Fast Indirect Robust Generalized Method of Moments

Sébastien Loisel[a], Marina Takane[b]

[a]*Department of Mathematics, 1805 North Broad Street, Temple University, Philadelphia, PA 19122, loisel@temple.edu.*
[b]*1003 Christian Street #2, Philadelphia, PA 19147, mtakane@gmail.com.*

Abstract

The Robust Robust Generalized Methods of Moments (RGMM) and the Indirect Robust GMM (IRGMM) are algorithms for estimating parameter values in statistical models, such as diffusion models for interest rates, in a robust way. The long computation time is one of the main challenge facing these methods. In this paper, we introduce accelerated variants of RGMM and IRGMM. The fixed point iteration in RGMM is accelerated using minimal polynomial extrapolation, and the simulation of pseudo-observations in IRGMM is sped up by using a higher order stochastic Runge-Kutta method. We illustrate the fast performance of these algorithms for an interest rate diffusion model on four datasets.

1. Introduction

A commonly studied problem in finance is modeling of the dynamics of the short-term riskless interest rate, often using the following stochastic differential equation (SDE):

$$dY = \mu(t, Y)dt + \sigma(t, Y)dW, \tag{1}$$

where $Y(t)$ is the modeled interest rate and $W(t)$ is a standard Wiener process. Models with various drift $\mu(t, Y)$ and volatility $\sigma(t, Y)$ functions have been proposed; the landmark study by Chan, Karolyi, Longstaff, and Sanders (CKLS, 1992) estimated parameters for nine such models using the Generalized Method of Moments (GMM) and compared them within a framework of nested models. However, a study by Dell'Aquila, Ronchetti and Trojani (2003) showed that the CKLS parameter estimates and test results were highly unstable, and depended crucially on a small number of influential observations. In particular, their sensitivity experiments showed that a moderate change in just a single observation resulted in the classical Hansen test being unable to distinguish between two competing models, demonstrating the importance of using robust

☆The authors would like to thank Elvezio Ronchetti and Veronika Czellar for their helpful comments.

estimation. Ronchetti and Trojani (2001) have developed a robustified version of GMM (RGMM), in which the influence function is bounded.

Due to the inherent complexity of financial phenomena and the influence of external factors, interest rate observations are almost certain to contain deviations from any financial model. Many of the classical methods commonly used in finance, such as GMM and Maximum Likelihood (ML), are prone to yielding biased results due to these model deviations. The advantage of robust methods is that they are stable in the presence of outliers; that is, robust methods tolerate situations where the model does not fit the data perfectly. One of the primary complaints concerning the practicability of robust methods such as RGMM is that they converge slowly. The current paper addresses these concerns by presenting accelerated versions of these algorithms.

1.1. The model problem

The interest rate model proposed by Brennan and Schwartz (1977) has three real parameters α, β and σ, and is written:

$$dY = (\alpha + \beta Y)dt + \sigma Y dW. \tag{2}$$

Although somewhat simpler than the CKLS model, it is a good SDE with which to demonstrate our acceleration techniques.

To simplify parameter estimation and analysis, SDEs are commonly discretized and solved using a Forward Euler scheme, which is the simplest Stochastic Runge-Kutta (SRK) solver (Burrage and Burrage, 1996). If Y_n is the approximation of the solution $Y(t)$ at time $t = t_n$, then the discretization of the SDE (2) can be written as

$$Y_{n+1} - Y_n = (\alpha + \beta Y_n)\Delta t + \sigma Y_n \omega_n, \tag{3}$$

where $\Delta t = t_{n+1} - t_n$, and $\omega_n = W_{n+1} - W_n \sim \mathcal{N}(0, \Delta t)$ are iid. Unfortunately, parameters estimated based on the discretized equation are biased. Indirect inference (Gouriéroux, Monfort & Renault, 1993) has been proposed as a method to correct this discretization bias. Czellar, Karolyi & Ronchetti (2007) applied Indirect Robust GMM (IRGMM) to three diffusion models and found that goodness-of-fit was improved, particularly for models with non-linear drift functions. Genton and Ronchetti (2003) have shown that using a robust auxiliary estimator confers robustness on its indirect estimator, and that IRGMM also corrects for bias due to model misspecification.

The IRGMM algorithm proceeds in two steps. In the first step, an RGMM estimation is performed on the real data. In the second step, a simplified RGMM algorithm runs iteratively, incorporating information from pseudo-observations to correct for model discretization bias. In the RGMM step, the essential numerical algorithm is a fixed point iteration which in practice can take many hours to complete. Our first contribution is to accelerate this iteration using the Minimum Polynomial Extrapolation (MPE) method for accelerating the convergence of vector sequences. (See Smith, Ford & Sidi (1987) for a review.) In the second step, a performance bottleneck is the simulation of large numbers of pseudo-observations from a given set of parameters. These pseudo-observations are generated by solving numerically the SDE (2). In order for the indirect inference to correct for bias, these pseudo-observations should be as free of discretization error as possible.

There are two main methods for obtaining higher precision solutions of an SDE: reducing the step size Δt, or increasing the order p of the stochastic Runge-Kutta method (Burrage and Burrage, 1996). In Czellar et al. (2007), the pseudo-observations are obtained using the Euler method, but with a step size of $\Delta t = 1/22$ instead of $\Delta t = 1$ (i.e. 22 times finer). This increases computation time enormously; however, since the Euler method has strong order 1/2, the pathwise errors are only improved by a factor of $\sqrt{22} \approx 4.69$, that is, by less than one digit. By using a higher order Runge-Kutta method, we are able to use a coarser fine grid and obtain similar results in less time.

This paper is organized as follows: Section 2 summarizes the GMM, RGMM and IRGMM parameter estimation algorithms. Section 3 introduces the new algorithms. Section 4 discusses our numerical results. We end with some concluding remarks in Section 5.

2. GMM, RGMM and IRGMM

In this section, we briefly recall the GMM (Hansen, 1982), RGMM (Ronchetti and Trojani, 2001) and IRGMM (Genton and Ronchetti, 2003; Czellar et al., 2007) parameter estimation techniques which compute values of α, β and σ in the SDE (2). Let Z_n denote the random variable for the interest rate at time t_n, and let $\mathbf{z} = [z_1, z_2, \ldots, z_N]$ be the interest rate data observed at regularly spaced time points $t = t_1, \ldots, t_N$ with interval Δt. Let $Y(t)$ denote the solution of the SDE (2), and let the symbols $Y_1, ..., Y_N$ denote the solution of the discretized equation (3). Our goal is to estimate the vector $\boldsymbol{\theta} = (\alpha, \beta, \sigma) \in \boldsymbol{\Theta} \subset \mathbb{R}^3$ parametrizing the SDE (2) using the given data $\mathbf{z}$.

Consider the residuals ε_n computed from the data,

$$\varepsilon_n = z_{n+1} - z_n - (\alpha + \beta z_n)\Delta t. \tag{4}$$

Note that if the interest rates Z_n were indeed generated from the discretized SDE (3), then we would expect ε_n to be iid and to follow a $\mathcal{N}(0, \Delta t \sigma^2 z_n^2)$ distribution.[1]

2.1. The Generalized Method of Moments (GMM)

GMM is an estimation method commonly used in econometrics. A set of orthogonality conditions $h(\boldsymbol{\theta}) = h(\boldsymbol{\theta}, Z_n, \varepsilon_n)$ is imposed such that $E[h(\boldsymbol{\theta}, Z_n, \varepsilon_n)] = \mathbf{0}$ if $\boldsymbol{\theta}$ are the true parameter values. It is usually not possible to choose $\boldsymbol{\theta}$ so

[1]We abuse the notation and use the same symbol ε_n for both the random variable, and the observation.

that $h(\boldsymbol{\theta}) = \mathbf{0}$, and so instead we minimize some norm of $h(\boldsymbol{\theta})$. Following CKLS (1992), we use the orthogonality conditions

$$h(\boldsymbol{\theta}, Z_n, \varepsilon_n) = \begin{bmatrix} \varepsilon_n \\ \varepsilon_n Z_n \\ \varepsilon_n^2 - \sigma^2 Z_n^2 \\ (\varepsilon_n^2 - \sigma^2 Z_n^2) Z_n \end{bmatrix}. \tag{5}$$

Let $g(\boldsymbol{\theta}, \mathbf{z}) = \frac{1}{N-1} \sum_{n=1}^{N-1} h(\boldsymbol{\theta}, z_n, \varepsilon_n)$ be the sample mean of $h(\boldsymbol{\theta}, Z_n, \varepsilon_n)$ computed on the data $\mathbf{z} = [z_1, z_2, \ldots, z_N]$. Iterative GMM finds the $\hat{\boldsymbol{\theta}}$ that minimizes the quadratic form $J(\boldsymbol{\theta}, \mathbf{z}) = g(\boldsymbol{\theta}, \mathbf{z})^T\, \mathbf{W}\, g(\boldsymbol{\theta}, \mathbf{z})$, where $\mathbf{W}$ is a symmetric positive definite weighting matrix.

In practice, $\hat{\boldsymbol{\theta}}$ and $\mathbf{W}$ are computed by an iterative process. We start with an initial value of $\mathbf{W}$ such as $\hat{\mathbf{W}}_0 = \mathbf{I}$, the identity matrix, and minimize $J(\boldsymbol{\theta}, \mathbf{z})$ as a function of $\boldsymbol{\theta}$ to obtain $\hat{\boldsymbol{\theta}}_0$. Given current values $\hat{\mathbf{W}}_i$ and $\hat{\boldsymbol{\theta}}_i$, the succeeding values in the iteration are computed using

$$\hat{\mathbf{W}}_{i+1} = \left(\frac{1}{N-1} \sum_{n=1}^{N-1} h(\hat{\boldsymbol{\theta}}_i, z_n, \varepsilon_n)\, h(\hat{\boldsymbol{\theta}}_i, z_n, \varepsilon_n)^T \right)^{-1}, \tag{6}$$

and

$$\hat{\boldsymbol{\theta}}_{i+1} = \operatorname*{argmin}_{\boldsymbol{\theta}} g(\boldsymbol{\theta}, \mathbf{z})^T\, \hat{\mathbf{W}}_{i+1}\, g(\boldsymbol{\theta}, \mathbf{z}). \tag{7}$$

Definition 1 (Iterative Generalized Method of Moments)**.** *Given* $\hat{\mathbf{W}}_0$ *and* $\hat{\boldsymbol{\theta}}_0$*, the iterated Generalized Method of Moments (GMM) procedure is the fixed point iteration*

$$(\hat{\mathbf{W}}_{i+1}, \hat{\boldsymbol{\theta}}_{i+1}) = f_{GMM}(\hat{\mathbf{W}}_i, \hat{\boldsymbol{\theta}}_i), \quad i = 0, 1, \ldots;$$

as defined by equations (6) and (7). This process is iterated until convergence.

2.2. Robust GMM

The GMM estimator is not robust, and if the data $\mathbf{z}$ do not fit the model (3), then the estimated parameters can be highly influenced by outliers. A robust method is one which is able to fit the parameters in a stable way provided that the departure from the model is not too large. A GMM estimator is robust if and only if the orthogonality conditions are bounded (Ronchetti and Trojani, 2001; Dell'Aquila et al., 2003). Ronchetti and Trojani construct a robustified GMM estimator by truncating the orthogonality conditions, and computing multiplicative and additive factors $\mathbf{A}$ and $\boldsymbol{\tau}$ to maintain consistency. The truncated orthogonality conditions $h_c^{\mathbf{A},\boldsymbol{\tau}}(\boldsymbol{\theta}, Z_n, \varepsilon_n)$ are defined in terms of the Huber function $\mathscr{H}_c(\mathbf{x}) = \mathbf{x}\, w_c(\mathbf{x})$, where

$$w_c(\mathbf{x}) = \begin{cases} \min(1, \frac{c}{\|\mathbf{x}\|}) & \text{for} \quad \mathbf{x} \neq \mathbf{0} \\ 1 & \qquad\quad\; \mathbf{x} = \mathbf{0}. \end{cases} \tag{8}$$

Then,

$$h_c^{\mathbf{A},\boldsymbol{\tau}}(\boldsymbol{\theta}, Z_n, \varepsilon_n) = \mathscr{H}_c(\mathbf{A}[h(\boldsymbol{\theta}, Z_n, \varepsilon_n) - \boldsymbol{\tau}])$$
$$= \mathbf{A}[h(\boldsymbol{\theta}, Z_n, \varepsilon_n) - \boldsymbol{\tau}]\, w_c(\mathbf{A}[h(\boldsymbol{\theta}, Z_n, \varepsilon_n) - \boldsymbol{\tau}])$$

is bounded by definition. In practice, $\mathbf{A}$, $\boldsymbol{\tau}$, and the RGMM estimator $\hat{\boldsymbol{\theta}}_{RGMM}$ are computed in an iterative process. We start with an initial guess of the parameters $\hat{\boldsymbol{\theta}}_0$ (for instance, $\hat{\boldsymbol{\theta}}_{GMM}$) and initial values $\boldsymbol{\tau}_0 = \mathbf{0}$ and $\mathbf{A}_0$ such that

$$\mathbf{A}_0^T \mathbf{A}_0 = \left(\frac{1}{N-1} \sum_{n=1}^{N-1} h(\hat{\boldsymbol{\theta}}_0, z_n, \varepsilon_n)\, h(\hat{\boldsymbol{\theta}}_0, z_n, \varepsilon_n)^T \right)^{-1}. \tag{9}$$

Subsequent values of $\boldsymbol{\tau}$ and $\mathbf{A}$ are computed as follows:

$$\mathbf{A}_{i+1}^T \mathbf{A}_{i+1} = \tag{10}$$
$$\left(\frac{1}{N-1} \sum_{n=1}^{N-1} [h(\hat{\boldsymbol{\theta}}_i, z_n, \varepsilon_n) - \boldsymbol{\tau}_i][h(\hat{\boldsymbol{\theta}}_i, z_n, \varepsilon_n) - \boldsymbol{\tau}_i]^T w_c^2(\mathbf{A}_i[h(\hat{\boldsymbol{\theta}}_i, z_n, \varepsilon_n) - \boldsymbol{\tau}_i]) \right)^{-1}$$

and

$$\boldsymbol{\tau}_{i+1} = \frac{E[h(\hat{\boldsymbol{\theta}}_i, Z_n, \varepsilon_n) w_c(\mathbf{A}_i[h(\hat{\boldsymbol{\theta}}_i, Z_n, \varepsilon_n) - \boldsymbol{\tau}_i])]}{E[w_c(\mathbf{A}_i[h(\hat{\boldsymbol{\theta}}_i, Z_n, \varepsilon_n) - \boldsymbol{\tau}_i])]}. \tag{11}$$

Equation (11) involves computing expected values under the model distribution. Following Dell'Aquila et al. (2003), a simulation procedure with 10000 normally distributed values of ε_n is used to compute this expected value. The parameter estimates $\hat{\boldsymbol{\theta}}_{i+1}$ are computed by using the GMM algorithm with the truncated orthogonality conditions $h_c^{\mathbf{A}_{i+1},\boldsymbol{\tau}_{i+1}}(\boldsymbol{\theta}, z_n, \varepsilon_n)$.

Definition 2 (Robust GMM)**.** *Given $\mathbf{A}_0, \boldsymbol{\tau}_0$, the Robust Generalized Method of Moments (RGMM) procedure is the fixed point iteration*

$$(\mathbf{A}_{i+1}, \boldsymbol{\tau}_{i+1}) = f_{RGMM}(\mathbf{A}_i, \boldsymbol{\tau}_i), \quad i = 0, 1, \ldots;$$

as defined by equations (10) and (11). This process is iterated until convergence.

Following Dell'Aquila et al. (2003) and Czellar et al. (2007), we take $c = 5.85$ for the tuning parameter.

2.3. Indirect Inference

Indirect inference is a method for estimating the parameters $\boldsymbol{\theta}$ which corrects for model bias due to discretization. This is achieved by introducing a simulation procedure. We suppose that there are two models, denoted $F_{\boldsymbol{\theta}}$ and $\tilde{F}_{\boldsymbol{\theta}}$, both parametrized by $\boldsymbol{\theta}$. We suppose that $F_{\boldsymbol{\theta}}$ is the *true model*, while $\tilde{F}_{\boldsymbol{\theta}}$ is the *auxiliary model*, for which the parameter inference problem is easier. In our model problem, $F_{\boldsymbol{\theta}}$ is the SDE (2), while $\tilde{F}_{\boldsymbol{\theta}}$ is its Euler discretization (3). Although the parameter inference problem for the true model is difficult, we assume that it is easy to simulate the true model. In our case, it is easy to simulate the true model with high precision, given parameter values $\boldsymbol{\theta}$, using either a small stepsize in the Euler step, or using a higher order Stochastic Runge-Kutta method.

Let $\tilde{\boldsymbol{\mu}}$ be the auxiliary parameter estimated from observations $\mathbf{z}$ using the auxiliary model $\tilde{F}_{\boldsymbol{\mu}}$. Given $\hat{\boldsymbol{\theta}}$, suppose we simulate S sets of pseudo-observations

$$\mathbf{Z}^*(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} \mathbf{z}^{*1}(\hat{\boldsymbol{\theta}}) \\ \mathbf{z}^{*2}(\hat{\boldsymbol{\theta}}) \\ \vdots \\ \mathbf{z}^{*S}(\hat{\boldsymbol{\theta}}) \end{bmatrix},$$

using the true model $F_{\hat{\boldsymbol{\theta}}}$, and compute the auxiliary parameter $\boldsymbol{\mu}^*(\hat{\boldsymbol{\theta}})$ from these pseudo-observations with the auxiliary model $\tilde{F}_{\boldsymbol{\mu}}$. The indirect estimate $\hat{\boldsymbol{\theta}}_I$ is the parameter value $\hat{\boldsymbol{\theta}}$ such that $\boldsymbol{\mu}^*(\hat{\boldsymbol{\theta}})$ is as close as possible to $\tilde{\boldsymbol{\mu}}$. See Gouriéroux et al. (1993), Genton and Ronchetti (2003) and Czellar et al. (2007) for more detailed descriptions of indirect inference and IRGMM.

Definition 3 (Indirect Inference)**.** *Given the auxiliary parameter $\tilde{\boldsymbol{\mu}}$ and the simulation procedure $\hat{\boldsymbol{\theta}} \mapsto \boldsymbol{\mu}^*(\hat{\boldsymbol{\theta}})$, as above, the Indirect Inference estimate $\hat{\boldsymbol{\theta}}_I$ is given by*

$$\hat{\boldsymbol{\theta}}_I = \underset{\hat{\boldsymbol{\theta}}}{\operatorname{argmin}} \left(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}^*(\hat{\boldsymbol{\theta}})\right)^T \boldsymbol{\Omega} \left(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}^*(\hat{\boldsymbol{\theta}})\right), \tag{12}$$

where $\boldsymbol{\Omega}$ is a symmetric positive definite weighting matrix.

To generate S sets of pseudo-observations $\mathbf{Z}^*(\hat{\boldsymbol{\theta}})$, Czellar et al. (2007) used a fine discretization of the model SDE; that is, Forward Euler was used to compute a trajectory using Equation (3) with $\Delta t = 1/22$. Equation (3) with $\Delta t = 1$, called the crude discretization model, is used as the auxiliary model. The values $S = 100$ and $\Delta t = 1/22$ were carefully chosen based on numerical experiments in Czellar et al. (2007).

3. Fast (I)(R)GMM

We now discuss a Fast GMM (FGMM) algorithm, a Fast RGMM (or FRGMM) algorithm and a Fast IRGMM (FIRGMM) algorithm. Our two basic tools are vector sequence convergence acceleration and higher order stochastic Runge-Kutta methods.

3.1. FGMM and FRGMM

Consider a fixed point iteration $\mathbf{x}_{i+1} = f(\mathbf{x}_i)$, where f is a (possibly non-linear) function which takes a long time to evaluate. As noted in Definitions 1 and 2, the GMM and RGMM algorithms consist of fixed point iterations

$$(\hat{\mathbf{W}}_{i+1}, \hat{\boldsymbol{\theta}}_{i+1}) = f_{GMM}(\hat{\mathbf{W}}_i, \hat{\boldsymbol{\theta}}_i) \text{ and}$$
$$(\mathbf{A}_{i+1}, \boldsymbol{\tau}_{i+1}) = f_{RGMM}(\mathbf{A}_i, \boldsymbol{\tau}_i).$$

Our approach is to accelerate the convergence of these fixed point iterations using the Minimum Polynomial Extrapolation algorithm (MPE). We briefly recall this algorithm and refer to Smith et al. (1987) for details.

A convergence acceleration technique is one that uses the iterates $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to produce an improved estimate $\mathbf{s}$ of the limit $\mathbf{x}_\infty$ of the iteration. For scalar sequences, there is the well-known Δ^2 process of Aitken (1926). The Minimum Polynomial Extrapolation (MPE) algorithm is a similar technique that works for vector sequences. Given a matrix A, the minimum polynomial $p(x)$ is the unique monic polynomial of least degree with scalar coefficients, such that $p(A) = 0$. If $p(x) = a_0 + a_1 x + \ldots + x^n$, then we have that

$$a_0 I = A \overbrace{(-a_1 - a_2 A - \ldots - A^{n-1})}^{q(A)},$$

and so $q(A)/a_0$ is the inverse of A. The idea of MPE is to find the fixed point of the iteration

$$\mathbf{x}_{i+1} = A\mathbf{x}_i + \mathbf{b},$$

by using such a minimal polynomial formula to compute the solution

$$\mathbf{x}_\infty = (I - A)^{-1}\mathbf{b},$$

using the iterates $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We summarize the algorithm here, and refer to Smith et al. (1987) for a development and for convergence results.

Definition 4 (Minimum Polynomial Extrapolation)**.** *Let*

$$U = [\mathbf{x}_2 - \mathbf{x}_1,\ \mathbf{x}_3 - \mathbf{x}_2,\ \ldots,\ \mathbf{x}_{n-1} - \mathbf{x}_{n-2}],$$

be a matrix with m rows, and let $\mathbf{c} = U^+(\mathbf{x}_n - \mathbf{x}_{n-1})$, where U^+ denotes the Moore-Penrose pseudo-inverse of U. The accelerated iterate $\mathbf{s} = (s_1, \ldots, s_m)^T = MPE(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is defined by $s_j = (\sum_{\ell=2}^{n} x_{j\ell} c_{\ell-1}) / \sum_{\ell=1}^{n-1} c_\ell$ for $j = 1, \ldots, m$; where we have defined $c_{n-1} = 1$.

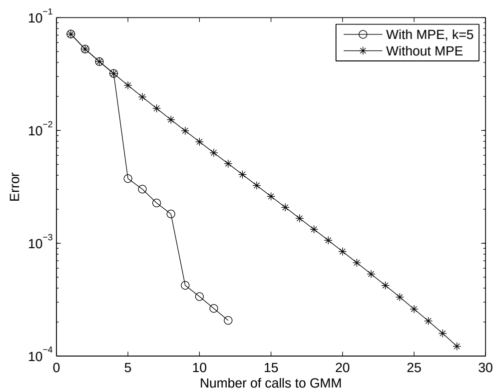
In this article, we have used $n = 5$, and so the MPE function is called every 5th iteration to obtain an accelerated iterate based on the 5 most recent steps.[2]

3.2. Numerical comparison of the convergence of RGMM and FRGMM

In Figure 1, the error $\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}$ of the parameters (α, β, σ) has been plotted for a call to RGMM and FRGMM for the LIBOR Dataset, as described in Section 4. Here, $\hat{\boldsymbol{\theta}}_i$ are the parameters at step i, and $\hat{\boldsymbol{\theta}}$ are the parameters at convergence of the RGMM algorithm. We observe that the MPE convergence acceleration is effective and that the accelerated algorithm converges in many fewer steps than the unaccelerated algorithm. The convergence may be quadratic (Smith et al., 1987), but the algorithm converged before this could become apparent.

[2] It is possible for MPE to fail to give an iterate with a smaller residual. The usual solution is to reject such MPE steps and increase n. Since $n = 5$ worked well with our data, we did not implement this “automatic” scheme.

Figure 1: Convergence graph for RGMM with and without MPE.



3.3. FIRGMM

For the IRGMM algorithm, the bottleneck is the high precision simulation of pseudo-observations (see Definition 3) drawn from the SDE (2). In Czellar et al. (2007), these "fine discretization" pseudo-observations are generated by using the same Euler method as with the coarse solver, but with a smaller time step. In order to obtain higher precision pseudo-observation without expending too much computation time, we have used the 4-stage stochastic Runge-Kutta method, which is of weak and strong order $p = 1.5$ (Burrage and Burrage, 1996).

Definition 5 (RK4 and SRK4 methods)**.** *The 4-stage Runge-Kutta (RK4) method for the ordinary differential equation $dy/dt = f(t, y)$ is*

$$y_{n+1} = y_n + \frac{1}{6}(RK_1 + 2RK_2 + 2RK_3 + RK_4), \tag{13}$$

where

$$RK_1 = \Delta t f(t_n, y_n), \tag{14}$$

$$RK_2 = \Delta t f(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}RK_1), \tag{15}$$

$$RK_3 = \Delta t f(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}RK_2), \tag{16}$$

$$RK_4 = \Delta t f(t_{n+1}, y_n + RK_3). \tag{17}$$

RK4 extends to a Stochastic Runge-Kutta method (SRK4) to solve the SDE (1). We formally replace $f(t, y)$ in the equations (14)–(17) with $f(t, y) = \mu(t, y) - \frac{1}{2}\sigma(t, y)\frac{\partial \sigma}{\partial y}(t, y) + \sigma(t, y)\Delta W/\Delta t$, where $\Delta W = W_n - W_{n-1}$ is the displacement of the Brownian motion between time t_{n-1} and t_n.

For the FIRGMM algorithm, the bottleneck is again the generation of pseudo-observations as per Definition 3. We generate the pseudo-observations using the SRK4 integrator with $\Delta t = 1/10$, as per Definition 5.

4. Numerical Experiments

We have already shown in Figure 1 that MPE provides a substantial speedup for the RGMM algorithm of Ronchetti and Trojani (2001). We now give numerical experiments on four datasets which show that that our algorithms are faster and equally accurate according to several goodness-of-fit measures. As the purpose of this paper is to give results pertaining to the fast RGMM and IRGMM algorithms, the reader is referred to Czellar et al. (2007), Dell'Aquila et al. (2003), Genton and Ronchetti (2003), and Ronchetti and Trojani (2001) for more thorough validations of the base algorithms.

4.1. The datasets

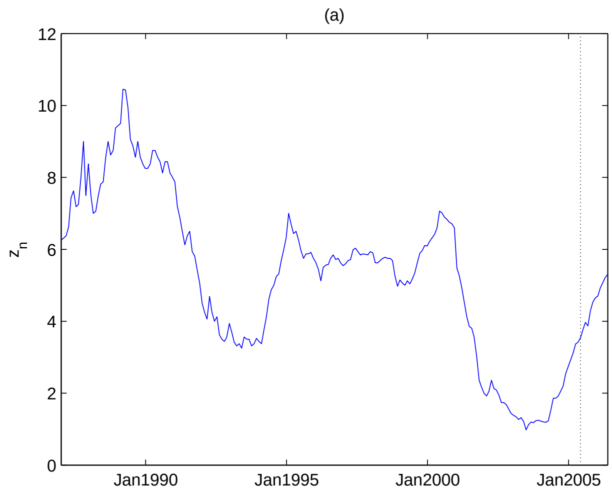
The estimation methods were tested on three sets of artificial data and one set of real data. Artificial data were generated using parameter values $\alpha = 0.25$, $\beta = -0.1$, and $\sigma = 0.01$ for Equation (2) using SRK4 with a stepsize of $\Delta t = 1$ and an initial value of $z_1 = 2.45$. Each dataset consists of 113 time points. The initial 101 time points were used for the training sample, and the remaining 12 time points were the out-of-sample test data. Normally distributed values of $\omega_n = \Delta W_n \sim \mathcal{N}(0, \Delta t)$ were used to generate Dataset 1, whereas 5% of the ω_n were perturbed for datasets 2 and 3. The contaminating distributions were $\mathcal{N}(0, 5^2\Delta t)$ (large standard deviation) and $\mathcal{N}(4, \Delta t)$ (large positive mean) for datasets 2 and 3, respectively. The fourth dataset was monthly London Interbank Offered Rate (LIBOR) data for the interest rate with a maturity of 1 month, in American dollars, from January 2, 1987 to May 26, 2006. It consisted of 222 data points in total. The first 210 were used to fit the parameters, and the last 12 were used to test predictions. The LIBOR data are plotted in Figure 2. More thorough statistical analyses of these data sets can be found in Takane (2008).

The mean, standard deviation, skewness, kurtosis, and the first five autocorrelations for the data z_n and the first order differences Δz_n are given in Table 1 for all datasets.

Table 1: Some descriptive statistics for the four datasets.

	Mean	SD	Skewness	Kurtosis	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5
Dataset 1									
z_n	2.4769	0.0414	-0.0755	2.5815	0.8806	0.7746	0.6817	0.5935	0.4708
Δz_n	0.0003	0.0202	-0.2044	2.7418	-0.0586	-0.0639	-0.0261	0.1419	-0.1680
Dataset 2									
z_n	2.4694	0.0648	0.2026	3.2221	0.8784	0.7631	0.6501	0.5519	0.4013
Δz_n	0.0000	0.0320	-1.5992	13.0036	-0.0294	-0.0116	-0.0610	0.2139	-0.1439
Dataset 3									
z_n	2.5305	0.0690	-0.1093	2.4910	0.8862	0.7833	0.6875	0.6105	0.5633
Δz_n	0.0006	0.0321	1.9693	9.3662	-0.0548	-0.0421	-0.0908	-0.1453	0.0743
LIBOR									
z_n	5.2277	2.2441	-0.0594	2.3858	0.9897	0.9766	0.9611	0.9422	0.9201
Δz_n	-0.0043	0.3154	-0.3781	6.3592	0.1401	0.1184	0.1594	0.1288	0.0293

Figure 2: Plot of the monthly LIBOR data. The sample used to fit parameters is to the left of the dotted line; data points used to test the model predictions are to the right side.



4.2. Goodness-of-fit measures

For each dataset, parameters were estimated on a sample, then goodness-of-fit measures were computed on within sample and out-of-sample data points. For comparison purposes, in addition to the robust algorithms RGMM and IRGMM, we include the non-robust algorithms ML and GMM. As in Czellar et al. (2007), three goodness-of-fit measures were computed to compare the performance of the different estimation methods: $\widehat{RAMSE}$, the root mean squared error; $\widehat{AMAD}$, the variability, and $\widehat{AMB}$, the average median bias. We generated $S = 10000$ simulated trajectories $\{[z_1^{(s)}, z_2^{(s)}, \ldots, z_N^{(s)}]\}_{s=1,\ldots,S}$ from the model Equation (2) using SRK4 with parameter estimates $\hat{\boldsymbol{\theta}}$ and $\Delta t = 1$. We define

$$\begin{aligned}\widehat{RAMSE} &= \left(\frac{1}{NS}\sum_{n=1}^{N}\sum_{s=1}^{S}(z_n^{(s)} - z_n)^2\right)^{\frac{1}{2}} \\ \widehat{AMAD} &= \frac{1}{N}\sum_{n=1}^{N}\operatorname*{median}_s(|\, z_n^{(s)} - \operatorname*{median}_l(z_n^{(l)})|) \\ \widehat{AMB} &= \frac{1}{N}\sum_{n=1}^{N}(|\operatorname*{median}_s(z_n^{(s)}) - z_n|).\end{aligned}$$

We also computed the goodness-of-fit measures with $\Delta t = 1/10$, but the differences in scores were not appreciable.

4.3. Experimental results

The results of our experiments, as well as running times, can be found in Table 3. When the data fit the model very closely, as in Dataset 1, the ML estimator performs very well and is very fast. However, when the data do not fit the model exactly, as in the other three datasets, then the robust techniques (such as IRGMM) tend to give very good results.

Since the fast algorithms FGMM, FRGMM and FIRGMM essentially give the same parameter estimates as the unaccelerated algorithms GMM, RGMM and IRGMM, we only give the goodness-of-fit measures of the fast algorithms (the unaccelerated algorithms have nearly the same goodness-of-fit measures, differing usually in the fourth decimal place). The exception is the LIBOR Dataset, for which IRGMM and FIRGMM gave different parameter estimates, and we list the goodness-of-fit measures for each one separately. We found that by changing the size of the time step for the indirect inference step of Definition 3, the optimization algorithm converges to different minima. This may be due to the presence of multiple local minima in the objective function. However, the focus of the current paper is on accelerating the existing parameter inference algorithms, so we did not further investigate this behavior. On these four datasets, the FGMM and FRGMM algorithms are always faster than the GMM and RGMM algorithms. This is because our MPE acceleration successfully reduces the number of iterations required to obtain convergence.

The running time listed in Table 3 for IRGMM and FIRGMM require a clarification. The (F)IRGMM algorithms start with an (F)RGMM calculation, followed by an indirect inference correction. The running time found in Table 3 is that of the indirect inference step. Hence, the total running time for FIRGMM (based on FRGMM) for Dataset 1 is $1016 + 125 = 1141$ seconds. The indirect inference step of FIRGMM is usually faster than that of IRGMM, and often converges in fewer iterations of the optimization function (we use MATLAB's `lsqnonlin`). In the case of Dataset 2, the FIRGMM was actually slower than IRGMM. However, on the LIBOR dataset, FIRGMM was over twice as fast as IRGMM.

Table 2: Estimated parameter values for Maximum Likelihood (ML), Generalized Method of Moments (GMM), Indirect GMM (IGMM), Robust GMM (RGMM), and Indirect Robust GMM (IRGMM), for the LIBOR data. The last column gives the mean to which the interest rate process (2) would revert for each method, given by $-\hat{\alpha}/\hat{\beta}$. The rows labeled "true" for the artificial datasets give the goodness-of-fit measures for the parameter values used to generate the data.

	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\sigma}$	$-\hat{\alpha}/\hat{\beta}$
ML	0.0285	-0.0081	0.0646	3.5354
(F)GMM	0.0393	-0.0086	0.0565	4.5789
(F)RGMM	0.0295	-0.0073	0.1273	4.0187
IRGMM	0.0940	-0.0134	0.0712	7.0030
FIRGMM	0.0866	-0.0197	0.0649	4.3875

Table 3: Goodness-of-fit measures computed on the true and estimated parameter values for the four datasets. For each dataset, and for each case (within-sample, and out-of-sample), the best (smallest) values for each measure are boxed. The "Time" columns denote the running time of the "Unaccelerated" algorithms (GMM, RGMM and IRGMM) compared to our new "Fast" algorithms (FGMM, FRGMM and FIRGMM), in seconds. The ML algorithm runs in a very small fraction of a second, which we denote by 0*. The running time for IRGMM and FIRGMM in this table does not include the running time of the preliminary RGMM or FRGMM calculation.

	Within sample			Out-of-sample			Time (s)	
	$\widehat{RAMSE}$	$\widehat{AMAD}$	$\widehat{AMB}$	$\widehat{RAMSE}$	$\widehat{AMAD}$	$\widehat{AMB}$	Unaccel.	Fast
Dataset 1								
True	0.0703	0.0363	0.0364	0.0480	0.0293	0.0148	–	–
ML	0.0588	0.0276	0.0342	0.0417	0.0229	0.0201	0*	–
(F)GMM	0.0637	0.0321	0.0344	0.0430	0.0238	0.0201	18.8	11.5
(F)RGMM	0.0790	0.0446	0.0345	0.0606	0.0368	0.0220	2347	1016
(F)IRGMM	0.0631	0.0317	0.0343	0.0448	0.0263	0.0173	133	125
Dataset 2								
True	0.0935	0.0441	0.0515	0.0588	0.0360	0.0202	–	–
ML	0.0935	0.0439	0.0515	0.0586	0.0358	0.0206	0*	–
(F)GMM	0.0945	0.0448	0.0515	0.0597	0.0363	0.0217	3.8	3.8
(F)RGMM	0.1055	0.0552	0.0508	0.0700	0.0446	0.0163	2348	1758
(F)IRGMM	0.0894	0.0401	0.0521	0.0519	0.0324	0.0138	111	140
Dataset 3								
True	0.0924	0.0363	0.0620	0.0495	0.0303	0.0154	–	–
ML	0.0932	0.0455	0.0536	0.0619	0.0383	0.0202	0*	–
(F)GMM	0.0954	0.0479	0.0533	0.0608	0.0372	0.0206	18.9	11.8
(F)RGMM	0.1236	0.0701	0.0555	0.0892	0.0577	0.0166	3531	2162
(F)IRGMM	0.0938	0.0456	0.0545	0.0610	0.0382	0.0184	467	298
Dataset 4 – LIBOR data								
ML	3.0652	1.1735	1.8158	1.2738	0.3443	1.0676	0*	–
(F)GMM	2.9547	1.1498	1.7259	1.1937	0.3040	1.0068	23	20
(F)RGMM	7.4696	1.5131	2.1760	1.5805	0.6507	1.1593	13135	5053
IRGMM	4.0570	1.4851	1.9311	1.0562	0.3891	0.8147	561	–
FIRGMM	2.5356	0.8329	1.8869	1.1664	0.3355	0.9659	–	254

For the three artificial datasets, the "true" parameter value is known, and so we include the goodness-of-fit measures for those parameter values in Table 3. The LIBOR Dataset comes from real data, so we do not have "true" parameters. The inferred parameters are in Table 2.

5. Conclusion

We have introduced fast versions of the RGMM and IRGMM parameter estimation algorithms and have shown that they are faster than the algorithms of Ronchetti and Trojani (2001) and Genton and Ronchetti (2003). We have also shown that FRGMM and FIRGMM work well on several artificial datasets as well as a LIBOR dataset.

Convergence acceleration such as MPE may be useful in other parameter estimation techniques that rely on a fixed point iteration, and higher order Runge-Kutta could also be used where increased precision leads to faster running times or more accurate parameter estimates.

References

Aitken, A., 1926. On Bernoulli's numerical solution of algebraic equations. Proceedings of the Royal Society of Edinburgh 46, 289–305.

Brennan, M. J., Schwartz, E. S., 1977. Saving bonds, retractable bonds, and callable bonds. Journal of Financial Economics 3, 133–155.

Burrage, K., Burrage, P., 1996. High Strong Order Explicit Runge-Kutta Methods for Stochastic Ordinary Differential Equations. Applied Numerical Mathematics: Transactions of IMACS 22 (1–3), 81–101.

Chan, K., Karolyi, G., Longstaff, F., Sanders, A., July 1992. An empirical comparison of alternative models of the short-term interest rate. The Journal of Finance 47 (3), 1209–1227.

Czellar, V., Karolyi, G., Ronchetti, E., 2007. Indirect robust estimation of the short-term interest rate process. Journal of Empirical Finance 14, 546–563.

Dell'Aquila, R., Ronchetti, E., Trojani, F., 2003. Robust GMM analysis of models for the short-rate process. Journal of Empirical Finance 10, 373–397.

Genton, M. G., Ronchetti, E., March 2003. Robust indirect inference. Journal of the American Statistical Association 98 (461), 67–76.

Gouriéroux, C., Monfort, A., Renault, E., 1993. Indirect inference. Journal of Applied Econometrics 8, S85–S118.

Hansen, L. P., 1982. Large sample properties of Generalized Method of Moments estimators. Econometrica, 1029–1054.

Ronchetti, E., Trojani, F., 2001. Robust Inference with GMM estimators. Journal of Econometrics 101, 37–69.

Smith, D. A., Ford, W. F., Sidi, A., June 1987. Extrapolation methods for vector sequences. SIAM Review 29 (2), 199–233.

Takane, M., June 2008. Robust Numerical Methods for Modeling Interest Rates, Master's Thesis, University of Geneva.