

УДК 519.766.4

М.М. Коновалюк

БАЙЄСІВСЬКИЙ АНАЛІЗ МОДЕЛІ СТОХАСТИЧНОЇ ВОЛАТИЛЬНОСТІ В СЕРЕДОВИЩІ OPENBUGS

This article illustrates Bayesian analysis of stochastic volatility models of daily dollar/hryvna exchange rates from 10/24/2006 to 4/15/2011, using a package of applied programs OpenBUGS. In addition, we determine that the estimation is completely accurate with acceptable computational cost as well as characterized by convergence at a given time gap.

Вступ

Модель стохастичної волатильності (МСВ), запропонована С. Тейлором у 1982 р. [1] та Г. Таученом і М. Піттсом у 1983 р. [2], використовується для опису фінансових часових рядів. Вона забезпечує реалістичніше та точніше моделювання фінансового часового ряду, ніж запропоновані Р. Інглом у [3] та Т. Болерслевом у [4] моделі типу авторегресії з умовною гетероскедастичністю (АРУГ). Це досягається завдяки тому, що вона включає два процеси білого шуму: один – для спостережуваних даних, інший – для прихованої волатильності. Спостереження мають волатильність через помилки вимірювання та помилки, притаманні процесу формування вибірки, а збурення самого досліджуваного процесу оцінюються варіацією основної динаміки волатильності. Переваги МСВ над моделями типу АРУГ розглянуто в [5].

Класичне оцінювання параметрів для МСВ – важке завдання внаслідок складного вигляду функції правдоподібності. Для оцінювання параметрів МСВ є ряд методів статистичного оцінювання, серед яких узагальнений метод моментів [6], метод квазімаксимальної правдоподібності [7], ефективний метод моментів [8], імітаційні методи максимальної правдоподібності [9] та апроксимації максимальної правдоподібності [10].

Байєсівський аналіз МСВ ускладнений через проблеми багатовимірного інтегрування при обчисленні апостеріорних ймовірностей. Ці труднощі подолано завдяки розвитку методу Монте-Карло для марковських ланцюгів (МКМЛ) протягом останніх десятиліть та збільшенню обчислювальних потужностей [11]. Процедури МКМЛ для МСВ запропоновано в [12].

Серед усіх згаданих методів МКМЛ є одним з найкращих інструментаріїв для обчислення оцінок стохастичної волатильності. Однак процедури МКМЛ вимагають складних об-

числень та непрості в реалізації, якщо не використовувати поширене програмне забезпечення.

МСВ можна легко реалізувати в BUGS, при цьому не вимагається, щоб програміст знав точні формули для будь-якої апіорної щільності або ймовірності.

Головна перевага BUGS полягає у простоті, з якою виконуються будь-які зміни у моделі, наприклад, вибір різних апіорних розподілів для параметрів, різні авторегресійні структури тощо. На відміну від BUGS, при реалізації алгоритму оцінювання МСВ на мовах низького рівня, наприклад на С, будь-які зміни потребують перепрограмування, що зазвичай вимагає значно більше часу для остаточного доведення програми.

Головна слабкість BUGS – надзвичайно слабка збіжність і неефективність з точки зору моделювання за одновимірним алгоритмом Гіббса. Внаслідок сильної апостеріорної кореляції між послідовними станами моделі волатильності в оновлених алгоритмах МКМЛ змішування даних відбувається дуже повільно [12].

Постановка задачі

Мета цієї статті полягає у тому, щоб проілюструвати процес виконання байєсівського аналізу моделей стохастичної волатильності, використовуючи пакет прикладних програм BUGS (Bayesian Analysis using Gibbs Sampling).

Поставлено завдання аналізу методів генерування псевдовипадкових послідовностей для розв'язання задачі оцінювання параметрів нелінійних моделей за методом Монте-Карло для марковських ланцюгів. Зокрема, необхідно оцінити значення волатильності стохастичного процесу та параметри моделі стохастичної волатильності для фінансового процесу, представленого статистичними даними щоденних обмінних курсів валют (долар/гривня) за вибраний період.

Метод Монте-Карло для марковських ланцюгів

Метод отримав назву Монте-Карло для марковських ланцюгів, тому що, використовуючи попередні значення вибірки для генерування випадкового наступного значення вибірки, генерується марковський ланцюг. Оскільки ймовірності переходу між значеннями вибірки є функцією лише попереднього значення, тобто виконується умова відсутності післядії, то при заданому теперішньому стані майбутній не залежить від минулих.

Нехай X_t – значення випадкової величини у момент часу t . Випадкова величина є марковською, якщо ймовірність переходу між різними значеннями у просторі станів залежить тільки від поточної випадкової величини, а саме:

$$\begin{aligned} \Pr(X_{t+1} = s_j \mid X_0 = s_k, \dots, X_t = s_i) = \\ = \Pr(X_{t+1} = s_j \mid X_t = s_i). \end{aligned}$$

Тому для прогнозування марковського процесу потрібна інформація тільки про поточний стан випадкової величини, а інформація про значення у попередні моменти часу не змінить ймовірність переходу.

При застосуванні методу Монте-Карло виникає проблема отримання вибірки з деякого багатовимірного розподілу ймовірностей $p(x)$. Вирішення цієї проблеми лежить в основі методів МКМЛ. Зокрема, були спроби інтегрувати дуже складні функції випадковим моделюванням. Результатом цих спроб є алгоритм Метрополіса–Хастінгса, який докладно розглянуто в [13].

Метод Монте-Карло для марковських ланцюгів – це набір алгоритмів, які реалізують ланцюг Маркова за певним правилом, серед них алгоритми Метрополіса, Метрополіса–Хастінгса, Гіббса.

Алгоритм Метрополіса. Нехай потрібно отримати вибірку з деякого розподілу $p(\theta)$, де $p(\theta) = f(\theta)/C$; C – нормуюча константа, яка може бути невідома. Алгоритм Метрополіса генерує послідовність випадкових вимірів з цього розподілу таким чином:

1. Вибираємо будь-яке початкове значення θ_0 , що задовольняє умову $f(\theta_0) > 0$.
2. Використовуючи поточне значення θ , генерується можливе значення θ^* з деякого розподілу $q(\theta_1, \theta_2)$ (його називають іноді стрибко-

подібним), що є ймовірністю отримання значення θ_2 при заданому попередньому значенні θ_1 . Цей розподіл називають ще розподілом для генерування кандидатів. Єдиним обмеженням на щільність стрибка в алгоритмі Метрополіса є симетричність $q(\theta_1, \theta_2) = q(\theta_2, \theta_1)$.

3. Маючи точку-кандидата θ^* , обчислюємо відношення щільності у точці-кандидаті (θ^*) та в поточній точці (θ_{t-1}):

$$\alpha = \frac{p(\theta^*)}{p(\theta_{t-1})} = \frac{f(\theta^*)}{f(\theta_{t-1})}.$$

Слід зауважити, що оскільки розглядається відношення $p(x)$ двох різних величин, то нормуюча константа C відсутня.

4. Якщо стрибок збільшує щільність ($\alpha > 1$), то приймаємо точку-кандидат ($\theta_t = \theta^*$) і повертаємося до кроку 2. Якщо стрибок зменшує значення щільності ($\alpha < 1$), то з ймовірністю α приймається точка-кандидат або з ймовірністю $1 - \alpha$ вона відхиляється і повертаємося до кроку 2.

Отже, алгоритм Метрополіса можна сформулювати так. Спочатку розраховуємо показник

$$\alpha = \min \left(\frac{f(\theta^*)}{f(\theta_{t-1})}, 1 \right),$$

після чого приймаємо точку-кандидат з ймовірністю α (ймовірність переходу). Так генерується ланцюг Маркова $(\theta_0, \theta_1, \dots, \theta_k, \dots)$ як ймовірність переходу зі стану θ_t до стану θ_{t+1} , що залежить тільки від значення θ_t та не залежить від $(\theta_0, \dots, \theta_{t-1})$. Через певний період ланцюг досягає стаціонарного розподілу та вибірка $(\theta_{k+1}, \dots, \theta_{k+n})$ береться як вибірка з розподілом $p(x)$.

Алгоритм Метрополіса–Хастінгса. Н. Хастінгс узагальнив алгоритм Метрополіса, використовуючи довільну функцію ймовірностей переходу $q(\theta_1, \theta_2) = \Pr(\theta_1 \rightarrow \theta_2)$ та встановивши ймовірність прийняття точки-кандидата таким чином:

$$\alpha = \min \left(\frac{f(\theta^*)q(\theta^*, \theta_{t-1})}{f(\theta_{t-1})q(\theta_{t-1}, \theta^*)}, 1 \right).$$

Це алгоритм Метрополіса–Хастінгса. Припустивши, що запропонований розподіл є симетричним $q(x, y) = q(y, x)$, Н. Хастінгс удосконалив алгоритм Метрополіса.

Алгоритм генерування вибірки за Гіббсом. Моделювання за Гіббсом є окремим випадком алгоритму Метрополіса–Хастінгса, де випадкова величина завжди приймається ($\alpha = 1$). Задача лишається незмінною, а саме: як побудувати ланцюг Маркова, значення якого збігатимуться до потрібного розподілу. Основним у генеруванні вибірки за Гіббсом є одновимірний умовний розподіл – розподіл, коли всі випадкові величини, крім однієї, приймаються фіксованими значеннями. Такі умовні розподіли значно легше моделювати, ніж складні багатовимірні спільні розподіли, і зазвичай вони мають простий вигляд (часто використовують нормальний розподіл, χ^2 -розподіл та інші відомі розподіли). Так моделюються n випадкових величин послідовно з n одновимірних умов замість генерування одного n -вимірного вектора за одну ітерацію, використовуючи спільний розподіл.

Для ілюстрації процедури генерування вибірки за Гіббсом розглянемо двовимірну випадкову величину (x, y) . Нехай потрібно розрахувати $p(x)$ та $p(y)$. Ідея полягає в тому, що значно легше розглядати послідовність умовних розподілів, $p(x|y)$ та $p(y|x)$, ніж отримувати безумовний розподіл однієї з компонент, інтегруючи щільність спільного розподілу, наприклад, $p(x) = \int p(x, y) dy$. Моделювання починається з певного початкового значення y_0 для y та отримання x_0 генеруванням випадкових величин з умовного розподілу $p(x|y = y_0)$. Потім генератор використовує x_0 для генерування нового значення y_1 , вибираючи його з умовного розподілу $p(y|x = x_0)$. Моделювання виконуємо таким чином:

$$x_i \sim p(x|y = y_{i-1}),$$

$$y_i \sim p(y|x = x_i).$$

Повторюючи цей процес k разів, генерується послідовність Гіббса довжиною k , де підмножина точок (x_j, y_j) для $1 \leq j \leq m < k$ береться як змодельована вибірка зі спільного розподілу. Послідовність Гіббса збігається до стаціонарного розподілу незалежно від початкового значення. Цей стаціонарний розподіл і є тим розподілом, який ми прагнули побудувати [14].

Аналогічно відбувається моделювання й у випадку, коли розглядається більше двох змін-

них. Зокрема, значення k -ї величини отримують з розподілу $p(\theta^{(k)} | \Theta^{(-k)})$, де $\Theta^{(-k)}$ визначає вектор, який містить всі змінні, крім k . Таким чином, на i -му кроці моделювання величину $\theta_i^{(k)}$ отримуємо з розподілу

$$\begin{aligned} \theta_i^{(k)} &\sim p(\theta^{(k)} | \theta^{(1)} = \theta_i^{(1)}, \dots, \theta^{(k-1)} = \theta_i^{(k-1)}, \\ &\theta^{(k+1)} = \theta_{i-1}^{(k+1)}, \dots, \theta^{(n)} = \theta_{i-1}^{(n)}). \end{aligned}$$

А. Гелфанд і А. Сміт у [15] показали потужність апарату Гіббса до вирішення широкого кола статистичних задач. А. Сміт та Г. Робертс в [16] продемонстрували тісний зв'язок моделювання за Гіббсом та байєсівської статистики (для отримання апостеріорного розподілу). Огляд методів моделювання вибірок можна знайти в [17].

Застосування методів МКМЛ до оцінювання моделі стохастичної волатильності

Модель стохастичної волатильності. Початковими даними вважатимемо щоденні обмінні курси валют (долар/гривня) $\{x_t\}$ за період з 24.10.2006 по 15.04.2011. Ці дані перетворимо у часовий ряд, який буде вказувати на середню відкориговану зміну курсу валют (тобто на середню волатильність курсу) за допомогою перетворення

$$\begin{aligned} y_t &= \log x_t - \log x_{t-1} - \frac{1}{n} \sum_{i=1}^n (\log x_i - \log x_{i-1}), \\ t &= 1, \dots, n. \end{aligned}$$

МСВ, що використовується для аналізу цих даних, може бути записана у формі нелінійної моделі [18]

$$y_t | \theta_t = \exp\left(\frac{1}{2} \theta_t\right) u_t, \quad u_t \sim N(0, 1), \quad t = 1, \dots, n, \quad (1)$$

де θ_t – прихована волатильність.

Припускається, що з часом невідомі стани θ_t утворюють ланцюг Маркова та задаються рівняннями

$$\begin{aligned} \theta_t | \theta_{t-1}, \mu, \phi, \tau^2 &= \mu + \phi(\theta_{t-1} - \mu) + v_t, \\ v_t &\sim N(0, \tau^2), \quad t = 1, \dots, n \end{aligned}$$

при цьому $\theta_0 \sim N(\mu, \tau^2)$. Стан θ_t визначає величину волатильності на момент t , а величина ϕ , $-1 < \phi < 1$, характеризує автокореляцію, наявну у квадраті прологарифмованих даних. От-

же, ϕ можна інтерпретувати як незмінність у волатильності, сталий множник $\beta = \exp(\mu / 2)$ – як модальну волатильність і τ – як волатильність лог-волатильності [12].

Байєсівська модель складається зі спільного апіорного розподілу всіх неспостережуваних величин, в цьому випадку трьох параметрів (μ, ϕ, τ^2) , невідомих станів $(\theta_0, \theta_1, \dots, \theta_n)$ та спільного розподілу спостережуваних даних y_1, \dots, y_n . Байєсівський аналіз даних ґрунтується на апостеріорному розподілі неспостережуваних величин на заданих даних. Позначивши функцію щільності ймовірності випадкової величини θ через $p(\theta)$, спільну апіорну щільність можна записати так:

$$p(\mu, \phi, \tau^2, \theta_0, \theta_1, \dots, \theta_n) = \\ = p(\mu, \phi, \tau^2) p(\theta_0 / \mu, \tau^2) \prod_{t=1}^n p(\theta_t / \theta_{t-1}, \mu, \phi, \tau^2).$$

При цьому припускається апіорна незалежність параметрів μ, ϕ, τ^2 . Оскільки апіорно параметр μ малоінформативний, то приймаємо $\mu \sim N(0, 10)$. Покладемо $\phi = 2\phi^* - 1$ і для бета-розподілу ϕ^* приймемо параметри $\alpha = 20$ і $\beta = 1,5$, при цьому нехай апіорне середнє для ϕ дорівнює 0,86. Для τ^2 вибираємо апіорні значення, обернені до гамма-розподілу, тобто $\tau^2 \sim IG(2, 5; 0, 025)$.

Ймовірність $p(y_1, \dots, y_n / \mu, \phi, \tau^2, \theta_0, \dots, \theta_n)$ визначається рівнянням (1) і припущенням умовної незалежності

$$p(y_1, \dots, y_n / \mu, \phi, \tau^2, \theta_0, \dots, \theta_n) = \prod_{t=1}^n p(y_t / \theta_t).$$

За теоремою Байєса спільний апостеріорний розподіл неспостережуваних величин при заданих $y_i, i = \overline{1, n}$ пропорційний добутку ймовірностей

$$p(\mu, \phi, \tau^2, \theta_0, \dots, \theta_n / y_1, \dots, y_n) \propto p(\mu) p(\phi) p(\tau^2) \times \\ \times p(\theta_0 / \mu, \tau^2) \prod_{t=1}^n p(\theta_t / \theta_{t-1}, \mu, \phi, \tau^2) \prod_{t=1}^n p(y_t / \theta_t).$$

Середовище OpenBUGS

Для реалізації моделі стохастичної волатильності використано програмну реалізацію алгоритму Гіббса в системі OpenBUGS. BUGS – це

програмне забезпечення, мета якого полегшити користувачам Windows виконання сучасного байєсівського аналізу даних, який ґрунтується на методі МКМЛ. Повна реалізація та документація BUGS розміщені у відкритому доступі в інтернеті, що дає можливість користувачам системи покращувати та розширювати систему. Ця версія називається OpenBUGS.

Значною мірою легкість користування програмним продуктом OpenBUGS забезпечується графічним інтерфейсом користувача та використанням широкого спектра інформації для розробленої програми.

Програмне забезпечення створює об'єкти, пов'язує разом і забезпечує їх взаємодію. Для подання байєсівської моделі будується динамічна структура цих об'єктів – спрямований ациклічний граф. Цей граф використовує припущення умовної незалежності для підвищення ефективності розрахунку умовних ймовірностей.

У середовищі OpenBUGS використовуються кілька інструментів, серед яких інструменти для моделювання параметрів моделі та спостереження за ними. Для моделювання параметрів моделі та копіювання їх у графічну структуру даних створено інструмент "a layer of updater objects", який задає кількість ітерацій алгоритму. Інструмент "a layer of monitor objects" створено для спостереження значень змодельованих параметрів та перегляду підсумкової статистики.

Розглянемо побудову графа об'єктів. Користувач у текстовому редакторі описує байєсівську модель на мові OpenBUGS. Ця модель є описом графа об'єктів, яку має реалізувати OpenBUGS. Компілятор перетворює текстове наведення байєсівської моделі в граф об'єктів.

Компіляція байєсівської моделі, написаної на мові OpenBUGS, складається з кількох етапів. По-перше, виконуються лексичний аналіз та сканування наведеної моделі – розбиття потоку символів на лексеми. По-друге, виконується синтаксичний аналіз для подання моделі у вигляді дерева. По-третє, будується граф об'єктів через послідовне внесення об'єктів з побудованого дерева, значення яких були позначені як дані. Нарешті, умовна незалежність використовується для створення переліку об'єктів графа для того, щоб розрахувати умовні розподіли.

Для отримання висновків OpenBUGS використовує алгоритми моделювання МКМЛ. Ці алгоритми вимагають значних обчислень, але вони стійкі до деяких подробиць задач, для яких вони застосовуються. Ця стійкість є важливою характеристикою таких систем як BUGS,

які автоматично вибирають алгоритм для отримання висновків. BUGS забезпечує широкий вибір алгоритмів МКМЛ.

OpenBUGS реалізовано на мові програмування Component Pascal (CP) у середовищі BlackBox Component Builder, розробленому компанією Oberon Microsystems. CP є досить сучасною мовою, що поєднує принципи процедурного й об'єктного програмування. Скомпільовані модулі містять метайнформацію, яка дає можливість користувачу модулів перевірити, що надають завантажені модульні сервіси, необхідні клієнту.

Специфікація моделі стохастичної волатильності у середовищі OpenBUGS

Для оцінки параметрів моделі стохастичної волатильності у середовищі OpenBUGS потрібно визначити такі три файли (з розширенням .odc):

1. Дані мають бути в окремому файлі, який названо "sv_data.odc".

```
list(y=c(0.0002, -0.0007, -0.0006, -0.0067,
0.0050, 0.0004, 0.0016, -0.0003, 0.0009, -0.0016, -
0.0020, ...
```

2. Для початку роботи алгоритму Гіббса потрібно задати початкові значення всіх параметрів, що знаходяться в окремому файлі; у цьому випадку з назвою "sv_init.odc".

```
list(mu=0, phistar=0.975, ntau=10,
theta0=0.5)
```

3. Опис моделі на мові OpenBUGS також знаходиться у файлі, який названо "sv_model.odc".

```
model{
  mu ~ dnorm(0, 0.1)
  phistar ~ dbeta(20, 1.5)
  ntau ~ dgamma(2.5, 0.025)
  phi<- 2*phistar-1
  theta0 ~ dnorm(mu, ntau)
  meantheta[1] <- mu + phi*(theta0 - mu)
  theta[1] ~ dnorm(meantheta[1], ntau)
  for (i in 2 : N){
    meantheta[i] <- mu +
phi * (theta[i-1] - mu)
    theta[i] ~
dnorm(meantheta[i], ntau)
  }
  for (j in 1 : N){
    meany[j] <- 1/exp(theta[j])
  }
}
```

```
y[j] ~ dnorm(0, meany[j])
}
```

Файл з даними може бути поданий у форматах списку або матриці. Файли з даними та початковими значеннями параметрів наведені у вигляді списку. Якщо файл з початковими значеннями параметрів відсутній, то ці значення будуть згенеровані у процесі моделювання з апріорного розподілу стосовно заданої моделі.

Подання моделі у вигляді спрямованого ациклічного графа (САГ) слугує не тільки внутрішнім потребам середовища OpenBUGS, але й допомагає представити модель на мові OpenBUGS. Файл зі специфікацією моделі складається з двох частин: декларацій та опису моделі. Частина декларацій встановлює назву моделі, вершинами якої є константи (прямокутники), і якщо є стохастичні величини, то вони задаються еліпсами, та дає назву файлам, що містять дані й початкові значення. Опис моделі утворює декларативне подання повної байєсівської моделі, яке задається у фігурних дужках {...}. Це і є перетворенням графічної моделі на синтаксис середовища OpenBUGS. Кожне твердження складається з відношення одного з двох видів:

1) символ "~" означає "є розподілений як" та графічно позначається одинарною стрілкою;

2) символ "<- " означає "замінюється на" та графічно позначається подвійною стрілкою.

Величина з лівої сторони символу "~" є стохастичною, а з лівої сторони символу "<- " – детермінованою. Кожна величина має визначатись лише один раз з лівої сторони твердження. Порядок виразів у фігурних дужках не має значення.

OpenBUGS містить модуль Doodle, який дає змогу користувачу описати модель графічно у вигляді САГ. Модуль використовує підхід гіперграфа для додавання додаткової інформації в граф з метою отримання повного опису моделі.

Важливо звернути увагу, що OpenBUGS використовує нестандартну параметризацію розподілів у термінах точності (1/дисперсія) замість дисперсії. Наприклад, нормальний розподіл, який позначається через $dnorm(v, \psi^2)$,

має щільність $f(x | v, \psi^2) = \frac{\psi}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-v)^2\psi^2}$. Це є

підставою для визначення вершини "ntau". OpenBUGS не дозволяє, щоб вирази були використані як параметри розподілу, тому потрібні детерміновані вершини "meantheta[t]" та "meany[t]".

Моделювання в середовищі OpenBUGS

Спочатку необхідно відкрити модель, описану на мові OpenBUGS, та перевірити її на синтаксичну коректність:

- в меню "File" вибрати пункт "Open" та відкрити файл "sv_model.odc";
- відкрити панель "Specification Tool", яка розміщена в меню "Model". У моделі стохастичної волатильності, описаній на мові OpenBUGS, виділити зарезервоване слово "model". У вікні "Specification Tool" натиснути кнопку "check model" для перевірки моделі на синтаксичну коректність (рис. 1).

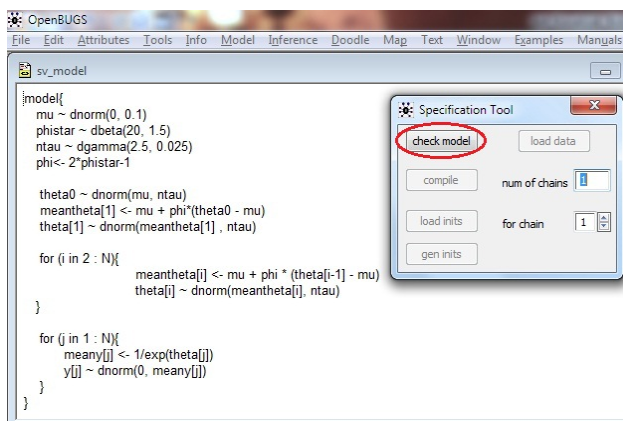


Рис. 1. Перевірка синтаксичної коректності моделі

Якщо модель задана коректно, то на панелі стану (у лівому нижньому кутку) буде виведено повідомлення "model is syntactically correct". В іншому випадку буде виведено повідомлення "invalid or unexrected token scanned".

Перед початком моделювання потрібно завантажити файл з даними, необхідними для моделювання, виконати компіляцію моделі та вказати деякі початкові значення. У файлі з даними задано такі значення:

y_i — часовий ряд, який вказує на середню відкориговану зміну курсу валют;

N — кількість даних часового ряду y_i .

Для завантаження даних потрібно у файлі з даними "sv_data.odc" виділити зарезервоване слово "list" та натиснути кнопку "load data" у вікні "Specification Tool" (рис. 2).

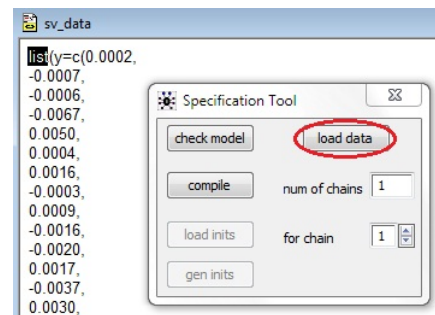


Рис. 2. Завантаження даних

Якщо дані завантажуться успішно, то на панелі стану буде виведено повідомлення "data loaded".

Перед компіляцією моделі потрібно визначити кількість ланцюгів, які будуть моделюватись. За замовчанням це значення встановлене на один ланцюг. Для перевірки збіжності моделювання встановимо значення кількості ланцюгів моделювання рівне двом. Змінити значення можна у полі "num of chains" панелі "Specification Tool".

Після встановлення кількості ланцюгів треба виконати компіляцію моделі. У вікні "Specification Tool" натиснути кнопку "compile". У разі успішної компіляції на панелі стану буде повідомлення "model compiled".

Залишилось вказати деякі початкові значення для кожної стохастичної вершини САГ. Для кожного ланцюга початкові значення мають вказуватись окремо. Величини цих початкових значень можуть бути довільними, оскільки це не впливає на саму збіжність значень параметрів, а має значення для швидкості збіжності та для кількості ітерацій. Є два варіанти отримання початкових значень:

- у меню "File" вибрати пункт "Open" та відкрити файл "sv_init.odc". У файлі з початковими значеннями виділити зарезервоване слово "list" та у вікні "Specification Tool" натиснути кнопку "load inits". На панелі стану з'явиться повідомлення "initial values loaded and chain initialized but another chain contains uninitialized variables". Тепер потрібно повторити процедуру для другого ланцюга, після чого на панелі стану буде повідомлення "model is initialized";

- у вікні "Specification Tool" натиснути кнопку "gen inits". Система OpenBUGS генерує початкові значення для кожного параметра.

Краще вказувати початкові значення відповідно до першого варіанту, щоб уникнути невідповідних значень.

Тепер модель стохастичної волатильності повністю задано і можна перейти до процедури моделювання. Перед цим доцільно вказати параметри, значення яких слід визначити. Для цього в меню “Inference” потрібно вибрати пункт “Samples”. У вікні “Sample Monitor Tool” у полі “node” вказати параметр і зафіксувати його, натиснувши кнопку “set”.

Для запуску процедури моделювання потрібно в меню “Model” вибрати пункт “Update”. У вікні “Update Tool” в полі “updates” ввести необхідне число оновлень (ітерацій моделювання). За замовчуванням це значення дорівнює 1000, але бажано встановити його більшим.

Після натиснення кнопки “update” OpenBUGS розпочне моделювання значення для кожного параметра моделі. Після завершення процедури моделювання на панель стану буде виведено повідомлення “updates took *** s”, де *** – кількість секунд, витрачених на моделювання.

Тепер можна перевірити збіжність моделі та переглянути результати моделювання (рис. 3). У вікні “Sample Monitor Tool” в полі “node” необхідно вибрати один із заданих параметрів. Натиснувши кнопку “history” для оцінювання збіжності моделі.

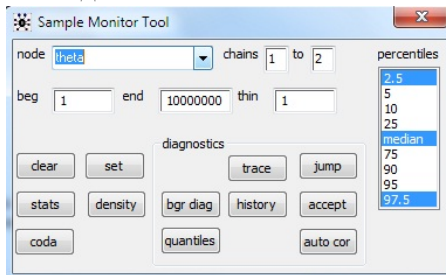


Рис. 3. Перегляд результатів моделювання

Якщо результати моделювання збіжні, то для отримання параметрів апостеріорного розподілу потрібно зробити ще кілька ітерацій. Процедуру моделювання можна оцінити після запуску достатньої кількості ітерацій. Для перегляду статистики потрібно у вікні “Sample Monitor Tool” натиснути кнопку “stats”.

У результаті роботи алгоритму в середовищі OpenBUGS отримано параметри моделі стохастичної волатильності та значення волатильності обмінних курсів (долар/гривня) (таблиця).

На рис. 4 наведено графік обчислювальних експериментів, який свідчить про збіжність процесу оцінювання для всіх параметрів моделі стохастичної волатильності, тобто оцінки наближаються до фактичних значень.

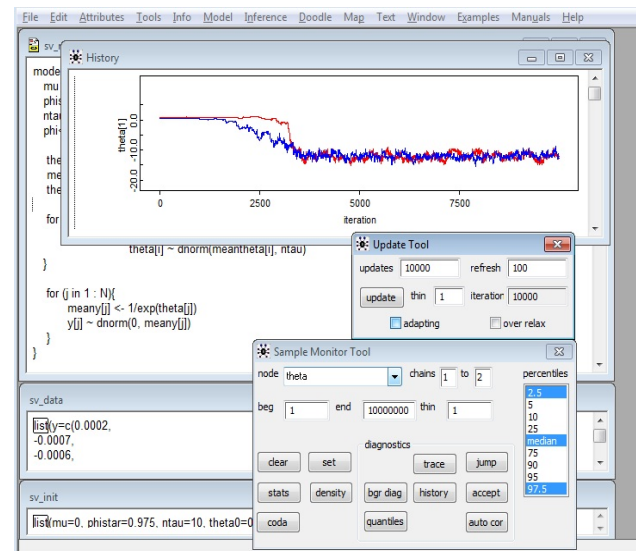


Рис. 4. Результати моделювання у середовищі OpenBUGS

Таблиця. Значення волатильності та параметрів μ та ϕ

Параметри	Середнє	Середньоквадратичне відхилення	Значення помилки	2,5 %	Медіана	97,5 %	Початкова ітерація	Кількість ітерацій
Mu	-7,84	5,125	0,3618	-12,35	-11,07	0,02938	1	20000
phi	0,9659	0,02287	0,001527	0,9296	0,9597	0,9991	1	20000
theta[1]	-8,306	5,555	0,3912	-13,86	-11,33	0,9169	1	20000
theta[2]	-8,864	4,811	0,3377	-13,72	-11,33	0,2557	1	20000
theta[3]	-9,158	4,092	0,286	-13,27	-11,16	-0,3523	1	20000
theta[4]	-9,096	3,333	0,2329	-12,13	-10,66	-0,9295	1	20000
theta[5]	-9,593	3,102	0,216	-12,36	-10,91	-1,388	1	20000
theta[6]	-10,37	3,153	0,2187	-13,47	-11,55	-1,754	1	20000
theta[7]	-10,85	3,106	0,2157	-13,87	-11,94	-2,136	1	20000
theta[8]	-11,23	3,098	0,2146	-14,36	-12,22	-2,327	1	20000
theta[9]	-11,34	2,981	0,2061	-14,28	-12,23	-2,484	1	20000
theta[10]	-11,27	2,824	0,1955	-13,81	-12,16	-2,601	1	20000

Висновки

Для оцінювання параметрів моделі стохастичної волатильності курсу валют (долар/гривня) за період з 24.10.2006 по 15.04.2011 вперше застосовано пакет прикладних програм OpenBUGS, який є ефективним інструментом для виконання байєсівського аналізу МСВ та, на відміну від WinBUGS, гнучкіший.

Для застосування середовища OpenBUGS створено специфікацію моделі стохастичної волатильності у середовищі OpenBUGS.

У результаті виконаного моделювання встановлено, що отримані оцінки є цілком задовільними за точністю з прийнятними обчислювальними витратами, всі оцінки характеризуються збіжністю на заданому часовому інтервалі.

Запропонований алгоритм може бути застосований для оцінки значень волатильності стохастичного процесу та параметрів моделі стохастичної волатильності для різних фінансових процесів, представлених статистичними даними обмінних курсів різних валют.

1. *Taylor S.J.* Financial Returns Modelled by the Product of Two Stochastic Processes—a Study of the Daily Sugar Prices 1961–75 // *Time Series Analysis: Theory and Practice*, 1 / O.D. Anderson (ed.). — Amsterdam: North-Holland. — 1982. — P. 203–226.
2. *Tauchen G., Pitts M.* The Price Variability-Volume Relationship on Speculative Markets // *Econometrica*. — 1983. — **51**. — P. 485–505.
3. *Engle R.F.* Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation // *Ibid.* — 1982. — **50**. — P. 987–1007.
4. *Bollerslev T.* Generalized Autoregressive Conditional Heteroskedasticity // *J. of Econometrics*. — 1986. — **31**. — P. 307–327.
5. *Taylor S.J.* Modelling Stochastic Volatility // *Math. Finance*. — 1994. — **4**. — P. 183–204.
6. *Sorensen M.* Prediction Based Estimating Equations // *Econometrics*. — 2000. — **3**. — P. 123–147.
7. *Harvey A.C., Ruiz E., Shephard N.* Multivariate Stochastic Variance Models // *Review of Economic Studies*. — 1994. — **61**. — P. 247–264.
8. *Gallant A.R., Hsie D., Tauchen G.* Estimation of Stochastic Volatility Models with Diagnostics // *J. of Econometrics*. — 1997. — **81**. — P. 159–192.
9. *Sandmann G., Koopman S.J.* Estimation of Stochastic Volatility Models Via Monte Carlo Maximum Likelihood // *Ibid.* — 1998. — **87**. — P. 271–301.
10. *Fridman M., Harris L.* A Maximum Likelihood Approach for Non-Gaussian Stochastic Volatility Models // *J. of Business and Economic Statistics*. — 1998. — **16**. — P. 284–291.
11. *Gilks W.R., Roberts G.O.* Strategies for Improving MCMC // *Markov Chain Monte Carlo in Practice* / W.R. Gilks, S. Richardson, D.J. Spiegelhalter (eds). — London: Chapman&Hall. — 1996. — P. 89–114.
12. *Kim S., Shephard N., Chib S.* Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models // *Review of Economic Studies*. — 1998. — **65**. — P. 361–393.
13. *Chib S., Greenberg E.* Understanding the Metropolis-Hastings Algorithm // *American Statistician*. — 1995. — P. 327–335.
14. *Tierney L.* Markov Chains for Exploring Posterior Distributions (with discussion) // *Ann. Statist.* — **22**. — P. 1701–1762.
15. *Gelfand A.E., Smith A.F.M.* Sampling-Based Approaches to Calculating Marginal Densities // *J. Am Stat. Asso.* — 1990. — **85**. — P. 398–409.
16. *Smith A.F.M., Roberts G.O.* Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte-Carlo Methods (with discussion) // *J. Roy. Stat. Soc. Series B*. — 1993. — **55**. — P. 3–23.
17. *Besag J., Green P.J., Higdon D.M., Mengersen K.* Bayesian Computation and Stochastic Systems (with discussion) // *Statist. Science*. — 1995. — **10**. — P. 3–66.
18. *Harvey A.* Forecasting Structural Time Series Models and the Kalman Filter. — New York: Cambridge University Press, 1989. — 572 p.

Рекомендована Радою
Навчально-наукового комплексу
“Інститут прикладного системного
аналізу” НТУУ “КПІ”

Надійшла до редакції
12 квітня 2011 року