Computationally-intensive Econometrics using a Distributed Matrix-programming Language

BY JURGEN A. DOORNIK, DAVID F. HENDRY AND NEIL SHEPHARD
NUFFIELD COLLEGE, UNIVERSITY OF OXFORD, OXFORD OX1 1NF

Abstract

This paper reviews the need for powerful computing facilities in econometrics, focusing on concrete problems which arise in financial economics and in macroeconomics. We argue that the profession is being held back by the lack of easy to use generic software which is able to exploit the availability of cheap clusters of distributed computers. Our response is to extend, in a number of directions, the well known matrix-programming interpreted language Ox developed by the first author. In particular we study 3 levels of extensions: (i) Ox with parallelization explicit in the Ox code; (ii) Ox with a parallelized run-time library; (iii) Ox with a parallelized interpreter. We give examples in the context of financial economics and time-series modelling. *Keywords:*

Econometrics; High-performance computing; Matrix-programming language.

JEL classification: C32†

1. Introduction

This paper considers high-performance computing from the perspective of one of the social sciences. The first question that comes to mind is whether the social sciences have any need for high-performance computing at all. In practice, the social sciences span a wide array of research activities, ranging from economics and sociology to psychology and political sciences. Of course, the boundaries between these fields are not well-defined, for example, political science increasingly uses econometric techniques for data analysis. Even the outside boundaries are not well-defined, illustrated by the existence of the society for econophysics, and the well-established field of psychophysics.

Within the social sciences, it is fairly safe to say that econometrics is one of the most technical and quantitative fields, with, in many cases, heavy use of computation to solve otherwise intractable problems.

In a narrow sense, the task of econometrics is to develop statistical techniques for the analysis of non-experimental data, and to study the properties of these techniques.‡ The problem is daunting: usually the observed data are a consequence of decisions made by millions of people, inaccurately measured, and often collected only at quarterly or annual intervals. Economies tend to be highly non-statioanry and subject to sudden major interventions. In recent years, there has also been a huge growth in financial econometrics, driven by the availability of data observed minute-by-minute or even tick-by-tick: examples are exchange rate and stock market data. In financial econometrics the emphasis is on modelling the volatility aspects of the data, rather than the central tendency.

† Correspondence to: jurgen.doornik@nuffield.ox.ac.uk
‡ Although experiments are also performed, especially for research into auction theory.

The question remains, though, whether there is a need for high-performance computing to achieve its ends. Some of the authors' research requires weeks and sometimes months of computation on an up-to-date PC. Other econometricians use similarly lengthy calculations, indicating that there is such a need. Also, for financial applications it can be important to obtain model results very quickly, so that they can be used in trading decisions. Section 4 provides some detailed illustrations.

The second question is whether there is any benefit in approaching high-performance computing from an econometric perspective. After all, why not just copy what has already been done in other sciences? To some extent the answer to this is yes: we will certainly borrow as much as possible from numerical analysis and use available computational tools, such as that illustrated by Hargrove, Hoffman and Sterling (2001). However, there are two reasons for a slightly different perspective, namely the cost of computing and the opportunity cost of labour. The former matters because equipment budgets in social sciences research tend to be small. They rarely go beyond the purchase of a state-of-the-art PC. Therefore, the availability now of low-cost supercomputing, possibly by combining standard PC equipment, is very important. A more substantial barrier is the cost of programming for high performance. Usually, the resulting program is specific to the problem at hand and hardware-specific, and may take weeks or months to develop. For example, Chong and Hendry (1986) developed Monte Carlo simulation code for a distributed array processor using DAP FORTRAN, but writing and testing the code took several months for a single problem. Unfortunately, because our subject area puts a low value on computational skills, and young researchers face much higher salaries outside academia, it is usually uneconomical to adopt high-performance computing when there is a substantive novel programming effort required.

In this paper, we propose a partial solution to that problem. Our approach is to take a matrix-programming language called Ox (see Doornik, 1999), and try to hide the parallelization within the language. This way, the same Ox program can be run unmodified on a stand-alone PC or on a cluster as discussed in section 5.

First, section 2 explains the main factors behind the need for powerful computing facilities in econometrics, then 3 provides some historical background to computing in econometrics. Next, section 4 describes three typical computationally-intensive econometric problems, all of which exceed the capabilities of the present generation of PCs. Section 5 outlines our proposed solution based on Ox.

2. The need for computer power in econometrics

There are four main facets of econometric analyses that lead to major demands on computing power.

First, economics data which are intrinsically high dimensional: there are literally billions of transactions per day in large economies; and economies are closely linked by trade and financial flows which amount to trillions of dollars per annum. While models necessarily simplify dramatically, the larger econometric systems comprise thousands of equations each with numerous parameters estimated from time-series observations. Simulation analyses of such large estimated econometric models is often the only way to obtain, say, interval forecasts, and doing so poses significant demands on computing resources.

Secondly, economics data are highly non-stationary, both evolving and subject to major shocks. The means and variances of most economic time series have changed greatly over the last few centuries, reflecting the Industrial Revolution, the Electrical Revolution

and more recently the Technological Revolution. Moreover, economies are subject to major political and legislative changes, including changes in the regimes of both macro and micro economic policy (with fiscal, monetary, and exchange-rate regimes for the former, and nationalization and privatization as salient examples of the latter), wars, and the creation and destruction of major trading blocks. Thus, economies are subject to sudden and unanticipated shifts, the effects of which are relatively long-lasting. Consequently, recursive methods are essential, and stochastic simulation of artificial processes provide one of the few ways of obtaining approximations to finite-sample distributions of estimators and tests. Subsection a illustrates the types of information needed to investigate one aspect of non-stationarity and highlights the resulting computational demands.

Thirdly, any quantitative description of an economy is inherently non-linear. At the most basic level, identities (such as those comprising National Income Accounts) are linear functions of the observations, but almost all models involve at least log-linear relationships (see Hendry, 1995, for discussion). Thus, the likelihood functions that require maximization are complicated, time-dependent and very high dimensional. The early researchers into econometric computations, such as Eisenpress and Greenstadt (1966), were doubtful that appropriate estimators could be feasibly calculated. Subsection b focuses on simulation-based inference where high-dimensional integrals are mapped into conditional expectations and estimated by the means of simulation samples.

Fourthly, the complexity of relationships between variables in economies requires data-based selection of relationships from a larger set of potential candidate variables, often using many different selection criteria and exploring all feasible simplifications (see e.g., Hendry and Krolzig, 2001). The statistical distributions of the outcomes from such preocedures have eluded formal analysis, so necessitate Monte Carlo simulation studies. While the conventional drawbacks of specificity and imprecision of simulation can be overcome (see e.g., Hendry, 1984), nevertheless the development of appropriately calibrated response surface analogues to theoretical distributions require computational capabilitites and speeds far in excess of those available in the most powerful PCs or work stations. Subsection c explains this aspect.

3. Historical background

Throughout its history, available computational power has provided a constraint on the feasible applications of econometrics. As Hendry (2001) note in their historical review:

> Bean (1929) reported that a single four variable regression analysis for 30 observations would take about 8 hours of work.

Despite an increase in speed of about 10^9, and significant improvements in accuracy, the scale of analyses have risen at least as fast. Bean's calculations were of the order of Tk^3, for a sample of size T and a model with k parameters. However, one complete path search for one equation involves approximately 2^n such regressions where $n = 40$ is not uncommon; and a Monte Carlo study thereof might require 10^4 replications, leading to around 10^{16} regression estimates varying in size from 1 through 40 variables, usually with $T > 100$. Even with clever shortcuts, investigators confront massive tasks. It can be no surprise that a substantive fraction of the research effort in econometrics has been devoted to devising computationally-feasible methods given existing computers (see e.g., Hendry, 1976).

The first major econometric methods for estimating macro-econometric systems of non-linear dynamic equations created computational demands that could not be met (see Eisenpress and Greenstadt, 1966), and led to a proliferation of 'short-cuts' to provide operational approaches (see e.g., Hendry, 1976). Phillips (1985) recounts the need to hard-wire early computers to achieve non-linear optimization.

Even with the advent of more powerful computers like the IBM 360/65, enhanced by fast FORTRAN compilers, available computational capabilities remained a binding constraint for the discipline. For example, the likelihood functions for even small macro-econometric systems might involve several hundred parameters and take hours to optimize, so detailed Monte Carlo simulation studies could not be performed.

By the early 1980s, matrix computers offered a feasible route, but as discussed above, posed much greater labour costs, necessitating complete rethinking of programs – and of course rewriting their code.

This background explains the considerable interest econometrics have in solutions to 'supercomputing' that are cheap in both capital and labour and motivates our approach.

4. Some typical problems

(*a*) *Response surfaces for cointegration analysis*

(*b*) *Simulation-based econometric inference*

(i) *Motivation*

One of the main motivations for the development of simulation based econometric methods has been the professions' interest in estimating of analytically intractable non-linear economic models. Some of this has been carried out in microeconometrics. See, for example, the work of McFadden (1989), Hajivassiliou and Ruud (1994) and Hajivassiliou and McFadden (1998). Our focus will be on the problem of carrying out inference for discretely observed continuous time processes. These diffusion based models play a crucial role in modern financial economics, providing the basis of most option pricing, asset allocation and term structure theory currently being used. However, traditionally we have not had strong methods for estimating such models, especially when some components of the model are not observed.

In the econometric literature at least two basic methods have been either invented or advanced to deal with this type of problem. Both are based on simulation based estimation. The first is the use of importance sampling and Markov chain Monte Carlo methods to perform likelihood inference for these models. Leading references to this include Danielsson and Richard (1993), Danielsson (1994), Jacquier, Polson and Rossi (1994), Kim, Shephard and Chib (1998), Sandmann and Koopman (1998), Elerian, Chib and Shephard (2001) and Durham and Gallant (2001).

More originally, econometricians have been developing simulation based moment dependent inference methods in their work on indirect inference. Leading references to this include Gourieroux, Monfort and Renault (1993), Smith (1993) and Gallant and Tauchen (1996). Important references in the context of continuous time models include Andersen and Lund (1997), Andersen, Benzoni and Lund (1999), Gallant, Hsieh and Tauchen (1997), Gallant, Hsu and Tauchen (1998).

In all of these papers the estimation of the above models is computationally intensive, often taking many minutes and sometimes many days to estimate the models. In some cases

the methods are so slow that there have not been any Monte Carlo studies of the sampling performance of the estimation methods. The simulation nature of the methods does mean that they are well suited to being sped up using distributed computing technology. To our knowledge this has not really been carried out.

(ii) *Continous time stochastic volatility process*

In this paper we will illustrate the potential use of our approach to distributed computers by applying it to the estimation of a stochastic volatility model. The starting point for this model is the so called Black-Scholes or Samuelson model which models the log of an asset price by the solution to the stochastic differential equation

$$\mathrm{d}x^*(s) = \left\{\mu + \beta\sigma^2\right\}\mathrm{d}s + \sigma \mathrm{d}w(s), \quad s \in [0, S], \tag{4.1}$$

where $w(s)$ is standard Brownian motion. This means aggregate returns over intervals of length $\Delta > 0$, are

$$\begin{aligned} y_t &= \int_{(t-1)\Delta}^{t\Delta} \mathrm{d}x^*(s) = x^*(n\Delta) - x^*\left\{(n-1)\Delta\right\} && (4.2)\\ &\sim NID(\mu\Delta + \beta\sigma^2\Delta, \sigma^2\Delta). && (4.3) \end{aligned}$$

Unfortunately for moderate to small values of Δ (corresponding to returns measured over 5 minute to one day intervals) returns are typically heavy-tailed, exhibit volatility clustering (in particular the $|y_n|$ are correlated) and are skew, although for higher values of Δ a central limit theorem seems to hold and so Gaussianity becomes a less poor assumption for $\{y_t\}$ in that case. This means that every single assumption underlying the Black-Scholes model is routinely rejected by the type of data usually used in practice.

This common observation, which carries over to the empirical rejection of option pricing models based on this model, has resulted in an enormous effort to develop empirically more reasonable models which can be integrated into finance theory. The most successful of these are the generalised autoregressive conditional heteroskedastic (GARCH) and the diffusion based stochastic volatility (SV) processes. This very large literature, which was started by Clark (1973), Engle (1982) and Taylor (1982), is reviewed in, for example, Bollerslev, Engle and Nelson (1994), Ghysels, Harvey and Renault (1996) and Shephard (1996).

The model we will work with will be of an SV type, based on a more general system of stochastic differential equations,

$$\begin{aligned} \mathrm{d}x^*(s) &= \left\{\mu + \beta\sigma^2(s)\right\}\mathrm{d}s + \sigma(s)\mathrm{d}w(s), && (4.4)\\ \mathrm{d}\log\sigma^2(s) &= -\lambda\left\{\log\sigma^2(s) - \xi\right\} + \eta\mathrm{d}b(s), \qquad \lambda > 0, && (4.5) \end{aligned}$$

where w and b are here assumed to be independent, standard Brownian motions. Our desire would be to carry out likelihood inference on μ, β, λ, ξ and η based on the discrete returns $\{y_t\}$. Of course this is difficult due to both the discretisation and the fact that we only partially observe the system.

(iii) *Analysis of discretised model*

In this paper we will only look at the problem of partial observation. It suffices for our purposes to note that the methods we look at are helpful at additionally tackling the

discretisation issue too. This is discussed at some length in Elerian *et al.* (2001). For now we focus on an Euler scheme approximation to the continuous time SV system (4.4). Then returns will be written as

$$\begin{aligned} y_t &= \mu + \beta \exp(\alpha_t) + e^{\alpha_t/2}\varepsilon_t \,,\, t \geq 1 \\ \alpha_{t+1} &= \xi + \phi(\alpha_t - \xi) + \sigma_\eta \eta_t \,,\, t \geq 2 \\ \alpha_1 &\sim \mathcal{N}\left(\xi, \sigma_\eta^2 / \left(1-\phi^2\right)\right). \end{aligned} \tag{4.6}$$

Then the likelihood function is just, writing $y = y_1, ..., y_T$ and $\alpha = \alpha_1, ..., \alpha_T$,

$$f(y) = \int f(y|\alpha) f(\alpha) d\alpha,$$

a T dimensional integral, which we do not know how to solve analytically. In practice T is almost always beyond 1000 and is often much larger and we have to use simulation to approximate $f(y)$. Importance sampling is used to deal with it (see Marshall (1956) and Liu (2001, Ch. 2)). An importance sampling density $g(\alpha|y)$ is introduced which is both easy to evaluate and simulate from. Then $f(y)$ is approximated by

$$\widehat{f}(y) = \frac{1}{R}\sum_{j=1}^{R} w_j, \qquad \text{where} \qquad w_j = \frac{f(y|\alpha^j) f(\alpha^j)}{g(\alpha^j|y)}, \tag{4.7}$$

and

$$\alpha^j \overset{i.i.d.}{\sim} g(\alpha|y),$$

with $g(\alpha|y)$ positive for all $\alpha \in R^T$. By construction we know that $\{w_j > 0\}$ are i.i.d. and that $E(w_j) = f(y)$. As a result, a simple application of Kolmogorov's strong law of large numbers (e.g. Geweke (1989, p. 1320) and Gallant (1997, p. 132)) shows that

$$\widehat{f}(y) \overset{a.s.}{\to} f(y), \qquad \text{as } R \to \infty,$$

whatever importance sampler we design.

Of course, the choice of the sampler $g(\alpha|y)$ will determine the efficiency of the method in practice. We design our importance sampler by using a Laplace approximation to the posterior of $\alpha_1, \ldots, \alpha_T$ given the data $y_1, \ldots, y_T$; see, for example, Gelman, Carlin, Stern and Rubin (1995, p.306). This means the proposals will be T dimensional Gaussian variables. The details of how such an approximation for SV type models is obtained quickly is given in Shephard and Pitt (1997) and Durbin and Koopman (2001, Chapter 10). The code for carrying out the computing is available at ??.

(iv) *Distributed computing and SV model estimation*

Each of the R samples from the sampler is T dimensional. We need R to be quite large for this to be a reasonably precise estimator of the true likelihood, which means each function evaluation of $\widehat{f}(y)$ is quite expensive (common random numbers are used in carrying out the computing and so $\widehat{f}(y)$ is smooth with respect to the parameters). This function has then to be maximised with respect to the parameters μ, β, ξ, ϕ and σ_η. This is quite a computationally demanding task.

We can use a distributed computing architecture when computing the numerical derivatives of $\widehat{f}(y)$ for this task requires around 10 function evaluations of $\widehat{f}(y)$ at different places in the parameter space. This load could be distributed over a number of computers.

(*c*) *Calibrating Gets by Monte Carlo experiments*

Gets denotes **ge**neral-**t**o-**s**imple methods for model selection. A three-stage process of specification, evaluation and selection is used:

A. Specification of the general unrestricted model (GUM).

B. Mis-specification testing of that GUM.

C. Selection of a specific model by simplifying the GUM:

(a) pre-search simplification of the GUM;

(b) multi-path selection of a specific model;

(c) post-search evaluation of that selected model.

At stage A., the GUM will be based on subject-matter considerations, but many aspects have to be data based, such as lag reactions, the precise set of relevant variables from a candidate list, and functional forms. Thus, the GUM is expected to nest the local approximation to the process generating the data, and can be large.

At stage B., there are five aspects needing decisions:

(a) the choice of the null hypotheses to be tested;

(b) the choice of the test types to be used;

(c) the choice of the parameters of tests;

(d) the choice of the critical values for each of the tests;

(e) the 'calibration' of the test sizes and powers.

An econometric model is defined to be congruent if it matches the evidence on all evaluated dimensions. The theory of model reduction in Hendry (1995) develops the theoretical basis for this appraoch, and provides excess corroborated content through a taxonomy of evaluation information which is an exhaustive set of null hypotheses necessary to sustain any congruence claim against past, present and potential future data, rival explanations, data measurement and theory information. Tests of such null hypotheses have long existed, and are optimized against possible alternative hypotheses. For example, the hypothesis of white-noise errors might be tested against first-order autocorrelation (or e.g., fourth). The power of mis-specification tests to reject any given null depends on the alternative as well as the state of nature. Thus, such tests are more useful if data variables are themselves:
i] non-normal;
ii] heteroscedastic;
iii] autocorrelated;
iv] subject to breaks.
If any such tests reject the GUM, the investigator must rethink, and begin the study again.

If the GUM is congruent against all the directions evaluated, it is essential to simplify it. The basis is not the usual statistical criterion that 'parsimony is necessary for precision of estimation', but because economies are non-stationary. Retaining a 'spurious' variable which then shifts will precipitate forecast failure (see Clements and Hendry (1999)) and can be detrimental for economic policy. Consequently, model selection is essential.

However, there are risks involved in every statistical decision, and Gets selection is designed to deliver a model that is 'close' to the local approximation by maximizing the probability of retaining relevant variables subject to the pre-set risk of also retaining irrelevant variables (Type I error). The program in Hendry and Krolzig (2001) first simplifies by eliminating 'highly irrelevant' variables, then explores simplifications along all feasible paths (i.e., never deleting significant effects), subject to no reduction inducing a significant value for a diagnostic check that congruence is retained. All candidate models are collected and cross-evaluated by encompassing tests, and the process repeated till a unique outcome results (although all non-dominated contenders are reported).

Thus, the following are open to the choice of the investigator:

- the n diagnostic checks in the test battery;
- the parameters of these diagnostic tests;
- the ignificance levels η of the n diagnostics;
- 'outlier' removal settings;
- the pre-search F-test simplifications;
- the significance levels κ of such tests;
- the multi-path simplification tests (t and F);
- the significance levels α of simplification tests;
- the significance levels γ of encompassing tests;
- the information criterion for final selection;
- the sub-sample split;
- the significance levels δ of the sub-sample tests.

A huge range of Monte Carlo simulations was required to calibrate the operational characteristics of the program, across differing artificial DGPs, with different sample sizes, numbers of relevant regressors, different numbers of irrelevant regressors, and different critical values of most of the selection test procedures. The outcomes of these experiments were needed to calibrate the in-built search strategies, which we denote by Liberal (minimize the non-selection probablilities) and Conservative (minimize the non-deletion probabilities).

A typical DGP for $m = k + n + 1$ variables was:

$$y_t = \sum_{i=1}^{k} \beta_{i,0} x_{i,t} + u_t, \qquad u_t \sim \mathsf{IN}\left[0, 1\right], \tag{4.8}$$

$$\mathbf{x}_t = \gamma + \mathbf{\Gamma} \mathbf{x}_{t-1} + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathsf{IN}_{k+n}\left[\mathbf{0}, \mathbf{\Omega}_{k+n}\right]. \tag{4.9}$$

As shown, only k current-dated values of these generated variables entered the DGP with potentially non-zero coefficients. However, the GUM was:

$$y_t = \sum_{i=1}^{2} \rho_i y_{t-i} + \sum_{j=1}^{k+n} \sum_{i=1}^{2} \beta_{j,i} x_{j,t-i} + \delta + u_t. \tag{4.10}$$

The desired design comprised four different sample sizes (of $T = 100, 150, 200, 500$) with $k = 4,$ or 8 relevant and either 6 or 12 irrelevant variables, with the regressors in (4.8) having population t-values of 0, 2, 3, or 4; and static, stationary dynamic, near-non-stationary, and cointegrated stationary processes for (4.9) which generates 256 experiments for each setting of every selection-test significance level (320 in total). Because selection probabilities were to be estimated, a large number M of replications per experiment was desirable. The relative percentage standard error $\mathsf{SE}\left[\widehat{\mathsf{p}}\right]$ of an estimated probability $\widehat{\mathsf{p}}$ for $\mathsf{p} = \Pr\left(X > x\right)$ (say) based on accept/reject calculations is:

$$\frac{100\mathsf{SE}\left[\widehat{\mathsf{p}}\right]}{\mathsf{p}} = \sqrt{\frac{100^2\mathsf{p}\left(1-\mathsf{p}\right)}{M\mathsf{p}^2}} = \sqrt{\frac{10\left(1-0.05\right)}{0.05}} \simeq 14\%, \tag{4.11}$$

when $M = 1000$, falling to 4.4% for $M = 10000$ so the former would suffice but hardly offer precision. Overall, therefore, a daunting computational was confronted, namely $8\times10^{\,7}$ trials, on each of which up to $2^{45} \simeq 3 \times 10^{13}$ possible models might need to be explored (if based purely on 'model selection criteria'), having up to 45 variables. The total load could never to carried out on a PC.

5. Parallel Ox

Ox is a matrix-programming language developed by the first author. Ox has a comprehensive mathematical and statistical library, and, although it is a matrix language, a syntax that is similar to C and C++. Ox is a relatively young language, with a first official release in 1996 (Doornik, 1996). Despite this, it has been quite widely adopted in econometrics and statistics. Two contributing factors are that there is a free version for academics, and that it is fast. The latter is also relevant within the current context.

In common with other matrix-programming languages (such as Matlab, GAUSS, etc.), Ox is an interpreted language,† with a commensurate speed penalty for unvectorized code such as loops (although the speed penalty is considerably less than in other programs‡).

We hope to turn the fact that is interpreted into an advantage, because it gives an additional level of control. The following table lists the levels of parallelization we consider in this paper:

level 1:	Ox with parallization explicit in the Ox code,
level 2:	Ox with parallelized run-time library,
level 3:	Ox with parallelized interpreter.

In our case, the three levels refer to cluster-based computing, based on a group of PCs, networked together, and controlled by a master PC. In this setting we shall only use the message-passing interface (MPI, see ??). Note that level 1 is not very different from using MPI from Fortran or C programs.

In addition, we shall consider implementation in a symmetric-multiprocessing (SMP) environment, using PCs with two or four processors. Such a shared memory implementation will be quite different from MPI, based on multithreading instead. For this setting, we will only implement level two and three.

The next three sections consider the parallelization levels in more detail.

† More precisely, Ox compiles code into an intermediate language, which is then interpreted.

‡ ... reports a speed comparison of many matrix languages, of which Ox is the fastest on the adopted metric.

(*a*) *Ox with parallization explicit in the Ox code*

At this level, the only requirement is to make the MPI functionality directly callable from Ox. Because Ox can be extended through dynamic link libraries, this level can be achieved without changing Ox. We would expect optimal performance gains for embarrasingly-parallel problems, in which case the implentation costs should be reasonable. The drawback is that the resulting Ox program cannot run unmodified on a single workstation. Because most problems that we consider are embarrasingly-parallel or close to it, we consider this level as the performance frontier for the next two levels.

(*b*) *Ox with parallelized run-time library*

At the next level, we hide the parallelization in the run-time library. Operations such as matrix multiplication, inversion, etc., will be distributed across the available hardware. Here we use available libraries for the implementation (scalapack for clusters, and MKL for SMP). Functions which work on matrix elements (logarithm, loggamma function, etc.) are also distributed using MPI or multithreading. The benefits to the user are that no knowledge of MPI is required. The speed benefit will be dependent on the problem: if only small matrices are used, the communication overhead would prevent the effective use of the cluster.

(*c*) *Ox with parallelized interpreter*

This level is the most interesting, and, insofar we are aware, has not been tried successfully before. The basic idea is to run the interpreter on the master, handing elements of expressions to the slaves. The main problem arises when a computation requires a previous result – in that case the process stalls until the result is available. On the other hand, there may be subsequent computations that can already be done. To tackle this issue we introduce *database computing*: components of expressions are handed to a database, and a database 'manager' decides on the order of computations based on the requests for results it receives. At this level it may happen that no satisfactory speed-up results.

(*d*) *Deterministic computing*

One concern that we have is that we have a strong preference for deterministic computation: when the same program is run twice on identical hardware the same results should be obtained. This is relevant when random numbers are used, which is the case in all our applications. To achieve deterministic results, we assign a separate seed to each node, to be decided by the master.

NB: this may not quite solve the problem

6. Some applications

7. Conclusions

TBW

Acknowledgements

Financial support from the U.K. Economic and Social Research Council under grant R000233447 is gratefully acknowledged. The computations were performed using the Ox programming language.

References

Andersen, T. G., Benzoni, L., and Lund, J. (1999). Estimating jump-diffusions for equity returns. Kellogg Graduate School of Management, Northwestern University.

Andersen, T. G., and Lund, J. (1997). Estimating continuous-time stochastic volatility models of the short term interest rate. *Journal of Econometrics*, **2**, 343–77.

Bean, L. H. (1929). A simplified method of graphic curvilinear correlation. *Journal of the American Statistical Association*, **24**, 386–397.

Bollerslev, T., Engle, R. F., and Nelson, D. B. (1994). ARCH models. In Engle, R. F., and McFadden, D. L. (eds.) , *Handbook of Econometrics*, Vol. 4, Ch. 49. Amsterdam: North-Holland.

Chong, Y. Y., and Hendry, D. F. (1986). Econometric evaluation of linear macro-economic models. *Review of Economic Studies*, **53**, 671–690. Reprinted in Granger, C. W. J. (ed.) (1990), *Modelling Economic Series*. Oxford: Clarendon Press.

Clark, P. K. (1973). A subordinated stochastic process model with fixed variance for speculative prices. *Econometrica*, **41**, 135–156.

Clements, M. P., and Hendry, D. F. (1999). *Forecasting Non-stationary Economic Time Series*. Cambridge, Mass.: MIT Press.

Danielsson, J. (1994). Stochastic volatility in asset prices: estimation with simulated maximum likelihood. *Journal of Econometrics*, **61**, 375–400.

Danielsson, J., and Richard, J. F. (1993). Accelerated Gaussian importance sampler with application to dynamic latent variable models. *Journal of Applied Econometrics*, **8**, S153–S174.

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/.

Doornik, J. A. (1999). *Object-Oriented Matrix Programming using Ox*. London: Timberlake Consultants Press. 3rd edition.

Durbin, J., and Koopman, S. J. (2001). *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press.

Durham, G., and Gallant, A. R. (2001). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes (with discussion). *Journal of Business and Economic Statistics*, **19**. Forthcoming.

Eisenpress, H., and Greenstadt, J. (1966). The estimation of non-linear econometric systems. *Econometrica*, **34**, 851–861.

Elerian, O., Chib, S., and Shephard, N. (2001). Likelihood inference for discretely observed non-linear diffusions. *Econometrica*, **69**, 959–993.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of the United Kingdom inflation. *Econometrica*, **50**, 987–1007.

Gallant, A. R., Hsu, C., and Tauchen, G. (1998). Calibrating volatility diffusions and extracting integrated volatility. Unpublished paper: Duke Economics Department.

Gallant, A. R., and Tauchen, G. (1996). Which moments to match. *Econometric Theory*, **12**, 657–81.

Gallant, A. R. (1997). *An Introduction to Econometric Theory*. Princeton: Princeton University Press.

Gallant, A. R., Hsieh, D., and Tauchen, G. (1997). Estimation of stochastic volatility models with diagnostics. *Journal of Econometrics*, **81**, 159–192.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian Data Analysis*. London: Chapman & Hall.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, **57**, 1317–39.

Ghysels, E., Harvey, A. C., and Renault, E. (1996). Stochastic volatility. In Rao, C. R., and Maddala, G. S. (eds.) , *Statistical Methods in Finance*, pp. 119–191. Amsterdam: North-Holland.

Gourieroux, C., Monfort, A., and Renault, E. (1993). Indirect inference. *Journal of Applied Econometrics*, **8**, S85–S118.

Hajivassiliou, V., and McFadden, D. (1998). The method of simulated scores for the estimation of ldv models. *Econometrica*, **66**, 863–896.

Hajivassiliou, V. A., and Ruud, P. A. (1994). Classical estimation methods for LDV models using simulation. In Engle, R. F., and McFadden, D. L. (eds.) , *Handbook of Econometrics, Volume 4*, pp. 2383–2441. Amsterdam: North-Holland.

Hargrove, W. M., Hoffman, F. M., and Sterling, T. (2001). The do it yourself supercomputer. *Scientific American*, **265**, 62–69.

Hendry, D. F. (1976). The structure of simultaneous equations estimators. *Journal of Econometrics*, **4**, 51–88. Reprinted in Hendry, D. F., *Econometrics: Alchemy or Science?* Oxford: Blackwell Publishers, 1993, and Oxford University Press, 2000.

Hendry, D. F. (1984). Monte Carlo experimentation in econometrics. In Griliches, Z., and Intriligator, M. D. (eds.) , *Handbook of Econometrics*, Vol. 2–3, Ch. 16. Amsterdam: North-Holland.

Hendry, D. F. (1995). *Dynamic Econometrics*. Oxford: Oxford University Press.

Hendry, D. F. (2001). How economists forecast. In Hendry, D. F., and Ericsson, N. R. (eds.) , *Understanding Economic Forecasts*, pp. 15–41. Cambridge, Mass.: MIT Press.

Hendry, D. F., and Krolzig, H.-M. (2001). *Automatic Econometric Model Selection*. London: Timberlake Consultants Press.

Jacquier, E., Polson, N. G., and Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models (with discussion). *Journal of Business and Economic Statistics*, **12**, 371–417.

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. *Review of Economic Studies*, **65**, 361–393.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York: Springer.

Marshall, A. (1956). The use of multi-stage sampling schemes in monte carlo computations. In Meyer, M. (ed.) , *Symposium on Monte Carlo Methods*, pp. 123–140. New York: Wiley.

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, **57**, 995–1026.

Phillips, P. C. B. (1985). The ET interview: Professor J.D. Sargan. *Econometric Theory*, **2**, 119–139.

Sandmann, G., and Koopman, S. J. (1998). Estimation of stochastic volatility models via Monte Carlo maximum likelihood. *Journal of Econometrics*, **87**, 271–301.

Shephard, N. (1996). Statistical aspects of ARCH and stochastic volatility. In Cox, D. R., Hinkley, D. V., and Barndorff-Nielsen, O. E. (eds.) , *Time Series Models in Econometrics, Finance and Other Fields*, pp. 1–67. London: Chapman & Hall.

Shephard, N., and Pitt, M. K. (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, **84**, 653–67.

Smith, A. A. (1993). Estimating nonlinear time series models using simulated vector autoregressions. *Journal of Applied Econometrics*, **8**, S63–S84.

Taylor, S. J. (1982). Financial returns modelled by the product of two stochastic processes — a study of daily sugar prices 1961-79. In Anderson, O. D. (ed.) , *Time Series Analysis: Theory and Practice, 1*, pp. 203–226. Amsterdam: North-Holland.