A deep dive into Reversible Jump Markov Chain Monte Carlo method, a practical alternative to Chain Ladder

By Marion Gremillet, José-Luis Vilar Zanón, Pierre Miehé

Contacts:
Marion Gremillet: marion.gremillet@actuaris.com
José Luis Vilar Zanon: jlvilarz@ccee.ucm.es
Pierre Miehé: pierre.miehe@actuaris.com

ABSTRACT

Most deterministic and stochastic reserves valuation based on triangles run-off use Chain-Ladder as a root; as is the case for Mack, Bootstrap, Bornhuetter Ferguson, Generalized Cape Cod, etc.
All of them apply a unique model to the cumulated triangles. However, the first columns contain a lot of data to estimate few points, whereas the last columns contain few data to estimate a lot of points.
The enhanced Reversible Jump Markov Chain Monte Carlo (RJMCMC) method presented in this paper relies on an Over-Dispersed Poisson model applied to incremental triangles. While a highly parameterized model is used for the left part of the triangle, the right part uses a two factor parametric model. All possible limits between the two parts of the triangle are tested.
The base of the presented approach follows the paper by Verrall and Wüthrich (2012). It is then enhanced to enable an application to a wide range of practical cases. In particular, this paper presents a way to deal with null or negative increments which are often found in real case triangles. It also tests other parametric models for the right part to better fit observed triangles and project the tail out of the triangle size.
The results of the cases in this paper illustrate that this enhanced RJMCMC is able not only to deal with all studied triangles, but also that it can result in less volatility than other methods such as Bootstrap or Mack. Therefore, the use of this RJMCMC model benefits insurers by highlighting a potential capital save, especially in the context of the risk measures imposed by the current solvency regulations (e.g. the famous VaR99.5% of the European Solvency II regulations).

Keywords

Monte Carlo, Markov Chain, Metropolis-Hastings, Gibbs, Bayesian Over-Dispersed Poisson, RJMCMC, Incremental Triangles, Reserves valuation, Stochastic reserves, Dynamic Financial Analysis, Tail Valuation, Value at Risk, Reserves Risk, Solvency II.

1. 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 us 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.

The fundamental assumption of the method is that each incremental value of row i and column j, denoted $X_{i,j}, 0 \leq i, j \leq I$ follows an over dispersed Poisson distribution with parameters: $\left(\frac{X_{ij}}{\varphi} \middle| \vartheta\right) \sim Poi(\frac{\mu_i \gamma_j}{\varphi})$ where φ is the over-dispersed parameter, μ_i is a row parameter and γ_j is a column parameter. Hereafter, we will consider that both the occurrence years of the underlying claim and the development years are numbered from 0 to I.
In section 2 we present the different assumptions of the model and explain the functioning of the algorithm. Section 3 describes two methodologies which allow managing with negatives and incremental values equal to zero, which is required to apply RJMCMC to most real case triangles. Section 4 proposes some extensions of the model by applying different functions to model the right

part (or "tail") of the triangle. In section 5 we provide documented examples and then conclude by highlighting the advantages and utility of the method.

2. 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.

2.1 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. Let $X_{i,j},\ 0 \leq i,j \leq I$ 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 $D_I^C = \{X_{ij}; i+j > I, 0 \leq i \leq I, 0 \leq j \leq I\}$, based on the upper triangle, defined by $D_I = \{X_{ij}; i+j \leq I, 0 \leq i \leq I, 0 \leq j \leq I\}$.

- 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:

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

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 $\boldsymbol{\Theta}^{(t)} = \left(k^{(t)}, \theta_{k^{(t)}}^{(t)}\right)$.

2.2 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

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

  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

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

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:

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

Finally, two cases are possible:

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

- 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:

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

  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:

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

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:

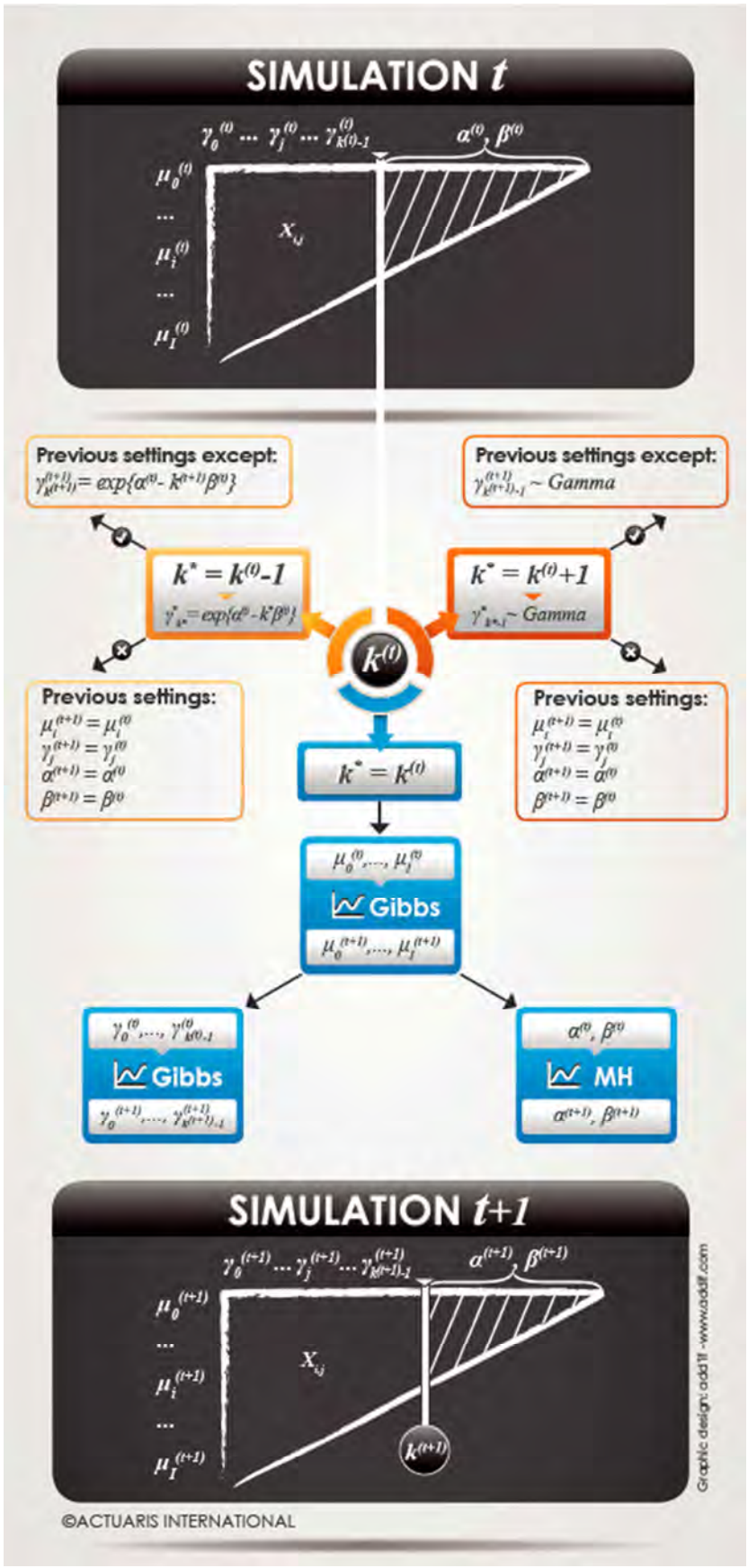

Figure 1: Functioning of the core of the RJMCMC algorithm

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.
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.

2.3 Impact of the number of simulations

We used professional software[1] which allowed us to easily test the method with different random seeds and strong random generators.
For the following results we used the Real Data example presented in the paper of Verrall and Wüthrich (2012).

We launched the algorithm ten times with different random seeds for several number of iterations: 10,000 ; 100,000 ; 500,000 ; 1,000,000 ; 2,000,000.
For the series of 10,000 simulations the burn-in was fixed to 2,000 but for all the other series it was fixed to 20,000.

We summed up the results that we get for the means of the reserves, the coefficients of reserves variations and the VaR 99.5%. We summarize the results in the following figures:

[1] IBNRS provided by Actuaris International

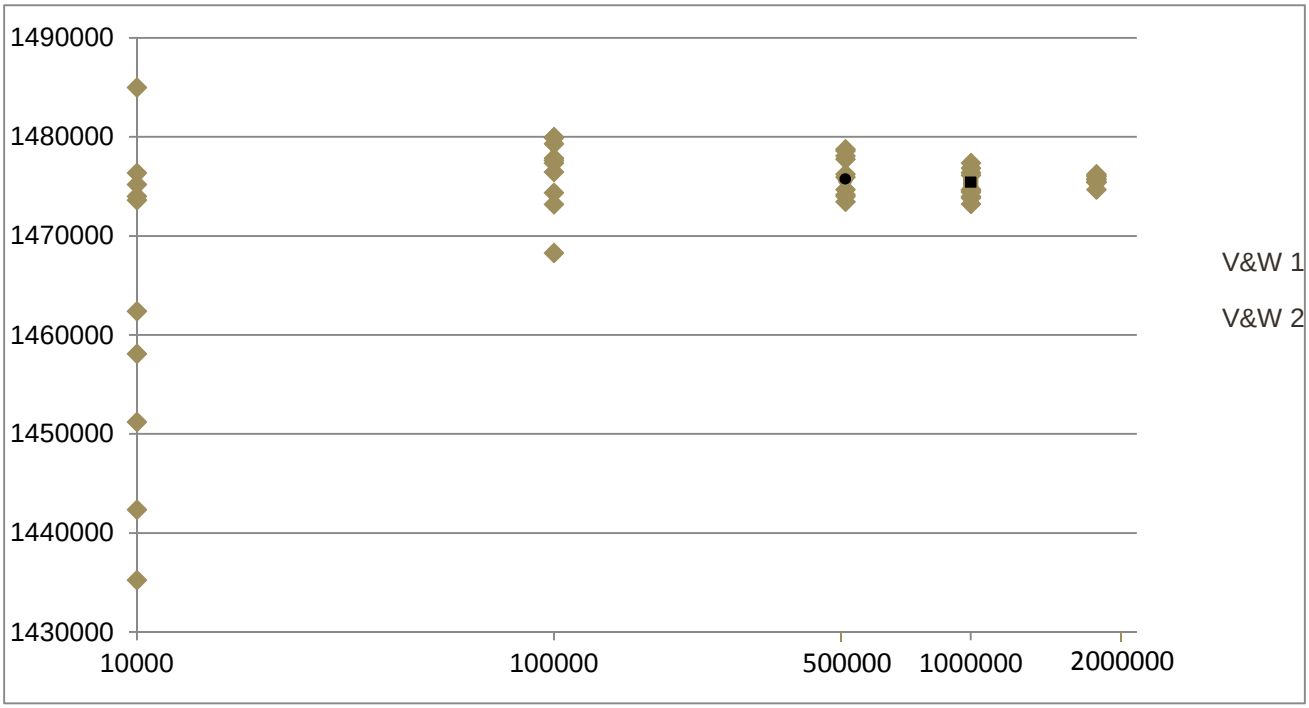

Figure 2: Mean of the reserves (Y-axis) obtained for different numbers of simulations (X-axis)

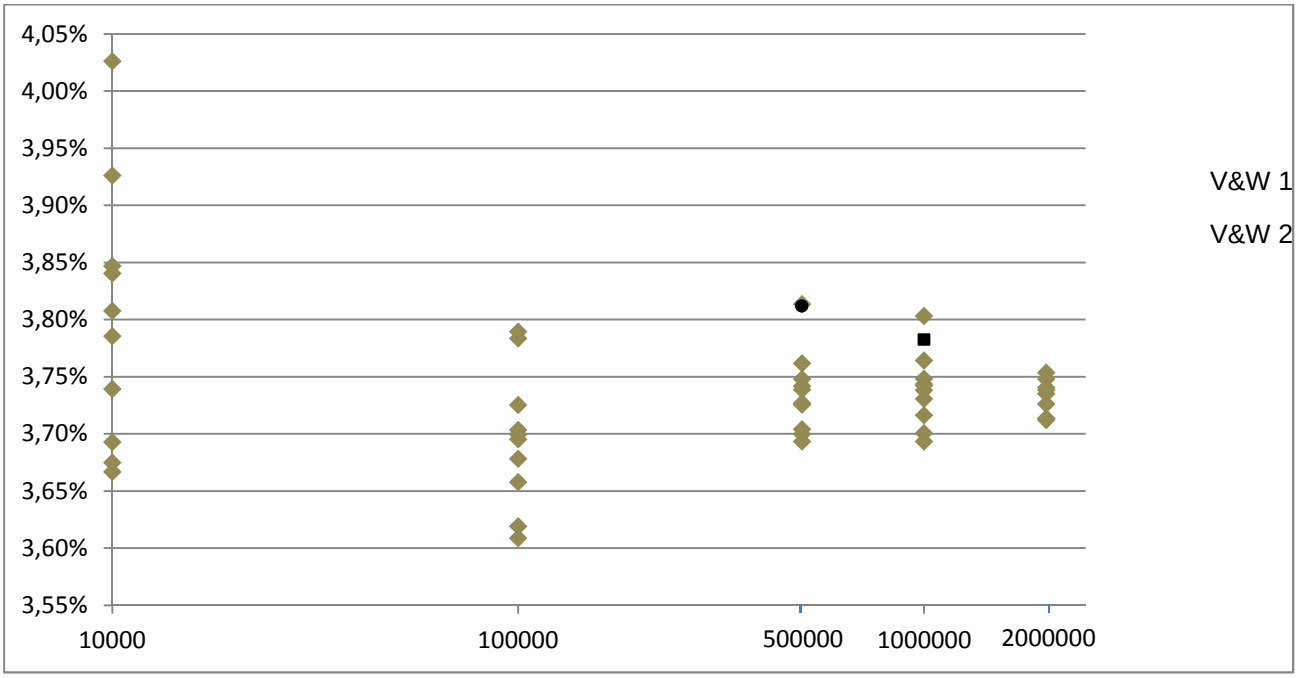

Figure 3: Coefficients of reserves variations (Y-axis) obtained for different numbers of simulations (X-axis)

On the figures 1 and 2, two points have been plotted: V&W 1 which corresponds to the values provided by Verrall and Wüthrich (2012) in their first paper about RJMCMC and V&W 2 which corresponds to the values provided by the previous authors in their second paper about RJMCMC.

The first results have been computed by using 1,000,000 simulations and the second results have been computed over 500,000 simulations.

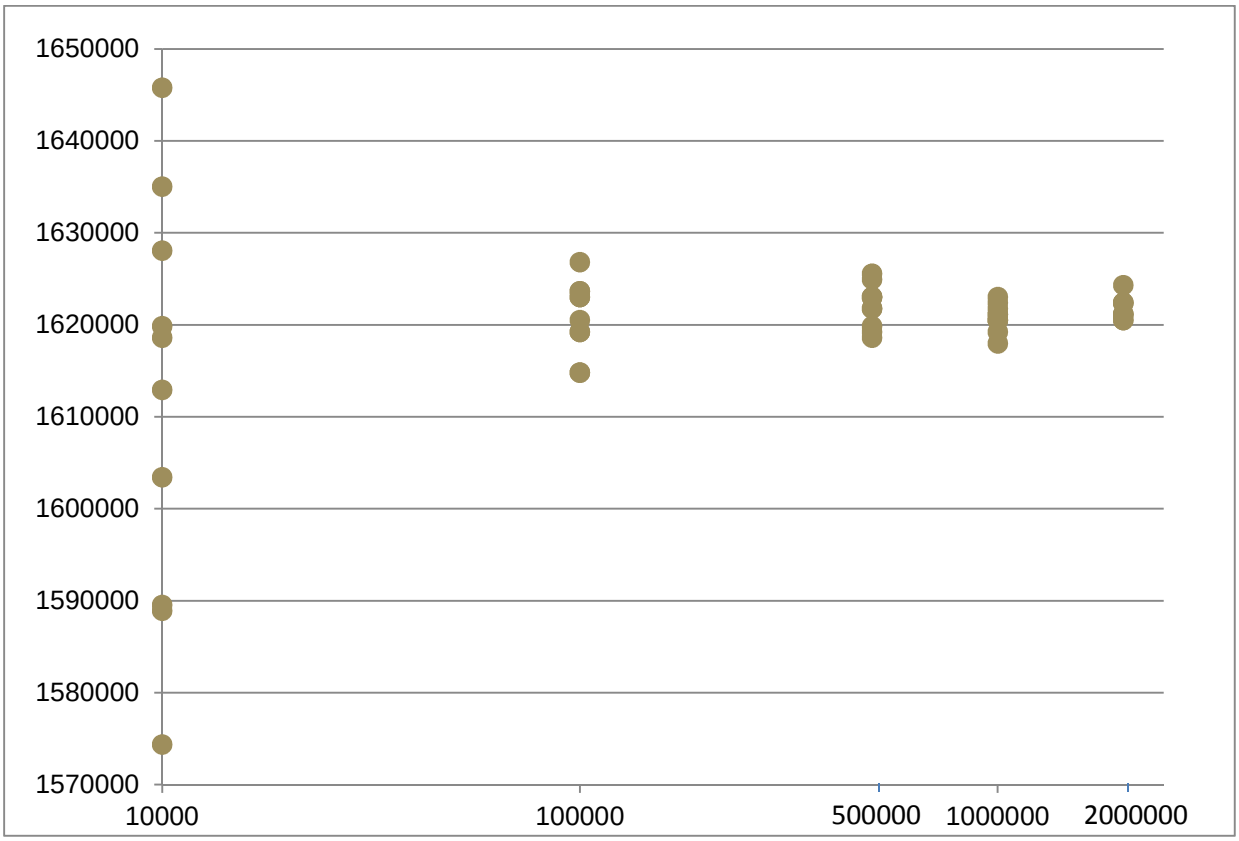

Figure 4: VaR 99.5% of the reserves (Y-axis) obtained for different numbers of simulations (X-axis)

We can observe a quick convergence of the results. This is especially true when the number of iterations grows from 10,000 to 500,000. After 500,000 simulations, increasing the number of simulations has a lesser impact on the convergence.
This trend was also observed for the other market triangles we studied. It seems that 500,000 simulations are a good compromise between speed and precision.

3. Modeling exclusions and negative values

Negative incremental amounts or increments equal to zero are typical issues when using over-dispersed Poisson models. Unfortunately, this often happens especially with incurred triangles.
In this section we propose some solutions to these problems for being able to apply the RJMCMC algorithm on nearly any kinds of real case input triangles.

3.1 Excluding values

The aim of this part is to propose a methodology which allows the expert to exclude some values of the initial data. Indeed, it is sometimes useful to have the ability to exclude cells of the input loss

data triangle from the model. For that purpose, we define an indicator function for each amount of the input triangle.
We build a triangle which represents the indicator function of each amount:

$w_{0,0}$	$w_{0,1}$	…	…	$w_{0,I}$
$w_{1,0}$	$w_{1,1}$	…	$w_{1,I-1}$	
…	…	…		
…	$w_{I-1,0}$			
$w_{I,0}$				

Table 1: Triangle of the indicator functions for each cell of the input triangle

The values of terms $w_{i,j}$ can be 1 if the corresponding amount is not excluded or 0 if the corresponding amount is excluded.
These indicator functions are then applied in each formula where the initial amounts $X_{i,j}$ are involved; each $X_{i,j}$ is replaced by the product $w_{i,j}X_{i,j}$. Thus, if the indicator is equal to 1 there will be no changes in the formulas. Conversely, if it is equal to 0 the corresponding amount will have no impact in all the formulas of the algorithm.

We then compute the indicator functions for each row and each column. If all the cells of a row or a column are excluded then the indicator function of the corresponding row or column will be equal to 0.
Let wc_j be the indicator function for the column j. If all the cells of the column j are excluded then the column j will be excluded. If $\forall i \;\; w_{i,j} = 0$ then $wc_j = 0$, otherwise the column j is included and $wc_j = 1$.
Let wr_i be the indicator function for the row i. If all the cells of the row i are excluded then the row i will be excluded. If $\forall j \;\; w_{i,j} = 0$ then $wr_i = 0$, otherwise the row i is included and $wr_i = 1$.

Thus, each time we compute a sum on rows we multiply the term of the sum by wr_i.
For instance the following sums are transformed:
$\sum_{i=0}^{\mathrm{I}} \mu_i^{(t)}$ becomes $\sum_{i=0}^{\mathrm{I}} wr_i \mu_i^{(t)}$

Each time we do a product on rows we put an exponent equal to wr_i.
For instance the following products are transformed:

$$\prod_{i=0}^{I}\left[\frac{e^{-\frac{\mu_i^{(t)}\gamma^*_{k(t)}}{\varphi}}\left(\gamma^*_{k(t)}\right)^{\frac{X_{ik(t)}}{\varphi}}}{e^{-\frac{\mu_i^{(t)}\gamma^{(t)}_{k(t)}}{\varphi}}\left(\gamma^{(t)}_{k(t)}\right)^{\frac{X_{ik(t)}}{\varphi}}}\right] \quad \text{becomes} \quad \prod_{i=0}^{\mathrm{I}}\left[\frac{e^{-\frac{\mu_i^{(t)}\gamma^*_{k(t)}}{\varphi}}\left(\gamma^*_{k(t)}\right)^{\frac{w_{i,k(t)}X_{ik(t)}}{\varphi}}}{e^{-\frac{\mu_i^{(t)}\gamma^{(t)}_{k(t)}}{\varphi}}\left(\gamma^{(t)}_{k(t)}\right)^{\frac{w_{i,k(t)}X_{ik(t)}}{\varphi}}}\right]^{wr_i}$$

Similar changes are made for the columns exclusions.
Thus, for computations of sums or products on columns, the indicator functions wc_j are used as multiplicative terms or as exponents depending on the case.

Case of null sums of columns and rows amount:
One of the main limitations of over dispersed Poisson distributions is that the sums of the incremental amounts in every column and in every row of the input triangle have to be different from zero.

We can generalize the exclusion methodology presented before to a complete column or row to solve this issue. Thus, if a column or a row contains only incremental amounts equal to zero we can exclude it from calculations and automatically set its weight to zero.

3.2 Modeling Negative values

Not only, increments equal to zero can lead to some problem in over-dispersed Poisson models but negative incremental amounts also represent an issue in such 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: $S_j^{(-)} = \{X_{i,j} \mid 0 \leq i \leq I - j \; and \; X_{i,j} < 0\}$. 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: $S_j^{(+)} = \{X_{i,j} \mid 0 \leq i \leq I - j \; and \; X_{i,j} \geq 0\}$. 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:

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

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.

3.3 Applications

We have applied RJMCMC with these extensions to a whole set of market anonymous triangles kindly provided by the Belgian supervisor (*Banque Nationale Belge*). Indeed, we have run RJMCMC on 18 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 1 to 18.

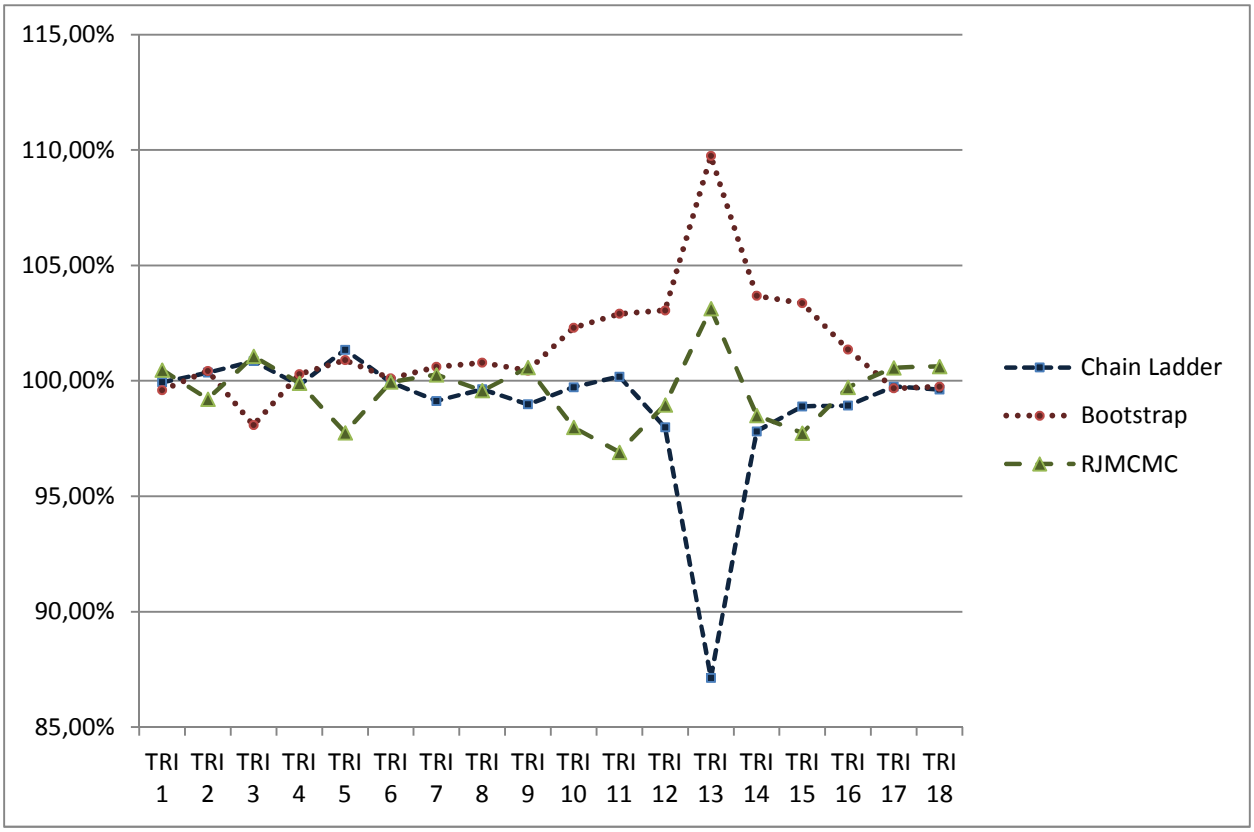

Figure 5: Means of the reserves

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. For triangle 13 the Chain Ladder method seems quite optimistic compared to the two other methods.

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

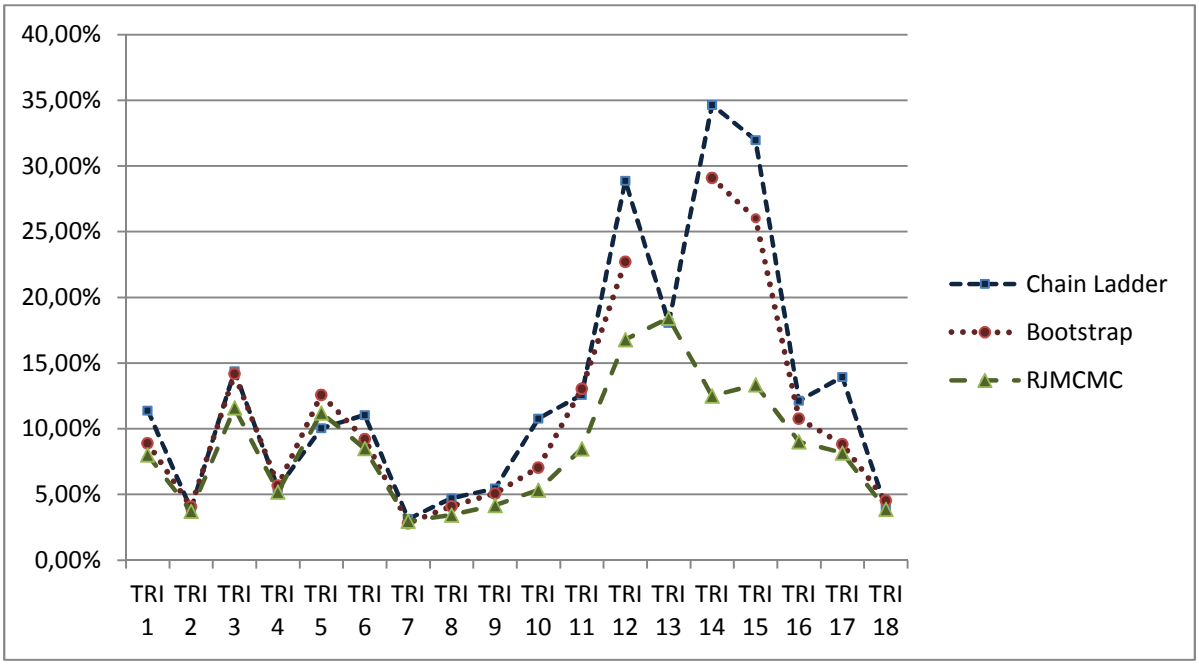

Figure 6: Coefficients of reserves variations

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.
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.
For triangle 13, the coefficient of variations obtained with the Bootstrap method is not represented as the methodology could not be applied to this specific triangle.

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.

4. 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.

4.1 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.

To illustrate it, some notations could be introduced:
Let U_i be the ultimate amount for the origin year i
Let $C_{i,j}$ be the cumulative amount for the origin year i and the development year j
Let f_j be the loss development factor of the development year j, with $f_0 = 1$

We now build a comparison between the estimation of the cumulative amounts with the two methods.
With the Chain Ladder method the cumulative amount of row i and column j is estimated by:

$$\hat{C}_{i,j} = X_{i,0} \times f_0 \times \dots \times f_j$$

With the RJMCMC assumptions the incremental amount of row i and column j is estimated by:

$$\hat{X}_{i,j} = U_i \times \gamma_j$$

Thus, the estimation of the cumulative amount of row i and column j is:

$$\hat{C}_{i,j} = U_i \times \sum_{n=0}^{j} \gamma_n$$

We can make equality between these two estimations so we get:

$$X_{i,0} \times f_0 \times \dots \times f_j = U_i \times \sum_{n=0}^{j} \gamma_n$$

As $X_{i,0} \times f_0 = U_i \times \gamma_0$ so we can simplify and delete the term U_i. Then, the formula above becomes:

$$\gamma_0 \times f_1 \times \dots \times f_j = \sum_{n=0}^{j} \gamma_n$$

This can be written:

$$f_j \times \sum_{n=0}^{j-1} \gamma_n = \sum_{n=0}^{j} \gamma_n$$

We can conclude that $\gamma_j = (f_j - 1) \times \sum_{n=0}^{j-1} \gamma_n$ for j>0.
Therefore, 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: $\forall j \in \{k, \dots, I\}\ \gamma_j = \frac{1}{1-e^{-\alpha \times j^\beta}} - 1$

γ_j

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 a good choice 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 $b = 0.5$

4.2 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 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.

5. Examples

In this section the presented methods are tested on two real data examples. Thus we compare the results obtained with the different tail parametric functions. We used four times the RJMCMC algorithm, one time for each function. We have used 500,000 simulations following Section 2.3 conclusions. However, as we have seen earlier, it is possible that some differences between random seeds can impact the final results. Thus, for the following examples we decided to compute the mean of five runs each of them with 500,000 simulations. We then present in the third part of this section the total results obtained for the data provided by the Belgium supervisor.

5.1 R. Verrall and M. Wüthrich Real Data example

For the first example, we choose to use the real data example provided by Verrall and Wüthrich (2012).
The use of different tail distribution functions implies that in some cases the truncation index will not converge to the same value.
We observe that the exponential and the power tail distributions have a similar behavior concerning the choice of the truncation index, whereas the inverse power and the Weibull tail distributions are similar to each other but different from the two first ones.

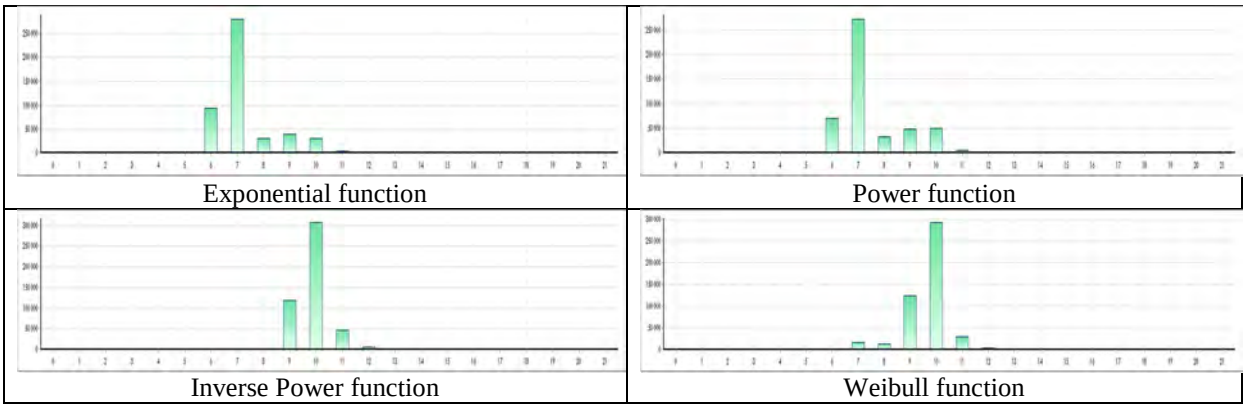

Figure 7: Example 1: Distribution of the truncation index for the different tail distribution functions

Concerning the exponential and the power functions the truncation index seems to become stable for the value , whereas for the inverse power and Weibull functions the highest probability is obtained for the truncation index .
It is then possible to compute the mean and the standard deviations of the reserves for each function. The comparison between the coefficients of reserves variations is the following:

	Mean	**Standard deviation**	**Coefficient of variations**
Exponential	1 476 794	54 840	3,71%
Power	1 470 727	55 889	3,80%
Inverse Power	1 485 757	52 608	3,54%
Weibull	1 460 584	55 260	3,78%

Table 2: Example 1: Means, standards deviations and coefficients of variations of the reserves obtained with the different tail distribution functions

The results are quite similar, even if we can observe that the inverse power function gives the highest mean whereas the Weibull gives the lowest which is often the case due to the structure of the functions. Regarding standard deviation it is more the goodness of fit which seems to impact the results.

To illustrate the behavior of the functions, we have estimated the means of parameters and for each function. Then, we have drawn the evolution of each function depending on the index of the column .

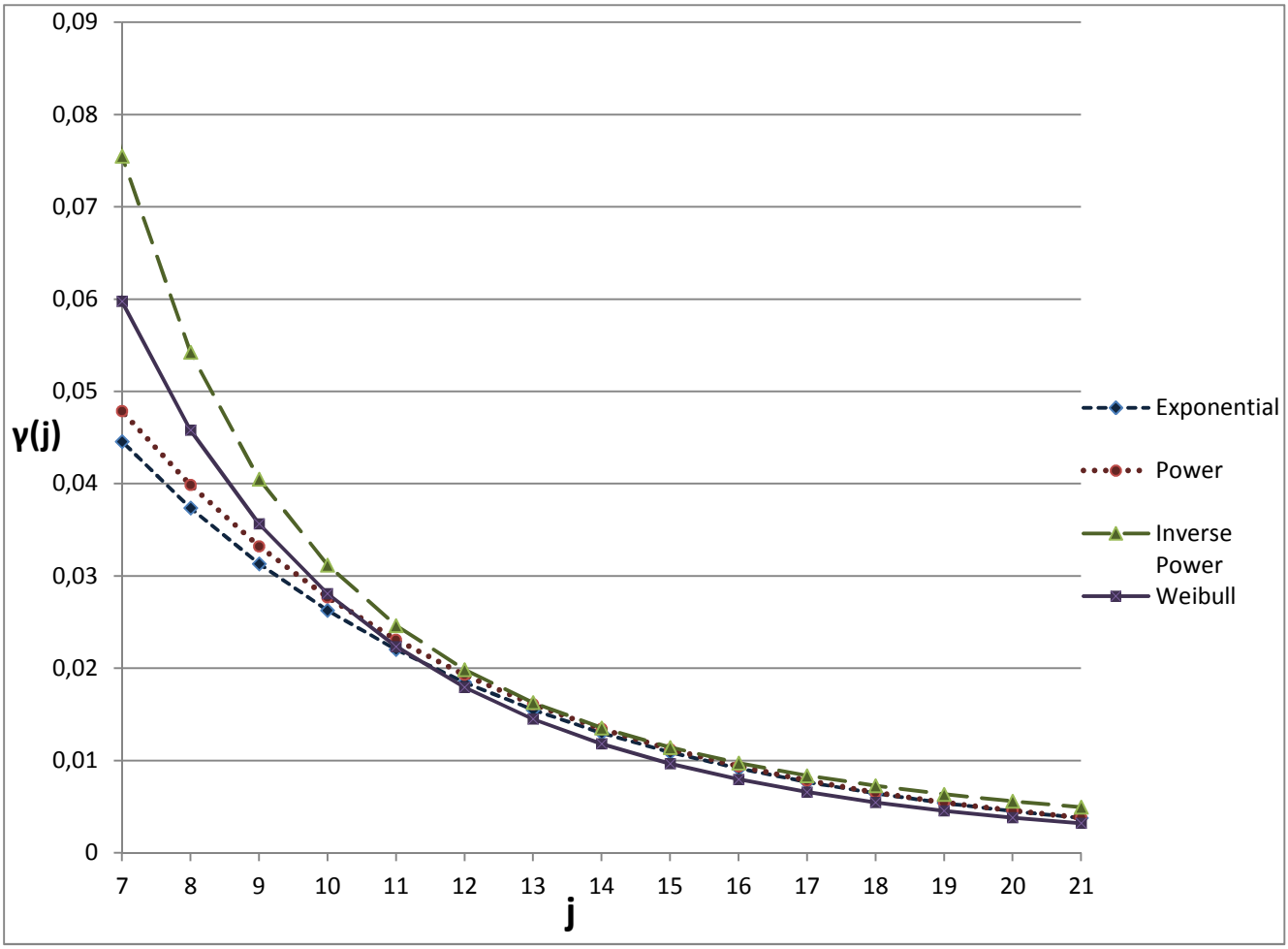

Figure 8: Example 1: Graph which represents the evolutions of each function for different values of *j*

The graph has been drawn for different values of j from 7 to 21 as the lowest observed truncation index is $k = 7$. The inverse power and the Weibull have a similar behavior which confirms the trend observed above concerning the choice of the truncation index. Conversely, the power and exponential functions are close to each other, especially for $j = 7, \dots 10$. The inverse power function presents is the highest, it explains why the highest mean of reserves is obtained with this function.

In order to choose the best function to fit the right part of the data, let's build a comparison between: the coefficient of reserves variations, which is equal to the ratio between the standard deviation and the mean; and the mean of $1 - R^2_{adj}$. We then can analyze whether there exists a correlation between these values.

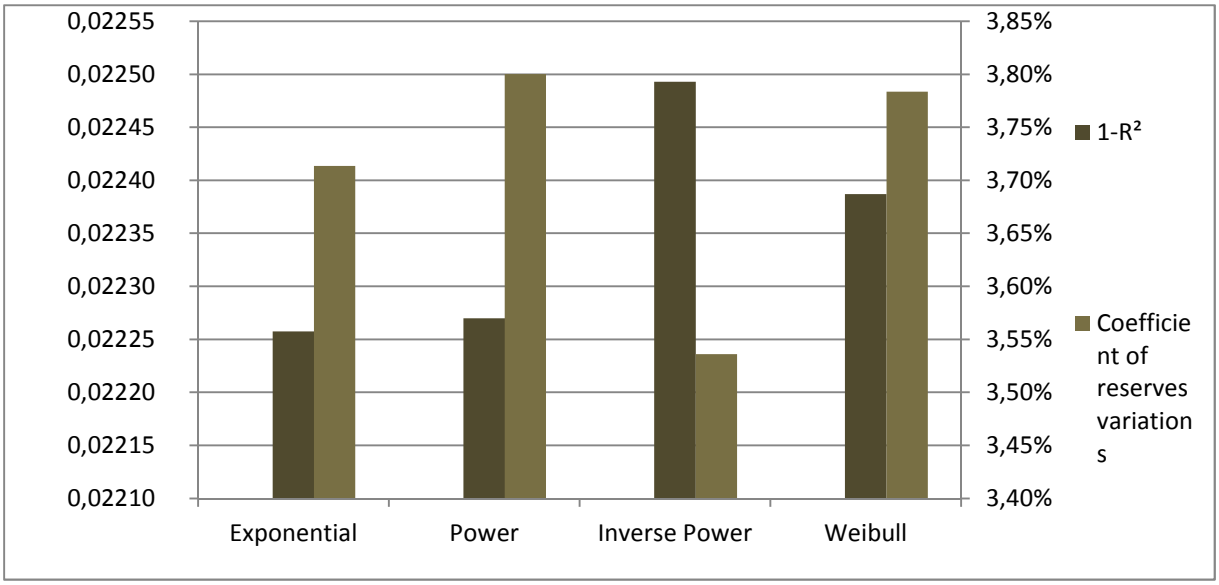

Figure 9: Example 1: Graph which represents the coefficient of reserves variations and $1 - R^2_{adj}$ for each function

The means of the adjusted coefficients of determination are very close, which is normal as the calculation is performed on the whole triangle and not only on the right part. Therefore, we should not consider the R^2_{adj} absolutely but relatively. Hence, the R^2_{adj} computed when the tail distribution is the exponential or power functions are a little bit greater than those computed with the inverse power or the Weibull functions of about 0.02%. In this example this means that the exponential is the function that fits best the data as the corresponding adjusted coefficient of determination is the highest.

Regarding the coefficient of determination, we can see two groups of fit: the inverse power and Weibull functions on one side, and the exponential and power functions on the other side. This follows logically, as the mean of k is different for these two groups and therefore, the adjusted coefficient of determination, being dependent on the number of parameters, is different.

This graph helps the user choose the best function to fit the right part of the column parameters. The function which has the best adjusted coefficient of determination (the closest to 1) and the lowest coefficient of reserves variations is the best function to be used.

For example 1 it is quite difficult to make differences between all the functions because they are very close to each other especially regarding the value of R^2_{adj}. Therefore, we can focus on the coefficient of determination which is the lowest for the Inverse Power; this could be the logical choice.

5.2 BNB Real Data example

For the second 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.

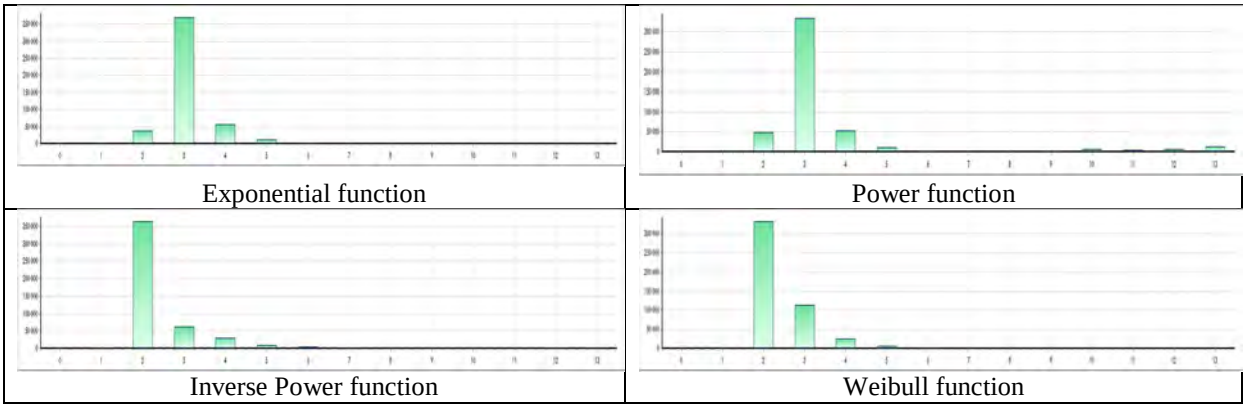

Figure 10: Example 2: Distribution of the truncation index for the different tail distribution functions

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 , whereas with the inverse power or Weibull functions the truncation index that appears most of the time is .

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 and for each function. We then have drawn the following curves.

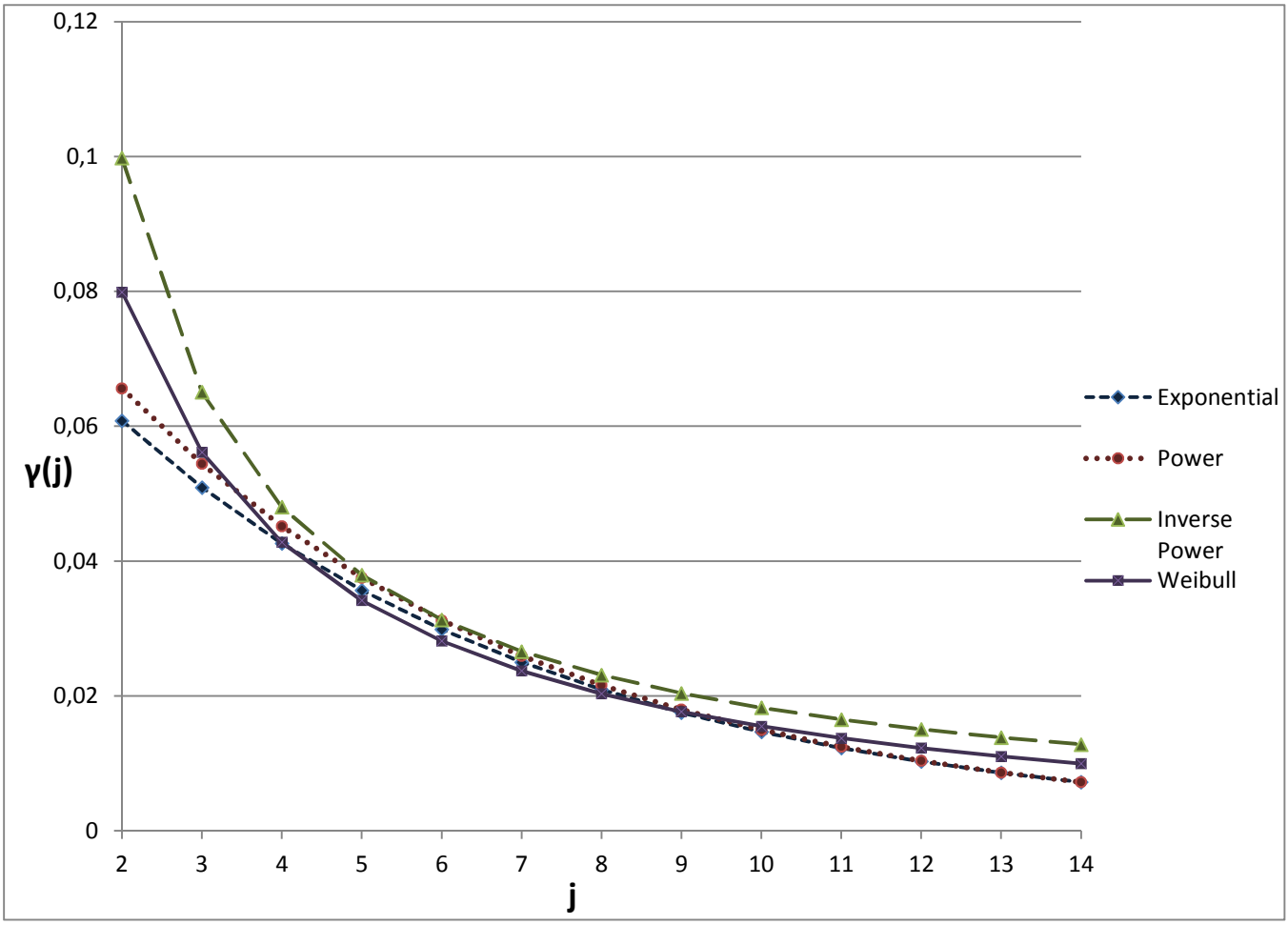

Figure 11: Example 2: Graph which represents the evolutions of each function for different values of *j*

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.

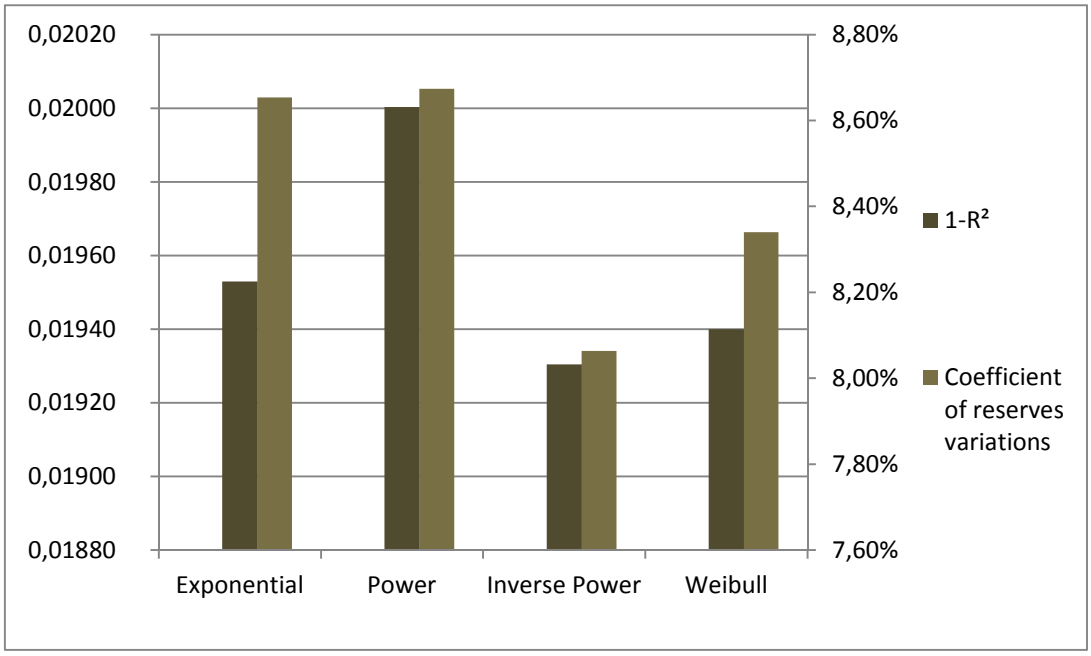

Figure 12: Example 2: Graph which represents the coefficient of reserves variations and $1 - R^2_{adj}$ for each function

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.

5.3 Applications on 18 real data triangles

To make a wide test on a full set of market data, we applied these methodologies on the 18 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 18 triangles
Exponential	7
Power	0
Inverse Power	8
Weibull	3

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 18 triangles the Weibull function appears three 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.

6. Conclusion

In this paper, we presented several enhancements to the RJMCMC method originally presented by Verrall and Wüthrich (2012) to use it on a wide set of real case triangles. This enables us to test the method against the reality faced by insurers. The results we 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. We make the assumption of an Over-Dispersed Poisson (ODP) distribution, however, our 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 we have often observed that the inverse power function gives better results than the exponential decay used in the original paper.

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

AKNOWLEDGMENT

We would like to thank Mario Wüthrich for his help during the development of the RJMCMC method, as well as the *Banque Nationale Belge* for providing us with a wide set of real case data.

REFERENCES

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

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

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

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