OPTIMAL REINSURANCE PER EVENT

by

Carl Axel Jean Pierre HOLTSMARK

MASTER THESIS

for the degree

Master of Science

(Master i Modellering og Dataanalyse)

The Faculty of Mathematics and Natural Sciences
University of Oslo

May 2015

Det matematisk- og naturvitenskapelige fakultet
Universitetet i Oslo

Abstract

Most insurance companies deal with reinsurance. One of the problems they have to solve is: What reinsurance treaty is optimal for their company? Optimal reinsurance for large portfolios has, during the past decades, been given much more attention in the academic world than individual claims have. In this thesis we will investigate the optimal reinsurance contract for the individual claims case.

The thesis will start by introducing some basic concepts in reinsurance. There will be a brief explanation of reinsurance mathematics. We will establish mathematical formulations of the different types of reinsurance contracts and the optimality criteria. Then we will see some of the existing literature on the topic and show some own results. This is going to point us towards the optimal reinsurance contract: the non proportional a x b contract with retention limits a and b where b is infinite.

We will introduce the reader to the Panjer recursion. The recursion will be used as a numerical tool to simulate the a x b contract. We will vary different key parameters and see how they affect the criteria and the retention limits. These results will back our assumption of the a x b contract with infinite b as the optimal one.

Acknowledgements

This is a thesis for the Master's degree in Modelling and Data Analysis (Modellering og Dataanalyse), with the programme option in Finance, Insurance and Risk (Finans, Forsikring og Risiko). This is a long thesis which corresponds to 60 credits and was done in the period between October 2013 and May 2015.

First of all, I would like to thank my thesis supervisor Erik Bølviken, he has been a great teacher and have provided me with the Fortran programs needed for the Panjer simulations. Besides having him as my supervisor, I have had the pleasure of taking 4 of his courses during my university years. I still remember the first course I had with him, which was STK2520: Problems and methods in Actuarial science. This course sparked my interest in actuarial science and is the main reason for my choice of studies. His presence in the university has been of great importance in my life.

I would also like to thank my current boss in PSS-GEO, a seismic processing company in Oslo, Jon Sandvik, for letting me use the office as a study place and giving me flexibility between work and studies.

I would also like to thank my parents and family for supporting me throughout my University years and always believing in me. They have kept me motivated and given me unconditional help when I needed it. Lastly, I want to mention my girlfriend Astrid, who has been a great listener and motivation during the last months of this thesis.

Carl Axel Holtsmark
May 2015, Oslo

Contents

1 Introduction

1.1 The world of reinsurance

Most of the insurance contracts are subject to reinsurance. Which is an insurance contract purchased by an insurance company, the cedent, from a reinsurance company. It can be property insurance, car insurance, or even workplaces (oil rigs, boats etc...). Half century ago, most reinsurance contracts were agreed upon in a mathematically primitive way, mostly based on "hunches". Nowadays it's a different story. The last two decades have seen a large number of unexpected and costly events, the 1999 storm in France, the terrorist attacks of 9/11, the 2004 and 2005 hurricanes of the United States etc...

All these events have pushed the industry to re-evaluate the way the prices of reinsurance (the premiums) are calculated. As the reinsurance business is global, a catastrophic and costly event such as hurricane Katrina in 2005 (the National Hurricane Center estimated the costs to 108 billion dollars), caused the reinsurance premiums to a rise not only in the United States, but also in the rest of the world. As we can see, we are dealing with an issue that is difficult to predict and often subject to changes according to world events. The table below shows the top ten reinsurers in terms of gross premium. The nationalities of the companies illustrate the international aspect of reinsurance.

1	Munich Reinsurance Company	$ 31,280	Germany
2	Swiss Reinsurance Company Limited	$ 24,756	Switzerland
3	Hannover Rueckversicherung AG	$ 15,147	Germany
4	Berkshire Hathaway Inc.	$ 14,374	USA
5	LLoyd's	$ 12,997	United Kingdom
6	SCOR S.E.	$ 8,872	France
7	Reinsurance Group of America Inc.	$ 7,201	USA
8	Allianz S.E.	$ 7,201	Germany
9	PartnerRe Ltd.	$ 4,881	Bermuda
10	Everest Re Group Ltd.	$ 4,201	Bermuda

Table 1: Top 10 reinsurance companies in the world, ranked by gross premiums written in 2010, it is taken from A.M. Best Co's website.

On average, an insurance company will lose money on reinsurance. However, there are several benefits. The insurance world is subject to a lot of uncertainty. Reinsurance is an important tool to cope with this uncertainty. It is a great way for an insurance company to protect itself against catastrophic events. There are several examples in history of events that have ruined insurance companies. One example is the 1906 San Francisco earthquake. This event bankrupted twenty companies. It also deleted the profit that American fire insurers made in the previous 47 years (Source: Aetna Life Insurance history www.aetna.com). Reinsurance may also reduce the capital requirements. The cedent will need less capital to satisfy the solvency directives. This can also increase the value per share of the company and allow the cedent to issue more policies. Other benefits include tax deductibility of reinsurance contracts, financing "startup" insurance companies, profit from the reinsurer's expertise ...

There are many different types of reinsurance contracts. They are divided into two main groups: the proportional contracts and the non-proportional contracts. The following paragraphs will explain how they work.

Proportional The concept of proportional reinsurance is as follows. The insurer (the cedent) and the reinsurer will agree to an assignment rate between 0% and 100% for all risks in the portfolio. In order to determine the reinsurance premium, we apply the assignment rate to the original premium. When we have a claim, the assignment rate is applied to the claim size to decide how much the reinsurer will give.

The assignment rate varies according to the insurance policies. In the proportional reinsurance world there are two major types of contracts:

- Quota share, the reinsurer shares an equivalent share of the premiums and claims of the cedent's portfolio. The quota share can be modified according to sub-portfolios defined in the contract. This becomes a varying quota share contract type.
- Surplus share, the assignment rate varies according to a "line" (a money amount). If the claim size exceeds this line, the reinsurer assumes the difference between the total amount and the line. The surplus share can also be modified and transformed into other slightly different contracts.

Non Proportional In non-proportional reinsurance, there is no fraction determining how the premiums and the losses will be shared between the cedent and the reinsurer. How much the reinsurer will pay depends on the amount of losses. The reinsurer and the cedent will instead agree upon the retention (also known as the priority). When the amount of losses exceeds the retention, the reinsurer takes over the financial compensations relative to the losses up to a certain limit (also agreed upon contract).

The reinsurance premium is the price the cedent pays the reinsurer for the cover it provides. In non-proportional, the reinsurer must anticipate the potential losses in order to fix a premium. This requires more sophisticated techniques than proportional reinsurance since there is no proportionality between premium and losses. Therefore, the reinsurer will receive more information about the potential losses. The actuarial models used are more advanced and are becoming more popular. The two major types of non-proportional contracts are stop loss and excess of loss:

- Excess of loss, also called the a x b contract (a times b). The contract covers the share of losses exceeding the retention and up to a certain limit (fixed in the contract). There are two distinct types of excess of loss contracts, one is per "event" (here an event is the cause of several losses, terrorist attacks, hurricanes etc...). The other one is per portfolio (car insurance portfolio, an aircraft etc...).
- Stop loss, similar to the excess of loss contract. It covers a percentage of the aggregate yearly losses over the retention and up to a certain limit. For example, in a 50% xs 70% (the percentages express the ratio losses to premiums), the reinsurer covers losses exceeding 70% with a maximum of 50%.

1.2 Optimal reinsurance

The topic of optimal reinsurance for the cedent was already researched upon in 1940 by the Italian mathematician de Finetti, see Wahlin (2012). He worked on optimal proportional reinsurance by minimizing the variance of the gain given a fixed gain. His results were pointing in the direction of a quota share type of contract. In 1960, the Norwegian professor Borch suggested an optimal contract by maximizing the reduction of the variance in the claim distribution of the cedent for a given net premium. He showed that, under fairly restricted conditions, the stop loss type of contract was the optimal one. The world of reinsurance has changed substantially since these papers were written and the topic needed an update. Nowadays, when dealing with optimal reinsurance, there are several criteria we must optimize with respect of. There has been a substantial increase in computational abilities which allows us to simulate with greater speed and accuracy. The reinsurance industry has changed a lot since the middle of the twentieth century.

In the last decade, the literature on optimal reinsurance in property and casualty insurance (P&C insurance) has been growing. With a notable summary of the prior results in Centento and Simoes (2009). Only a small part of the literature is focusing on the individual claim case. For example the papers of Dickson and Waters (2006) and Centento and Guerra (2010). However, the case of reinsurance for portfolio aggregates have so far received much more attention.

The idea that the deciding parts of an insurance company are going to base their choice of reinsurance contract only on a mathematical argument is probably a little far-fetched. Nevertheless, the research on this topic can help them make better decisions, based on good evidence. By carefully choosing the optimality criteria, the decision makers can consider the results from the different criteria and choose the reinsurance contract satisfying their company's needs. Cai and Wei (2012) give solutions in terms of a utility function. Although the utility functions are interesting in a theoretical sense, their impact in the industry is rather limited as they are sometimes based upon unclear conditions, which are not always as easy to interpret in practice. In Chung, Sung, Yam and Yung (2011), they use the expected profits against "Value at Risk" (V@R, although it expresses almost the same value as the reserve based on percentiles, V@R is not to be confused with the reserve ϵ that we will introduce later on) as the criteria.

As of today, the preferred methods for reinsurance optimizing seem to be expected profits against variance (stability of the results) and expected profits against total solvency capital required (the reserve ϵ).

In this thesis, we are going to focus on the optimal reinsurance contract **per event**. As we have seen, there are several papers addressing the topic of optimal reinsurance, however most of them are on the portfolio level, and not per event. They are also excessively theoretical and do not focus on the practical aspects of the reinsurance industry. Decisions need to be made. It is therefore interesting to focus on these individual events and see if they coincide with the current literature. Our simulations in section 4 will try to answer the following question: **Which reinsurance contract is optimal for the cedent when we focus on individual events?**

I will start this thesis by introducing some insurance concepts and notation. The different types of contracts and also the Monte Carlo method for simulating the claims. Then, I will show some results already existing in the literature about this topic and also show some results developed independently from the literature. Finally, I will introduce the reader to the Panjer recursion for compound distribution. This recursion will be used as a tool for optimization. I will focus on the non-proportional contract and the expected profits against solvency capital requirements criteria for results.

2 Reinsurance Mathematics

In order to understand the mathematics behind reinsurance, it is essential to explain how insurance loss works. Doing this, we can establish a mathematical relationship between the cedent and the reinsurer. From the introduction we understand that the losses are organised this way:

$$\begin{array}{ccccc} Z & \longrightarrow & Z^{ce} & \longrightarrow & Z^{re} \\ \text{First Clients} & & \text{Cedent} & & \text{Reinsurer} \end{array}$$

An insurance company (the cedent) and a person (or company) agree upon a contract, also known as a policy which makes the insurance company economically responsible for incidents which affect the item or person that is insured. In this theses, the amount that the insurance company must pay to the person in case of a claim will be called Z. The insurance company will have several policies, this makes the total of the Z highly uncertain. The same way as the first client cedes his risk to the insurer, the insurer can cede his risk to a reinsurer. The reinsurer's risk Z^{re} is formulated as a function of the insurer's risk $H^{re}(Z) = Z^{re}$. The cedent's net risk Z^{ce} is then defined as $Z^{ce} = H^{ce}(Z) = Z - Z^{re} = Z - H^{re}(Z)$. H^{re} and H^{ce} can be considered as the same function but with different point of views. They are defined by contract and the reinsurers and cedent's net risk Z^{re} and Z^{ce} are directly affected by the insurance total net risk Z.

2.1 Basic formulations

There are two uncertainty factors for the insurance company. The claim frequency and the claim intensity. We want to have a model for the insurance claims that will predict the expected amount the insurance company is accountable for:

$$X = Z_1 + Z_2 + ... + Z_N$$

where N is the number of claims and $Z_1, Z_2, ...$ the amount of each claims.
The claim frequency is often modelled with the Poisson distribution. The Poisson distribution is a discrete probability distribution which expresses the probability that a number of events will happen independently in a predefined time interval. The probability mass function is

$$P(N = n) = \frac{\lambda^n}{n!} e^{-\lambda}$$

where the mean is $\mathrm{E}[N] = \lambda$ and the standard deviation $\mathrm{sd}(\lambda) = \sqrt{\lambda}$. It seems fit to express the claim number of a policy N as poisson distributed with parameters $\lambda = \mu T$. The intensity is an average over time and policies, see Bølviken (2014), page 283.

In order to model the claim size Z, there are several distributions that could seem adequate. In this thesis we want to deal with individual claims that occur rarely, but are very costly. Since these claims occur rarely, it would be wrong to try to fit a distribution using historical events. The non-parametric approach being ruled out because of the scarceness of historical data, we should use the parametric approach. We must then find a distribution that agrees with these principles. To model the claim size there are several potential distributions, for example the Log-normal, Pareto, Gamma etc ... For our problem, which is to find the optimal reinsurance, the distribution choice should not be a crucial factor. Once we have build a program that takes the distributions as input, it will be easy to simulate with different distributions and vary their parameters.

In the introduction we explained how different types of reinsurance contracts function. In order to simulate the effects of reinsurance, it is important to express these contracts mathematically. In the previous section, we expressed the cedent's net loss. Since the reinsurance function $H^{re}(Z) = Z^{re}$ is derived from the cedent's net loss we can give mathematical formulas for the reinsurance contracts. Since the reinsurance function is equal to $Z - H^{ce}(Z)$, it should always satisfy $0 \leq H^{re}(Z) \leq Z$.

The **proportional** contracts are the quota share type of contracts and the surplus share. In the quota share contract, the total net risk is shared using a fixed percentage $0 \leq c \leq 1$, the risk kept by the cedent and the reinsurer are $H^{ce}(Z) = c(1 - Z)$ and $H^{re}(Z) = cZ$.

The surplus share contracts are mathematically a little different, define a as the retention limit of the cedent, s the maximum insured sum. If one claim exceeds the retention limit, the reinsurer pays the difference between the claims and the limit a. The percentage ceeded is then $c = \max(0, 1 - \frac{a}{s})$. Which gives us from the cedent's point of view:

$$H^{ce}(Z) = \begin{cases} Z & \text{if } a \geq s \\ a & \text{if } a < s \end{cases}$$

and from the reinsurer's point of view:

$$H^{re}(Z) = \begin{cases} 0 & \text{if } a \geq s \\ (1 - \frac{a}{s})Z & \text{if } a < s \end{cases}$$

The **non proportional** contracts are the excess of loss and stop loss. The excess of loss contract is as follows, assume that the cedent's net risk is $\sum_{i=1}^{N} H^{ce}(Z_i)$ where N is the number of claims, $Z_1, Z_2, ...$ independent claims for incidents, $1, 2, ...$ and $H^{ce}(Z)$ the risk kept by the cedent. The contract is then as follows (from the cedent's point of view):

$$H^{ce}(Z) = \begin{cases} Z & \text{if } Z < a \\ a & \text{if } a \leq Z \leq a + b \\ Z - a & \text{if } Z \geq a + b \end{cases}$$

from the reinsurer's point of view the same contract will look like this:

$$H^{re}(Z) = \begin{cases} 0 & \text{if } Z < a \\ Z - a & \text{if } a \leq Z \leq a + b \\ b & \text{if } Z \geq a + b \end{cases}$$

The stop loss contracts are similar to the excess of loss contract, the difference is that the limit b is undefined and can even be considered as infinite, see Bølviken (2014), page 367. These non-proportional contracts will be called a x b contracts (because of the retention limits), we will differentiate them by the b, which will either be given a numerical value, or defined to be infinite.

In order to deal with the different levels of uncertainty in reinsurance, we can use a set of statistical rules called the double rules for expectations and variance, see Bølviken (2014), page 187. These rules will be used in the next section to show some interesting theoretical results on the choice of optimal reinsurance contract and are as follows. Suppose the distribution of Y depends on a random vector $\mathbf{X}$, the double expectation and the double variance are then

$$\mathrm{E}[Y] = \mathrm{E}\{\xi(\mathbf{X})\}, \quad \text{for } \xi(\mathbf{x}) = \mathrm{E}[Y|\mathbf{x}] \tag{2.1}$$

and

$$\mathrm{var}(Y) = \mathrm{var}\{\xi(\mathbf{X})\} + \mathrm{E}\{\sigma^2(\mathbf{X})\}, \quad \text{for } \sigma(\mathbf{x}) = \mathrm{sd}(Y|\mathbf{x}) \tag{2.2}$$

2.2 Monte Carlo in reinsurance

In actuarial science, claims are often simulated using the Monte Carlo method. Here, I will show how to simulate the a x b contract with this method. The cedent net risk is $X^{ce} = \sum_{i=1}^{N} H^{ce}(Z_i)$, where N is the number of claims, Z the size of the claims. The reinsurer's net risk is the then $X^{re} = \sum_{i=1}^{N} H^{re}(Z_i)$. In order to simulate the cedent's and the reinsurer's liabilities we must first simulate the claims using the Monte Carlo simulation method.

The Monte Carlo method is a method to find a numerical value using random procedures. After selecting the suitable distributions for the events, we draw a large number of simulations in order to select the probability of each events we are interested in. In our case, the claim number will be simulated using the Poisson distribution, the claim size can be simulated with several distributions, such as the Log-normal, the Gamma and the Pareto distributions. Algorithm 1 describes how the R-program I developed deals with the simulation of the claims. The input parameters (m, λ, $J\mu T$, σ and ξ) are easy to change according to the desired simulations. Because of the large number of simulations required and the loops in the code, running the program may take several minutes.

Algorithm 1 The claims

Input: Integers m, $\lambda = J\mu T$, σ, ξ

```
1: m ← 100000                              ▷ Number of simulations for the claim number
2: for i in m do
3:     N[i] ← Poisson(λ)
4: Nsum = sum(N)
5: for j in Nsum do
6:     if Nsum[j] > 0 then                 ▷ Need values > 0 to avoid problems later
7:         Z ← Distribution(ξ, σ)          ▷ Log-Normal, Pareto, Gamma ...
8: return Z
```

The simulation from Algorithm 1 gives us all the potential claims and their values. These values are grouped in their respective Poisson group: assuming we have a Poisson simulation of n claims, the program will group the claim sizes corresponding to the same nth group in an array. For later calculations we need to compress these groups into averages in order to avoid numbers of simulations exceeding the computer's capacity.

Figure 1 shows the density of the claim size for each compound distribution. The parameters are: number of simulations $m = 1000000$, Poisson distribution intensity $\lambda = 10$. For the claim size distributions: Log-normal distribution with $\xi = 2$ and $\sigma = 0.3$, Pareto distribution with $\alpha = 3$ and $\beta = 1$ and finally the Gamma distribution with $\alpha = 0.5$ and $\xi = 1$. The plots have been cut in the top to make it easier for the reader to visualise the difference between them.

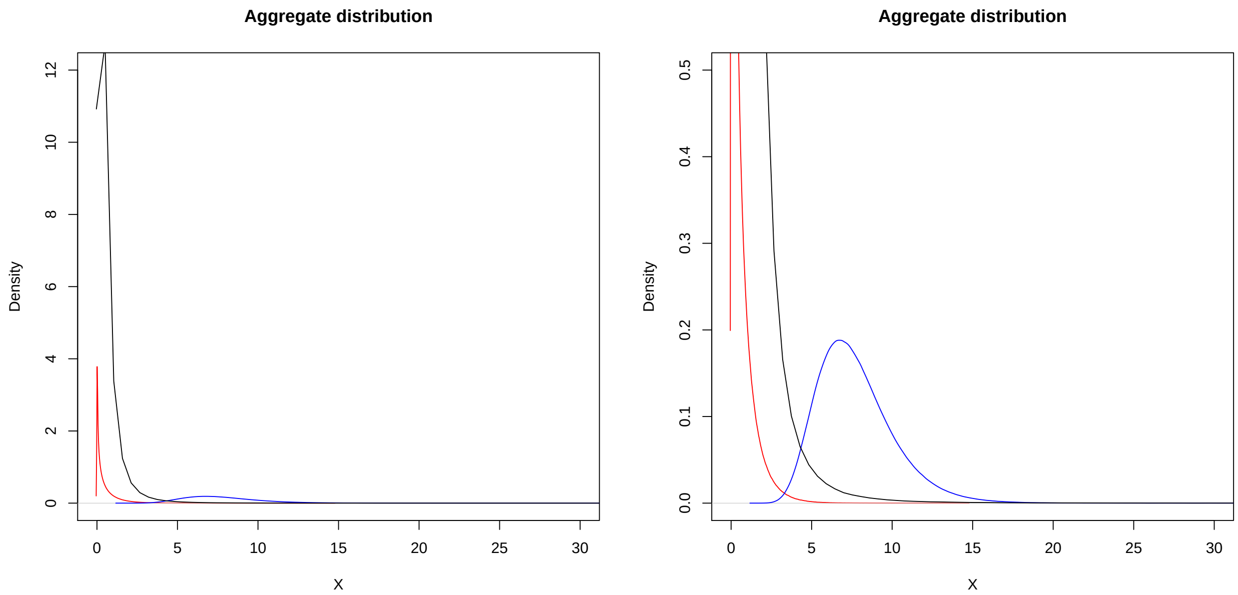


Figure 1: Monte Carlo simulations of the aggregate distribution of the Poisson and Log-normal (blue), Pareto (black) and Gamma (red) distributions. The figure on the right is the same simulations with different axes.

Now that we have all the claims, we can find the reinsurance and cedent liabilities (actually, finding one is enough as they are complementary events). The following algorithms show the procedures we must follow. We start with the algorithm for the cedent, Algorithm 2 shows the procedures needed to find the total liabilities for the cedent **before** singing a reinsurance treaty, it is an average of all the claims simulated in Algorithm 1.

Algorithm 2 Algorithm for the cedent

Input: Integers m

$X \leftarrow 0$
$s2 \leftarrow 0$
$i \leftarrow 1$
while $i \leq m$ **do**
    $s1 = s2 + 1$
    $s2 = s1 + N[i] - 1$
    $X[i] = \text{mean}(Z[s1 : s2])$
    $i \leftarrow i + 1$
return X

Algorithm 3 is the algorithm for the reinsurer, it gives us the procedure to find the reinsurer's liabilities. We have added the reinsurance function H^{re} (in this algorithm the a x b contract) to the procedure. As we can see in the algorithm, the values of a and b are easily changed and we can therefore create a function taking them as parameters.

Algorithm 3 Algorithm for the reinsurer

Input: Integers a, and b

$X \leftarrow 0$

$s2 \leftarrow 0$

$i \leftarrow 1$

while $i \leq m$ **do**

  $s1 = s2 + 1$

  $s2 = s1 + N[i] - 1$

  $X[i] = \text{mean}(\min(\max(Z[s1 : s2] - a, 0), b))$

  $i \leftarrow i + 1$

return X

Once we have both these procedures up and running (for the R programs, consult appendix B.1), we can find Z^{ce} by simply subtracting the results from Algorithm 3 from the results from Algorithm 2. Once we have these values, it is possible to find the gain and expected gain and use them to set up the criteria and optimise them with the retention limits as variables for example.

3 Optimal Reinsurance I: Theory

On average, the cedent loses money on reinsurance, when the cession rate increases, the reinsurance price increases too. However, there are several crucial benefits associated with reinsurance, an insurance company cannot avoid reinsurance. We want to find a contract which allows us to optimize the situation of the cedent according to the optimality criteria. When an insurance company wants to cede part of their risk, there are several types of contracts to choose from. There are also different criteria we can use to optimize on. We are going to focus on the criteria C_σ^{ce} and C_ϵ^{ce} (C_σ and C_ϵ in case there is no reinsurance contract). Their mathematical formulations are:

$$C_\sigma^{ce} = \mathrm{E}[G^{ce}]/\mathrm{sd}(G^{ce}) \tag{3.1}$$

and

$$C_\epsilon^{ce} = \mathrm{E}[G^{ce}]/x_\epsilon^{ce} \tag{3.2}$$

where $\mathrm{E}[G^{ce}]$, $\mathrm{sd}(G^{ce})$ and x_ϵ^{ce} are the expected gain of the cedent, the standard deviation of the cedent and the cedent's capital percentile. It seems that **stability**, keeping the standard deviation low while maximizing the gain and **the value at risk**, keeping the solvency capital low while maximizing the gain are the most relevant criteria when choosing reinsurance contract. Why? Because stability and freeing capital are the main benefits from reinsurance for the cedent.

In this section we are going to examine the stability and value at risk criteria. To do this, we are going to use the literature available on this topic, and also elaborate some own results. Thanks to this process, we are going to be able to rule out some types of contracts for optimality. This will point us toward the optimal contract.

3.1 Maximum stability

Here we are examining the C_σ^{ce} criteria. This is the criteria for stability, we are going to elaborate on this particularly important consequence of reinsurance. In order to maximize C_σ^{ce}, we need to maximize the expected gain and minimize the standard deviation. We are going to investigate the effects of reinsurance on this criteria.

Using the double rules for the expectation and the variance we introduced earlier, we will now be able to state a formula for both the expected gain and the standard deviation of the gain. The portfolio risk is the sum of all the claims, $X = Z_1 + ... + Z_N$ where N, Z_1, Z_2, ... are stochastically independent. Let $\mathrm{E}[Z_i] = \xi$ and $\mathrm{sd}(Z_i) = \sigma$. Elementary rules for random sums imply

$$\mathrm{E}[X|N] = N\xi \quad \text{and} \quad \mathrm{var}(X|N) = N\sigma^2$$

using (2.1) and (2.2),

$$\mathrm{E}[X] = \mathrm{E}[N]\xi \quad \text{and} \quad \mathrm{var}(X) = \mathrm{E}[N]\sigma^2 + \mathrm{var}(N)\xi^2.$$

If N is Poisson distributed so that $\mathrm{E}[N] = \mathrm{var}(N) = J\mu T = \lambda$, then

$$\mathrm{E}[X] = \lambda\xi \quad \text{and} \quad \mathrm{var}(X) = \lambda(\sigma^2 + \xi^2)$$

Defining $X = \sum_{i=1}^{N} Z_i$ as the total of the claims the cedent is accountable for and $X^{re} = \sum_{i=1}^{N} H^{re}(Z_i)$ as the share of the losses the reinsurer covers. The gain of having a reinsurance contract H^{re} is then

$$G^{ce} = (1+\gamma)\mathrm{E}[X] - (1+\gamma^{re})\mathrm{E}[X^{re}] - (X - X^{re}) \tag{3.3}$$

where γ and γ^{re} are respectively the loading of the insurer and the loading of the reinsurer. The loadings express the "price" of the insurance contracts. It is an extra fee the company charges, usually covering their expenses and also giving them some profit. $\mathrm{E}[X] = \pi$ and $\mathrm{E}[X^{re}] = \pi^{re}$ are the pure premiums of the insurer and the reinsurer. $(1+\gamma)\mathrm{E}[X]$ and $(1+\gamma^{re})\mathrm{E}[X^{re}]$ are then, respectively, the true premium charged from the insurance company to the client and from the reinsurer to the cedent. We can express the mean and the standard deviation of the cedent like this,

$$\xi^{ce} = \mathrm{E}[Z - H^{re}(Z)] \quad \text{and} \quad \sigma^{ce} = \mathrm{sd}(Z - H^{re}(Z))$$

we can then write

$$\mathrm{E}[X^{ce}] = \lambda\xi^{ce} \tag{3.4}$$

and

$$\mathrm{var}(X^{ce}) = \lambda\left((\xi^{ce})^2 + (\sigma^{ce})^2\right) \tag{3.5}$$

The expected gain and the variance of the gain are then,

$$\begin{aligned}
\mathrm{E}[G^{ce}] &= (1+\gamma)\mathrm{E}[X] - (1+\gamma^{re})\mathrm{E}[X^{re}] - \mathrm{E}[X] + \mathrm{E}[X^{re}] \\
&= \gamma\mathrm{E}[X] - \gamma^{re}\mathrm{E}[X^{re}] \\
&= \gamma\lambda\xi - \gamma^{re}\lambda\xi^{re} = \lambda(\gamma\xi_Z - \gamma^{re}\xi^{re}) \\
&= \lambda(\gamma\xi - \gamma^{re}H^{re}(Z)) \\
\mathrm{var}(G^{ce}) &= \lambda((\xi^{ce})^2 + (\sigma^{ce})^2)
\end{aligned}$$

There is no right answer to the ratio between $\mathrm{E}[G^{ce}]$ and $\mathrm{sd}(G^{ce})$ that gives optimality for the cedent, this depends on the companies risk profile. However, it is mathematically possible to find the reinsurance contract which maximizes the criteria C_{σ}^{ce}. This is known as an efficient frontier, a term introduced by Markowitz (1952). A combination of assets is efficient if the expected gain is the highest, given its level of risk. For reinsurance we can plot the expected gain versus the standard deviation of the gain we get from changing the reinsurance parameters. The efficient frontier will be the portion of the plot which gives the highest expected gain given its standard deviation.

The first result we will be looking at is a simple mathematical argument using the properties of the non-proportional reinsurance contracts. For the C_σ criteria, we can use the formulas acquired above to show some interesting outcomes. Assume we have a reinsurance contract with function H_a^{re} such that

$$H_a^{re}(Z) = \min(Z - a, 0), \qquad \text{for all } Z > a.$$

This is an a x b contract with an infinite b. Now, consider an arbitrary reinsurance contract $H^{re}(Z)$ which must only satisfy $H^{re}(Z) \leq Z$. This leaves us with the following two possible scenarios,

$$Z > a: \quad |Z - Z + a - a| = |Z - H_a^{re}(Z) - a| = 0$$
$$Z < a: \quad |Z - a| \leq |Z - H^{re}(Z) - a|$$

which then leaves us with

$$|Z - H_a^{re}(Z) - a| \leq |Z - H^{re}(Z) - a|, \qquad \text{for all } H^{re}(Z) \leq Z$$

using the above arguments for the variance of the claims, we can show the following interesting result:

$$\begin{aligned}
\operatorname{var}(Z - H_a^{re}(Z)) &= \operatorname{var}(Z - H_a^{re}(Z) - a) \\
&= \mathrm{E}[Z - H_a^{re}(Z) - a]^2 - (\mathrm{E}[Z - H_a^{re}(Z) - a])^2 \\
&\leq \mathrm{E}[Z - H^{re}(Z) - a]^2 - (\mathrm{E}[Z - H^{re}(Z) - a])^2 \\
&= \operatorname{var}(Z - H^{re}(Z))
\end{aligned}$$

This shows us that the a x b contract with infinite b gives us a smaller or equal variance, for a fixed expected gain, than any other arbitrary reinsurance contract. This is an important result, because it tells us that the a x b contract with infinite b can give better results for the C_σ^{ce} criteria compared to any other reinsurance contract.

Another interesting argument we can use is based upon the convexity of the variance. In the introduction we briefly talked about the utility function, but we rejected it as a potential criteria because of its lack of practical use. Looking at (3.5), we see that the variance is composed of the square of the functions ξ^{ce} and σ^{ce}. Cai and Wei (2012) showed that the a x b contract is the optimal reinsurance contract for individual claims under some preconditions. They assume that the risks are positively dependent through stochastic ordering. They prove the convolution preservation of the convex order for positively dependent through the stochastic ordering random vectors. Their result is that for any convex function u, which they see as a risk measure, the expected value of this risk measure for the a x b contract is less than or equal to any other individualized reinsurance treaty. Transposing this argument to our notation, we can express these results in the following way:

$$\mathrm{E}[u(H_{ab}(Z))] \leq \mathrm{E}[u(H(Z))]$$

where H_{ab} is the a x b contract and H an arbitrary individualized reinsurance contract. The function u can be seen as the variance in our case.

Since the variance in our criteria C_σ^{ce} is a convex function, this argument is usable for the criteria. Therefore it is another clue for our choice of contract. However, these arguments show no numerical examples and are not giving us some concrete examples of contracts. These arguments are also difficult to read for anyone with no mathematics background. This shows that there is a need of simpler and more intuitive arguments, such as numerical arguments.

3.2 Value at Risk

Another important criteria to consider is the capital requirements of an insurance company. The capital requirements of insurance companies are based upon worst case scenarios. We wish to find the upper percentiles of the claims handled. This is the root to several values one can consider as an optimality criteria: the reserve, the value at risk and also the conditional value at risk. The reserve is the upper ϵ percentile x_ϵ^{ce} of X^{ce}. This expresses the liquidity a reinsurance company must keep in its books to meet the requirements given by the financial authorities where the company is based. They are usually expressed in percentiles. The percentile level ϵ of a liability Z is a threshold loss value x such that the probability of the loss being greater than x is ϵ:

$$\mathrm{P}(Z > x) = \epsilon. \tag{3.6}$$

The insurance companies wish to minimize these values. The capital required by the authorities is capital they cannot invest and therefore a loss of potential profit for the company. Another optimality criteria could be to maximize the expected gain while keeping the reserve low. This is the C_ϵ^{ce} we introduced earlier:

$$C_\epsilon^{ce} = \mathrm{E}[G^{ce}]/x_\epsilon^{ce} \tag{3.7}$$

we could also have used another version of this, using the value at risk as the denominator

$$\mathrm{E}[G^{ce}]/\mathrm{V@R}_\epsilon(Z).$$

But these two values are more or less expressing the same. In conclusion, the optimality of reinsurance is decided by balancing the risk and reward factors implied by the reinsurance contract the cedent is under.

The argument of Cai J. and Wei W. (2012) from the previous section is not applicable for the C_ϵ^{ce} criteria. Their results are only suitable for convex functions such as the variance and x_ϵ^{ce} is not convex. As we can see from figure 2, the x_ϵ^{ce} has a structure which makes Cai and Wei's arguments not applicable for C_ϵ^{ce}. We will therefore have to find other sources for arguments.

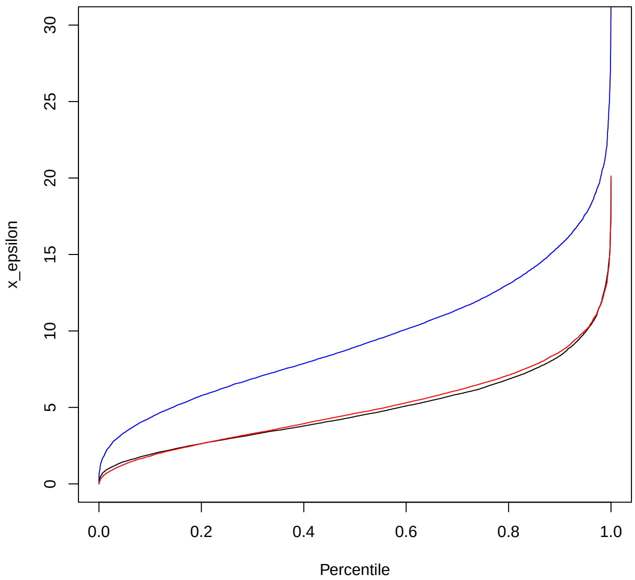


Figure 2: Plot of the reserve for Log-normal (blue), Pareto (black) and Gamma (red) claim sizes with parameters $\xi_{LN} = -0.5$, $\sigma_{LN} = 1$, $\alpha_G = 0.5$, $\xi_G = 1$, $\alpha_P = 3$, $\beta_P = 1$, $\lambda = 10$ and $a = 5$. This is done the Monte Carlo way with 10000 simulations, x-axis go from $\epsilon = 1$ to 0 and the y-axis shows the capital requirements x_ϵ^{ce}.

We can show interesting results for the C_ϵ^{ce} criteria as well. We start with an argument showing that the proportional type of contract is not optimal for the cedent. The expected gain of the cedent under a proportional contract, when c is the percentage agreed upon and γ and γ^{re} are the loadings is

$$\mathrm{E}[G^{ce}] = \gamma\pi - \gamma^{re}\mathrm{E}[\pi^{re}] = \gamma\lambda\mathrm{E}[Z] - \gamma^{re}c + \mathrm{E}[Z] = (\gamma - \gamma^{re}c)\lambda\mathrm{E}[Z]$$

the reserve criteria is then

$$\frac{\mathrm{E}[G^{ce}]}{x_\epsilon^{ce}} = \frac{(\gamma - c\gamma^{re})\lambda\mathrm{E}[Z]}{(1-c)x_\epsilon} = \frac{(1 - c\gamma^{re}/\gamma)C_\epsilon}{1-c}$$

which concludes to the following for the criteria

$$C_\epsilon^{ce} = \frac{(\gamma - c\gamma^{re})\lambda}{1-c}C_\epsilon < C_\epsilon, \quad \text{when } \gamma < \gamma^{re} \tag{3.8}$$

where C_ϵ is the reserve criteria with no reinsurance. In order for the argument to work, we need the loading of the cedent to be lower than the loading of the reinsurer, but this is actually what we observe in the industry. We see in equation (3.8) that when we choose a proportional type of contract, the solvency criteria with reinsurance is less than or equal the criteria without reinsurance. This means that at best, the cedent gets the same criteria with reinsurance than without. In other words, when considering the reserve criteria, the proportional types of reinsurance contract is not optimal. This points us towards the non-proportional types of contracts.

Another interesting result for large portfolios is given in Cheung (2011). It is shown there that the optimal reinsurance contract for large portfolios under value at risk and conditional tail expectation is insurance layers (which can be translated into our excess of loss type of contracts). Under law-invariant convex risk measure (average value at risk), the optimal contract is the stop-loss type of contract. This is another clue pointing towards the non-proportional types of contracts. But, the arguments they use in this paper are as theoretical as the ones in Cai and Wei (2012). They also lack the numerical examples for easier interpretation and practical use.

3.3 Several large portfolios

What happens when we are dealing with large portfolios and reinsurance? Examining this question can help us understand optimal reinsurance and possibly point us towards the reinsurance contracts that are also optimal for individual claims. A lot more research has been made on this topic than for the individual claims, it is therefore interesting to see results in this case. We are going to look at some results for the solvency capital.

Assume we have a portfolio loss X and a number of policies $J \longrightarrow \infty$. When we have a large number of random numbers, the central limit theorem can be applied to formulate the aggregated losses. The central limit theorem states that the mean of a sufficiently large number of independent random variables, each with well defined expected value and well defined variance, will be approximatively normally distributed. Since our portfolio losses are independent from each other we get

$$\mathrm{E}[X] = \lambda\xi J \qquad \text{and} \qquad \mathrm{sd}(X) = \sqrt{\lambda}\sqrt{((\xi)^2 + (\sigma)^2)}\sqrt{J}$$

Now, let us consider the gain of the cedent G^{ce} for large portfolios. I will not include the loadings in this formula for simplification. The gain is now

$$G^{ce} = \lambda\xi^{ce}J + \sqrt{J} \cdot \mathrm{sd}(X^{ce})N(0,1) + o(\frac{1}{\sqrt{J}})$$

where $\sqrt{J}o(\frac{1}{\sqrt{J}}) \to \infty$ as $J \to \infty$. The function $o(\frac{1}{\sqrt{J}}) < \sqrt{J}\mathrm{sd}(X^{ce})N(0,1)$ for large portfolios. Which means that for a large J, we can have some fixed results for the gain of the cedent which can help us find results. This leads to the following approximation for the reserve percentile q_ϵ:

$$q_\epsilon = \lambda\xi^{ce} + \sqrt{\lambda}\sqrt{(\xi^{ce})^2 + (\sigma^{ce})^2}\phi_\epsilon = \sqrt{J}\mathrm{sd}(X^{ce})\phi_\epsilon$$

where ϕ_ϵ is the upper ϵ-percentile of the standard normal distribution, for example $\phi_{0.99} = 2.33$. This is due to the Lindeberg extension of the central limit theorem see Appendix A.4. in Bølviken (2014).

When we wish to maximize the C_ϵ criteria, we want to have the smallest q_ϵ possible for the highest corresponding **fixed** gain. Now, if we focus on the formula for q_ϵ, we see that in order to minimize q_ϵ, we must minimize $\mathrm{sd}(X^{ce})$. This is the same problem as the stability criteria C_σ. We saw that the optimal contract for minimizing the variance is the a x b type of contract with infinite b. Since for large portfolios, the criteria have the same optimizing procedure of minimizing the variance, we can conclude that they have the same optimal reinsurance contract: a x b with infinite b.

After investigating the theory on optimal reinsurance, we have found some interesting results. Generally, it seems that the optimal contract is of the non-proportional type. More specifically an a x b contract with b infinite. The arguments for this have all been theoretical, we have no numerical evidence for this statement. Therefore we need to back our arguments with some numerical examples. We will do this in the next section.

4 Optimal Reinsurance II: Numerics

We have seen in the previous section that finding an optimal reinsurance contract for the cedent, when we focus on individual events, is a very relevant issue for today's industry. The number of papers addressing this topic has increased the last decade and generally, they point towards a non-proportional type of contract and more specifically, the a x b contracts (with some variations). We will therefore focus on this type of contract when we are going to examine the numerical results in the following section section. The Monte Carlo method was introduced earlier but is a little too slow and not robust enough. We have a way around these obstacles: The Panjer Recursion. The idea is to find the optimal retention limits by finding the compound distribution of the claim size and amount, and simulating step by step for small increments.

It should be noted that before selecting the Panjer recursion as a tool for numerical results, different approaches were tried. The thesis started off with smoothing splines as a potential tool for optimization. Then when the Panjer recursion was finally selected, different approaches with R and C running in parallel where tried.

4.1 The Panjer Recursion

The Panjer recursion is an algorithm computing the distribution of a compound random variable

$$X = Z_1 + Z_2 + ... + Z_N$$

where both N and $Z_1, Z_2, ...$ are random variables with specific attributes, typically assumed to be independent. It is also assumed that $Z_1, Z_2, ...$ are identically distributed as a random variable Z. In our context, we have each claim size distributed according to a distribution such as the Pareto, the Gamma and the Log-Normal distribution. The number of claims occuring is Poisson distributed. Our compound distribution is then representing the total sum of all the claims. A recursive definition of the distribution of the total claims, for a specific family of claim number and size distributions was introduced in a paper by Panjer (1981). This recursion can be used for different applications, in our case, we have N insurance claims, each of size $Z_1, Z_2, ...$

We are interested in the compound random variable X, where X and the Z_i fulfill the following preconditions. We assume the Z_i to be independent and identically distributed random variables, independent of N. Furthermore the Z_i have to be distributed on small increments $h > 0$, such that

$$f_j = P[Z = jh]$$

the probability that Z is in the jth increment.

For the Panjer recursion, the probability distribution of N has to be a member of the so-called (a,b,0) class of distributions, which will in this thesis be rebranded as the (u,v,0) class for notational purposes. It consists of all counting random variables which satisfy the following relation:

$$p_n = \left(u + \frac{v}{n}\right) p_{n-1}, \text{ for } n = 1, 2, 3... \tag{4.1}$$

for some u and v which fulfill $u + v \geq 0$. The four members of this family of distributions are

1. The Poisson distribution: $p_n = \frac{\lambda}{n!} e^{-\lambda}$

   $u = 0, \ v = \lambda$

2. The Binomial distribution: $p_n = \binom{N}{n} p^n (1-p)^{N-n}$

   $u = -p/(1-p), \ v = (N+1)p/(1-p)$

3. The Negative Binomial distribution: $p_n = \binom{\alpha+n-1}{n} p^n (1-p)^{\alpha}$

   $u = p, \ v = 0$

4. Geometric distribution (Negative binomial with $\alpha = 1$)

The Panjer recursion makes use of this iterative relationship to specify a recursive way of constructing the probability distribution of X. In this project, we will only look at the case of discrete severities. Before we start with the recursion, we need to calculate f_j. After choosing a h, which can be seen, in our context, as a rounding off to the nearest multiple of monetary unit, we discretize the continuous distribution using the central difference approximation:

$f_0 = F(h/2)$
$f_j = F(jh + h/2) - F(jh - h/2)$ for $j = 1, 2, ...$

see the next section for more calculation details.

Now that we have discretized the claim size part, we can go further and calculate the compound distribution by using the procedure given by Algorithm 4.

Algorithm 4 The Panjer recursion

Input: Starting value g_0 and integers a, b, h

```
1: if a = 0 then
2:     g_0 = p_0 · exp(f_0 b)
3: if a ≠ 0 then
4:     g_0 = p_0 / (1 − f_0 a)^(1+b/a)
5: for g_j = P[X = hj] do
6:     g_j = 1/(1 − f_0 a) · Σ_{k=1}^{j} (a + b·k/j) f_k · g_{j−k}
7:     G_j = G_{j−1} + g_j
8: return g_j and G_j
```

$$g_0 = \frac{p_0}{(1-f_0 a)^{1+b/a}}$$

$$g_j = \frac{1}{1 - f_0 a} \sum_{k=1}^{j} \left(a + \frac{b \cdot k}{j}\right) f_k \cdot g_{j-k}$$

Panjer proves that the recursion holds using arguments such as the recursive definition of convolutions and symmetry of the elements in question. The proof for a continuous claim size distribution is in Panjer H. (1981). However it should be noted that Adelson in 1966 came with an article on compound Poisson distributions already addressing this topic.

Before we start with the proof of the recursion, it is interesting to show why the Poisson distribution is said to be a member of the (u,v,0) class of distributions. Assuming that $p_0 > 0$, we can observe that $p_1 = (u+v)p_0$ and since $(u+v) \geq 0$, the value of p_1 is positive. If $(u+v) = 0$, then $p_1 = 0$ and all the $p_n = 0$ for $n \geq 1$. This is then to be ruled out of our potential values for $u+v$. If $u = 0$, like suggested for the Poisson distribution, then $p_n = \frac{v}{n} p_{n-1}$ and

$$p_n = \frac{v^n}{n!} p_0, \text{ for all } n \geq 0$$

From the definition of the (u,v,0) class, we know that $\sum_{n=0}^{\infty} p_n = 1$, which gives us the following,

$$\sum_{n=0}^{\infty} p_n = p_0 \sum_{n=0}^{\infty} \frac{v^n}{n!} = 1$$

We, recognize $\sum_{n=0}^{\infty} \frac{v^n}{n!}$ to be the Taylor series of e^v,

$$1 = p_0 \sum_{n=0}^{\infty} \frac{v^n}{n!} = p_0 e^b$$

We then get that, if $u = 0$, $\{p_n\}_{n=0}^{\infty}$ is the Poisson distribution with parameter v.

For the recursion of the compound distribution, Panjer proves it for the continuous claim size distributions, however this proof is similar to the proof for the discrete distributions and is as follows, we start with the continuous compound distribution function

$$G(x) = \begin{cases} \sum_{n=1}^{\infty} p_n F^{*n}(\lambda) & \text{if } x > 0 \\ p_0 & \text{if } x = 0 \end{cases}$$

for arbitrary claim amount distribution $F(\lambda)$, $x > 0$. The density of total claims is

$$g(x) = \begin{cases} \sum_{n=1}^{\infty} p_n f^{*n}(x) & \text{if } x > 0 \\ p_0 & \text{if } x = 0 \end{cases} \tag{4.2}$$

Panjer uses these two relations in his proof:

$$\int_0^x f(y) f^{*n}(x-y) dy = f^{*(n+1)}(x), \text{ for } n = 1, 2, 3, ... \tag{4.3}$$

$$\int_0^x y f(y) f^{*n}(x-y) / f^{*(n+1)}(x) dy = x/(n+1), \text{ for } n = 1, 2, 3, ... \tag{4.4}$$

Relation (4.3) is the recursive definition of a convolution. The left side of relation (4.4) is the conditional mean of any element of a sum consisting of $n+1$ independent and identically distributed elements, given that the sum is exactly x. The mean is $x/(n+1)$ as a result of the symmetry in the elements of the sum. Now, the theorem for the recursion, given in Panjer (1981), is as follows:

Theorem 4.1. *For p_n and $g(x)$ defined by the previous comments, and $f(x)$ any distribution of the continuous type for $x > 0$, the following recursion holds*

$$g(x) = p_1 f(x) + \int_0^x (u + vy/x) f(y) g(x-y) dy, \text{ for } x > 0 \tag{4.5}$$

The proof is as follows, we start by substituting (4.2) in the right side of (4.5) which gives us:

$$p_1 f(x) + \int_0^x (u + \frac{vy}{x}) f(y) g(x-y) dy = p_1 f(x) + \int_0^x (u + \frac{vy}{x}) f(y) \sum_{n=1}^{\infty} p_n f^{*n}(x-y) dy$$

We develop the right hand side (**RHS**),

$$\begin{aligned}
\mathbf{RHS} &= p_1 f(x) + \sum_{n=1}^{\infty} p_n \int_0^x (u + \frac{vy}{x}) f(y) f^{*n}(x-y) dy \\
&= p_1 f(x) + \sum_{n=1}^{\infty} p_n \left(u + \frac{v}{(n+1)} \right) f^{*n}(x-y) dy && \text{(from (4.3) and (4.4))} \\
&= p_1 f(x) + \sum_{n=1}^{\infty} p_{n+1} f^{*(n+1)}(x) && \text{(from (4.1))} \\
&= p_1 f(x) + \sum_{n=2}^{\infty} p_n f^{*n}(x) \\
&= \sum_{n=1}^{\infty} p_n f^{*n}(x) && \text{(since } f^{*1}(x) = f(x)) \\
&= g(x)
\end{aligned}$$

This is the result we wanted to prove the relation.

4.2 Implementation

To understand the Panjer recursion, it is important to see how it works by showing some simple numerical examples. We look at the case where the distribution of the claim size is Log-normal with parameters $\mu = 0$ and $\sigma = 2$. For the "money" (precision) parameter, we chose $h = 1\$$.

We start with the discretization of the Log-normal distribution:

$f_0 = F(h/2) = F(0.5) = 0.364$
$f_1 = F(h + h/2) - F(h - h/2) = F(1.5) - F(0.5) = 0.216$
$f_2 = F(2h + h/2) - F(2h - h/2) = F(2.5) - F(1.5) = 0.096$
...

In order to find the discrete version of the log-normal distribution, I have developed a R-program called "discretize", which finds the results. The following table is a calculation of the first 5000 f_j.

j	f_j
0	3.645×10^{-1}
1	2.159×10^{-1}
2	9.625×10^{-2}
3	5.789×10^{-2}
.	.
.	.
.	.
4999	4.606×10^{-9}
5000	4.603×10^{-9}

Table 2: Results for f_j using the R program discretize.

Comparing the plots of the simulated log-normal distribution against the discretized version shows that they are close to each other. This means that the discretized version is giving accurate enough precision, however it should be noted that the smaller the lattice, the greater the precision. We will see later that decreasing the increment size increases the computing time a lot.

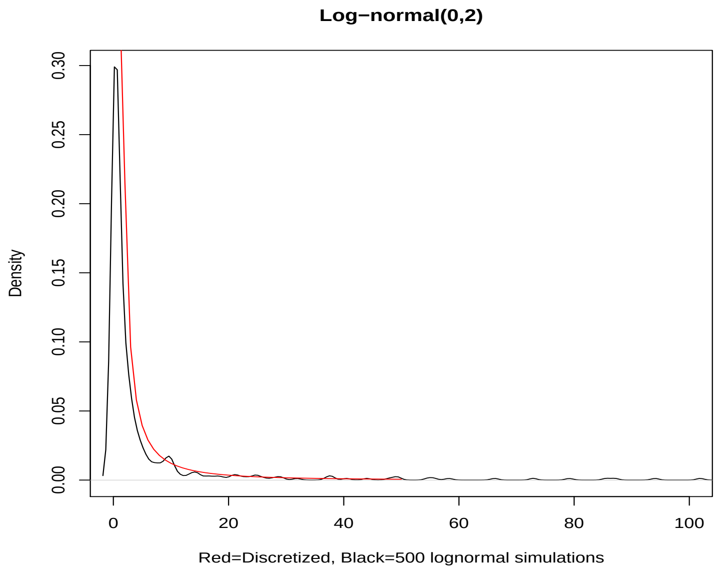


Figure 3: Simulated Log-normal distribution (black) versus discretized Log-normal distribution (red) using the R software

For the claim numbers, we assume that they are Poisson distributed with intensity $\lambda = 10$. We know that for the Poisson distribution, which is a member of the (u,v,0) distribution class, $u = 0$ and $v = \lambda$. Inserting this into the formula for p_n we now have all the information we need to calculate g_j:

$$g_0 = p_0 \cdot \exp(f_0 b) = \exp(-10) \cdot \exp(0.3644584 \cdot 10) = 1.7373 \times 10^{-3}$$

$$g_1 = \left(\frac{b \cdot 1}{1}\right) f_1 \cdot g_0 = 3.75 \times 10^{-3}$$

$$g_2 = \left(\frac{b \cdot 1}{2}\right) f_1 \cdot g_1 + \left(\frac{b \cdot 2}{2}\right) f_2 \cdot g_0 = 5.71 \times 10^{-3}$$

$$\ldots$$

When the frequency is large $\lambda \geq 700$, see Cruz (2015) chapter 13, we get numbers outside the range of the computer (the number is too small), we can overcome this by scaling the Poisson distribution and calculate for some large m:

$$G^{(m)*}(z;\lambda/m) = G(z;\lambda)$$

When computing the recursion, a large number of simulation may be needed in order to find the aggregated loss distribution. In order to implement the Panjer recursion into our optimal reinsurance problem, I had to build a program computing the recursion. The first programming language I used to build this program was the statistical computing language R. However, the R-program ran slow when the increment width was small. This forced me to try another programming language: Fortran.

The first computation required is the discretization of the claim size distribution. This is a straightforward calculation, following the Algorithm 5 and the formula for f_j. In R, there are built-in functions calculating the distribution function of the distributions we need for the recursion. The discretization program I build in R works as in the following way:

Algorithm 5 The discrete claim size

Input: Integers h, $from$ and to

$s \leftarrow \text{sequence}(from, to, \text{by} = h)$ ▷ Sequence with increment h

$f_0 = \text{Distribution}(\frac{h}{2})$

for j in s **do**

  $f_j \leftarrow \text{Distribution}(j \cdot h + \frac{h}{2}) - \text{Distribution}(j \cdot h - \frac{h}{2})$

return f_j

When the discretization is done, the recursion for the compound distribution can start. This is the part of the program which is demanding a lot of computer resources. For each extra decimal of precision added, the number of calculations is squared. For example, changing h from 0.01 to 0.001 requires about a 100 more calculations and naturally takes a lot more time to compute. Again, using Algorithm 4, I build a program computing the g_j's. The idea is to use the f_j we get from the discretize program as input and use it in the recursion. We also need the claim number distribution, which in our case is the Poisson distribution. The R program developed for this is called PanjerPoisson, and works like this:

Algorithm 6 The discrete claim size

Input: Integer λ, and the f_j

$s \leftarrow \text{sequence}(from, to, \text{by} = h)$

$g_0 = \exp(-\lambda(1 - f_0))$

$R \leftarrow \text{length}(j)$

for j in s **do**

  $g_j \leftarrow \frac{\lambda}{j} \sum_{k=1}^{j} k \cdot f_j \cdot g_{j-k}$

return g_j

Once the procedures are up and running, the problem of the optimal reinsurance can be implemented in the program. The R program I build is suited for increments larger than 1, however when the increments are less than 1, the computations take too long and crash most of the time. Therefore, a Fortran program was developed in order to have greater precision without taking too long. We want to find the optimal lattices for the a and the b of our reinsurance contract, in order to find this, we must implement these values and the criteria in our program, the net claims the cedent must cover after reinsurance is expressed as

$$X^{ce} = \sum_{i=1}^{N} \{Z_i - H^{re}(Z_i)\}$$

where H is the function for the reinsurance contract.

We are considering the a x b contract, the factors from the contract that can vary the outcome of the criteria are the retention limits a and b. Now for our Panjer program we introduce the retention limits induced from the increments in our recursion

$$a = j_a h \qquad \text{and} \qquad b = j_b h$$

Just like the Monte Carlo method, we need to decide which pair of a and b that gives optimality according to the criteria we established. In the Panjer recursion we split the compound distribution into small intervals with increment size h, letting

$$g_j^{ce} = \mathrm{P}(Z^{ce} = jh).$$

This configuration of the recursion gives us the following values for the claims:

$$Z_j^{ce} = \begin{cases} jh & \text{if } j < j_a \\ j_a h & \text{if } j_a \leq j \leq j_a + j_b \\ (j - j_b)h & \text{if } j > j_a + j_b \end{cases}$$

This gives us the following function and values for the recursion function:

$$g_j^{ce} = \begin{cases} g_j & \text{if } j < j_a \\ g_{j_a} + ... + g_{j_a + j_b} & \text{if } j_a \leq j \leq j_a + j_b \\ g_{j_a} + ... + g_{j_a + j_b} & \text{if } j > j_a + j_b \end{cases}$$

The Fortran program, makes us of the recursive bisection method to find the optimal criteria. This method is simple and robust, it works the following way, we start with an interval $[i_a, i_b]$, usually the endpoints, from our function g_j^{ce}. The first procedure is to find the midpoint of the interval, if the function value from this point is closer to one of the initial points, you replace the point with the midpoint and continue until the interval between midpoint and interval points is below a value we have fixed, see Algorithm 7.

Algorithm 7 The recursive bisection method

Input: Integer i_a and i_b, $i_a < i_b$, and the function g_j

```
N ← 1
while N ≤ NMAX do                                  ▷ To avoid infinite loops
    i_c ← (i_a + i_b)/2
    if (i_a + i_b)/2 <MIN then Output(i_c)         ▷ MIN is the precision value
    N ←N+1
    if g(i_c) closer to g(i_a) then i_b ← i_c
    else i_b ← i_c
return i_c
```

The way to use the Fortran programs on a Linux command window and some time tests are explained in the appendix. After doing some tests with the program, we realize that the more the precision, the more time the program takes to run, however, after varying the h from 1 to 0.001, we notice that there is no major increase in the precision below an increment size of 0.01. However the time increase from 0.01 to 0.001 is very large, for the Gamma distribution the time it takes to run the program is multiplied by 152 and for the Pareto distribution it is multiplied by 125 (see Appendix B).

h	**Gamma claims**				**Pareto claims**				**Log-normal claims**			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
1	8.4	7.9	4.0	32.0	9.0	7.1	2.0	105	2.0	10.6	1.0	1.0
0.5	8.2	7.8	3.5	31.0	8.5	6.8	1.5	50	6.9	10.9	1.0	1.0
0.1	8.1	7.7	3.8	31.7	8.2	6.6	1.7	67	10.6	10.8	1.2	1.2
0.01	8.1	7.7	3.8	31.4	8.1	6.5	1.7	65.2	10.8	10.8	1.4	1.4
0.001	8.1	7.7	3.8	31.4	8.1	6.5	1.7	65.5	10.8	10.8	1.4	1.4

Table 3: Criteria with and without reinsurance (in percent) and retention limits with varying increment size h. The parameters are $\lambda = 10$, $\alpha_G = 0.5$, $\xi_G = 1$,$\alpha_P = 3$, $\beta_P = 1$, $\xi_{LN} = 1$, $\sigma_{LN} = 0.3$, $\gamma = 0.2$, $\gamma^{re} = 0.3$ and $\epsilon = 0.01$.

h	**Gamma claims**				**Pareto claims**				**Log-normal claims**			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
1	9.7	9.9	3.0	3.0	9.0	8.4	4.0	57	10.0	10.1	3.0	3.0
0.5	9.9	9.9	4.0	20.5	8.9	8.3	3.5	57	10.4	10.4	3.5	4.0
0.1	9.9	9.8	4.2	19.2	8.8	8.2	3.6	57.1	10.3	10.3	3.1	3.3
0.01	9.9	9.8	4.5	20.3	8.8	8.2	3.6	57.2	10.3	10.3	3.0	3.2
0.001	9.9	9.8	4.5	20.3	8.8	8.2	3.6	57.2	10.3	10.3	3.0	3.2

Table 4: Criteria with and without reinsurance (in percent) and retention limits with varying increment size h. The parameters are $\lambda = 10$, $\alpha_G = 2$, $\xi_G = 2$, $\alpha_P = 7$, $\beta_P = 7$, $\xi_{LN} = 1$, $\sigma_{LN} = 0.5$, $\gamma = 0.2$, $\gamma^{re} = 0.3$ and $\epsilon = 0.01$.

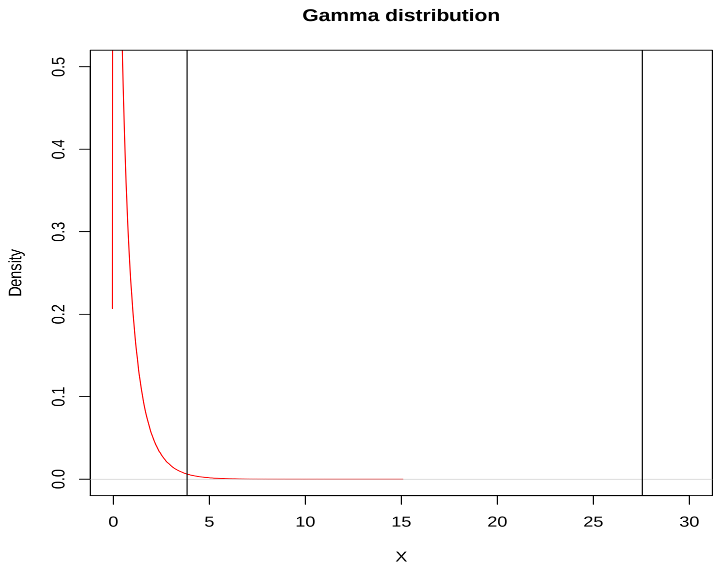


Figure 4: Simulated compound Gamma and Poisson distributions (red) with distribution parameters $\xi = 1$, $\alpha = 0.5$ and $\lambda = 10$. The vertical bars are, from left to right the a and b retention limits we get from the increment $h = 0.001$.

The tables are a synthesis of the results we get from running the Fortran programs we have (see Appendix B). It shows the retention limits and the criteria for different increment size h. The tables provide C_ϵ^{ce} (the criteria with reinsurance) and C_ϵ(the criteria without reinsurance), with the corresponding a and b, it is interesting to see how reinsurance affect the criteria. In our example, we see that having reinsurance increases the criteria for both the Gamma and the Pareto distributions. This means that the company will have a larger expected gain for less required capital when they have reinsurance. This results could be expected, reinsurance is a stabilizer of risk and is also increasing the companies ability to underwrite more profitable risks. All the results in these tables are with the same distribution parameters.

For the Log-normal distribution, the results are not the same, the Fortran program shows that the optimal situation is no reinsurance. It also shows that it is important to use a small enough increment size for precision. The results for $h = 1, 0.5$, and 0.1 are contradicting, the retention limits tell us that the interval between a and b is 0, which is the same as saying no reinsurance. However, $C_\epsilon^{ce} < C_\epsilon$, which means that reinsurance gives a higher criteria and is then optimal for the cedent. But, we can see that when the $h < 0.1$, the criteria are identical, the results for the retention limits and the criteria are no longer contradicting.

Changing h changes the criteria, when h varies from 1 to 0.001, C_ϵ^{ce} goes from 0.0895 to 0.0811 and C_ϵ from 0.0712 to 0.0654 for the Pareto distribution which is not more than a 10% change. These variation are as small for the Gamma and the Log-normal distributions.

The retention limits seem to vary a lot more, especially for the Pareto distribution: changing h from 1 to 0.001 changes a from 2.00 to 1.67 and b from 103.00 to 63.85, which is a 38% variation. Even with the same distributions parameters, loadings and percentiles, a variation of h can change the value of the retention limits by almost half of its original value. For the Gamma distribution, the change is less dramatic. An explanation to this is the shape of the distributions, the Pareto distribution has longer tails, which might go unnoticed if the precision is not good enough. From figure (4), we see that the retention limits are more on the extreme values, could this support the idea that the optimal contract is optimal with an infinite b?

Now that we have the program up and running, we can use the program to investigate the results with varying parameters.

4.3 Results

Our Fortran programs simulates an a x b contract. It calculates the a and b that optimize C_ϵ^{ce} and C_ϵ, the ratios between the expected gain and the net reserve with and without reinsurance. As we saw in the previous sections, several arguments pointed towards the a x b contract as optimal for the individualised claims. In this section our main goal is to give results for different parameters and vary these in order to see if the numerical results validate the theoretical results. Thanks to the programs, we are now able to vary the distribution parameters which affect both the claim size and the claim numbers. We can also vary the percentiles for the capital requirements, the loadings for both the cedent and the reinsurer and finally the increment size h. This is a very important part of the thesis, here we will be able to see if the theory fits with the results from the Panjer recursion.

An interpretation of the tables that will follow is that the larger the difference between the criteria C_ϵ and C_ϵ^{ce} the more the reinsurance contract is needed. Usually we will have $C_\epsilon \leq C_\epsilon^{ce}$. As for the retention limits, the larger the interval between a and b, the more reinsurance contract is needed too. We will see that these two values are related.

Variation of distribution parameters

First we vary the λ of the Poisson distribution for the claim number, this parameter describes the intensity of the claims. How often they occur in a determined time interval. It is therefore interesting to see how this value varies the retention limits or the criteria. I will present the results of this in tables and figures representing the criteria and the retention limits. There will also be two versions of the tables, each with different claim size distributions. This is done to have more evidence in case of a trend. I encourage the reader to try the program with different parameters to see how they vary.

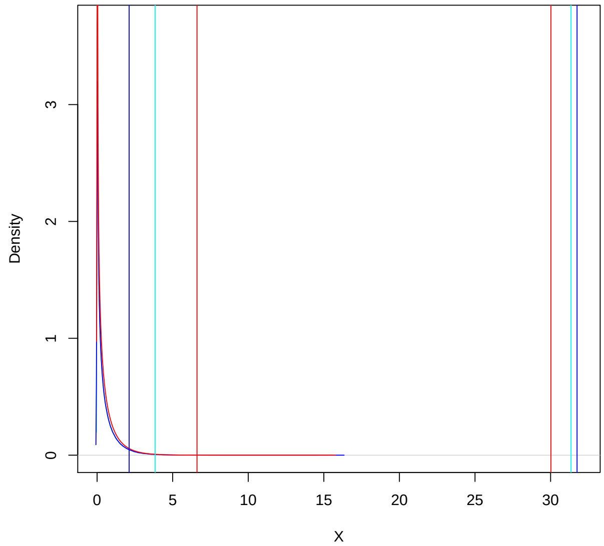
λ	Gamma claims				Pareto claims				Log-normal claims			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
1	3.0	2.5	2.1	31.8	2.4	2.4	42.9	42.9	3.7	3.6	6.7	8.5
2	4.2	3.7	3.1	32.3	4.3	3.3	1.3	16.0	5.1	5.1	7.6	9.7
5	6.3	5.8	3.4	31.7	6.4	4.9	1.5	35.3	7.5	7.4	8.7	10.8
10	8.1	7.6	3.8	31.4	8.1	6.5	1.7	65.2	9.4	9.4	9.7	11.7
20	10.0	9.6	4.4	31.0	9.9	8.4	2.0	116.4	11.4	11.4	10.7	12.4
50	12.3	12.2	5.5	30.5	12.2	10.9	2.5	198.7	13.7	13.7	12.3	13.7
100	14.0	13.9	6.6	30.0	13.8	12.8	3.1	195.3	15.2	15.2	13.9	14.9

Table 5: Criteria and retention limits when varying the intensity λ, the other parameters are $h = 0.01$,$\alpha_G = 0.5$,$\xi_G = 1$,$\alpha_P = 3$,$\beta_P = 1$,$\xi_{LN} = 2$,$\sigma_{LN} = 0.7$,$\gamma = 0.2$,$\gamma^{re} = 0.3$ and $\epsilon = 0.01$. The criteria are in percent.

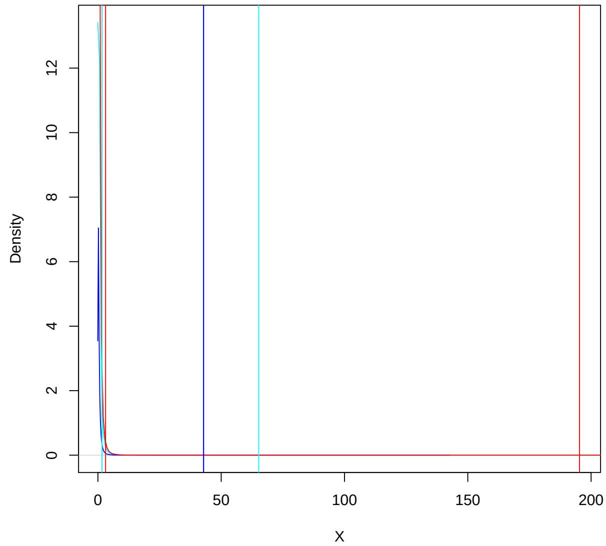
λ	Gamma claims				Pareto claims				Log-normal claims			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
1	4.1	3.9	3.7	21.3	3.5	2.9	2.5	48.3	4.3	4.3	2.3	2.6
2	5.6	5.4	3.8	21.0	4.8	4.1	2.8	58.6	5.9	5.9	2.6	2.9
5	8.0	7.9	4.3	31.7	6.9	6.3	3.2	35.3	8.4	8.3	2.7	2.9
10	9.9	9.8	4.5	20.8	8.8	8.2	3.6	57.9	10.3	10.3	3.1	3.3
20	11.7	11.7	3.9	3.9	10.6	10.2	4.2	56.2	12.2	12.2	3.6	3.8
50	14.0	14.0	4.5	4.5	13.0	12.7	5.2	54.6	14.3	14.3	3.8	3.9
100	15.4	15.4	4.5	4.5	14.5	14.4	6.4	53.2	15.7	15.7	4.2	4.3

Table 6: Criteria and retention limits when varying the intensity λ, the other parameters are $h = 0.01$,$\alpha_G = 2$,$\xi_G = 2$,$\alpha_P = 7$,$\beta_P = 7$,$\xi_{LN} = 1$,$\sigma_{LN} = 0.5$,$\gamma = 0.2$,$\gamma^{re} = 0.3$ and $\epsilon = 0.01$. The criteria are in percent.

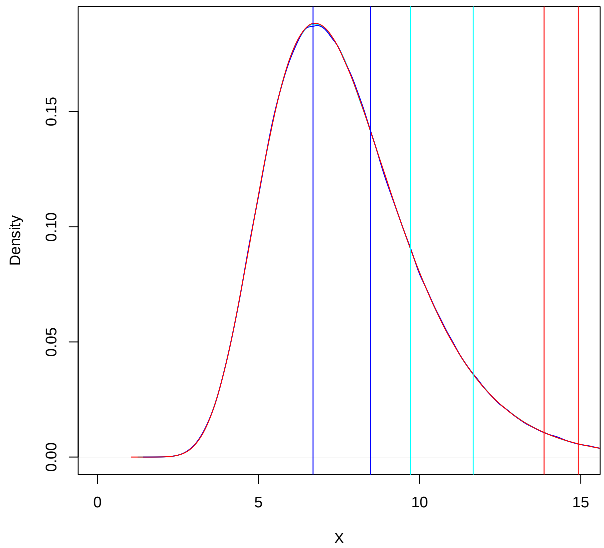
In figure 5, I have plotted the Monte-Carlo simulations of the density of the three distributions for $\lambda = 1, 10$ and 100. I have also included the retention limits we get from the Fortran program. The retention limits and the claims density with the same λ have the same colors. The densities seem to be very close to each other, it is not easy to see the difference between them.



(a) Gamma distribution



(b) Pareto distribution



(c) Log-normal distribution

Figure 5: Aggregated distributions with retention limits. $\lambda = 1$ (dark blue) $\lambda = 10$ (light blue) and $\lambda = 100$ (red); $h = 0.01$,$\alpha_G = 0.5$,$\xi_G = 1$,$\alpha_P = 3$,$\beta_P = 1$,$\xi_{LN} = 2$,$\sigma_{LN} = 0.7$,$\gamma = 0.2$,$\gamma^{re} = 0.3$ and $\epsilon = 0.01$

The trend here is that increasing λ increases the need for reinsurance for the Pareto claim size and decreases the need for reinsurance for the two other distributions. The difference between a and b are increasing when λ is increasing. However, for the Gamma and the Log-normal distributions, the results are opposite (the Log-normal claims do not vary a lot from $\lambda = 1$ to $\lambda = 100$). These are expected results, it seems that they are linked to the compound claims for the cedent. The three distributions have different characteristics when λ varies.

Note that for small values of λ, the Pareto distribution finds that optimality is without reinsurance, we see that $a = b$ when $\lambda = 1$. The densities of the distributions are not much affected by the change of λ, except for the Pareto distribution which sees it tail increase a lot when λ increases. Another interesting we get from the tables is that it seems that an increase in the interval between a and b seems to increase the difference between C_ϵ and C_ϵ^{ce}.

Now we vary the parameters for the distribution of the claim size, will this affect the criteria and the retention limits the same way as the claim number?

α_G	ξ_G	**Gamma claims**				α_P	β_P	**Pareto claims**				ξ_{LN}	σ_{LN}	**Log-normal claims**			
		C_ϵ^{ce}	C_ϵ	a	b			C_ϵ^{ce}	C_ϵ	a	b			C_ϵ^{ce}	C_ϵ	a	b
1	2	9.1	8.9	5.9	35	10	4	8.9	8.4	1.4	14.4	1	0.5	10.3	10.3	3.1	3.3
1	4	9.1	8.9	12	70	10	3	8.9	8.4	1.0	10.8	1	1	7.8	7.6	8.8	13.1
1	8	9.1	8.9	23	140	10	1	8.9	8.4	0.3	3.6	2	0.5	10.3	10.3	6.1	6.5
2	2	9.9	9.8	4.5	20	3	5	8.1	6.5	8.4	328	2	1	7.8	7.6	17.5	26.0
3	2	10.2	10.2	3.2	3.2	3	6	8.1	6.5	10	393	2	1.5	4.5	4.5	1315	1315
5	1	10.5	10.5	1.2	1.2	3	7	8.1	6.5	12	459	10	1.5	4.5	4.5	6577	6577
9	0.5	10.7	10.8	0.4	0.4	7	7	8.8	8.2	3.6	54	0	0	0	0	4.2	4.3

Table 7: Criteria and retention limits with varying distribution parameters. The other parameters are $h = 0.01$,$\lambda = 10$,$\gamma = 0.2$,$\gamma^{re} = 0.3$ and $\epsilon = 0.01$. The criteria are in percent.

From the table we see that the variation of the distribution parameters affect the value of the criteria and the retention limits a lot, for the Log-normal claims we get proportion differences between the retention limits for $\xi_{LN} = 1$ and $\sigma_{LN} = 0.5$ and $\xi_{LN} = 10$ and $\sigma_{LN} = 1.5$ to be approximatively 2100. In appendix A, I have plotted the distributions with different parameters from the table so we can see how the shape of the density affects the criteria and retention limits. It seems that the longer the tail, the less reinsurance is needed for the Gamma and the Log-normal distributions, the opposite for the Pareto distribution.

We can conclude that the distribution parameters for both the claim number and claim size affect the need for reinsurance. They should be chosen carefully.

Variation of the loadings

The loadings are very important values when considering reinsurance, the reinsurance companies usually don't reveal their loading factors because of the competition, but it is safe to assume that $\gamma < \gamma^{re}$ and that the difference between these two is normally not very large. The variation of these values from year to year can be quite substantial. A year where lot of catastrophic events occur can increase the price of next year's premiums. Therefore it is of great importance to see how changing these values change the cedent's need for reinsurance as they appear to be volatile.

γ^{re}	Gamma claims				Pareto claims				Log-normal claims			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.1	∞	7.6	0	17.0	∞	6.5	0	107	9.7	9.4	3.2	4.5
0.2	∞	7.6	0	17.0	∞	6.5	0	107	9.5	9.4	7.4	9.3
0.4	7.8	7.6	5.7	31.1	7.6	6.5	2.7	52.3	9.4	9.4	11.4	13.3
0.6	7.7	7.6	6.8	29.3	7.2	6.5	4.2	33.7	9.4	9.4	13.7	15.4
0.8	7.6	7.6	6.8	6.8	7.0	6.5	5.4	23.8	9.4	9.4	15.3	16.7

Table 8: Criteria and retention limits with varying γ^{re}. The other parameters are $h = 0.01$, $\lambda = 10$, $\alpha_G = 0.5$, $\xi_G = 1$, $\alpha_P = 3$, $\beta_P = 1$, $\xi_{LN} = 2$, $\sigma_{LN} = 0.7$, $\gamma = 0.2$, and $\epsilon = 0.01$. The criteria are in percent.

γ^{re}	Gamma claims				Pareto claims				Log-normal claims			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.1	∞	9.8	0	11.3	∞	8.2	0	31.5	12.1	10.3	0.1	0.6
0.2	∞	9.8	0	11.3	∞	8.2	0	31.5	10.3	10.3	2.1	2.3
0.4	9.8	9.8	4.5	4.5	8.5	8.2	5.3	56.4	10.3	10.3	3.7	3.9
0.6	9.8	9.8	4.5	4.5	8.3	8.2	7.8	55.6	10.3	10.3	4.2	4.3
0.8	9.8	9.8	4.5	4.5	8.3	8.2	9.5	54.9	10.3	10.3	4.2	4.2

Table 9: Criteria and retention limits with varying γ^{re}. The other parameters are $h = 0.01$, $\lambda = 10$, $\alpha_G = 2$, $\xi_G = 2$, $\alpha_P = 7$, $\beta_P = 7$, $\xi_{LN} = 1$, $\sigma_{LN} = 0.5$, $\gamma = 0.2$, and $\epsilon = 0.01$. The criteria are in percent.

From tables 8 and 9 we get the following results, we see that for the Gamma and Pareto distributions, a cheap reinsurance contract (loadings equal to 0.1 and 0.2 which is in this case $\leq \gamma$) gives the cedent unlimited criteria, the ratio between the expected gain and the reserve is infinite, the interval between the retention limits is also at the largest (see Appendix A for figures). This is an arbitrage opportunity. For these two values of γ^{re}, the cedent can just reinsure all the claims and make an unlimited gain for very small capital requirements and without any risk of losing money. For the Log-normal distribution the results were a little different. I had to go to as low as $\gamma^{re} = 0.05$ to get the same results than for the Gamma and Pareto distributions. The cases where $\gamma > \gamma^{re}$ are not realistic and we cannot expect to see these scenarios in the insurance world, see Wahlin (2012).

Now, when the the reinsurance price increases in comparison to the cedent's loading, optimality is achieved with less reinsurance. The difference between C_ϵ^{ce} and C_ϵ is also shrinking and even becoming 0 for large enough γ^{re}. This means that when the loading of the reinsurance company is too large compared to the one for the insurance company, the most profitable scenario for the cedent is to reinsure as little as possible. This is a very intuitive result. In tables 8 and 9, I only varied γ^{re}, in order to see how the loading factor affect the cedent's optimal reinsurance, we need some more results for other configurations of γ and γ^{re}. Will we see the same trend here?

γ	γ^{re}	**Gamma claims**				**Pareto claims**				**Log-normal claims**			
		C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.5	0.5	∞	19.1	0	17	∞	16.4	0	107	23.7	23.5	7.4	9.3
0.5	1.0	19.6	19.1	5.7	31	19.1	16.4	2.7	52.3	23.6	23.5	11.4	13.3
1.0	1.0	∞	38.2	5.7	17	∞	32.7	0	107	47.5	47.0	7.4	9.3
1.0	1.1	44.5	38.2	1.4	31	46.3	32.7	0.6	58.7	47.4	47.0	7.9	9.8
1.0	1.3	41.5	38.2	2.8	32	42.4	32.7	1.2	68.5	47.3	47.0	8.8	10.7

Table 10: Criteria and retention limits with varying both γ and γ^{re}. The other parameters are $h = 0.01$,$\lambda = 10$,$\alpha_G = 0.5$,$\xi_G = 1$,$\alpha_P = 3$,$\beta_P = 1$,$\xi_{LN} = 2$,$\sigma_{LN} = 0.7$ and $\epsilon = 0.01$. The criteria are in percent.

γ	γ^{re}	**Gamma claims**				**Pareto claims**				**Log-normal claims**			
		C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.5	0.5	∞	24.5	0	11	∞	20.4	0	31.5	25.8	25.7	2.1	2.3
0.5	1.0	24.5	24.5	4.5	4.5	21.3	20.4	5.3	56.4	25.7	25.7	3.7	3.9
1.0	1.0	∞	49.0	0	11	∞	40.8	0	31.5	51.6	51.4	2.1	2.3
1.0	1.1	51.3	49.0	2.3	21	47.8	40.8	1.4	58.8	51.5	51.4	2.4	2.6
1.0	1.3	49.9	49.0	3.8	21	45.0	40.8	2.6	57.7	51.5	51.4	2.7	2.9

Table 11: Criteria and retention limits with varying both γ and γ^{re}. The other parameters are $h = 0.01$,$\lambda = 10$,$\alpha_G = 2$,$\xi_G = 2$,$\alpha_P = 7$,$\beta_P = 7$,$\xi_{LN} = 1$,$\sigma_{LN} = 0.5$ and $\epsilon = 0.01$. The criteria are in percent.

Tables 10 and 11 confirm the results we got from tables 8 and 9. We see that when $\gamma^{re} = \gamma$, the cedent still have arbitrage opportunities for the Gamma and the Pareto distributions with infinite C_ϵ^{ce} and large retention limits (see Appendix A for figures). When γ^{re} is increasing compared to γ, we see C_ϵ getting closer to C_ϵ^{ce} and the retention limit interval shrinking.

To conclude, we see that the loading factors have a lot to say on the choice of the reinsurance retention limits. When a ceding insurance company is looking at different contracts, it should take a look at the loadings offered in the market and also see how much reinsurance is optimal for given loadings.

Variation of the reserve percentile

Another parameter we can vary in the program is the percentile for the capital reserve. It seems that requiring a 99% capital reserve is getting more popular in European countries, however this is not the case everywhere and it surely is interesting to see how the reserve requirements affect the criteria and retention limits. As a reminder, the reserve is defined as the solution to this equation

$$\mathbb{P}[X > q_\epsilon] = \epsilon$$

where q_ϵ is the estimated capital required at $(1 - \epsilon)100$ percentile. The values we are going to vary in our program is ϵ. At first we are going to vary ϵ from 1% to 10%. When we have

$\epsilon = 1\%$, q_ϵ is the solvency capital (reserve) required.

ϵ	Gamma claims				Pareto claims				Log-normal claims			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.01	8.1	7.6	3.8	31.4	8.1	6.5	1.7	65.2	9.4	9.4	9.7	11.7
0.025	9.1	8.7	4.2	31.1	9.1	8.1	1.8	25.4	9.4	9.4	9.7	11.7
0.05	10.1	9.9	4.7	30.6	10.1	9.6	2.0	10.9	9.5	9.4	9.8	11.7
0.1	11.6	11.5	5.7	29.3	11.7	11.5	2.3	4.4	9.5	9.4	9.8	11.8

Table 12: Criteria and retention limits when varying the capital requirements, $h = 0.01$, $\lambda = 10$, $\alpha_P = 3$, $\beta_P = 1$, $\alpha_G = 0.5$, $\xi_G = 1$, $\xi_{LN} = 2$, $\sigma_{LN} = 0.7$, $\gamma = 0.2$, and $\gamma^{re} = 0.3$. The criteria are in percent.

ϵ	Gamma claims				Pareto claims				Log-normal claims			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.01	9.9	9.8	4.5	20.3	8.8	8.2	3.6	57.2	10.3	10.3	3.1	3.3
0.025	10.9	10.8	4.5	19.6	9.8	9.3	3.9	55.9	11.3	11.3	3.0	3.1
0.05	11.8	11.8	4.4	4.4	10.8	10.5	4.4	53.1	12.3	12.3	3.3	3.4
0.1	13.2	13.2	4.5	4.5	12.2	12.1	5.4	15.9	13.6	13.6	4.2	4.3

Table 13: Criteria and retention limits when varying the capital requirements, $h = 0.01$, $\lambda = 10$, $\alpha_P = 7$, $\beta_P = 7$, $\alpha_G = 2$, $\xi_G = 2$, $\xi_{LN} = 1$, $\sigma_{LN} = 0.5$, $\gamma = 0.2$, and $\gamma^{re} = 0.3$. The criteria are in percent.

Since the capital requirements are not decided by the cedent, merely imposed, it cannot be considered as an optimization parameter which the cedent can affect. But it is interesting to see how the criteria and the retention limits vary with the value of ϵ.

The results from tables 12 and 13 are not controversial for the Gamma and the Pareto distributions. When the capital requirements decrease, the C_ϵ^{ce} and C_ϵ increase, which is an expected effect. The need to put away funds for solvency issues is decreasing when capital requirements set by the authorities decreases. The retention limits are forming smaller intervals too when the capital requirements decrease, the cedent needs less reinsurance to cope with the ϵ required by the regulators. For the Log-normal distribution, the capital requirements don't seem to affect the need for reinsurance or the criteria as much. The explanation for this must be the configurations of the distribution. In figure 8, I have plotted the density and x_ϵ side by side for each distribution.

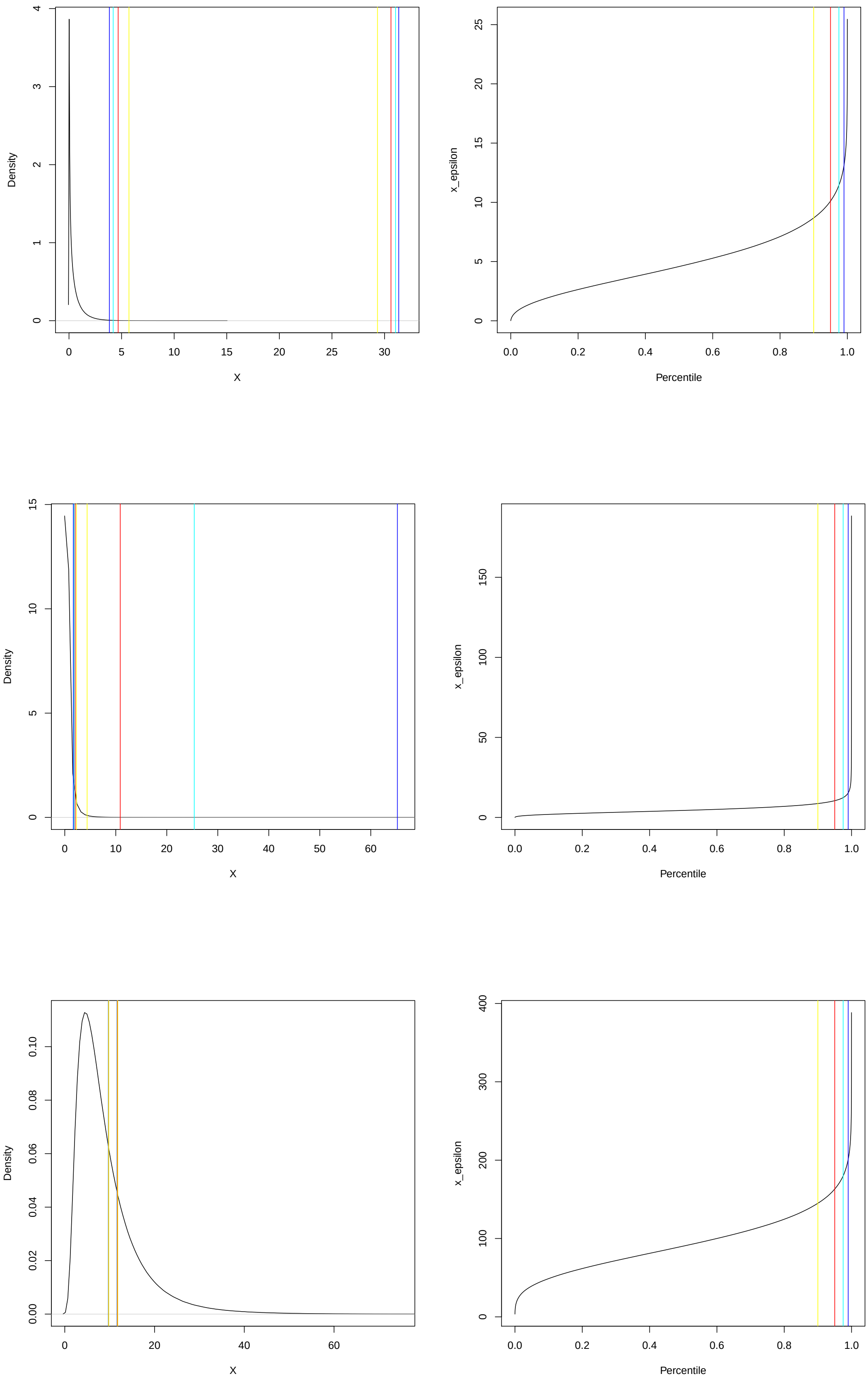


Figure 8: Aggregated distributions with retention limits on the left, q_ϵ on the right. From top to bottom the distributions are Gamma, Pareto and Log-normal. The parameters are $h = 0.01$,$\lambda = 10$,$\alpha_P = 3$,$\beta_P = 1$,$\alpha_G = 0.5$,$\xi_G = 1$,$\xi_{LN} = 2$,$\sigma_{LN} = 0.7$,$\gamma = 0.2$, and $\gamma^{re} = 0.3$. The colour code is the following: $\epsilon = 0.1$: yellow, $\epsilon = 0.05$: red, $\epsilon = 0.25$: light blue and $\epsilon = 0.01$: blue.

The figures on the left side represent the aggregated distributions (simulated) and the respective retention limits. On the right hand side we have the reserve versus the percentile. The retention limits the Log-normal distribution and the as of the Pareto distribution are hard to see because the numbers are close to each other, the left hand side figure of the Log-normal distribution, the retention limits are almost equal (see tables 12 and 13).

One interesting result we can see from each of the couple figures, is that for the Gamma and the Pareto distributions, we notice that the retention limits are following q_ϵ. It seems that b is always greater than q_ϵ. For example, when $\epsilon = 0.01$, we get $b = 31.4$ for the Gamma distribution and $b = 65.2$ for the Pareto distribution (table 12). These values are much larger than the extreme right of the tails of our aggregated distributions on the left. This is actually an important result because it could mean that the optimal reinsurance contract is the a x b with infinite b, which would be another clue for the optimal reinsurance contract. For the Log-normal distribution, the same cannot be said, the retention limits are very stable when we vary ϵ. It can consequently be interesting to investigate the results for stricter capital requirements.

ϵ	**Gamma claims**				**Pareto claims**				**Log-normal claims**			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.0001	5.6	4.9	3.2	31.9	5.7	2.0	1.3	210	6.6	6.3	11.9	20.9
0.0005	6.2	5.6	3.3	31.8	6.3	3.1	1.4	209	9.4	9.4	11.2	17.1
0.001	6.5	5.9	3.4	31.7	6.6	3.8	1.4	209	7.7	7.6	10.9	15.7
0.005	7.5	7.0	3.7	31.5	7.6	5.6	1.6	124	8.9	8.8	10.1	12.8

Table 14: Criteria and retention limits, $h = 0.01$, $\lambda = 10$, $\alpha_P = 3$, $\beta_P = 1$, $\alpha_G = 0.5$, $\xi_G = 1$, $\xi_{LN} = 2$, $\sigma_{LN} = 0.7$, $\gamma = 0.2$, and $\gamma^{re} = 0.3$. The criteria are in percent.

ϵ	**Gamma claims**				**Pareto claims**				**Log-normal claims**			
	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b	C_ϵ^{ce}	C_ϵ	a	b
0.0001	7.2	7.0	4.0	21.0	6.2	5.0	2.9	59.0	7.5	7.5	3.0	3.4
0.0005	7.9	7.7	4.2	20.9	6.8	5.8	3.1	58.6	8.2	8.2	3.1	3.5
0.001	8.3	8.1	4.3	20.8	7.2	6.2	3.2	58.4	8.6	8.6	3.0	3.3
0.005	9.3	9.2	4.5	20.6	8.2	7.5	3.4	57.7	9.7	9.7	3.0	3.2

Table 15: Criteria and retention limits, $h = 0.01$, $\lambda = 10$, $\alpha_P = 7$, $\beta_P = 7$, $\alpha_G = 2$, $\xi_G = 2$, $\xi_{LN} = 1$, $\sigma_{LN} = 0.5$, $\gamma = 0.2$, and $\gamma^{re} = 0.3$. The criteria are in percent.

Tables 14 and 15 have the same structure as tables 12 and 13. We see that the b of all the distributions seem to converge towards a fixed value as ϵ goes to zero. The hypothesis of an infinite b is backed up by the choice of ϵ for all the distributions. These last capital requirements are not realistic, but they show that in case of extremely large claims, which are rare, the optimal reinsurance contract is the a x b contract with infinite b. These types of claims (rare and large), are exactly the type of claims we are interested in when we introduced the "per event" principle. The capital requirements are important to verify that the reinsurance contract we have chosen is really the optimal one.

To conclude this section about using the Panjer recursion as a tool for finding the optimal reinsurance contract for the cedent, we can first say that the distribution and its parameters play a very important role for the choice of optimal reinsurance contract. We saw for example that the behaviour of the Log-normal distribution was not the same as the behaviour of the two other distributions. The key is to chose carefully which distribution and parameters we want to use to model the claim size and claim number.

The loadings of the cedent and the reinsurer also play an important role for optimal reinsurance. They are most of the time decided by market situations and fluctuate a lot from year to year. The closer γ^{re} is to γ, the more profits the cedent can make out of signing up to a reinsurance contract. The strategy is then to assess the market situation and than decide the amount of reinsurance.

Lastly, the capital requirements set by the financial authorities also play an important role when choosing the reinsurance contracts. These are not values we can control as easily as the others, but it is important to be aware that the capital requirements affect the optimum a lot, especially nowadays, when all Europe is preparing to change to the "Solvency II" directive.

5 Concluding Remarks

In this thesis, a framework for optimizing reinsurance per event has been established. The known relevant theoretical results for this topic have been introduced and commented. The conclusion is that there is a lack of numerical examples and of research on the individual claims case, especially for the reserve criteria C_ϵ. We have also introduced some theoretical results for the individual claims. All of this pointed us towards an optimal contract of the non-proportional type, more specifically an a x b contract with infinite b.

In order to investigate the problem of optimal reinsurance we introduced the Panjer recursion for compound distribution. This recursion was used as a tool for optimizing and was implemented in our optimal reinsurance problem as a Fortran program. We used C_ϵ as the optimality criteria. The program allowed us to vary crucial parameters such as the distribution parameters, the loadings, the reserve percentiles ϵ and the precision increment h. This let us investigate the effect on the criteria and the retention limits of the variation of the parameters which allowed us to back the idea that the optimal contract is the a x b with infinite b. We have seen that varying some of the parameters, such as the distribution parameters, caused large oscillations in the results.

The research that has been undertaken for this thesis has highlighted the need for better numerical solutions to the optimal reinsurance contract. Before the industry can base their choice of reinsurance contracts on this kind of numerical analysis, they should try to include as many criteria as possible in their calculations. In this thesis, we only investigated C_ϵ and this alone is not enough to be completely sure about the decision to be made. A careful selection of input parameters is also required, especially for the distributions. Insurance companies usually hold a large database of claims which they can use for this. Other tools than the Panjer recursion can be used for calculations. The Monte Carlo method for simulation and smoothing splines are possibilities.

Appendices

A Figures

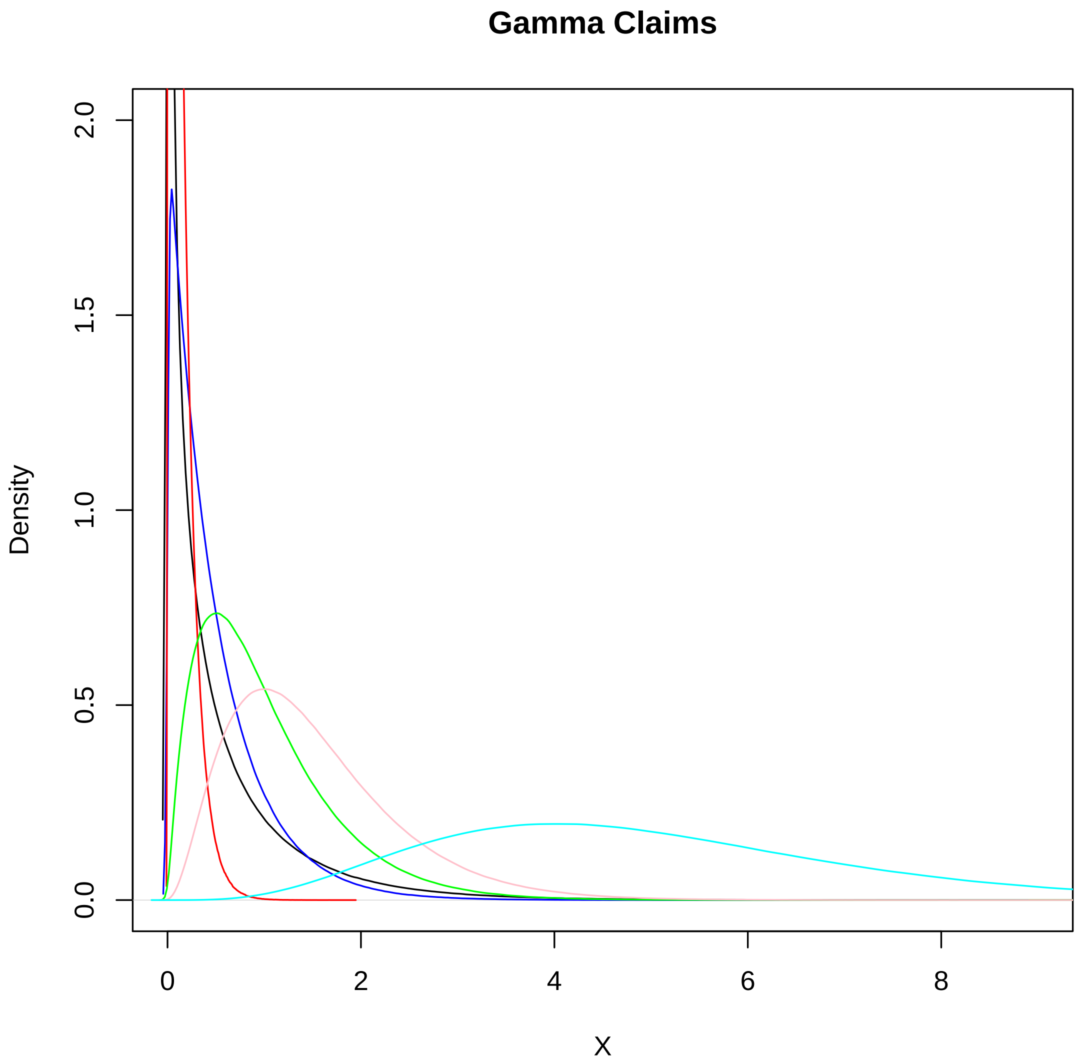


Figure 9: Density of the Gamma distribution with different parameters. $\alpha_G = 0.5$, $\xi_G = 1$ (black), $\alpha_G = 1$, $\xi_G = 2$ (blue), $\alpha_G = 1$, $\xi_G = 8$ (red), $\alpha_G = 2$, $\xi_G = 2$ (green), $\alpha_G = 3$, $\xi_G = 2$ (pink), $\alpha_G = 5$, $\xi_G = 1$ (light blue).

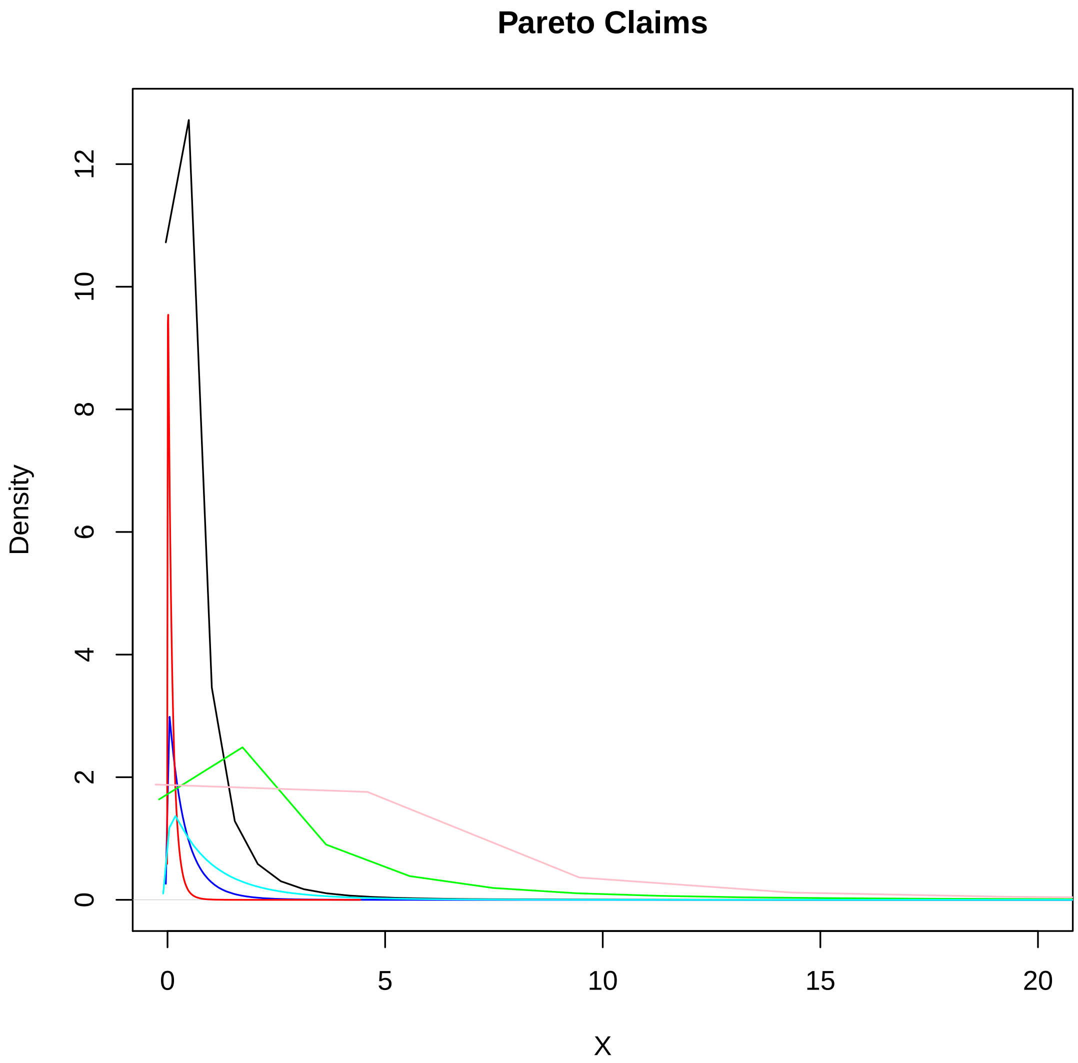


Figure 10: Density of the Pareto distribution with different parameters. $\alpha_P = 3$, $\beta_P = 1$ (black), $\alpha_P = 10$, $\beta_P = 4$ (blue), $\alpha_P = 10$, $\beta_P = 1$ (red), $\alpha_P = 3$, $\beta_P = 5$ (green), $\alpha_P = 3$, $\beta_P = 7$ (pink), $\alpha_P = 7$, $\beta_P = 7$ (light blue).

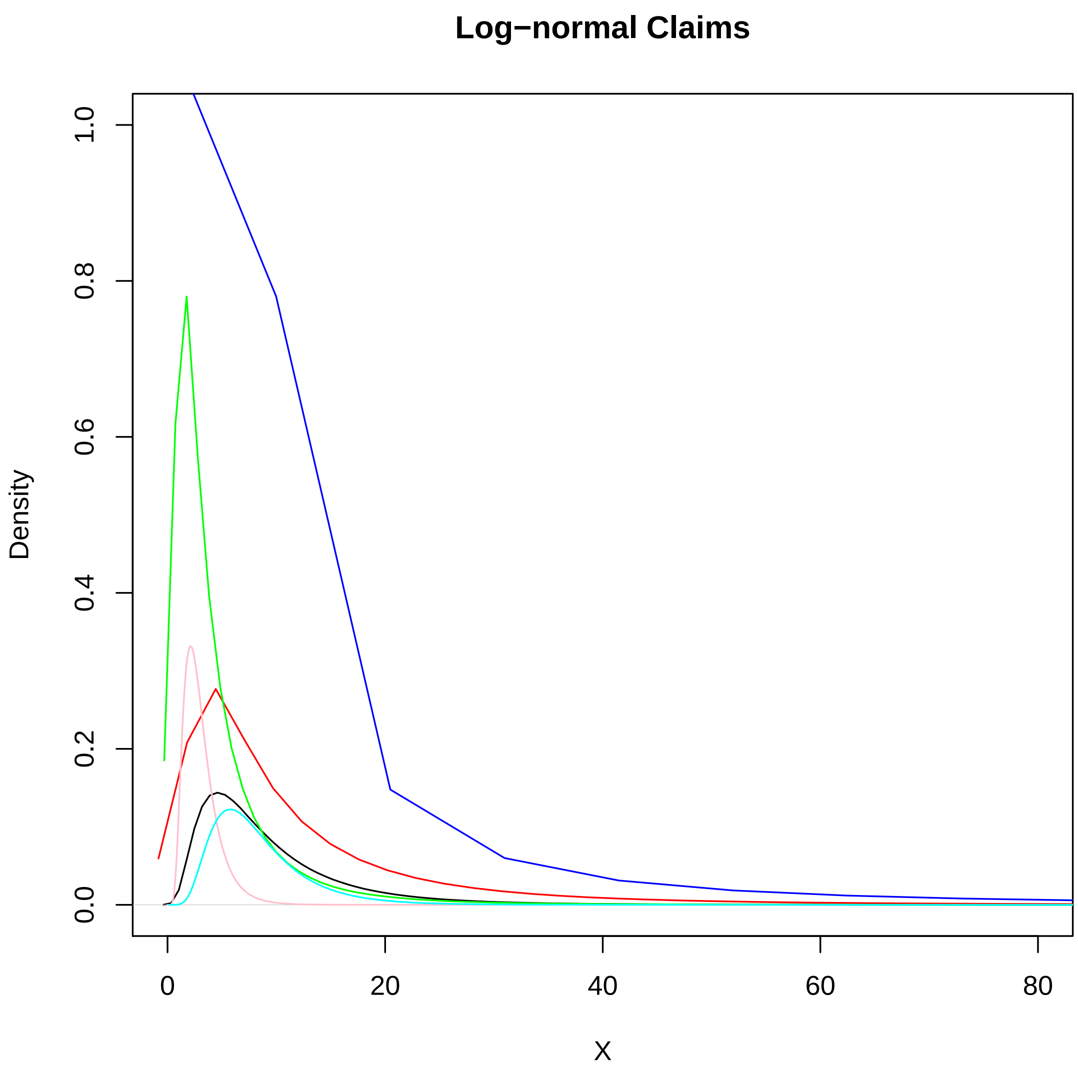


Figure 11: Density of the Log-normal distribution with different parameters. $\xi_{LN} = 2$, $\sigma_{LN} = 0.7$ (black), $\xi_{LN} = 1$, $\sigma_{LN} = 1.5$ (blue), $\xi_{LN} = 2$, $\sigma_{LN} = 1$ (red), $\xi_{LN} = 1$, $\sigma_{LN} = 1$ (green), $\xi_{LN} = 1$, $\sigma_{LN} = 0.5$ (pink), $\xi_{LN} = 2$, $\sigma_{LN} = 0.5$ (light blue).

B Scripts

B.1 R-Scripts

In this appendix, I have listed all the relevant R codes used in the thesis. First, I have the code used for Monte Carlo simulations of the claims.

```
#Parameters


m    <- 100000     #Number of simulations
JmuT <- 10       #Poisson parameter (lambda)

a    <- seq(0.4,2.5,by=0.1)    #Retention
b    <- 2.5                    #Retention upper


#Stochasticity in the model, we must remove the zeros from
#N because of pmin pmax functions in R.
#N[N == 0]     this code to show number 0 in vector

N       <- rpois(m,JmuT)
mm      <- length(N)
remove0 <- c(0)
id0     <- which(N %in% remove0)
if(is.na(id0[1])==TRUE) {
        N <- N
       } else {
        N <- N[-id0]
}

m       <- length(N)
ns      <- sum(N)
S       <- rlnorm(ns,1,0.3)
```

```
#The ZRE function, calculates Zre for each poisson output
#Returns m numbers for each different a (re-contract), loss
#Taken by reinsurers


ZRE <- function(a){
    Zre <- function(x) {pmin(pmax(S[x]-a,0),b)}
    X <- array(0,m)
    s2 <- 0
    i <- 1
    while(i<=m){
        s1 <- s2+1
        s2 <- s1+N[i]-1
        X[i] <- (sum(mapply(Zre,s1:s2)))/N[i]
        i <- i+1
        }
        return(X)
}

LossRe      <- lapply(a,ZRE)
meanLossRe <- mapply(mean,LossRe)
```

The next code calculates the mean of the claims, organized in each of the Poisson N_i.

```
#We need this to calculate the premium paid to the
#Cedent, and then the gain.

Z <- function(x){
    X <- array(0,m)
    s2 <- 0
    i <- 1
    while(i<=m){
        s1 <- s2+1
        s2 <- s1+N[i]-1
        X[i] <- (sum(S[s1:s2]))/N[i]
        i <- i+1
        }
        return(X)
}

Loss     <- lapply(a,Z)
meanLoss <- mapply(mean, Loss)
```

The next code is the discretize program mentioned in section 4.2, it should be noted that this particular code only works for round numbers.

```
discretize <- function(mu,sigma,from,to,delta){
        d    <- seq(from,to,by=(delta))
        disc <- array(0,length(d)-1)
        disc[1] <- plnorm(delta/2,mu,sigma)
        for(i in (2:to)) {
                disc[i] <- plnorm((i-1)*delta+delta/2,mu,sigma)-
                           plnorm((i-1)*delta-delta/2,mu,sigma)
        }

        return(disc)
}

ab <- discretize(0,2,0,22,0.5)
```

```
#This is the code for plotting the reserves
m <- 10000
JmuT <- 10
theta <- -0.5
sigma <- 1
a <- 5

Xce <- array(0,m)
N = rpois(m,JmuT)
for (i in 1:m) {
Z = rlnorm(N[i],theta,sigma)
if(N[i]>0) Xce[i]=sum(pmin(Z,a))
}

resfunc <- function(x){
  res <- sort(Xce)[(x)*m]

  return(res)
}
aa <- seq(0,1,by=0.0001)



#Reserve for Log-normal
theta <- -0.5
sigma <- 1
plot(seq(0,0.9999,by=0.0001),ylim=c(0,30),resfunc(aa),type='l',
col='blue',xlab='Percentile',ylab='x_epsilon')




#Reserve for Pareto
alpha <- 3
beta  <- 1
for (i in 1:m) {
Z = rpareto(N[i],alpha,beta)
if(N[i]>0) Xce[i]=sum(pmin(Z,a))
}
lines(seq(0,0.9999,by=0.0001),resfunc(aa),type='l')


#Reserve for Gamma
xi   <- 0.5
sigma<- 1
for (i in 1:m) {
Z = rgamma(N[i],xi,sigma)
if(N[i]>0) Xce[i]=sum(pmin(Z,a))
}
lines(seq(0,0.9999,by=0.0001),resfunc(aa),col='red',type='l')
```

```
PanjerPoisson <- function(p,lambda)
{
  cumul <- f <- exp(-lambda*sum(p))
  r <- length(p)
  j <- 0
  repeat
  { j <- j+1
```

```
    m <- min(j , r )
    last <- lambda/j * sum(  1:m * head(p,m)  * rev( tail (f ,m))  )
    f <- c(f , last )
    cumul <- cumul + last
    if(cumul>0.999999)  break
    }
  return(m)
}


PanjerPoisson(ab,10)
```

B.2 How to use the Fortran Programs

Here is a brief explanation of how to use the Fortran programs developed for the Panjer recursion. There are also some descriptions of the different parameters. In order to run the Fortran programs, we must first compile them using a command window. This is done by calling

```
> gfortran pareto.f -o pareto

> gfortran gamma.f -o gamma
```

Once the programs are compiled, we must put the parameters in separate files: 'gamma.par' and 'pareto.par'. For each of the programs, the parameters we can vary are the distribution parameters, λ, the loading γ and γ^{re}, the lattice width h and the percentile ϵ:

```
gamma.par:
lam     alf    xi    gam    gamre
 10      1      2     0.2    0.3
h       eps
0.1     0.05

pareto.par:
lam     alf    bet    gam    gamre
 10      3      1      0.2    0.3
h        eps
0.1      0.05
```

When the wanted parameters are selected we get the following output in our command window, first the lower retention limit of our a and b contract:a, then we get the length of the interval between a and b. In order to find the value of the second retention limit b, you sum a and the length of the interval.

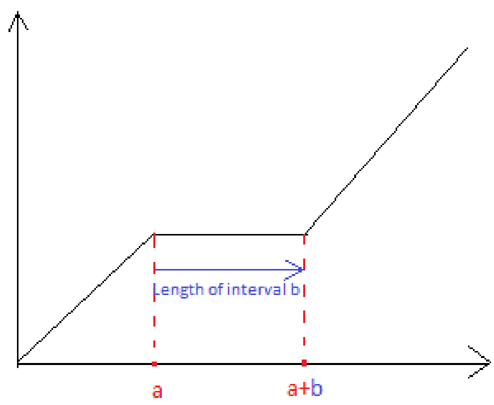


Figure 12: Model of the retention limits we get from the Fortran program

Finally we get the value of the ratio between the expected gain and the net reserve of the cedent:

```
> ./pareto
    Lower limit (a) Length of interval (b) Value ratio
         2.0000                9.0000          0.1017
 Value ratio without reinsurance
         0.0960

> ./gamma
    Lower limit (a), Length of interval (b), Value ratio
         7.1000               26.0000             0.1115
 Value ratio without reinsurance
         0.1105
```

Here is the time it takes to run the program, using linux built-in function:

```
#With h=0.01
>time ./gamma
    Lower limit (a), Length of interval (b), Value ratio
         4.6700               25.9400             0.1010
 Value ratio without reinsurance
         0.0989
real    0m2.572s
user    0m2.533s
sys     0m0.001s
#With h=0.001
>time ./gamma
    Lower limit (a), Length of interval (b), Value ratio
         4.6720               25.9830             0.1009
 Value ratio without reinsurance
         0.0989
real    6m20.075s
user    6m19.999s
sys     0m0.007s
```

```
#h=0.01
>time ./pareto
    Lower limit (a) Length of interval (b) Value ratio
         2.0300                8.8400          0.1010
 Value ratio without reinsurance
         0.0955

real    0m1.491s
user    0m1.447s
sys     0m0.001s
#h=0.001
>time ./pareto
    Lower limit (a) Length of interval (b) Value ratio
         2.0340                8.8810          0.1009
 Value ratio without reinsurance
         0.0954

real    3m6.994s
user    3m6.940s
sys     0m0.003s
```

B.3 Fortran Programs

The program for the pareto function:

```
      implicit none
      real* 8  lam,alf,bet,gam,gamre,h,eps,f(0:1000000)
      real*8 func,crit,crito,ao,bo,fo,fao,a,b,ffao,fac,ffaom
      integer  i,ia,ib,iao,ibo,imax,ia1,ia3,niter
      parameter(niter=20)

      open(unit=10,file='pareto.par')
      open(unit=20,file='pareto.res')

      read(10,*)
      read(10,*)lam,alf,bet,gam,gamre
      read(10,*)
      read(10,*)h,eps

      call density(alf,bet,h,imax,f)

c-----------------------------------------------------------------------
c     Optimizing with respect to the limits of the reinsurance treaties

c     First find the initial bracket on ia
      fac=imax/50.0
      ffaom=-1000
      do i=0,9
        ia=i*fac
        ffao=fao(lam,alf,bet,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
            ffaom=ffao
            iao=ia
            ibo=ib
        endif
      enddo

c     Then the optimum with respect to ia itself
      ia1=iao-fac
      if(ia1.lt.0)ia1=0
      ia3=iao+fac
      if(ia3.gt.imax)ia3=imax
      do i=1,niter
        ia=0.5*(ia1+iao)
        ffao=fao(lam,alf,bet,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
          ia3=iao
          iao=ia
          ffaom=ffao
          ibo=ib

                 else
          ia1=ia
        endif
        if(iao-ia1.eq.1)  ia1=iao
        ia=0.5*(ia3+iao)
        ffao=fao(lam,alf,bet,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
          ia1=iao
          iao=ia
          ffaom=ffao
          ibo=ib
                 else
          ia3=ia
        endif
        if(ia3-iao.eq.1)ia3=iao
       if(ia3-ia1.lt.2)goto 1
      enddo
 1    continue

c     Writing results
      write(*,101)
      write(*,100)h*iao,h*ibo,ffaom
      write(*,102)
      write(*,100)func(lam,alf,bet,20,0,gam,gamre,h,imax,f,eps)
```

```
      write(20,101)
      write(20,100)h*iao,h*ibo,ffaom
      write(20,102)
      write(20,100)func(lam,alf,bet,20,0,gam,gamre,h,imax,f,eps)

 101  format(1x,'   Lower limit (a)   Length of interval (b)   Value rat
     1io')
 102  format(1x, 'Value ratio without reinsurance')
 100  format(1x,f14.4,f24.4,f17.4)
      end

c-----------------------------------------------------------------------

      real*8 function fao(lam,alf,bet,gam,gamre,h,imax,f,eps,ia,ibo)
      implicit none
      real* 8  lam,alf,bet,gam,gamre,h,eps,f(0:1000000)
      real*8 a,b,resce,xeps,ece,ere,func,f2,fo,f1
      integer ia,ibo,imax,ib1,ib2,ib3,iter,niter,ia1,ia2,ia3,iter2
      parameter(niter=20)

c     Optimizing with respect to ib using bisection

        ib1=0
        f1=func(lam,alf,bet,ia,ib1,gam,gamre,h,imax,f,eps)
        ib3=imax-ia
        ibo=0.5*ib3
        fo=func(lam,alf,bet,ia,ibo,gam,gamre,h,imax,f,eps)
        do iter=1,niter
          ib2=0.5*(ib1+ibo)
          f2=func(lam,alf,bet,ia,ib2,gam,gamre,h,imax,f,eps)
          if(f2.gt.fo) then
            ib3=ibo
            ibo=ib2
            fo=f2
                    else
            ib1=ib2
          endif
          if(ibo-ib1.eq.1)then
                if(fo.gt.f1) then
                  ib1=ibo
                           else
                  ibo=ib1
                endif
          endif
          ib2=0.5*(ib3+ibo)
          f2=func(lam,alf,bet,ia,ib2,gam,gamre,h,imax,f,eps)
          if(f2.gt.fo) then
            ib1=ibo
            ibo=ib2
            fo=f2
                    else
            ib3=ib2
          endif
          if(ib3-ibo.eq.1)ib3=ibo
          if(ib3-ib1.lt.2)goto 1
        enddo
 1      continue
        fao=fo
      end

c-----------------------------------------------------------------------

      real*8 function func(lam,alf,bet,ia,ib,gam,gamre,h,imax,f,eps)
      implicit none
      real* 8  lam,alf,bet,gam,gamre,h,eps
      real*8 resce,xeps,ece,ere,f(0:1000000)
      integer ia,ib,imax,i

c     Expected gain divided on cedent net reserve returned

      ece=lam*gam*(bet/(alf-1))
      ere=lam*gamre*(bet/(alf-1))*((1+ia*h/bet)**(-alf+1))
     1        *(1-((1+ib*h/(ia*h+bet))**(-alf+1)))
      xeps=resce(lam,ia,ib,h,imax,f,eps)
      if(xeps.lt.0.000001) then
         if(ece.gt.ere) then
```

```
            func=1000000
                         else
            func=-100000
          endif
                              else
           func=(ece-ere)/xeps
        endif

      end

c------------------------------------------------------------
      real*8 function resce(lam,ia,ib,h,imax,f,eps)
      implicit none
      real*8 lam,h,eps,density,dist
      real*8 f(0:1000000),g(0:1000000),sum,f1(0:1000000)
      integer i,ia,ib,imax,i1,j,imf,img,imf2

c     Cedent net  reserve using the Panjer recursion

      do i=0,ia-1
        f1(i)=f(i)
      enddo
      f1(ia)=0.0
      do i=ia,ia+ib
         f1(ia)=f1(ia)+f(i)
      enddo
      imf=imax-ib
      do i=ia+1,imf
         f1(i)=f(i+ib)
      enddo

      img=imf*(lam+6.0*dsqrt(lam))
      g(0)=dexp(-lam+lam*f1(0))
      sum=g(0)
      do i=1,img
        i1=i-1
        if(sum.gt.1-eps) goto 1
        g(i)=0
        imf2=min(i,imf)
        do j=1,imf2
          g(i)=g(i)+j*f1(j)*g(i-j)
        enddo
        g(i)=g(i)*lam/i
        sum=sum+g(i)
      enddo
      write(*,100)ia,ib,imax
 100  format(1x,'Error: h too large or imax too small',4i6)
      stop

 1    continue
      if(i1.gt.0) then
         resce=((i1-1)+(1-eps-(sum-g(i1)))/g(i1))*h
                 else
         resce=0.0
      endif
      end

c------------------------------------------------------------------


      subroutine density(alf,bet,h,imax,f)
      implicit none
      real*8 alf,bet,h,f(0:1000000),sum,eta,dold,dnew
      integer imax,i
      parameter(eta=0.0000001)

c     Computes the input density function

      imax=(eta**(-1.0/alf)-1)*bet/h

c      Former version
c      sum=0.0
c      do i=0,imax
c         f(i)=(alf/bet)*(1+i*h/bet)**(-1-alf)
c        sum=sum+f(i)
c      enddo
```

```
c          do i=0,imax
c             f(i)=f(i)/sum
c          enddo

c          New version
           dold=0
           do i=0,imax
              dnew=1-(1+(i+0.5)*h/bet)**(-alf)
              f(i)=dnew-dold
              dold=dnew
           enddo

         end
```

The program for the gamma function:

```
      implicit none
      real*8  lam,alf,xi,gam,gamre,h,eps,f(0:1000000)
      real*8 func,crit,crito,ao,bo,fo,fao,a,b,ffao,fac,ffaom
      integer  i,ia,ib,iao,ibo,imax,ia1,ia3,niter
      parameter(niter=20)


      open(unit=10,file='gamma.par')
      open(unit=20,file='gamma.res')

      read(10,*)
      read(10,*)lam,alf,xi,gam,gamre
      read(10,*)
      read(10,*)h,eps


      call density(alf,xi,h,imax,f)

c-----------------------------------------------------------------------
c     Optimizing with respect to the limits of the reinsurance treaties

c     First find the initial bracket on ia
      fac=imax/50.0
      ffaom=-1000
      do i=0,9
        ia=i*fac
        ffao=fao(lam,alf,xi,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
            ffaom=ffao
            iao=ia
            ibo=ib
        endif
      enddo


c     Then the optimum with respect to ia itself
      ia1=iao-fac
      if(ia1.lt.0)ia1=0
      ia3=iao+fac
      if(ia3.gt.imax)ia3=imax
      do i=1,niter
        ia=0.5*(ia1+iao)
        ffao=fao(lam,alf,xi,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
          ia3=iao
          iao=ia
          ffaom=ffao
          ibo=ib

                 else
          ia1=ia
        endif
        if(iao-ia1.eq.1)  ia1=iao
        ia=0.5*(ia3+iao)
        ffao=fao(lam,alf,xi,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
          ia1=iao
          iao=ia
          ffaom=ffao
          ibo=ib
                 else
          ia3=ia
        endif
        if(ia3-iao.eq.1)ia3=iao
       if(ia3-ia1.lt.2)goto 1
      enddo
 1    continue

c     Writing results
      write(*,101)
      write(*,100)h*iao,h*ibo,ffaom
      write(*,102)
      write(*,100)func(lam,alf,xi,10,0,gam,gamre,h,imax,f,eps)
      write(20,101)
```

```fortran
      write(20,100)h*iao,h*ibo,ffaom
      write(20,102)
      write(20,100)func(lam,alf,xi,20,0,gam,gamre,h,imax,f,eps)

 101  format(1x,'   Lower limit (a),   Length of interval (b),   Value r
     1atio')

 100  format(1x,f14.4,f24.4,f19.4)
 102  format(1x, 'Value ratio without reinsurance')

      end

c-----------------------------------------------------------------------

      real*8 function fao(lam,alf,xi,gam,gamre,h,imax,f,eps,ia,ibo)
      implicit none
      real* 8  lam,alf,xi,gam,gamre,h,eps,f(0:1000000)
      real*8 a,b,resce,xeps,ece,ere,func,f2,fo,f1
      integer ia,ibo,imax,ib1,ib2,ib3,iter,niter,ia1,ia2,ia3,iter2
      parameter(niter=20)

c     Optimizing with respect to ib using bisection

        ib1=0
        f1=func(lam,alf,xi,ia,ib1,gam,gamre,h,imax,f,eps)
        ib3=imax-ia
        ibo=0.5*ib3
        fo=func(lam,alf,xi,ia,ibo,gam,gamre,h,imax,f,eps)
        do iter=1,niter
          ib2=0.5*(ib1+ibo)
          f2=func(lam,alf,xi,ia,ib2,gam,gamre,h,imax,f,eps)
          if(f2.gt.fo) then
            ib3=ibo
            ibo=ib2
            fo=f2
                      else
            ib1=ib2
          endif
          if(ibo-ib1.eq.1)then
                if(fo.gt.f1) then
                  ib1=ibo
                            else
                  ibo=ib1
                endif
          endif
          ib2=0.5*(ib3+ibo)
          f2=func(lam,alf,xi,ia,ib2,gam,gamre,h,imax,f,eps)
          if(f2.gt.fo) then
            ib1=ibo
            ibo=ib2
            fo=f2
                      else
            ib3=ib2
          endif
          if(ib3-ibo.eq.1)ib3=ibo
          if(ib3-ib1.lt.2)goto 1
        enddo
 1      continue
        fao=fo
      end

c-----------------------------------------------------------------------

      real*8 function func(lam,alf,xi,ia,ib,gam,gamre,h,imax,f,eps)
      implicit none
      real* 8  lam,alf,xi,gam,gamre,h,eps
      real*8 resce,xeps,ece,ere,f(0:1000000)
      real*8 a,b,p1,p2,p3,p4,gammp
      integer ia,ib,imax,i

c     Expected gain divided on cedent net reserve returned

      ece=lam*gam*xi

      a=ia*h
      b=ib*h
```

```
      p1=gammp(alf+1,a*alf/xi)
      p2=gammp(alf+1,(a+b)*alf/xi)
      p3=gammp(alf,a*alf/xi)
      p4=gammp(alf,(a+b)*alf/xi)
      ere=lam*gamre*(xi*(p2-p1)-a*(p4-p3)+b*(1-p4))

      xeps=resce(lam,ia,ib,h,imax,f,eps)
      if(xeps.lt.0.000001) then
         if(ece.gt.ere) then
           func=1000000
                        else
           func=-100000
         endif
                             else
         func=(ece-ere)/xeps
       endif

      end

c-----------------------------------------------------------
      real*8 function resce(lam,ia,ib,h,imax,f,eps)
      implicit none
      real*8 lam,h,eps,density,dist
      real*8 f(0:1000000),g(0:1000000),sum,f1(0:1000000)
      integer i,ia,ib,imax,i1,j,imf,img,imf2

c     Cedent net  reserve using the Panjer recursion

      do i=0,ia-1
        f1(i)=f(i)
      enddo
      f1(ia)=0.0
      do i=ia,ia+ib
         f1(ia)=f1(ia)+f(i)
      enddo
      imf=imax-ib
      do i=ia+1,imf
         f1(i)=f(i+ib)
      enddo

      img=imf*(lam+6.0*dsqrt(lam))
      g(0)=dexp(-lam+lam*f1(0))
      sum=g(0)
      do i=1,img
        i1=i-1
        if(sum.gt.1-eps) goto 1
        g(i)=0
        imf2=min(i,imf)
        do j=1,imf2
          g(i)=g(i)+j*f1(j)*g(i-j)
        enddo
        g(i)=g(i)*lam/i
        sum=sum+g(i)
      enddo
      write(*,100)ia,ib,imax
 100  format(1x,'Error: h too large or imax too small',4i6)
      stop

 1    continue
      if(i1.gt.0) then
         resce=((i1-1)+(1-eps-(sum-g(i1)))/g(i1))*h
                  else
         resce=0.0
      endif
      end

c-------------------------------------------------------------------


      subroutine density(alf,xi,h,imax,f)
      implicit none
      real*8 alf,xi,h,f(0:1000000),sum,eta,gammp,z,dnew,dold
      integer imax,i
      parameter(eta=0.00000001)

c     Computes the input density function
```

```
      imax=xi+4.0*xi/dsqrt(alf)
      do i=1,1000
         z=imax*h*alf/xi
         if(gammp(alf,z).gt.1-eta) goto 1
         imax=imax+(xi/dsqrt(alf))/h
      enddo
 1    continue

c       Former version
c       sum=0.0
c       f(0)=0.0
c       do i=1,imax
c          f(i)=dexp(dlog(i*h)*(alf-1)-alf*(i*h)/xi)
c          sum=sum+f(i)
c       enddo
c       do i=0,imax
c          f(i)=f(i)/sum
c       enddo

c       New version
        dold=0
        do i=0,imax
           z=(i+0.5)*h*alf/xi
           dnew= gammp(alf,z)
           f(i)=dnew-dold
           dold=dnew
        enddo


      end
c--------------------------------------------

      real*8 function gammp(a,x)
      implicit none
      real*8 a, x,gammcf,gamser,gln

      if(x.lt.a+1) then
        call gser(gamser,a,x,gln)
        gammp=gamser
                  else
        call gcf(gammcf,a,x,gln)
        gammp=1.0-gammcf
      endif

      end

c------------------------------------------------------------------

      subroutine gser(gamser,a,x,gln)
      real*8 a,gamser,gln,x,eps,ap,del,sum,gamfl
      integer itmax,n
      parameter(itmax=100,eps=3.0d-7)

      gln=gamfl(a)
      if(x.le.0) then
         gamser=0
         return
      endif
      ap=a
      sum=1.0/a
      del=sum
      do n=1,itmax
        ap=ap+1
        del=del*x/ap
        sum=sum+del
        if(dabs(del).lt.dabs(sum)*eps) goto 1
      enddo
      pause 'a too large, itmax too smal in gser'
 1    continue
      gamser=sum*dexp(-x+a*dlog(x)-gln)

      end

c----------------------------------------------------------
```

```
      subroutine gcf(gammcf,a,x,gln)
      implicit none
      real*8 gammcf,a,gln,x,eps,fpmin,an,b,c,d,del,h,gamfl
      integer itmax,i
      parameter(itmax=100,eps=3.0d-7,fpmin=1.0d-30)

      gln=gamfl(a)
      b=x+1.0-a
      c=1.0/fpmin
      d=1.0/b
      h=d
      do i=1,itmax
         an=-i*(i-a)
         b=b+2.0
         d=an*d+b
         if(dabs(d).lt.fpmin)d=fpmin
         c=b+an/c
         if(dabs(c).lt.fpmin)c=fpmin
         d=1.0/d
         del=d*c
         h=h*del
         if(dabs(del-1.0).lt.eps)goto 1
      enddo
      pause 'a too large, itmax too small in gcf'
 1    continue
      gammcf=dexp(-x+a*dlog(x)-gln)*h
      end

c---------------------------------------------------

      real*8 function gamfl(z)
      implicit none
      real*8 z,coef(6),tmp,sum,stp,x
      integer j

      coef(1)=76.18009173
      coef(2)=-86.50532033
      coef(3)=24.01409822
      coef(4)=-1.231739516
      coef(5)=0.0012085003
      coef(6)=-0.536382/(10**5)
      stp=2.50662827465

      x=z-1
      tmp=x+5.5
      tmp=(x+0.5)*dlog(tmp)-tmp
      sum=1.0
      do j=1,6
         x=x+1
         sum=sum+coef(j)/x
      end do

      gamfl=tmp+dlog(stp*sum)

      end
```

The program for the Log-normal function:

```
      implicit none
      real*8  lam,sig,xi,gam,gamre,h,eps,f(0:1000000)
      real*8 func,crit,crito,ao,bo,fo,fao,a,b,ffao,fac,ffaom
      integer  i,ia,ib,iao,ibo,imax,ia1,ia3,niter
      parameter(niter=20)


      open(unit=10,file='logn.par')
      open(unit=20,file='logn.res')

      read(10,*)
      read(10,*)lam,sig,xi,gam,gamre
      read(10,*)
      read(10,*)h,eps


      call density(sig,xi,h,imax,f)

c-----------------------------------------------------------------------
c     Optimizing with respect to the limits of the reinsurance treaties

c     First find the initial bracket on ia
      fac=imax/40.0
      ffaom=-1000
      do i=0,9
        ia=i*fac
        ffao=fao(lam,sig,xi,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
            ffaom=ffao
            iao=ia
            ibo=ib
        endif
c         write(*,*)ia,ib,ffao
      enddo
c      stop

c     Then the optimum with respect to ia itself
      ia1=iao-fac
      if(ia1.lt.0)ia1=0
      ia3=iao+fac
      if(ia3.gt.imax)ia3=imax
      do i=1,niter
        ia=0.5*(ia1+iao)
        ffao=fao(lam,sig,xi,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
          ia3=iao
          iao=ia
          ffaom=ffao
          ibo=ib

                  else
          ia1=ia
        endif
        if(iao-ia1.eq.1) ia1=iao
        ia=0.5*(ia3+iao)
        ffao=fao(lam,sig,xi,gam,gamre,h,imax,f,eps,ia,ib)
        if(ffao.gt.ffaom) then
          ia1=iao
          iao=ia
          ffaom=ffao
          ibo=ib
                  else
          ia3=ia
        endif
        if(ia3-iao.eq.1)ia3=iao
       if(ia3-ia1.lt.2)goto 1
      enddo
 1    continue

c     Writing results
      write(*,101)
      write(*,100)h*iao,h*ibo,ffaom
      write(*,102)
      write(*,100)func(lam,sig,xi,10,0,gam,gamre,h,imax,f,eps)
```

```
      write(20,101)
      write(20,100)h*iao,h*ibo,ffaom
      write(20,102)
      write(20,100)func(lam,sig,xi,20,0,gam,gamre,h,imax,f,eps)

 101  format(1x,'   Lower limit (a),   Length of interval (b),   Value r
     1atio')

 100  format(1x,f14.4,f24.4,f19.4)
 102  format(1x, 'Value ratio without reinsurance')

      end

c----------------------------------------------------------------------

      real*8 function fao(lam,sig,xi,gam,gamre,h,imax,f,eps,ia,ibo)
      implicit none
      real* 8  lam,sig,xi,gam,gamre,h,eps,f(0:1000000)
      real*8 a,b,resce,xeps,ece,ere,func,f2,fo,f1
      integer ia,ibo,imax,ib1,ib2,ib3,iter,niter,ia1,ia2,ia3,iter2
      parameter(niter=20)

c     Optimizing with respect to ib using bisection

        ib1=0
        f1=func(lam,sig,xi,ia,ib1,gam,gamre,h,imax,f,eps)
        ib3=imax-ia
        ibo=0.5*ib3
        fo=func(lam,sig,xi,ia,ibo,gam,gamre,h,imax,f,eps)
        do iter=1,niter
          ib2=0.5*(ib1+ibo)
          f2=func(lam,sig,xi,ia,ib2,gam,gamre,h,imax,f,eps)
          if(f2.gt.fo) then
            ib3=ibo
            ibo=ib2
            fo=f2
                    else
            ib1=ib2
          endif
          if(ibo-ib1.eq.1)then
                if(fo.gt.f1) then
                  ib1=ibo
                           else
                  ibo=ib1
                endif
          endif
          ib2=0.5*(ib3+ibo)
          f2=func(lam,sig,xi,ia,ib2,gam,gamre,h,imax,f,eps)
          if(f2.gt.fo) then
            ib1=ibo
            ibo=ib2
            fo=f2
                    else
            ib3=ib2
          endif
          if(ib3-ibo.eq.1)ib3=ibo
          if(ib3-ib1.lt.2)goto 1
        enddo
 1      continue
        fao=fo
      end

c----------------------------------------------------------------------

      real*8 function func(lam,sig,xi,ia,ib,gam,gamre,h,imax,f,eps)
      implicit none
      real* 8  lam,sig,xi,gam,gamre,h,eps
      real*8 resce,xeps,ece,ere,f(0:1000000)
      real*8 a,b,p1,p2,p3,p4,z
      integer ia,ib,imax,i

c     Expected gain divided on cedent net reserve returned

      ece=lam*gam*xi

      a=ia*h
```

```
      b=ib*h
      z=dlog((a+b)/xi)/sig -0.5*sig
      call cumnorm(z,p1)
      z=dlog(a/xi)/sig -0.5*sig
      call cumnorm(z,p2)
      z=dlog((a+b)/xi)/sig+0.5*sig
      call cumnorm(z,p3)
      z=dlog(a/xi)/sig+0.5*sig
      call cumnorm(z,p4)
      ere=lam*gamre*(xi*(p2-p1)-a*(p4-p3)+b*p4)

      xeps=resce(lam,ia,ib,h,imax,f,eps)
      if(xeps.lt.0.000001) then
         if(ece.gt.ere) then
           func=1000000
                      else
           func=-100000
         endif
                        else
         func=(ece-ere)/xeps
       endif

      end

c-----------------------------------------------------------
      real*8 function resce(lam,ia,ib,h,imax,f,eps)
      implicit none
      real*8 lam,h,eps,density,dist
      real*8 f(0:1000000),g(0:1000000),sum,f1(0:1000000)
      integer i,ia,ib,imax,i1,j,imf,img,imf2

c     Cedent net  reserve using the Panjer recursion

      do i=0,ia-1
        f1(i)=f(i)
      enddo
      f1(ia)=0.0
      do i=ia,ia+ib
         f1(ia)=f1(ia)+f(i)
      enddo
      imf=imax-ib
      do i=ia+1,imf
         f1(i)=f(i+ib)
      enddo

      img=imf*(lam+6.0*dsqrt(lam))
      g(0)=dexp(-lam+lam*f1(0))
      sum=g(0)
      do i=1,img
        i1=i-1
        if(sum.gt.1-eps) goto 1
        g(i)=0
        imf2=min(i,imf)
        do j=1,imf2
          g(i)=g(i)+j*f1(j)*g(i-j)
        enddo
        g(i)=g(i)*lam/i
        sum=sum+g(i)
      enddo
      write(*,100)ia,ib,imax
 100  format(1x,'Error:␣h␣too␣large␣or␣imax␣too␣small',4i6)
      stop

 1    continue
      if(i1.gt.0) then
         resce=((i1-1)+(1-eps-(sum-g(i1)))/g(i1))*h
                   else
         resce=0.0
      endif
      end

c------------------------------------------------------------------


      subroutine density(sig,xi,h,imax,f)
      implicit none
```

```
      real*8 sig,xi,h,f(0:1000000),sum,z,dnew,dold,g
      integer imax,i

c     Computes the input density function
      imax=xi*dexp(-0.5*sig*sig+6.0*sig)/h

c      Former version
c      sum=0.0
c      f(0)=0.0
c      do i=1,imax
c         f(i)=dexp(dlog(i*h)*(alf-1)-alf*(i*h)/xi)
c         sum=sum+f(i)
c      enddo
c      do i=0,imax
c         f(i)=f(i)/sum
c      enddo

c      New version
       dold=0
       do i=0,imax
          z=dlog((i+0.5)*h/xi)/sig+0.5*sig
          call cumnorm(z,g)
          dnew=1-g
          f(i)=dnew-dold
          dold=dnew
       enddo


      end
c-------------------------------------------------------------




      subroutine cumnorm(x,g)
      implicit none
      real*8 x,g,z,q,y

c     The function computes 1- the standard normal integral by a method
c     provided by H.P.Langtangen (it seems to be one of the algorithms
c     given in Abramowitz and Stegun: Handbook of mathematical functions).
c     It has been checked against Owen's tables and found to be correct
c     to 6 decimals in the range -4.0 to 4.0 for the argument.
c
c     Input variable: x = the argument to the integral.
c
c

      z=1.0/(1.0+0.2316419*abs(x))
      q=z*(0.127414796+z*(-0.142248368+z*(0.7107068705+
     *  z*(-0.7265760135+z*0.5307027145))))*exp(-0.5*x*x)
      y=q
      if(x.le.0.0) y=1.0-q


      g=y



      end
```

References

[1] Adelson R.M. (1966). Compound Poisson Distributions. *Operational Research Quarterly*, 17, 73-75

[2] Borch K. (1960) An attempt to determine the optimum amount of stop loss reinsurance. *XVIth International Congress of Actuaries*, Brussels, Volume 1, 597.

[3] Bølviken E. (2014), *Computation and Modelling in Insurance and Finance*. International Series on Actuarial Science, Cambridge.

[4] Cai J. and Wei W. (2012). Optimal reinsurance with positively dependent risks. *Insurance: Mathematics and Economics*, 50, 57-63.

[5] Centento M.L. and Simoes, O. (2009). Optimal reinsurance. *RACSAM*, 103, 387-404.

[6] Centento M.L. and Guerra M. (2010). The optimal reinsurance strategy - the individual claim case. *Insurance: Mathematics and Economics*, 46, 450-460.

[7] Cheung K.C., Sung K.C.J., Yam S.C.P. and Yung S.P. (2011). Optimal reinsurance under law-invariant risk measures. *Scandinavian Actuarial Journal*, 1-20.

[8] Cheung K.C., Sung K.C.J. and Yam S.C.P. (2014). Risk-minimizing reinsurance protection for multivariate risks. *Journal of Risk and Insurance*, 81, 219-236.

[9] Cruz M.G., Peters G.W. and Shevchenko P. (2015). *Fundamental Aspects of Operational Risk and Insurance Analytics: A Handbook of Operational Risk*. John Wiley and Sons.

[10] Dickson, D.C.M and Waters, H.R. (2006). Optimal Dynamic Reinsurance. *Astin Bulletin*, 36, 415-432

[11] Markowitz H. (1952). Portfolio Selection. *The Journal of Finance.*

[12] Matloff N. (2011). *The Art of R Programming: A Tour of Statistical Software Design*. No Starch Press, USA

[13] Panjer H. (1981). Recursive Evaluation of a Family of Compound Distributions. *Astin Bulletin.*

[14] Walhin J.F. (2012). *La Réassurance*. Larcier, Bruxelles, 2^e édition.