ACTUARIS
INTERNATIONAL

The Reversible Jump Markov Chain Monte Carlo: extensions and practical applications

Marion Gremillet marion.gremillet@actuaris.com

INTRODUCTION

Insurers increasingly require highly robust stochastic models to obtain credible valuations of their outstanding claims reserves best estimate or Value at Risk. This is particularly true for firms subject to the EU's Solvency II regulations.

Outstanding claims reserves represent most of insurers' liabilities under Solvency II. For non-life companies, the valuation of these reserves is mostly based on the study of a run-off triangle which represents the evolution of insurers' payments, depending on the development year and the occurrence year of the underlying claim. The actuary's aim is to "complete" this triangle, i.e. evaluate the outstanding claims for future accounting years.

Traditional methods, including the famous Chain Ladder approach, proceed by evaluating column by column each element of the lower triangle according to the upper triangle data. In the case of the Chain Ladder method, this evaluation is based on an estimated development factor which determines one column's data from earlier ones. Estimates for the columns on the left-hand side contain a lot of data used to project very few points. Conversely, the right-hand columns contain relatively little data which are used to project many points, which appears counter-intuitive and generally leads to a high estimate error.

These observations prompt me to research the potential use of the RJMCMC method, proposed by Verrall and Wüthrich (2012). This method assumes that amounts follow an over dispersed Poisson distribution with parameters for each line and column. The triangle is split in two parts with a dedicated model in each part to estimate the column parameters: one for the left part of the triangle, based on more parameters, which allows a better fit to the data; and one for the right part of the triangle using only two parameters and reference statistical curves. This allows a more robust valuation of the tail, the last columns corresponding mainly to the development of the claims that have already occurred which can more easily fit a simple parametric model.

One of the main issues therefore, is defining the column where the split from one methodology to the other will occur. Fortunately, the RJMCMC method addresses this issue with a solution that is both complex and pragmatic.

In section 1 the different assumptions of the model and the functioning of the algorithm are detailed. Section 2 describes a methodology which allows managing with negatives values, which is required to apply RJMCMC to most real case triangles. Section 3 proposes some extensions of the model by applying different functions to model the right part (or "tail") of the triangle. Finally the section 4 is devoted to a proposition of the one-year uncertainty estimation in order to apply RJMCMC in the context of the Solvency II European directive. Each part is illustrated by concrete applications of the method on real data and comparison to traditional methods, which allows us to conclude by highlighting the advantages and utility of the method.

1. Proposed methodology to apply Reversible Jump Markov Chain Monte Carlo algorithm to incremental triangles

This section presents the different assumptions of the studied methodology stressing the Bayesian Over-Dispersed Poisson model on which RJMCMC is based. Then, step by step, the description of the algorithm will be given.

1.1 Model Assumptions

The aim of the RJMCMC methodology is to predict the lower part of the triangle, using the data provided by the upper part of the triangle. Hereafter, we will consider that both the occurrence years of the underlying claim and the development years are numbered from 0 to I. Let $X_{i,j},\ 0 \le i,j \le I$ be the values of the incremental amounts. As for any other reserving methodologies, the aim is the prediction of the lower triangle which will be noted $D_I^C = \{X_{ij}; i+j > I, 0 \le i \le I, 0 \le j \le I\}$, based on the upper triangle, defined by $D_I = \{X_{ij}; i+j \le I, 0 \le i \le I, 0 \le j \le I\}$.

- The first assumption of the model is that conditionally to the values of $\vartheta = (\mu_0, \dots, \mu_I, \gamma_0, \dots, \gamma_I, \varphi)$ each incremental amount follows an over-dispersed Poisson distribution with the following parameters:

$$\left(\frac{X_{ij}}{\varphi}\middle|\vartheta\right) \sim Poi\left(\frac{\mu_i\,\gamma_j}{\varphi}\right)$$

Where μ_i is the parameter for row i and γ_j is the parameter for column j. φ is a computed constant known as the over-dispersion parameter.

$\vartheta = (\mu_0, \dots, \mu_I, \gamma_0, \dots, \gamma_I, \varphi)$ is the vector of parameters that we need to estimate.

Negative increments can lead to some issues in over-dispersed Poisson models, thus we will propose methodologies to manage with negatives in section 3.

- The second assumption concerns the estimation of the row parameters. They are supposed to be independent random variables and gamma distributed with the following parameters:

$$\forall i \in \{0, \dots, I\} \quad \mu_i \sim \Gamma(s, \frac{s}{m_i})$$

Where s and m_i are positive prior estimates, computed in the way detailed in section 2.2.

- The third main assumption is that two different models are used to estimate the vector of column parameters $(\gamma_0, \dots, \gamma_I)$. Until a truncation column index the column parameters are independent and gamma distributed; and starting from this truncation index, an exponential decay is used to estimate the column parameters. Let k be the truncation index:
  - $\forall j \in \{0, \dots, k-1\} \quad \gamma_j \sim \Gamma(v, \frac{v}{c_j})$
  - $\forall j \in \{k, \dots, I\} \quad \gamma_j = \exp(\alpha - j\beta)$

The exponential decay implies the estimate of the two parameters α and β. In that aim we define prior distributions for these two parameters:

$\alpha \sim N(a, \sigma^2)$ and $\beta \sim N(b, \tau^2)$

Where v, c_j, σ and τ are positive prior estimates and a and b are real prior parameters. The choices of these estimates are described in section 2.2.

As detailed in Section 4, we could instead of using an exponential decay decide to use any other two parameter curve that could fit better.

Finally, the initial parameters vector $\vartheta = (\mu_0, \dots, \mu_I, \gamma_0, \dots, \gamma_I, \varphi)$ can be replaced by a new vector to be estimated $\theta_k = (\alpha, \beta, \mu_0, \dots, \mu_I, \gamma_0, \dots, \gamma_{k-1})$.

It is then possible to express the joint density of the data $\left(X_{i,j}\right)_{(i,j)\in\{0,\dots,I\}^2}$ and the parameter vector θ_k:

$$f_k\left(\left(X_{i,j}\right)_{(i,j)\in\{0,\dots,I\}^2}, \theta_k\right) = f_k\left(\left(X_{i,j}\right)_{(i,j)\in\{0,\dots,I\}^2} \middle| \theta_k\right) p_k(\theta_k)$$

Where

$$f_k\left(\left(X_{i,j}\right)_{(i,j)\in\{0,\dots,I\}^2} \middle| \theta_k\right) = \prod_{(i,j)\in\{0,\dots,I\}^2} e^{-\frac{\mu_i \gamma_j}{\varphi}} \frac{\left(\frac{\mu_i \gamma_j}{\varphi}\right)^{\frac{X_{ij}}{\varphi}}}{\left(\frac{X_{ij}}{\varphi}\right)!}$$

And

$$p_k(\theta_k) \propto \prod_{i=0}^{I} \mu_i^{s-1} e^{-\frac{s}{m_i}\mu_i} \times \prod_{j=0}^{k-1} \gamma_j^{v-1} e^{-\frac{v}{c_j}\gamma_j} \times exp\left\{-\frac{1}{2\sigma^2}(\alpha - a)^2\right\} \times exp\left\{-\frac{1}{2\tau^2}(\beta - b)^2\right\}$$

The sign "$\propto$" express the proportion, as the normalizing constants are not calculated. The term $p_k(\theta_k)$ corresponds to the product of the prior densities of the row parameters μ_i, the column parameters γ_j and the prior densities of the parameters α and β.

We can define a Markov Chain, for which each state t is characterized by the truncation index and the parameter vector $\mathbf{\Theta}^{(t)} = \left(k^{(t)}, \theta_{k^{(t)}}^{(t)}\right)$.

1.2 Application of the RJMCMC algorithm

Starting from the paper of Verrall and Wüthrich (2012), we summarize below the main steps of the RJMCMC algorithm.

First of all, the algorithm has to be initialized. We compute the maximum likelihood estimators of the row and column parameters normalized such that the sum of the column parameters is equal to 1. This is a convention which has no impact on the future calculations. This choice is quite convenient as the column could be, in this way, associated to development patterns and the row parameters could be seen as the expected ultimate amounts.

From these estimators, we can compute different prior parameters with Maximum Likelihood (an acceptable but less accurate alternative could be to use mean square approach):

$$m_i = \mu_i^{MLE}$$

$$c_j = \gamma_j^{MLE}$$

These prior estimates are used in the distributions of the row and the column parameters:

$$\forall i \in \{0, \dots, I\} \quad \mu_i \sim \Gamma(s, \frac{s}{m_i})$$

$$\forall j \in \{0, \dots, k-1\} \quad \gamma_j \sim \Gamma(v, \frac{v}{c_j})$$

Parameters s and v reflect the prior uncertainties associated with the estimate of row and column parameters.

They have to be chosen such that they reflect the best the uncertainty linked with the prior estimations. For instance the following coefficients of variation can be allocated: 10% to row parameters and 100% to column parameters. Indeed, it seems more coherent to choose the priors of the column parameters to be rather non-informative because the smoothing effect is not taken into account in the prior estimate of the γ_j.

While the initialization has been performed, we can go through the recursive algorithm.

Let's suppose we have finished calculating estimators for step t. Starting from this, and based on the Markov Chain principal, we want to produce the calculations for step $t+1$.

Step A: the choice of a new truncation index

We choose a new truncation index k^* from the previous one $k^{(t)}$. The following discrete probability distribution is defined:

$$\forall k^{(t)} \in \{2, \dots, I-1\} \quad q\left(k^* = k^{(t)} - 1 \middle| k^{(t)}\right) = q\left(k^* = k^{(t)} + 1 \middle| k^{(t)}\right) = q\left(k^* = k^{(t)} \middle| k^{(t)}\right) = \frac{1}{3}$$

$$q\left(k^* = 1 \middle| k^{(t)} = 1\right) = \frac{2}{3} \quad q\left(k^* = 2 \middle| k^{(t)} = 1\right) = \frac{1}{3}$$

$$q\left(k^* = I \middle| k^{(t)} = I\right) = \frac{2}{3} \quad q\left(k^* = I-1 \middle| k^{(t)} = I\right) = \frac{1}{3}$$

This distribution implies that it is possible to jump to next neighbor models, which means that the parameters vector dimension may change by one unit (plus or minus) or remain unchanged.

Step B: updating all parameters when $k^* = k^{(t)}$

If $k^* = k^{(t)}$ then we can directly set $k^{(t+1)} = k^{(t)}$. And we apply the Metropolis Hastings block sampler to update each parameter, which is decomposed in three steps:

- The updating of $(\mu_0^{(t)}, \dots, \mu_I^{(t)})$ using the Gibbs sampler. Conditionally to the other parameters, they are mutually independent and follow gamma distributions with parameters:

$$\mu_i^{(t+1)} \sim \Gamma\left(s_i^{post}, \left(\frac{s}{m_i}\right)_i^{post}\right)$$

Where $\quad s_i^{post} = s_i + \frac{1}{\varphi}\sum_{j=0}^{I-i} X_{ij} \quad$ and $\quad \left(\frac{s}{m_i}\right)_i^{post} = \frac{s}{m_i} + \frac{1}{\varphi}\sum_{j=0}^{I-i} \gamma_j^{(t)}$

Note that the updating of the row parameters μ_i uses the values of the column parameters but at the previous state, $\gamma_j^{(t)}$, as they have not been yet updated.

This concludes the updating of the row parameters.

- The updating of $(\gamma_0^{(t)}, \dots, \gamma_{k^{(t)}-1}^{(t)})$ using the Gibbs sampler. Conditionally to the other parameters, they are mutually independent and follow gamma distributions with parameters:

$$\gamma_j^{(t+1)} \sim \Gamma\left(v_j^{post}, \left(\frac{v}{c_j}\right)_j^{post}\right)$$

Where $v_j^{post} = v + \frac{1}{\varphi}\sum_{j=0}^{I-j} X_{ij}$ and $\left(\frac{v}{c_j}\right)_j^{post} = \frac{v}{c_j} + \frac{1}{\varphi}\sum_{i=0}^{I-j}\mu_i^{(t+1)}$

Note that the updating of the column parameters γ_j uses the values of the row parameters but at the current state, $\mu_i^{(t+1)}$, as they have been previously updated.

This concludes the updating of the column parameters.

- The updating of $(\alpha^{(t)}, \beta^{(t)})$ using the Metropolis Hastings algorithm.

We propose new values for this vector by generating a two-dimensional Gaussian distribution with parameters:

$$(\alpha^*, \beta^*) \sim \mathcal{N}\left(\begin{pmatrix}\alpha^{(t)}\\ \beta^{(t)}\end{pmatrix}, \Sigma\right)$$

Where Σ represents the covariance matrix. For more simplicity, we consider that α and β are independent:

$$\Sigma = \begin{pmatrix} Var_\alpha & 0 \\ 0 & Var_\beta \end{pmatrix}$$

We then need to calculate an acceptance probability which uses the following general formula, as described by Green (1995):

$$\alpha(t \to *) = min\left(1, \frac{f(\alpha^*, \beta^*) \times q\left((\alpha^{(t)}, \beta^{(t)})\middle|(\alpha^*, \beta^*)\right)}{f(\alpha^{(t)}, \beta^{(t)}) \times q\left((\alpha^*, \beta^*)\middle|(\alpha^{(t)}, \beta^{(t)})\right)}\right)$$

The last terms correspond to the proposal distribution. In our case this is equal to the density function of the two-dimensional Gaussian distribution previously written, which is an even function.

Thus, we have $q\left((\alpha^{(t)}, \beta^{(t)})\middle|(\alpha^*, \beta^*)\right) = q\left((\alpha^*, \beta^*)\middle|(\alpha^{(t)}, \beta^{(t)})\right)$

So the acceptance probability can be written as:

$$\alpha(t \to *) = min\left(1, \frac{f(\alpha^*, \beta^*)}{f(\alpha^{(t)}, \beta^{(t)})}\right)$$

Where the density f is proportional to:

$$f(\alpha, \beta) \propto \prod_{j=k^{(t)}}^{I}\left[e^{-\exp(\alpha - j\beta)\frac{\sum_{i=0}^{I-j}\mu_i^{(t+1)}}{\varphi}}(\exp(\alpha - j\beta))^{\frac{\sum_{i=0}^{I-j}X_{ij}}{\varphi}}\right] \times exp\left\{-\frac{1}{2\sigma^2}(\alpha - a)^2\right\} \times exp\left\{-\frac{1}{2\tau^2}(\beta - b)^2\right\}$$

Finally, two cases are possible:

- If the proposal values are accepted we set $\left(\alpha^{(t+1)}, \beta^{(t+1)}\right) = (\alpha^*, \beta^*)$
- If the proposal values are rejected we set $\left(\alpha^{(t+1)}, \beta^{(t+1)}\right) = \left(\alpha^{(t)}, \beta^{(t)}\right)$

This concludes the updating of the tail factors.

These three steps provide the updated parameters:

$$\boldsymbol{\Theta}^{(t+1)} = \left(k^{(t+1)}, \theta_{k^{(t+1)}}^{(t+1)}\right) = \left(k^{(t+1)}, \left(\alpha^{(t+1)}, \beta^{(t+1)}, \mu_0^{(t+1)}, \dots, \mu_I^{(t+1)}, \gamma_0^{(t+1)}, \dots, \gamma_{k^{(t+1)}-1}^{(t+1)}\right)\right)$$

Step C: cases corresponding to $\boldsymbol{k^* \neq k^{(t)}}$

These are the cases when the dimension of the parameter vector changes. The only parameter to consider is the column parameter that is supposed to jump from one model to the other.

- Case 1: $k^{(t)} < I$ and $k^* = k^{(t)} + 1$

This means that the column parameter $\gamma_{k^{(t)}}^{(t)}$ will leave the tail distribution and join the left part of the column parameters vector.

All the other parameters will not be updated and are equal to the ones of the previous state.

We propose a new value for the column parameter that jumps from one model to the other:
$\gamma_{k^{(t)}}^{*} \sim \Gamma\left(v^*, \frac{v^*}{exp\{\alpha^{(t)} - k^{(t)}\beta^{(t)}\}}\right)$

The following acceptance probability is then computed as mentioned in Verrall and Wüthrich (2012):

$$\alpha(t \to *) = min\left\{1, \prod_{i=0}^{I-k^*}\left[\frac{e^{-\frac{\mu_i^{(t)}\gamma_{k^{(t)}}^{*}}{\varphi}}\left(\gamma_{k^{(t)}}^{*}\right)^{\frac{X_{ik^{(t)}}}{\varphi}}}{e^{-\frac{\mu_i^{(t)}\gamma_{k^{(t)}}^{(t)}}{\varphi}}\left(\gamma_{k^{(t)}}^{(t)}\right)^{\frac{X_i}{\varphi}}}\right] \frac{\frac{(\frac{v}{c_{k^{(t)}}})^{v}}{\Gamma(v)}\left(\gamma_{k^{(t)}}^{*}\right)^{v-1} e^{-\frac{v}{c_{k^{(t)}}}\gamma_{k^{(t)}}^{*}}}{\frac{\left(\frac{v^*}{\gamma_{k^{(t)}}^{(t)}}\right)^{v^*}}{\Gamma(v^*)}\left(\gamma_{k^{(t)}}^{*}\right)^{v^*-1} e^{-\frac{v^*}{\gamma_{k^{(t)}}^{(t)}}\gamma_{k^{(t)}}^{*}}}\right\}$$

Thus, two cases are possible:

- If the proposal value is accepted we set $\gamma_{k^{(t)}}^{(t+1)} = \gamma_{k^{(t)}}^{*}$
- If the proposal value is rejected we set $\gamma_{k^{(t)}}^{(t+1)} = \gamma_{k^{(t)}}^{(t)}$, which means that we keep the value of the previous state of the Markov Chain

- Case 2: $k^{(t)} > 1$ and $k^* = k^{(t)} - 1$

This means that the column parameter $\gamma_{k^*}^{(t)}$ will leave the left part of the column parameters vector and join the tail distribution.

All the other parameters will not be updated and remain equal to their values of the previous state.

We propose a new value for the column parameter that jumps from one model to the other: $\gamma_{k^*}^{*} = \exp\left(\alpha^{(t)} - k^*\beta^{(t)}\right)$

The following acceptance probability is then computed:

$$\alpha(t \to *) = min\left\{1, \prod_{i=0}^{I-k^*}\left[\frac{e^{-\frac{\mu_i^{(t)}\gamma_{k^*}^{*}}{\varphi}}\left(\gamma_{k^*}^{*}\right)^{\frac{X_{ik^*}}{\varphi}}}{e^{-\frac{\mu_i^{(t)}\gamma_{k^*}^{(t)}}{\varphi}}\left(\gamma_{k^*}^{(t)}\right)^{\frac{X_{ik^*}}{\varphi}}}\right] \frac{\frac{(\frac{v^*}{\gamma_{k^*}^{*}})^{v^*}}{\Gamma(v^*)}\left(\gamma_{k^*}^{(t)}\right)^{v^*-1} e^{-\frac{v^*}{\gamma_{k^*}^{*}}\gamma_{k^*}^{(t)}}}{\frac{\left(\frac{v}{c_{k^*}}\right)^{v}}{\Gamma(v)}\left(\gamma_{k^*}^{(t)}\right)^{v-1} e^{-\frac{v^*}{c_{k^*}}\gamma_{k^*}^{(t)}}}\right\}$$

Thus, two cases are possible:

- If the proposal value is accepted we set $\gamma_{k^*}^{(t+1)} = \gamma_{k^*}^{*}$
- If the proposal value is rejected we set $\gamma_{k^*}^{(t+1)} = \gamma_{k^*}^{(t)}$, which means that we keep the value of the previous state of the Markov Chain.

Finally, we get a new parameter vector $\boldsymbol{\Theta}^{(t+1)} = (k^{(t+1)}, \theta_{k^{(t+1)}}^{(t+1)})$.

The graph below summarizes the steps from the state t to the state $t+1$ of the Markov Chain:

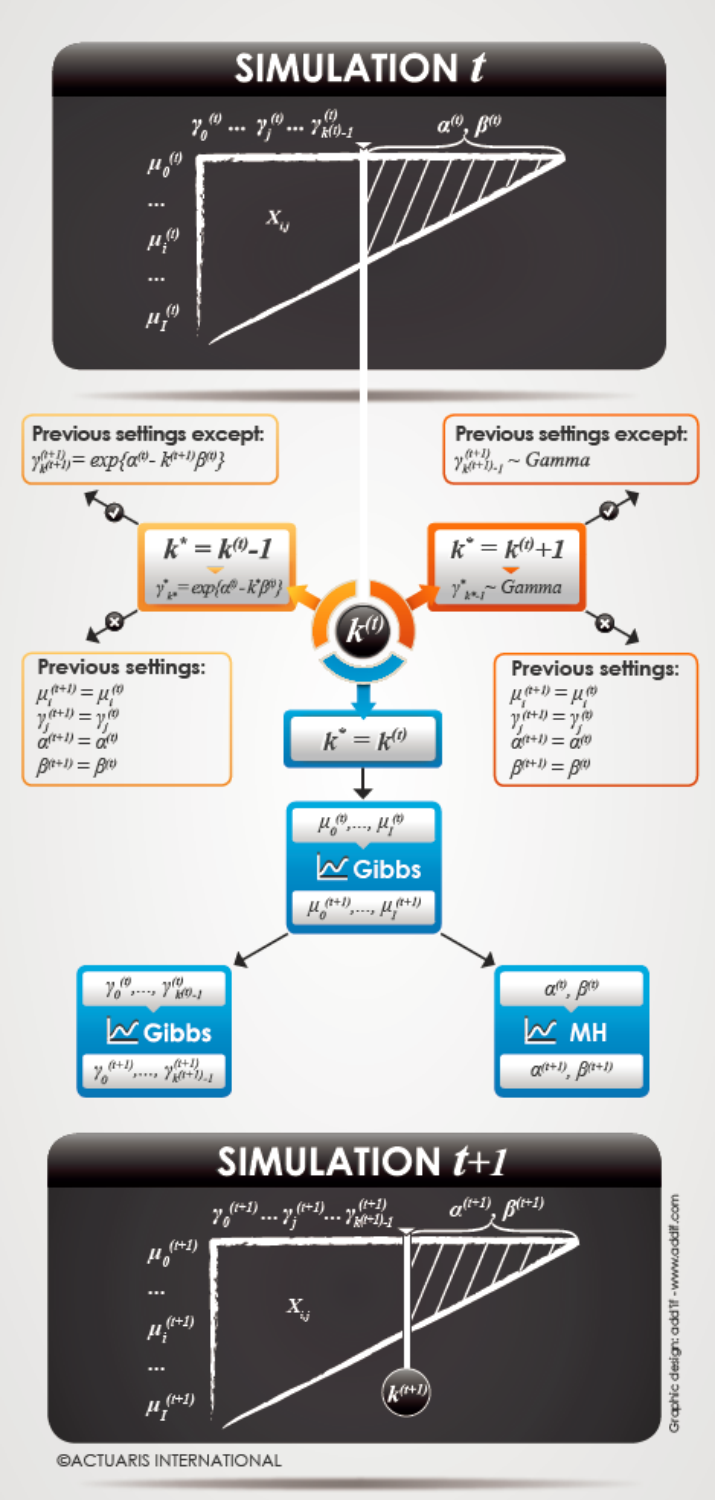


Figure 1: Functioning of the core of the RJMCMC algorithm

It is then possible at each iteration to estimate the lower part of the triangle. We generate random over-dispersed Poisson values with the estimated parameters to take into account the process error:

$$\forall i+j>I \qquad \left(\frac{\hat{X}_{i,j}^{(t)}}{\varphi}\middle|\vartheta\right)\sim Poi\left(\frac{\hat{\mu}_i^{(t)}\,\hat{\gamma}_j^{(t)}}{\varphi}\right)$$

Thus, it is easy to obtain an estimation of the total reserve $\hat{R}^{(t)}$ by summing the estimated increments of the lower part of the triangle.

Iterating this steps, at the end of all the simulations we obtain a distribution of the total reserve. This makes possible the computation of the mean and several risk measures. However, it is important to exclude the first simulations from final calculations as they correspond to the research of stability of the RJMCMC algorithm; this phase is called the Burn-in. The computations of the mean and of other risk measures will be done on the latest simulations and they will not be polluted by the Burn-in phase.

2. MODELING NEGATIVE VALUES

Negative incremental amounts are typical issues when using over-dispersed Poisson models. Unfortunately, this often happens especially with incurred triangles.

In this section we propose a solution to this problem for being able to apply the RJMCMC algorithm on nearly any kinds of real case input triangles.

2.1 METHODOLOGY TO MANAGE NEGATIVE VALUES

Negative incremental amounts represent an issue in over-dispersed Poisson models. Indeed, it is assumed that the sums of the incremental values in every development periods and origin periods of the loss data triangle need to be greater than zero.

We could think of excluding these points from the model. However, we would misestimate the claims because we would model it as being equal to 0 for these points. This is obviously not the case; therefore we have to find another way to deal with it.

The methodology we propose to use is based on the paper of Kunkler (2006).

For each column (development year) we split the values into two sets:

- The first set contains the strictly negative incremental values of the column j:

$S_j^{(-)} = \{X_{i,j} \mid 0 \leq i \leq I - j \text{ and } X_{i,j} < 0\}$. Let $n_j^{(-)}$ be the number of values contained in the set $S_j^{(-)}$.

- The second set contains the positive incremental values of the column j:

$S_j^{(+)} = \{X_{i,j} \mid 0 \leq i \leq I - j \text{ and } X_{i,j} \geq 0\}$. Let $n_j^{(+)}$ be the number of values contained in the set $S_j^{(+)}$.

For each column it is then possible to compute what we call the *probability of being negative*. This probability corresponds to the number of negative values divided by the total number of values:

$\forall\, 0 \leq j \leq I,\ p_j^{(-)} = \frac{n_j^{(-)}}{n_j^{(-)} + n_j^{(+)}}$ stands for the *probability of being negative* for the column j.

The *probability of being negative* is then computed for each column.

From the input data triangle, it is possible to compute the *pseudo data triangle* which is composed of the absolute values of the incremental claims.

Let $\mathcal{P}_I = \{|X_{i,j}|;\ i + j \leq I, 0 \leq i \leq I, 0 \leq j \leq I\}$ be this triangle.

We can then apply the RJMCMC algorithm on the *pseudo data triangle* as each incremental value is positive in this triangle.

Let $\hat{X}_{i,j}^{P}$, $i+j>I, 0 \leq i \leq I, 0 \leq j \leq I$, represent the expected amounts for the lower triangle after applying the algorithm on the pseudo data triangle $\mathcal{P}_I$.

At each iteration, in order to take into account the negatives contained in the input data triangle, we apply the *probability of being negative* computed by the formula above.

In that aim we use Bernoulli distributions with parameters $p_j^{(-)}$. For each amount of the lower triangle we generate a random number equal to 1 with probability $p_j^{(-)}$ and equal to 0 with probability $1-p_j^{(-)}$. Let $t_{i,j}$ represent the random number that we generate for the cell row i and column j.

We then apply the following formula:

$\hat{X}_{i,j} = (-1)^{t_{i,j}} \times \hat{X}_{i,j}^{P}$ for $i+j>I, 0 \leq i \leq I, 0 \leq j \leq I$

This methodology allows taking into account all the information of the upper triangle and reproducing negatives in the estimated triangle.

These two extensions presented in section 3 allow the application of RJMCMC on most triangles, including the ones which present negative and incremental amounts equal to zero. We will then be able to apply the methodology to a whole range of market triangles as shown in the Examples section.

2.2 Applications

We have applied RJMCMC with this extension to a whole set of market anonymous triangles kindly provided by the Belgian supervisor (*Banque Nationale Belge*). Indeed, we have run RJMCMC on 17 paid triangles for different lines of business: motor liability, general liability, legal protection and fire.

We have launched RJMCMC for each of these triangles; we have summarized the means and the standard deviations in the following graphs. Results obtained with the RJMCMC algorithm are then compared with the traditional methods of Chain Ladder / Mack and Bootstrap.

The triangles are numbered from A to Q.

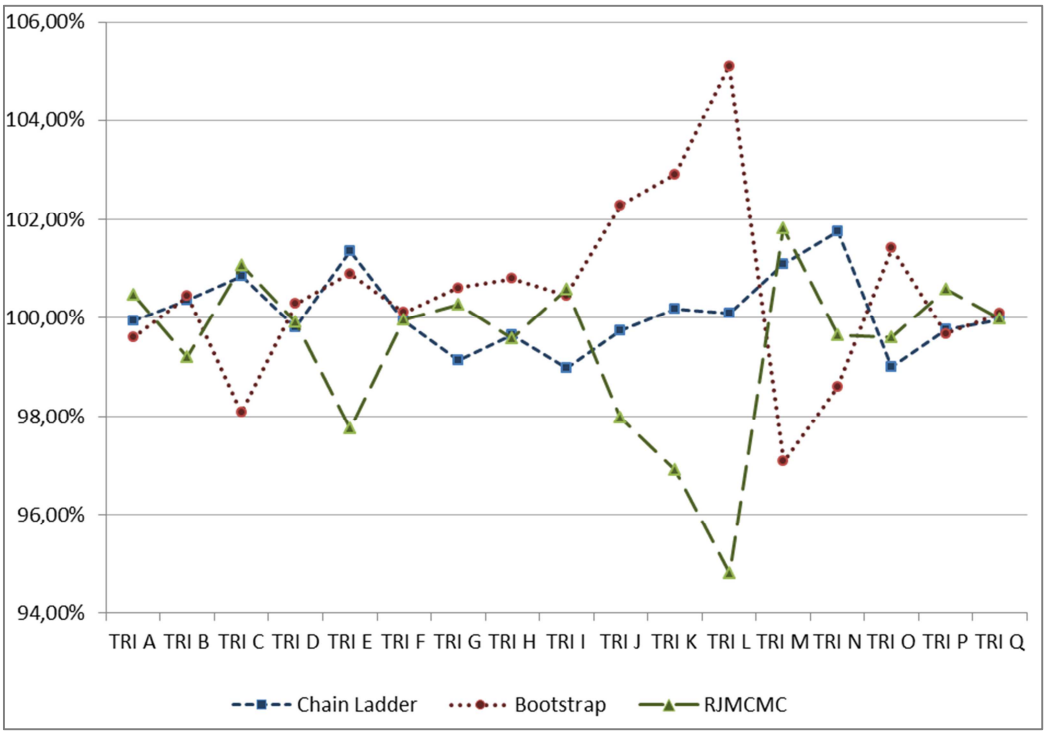


Figure 5: Means of the reserves in percentage of the mean of the three methods

To draw a comparison between methods we have compared the results with the mean of the three methods for each triangle. We can observe that the results of the mean valuation are quite similar, but the mean obtained with the RJMCMC method is often lower than the one estimated with the Bootstrap.

Let's now consider the coefficients of reserves variations obtained with the different methods.

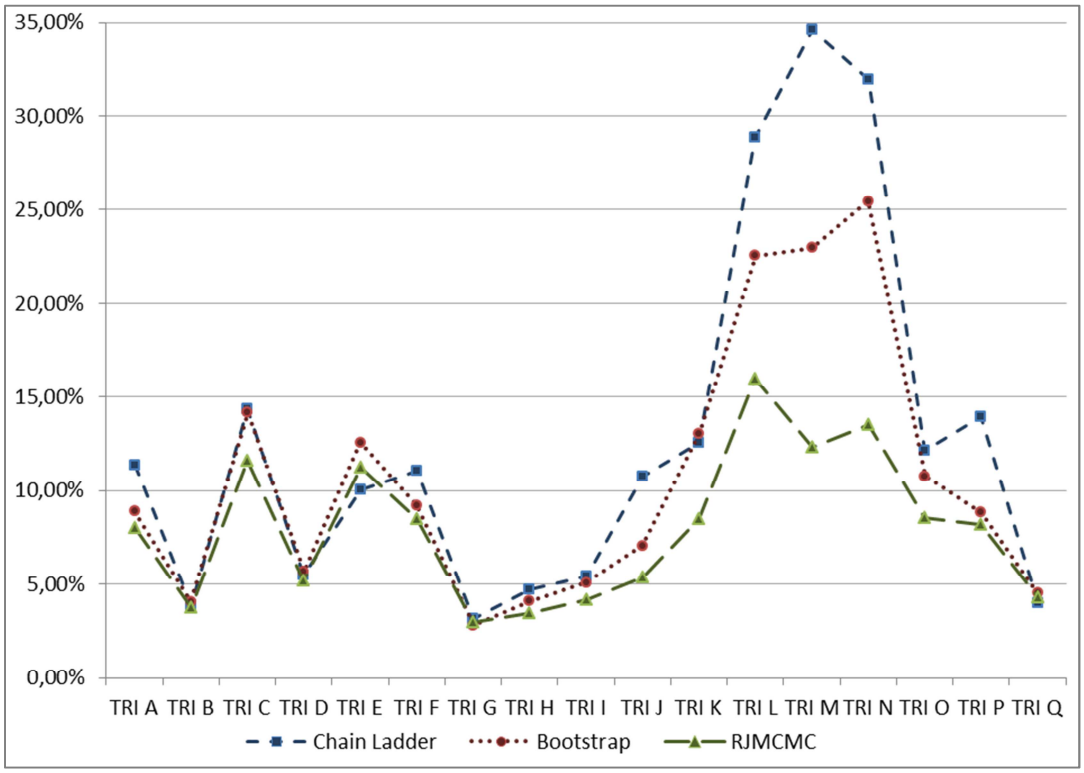


Figure 6: Coefficients of reserves variations

We can observe that for most triangles, the coefficients of reserves variations estimated by the RJMCMC algorithm are lower than the ones computed with the Bootstrap. It is also most of the time true for the Chain Ladder / Mack method with respect to the bootstrap method.

This seems quite logical because RJMCMC uses two different models: one for the right part and one for the left part of the triangle, these models being adapted to the number of data available. Thus, the volatility is reduced compared to the other methods for which a unique model is applied on the whole triangle, which increases the risk of uncertainty especially for the tail distribution.

As a conclusion, without any required manual adjustment:

- RJMCMC method leads to quite similar means in comparison with the other traditional methods;
- However, its coefficient of variation is often lower than the Mack and the Bootstrap methodologies.

3. EXTENSIONS OF THE TAIL DISTRIBUTION

One of the main commonly admitted advantages of the RJMCMC method is that it does not require any manual procedure from the expert because the algorithm will find by itself the best model to apply.

However, limiting it to an exponential decay for the right part of the triangle might seem a little bit restrictive; several other functions could be more adapted in some cases.

This is what we propose to study here with a measure which enables to get an idea of the goodness of fit for each other used function.

3.1 APPLICATION OF RJMCMC TO OTHER TAIL DISTRIBUTION FUNCTIONS

This section is dedicated to test other tail distribution functions than the exponential decay. Indeed, we propose to compare it against the power, inverse power and Weibull functions.

We propose to base our work on the classical curve fitting used to estimate the Loss Development Factors in the Chain Ladder methodology. Actually, it is possible to build a parallel between the formulas of the Loss Development Factors in Chain Ladder and the column parameters in RJMCMC. The main difference is that Chain Ladder deals with cumulative amounts whereas RJMCMC is based on incremental amounts.

Let f_j be the loss development factor of the development year j, with $f_0 = 1$

It is then possible to proof that γ_j is proportional to $(f_j - 1)$ according to a factor $g(j)$, g being a discrete increasing function from $[1,2, \dots, J]$ to $[a, 1]$, $0 < a \leq 1$. For high values of j (in the right part of the triangle) $g(j)$ is generally near 1 and therefore γ_j is not far from $(f_j - 1)$.

Starting from this conclusion, we propose to use the curve fitting formulas commonly used in Chain Ladder with the following transformations:

- Exponential function: $\forall j \in \{k, \dots, I\}\ \gamma_j = \exp(\alpha - j\beta)$, which is the one chosen by Verrall and Wüthrich (2012)
- Power function: $\forall j \in \{k, \dots, I\}\ \gamma_j = \alpha^{\beta j} - 1$
- Inverse power function: $\forall j \in \{k, \dots, I\}\ \gamma_j = \frac{\alpha}{j^{\beta}}$

- Weibull function: $\forall j \in \{k, \dots, I\}\ \gamma_j = \frac{1}{1-e^{-\alpha \times j^{\beta}}} - 1$

These functions share a desired behavior in consideration of the assumptions of the model. They are decreasing in j, they are convex and their limit when $j \to +\infty$ is equal to zero.

For each function we have to choose prior values for the two parameters α and β. The values themselves have no real impact on the final results as the aim of the algorithm is to converge to real estimates but chosen by trial and error, it may help a faster convergence.

The chosen values are:

- Exponential function: $a = -1$ and $b = 0.5$
- Power function: $a = 1.5$ and $b = 0.5$
- Inverse power function: $a = 0.5$ and $b = 1.5$
- Weibull function: $a = 1$ and $b = 0.5$

3.2 Computing the adjusted coefficient of determination for the different tail distribution functions

The aim is to run the algorithm with each function. In our case, we launch the algorithm four times changing the tail distribution function each time. For each function it is possible to compute the corresponding adjusted coefficient of determination. It is first necessary to estimate the coefficient of determination for which we compute two different terms.

The first one can be computed from the beginning because it only uses the input triangle. This is called the Total Sum of Squares and it consists in evaluating the variability of the initial data triangle. It is equal to the sum of the squared differences between each amount and the mean of all amounts:

$$SS_{Total} = \sum_{i=0}^{I} \sum_{j=0}^{I-i} \left(X_{i,j} - \bar{X}\right)^2$$

Where $\bar{X}$ represents the mean of all the incremental amounts contained in the input triangle. This last is computed with the formula:

$$\bar{X} = \frac{\sum_{i=0}^{I} \sum_{j=0}^{I-i} X_{i,j}}{\frac{I(I+1)}{2}}$$

The second term needed for the calculation of the coefficient of determination is called the Residual Sum of Squares. It consists in evaluating the variability of the residuals estimation. It is equal to the sum of the squared differences between each residual and the initial amount. Thus, it must be computed for each iteration:

$$SS_{Residual}^{(t)} = \sum_{i=0}^{I} \sum_{j=0}^{I-i} \left(\hat{X}_{i,j}^{(t)} - X_{i,j}\right)^2$$

Where $\hat{X}_{i,j}^{(t)}$ correspond to the expected amounts of the upper triangle. For the coefficient of determination, we only need to compute the mean of the expected amounts. Thus, the following formula is used to estimate the upper triangle:

$$\hat{X}_{i,j}^{(t)} \approx \mu_i^{(t)}\, \gamma_j^{(t)}$$

Finally, we calculate the coefficient of determination which uses the ratio between the total sum of squares and the residual sum of squares.

$$R^{2(t)} = 1 - \frac{SS_{Residual}^{(t)}}{SS_{Total}}$$

This coefficient has to be adjusted in order to take into account the number of estimated parameters:

$$R_{adj}^{2\ (t)} = 1 - \left(1 - R^{2(t)}\right) \times \frac{n-1}{n - p^{(t)} - 1}$$

Where n is the sample size: $n = \frac{(I+1)(I+2)}{2}$

The term $p^{(t)}$ is the number of parameters; this value changes at each simulation following the variations of the truncation index as the truncation index can move, $p^{(t)} = k^{(t)} + 2 + I$.

The adjusted coefficient of determination is then computed at each iteration so we get a distribution for this indicator. It is then possible to compute the mean and standard deviation of the adjusted coefficients of determination and several risk measures.

The aim of this methodology is to help the expert choose the function that fits the best the input data. This could be done following some simple rules as for instance: the more the adjusted coefficient of determination is closed to 1, the better the function is. Therefore it can be useful to build comparison between the mean of the adjusted coefficients of determination computed for each tail distribution function and choose the one which is the closest to 1.

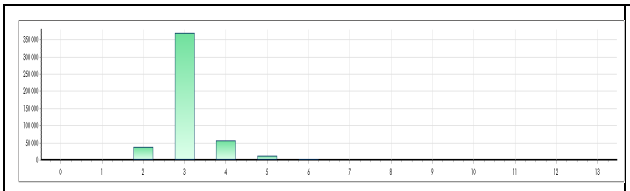
3.3 Applications

The aim is to run the algorithm with each function. In our case, we launch the algorithm four times changing the tail.

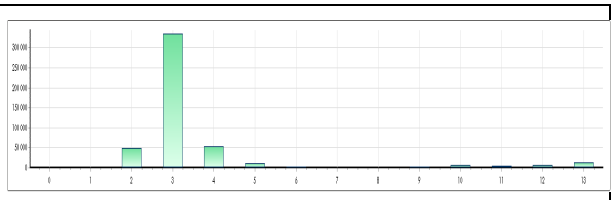
a. BNB Real Data example

For this example, we choose to use the "real data portfolio" which has kindly been provided by the Belgian supervisor *Banque Nationale Belge* (BNB). This triangle has been taken from real data (multiplied by a factor for remaining anonymous) of the Motor Liability line of business based on 14 years of history.

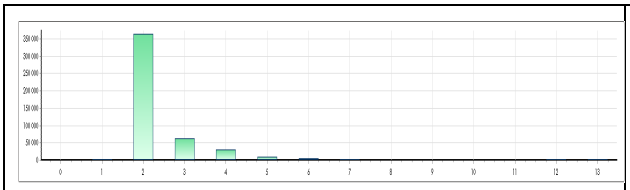
Once again we observe that the exponential and the power tail distributions have a similar behavior concerning the choice of the truncation index. Conversely, the inverse power and the Weibull tail distributions are similar to each other but different from the two first ones.



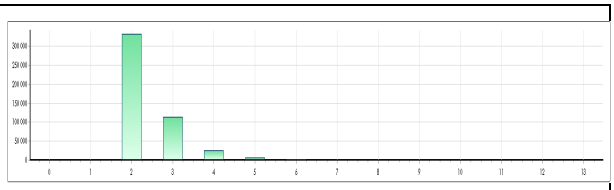
Exponential function



Power function



Inverse Power function



Weibull function

Figure 10: Example 2: Distribution of the truncation index for the different tail distribution functions

The behavior observed here is quite different from the previous case. Indeed, regarding the exponential or power functions the truncation index that presents the higher probability is $k = 3$, whereas with the inverse power or Weibull functions the truncation index that appears most of the time is $k = 2$.

Let's now consider the means, standard deviations and coefficients of variations of the reserves obtained with each function.

	Mean	Standard deviation	Coefficient of variations
Exponential	17 735 033	1 534 723	8,65%
Power	17 620 910	1 528 349	8,67%
Inverse Power	18 342 090	1 479 051	8,06%
Weibull	18 169 488	1 515 304	8,34%

Table 3: Example 2: Means, standards deviations and coefficients of variations of the reserves obtained with the different tail distribution functions

The lowest coefficient of variation is obtained with the inverse power function, whereas the highest corresponds to the use of the exponential function. As to the means of the reserves they are quite close even though the mean of the reserves computed with the inverse power function is higher than the other ones. In particular, it presents about 4% more than the mean of the reserves computed with the power function.

For example 2, we have estimated the means of parameters α and β for each function. We then have drawn the following curves.

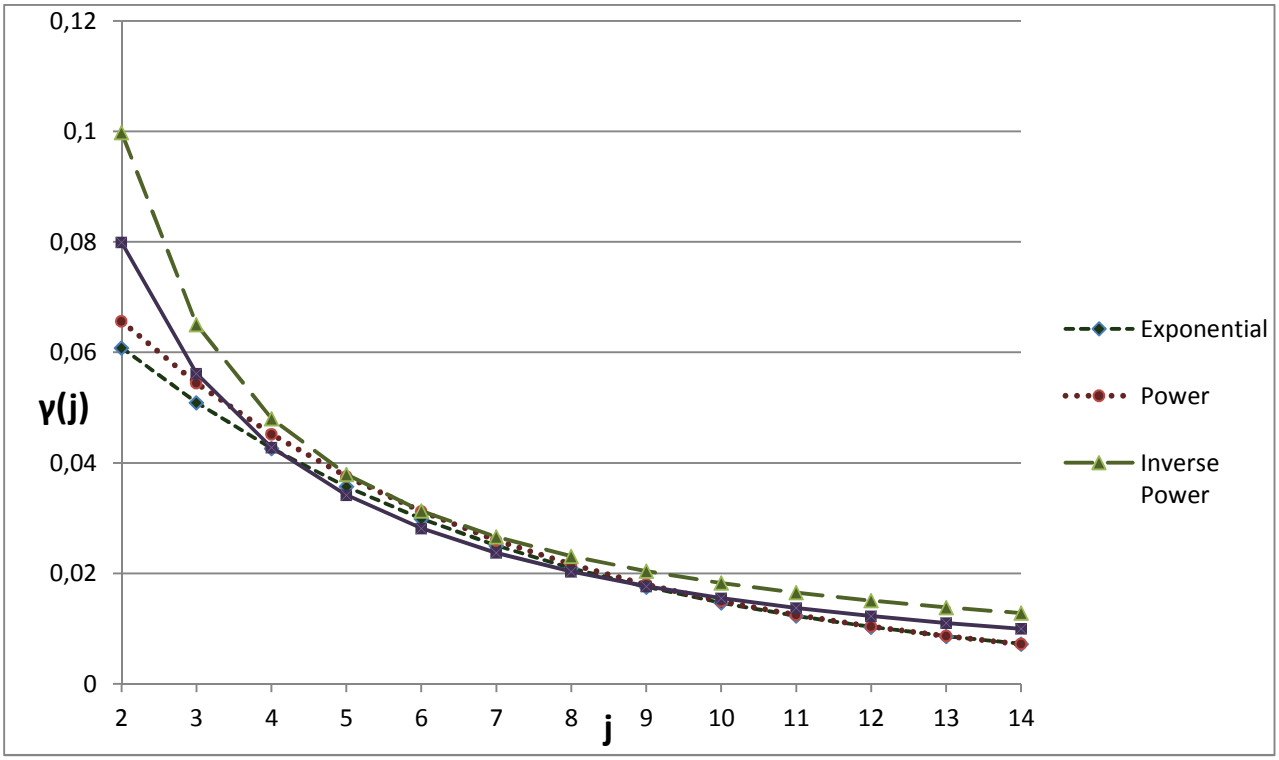


Figure 11: Example 2: Graph which represents the evolutions of each function for different values of *j*

For this example, the lowest truncation index that we observed is $k = 2$, so in this graph j goes from 2 to 14. Once again, the differences of behavior between the group composed by the inverse power and Weibull functions, and the group composed by the power and exponential functions explains the two different truncation indexes obtained above. The inverse power is the most prudent curve as it presents the highest values. Thus, it justifies that the highest mean is obtained with this function.

Let's now consider the adjusted coefficient of determination and the coefficient of reserves variations.

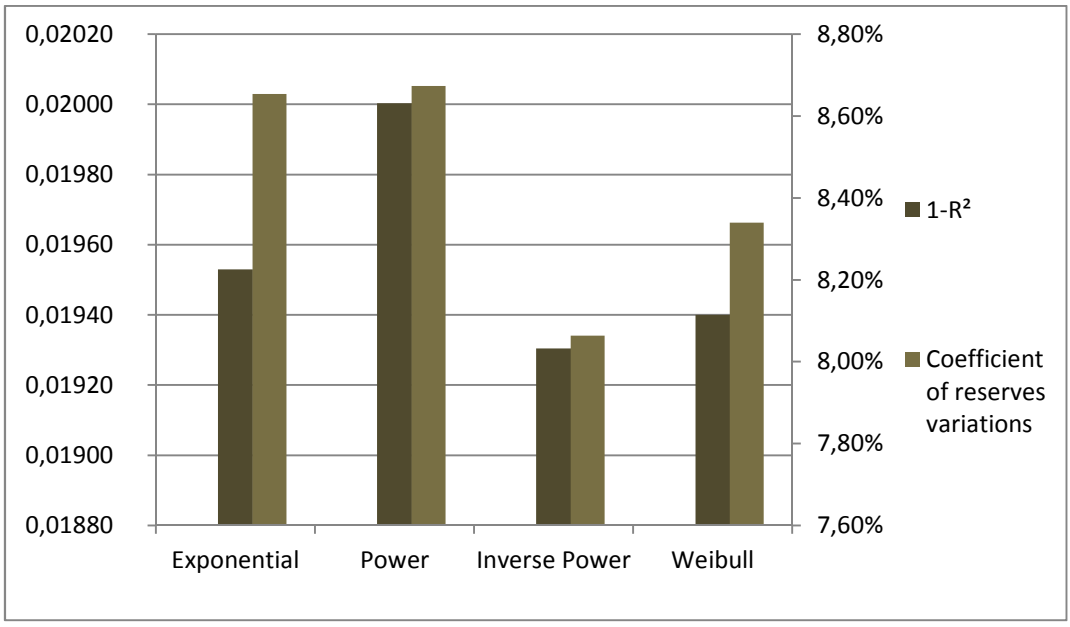


Figure 12: Example 2: Graph which represents the coefficient of reserves variations and $1 - R^2_{adj}$ for each function

In this example the inverse power function is the one that presents the adjusted coefficient of determination closest to 1. Even though, the mean of R^2_{adj} computed with the Weibull function is quite close with just 0.01% less. The power function seems to be the one that fits the worst the data as it has the lowest mean for the adjusted coefficient of determination.

For this example a negative correlation between the adjusted coefficient of determination and the coefficient of reserves variations can be observed.

Hence, the best choice seems quite obvious as the inverse power function presents at the same time the highest R^2_{adj} and the lowest coefficient of reserves variation.

b. Applications on 17 real data triangles

To make a wide test on a full set of market data, we applied these methodologies on the 17 triangles provided by the Belgian supervisor. The aim was to see which function with the highest adjusted coefficient of determination appears most of the time. We have therefore counted how many times each function was the best choice for each triangle.

The occurrences that we obtained are the following:

Tail distribution function	Occurrences on the 17 triangles
Exponential	7
Power	0
Inverse Power	8
Weibull	2

Table 4: Number of occurrences for which each function presented the best adjusted coefficient of determination

To conclude, we can say that for most triangles the function that has the highest adjusted coefficient of determination is the inverse power function. Then, comes the exponential function and over 17 triangles the Weibull function appears two times as the best tail distribution function. Finally, it seems that the power function does not fit very well the tail distribution as it never appears as the best fitting curve.

4. TOWARDS ONE-YEAR UNCERTAINTY

The capital calculation required by the Solvency II European directive is based on the one-year uncertainty valuation. However, at this stage RJMCMC is a methodology which allows the ultimate volatility. Thus, an extension of the algorithm is proposed in this section.

4.1 TRADITIONAL "ACTUARY IN THE BOX" METHOD

This methodology is the one commonly used to estimate the one-year uncertainty, in particular in the stochastic Bootstrap method. We propose to apply the same methodology to the RJMCMC algorithm.

The underlying idea of the "Actuary in the box" methodology consists in evaluating for each iteration the first diagonal of the lower triangle with one stochastic reserving method in order to get a new triangle with $D+1$

diagonals. Then, for each of these iterations the mean of the rest of the diagonals is estimated by applying again the same stochastic reserving method.

More precisely, starting from an input triangle with D diagonals, the methodology can be described by the following methodology, simulation by simulation:

- Estimation of the first diagonal of the lower triangle by applying one simulation of the underlying stochastic method;
- Construction of the triangle which contains $\boldsymbol{D+1}$ diagonals: adding the diagonal that has just been estimated to the input triangle;
- On the triangle with $\boldsymbol{D+1}$ diagonals: apply the underlying stochastic method with all required simulations to estimate the rest of the diagonals. At each iteration, of the stochastic method, compute the corresponding reserves;
- Store the mean of the obtained reserves.

These steps should be repeated for the chosen number of iterations. At the end of the entire method, we get a full distribution of the "means of the year+1 reserves" which correspond to the one-year uncertainty. From this we can derive the one-year mean , VaR, TVaR, confidence intervals or any other risk measure.

4.2 "ACTUARY IN THE BOX" APPLIED TO RJMCMC

In this section, the application of the "Actuary in the box" method to the specific case of RJMCMC is proposed.

The original functioning would consist in applying a new RJMCMC algorithm at each simulation, when a new triangle with $D+1$ diagonals has been previously estimated by one simulation of RJMCMC.

The scheme below describes the different steps of the method "Actuary in the box" applied to RJMCMC:

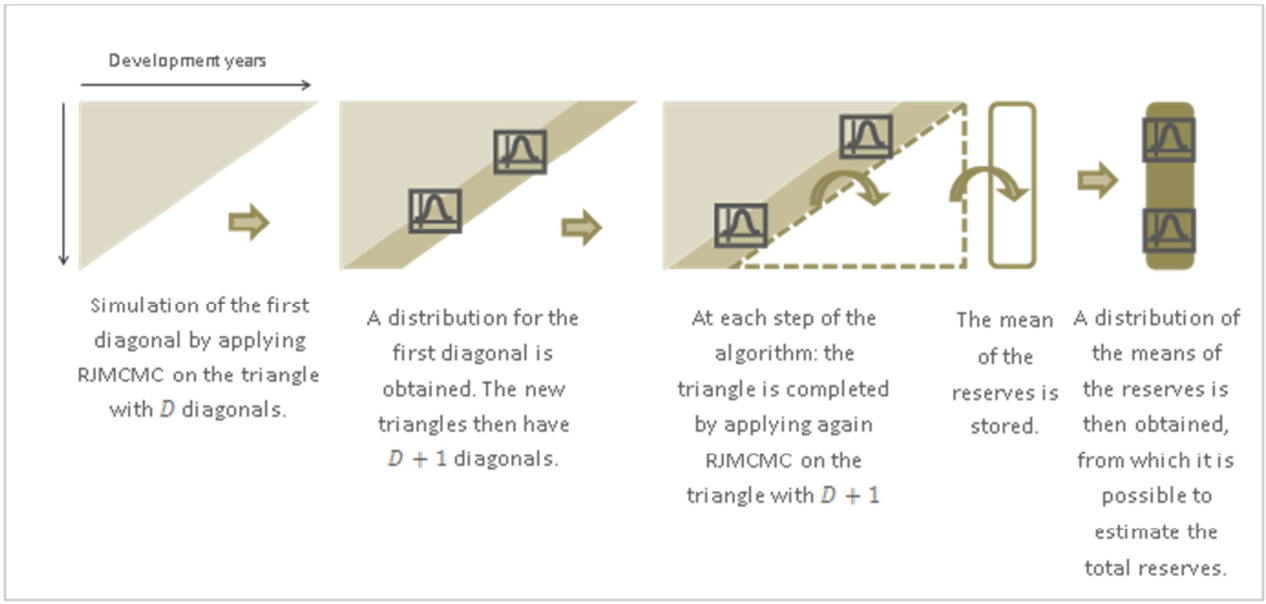


Figure 13: Scheme describing the functioning of the method « Actuary in the box » applied to the algorithm RJMCMC

In theory, the original method would require an important number of simulations (e.g. 1 million * 1 million which is over 10^{12}). The calculation times required would be huge in the context of the computers of today. This is why, for the practical application of the method in the next section, we made the same assumption as the one that is currently used when the "Actuary in the box" is applied to the Bootstrap. Thus, the estimation

of the first diagonal is done by the RJMCMC algorithm whereas the estimate of the mean of the rest of the diagonals is estimated by applying Chain Ladder instead of applying a whole RJMCMC. As Chain Ladder is a deterministic method, it does not require any simulation, so instead of requiring N^2 simulations, we just need N simulations. As the means between RJMCMC and Chain Ladder are quite similar (<1% difference in general), this assumption allowed us to value a proxy based on a large panel of triangles, without calculation time issues.

4.3 Applications

In this section, we compare the estimation of the reserves volatility obtained with different methodologies for the 17 triangles of the Belgian market: the Merz & Wüthrich ("Mack one-year"), the one-year bootstrapping, and the Solvency II standard formula.

a. Particular case of the Solvency II standard formula

We will use the coefficients provided by the European Insurance and Occupational Pensions Authority (EIOPA) to compute an estimation of the standard deviation and the Value at Risk 99.5% of the reserves under Solvency II requirements.

Depending on the considered Line of Business (LoB), the QIS 5 of the EIOPA gives directly coefficients to apply on the Best Estimate to get the standard deviation and the VaR 99.5%.

For the calculation of the standard deviation, the EIOPA gives the following rates, different for each LoB:

Standard deviation calculation per lob		Standard deviation for reserve risk		
Reserve risk	sres		Market	USP
Motor vehicle liability	9,5%		*9,5%*	
Motor, other classes	10,0%		*10,0%*	
Marine, aviation, transport (MAT)	14,0%		*14,0%*	
Fire and other property damage	11,0%		*11,0%*	
Third-party liability	11,0%		*11,0%*	
Credit and suretyship	19,0%		*19,0%*	
Legal expenses	9,0%		*9,0%*	
Assistance	11,0%		*11,0%*	
Miscellaneous	15,0%		*15,0%*	
Non-proportional reinsurance - property	20,0%		*20,0%*	
Non-proportional reinsurance - casualty	20,0%		*20,0%*	
Non-proportional reinsurance - MAT	20,0%		*20,0%*	

Table 5: Table provided by the EIOPA in relation to the standard deviations for each LoB

The Best Estimate of each LoB is then multiplied by the corresponding rate in order to get the standard deviation.

For the calculation of the VaR 99.5%, the EIOPA gives a formula which allows the deduction of new coefficients. These coefficients simulate the use of a log normal distribution based on the standard deviation values:

$$Coef_{VaR\ 99.5} = \frac{\text{Exp}\left(\mathcal{N}_{0,995} \times \sqrt{\text{LN}\left(Coef_{Std\ Dev}{}^2 + 1\right)}\right)}{\sqrt{\text{LN}\left(Coef_{Std\ Dev}{}^2 + 1\right)}} - 1$$

$Coef_{Std\ Dev}$ being the corresponding coefficient given by the EIOPA in the table 5.

$\mathcal{N}_{0,995}$ being the 99.5% quintile of the standard normal distribution.

In the following table, the values of the standard deviation and the VaR 99.5% are summarized by LoB:

	Standard deviation for the reserve risk	VaR 99,5% for the reserve risk
Motor vehicle liability	9,5 %	27 %
Motor, other classes	10,0 %	29 %
Marine, aviation, transport (MAT)	14,0 %	42 %
Fire and other property damage	11,0 %	32 %
Third-party liability	11,0 %	32 %
Credit and suretyship	19,0 %	60 %
Legal expenses	9,0 %	26 %
Assistance	11,0 %	32 %
Miscellaneous	15,0 %	45 %
Non-proportional reinsurance – property	20,0 %	63 %
Non-proportional reinsurance – casualty	20,0 %	63 %
Non-proportional reinsurance – MAT	20,0 %	63 %

Table 6: Table summarizing the coefficients provided by the EIOPA to compute standard deviations and VaR 99.5% of the reserves

Therefore the EIOPA provides rates directly applicable on the estimation of the reserves in order to compute the standard deviation and the VaR 99.5% of the reserves. These coefficients are based on the estimation of the one-year uncertainty.

b. Results

Each of the methods previously mentioned has an extension which allows us to estimate the one-year uncertainty:

- Mack: the extension brought by Merz & Wüthrich allows the estimation of the one-year uncertainty starting from Chain Ladder;
- Bootstrap: the application of the "Actuary in the box" allows the estimation of this volatility. For the practical application of this method in this part we made the assumption of Chain Ladder instead of using simulations inside simulations, as proposed in the reference papers;
- RJMCMC: we also applied the "Actuary in the box" method. As for the Bootstrap the assumption of Chain Ladder is made.

The graph below enables comparison of the capital calculated with these methodologies for the one-year uncertainty estimation:

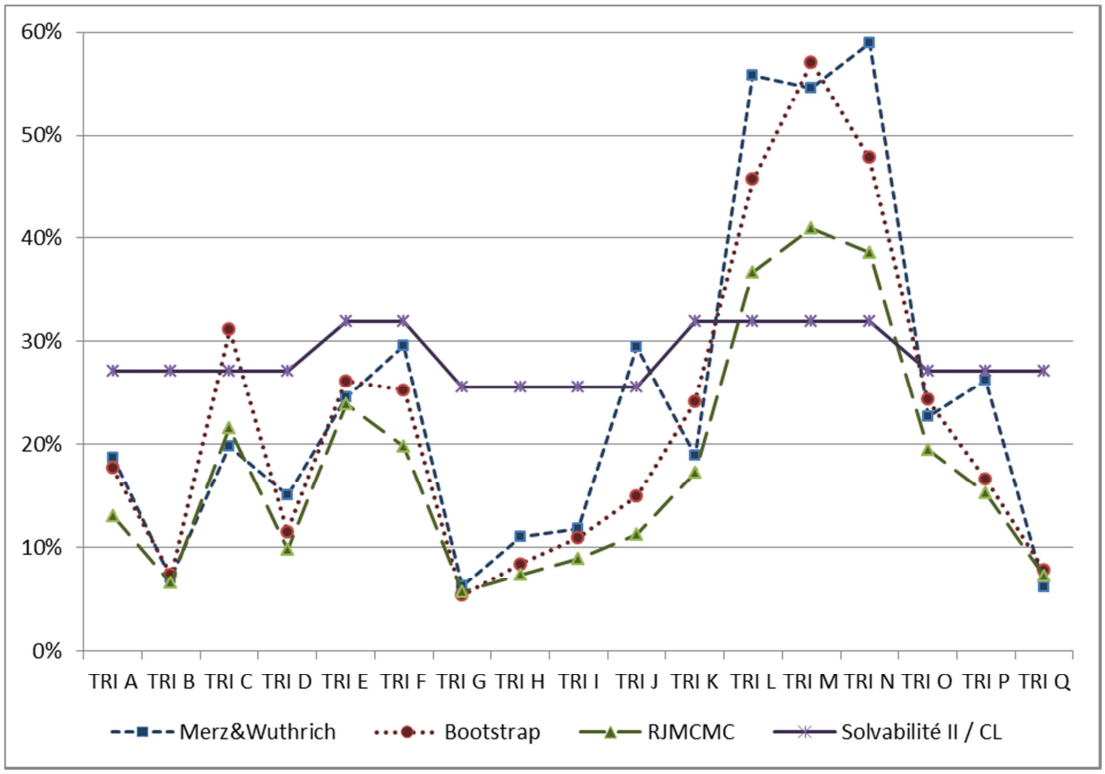


Figure 14: Graph representing capital as a percentage of the reserves for the different triangles

There are only three triangles over the seventeen for which the capital estimation by Solvency II is lower than the RJMCMC estimation. The RJMCMC results also lead to lower estimations than the Bootstrap results. It is also the case in comparison to the Merz & Wüthrich results, except for one triangle.

The table below represents the capital (as percentage of the mean of the reserves) obtained over the seventeen triangles and for each method. On the second row, a comparison is done with the results obtained with Solvency II.

	Mack	Bootstrap	RJMCMC	Solvency II
$\frac{Capital}{Mean\ of\ reserves}$	23 %	20 %	16 %	28 %
Differences with Solvency II	- 20 %	- 29 %	- 42 %	_

Table 7: Table summarizing the mean of capital obtained over the 17 triangles for the different methods

When these capitals including the one-year uncertainty are compared, it is possible to observe that the capital economy realized with the RJMCMC method is quite significant: 42% capital save in mean in comparison to the application of the Solvency II standard calibration. The capitals calculated with the Bootstrap come in the second place with a 29% capital save and the Merz & Wüthrich method represents a capital save of 20%.

CONCLUSION

In this paper several enhancements to the RJMCMC method originally presented by Verrall and Wüthrich (2012) are presented in order to use it on a wide set of real case triangles. This enables testing the method against the reality faced by insurers. The results obtained have been extremely encouraging. We get very plausible means without having to make any manual additional setup, and the standard deviation is lower than traditional methods. This follows logically, as the methodology uses two different models for the left and right parts of the triangle, instead of using one model which could lead to more uncertainty, especially in the tail.

Obviously, this methodology has drawbacks, e.g. it is based on the assumption of an Over-Dispersed Poisson (ODP) distribution, however, the improvements regarding the treatment of negative increments and increments equal to zero solve one of the biggest issues of the ODP.

This paper also describes the use of other parametric curves for the right part of the triangle; here it is often observed that the inverse power function gives better results than the exponential decay used in the original paper.

The final extension brought deals with the estimation of the one-year uncertainty and its comparison with calculations resulting from Solvency II European directive standard formula. This methodology leads to a substantial capital save, which can be explained in an intuitive way by the nature of the method applying adapted models on the right and on the left parts of the triangle.

Obviously, it is too early to say that this methodology is a genuine alternative to Chain Ladder. At this point, as it is quite new, it has not yet been tested intensively by insurance companies.

However, the results achieved applying the methodology to the set of triangles kindly provided by the Belgian regulator appeared quite promising. I hope it will encourage the readers of this paper to try this methodology on their own triangles, and I will obviously be more than happy to discuss their findings with them.

ACKNOWLEDGMENT

I would like to thank Mario Wüthrich, Pierre Miehé and Jose-Luis Vilar Zanon for their help during the development of the RJMCMC method, as well as the *Banque Nationale Belge* for providing me with a wide set of real case data.

REFERENCES

ENGLAND, P.D. (2010): Bootstrapping: Lessons Learnt in the Last 10 Years. Swiss Association of Actuaries

ENGLAND, P.D., and R.J. VERRALL (2001): A Flexible Framework for Stochastic Claims Reserving. Proceedings of the Casualty Actuarial Society 88: 1-38

ENGLAND, P.D., and R.J. VERRALL (2002): Stochastic Claims reserving in General Insurance. British Actuarial Journal 8/3: 443-518

ENGLAND, P.D., VERRALL, R.J. and WÜTHRICH, M.V. (2012): Bayesian Overdispersed Poisson Model and the Bornhuetter Ferguson Claims Reserving method. Annals of Actuarial Science, in press

ENGLAND, P.D., VERRALL, R.J. and WÜTHRICH, M.V. (2012): Bayesian Overdispersed Poisson Model and the Bornhuetter Ferguson Claims Reserving method. Annals of Actuarial Science, in press

EVERS, L. and JOHANSEN, A. M. (2007) : Monte Carlo Methods. Lecture Notes

GEMAN and GEMAN (1984), GELFAND and SMITH (1990): Gibbs Sampling

GREEN, P.J (1995): Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. Biometrika 82: 711-732

GREMILLET, M., P. MIEHE and J.L. VILAR ZANON (2013): A deep dive into RJMCMC, a practical alternative to Chain Ladder. ASTIN Colloquium

HASTINGS, W.K. (1970): Monte Carlo Sampling Using Markov Chains and Their Applications. Biometrika 57: 97-109

JOHANSEN, A.M., L. EVERS and WHITELEY, N. (2010): Monte Carlo Methods. Lecture Notes, Department of Mathematics, University of Bristol

KUNKLER, M. (2006): Modelling Negatives in Stochastic Reserving Models. Insurance: Mathematics and Economics 38: 540-555

MACK, T. (1993): Distribution-free calculation of the standard error of Chain Ladder reserve estimates. ASTIN Bulletin 23: 213-225

MACK, T. (1999): The standard error of Chain Ladder reserve estimate: recursive calculation and inclusion of a tail factor. ASTIN Bulletin 29, n°2: 361-366

MERZ, M. and M.V. WÜTHRICH (2008): Modelling the claims development result for Solvency purposes. ASTIN Colloquim, July 2008

TAYLOR, G. (2000): Loss reserving – An actuarial perspective. Kluwer Academic Press

VERRALL, R.J., HÖSSJER, O. and BJÖRKWALL (2012): Modelling Claims Run-off with Reversible Jump Markov Chain Monte Carlo Methods. ASTIN Bulletin, in press.

WÜTHRICH, M.V. and M. MERZ (2008): Stochastic Claims Reserving Methods in Insurance. New York: Wiley

WÜTHRICH, M.V. and R.J. VERRALL (2012): Reversible Jump Markov Chain Monte Carlo Method for Parameter Reduction in Claims Reserving. North American Actuarial Journal vol 16, number 2: 240-259