARMAV(1,0) and ASARMAV(1,0) models. The small sample properties of EMM with this class of non-Gaussian auxiliary models is unknown. For SV models with an nonnormal error structure it is likely that an high order Hermite polynomial is needed. As argued in van der Sluis (1997c) Monte Carlo experiments with high order Hermite polynomials are thought to be computationally infeasible at the current state of computer technology. In figure 3 the progressive nature of the computing times with higher order Hermite polynomials is displayed. For more on the the choice of an auxiliary model the interested reader is referred to Andersen *et al.* (1997) and van der Sluis (1997a,b and c). The procedure `diagnos` returns a summary of the auxiliary estimations. Among the parameters itself, standard errors and  $t$ -values, the following model selection criteria for maximization are provided AIC (Akaike (1973)), BIC (Schwarz (1978)) and HQC (Hannan-Quinn (1979) and Quinn (1980)). These criteria may be used to determine the order of the auxiliary model<sup>14</sup>.

In short, the following algorithm was used

- (i) Determine the order of the EGARCH process. For this particular EGARCH process calculate  $\hat{\beta}$ . Set the corresponding weighting matrix  $\hat{T}_n$  equal to the outer product of the scores, i.e.  $\frac{1}{n} \sum_{t=1}^n \left[ \frac{\partial}{\partial \beta} \ln f_t(y_t | w_t, \hat{\beta}_n) \right] \left[ \frac{\partial}{\partial \beta} \ln f_t(y_t | w_t, \hat{\beta}_n) \right]'$
- (ii) Determine a value for  $\theta : \hat{\theta}$
- (iii) Simulate  $y_1^N(\hat{\theta})$  and  $y_2^N(\hat{\theta})$  using *antithetic variables* (see appendix C)

<sup>14</sup>For some properties in the context of multivariate time-series see Lütkepohl (1990).

(iv) Calculate  $m_N(\hat{\theta}, \hat{\beta}_n) = \frac{1}{2N} \sum_{t=1}^N [\frac{\partial}{\partial \beta} \ln f_t(y_1^N(\hat{\theta})|w_t, \hat{\beta}_n) + \frac{\partial}{\partial \beta} \ln f_t(y_2^N(\hat{\theta})|w_t, \hat{\beta}_n)]$

(v) Repeat (ii) till (iv) until the quantity  $m'_N(\hat{\theta}, \hat{\beta}_n)(\hat{\mathcal{I}}_n)^{-1}m_N(\hat{\theta}, \hat{\beta}_n)$  is minimised

We set  $N = 50,000$ <sup>15</sup> and as explained in van der Sluis (1997b) for the stochastic volatility models considered here no starting values have to be used<sup>16</sup>.

We end this section by stating the  $J$ -test for overidentifying restrictions for EMM (see Gallant, Hsieh and Tauchen (1994)). Under the null that the structural model is true one may deduce:

$$n \cdot m'_N(\hat{\theta}_n, \hat{\beta}_n)(\hat{\mathcal{I}}_n)^{-1}m_N(\hat{\theta}_n, \hat{\beta}_n) \xrightarrow{d} \chi^2_{|\beta|-|\theta|}$$

and the direction of the misspecification may be indicated by the quasi-t ratios

$$\begin{aligned} \hat{T}_n &:= \hat{S}_n^{-1} \sqrt{n} m_N(\hat{\theta}_n, \hat{\beta}_n) \\ \hat{S}_n &:= [\text{diag}(\hat{\mathcal{I}}_n - \hat{\mathcal{M}}_n'(\hat{\mathcal{M}}_n' \hat{\mathcal{I}}_n^{-1} \hat{\mathcal{M}}_n)^{-1} \hat{\mathcal{M}}_n)]^{1/2} \end{aligned}$$

Here  $\hat{T}_n$  is distributed as  $t_{|\beta|-|\theta|}$ . These statistics are provided by EmmPack.

## 4 Description of EmmPack and Ox

The Ox matrix programming language is new to the market of econometric software. It is by far the fastest matrix oriented programming language see Cribari-Neto (1997). Several other advantages are: it is a very open language, its syntax is very similar to C/C++ and there are currently versions for DOS, Windows 3.xx, Windows 95, Windows NT, AIX, SunOs, Solaris, HP-UX, Irix, Linux and Unix. Most of these versions are free. There are no differences between these versions, except that only in the Windows version one can use Ox's sister program *Givewin* which provides several graphic capabilities. Another small difference is that for some versions one cannot dynamically link C/C++ code to Ox, for these platforms one has to resort to static linking which is not more difficult but less elegant. One may also try to link the library to the more common programs as GAUSS or Matlab. One should probably have to write a DLL which translates exported C/C++ functions from the Ox DLL to C/C++ functions that can be imported from GAUSS or Matlab. How the C/C++ functions are exported from an Ox DLL can be found in Doornik (1996) pp 286-328. It should be mentioned that the DLL in EmmPack uses mathematical C/C++ function from Ox, so one should own a copy Ox. Therefore one may better save the trouble and directly use Ox.

The DLL is optimized for the Intel Pentium. For the RS/6000 the author has used the IBM compiler which should generate very efficient code on the RS/6000. In spite of this it is worth mentioning that the gain in speed

<sup>15</sup>Two antithetic series are used, each of size 50,000: Experimentation has shown that for the score-generator used here virtually no Monte Carlo variance is presented for simulated series of this size. With virtually no Monte Carlo variance the author means that the estimates are the same for four digits when different simulated series are used. Smaller sizes, say 5,000 could be used for getting fast estimates.

<sup>16</sup>The model is started up in

$$\begin{aligned} y_0 &= \epsilon_0 \sigma_0, \\ \ln \sigma_0^2 &\sim N(\omega / (1 - \sum_{i=1}^p \rho_i), \sigma_\eta^2 (1 + \sum_{j=1}^q \zeta_j^2) (1 - \sum_{i=1}^p \rho_i^2)^{-1}) \\ \epsilon_0 &\sim N(0, 1), \text{Corr}(\epsilon_0, \ln \sigma_0^2) = \lambda / \sqrt{(1 + \sum_{j=1}^q \zeta_j^2) (1 - \sum_{i=1}^p \rho_i^2)^{-1}} \end{aligned}$$

on both the Intel and the RS/6000 was immense, with the highest relative gain on the Intel. On a P5-166 with 32 MB under Win95 it took in pure Ox 1.20a code 117.7 seconds to evaluate an EGARCH(1,1)-H(5,0) for  $2 \times 5,000$  antithetic variables. With the C/C++ code in a DLL it took only 9.5 seconds under the same configuration: a twelve-fold improvement. Whereas this gain in speed is considerable this does not mean that one should always program in C/C++ or any other low level programming language. Only for the types of problems that we are dealing with in this particular problem, namely a loop with allocations, hard-coding gives an enormous speed improvements.

EmmPack can be downloaded from

[http://www.fee.uva.nl/vak\\_groep/AKE/vdsluis.htm](http://www.fee.uva.nl/vak_groep/AKE/vdsluis.htm)

The files are *zipped* as emmpack.zip. One can *unzip* them with a utility such as *pkunzip*. This file contains the following: s&p500.mat, xr.mat, emm.dll, sv\_model.oxo, sv\_model.h, max1sid.oxo, max1sid.h, emm.h, and the central program, emm.ox, containing modifiable Ox source code.

Unless you are an *expert*, it is wise to create a directory `\ox\packages\emm\` and move the emm.ox file to this directory. Next, place the files sv\_models.oxo, sv\_models.h, max1sid.oxo and max1sid.h to the directory `\ox\include\` and move the files emm.dll to the directory `\ox\bin\`. Additional description of the program can be found in the source code emm.ox.

A few things to note here: emm.dll contains a dynamic link library for the Windows 3.xx, Windows 95 and Windows NT operating systems. These require Ox version 1.20a. The AIX version is available on request from the author. The sv\_models.oxo and max1sid.oxo contain compiled Ox code. The max1sid.oxo is a modification of the maximize.oxo code that is included in the official release of Ox, except that instead of two-sided derivatives only one-sided derivatives are taken in the BFGS algorithm. Although less accurate it is twice as fast, which is very important in this application. The loss of accuracy is not relevant for the problem at hand (see also section 5). The file sv\_models.oxo contains the Ox compiled code of several procedures to generate antithetic series from stochastic volatility models. There is no need to hard code these procedures, because the author extensively used hard-coded built in procedures from Ox. So there will be virtually no gain, maybe even a loss, in hard coding these procedures. These are not all the stochastic volatility models that can be estimated with this program. The user can use his own. In the module sv\_models.oxo the following procedures have been currently provided:

- sarmav10(const total,const theta)
- sarmav20(const total,const theta)
- sarmav30(const total,const theta)
- sarmav11(const total,const theta)
- sarmav12(const total,const theta)
- asarmav10(const total,const theta)
- asarmav20(const total,const theta)
- asarmav30(const total,const theta)



- `asarmav11(const total,const theta)`
- `asarmav12(const total,const theta)`

*in*: an integer `total` denoting the number of variables you want to simulate, `theta` a vector of parameters. The first element of `theta` denotes the  $\omega$  variable, the next variables denote the ARMA variables. Then comes the  $\sigma_\eta$  variable. If applicable the last element is the asymmetry variable  $\lambda$ .  
*out*: returns a `total` $\times$ 2 series of *antithetic variables*<sup>17</sup> from the specified process.

Before estimating the structural stochastic volatility model, one has to specify an auxiliary EGARCH-H model. As mentioned before, one can also use this program for estimating EGARCH-H models. As described above the auxiliary models are taken from the SNP densities with an EGARCH( $p, q$ ) leading term. In the program the integers `k_x`, `k_z`, `p` and `q` refer to the variables  $k_x$ ,  $k_z$ ,  $p$  and  $q$  respectively, as defined in appendix A. The order of input and output of the variables is follows:  $\{\alpha_0, \alpha_1, \dots, \alpha_q, \gamma_1, \dots, \gamma_p, \kappa_0, \kappa_1, \Theta\}$  where  $\Theta$  is a matrix defined as

$$\Theta = \begin{bmatrix} 1 & a_{10} & \cdots & a_{k_x 0} \\ a_{01} & \ddots & & a_{k_x, 1} \\ \vdots & & \ddots & \vdots \\ a_{0k_z} & a_{1k_z} & \cdots & a_{k_x, k_z} \end{bmatrix}$$

In the program this matrix is vectorized as  $(1, a_{01}, \dots, a_{0k_z}, a_{10}, \dots, a_{1k_z}, \dots, a_{k_x k_z})$  by the procedure `decomp`. Other procedures that are included in the source-code of `emm.ox` include:

- `likeli(const vP, const adFunc, const avScore, const amHessian)`

This function is the likelihood of the auxiliary EGARCH-H model, with *in* and *output* in the generic format as on page 114 of Doornik (1996)

- `gradproc(const vP)`

This function returns the score of the auxiliary model, the auxiliary parameter vector `vP` denotes the parameters of the auxiliary model in which the score has to be evaluated.

- `createin(const vP)`

This function returns the outer product of the scores in the auxiliary parameter vector `vP`.

- `diagnos(const l, const paras, const diags, const file_diagnos)`

The function writes several diagnostic statistics for the auxiliary model to an *open* file `file_diagnos`. Here `l` denotes the loglikelihood in the optimum, `paras` denotes the parameters in the optimum and `diags` is the outer product of the scores in the optimum. Details are also given in the next section.

<sup>17</sup>For an explanation and motivation of antithetic variables appendix C below.

- `dist(const vP, const adFunc, const avScore, const amHessian)`

This function returns the value of the minimum chi-squared criterium of the EMM estimation of the structural model. Its format is the same as on page 114 of Doornik (1996).

- `jtest(const l,const inv_l, const tot, const m_hat, const M_hat)`

The function  $J$  test returns a vector of which the first element denotes the value of the  $J$  test and the other elements the individual  $t$ -values as described in section 3. The inputs are `l`, `inv_l`, `tot`, `m_hat` and `M_hat` which are mnemonics for  $\widehat{\mathcal{I}}_n, \widehat{\mathcal{I}}_n^{-1}, n, m_N(\widehat{\theta}_n, \widehat{\beta}_n)$  and  $\mathcal{M}_n(\widehat{\theta}_n, \widehat{\beta}_n)$  respectively.

- `gradproc.theta2 (const adFunc, const sv_theta)`

This function returns the score of auxiliary model at the parameters of the structural model `sv_theta`, i.e.  $m(\theta, \widehat{\beta}_n)$ . It is of the same format as on page 125 and 126 in Doornik (1996) and therefore by using this function with the Ox procedure `NumJacobian` we may calculate the numerical derivative of  $m(\theta, \widehat{\beta}_n)$  with respect to  $\theta$ , i.e.  $\mathcal{M}_n(\theta, \widehat{\beta}_n)$ .

After the auxiliary model is fitted the global variable `z` contains the  $z_t(\beta)$  from the auxiliary model. These may be used for specification tests. More information on the procedures and the variables can be found in the source file `emm.ox`. The file `s&p500.mat` contains the S&P500 series that are analysed in van der Sluis (1997b) and the file `xr.mat` contains the exchange rate series that have been analysed in van der Sluis (1997a).

As for all nonlinear optimization problems: be aware of local optima. The author encountered some in this context. One can check this by trying at different paramete values, possibly with a lower value of  $N$ . A sensible choice may be to set  $N = 20,000$ .

## 5 Application

In this section two examples of the performance of `EmmPack 1.01` are given. In van der Sluis (1997a) the British Pound / Canadian Dollar exchange rates were investigated. In van der Sluis (1997b and c) the S&P500 was investigated. Both series are in 100\* log differences. Figure 1 shows the exchange rates. Figure 2 shows the S&P500. Salient features of these data sets are provided in table 1.

Calculations were performed on Pentiums<sup>18</sup> and on several *nodes* of the SP2 at SARA<sup>19</sup>. It was found that using anti-thetic variates there is no reason to take more than  $N = 2 * 50,000$ . In the program the variable  $N$  is called `tau`. The outcomes are stable in four or five digits in case the initial seed is set differently. The author has the opinion that a simulation method should provide estimates that are virtually independent of initial seed and therefore the Monte Carlo variance should be virtually zero. The specification search of the auxilliary model is

<sup>18</sup>To be precise, at the time of writing, the author had a Pentium 90 with 16 Mb of RAM and a Pentium 133 with 24 Mb of EDO RAM at his disposal. The P90 ran under Windows 3.1 and the P133 ran under Windows 95.

<sup>19</sup>Stichting Academisch Rekencentrum Amsterdam. The SP2 consists of 76 parallel IBM RS/6000 processors, running under AIX 4.1.4. The SP2 at SARA has 76 nodes, which makes it the largest one in Europe. The SP2's theoretical maximum rate is 20 Gigaflops/s. Four of the nodes are wide nodes with 512 Mbyte of memory each. Two wide nodes function as file servers with 180 Gbyte of disk space. Among the 72 thin nodes, 58 are equipped with 128 Mbyte internal memory, the remaining 14 with 256 Mbyte. Each thin node is equipped with 4.4 Gbyte local disk space, primarily intended for scratch files. For long term data storage a Magstar tape robot is available with a total capacity of 5.5 Terabyte (5.5 million Megabyte).

described in the above papers. The upshot is that for the exchange rates an EGARCH(1, 3)-H(4, 0) was found to be BIC-optimal for the S&P500 an EGARCH(1, 2)-H(5, 0) was found to be BIC-optimal.

To the exchange rates a SARMAV(1, 0), a SARMAV(2, 0) and a SARMAV(3, 0) were fitted. To the S&P500 data set a SARMAV(1, 0) and a SARMAV(1, 1) was fitted. The following results<sup>20</sup> for the exchange rate models were obtained for the SARMAV(1, 0) model

$$y_t = \sigma_t \epsilon_t$$

$$\ln \sigma_t^2 = -.103 + .894 \ln \sigma_{t-1}^2 + .270 \eta_t$$

(-4.20)
(34.9)
(17.4)

for the SARMAV(2, 0) model

$$y_t = \sigma_t \epsilon_t$$

$$\ln \sigma_t^2 = -.160 + .268 \ln \sigma_{t-1}^2 + .570 \ln \sigma_{t-2}^2 + .415 \eta_t$$

(-9.32)
(58.5)
(127)
(34.4)

and for the SARMAV(3, 0) model

$$y_t = \sigma_t \epsilon_t$$

$$\ln \sigma_t^2 = -.094 + 1.20 \ln \sigma_{t-1}^2 - 1.14 \ln \sigma_{t-2}^2 + .851 \ln \sigma_{t-3}^2 + .267 \eta_t$$

(-5.84)
(311)
(-335)
(108)
(31.9)

The estimates for the S&P500 series for the SARMAV(1, 0) model

$$y_t = \sigma_t \epsilon_t$$

$$\ln \sigma_t^2 = -.006 + .975 \ln \sigma_{t-1}^2 + .101 \eta_t$$

(-1.12)
(50.0)
(2.08)

and for the SARMAV(1, 1) model

$$y_t = \sigma_t \epsilon_t$$

$$\ln \sigma_t^2 = -.007 + .973 \ln \sigma_{t-1}^2 + (1 + .839 L) .058 \eta_t$$

(-.048)
(28.6)
(24737)
(54.3)

As far specification of the (A)SARMAV model is concerned we only report the Hansen  $J$ -test here in tables 2 and 3. For other specification tests in this context see van der Sluis (1997a and b). Although a  $P$ -value is a monotonic function of the actual evidence against  $H_0$ , it is very dangerous to choose the best model of these specifications on basis of the  $P$ -values (see Berger and Delampady (1987)). Different criteria will certainly be found in future research. For a more in-depth discussion of the above models the reader should consult van der Sluis (1997a) for the exchange rates and van der Sluis (1997b and c) for the S&P500 series.

## 6 Conclusion

This paper documents the software package EmmPack 1.01. A wide variety of stochastic volatility models can be estimated, although there are several limitations to the program. No auxiliary process for the mean is implemented in the C library yet and only univariate models can be considered. Currently the author is expanding his code in these two directions. A multivariate generalization is investigated in van der Sluis (1997d). Some generalizations of the stochastic volatility model (2) can also be estimated with EmmPack 1.01 see van der Sluis (1997c). However, the user should keep in mind that the auxiliary model should embed the stochastic volatility model that one wants to estimate. The mailbox is open for any comments.

---

<sup>20</sup>  $t-$  values are between brackets

## **Disclaimer**

THIS SOFTWARE AND SOURCE CODE ARE DISTRIBUTED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THESE ITEMS MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. WHILE EVERY EFFORT HAS BEEN MADE TO TEST THE PRODUCT IN A WIDE VARIETY OF OPERATING ENVIRONMENTS, GOOD PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF THE AUTHOR WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT.

## References

- [1] Abramowitz, M. and I.A. Stegun (1972): *Handbook of Mathematical Functions*. New York: Dover.
- [2] Akaike, H, (1973): "Information theory and an extension of the maximum likelihood principle," *Second International Symposium on Information Theory*, 267-281. Budapest: Akademia Kiado.
- [3] Andersen, T.G. (1992): "Volatility," Working paper No. 144, Northwestern University.
- [4] Andersen, T.G. (1994): "Stochastic autoregressive volatility: A framework for volatility modeling," *Mathematical Finance*, 4, 75-102.
- [5] Andersen, T.G., H-J Chung and B.E. Sørensen (1997), Efficient method of moments estimation of a stochastic volatility model: a monte carlo study. Working paper no. 229 Northwestern University.
- [6] Andersen, T.G. and J. Lund (1997): "Estimating continuous time stochastic volatility models of the short term interest rate," *Journal of Econometrics*, 77, 343-377.
- [7] Andersen, T.G. and J. Lund (1996): "Stochastic volatility and mean drift in the short rate diffusion: sources of steepness, level and curvature in the yield curve," Working paper No.214 Northwestern University.
- [8] Andersen, T.G. and B.E. Sørensen (1996): "Estimation of a stochastic volatility model: A Monte Carlo Study," *Journal of Business and Economic Statistics*, 14, 328-352.
- [9] Berger, J.O. and M. Delampady (1987): "Testing precise hypotheses," *Statistical Science*, 2, 317-352.
- [10] Bollerslev, T. (1986): "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, 31, 307-327.
- [11] Bollerslev, T., R. F. Engle and D. B. Nelson (1994): "ARCH models," in R.F. Engle and D.L. McFadden eds., *Handbook of Econometrics IV*, Chapter 49. Amsterdam: North-Holland. pp. 2959-3038.
- [12] Clark, P.K. (1973): "A subordinated stochastic process model with finite variance for speculative prices," *Econometrica*, 41, 135-155.
- [13] Cribari-Neto, F. (1997): "Econometric Programming Environments: GAUSS, Ox and S-PLUS," *Journal of Applied Econometrics*, 12, 77-89 and <http://www.siu.edu/~econ/software.zip>
- [14] Danielsson, J. and J.F. Richard (1993): "Accelerated gaussian importance sampler with application to dynamic latent variable models," *Journal of Econometrics*, 8, S153-S174.
- [15] Danielsson, J. (1994): "Stochastic volatility in asset prices: estimation with simulated maximum likelihood," *Journal of Econometrics*, 61, 375-400.
- [16] Danielsson, J. (1996a): "Multivariate stochastic volatility," Working paper University of Iceland. <http://www.hag.hi.is/~jond/msvm.ps>

- [17] Danielsson, J. (1996b): "Estimation of stochastic volatility models by simulated maximum likelihood: C++ code," *Studies in Non-Linear Dynamics and Econometrics* 1.
- [18] Doornik, J.A. (1996): *Object-Oriented Matrix Programming using Ox*, London: International Thomson Business Press and Oxford: <http://www.nuff.ox.ac.uk/Users/Doornik/>
- [19] Eastwood, B. (1991): "Asymptotic normality and consistency of seminonparametric regression estimators using an upward F-test rule," *Journal of Econometrics*, 48, 151-183.
- [20] Fenton, V.M. and A.R. Gallant (1996a): "Qualitative and asymptotic performance of SNP density estimators," Working Paper, University of North Carolina.
- [21] Fenton, V.M. and A.R. Gallant (1996b): "Convergence rates of SNP density estimators," *Econometrica*, 64, 719-727.
- [22] Fridman, M. and L. Harris, (1997), "A maximum likelihood approach for non-gaussian stochastic volatility models", Working paper University of Southern California. Forthcoming *Journal of Business Economics and Statistics*.
- [23] Gallant, A.R., D.A. Hsieh and G.E. Tauchen (1991): "On fitting a recalcitrant series: the pound/dollar exchange rate, 1974-1983," in W.A. Barnett, J. Powell and G.E. Tauchen (eds), *Nonparametric and semi-parametric methods in econometrics and statistics*, Fifth International Symposium in Economic Theory and Econometrics, 199-240, Cambridge: Cambridge University Press.
- [24] Gallant, A.R., D.A. Hsieh and G.E. Tauchen (1997): "Estimation of stochastic volatility models with diagnostics," *Journal of Econometrics*, 81, 159-192.
- [25] Gallant, A.R. and J.R. Long (1997): "Estimating Stochastic Differential Equations Efficiently by Minimum Chi-Square," *Biometrika*, 84, 125-141.
- [26] Gallant, A.R. and D.W. Nychka (1987): "Semi-nonparametric maximum likelihood estimation," *Econometrica*, 55, 363-390.
- [27] Gallant, A. R. and G.E. Tauchen (1996): "Which moments to match?," *Econometric Theory*, 12, 657-681.
- [28] Ghysels, E., A. Harvey and E. Renault (1996): "Stochastic Volatility," in G.S. Maddala and C.R. Rao eds., *Handbook of Statistics: Statistical Methods in Finance*, vol 14: *Statistical Methods in Finance*. Amsterdam: Elsevier.
- [29] Gouriéroux, C., A. Monfort and E. Renault (1993): "Indirect Inference," *Journal of Applied Econometrics*, 8, S85-S119.
- [30] Hannan, E.J. and B.G. Quinn (1979): "The determination of the order of an autoregression," *Journal of the Royal Statistical Society*, B 41, 190-195.
- [31] Harvey A.C. E. Ruiz and N. Shephard (1994): "Multivariate stochastic variance models," *Review of Economic Studies*, 61, 247-264.
- [32] Harvey A.C. and N. Shephard (1996): "Estimation of an asymmetric stochastic volatility model for asset returns," *Journal of Business and Economic Statistics*, 14, 429-434.

- [33] Hull, J. and A. White (1987): "The pricing of options on assets with stochastic volatilities," *Journal of Finance*, 42, 281-300.
- [34] Jacquier, E., N.G. Polson and P.E. Rossi (1994): "Bayesian analysis of stochastic volatility models (with discussion)," *Journal of Business and Economics Statistics*, 12, 371-417.
- [35] Jiang, G.J. and P.J. van der Sluis (1997): "Pricing of stock options with systematic stochastic volatility and uncertain interest rates," Working paper University of Groningen and University of Amsterdam.
- [36] Kim, S., N. Shephard and S. Chib (1996): "Stochastic volatility: optimal likelihood inference and comparison with ARCH models," Discussion paper, Nuffield College, Oxford. Forthcoming *Review of Economic Studies*.
- [37] Koopman, S.J., N. Shephard and J.A. Doornik (1997), SsfPack 2.0: Statistical algorithms for models in state space, <http://cwis.kub.nl/~few5/center/staff/koopman/ssfpack.htm>
- [38] Lütkepohl, H. (1990): *Introduction to multiple time series analysis*. Berlin: Springer Verlag
- [39] Monfardini, C. (1996): "Estimating stochastic volatility models through indirect inference," Working paper, European University Institute, Florence.
- [40] Nelson, D.B. (1991): "Conditional heteroskedasticity in asset returns: a new approach," *Econometrica*, 59, 347-370.
- [41] Phillips, P.C.B. (1983): "ERA's: A new approach to small sample theory," *Econometrica*, 51, 1505-1527.
- [42] Quinn, B.G. (1980): "Order determination for a multivariate autoregression," *Journal of the Royal Statistical Society*, B 42, 182-185.
- [43] Richard, J.F. and Zhang, W. (1995a): "Accelerated importance sampling," University of Pittsburgh Mimeo.
- [44] Richard, J.F. and Zhang, W. (1995b): "Accelerated Monte Carlo integration: an application to dynamic latent variable models," University of Pittsburgh Mimeo.
- [45] Ross, S.M. (1990): *A course in simulation*, New York: MacMillan.
- [46] Ruiz, E. (1994): "Quasi-maximum likelihood estimation of stochastic volatility models," *Journal of Econometrics*, 63, 289-306.
- [47] Sandmann, G. and S.J. Koopman (1997): "Estimation of stochastic volatility models via Monte Carlo simulation", Working paper London School of Economics.
- [48] Schwarz, G. (1978): Estimating the dimension of a model, *Annals of Statistics*, 6, 461-464.
- [49] Schotman, P. and R. Mahieu (1994): "Stochastic volatility and the distribution of exchange rate news," LIFE-University of Maastricht mimeo.
- [50] Shephard, N. (1996a): "Ox software for volatility models," Discussion paper and freeware, Nuffield College, Oxford and <http://hicks.nuff.ox.ac.uk/users/shephard/ox/>

- [51] Shephard, N. (1996b), "Statistical aspects of ARCH and stochastic volatility", in D.R.Cox, D.V.Hinkley and O.E. Barndorff-Nielsen (eds), *Time Series Models: In econometrics, finance and other fields*, 1-67, London: Chapman and Hall.
- [52] Tauchen, G.E. and M. Pitts (1983): "The price variability-volume relationship on speculative markets," *Econometrica* 51, 485-505.
- [53] Tauchen, G.E. (1997): "New minimum chi-square methods in empirical finance," in David M. Kreps and Kenneth F. Wallis eds., *Advances in Economics and Econometrics: Theory and Applications*, Vol. III, Econometric Society Monographs No. 28, Cambridge: Cambridge University Press, pp. 279-317.
- [54] Taylor, S.J. (1986): *Modeling Financial Time Series*. Chichester: Wiley.
- [55] Taylor, S.J. (1994): "Modeling stochastic volatility: A review and comparative study," *Mathematical Finance* 4, 183-204.
- [56] van der Sluis, P. J. (1997a): "Post-sample prediction tests for the Efficient Method of Moments," Tinbergen Institute Discussion Paper TI97-054/4. [http : //www.fee.uva.nl/vak.groep/AKE/vdsluis.htm](http://www.fee.uva.nl/vak.groep/AKE/vdsluis.htm)
- [57] van der Sluis, P.J. (1997b): "Computationally attractive stability tests for the Efficient Method of Moments," Tinbergen Institute Discussion Paper TI97-087/4. [http : //www.fee.uva.nl/vak.groep/AKE/vdsluis.htm](http://www.fee.uva.nl/vak.groep/AKE/vdsluis.htm)
- [58] van der Sluis, P.J. (1997c): "Estimation of fat-tailed and asymmetric stochastic volatility models with the efficient method of moments: an application to the S&P500 and some Monte Carlo results", Working paper, University of Amsterdam.
- [59] van der Sluis, P.J. (1997d): "Multivariate stochastic volatility: estimation and inference with the efficient method of moments," Working paper and unpublished freeware, University of Amsterdam.
- [60] White, H. (1994): *Estimation, inference and specification analysis*, Cambridge: Cambridge University Press.



## A Description of the auxiliary model

In this appendix the class of auxiliary models that is used in `EmmPack` will be described. As in section 3 consider the stochastic process  $y_t(\theta_0)$ . The model for the conditional mean is  $\mu_t(\beta) := \mathbb{E}_{t-1}[y_t(\theta)]$ . The zero mean process is then defined as  $\epsilon_t(\beta) := y_t(\theta) - \mu_t(\beta)$ . The model for the conditional variance as  $\sigma_t^2(\beta) := \text{Var}_{t-1}[\epsilon_t(\beta)] = \mathbb{E}_{t-1}[\epsilon_t^2(\beta)]$ . This leads to the standardized process:  $z_t(\beta) := \epsilon_t(\beta)[\sigma_t^2(\beta)]^{-1/2}$ . Let  $f(z_t; \eta)$  be the density for  $z_t(\beta)$ , and let  $\eta$  denote the nuisance parameters  $\eta \in H \subseteq \mathbb{R}^k$ .

Let  $\psi := (\beta', \eta')$ . The loglikelihood of  $y_t$  equals

$$l_t(y_t, \psi) := \ln[f\{z_t(\beta); \eta\}] - 0.5 \ln[\sigma_t^2(\beta)], t = 1, 2, \dots$$

By the *prediction error decomposition* we get the following expression for the likelihood of full sample  $L_T(y_1, \dots, y_T; \psi) := \sum_{t=1}^T l_t(y_t; \psi)$  Let the score function be  $S_T(y_1, \dots, y_T; \psi) := \sum_{t=1}^T \nabla_\psi l_t(y_t; \psi)$

For conditional mean and variance parameters we get

$$\nabla_\theta l_t(y_t; \psi) = f[z_t(\beta); \eta]^{-1} f'[z_t(\beta); \eta] \nabla_\beta z_t(\beta) - 0.5 \sigma_t^{-2}(\beta) \nabla_\theta \sigma_t^2(\beta)$$

here  $f'[z_t(\beta); \eta] := \frac{df[z_t(\beta); \eta]}{dz_t(\beta)}$  and  $\nabla_\beta z_t(\beta) = -\nabla_\beta \mu_t(\beta) \sigma_t^{-1}(\beta) - 0.5 \epsilon_t(\beta) \sigma_t^{-3}(\beta) \nabla_\beta \sigma_t^2(\beta)$

In `EmmPack` we specify the standard normal density for  $z_t$ . Now there are no nuisance parameters so  $\psi = \beta$ . The gradient becomes

$$\nabla_\theta l_t(y_t; \beta) = \nabla_\beta \mu_t(\beta) \epsilon_t(\beta) \sigma_t^{-2}(\beta) + 0.5 \nabla_\beta \sigma_t^2(\beta) \sigma_t^{-2}(\beta) [\epsilon_t(\beta)^2 \sigma_t^{-2}(\beta) - 1]$$

It is known that in many models the normal distribution cannot capture all the excess kurtosis that is often observed in financial markets. An alternative may be to use Student's  $t$  distribution or the Generalized Error Distribution (GED) of Nelson (1991). However since we will use a SNP density for the  $z_t$  we follow the easiest way and use the Gaussian density.

In this version of `EmmPack`, the leading term of the SNP model, specified by  $\mu_t(\beta)$  and  $\sigma_t^2(\beta)$  will be,  $\mu_t(\beta) = 0$  and  $\sigma_t^2(\beta)$  following the EGARCH model of Nelson (1991). A formal representation of the EGARCH( $p, q$ ) model is given by

$$\begin{aligned} y_t &= \sigma_t z_t \\ \sigma_t^2 &= \exp\left\{\omega + \sum_{i=1}^p \gamma_i \ln \sigma_{t-i}^2 + (1 + \alpha_1 L + \dots + \alpha_q L^q) [\kappa_1 z_{t-1} + \kappa_2 \langle b(z_{t-1}) - \sqrt{2/\pi} \rangle]\right\} \end{aligned}$$

here  $b(z)$  denotes a two times differentiable approximation to the absolute value function,  $|z|$ .<sup>21</sup> This is important for numerical reasons. Interpretation the coefficient of the EGARCH parameters is as follows: The  $\gamma$  parameters give us a idea of the persistence in the data, in particular  $\sum_{i=1}^p \gamma_i$  will serve as a measure of persistence. The  $\omega$  parameter is a measure of the unconditional variance. The term  $1 + \sum_{i=1}^q \alpha_i$  reflects the short-run sensitivity to innovations. The term  $\kappa_1$  reflects the asymmetric volatility effect that is often present in financial time series. Finally, the  $\kappa_2$  parameter measures the changes in  $z_t$  in reaction with the conditional heteroskedasticity.

<sup>21</sup>A good choice is e.g.  $b(z) = |z|$  for  $|z| \geq \pi/2K$  and  $b(z) = (\pi/2 - \cos(Kz))/K$  for  $|z| < \pi/2K$  and set e.g.  $K = 100$ . The derivatives are  $b'(z) = -1$  for  $z < -\pi/2K$ ,  $b'(z) = 1$  for  $z > \pi/2K$ ,  $b'(z) = \sin(Kz)$  for  $|z| < \pi/2K$

The SNP density based on the Gaussian distribution is given by formula (3). For identification we need  $a_{00} = 1$ . Let  $K = \max(K_z, K_x)$ . In what follows  $M = 1$  or  $M = 0$ . In case  $K_x = 0$ , letting  $K_z > 0$  induces a time-homogeneous non-Gaussian error structure. The case  $K_x > 0$  induces heterogeneous innovation densities beyond the EGARCH model. Since we believe the EGARCH model captures all this heterogeneity,  $K_x > 0$  will not be necessary. This was also very much supported by our empirical findings. To the author's knowledge there are no guidelines provided yet for assessment of the individual coefficients of the polynomial. For computational ease we employ polynomials that are of Hermite form:

$$\text{He}_i(z) = \sum_{j=0}^{\lfloor i/2 \rfloor} (-1)^j \frac{i!}{j!2^j(i-2j)!} z^{i-2j}$$

$$\int \text{He}_i \text{He}_j \exp\{-\frac{1}{2}z^2\} dz = \begin{cases} 0 & i \neq j \\ \sqrt{2\pi}i! & i = j \end{cases}$$

Let

$$\overline{\text{He}}_i(z) := (i!)^{-1/2} \text{He}_i(z)$$

A nice result of this normalization is that

$$\int \overline{\text{He}}_i(u) \overline{\text{He}}_j(u) \phi(u) dz = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

The polynomial is now taken as

$$P_K(z, x) = \sum_{i=0}^{K_z} \gamma_i(x) \overline{\text{He}}_i(z)$$

where  $\gamma_i(x)$  is a monomial in  $x$ , e.g.  $\gamma_i(x) = \gamma_{i0} + \gamma_{i1}x$

$$\int P_K(z, x)^2 \phi(z) dz = \sum_{i=0}^{K_z} \gamma_i(x)^2$$

Later on we will need

$$\frac{\partial z_t}{\partial \psi_k} = \frac{\partial}{\partial \psi_k} \left( \frac{y_t - \mu_t}{\sqrt{\sigma_t^2}} \right) = - \left( \frac{1}{\sqrt{\sigma_t^2}} \frac{\partial \mu_t}{\partial \psi_k} + \frac{z_t}{2\sigma_t^2} \cdot \frac{\partial \sigma_t^2}{\partial \psi_k} \right)$$

and the recurrence relation

$$\begin{aligned} \text{He}_{i+1}(z) &= z \text{He}_i(z) - i \text{He}_{i-1}(z) \\ \text{He}_0(z) &= 1 \\ \text{He}_1(z) &= z \end{aligned}$$

A differential relation that is useful is given by

$$\frac{\partial \text{He}_i(z)}{\partial z} = i \text{He}_{i-1}(z)$$

An individual component of prediction error decomposition of the likelihood function now reads

$$\ln f_K(y_t | x_t; \psi) = \underbrace{\ln[P_K(z_t, x_t)^2]}_{\text{I}} + \underbrace{\ln \phi(z_t) - 0.5 \ln(\sigma_t^2)}_{\text{II}} - \underbrace{\ln\left[\sum_{i=0}^{K_z} \gamma_i(x_t)^2\right]}_{\text{III}}$$

Only I and III require special treatment as regards the analytic expression of the derivative. The remaining part of the score is simple and well-known. For completeness, expressions of the derivative of part II is given in appendix B. What is left are some derivatives which are needed in the score generator or that can be used as analytical derivatives. Differentiation of part III

$$\frac{\partial[\ln \int P_K(u, x)^2 \phi(u) du]}{\partial \psi_k} = \frac{\partial[\ln \sum_{i=0}^{K_z} \gamma_i(x)^2]}{\partial \psi_k} = \frac{2}{\sum_{i=0}^{K_z} \gamma_i(x)^2} \sum_{i=0}^{K_z} \gamma_i(x) \underbrace{\frac{\partial \gamma_i(x)}{\partial \psi_k}}_{\text{Jacobian}}$$

note

$$\frac{\partial \gamma_i(x)}{\partial \psi_k} = 0 \text{ if } \psi_k \text{ is a leading term parameter}$$

Differentiation of part I

$$\frac{\partial \ln P_K(z, x)^2}{\partial \psi_k} = \frac{2 \cdot \frac{\partial P_K(z, x)}{\partial \psi_k}}{P_K(z, x)}$$

$$\frac{\partial P_K(z, x)}{\partial \psi_k} = \frac{\partial}{\partial \psi_k} \left\{ \sum_{i=0}^{K_z} \gamma_i(x) \overline{\text{He}}_i(z) \right\} = \sum_{i=0}^{K_z} \underbrace{\frac{\partial \gamma_i(x)}{\partial \psi_k}}_{\text{Jacobian}} \overline{\text{He}}_i(z) + \sum_{i=0}^{K_z} \gamma_i(x) \frac{\partial \overline{\text{He}}_i(z)}{\partial \psi_k}$$

where

$$\frac{\partial \overline{\text{He}}_i(z)}{\partial \psi_k} = (i!)^{-1/2} \frac{\partial \text{He}_i(z)}{\partial z} \frac{\partial z}{\partial \psi_k}$$

About the Jacobian  $\frac{\partial \gamma_i(x)}{\partial \psi_k}$  in case  $\psi_k$  is not a leading term parameter. For example in the case  $K_x = 1$  and all non-leading term parameters are vectorized as  $\gamma = (\gamma_{00}, \gamma_{01}, \gamma_{10}, \gamma_{11}, \dots, \gamma_{K_z 0}, \gamma_{K_z 1})'$  we have

$$\frac{\partial \gamma_i(x)}{\partial \gamma} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ x \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ where the 1 occurs at position } 2i - 1 \text{ and the } x \text{ at } 2i$$

## B Derivatives of EGARCH process

If  $\sigma_t^2$  follows an EGARCH( $p, q$ ) process then

- $\nabla_{\omega} \sigma_t^2(\theta) = \sigma_t^2(\theta) [1 + \sum_{i=1}^p \frac{\beta_i}{\sigma_{t-i}^2(\theta)} \nabla_{\omega} \sigma_{t-i}^2(\theta) + (1 + \sum_{i=1}^q \alpha_i L^i) (\kappa_1 \nabla_{\omega} z_{t-1} + \kappa_2 b'(z_{t-1}) \nabla_{\omega} z_{t-1})]$
- $\nabla_{\alpha_j} \sigma_t^2(\theta) = \sigma_t^2(\theta) [\sum_{i=1}^p \frac{\beta_i}{\sigma_{t-i}^2(\theta)} \nabla_{\alpha_j} \sigma_{t-i}^2(\theta) + \kappa_1 z_{t-j-1} + \kappa_2 \{b(z_{t-j-1}) - \sqrt{\frac{2}{\pi}}\} + (1 + \sum_{i=1}^q \alpha_i L^i) \{\kappa_1 \nabla_{\alpha_j} z_{t-1} + \kappa_2 b'(z_{t-1}) \nabla_{\alpha_j} z_{t-1}\}]$
- $\nabla_{\beta_i} \sigma_t^2(\theta) = \sigma_t^2(\theta) [\ln \sigma_{t-i}^2 + \sum_{j=1}^p \frac{\beta_j}{\sigma_{t-j}^2(\theta)} \nabla_{\beta_i} \sigma_{t-j}^2(\theta) + (1 + \sum_{i=1}^q \alpha_i L^i) \{\kappa_1 \nabla_{\beta_i} z_{t-1} + \kappa_2 b'(z_{t-1}) \nabla_{\beta_i} z_{t-1}\}]$
- $\nabla_{\kappa_1} \sigma_t^2(\theta) = \sigma_t^2(\theta) [\sum_{i=1}^p \frac{\beta_i}{\sigma_{t-i}^2(\theta)} \nabla_{\kappa_1} \sigma_{t-i}^2(\theta) + (1 + \sum_{j=1}^q \alpha_j L^j) \{z_{t-1} + \kappa_1 \nabla_{\kappa_1} z_{t-1} + \kappa_2 b'(z_{t-1}) \nabla_{\kappa_1} z_{t-1}\}]$

- $\nabla_{\kappa_2} \sigma_t^2(\theta) = \sigma_t^2(\theta) [\sum_{i=1}^p \frac{\beta_i}{\sigma_{t-i}^2(\theta)} \nabla_{\kappa_2} \sigma_{t-i}^2(\theta) + (1 + \sum_{j=1}^q \alpha_j L^j) \{ \kappa_1 \nabla_{\kappa_2} z_{t-1} + b(z_{t-1}) - \sqrt{\frac{2}{\pi}} + \kappa_2 b'(z_{t-1}) \nabla_{\kappa_2} z_{t-1} \}]$

The recursive formulas must be started up. In the program we set  $\sigma_{-1}^2, \dots, \sigma_{-p}^2 = z'z/n$ , where  $z$  is based on the current parameter estimates.

## C Antithetic Variables

In this context using antithetic variables as a variance reduction technique is very important. See e.g. Ross (1990, chapter 8), among others, for a review of variance reduction techniques. In short the antithetic variates technique is based on the fact that if we wish to estimate by simulation  $\theta = E[X]$  on basis of two generated variables  $X_1$  and  $X_2$ . We are better off in case  $X_1$  and  $X_2$  are negatively correlated than independent, because

$$\text{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4} [\text{Var}(X_1) + \text{Var}(X_2) + 2\text{Cov}(X_1, X_2)]$$

Likewise here in the generation of the stochastic volatility model (2), for one series the variables  $\epsilon_t$  and  $\eta_t$  are generated and for the other the variables  $-\epsilon_t$  and  $-\eta_t$  are used. The scores are now correlated leading to a considerable variance reduction.

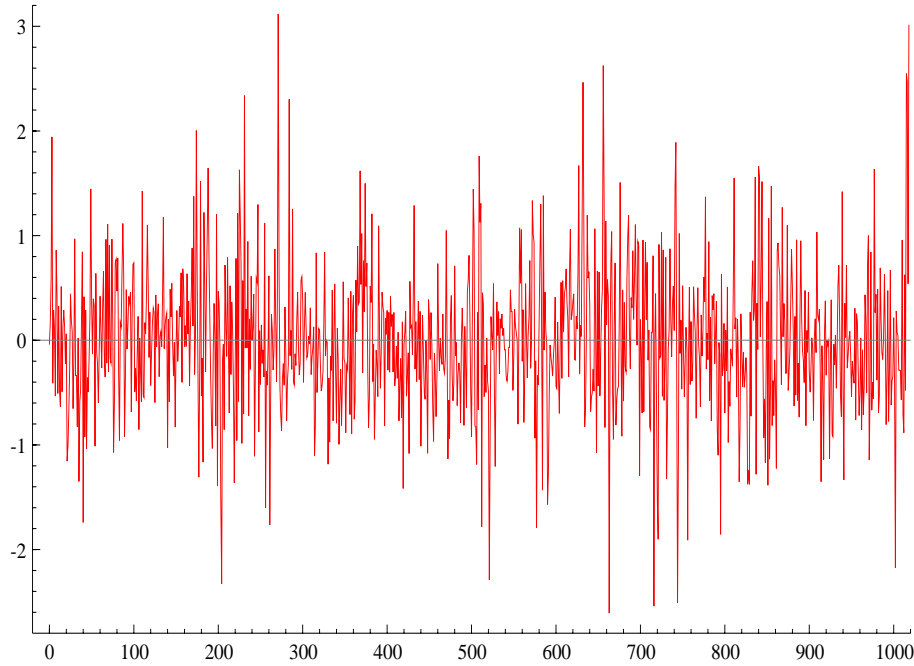


Figure 1: British Pound / Canadian Dollar exchange rates

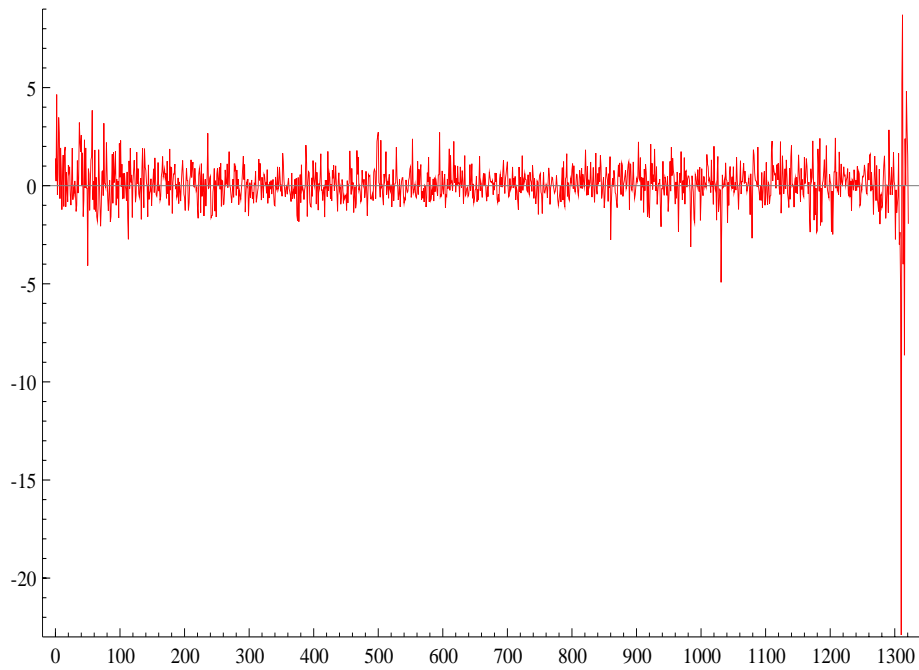


Figure 2: S&P500

	XR	S&P500
Mean	-.005	.068
Std.Dv.	.695	1.19
Skewness	.269	-5.36
Excess Kurtosis	1.81	111
Minimum	-2.61	-22.9
Maximum	3.12	8.71
Normality Chi <sup>2</sup>	85.7	1894
#observations	1019	1322
period	9-1988 till 9-1992	9-1981 till 10-1987

Table 1: Some preliminary statistics of the series.

<i>J</i> -test	SARMAV(1, 0)	SARMAV(1, 1)
<i>J</i>	33.8	33.6
df	8	7
<i>P</i> -value	.000	.000
$\alpha_0$	1.33	.212
$\alpha_1$	-1.52	-1.52
$\alpha_2$	-2.06	-2.05
$\gamma_1$	-2.30	-2.33
$\kappa_1$	1.85	1.87
$\kappa_2$	-.883	-.875
$a_{10}$	-3.57	-3.59
$a_{20}$	-2.46	-3.53
$a_{30}$	-.367	-.408
$a_{40}$	-2.40	-2.50
$a_{50}$	2.74	3.06

Table 2: Hansen *J*-test and individual *t*-values for the models for the S&P 500

<i>J</i> -test	SARMAV(1, 0)	SARMAV(2, 0)	SARMAV(3, 0)
<i>J</i>	13.4	11.6	6.87
df	8	7	6
<i>P</i> -value	.100	.113	.333
$\alpha_0$	-.846	-.847	-.259
$\alpha_1$	.397	.435	-1.43
$\alpha_2$	-.754	-.950	-1.17
$\alpha_3$	-2.28	-2.22	-1.31
$\gamma_1$	.271	.258	-.333
$\kappa_1$	-1.79	-1.78	-1.95
$\kappa_2$	.030	-.755	-.220
$a_{10}$	.951	.942	.914
$a_{20}$	-1.71	-1.38	-.649
$a_{30}$	-1.17	-1.14	-1.25
$a_{40}$	-1.84	-1.37	-.433

Table 3: Hansen *J*-test and individual *t*-values for the models for the Exchange Rate series

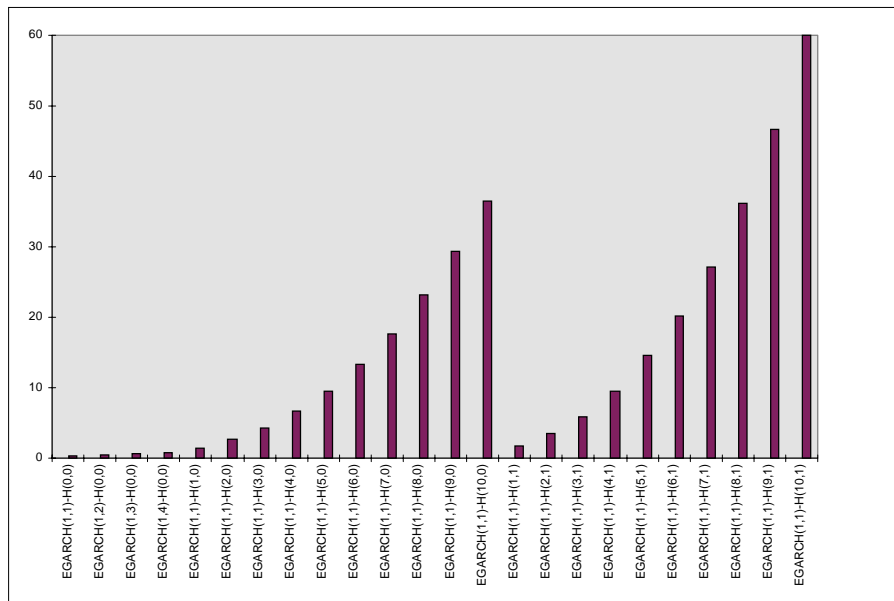


Figure 3: Times in seconds for calculation of score generator, run on a P5-166 with 32Mb under Win95. Times are for  $N = 2 \times 5,000$ . using EmmPack 1.01.