Mémoire présenté le :

pour l'obtention du Diplôme Universitaire d'actuariat de l'ISFA et l'admission à l'Institut des Actuaires

Par : Alexandre Bonicel

Titre : Adapter la modélisation de la sévérité en réassurance dans le contexte de forte inflation

Confidentialité : ☒ NON ☐ (Durée : ☐ 1 an ☐ 2 ans)
Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membres présents du jury de l'Institut des Actuaires — Signature

. . . . . .

. . . . . .

. . . . . .

Membres présents du jury de l'ISFA

. . . . . .

. . . . . .

. . . . . .

Entreprise :
Nom : QBE Europe

Signature :

Directeur de mémoire en entreprise :
Nom : Alexandre Masquelein

Signature :

Invité :
Nom :

Signature :

Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels *(après expiration de l'éventuel délai de confidentialité)*

Signature du responsable entreprise

Signature du candidat

Résumé

Lorsqu'un réassureur tarifie un traité en utilisant le modèle collectif (fréquence sévérité), il dispose d'un historique de sinistres tronquées aléatoirement à gauche. Pour éviter les biais dans le modèle, il doit seulement considérer les sinistres supérieurs à la troncature maximale. Beaucoup de sinistres sont alors abandonnés.

L'objectif de ce mémoire est de proposer un modèle pour la sévérité qui puisse utiliser tous les sinistres quelles que soient leurs troncatures. Ce modèle se divise en trois parties. D'abord, le corps de la distribution est modélisé avec une mixture de lois d'Erlang. Un algorithme existant dans le cas d'une troncature aléatoire est adapté. Ensuite, la queue de distribution est modélisée avec les lois de Pareto, Pareto tronquée et exponentielle. Les estimateurs de maximum de vraisemblance de ces trois lois sont adaptés dans le cas de troncatures aléatoires. Enfin, les probabilités d'être dans la queue ou le corps de la distribution sont calculées pour joindre les deux.

Pour finir, cette nouvelle proposition de modèle est testée sur des données réelles et les résultats sont comparés avec les modèles retenus par QBE Re sur ces mêmes données.

Mots-clés : *troncature aléatoire*, *sévérité*, *réassurance*.

Abstract

When a reinsurer prices a treaty using the collective model (frequency severity), it has a dataset of randomly left truncated claims. It can only consider the claims above the upper left truncation level to avoid bias in the model. Many claims are lost for the fitting.

The goal of this master thesis is to provide a model for the severity that can use every claim no matter what their truncation levels are. This model has three parts. First, the body distribution is modeled with an Erlang mixture. A pre-existing algorithm is adapted to the random truncation levels context. Then, the tail distribution is modeled with the Pareto, truncated Pareto, and exponential distributions. The maximum likelihood estimators of those three distributions are also adapted to the random truncation levels context. Finally, the probabilities to be either in the body or in the tail of the distribution are computed to join them.

At the end of this thesis, this new proposal is tested on real data and the results are compared with those from QBE Re for the same data.

Keywords : *random truncation*; *severity*; *reinsurance*.

Remerciements

Ce mémoire marque la fin de mes études d'actuaire. Il est le fruit de beaucoup de travail de ma part mais aussi de l'aide précieuse de plusieurs personnes que je souhaite remercier ici.

D'abord, je tiens à remercier mon tuteur en entreprise Alexandre Masquelein. Il a toujours été disponible pour m'orienter dans mon travail et répondre à mes questions. Tout au long de l'année, il m'a consacré énormément de temps et ce fut toujours un plaisir d'apprendre avec lui.
Je tiens aussi à remercier mes collègues au sein de QBE Re. Le sérieux et la bonne ambiance m'ont permis de m'investir sans difficulté dans mon travail et de passer cette année dans les meilleures conditions possibles.

Enfin, je voudrais remercier ma famille que ce soit pour la relecture, ou plus généralement le soutien au quotidien et tout au long de mes études. J'ai la chance d'être bien entouré.

Table des matières

Introduction

Lorsqu'un réassureur tarifie un contrat, il commence par déterminer combien vont lui coûter les sinistres qu'il réassure. Pour cela, il peut utiliser le modèle collectif. Cela consiste à modéliser la fréquence d'occurrence des sinistres d'une part et la sévérité (valeur) des sinistres d'autre part. En multipliant les deux, il obtient l'espérance des sinistres qu'il devra payer.

Pour faire ses modélisations, le réassureur dispose d'un historique de sinistres fourni par les cédantes qui souhaitent se faire réassurer. Cet historique est fourni au-dessus d'un certain seuil. On dit que les données sont tronquées à gauche. Pour utiliser ses sinistres dans sa modélisation, le réassureur calcule la valeur qu'ils auraient s'ils se produisaient l'année à modéliser (l'année de cotation). Ils sont indexés en fonction de leur année de survenance. Le seuil (troncature à gauche) est aussi réévalué. En fonction des années, il est plus ou moins augmenté. Les sinistres deviennent tronqués aléatoirement à gauche.

Il est possible de modéliser une distribution de probabilité pour des données tronquées à gauche seulement si la troncature est fixe et unique. Le réassureur doit prendre le plus grand seuil obtenu après indexation comme étant la troncature à gauche fixe et commune pour éviter un biais. Certains sinistres sont alors plus petits que le seuil retenu et ne sont plus utilisables. Il perd alors de l'information utilisable alors qu'il souffre déjà d'un manque de données par rapport aux assureurs. Plus l'inflation est importante et plus ce phénomène est important.

Malheureusement ce problème devient une actualité et il serait bien de trouver une méthode pour modéliser les distributions de probabilités qui puisse prendre en compte des troncatures à gauche aléatoires. C'est la tâche qui a été menée tout au long du stage et qui est présentée dans ce mémoire.

L'objectif est de modéliser une distribution de sévérité d'après des données tronquées aléatoirement à gauche en utilisant une mixture d'Erlang pour le corps de distribution (*body*) et la loi de Pareto, la loi de Pareto tronquée et la loi exponentielle pour la queue de distribution (*tail*).

Chapitre 1

Définition du problème

1.1 Le modèle fréquence sévérité

Comme en assurance, la réassurance est caractérisée par l'inversion du cycle de production. Le réassureur vend la couverture d'un risque avant de savoir combien la vente de ce service va lui coûter en réalité. Il doit donc définir le prix de ce qu'il vend avant de savoir ce que ça lui coûte. L'enjeu est de déterminer les prix des futurs sinistres couverts par le traité.

Il est fréquent que le réassureur vende des traités de réassurance en excédent de sinistres par risque. Dans ce type de traités, on définit une priorité et une portée. Pour chaque sinistre, le réassureur paie ce qui dépasse la priorité dans la limite de la portée. Autrement dit, pour chaque sinistre, le réassureur paie :

$$S = min(max(0, Sinistre - Priorite), Portee) \tag{1.1}$$

Pour définir cette valeur, le réassureur peut utiliser la méthode de Monte Carlo. Il va simuler un grand nombre de sinistres et leur appliquer cette formule pour estimer sa valeur moyenne. Cette valeur est l'espérance de la valeur d'un sinistre pour le réassureur.

La valeur d'un sinistre est appelée sévérité. Ensuite, en estimant la fréquence d'occurrence d'un sinistre et en la multipliant par la sévérité, le réassureur obtient ce qu'il espère payer pour un traité. Cette méthode s'appelle le modèle collectif ou encore modèle fréquence sévérité.

Dans le contexte de traités de réassurance en excédent de sinistres par risque, ce modèle est utilisé en faisant des simulations. Pour cela il faut une distribution de probabilité. Cela s'estime très bien avec une liste de sinistres historiques. La seule condition est que si les sinistres sont tronqués à gauche, la troncature doit être la même pour tous les sinistres. En d'autres termes, si les sinistres sont connus sachant qu'ils sont supérieurs à une valeur, cette valeur doit être la même pour tous.

1.2 L'information à disposition de l'assureur

Lorsqu'un assureur veut obtenir un contrat de réassurance, il transmet son historique de sinistres au-dessus d'un certain seuil sur plusieurs années. Les sinistres transmis sont tronqués à gauche. En général, pour des développements courts (*short tail*), l'assureur fournit l'information au-dessus de 75% de la priorité demandée. Par exemple pour un traité dans lequel le réassureur paie la partie des sinistres

qui dépasse 1 000 000 €, il transmet son historique de sinistres au-dessus de 750 000 €.

Cette liste de sinistres ne peut pas être utilisée directement pour la modélisation. Les sinistres sont issus d'années différentes et un sinistre qui a eu lieu il y a plusieurs années aura un prix plus important aujourd'hui. Il faut donc réévaluer les sinistres des années antérieures au prix qu'ils auraient s'ils se produisaient l'année à modéliser. Ils sont exprimés « comme si » ils se passaient l'année de cotation. Ce processus s'appelle processus « *as if* ».

Pour calculer la valeur d'un sinistre *as-if*, chaque année a un indice associé. On exprime la valeur d'un sinistre survenu l'année j comme s'il se passait l'année i comme cela :

$$Sinistre_{(as-if\,annee\,i)} = Sinistre_{(annee\,j)} * \frac{index_{(i)}}{index_{(j)}} \tag{1.2}$$

Cette formule concerne les développements courts. Pour des développements longs, il faut décomposer le sinistre puisqu'il est payé en plusieurs fois et sur plusieurs années.

1.3 L'indexation fait perdre de l'information, c'est accentué par l'inflation

Quand les sinistres sont indexés, les troncatures doivent l'être aussi. Certains sinistres qui sont juste en dessous de la troncature et qui ne sont pas transmis par l'assureur passeraient au-dessus de la troncature initiale après indexation. N'ayant pas cette information, il faut rehausser les troncatures.

Cela ne parait pas être un problème mais il faut que la troncature de tous les sinistres soit unique pour pouvoir estimer une loi dessus. Dans ce cas, il n'y a pas d'autres choix que de prendre la troncature la plus élevée comme étant la troncature commune. Cela veut dire que les sinistres de valeurs inférieures à cette troncature ne sont plus utilisables.

Lors de la modélisation, la troncature doit rester inférieure à la priorité. Les années pour lesquelles la troncature après indexation est plus élevée que la priorité ne sont plus utilisables et tous les sinistres sont perdus en termes d'information. Plus l'inflation est importante et plus la perte d'information va être accentuée. Cet effet est montré à travers un exemple.

Une entreprise d'assurance souhaite se faire réassurer en excédent de sinistres par risque avec une priorité de 1 000 000 € sur de la responsabilité civile en 2021. C'est-à-dire que le réassureur va payer tout ce qui dépasse 1 000 000 € pour chaque sinistre en 2021. L'assureur fournit son historique de sinistres sur 15 ans au-dessus du seuil de 750 000€. En utilisant les indices de QBE Re d'un marché de responsabilité civile, les seuils de chaque année sont indexés :

Année	Indice	Inflation	Troncature initiale	Troncature indexée 2021
2021	313,73	2%	750 000	750 000
2020	306,47	3%	750 000	767 767
2019	298,39	3%	750 000	788 557
2018	288,41	3%	750 000	815 844
2017	279,85	3%	750 000	840 799
2016	272,13	2%	750 000	864 651
2015	267,83	1%	750 000	878 533
2014	264,87	2%	750 000	888 351
2013	259,78	3%	750 000	905 757
2012	252,26	4%	750 000	932 758
2011	242,51	4%	750 000	970 259
2010	233,89	2%	750 000	1 006 018
2009	230,18	4%	750 000	1 022 233
2008	222,13	4%	750 000	1 059 278
2007	212,65	3%	750 000	1 106 501
2006	206,72	3%	750 000	1 138 243

FIGURE 1.1 – 5 années d'information sont perdues

On observe que le seuil indexé dépasse 1 000 000 € à partir de l'année 2010. Cela signifie qu'on ne peut pas utiliser les années de 2006 à 2010 pour tarifer ce contrat. Sur quinze années, seules onze sont exploitables.

Si maintenant on reprend le même exemple mais avec un contrat prévu pour 2023, la forte inflation de 2022 et 2023 va accentuer ce phénomène.

Année	Indice	Inflation	Troncature initiale	Troncature indexée 2023
2023	367,97	8%	750 000	750 000
2022	341,34	9%	750 000	808 512
2021	313,73	2%	750 000	879 666
2020	306,47	3%	750 000	900 504
2019	298,39	3%	750 000	924 889
2018	288,41	3%	750 000	956 893
2017	279,85	3%	750 000	986 162
2016	272,13	2%	750 000	1 014 138
2015	267,83	1%	750 000	1 030 420
2014	264,87	2%	750 000	1 041 936
2013	259,78	3%	750 000	1 062 351
2012	252,26	4%	750 000	1 094 020
2011	242,51	4%	750 000	1 138 005
2010	233,89	2%	750 000	1 179 946
2009	230,18	4%	750 000	1 198 964
2008	222,13	4%	750 000	1 242 414

FIGURE 1.2 – 9 années d'information sont perdues

Cette fois-ci, dès l'année 2016 le seuil dépasse 1 000 000 €. On ne peut plus utiliser que sept années sur les quinze. L'inflation a divisé le nombre d'années utilisables par 1,57 soit une augmentation de l'incertitude d'échantillonnage de plus de 25%.

1.4 La proposition

Le problème vient du fait qu'il faut une troncature unique pour estimer une distribution de probabilité. En imaginant qu'on puisse casser cette contrainte, on pourrait utiliser toutes les informations dont on dispose peu importe que leurs troncatures dépassent le seuil de modélisation ou non. L'objectif de ce travail est donc de faire un estimateur capable de trouver une distribution de probabilités à partir de données tronquées aléatoirement.

L'estimateur va se décomposer en deux parties, le corps de la distribution (*body*) et la queue de distribution (*tail*).
Pour la modélisation de la loi du corps de la distribution, une mixture d'Erlang est utilisée. Cette distribution est dense dans l'espace des distributions positives et continues et permet ainsi de modéliser n'importe quelle distribution. Lee & Lin (2010) proposent cette distribution pour évaluer des sinistres en assurance et décrivent un algorithme permettant de l'estimer. Dans ce mémoire, la recherche s'appuie sur l'article de Verbelen et al. (2015) qui détaille comment modéliser une mixture d'Erlang en ajoutant une troncature unique. Cette méthode est adaptée dans le cas où il y a plusieurs troncatures différentes.
Ensuite, la queue de distribution est modélisée avec trois distributions possibles, choisies en fonction des données. Il y a la loi de Pareto, la loi de Pareto tronquée et la loi exponentielle. Ici aussi l'idée est d'adapter des estimateurs dans le cas de troncatures aléatoires.
Enfin, pour pouvoir joindre ces deux distributions et en obtenir une qui couvre toutes les données, il faut estimer la probabilité d'être dans l'une ou l'autre. En utilisant l'algorithme de Newton-Raphson, ces probabilités sont calculées en prenant en compte les différentes troncatures pour chaque sinistre.

Chapitre 2

Modéliser le corps de la distribution

2.1 Le modèle de marché actuel de l'entreprise peut être amélioré

2.1.1 Le modèle de marché actuel de l'entreprise

Dans ce chapitre, le corps de la distribution est modélisé sur des données tronquées aléatoirement avec une mixture d'Erlang. Ce travail a pour but d'améliorer le modèle de marché qu'utilise QBE Re jusqu'ici. Le problème des troncatures aléatoires n'est pas nouveau et le modèle de marché actuel tente de le gérer en proposant une distribution empirique calculée par tranche.

Son fonctionnement est le suivant : Il commence en sélectionnant la plus grande troncature qu'il y a dans les données et calcule une distribution empirique entre cette troncature et la troncature à droite (séparation entre le corps et la queue de distribution). Ensuite, il choisit une autre troncature plus petite et ajuste une autre distribution empirique sur les données entre ce nouveau seuil et la troncature retenue pour la distribution précédente. Il faut aussi que les troncatures individuelles des sinistres soient inférieures ou égales au seuil pour pouvoir les utiliser. Le modèle continue ainsi jusqu'à ce qu'il atteigne le plus petit niveau de troncature et obtienne une distribution de probabilité pour chaque couche. Pour schématiser, la méthode est comme suit :

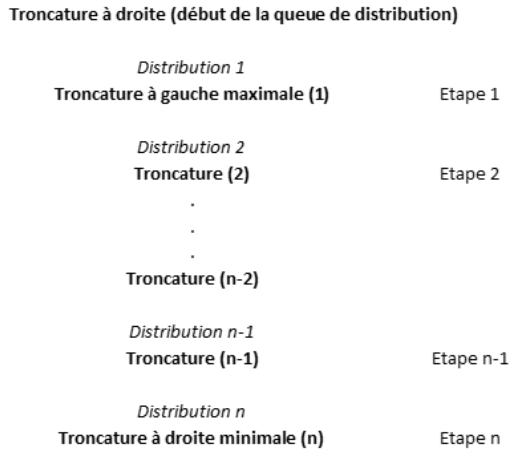

FIGURE 2.1 – Le modèle de marché actuel est découpé en tranches

Une fois que toutes les distributions sont définies, elles sont "collées" pour obtenir une seule distribution qui couvre l'ensemble des données. Ce collage est compliqué parce que les tranches ne sont pas homogènes, elles ont chacunes des cédantes différentes. Il n'est pas possible de calculer la probabilité empirique d'être dans une tranche en comptant le nombre de cédantes de la tranche.

Pour mieux comprendre cela, il est possible de prendre le cas d'une tranche entre 1 million et 2 millions pour laquelle il y a 20 sinistres. Par ailleurs, il y a 100 sinistres au total dans les données. Intuitivement, la probabilité d'être dans cette tranche est 0,2 (20/100). Cependant si une des cédantes communique seulement ses sinistres au-dessus de 2 millions mais a 5 sinistres entre 1 et 2 millions, il va manquer cette information. La probabilité d'être dans cette tranche est alors fausse. Cet exemple montre bien qu'il faut trouver une autre méthode.

Le modèle qu'utilise QBE Re considère que la tranche supérieure ne manque pas d'information et va redimensionner les distributions des tranches inférieures en fonction d'elle.

2.1.2 Un modèle de marché avec des faiblesses

Avec le modèle de marché actuel, le choix des niveaux de troncature pour définir les tranches a un impact sur les distributions et donc sur la cotation finale. Comme ce choix est fait par le technicien qui calcule le modèle, cela peut engendrer une différence de prix importante en fonction de qui effectue ce travail. L'un des premiers objectifs est de se débarrasser de ce découpage en tranches et du processus de collage en estimant directement une seule distribution pour l'ensemble des données.

Pour cela, Masquelein (2022*a*) suggère d'utiliser une loi paramétrique et la loi mixture d'Erlang est un très bon candidat. Ce choix vient de sa propriété de densité dans l'espace des lois de probabilités à valeurs positives. Cela permet d'exprimer n'importe quelles distributions de coûts de sinistres.

Dans ce chapitre, il va être montré comment un estimateur de mixture d'Erlang avec une unique troncature à gauche a été adapté dans le cas d'une troncature à gauche aléatoire. Ce travail s'appuie sur l'article de Verbelen et al. (2015) qui décrit une méthode pour modéliser une mixture d'Erlang sur des données avec une troncature fixe à gauche.

2.2 Un nouvel estimateur à partir de la loi mixture d'Erlang

Pour la modélisation, une mixture d'Erlang avec un paramètre d'échelle commun est utilisée. Une loi d'Erlang est positive, continue et définie comme :

$$f(x;r,\theta) := \frac{x^{r-1}e^{-x/\theta}}{\theta^r(r-1)!} \quad \text{pour } x>0, \tag{2.1}$$

où r, un entier positif, est le paramètre de forme et $\theta > 0$ est le paramètre d'échelle.

La mixture d'Erlang est une somme pondérée de lois d'Erlang. La densité d'une mixture de M lois d'Erlang est définie comme :

$$f(x;\alpha,r,\theta) = \sum_{j=1}^{M} \alpha_j \frac{x^{r_j-1}e^{-x/\theta}}{\theta^{r_j}(r_j-1)!} = \sum_{j=1}^{M} \alpha_j f(x;r_j,\theta) \quad \text{pour } x>0, \tag{2.2}$$

où θ est le paramètre d'échelle commun, $r=(r_1,...,r_M)$ avec $r_1 < ... < r_M$ sont les paramètres de forme de chaque loi d'Erlang et $\alpha=(\alpha_1,...,\alpha_M)$ avec $\alpha_j > 0$ et $\sum_{j=1}^{M}\alpha_j = 1$ sont les probabilités (poids) d'être dans chacune des lois d'Erlang.

2.3 Un simulateur de mixture d'Erlang pour tester l'estimateur

2.3.1 Un premier simulateur

Pour développer un estimateur de mixture de lois d'Erlang, il faut être capable de le tester. Des jeux de données dont les lois sont connues sont nécessaires. La meilleure façon de les obtenir est de les simuler. Un outil permettant de simuler des échantillons de sinistres est donc créé. Il prend en entrée les paramètres de la distribution :

— M : le nombre de lois d'Erlang dans la mixture (jusqu'à 10)
— θ : le paramètre d'échelle commun
— n : le nombre de simulations ou nombre de sinistres
— α_j : la probabilité d'être dans la j^e loi d'Erlang
— r_j : le paramètre de forme de la loi d'Erlang j

Ensuite le simulateur renvoie un échantillon de n simulations suivant la loi mixture choisie.

En pratique :

Pour chaque simulation, une des lois d'Erlang de la mixture est choisie aléatoirement en fonction des probabilités α_j puis elle est simulée.

2.3.2 L'ajout d'une troncature aléatoire à chaque simulation

Pour tester le modèle, il faut aussi que les données soient aléatoirement tronquées à gauche et que cette troncature soit indépendante. La simulation d'une troncature aléatoire à gauche est alors ajoutée dans le simulateur de mixtures de lois d'Erlang. Certaines simulations sont inférieures à leurs troncatures simulées et donc enlevées de l'échantillon.

Pour ce faire, une liste de dix troncatures possibles est créée. Comme elles doivent être cohérentes avec le jeu de données à simuler, elles sont définies en fonction du paramètre d'échelle θ et d'un facteur k à définir en entrée de l'outil. Chaque troncature est définie comme :

$$Troncature_i := \theta * k * i \tag{2.3}$$

pour $i \in [1, 10]$

Ensuite, pour chaque sinistre simulé, une des dix troncatures possibles est choisie aléatoirement.

2.4 Modéliser une mixture d'Erlang sur des données avec troncature fixe

Dans cette section, il est décrit comment ajuster une loi mixture d'Erlang sur des données avec une troncature à gauche fixe. Il faut estimer le paramètre commun d'échelle θ, le nombre d'Erlang dans la mixture M, et pour chaque Erlang j, la probabilité d'y appartenir α_j et le paramètre de forme r_j. La méthode se base sur l'article de Verbelen et al. (2015). L'idée est de calculer un estimateur initial $(M^{(k)}, (\alpha_j^{(k)}, r_j^{(k)})_j, \theta^{(k)})$ puis de l'affiner avec l'algorithme Espérance-Maximisation (*Expectation-Maximisation*).

Pour rappel, l'algorithme Espérance-Maximisation est un algorithme itératif à deux étapes dont le fonctionnement est le suivant (Borman (2004)) :

— Etape Espérance : L'espérance de la vraisemblance est calculée en tenant compte des derniers paramètres estimés. $\mathbf{E}[\mathcal{L}(x,\theta)|\hat{\theta}]$.
— Etape Maximisation : Le maximum de vraisemblance des paramètres est estimé en maximisant la vraisemblance trouvée à l'étape Espérance.

2.4.1 Notations

$$F(x,\Theta) := \sum_{j=1}^{M} \alpha_j \mathbb{P}(X_{r_j,\theta} \leq x) \quad \text{pour } x > 0, \tag{2.4}$$

Avec :
— M : Le nombre d'Erlang dans la mixture.
— α_j : La probabilité d'être dans la j^e Erlang.
— $X_{r_j,\theta} \sim Gamma(\theta, r_j)$: La j^e Erlang définie par :

$$F(x; r_j, \theta) := \mathbb{P}(X_{r_j,\theta} \leq x) \tag{2.5}$$

$$f(x; r_j, \theta) := \frac{x^{r_j-1} e^{-x/\theta}}{\theta^{r_j}(r_j-1)!} \tag{2.6}$$

$$f(x; r_j, \theta, t^u, t^l) := \frac{f(x; r_j, \theta)}{F(t^u; r_j, \theta) - F(t^l; r_j, \theta)} \tag{2.7}$$

$$\beta_j := \alpha_j \frac{F(t^u; r_j, \theta) - F(t^l; r_j, \theta)}{F(t^u; \Theta) - F(t^l; \Theta)} \tag{2.8}$$

2.4.2 Algorithme

Basé sur un estimateur a priori $(M^{(k)}, (\alpha_j^{(k)}, r_j^{(k)})_j, \theta^{(k)})$:

Etape 1 - Estimation de $(\theta^{(k+1)}, (\alpha_j^{(k+1)})_j)$ avec les $(r_j^{(k)})_j$ fixés

Ces paramètres sont estimés avec l'algorithme Espérance-Maximisation (EM).

Espérance

Verbelen et al. (2015) calculent $z_{i,j}$, la probabilité que le sinistre x_i soit dans la j^e Erlang de densité $f(x; r_j, \theta)$.

$$
\begin{aligned}
z_{i,j}^{(k+1)} &= \frac{\beta_j^{(k)} f(x_i; t^l, t^u, r_j, \theta^{(k)})}{\sum_{m=1}^{M} \beta_m^{(k)} f(x_i; t^l, t^u, r_m, \theta^{(k)})} \\
&= \frac{\beta_j^{(k)} f(x_i; r_j, \theta^{(k)}) / (F(t^u; r_j, \theta^{(k)}) - F(t^l; r_j, \theta^{(k)}))}{\sum_{m=1}^{M} \beta_m^{(k)} f(x_i; r_m, \theta^{(k)}) / (F(t^u; r_m, \theta^{(k)}) - F(t^l; r_m, \theta^{(k)}))} \\
&= \frac{\alpha_j^{(k)} f(x_i; r_j, \theta^{(k)})}{\sum_{m=1}^{M} \alpha_m^{(k)} f(x_i; r_m, \theta^{(k)})}
\end{aligned}
\tag{2.9}
$$

Maximisation

Dans cette étape, ils maximisent l'espérance de la log-vraisemblance de l'ensemble des données. D'après les calculs faits et présentés dans leur article, cela revient à calculer :

$$
\beta_j^{(k+1)} = \frac{\sum_{i=1}^{M} z_{i,j}^{(k+1)}}{n} \tag{2.10}
$$

$$
T_j^{(k+1)} := \frac{(t^l)^{r_j} e^{-t^l/\theta^{(k+1)}} - (t^u)^{r_j} e^{-t^u/\theta^{(k+1)}}}{(\theta^{(k+1)})^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta^{(k+1)}) - F(t^l; r_j, \theta^{(k+1)}))} \tag{2.11}
$$

$$
\theta^{(k+1)} := \frac{\sum_i x_i - \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} T_j^{(k+1)}}{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} r_j} \tag{2.12}
$$

Il n'est pas possible d'avoir directement une valeur pour les $T_j^{(k+1)}$ et $\theta^{(k+1)}$ étant donné qu'ils sont interdépendants. Ils calculent une première valeur pour $\theta^{(k+1)}$ en utilisant les $T_j^{(k)}$ puis réitèrent les étapes Espérance et Maximisation jusqu'à ce que la log-vraisemblance ne soit pas significativement améliorée. En pratique, on réitère jusqu'à ce que la logvraisemblance se stabilise, soit tant que $abs(logvraisemblance^{(k+1)} - logvraisemblance^{(k)}) > 0,001$. La valeur absolue est utilisée car la valeur de la logvraisemblance peut osciller pendant l'optimisation. Ça évite de s'arrêter trop tôt à cause d'une logvraisemblance plus petite sur une étape.

Recalcul des probabilités

A la fin de cette première étape, les probabilités α_j sont calculées pour chaque Erlang j. D'abord Verbelen et al. (2015) calculent :

$$
\widetilde{\alpha_j} = \frac{\hat{\beta}_j}{F(t^u; r_j, \hat{\theta}) - F(t^l; r_j, \hat{\theta})} \tag{2.13}
$$

où $\hat{\beta}_j$ et $\hat{\theta}$ sont les estimations finales de l'algorithme EM. Ensuite ils normalisent les poids pour que leur somme soit égale à 1 (probabilités) :

$$
\alpha_j = \frac{\widetilde{\alpha_j}}{\sum_{m=1}^{M} \widetilde{\alpha_m}} \tag{2.14}
$$

Etape 2 - Estimation de $(\theta^{(k+1)}, (\alpha_j^{(k+1)})_j, (r_j^{(k+1)})_j)$ pour M fixé

Pour j allant de M à 1

Ils remplacent $r_j^{(k)}$ par $r_j^{(k+1)} := r_j^{(k)} + 1$. Puis ils relancent **l'étape 1**.
Ils refont ça jusqu'à un critère d'arrêt. Dans ce mémoire, le critère AICc a été choisi. La boucle s'arrête donc lorsque l'AICc n'est pas significativement amélioré. Pour rappel, le critère AICc est défini par :

$$AICc := AIC + \frac{2k(k+1)}{n-k-1} = 2k - 2ln(L) + \frac{2k(k+1)}{n-k-1} \tag{2.15}$$

avec :
- n : le nombre de sinistres dans les données
- k : le nombre de paramètres estimés dans le modèle
- L : le maximum de vraisemblance

Plus la valeur de l'AICc est petite, meilleur est le modèle. Ce critère a été retenu ici pour travailler au lieu du critère AIC parce qu'il est plus performant pour des petits jeux de données (Brewer et al. (2016)) or c'est souvent ce à quoi la réassurance est confrontée.

Pour j allant de 1 à M

Si $r_j^{(k)} + 1 > r_{j-1}^{(k)}$
Ils remplacent $r_j^{(k)}$ par $r_j^{(k+1)} := max(r_j^{(k)} - 1; 0)$. Puis il relancent **l'étape 1**.
Cela est refait jusqu'à ce que le critère AICc ne soit pas significativement amélioré avec les nouveaux paramètres.

Etape 3 - Estimation de $(M^{(k+1)}, \theta^{(k+1)}, (\alpha_j^{(k+1)})_j, (r_j^{(k+1)})_j)$

Ils suppriment l'Erlang qui a la plus petite probabilité associée α. Ils réduisent alors le nombre d'Erlang dans la mixture puis refont **l'étape 2** jusqu'à ce que le critère AICc ne soit plus significativement amélioré. Cette dernière étape permet d'éviter le surapprentissage (*over-fitting*).

L'estimateur initial (a priori)

En pratique, pour éviter de se retrouver sur un minimum local du critère AICc, Verbelen et al. (2015) suggèrent de calculer plusieurs estimateurs et de retenir celui qui a le meilleur critère AICc. Dans ce but, cinq estimateurs a priori sont initialisés grâce à un facteur d'étalement s (*spread factor*) pour calculer les paramètres de forme r_j.
Ainsi, pour s allant de 1 à 5, les estimateurs a priori sont :

$$\begin{cases} M^{(0)} = 6 \\ r_j^{(0)} = j * s & \text{pour chaque } j \\ \theta^{(0)} = max_i(x_i)/r_M^{(0)} \end{cases} \tag{2.16}$$

et

$$\alpha_j^{(0)} := \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{I}_{\{x_i > r_{j-1}^{(0)} \theta^{(0)}\}} - \mathbb{I}_{\{x_i > r_j^{(0)} \theta^{(0)}\}} \right) \tag{2.17}$$

$M^{(0)} = 6$ a été choisi. Biber (2021) a montré qu'il ne faut pas utiliser plus de lois d'Erlang dans la mixture qu'il n'y a de bosses dans la distribution à représenter. De plus, dans une précédente étude interne de QBE Re, l'algorithme de l'article de Verbelen et al. (2015) a été testé et n'a jamais atteint un nombre d'Erlang supérieur à 6.

Avec le facteur d'étalement, l'algorithme peut atteindre de grandes valeurs et ne pas être capable d'aller au bout des calculs. Pour remédier à ce problème, il a fallu programmer en base logarithme népérien.

2.5 Adaptation dans le cas d'une troncature aléatoire

Avec l'aide de la note de Masquelein (2022*a*), l'algorithme précédemment décrit est adapté dans le cas d'une troncature à gauche **aléatoire** des données. Cela veut dire que chaque sinistre x_i peut avoir sa propre troncature à gauche t_i^l.

2.5.1 Notations

$$F(x, \Theta) := \sum_{j=1}^{M} \alpha_j \mathbb{P}(X_{r_j,\theta} \leq x) \quad \text{pour } x > 0, \tag{2.18}$$

avec :

- M : le nombre d'Erlang dans la mixture
- α_j : la probabilité d'être dans la j^e Erlang
- $X_{r_j,\theta} \sim Gamma(\theta, r_j)$: la j^e Erlang définie par

$$F(x; r_j, \theta) := \mathbb{P}(X_{r_j,\theta} \leq x) \tag{2.19}$$

$$f(x; r_j, \theta) := \frac{x^{r_j-1} e^{-x/\theta}}{\theta^{r_j}(r_j - 1)!} \tag{2.20}$$

$$f(x; r_j, \theta, t^u, t_i^l) := \frac{f(x; r_j, \theta)}{F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)} \tag{2.21}$$

$$\beta_{i,j} := \alpha_j \frac{F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)}{F(t^u; \Theta) - F(t_i^l; \Theta)} \tag{2.22}$$

2.5.2 Algorithme

L'estimateur de $(M, (\alpha_i, r_i)_i, \theta)$ est toujours basé sur un processus itératif exécuté à partir d'un estimateur initial $(M^{(k)}, (\alpha_j^{(k)}, r_j^{(k)})_j, \theta^{(k)})$.

Etape 1 - Estimation de $(\theta^{(k+1)}, (\alpha_j^{(k+1)})_j)$ avec les $(r_j^{(k)})_j$ fixés

Comme dans l'article de Verbelen et al. (2015), l'algorithme Espérance-Maximisation est utilisé pour trouver ces paramètres.

Espérance

Estimation de $z_{i,j}$, la probabilité que le sinistre x_i soit dans la j^e Erlang :

Avec la troncature aléatoire, l'espérance du maximum de log-vraisemblance devient :

$$\begin{aligned} Q(\Theta^{(k+1)}|\Theta^{(k)}) := &\sum_i \sum_{j=1} M z_{i,j}^{(k+1)} (ln(\beta_{i,j} + (r_j - 1) ln(x_i) - \frac{x_i}{\theta^{(k)}} - r_j ln(\theta^{(k)}) \\ &- ln((r_j - 1)!) - ln(F(t^u; r_j, \theta^{(k)}) - F(t_i^u; r_j, \theta^{(k)})) \end{aligned} \tag{2.23}$$

Et l'estimateur de $z_{i,j}^{(k+1)}$ reste inchangé :

$$\begin{aligned} z_{i,j}^{(k+1)} &= \frac{\beta_{i,j}^{(k)} f(x_i; t_i^l, t^u, r_j, \theta^{(k)})}{\sum_{m=1}^M \beta_{i,m}^{(k)} f(x_i; t_i^l, t^u, r_m, \theta^{(k)})} \\ &\overset{(2.21)}{=} \frac{\beta_{i,j}^{(k)} f(x_i; r_j, \theta^{(k)}) / (F(t^u; r_j, \theta^{(k)}) - F(t_i^l; r_j, \theta^{(k)}))}{\sum_{m=1}^M \beta_{i,m}^{(k)} f(x_i; r_m, \theta^{(k)}) / (F(t^u; r_m, \theta^{(k)}) - F(t_i^l; r_m, \theta^{(k)}))} \\ &\overset{(2.22)}{=} \frac{\alpha_j^{(k)} f(x_i; r_j, \theta^{(k)})}{\sum_{m=1}^M \alpha_m^{(k)} f(x_i; r_m, \theta^{(k)})} * \frac{F(t^u; \Theta^{(k)}) - F(t_i^l; \Theta^{(k)})}{F(t^u; \Theta^{(k)}) - F(t_i^l; \Theta^{(k)})} \\ &= \frac{\alpha_j^{(k)} f(x_i; r_j, \theta^{(k)})}{\sum_{m=1}^M \alpha_m^{(k)} f(x_i; r_m, \theta^{(k)})} \end{aligned} \tag{2.24}$$

Maximisation

Estimation de $\beta_{i,j}$

Ici, il faut trouver un estimateur de $\beta_{i,j}$. Malheureusement, ce n'est pas possible car $\beta_{i,j}$ dépend de la troncature t_i^l propre au sinistre x_i. La partie à optimiser est alors :

$$\sum_i \sum_{j=1}^M z_{i,j}^{k+1} (ln(\beta_{i,j}))$$

avec

$$\beta_{i,M} := 1 - \sum_{j=1}^{M-1} \beta_{i,j}$$

sous la contrainte $\beta_{i,j} = \alpha_j \frac{F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)}{F(t^u; \Theta) - F(t_i^l; \Theta)}$

soit

$$\frac{\partial \sum_i \sum_{j=1}^M z_{i,j}^{(k+1)} (ln(\beta_{i,j}))}{\partial \beta_{i,j}} = \frac{z_{i,j}^{(k+1)}}{\beta_{i,j}} - \frac{z_{i,M}^{(k+1)}}{\beta_{i,M}}$$

$$\beta_{i,j} = \frac{z_{i,j}^{(k+1)}}{z_{i,M}^{(k+1)}} \beta_{i,M}$$

et finalement

$$\beta_{i,j} = z_{i,j}$$

Il n'y a pas de raison d'obtenir la relation souhaitée :

$$\beta_{i,j} := \alpha_j \frac{F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)}{F(t^u; \Theta) - F(t_i^l; \Theta)} \tag{2.25}$$

L'approche sera donc différente. Dans l'étape de maximisation, il faut essayer de calculer $\hat{\alpha_j}$ sans passer par $\beta_{i,j}$. Seuls θ et $T_{i,j}$ seront calculés.

Estimation of θ

$$\frac{\partial Q(\Theta|\Theta^{(k)})}{\partial \theta} = \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} \left(\frac{x_i}{\theta^2} - \frac{r_j}{\theta} - \frac{\frac{\partial}{\partial\theta} F(t^u; r_j, \theta) - \frac{\partial}{\partial\theta} F(t_i^l; r_j, \theta)}{F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)} \right)$$

$$= \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} \left(\frac{x_i}{\theta^2} - \frac{r_j}{\theta} - \frac{1}{\theta^2} \frac{(t_i^l)^{r_j} e^{-t_i^l/\theta} - (t^u)^{r_j} e^{-t^u/\theta}}{\theta^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta))} \right)$$

Donc

$$\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} \left(\frac{x_i}{\theta^2} - \frac{r_j}{\theta} - \frac{1}{\theta^2} \frac{(t_i^l)^{r_j} e^{-t_i^l/\theta} - (t^u)^{r_j} e^{-t^u/\theta}}{\theta^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta))} \right) = 0$$

Ce qui donne :

$$\begin{aligned} \theta^{(k+1)} &= \frac{\sum_i x_i - \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} \left(\frac{(t_i^l)^{r_j} e^{-t_i^l/\theta} - (t^u)^{r_j} e^{-t^u/\theta}}{\theta^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta))} \right)}{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} r_j} \\ &= \frac{\sum_i x_i - \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} T_{i,j}^{(k+1)}}{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} r_j} \end{aligned} \tag{2.26}$$

où :

$$T_{i,j}^{(k+1)} := \frac{(t_i^l)^{r_j} e^{-t_i^l/\theta^{(k+1)}} - (t^u)^{r_j} e^{-t^u/\theta^{(k+1)}}}{(\theta^{(k+1)})^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta^{(k+1)}) - F(t_i^l; r_j, \theta^{(k+1)}))} \tag{2.27}$$

Comme dans le processus original de Verbelen et al. (2015), $\theta^{(k+1)}$ et $T_{i,j}^{(k+1)}$ sont interdépendants. $T_{i,j}^{(k)}$ est utilisé au lieu de $T_{i,j}^{(k+1)}$ dans la formule de $\theta^{(k+1)}$. Ensuite, $\theta^{(k+1)}$ est obtenu en réitérant les étapes Espérance et Maximisation jusqu'à ce que la log-vraisemblance ne soit plus significativement améliorée. La même condition est gardée et le processus est réitéré tant que la différence entre deux log-vraisemblances successives est supérieure à 0,001 en valeur absolue.

Recalcul des probabilités

Pour conclure cette première étape, il faut calculer α_j pour tout j. Comme il n'a pas été possible de calculer $\beta_{i,j}$, il faut essayer de calculer directement $\hat{\alpha_j}$.

En définissant la variable aléatoire suivante :

$$Z_{i,j} = \begin{cases} 1 & \text{si l'observation } x_i \text{ appartient à la } j^e \text{ Erlang de densité } f(x; r_j, \theta) \\ 0 & \text{sinon} \end{cases}$$

Il est possible d'écrire :

$$z_{i,j}^{(k+1)} = \mathbb{P}(Z_{i,j} = 1 | x_i, t_i^l, t^u; \Theta^{(k)})$$

Dans le contexte d'une troncature fixe $t_i^l = t^l$ pour tout i :

$$\begin{aligned}
\hat{\alpha_j} &= \frac{\beta_j}{F(t^u; r_j, \theta) - F(t^l; r_j, \theta)} \\
&= \sum_i (\frac{z_{i,j}}{n}) * \frac{1}{F(t^u; r_j, \theta) - F(t^l; r_j, \theta)} \\
&= \sum_i (\frac{1}{n}) * z_{i,j} * \frac{1}{\mathbb{P}(x \in [t^l, t^u] | Z_j = 1)} \\
&= \frac{1}{\mathbb{P}(x \in [t^l, t^u] | Z_j = 1)} \sum_i z_{i,j} * \mathbb{P}(x_i \in [t^l, t^u], X = x_i) \\
&= \frac{\sum_i \mathbb{P}(Z_{i,j} = 1 | x_i \in [t^l, t^u], X = x_i) * \mathbb{P}(x_i \in [t^l, t^u], X = x_i)}{\mathbb{P}(x \in [t^l, t^u] | Z_j = 1)} \\
&= \frac{\mathbb{E}(\mathbb{P}(Z_{i,j} = 1, x_i \in [t^l, t^u], X = x_i))_{x_i}}{\mathbb{P}(x \in [t^l, t^u] | Z_j = 1)} \\
&= \mathbb{E}\left(\frac{\mathbb{P}(Z_{i,j} = 1, x_i \in [t^l, t^u], X = x_i)}{\mathbb{P}(x \in [t^l, t^u] | Z_j = 1)} \right)_{x_i}
\end{aligned} \tag{2.28}$$

Dans le cas où l'intervalle de troncature $I^T = [t^l, t^u]$ n'est pas fixe mais aléatoire :

$$\begin{aligned}
\hat{\alpha_j} &= \mathbb{E}\left(\frac{\mathbb{P}(Z_{i,j} = 1, x_i \in I^T, x_i}{\mathbb{P}(x \in I^T | Z_j = 1)} \right)_{(x_i, I^T)} \\
&= \sum_i \mathbb{P}(I^T = [t_i^l, t^u], X = x_i) * \frac{\mathbb{P}(Z_{i,j} = 1 | I^T = [t_i^l, t^u], X = x_i)}{\mathbb{P}(x \in [t_i^l, t^u] | Z_j = 1)} \\
&= \sum_i \frac{1}{n} * \frac{z_{i,j}}{F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)}
\end{aligned} \tag{2.29}$$

Et finalement :

$$\alpha_j = \frac{\hat{\alpha_j}}{\sum_{m=1}^M \hat{\alpha_m}} \tag{2.30}$$

Etape 2 - Estimation de $(\theta^{(k+1)}, (\alpha_j^{(k+1)})_j, (r_j^{(k+1)})_j)$ pour M fixé

Cette étape demeure inchangée par rapport au processus initial de Verbelen et al. (2015).

Ainsi pour j allant de M à 1

$r_j^{(k)}$ est remplacé par $r_j^{(k+1)} := r_j^{(k)} + 1$. Puis **l'étape 1** est relancée.
Cela est refait jusqu'à ce que le critère AICc ne soit plus significativement amélioré avec les nouveaux paramètres estimés.

Pour j allant de 1 à M

Si $r_j^{(k)} + 1 > r_{j-1}^{(k)}$
$r_j^{(k)}$ est remplacé par $r_j^{(k+1)} := max(r_j^{(k)} - 1; 0)$. Puis **l'étape 1** est refaite.
Toujours de la même manière, cette étape est réitérée jusqu'à ce que le critère AICc ne soit plus significativement amélioré avec les nouveaux paramètres estimés.

Etape 3 - Estimation de $(M^{(k+1)}, \theta^{(k+1)}, (\alpha_j^{(k+1)})_j, (r_j^{(k+1)})_j)$

Cette étape reste aussi inchangée par rapport au cas d'une troncature fixe. Pour limiter le surapprentissage, l'Erlang qui a la plus petite probabilité associée α est enlevée puis **l'étape 2** est réitérée jusqu'à ce que le critère AICc ne soit plus significativement amélioré.

L'estimateur initial (a priori)

L'estimateur initial de M, $(r_j)_j$ et θ est le même que dans le cas d'une troncature fixe. Un facteur d'étalement s est aussi utilisé pour éviter d'être coincé sur un minimum local du critère AICc et cinq estimateurs a priori sont donc obtenus.

Ils sont définis pour tout entier $s \in [1, 5]$ comme :

$$\begin{cases} M^{(0)} = 6 \\ r_j^{(0)} = j * s & \text{pour tout } j \\ \theta^{(0)} = max_i(x_i)/r_M^{(0)} \end{cases}$$

Comme dans le cas d'une troncature fixe, $M^{(0)} = 6$ est choisi.

En revanche, les estimateurs a priori des $(\alpha_j)_j$ sont modifiés. Les données sont tronquées à gauche donc moins de sinistres sont observés parmi les petits sinistres qu'en réalité. Pour prendre ça en compte dans le modèle, il faut allouer plus de poids aux sinistres de valeurs plus faibles. Cette idée doit être traduite dans le calcul des estimateurs initiaux pour $(\alpha_j^{(0)})_j$.

Il faut commencer par calculer ce poids réajusté pour chaque observation de l'échantillon en s'inspirant de la méthode pour calculer la distribution empirique du modèle de marché actuel de QBE Re. Les poids des sinistres sont recalculés en fonction d'intervalles de troncatures et en commençant par le plus élevé :

Soient $t^1 < ... < t^T < t^u$ les T troncatures à gauche aléatoires et t^u la troncature à droite (pour borner l'échantillon).

D'abord les poids w^T des sinistres de valeurs comprises entre t^T et t^u sont évalués :

$$w^T = \frac{1}{\#\{x_i; x_i \in [t^T, t^u]\}}$$

J $\sum w^T = 1$

Ensuite, les poids w^{T-1} sont calculés pour tous les sinistres $x_i \in [t^{T-1}, t^T]$:

$$w^{T-1} = \frac{1}{\#\{x_i; x_i \geq t^{T-1}, t_i^l \leq t^{T-1}\}}$$

Et les poids w^T doivent être redimensionnés pour les $x_i \geq t^T$:

$$w^T_{redimensionne} = \frac{\#\{x_i; x_i \geq t^T, t_i^l \leq t^{T-1}\}}{\#\{x_i; x_i \geq t^{T-1}, t_i^l \leq t^{T-1}\}} * w^T$$

De cette façon, $\sum w^{T-1} + \sum w^T_{redimensionne} = 1$ est toujours vraie.

En suivant le même modèle, tous les poids sont calculés pour les intervalles inférieurs et ceux déjà calculés sont redimensionnés. Une fois que les poids $(w_i)_i$ de chaque observation x_i sont définis, les probabilités α_j sont calculées :

$$\alpha_j^{(0)} := \sum_{i=1}^{n} w_i * (\mathbb{I}_{\{x_i > r_{j-1}^{(0)}\theta^{(0)}\}} - \mathbb{I}_{\{x_i > r_j^{(0)}\theta^{(0)}\}}) \tag{2.31}$$

Cette méthode pour calculer les estimateurs des probabilités a priori n'améliore pas le résultat final (du point de vue de l'AICc) mais augmente la vitesse de convergence de l'algorithme.

2.5.3 Quelques corrections pour l'algorithme

Lorsque l'algorithme prévu pour des données tronquées aléatoirement est exécuté, il trouve les bonnes bosses dans la distribution mais reste loin de la distribution à trouver. Voici un exemple de distribution obtenue par l'estimateur, comparée à la distribution obtenue par le modèle de marché de QBE Re pour le même jeu de données (180 points).

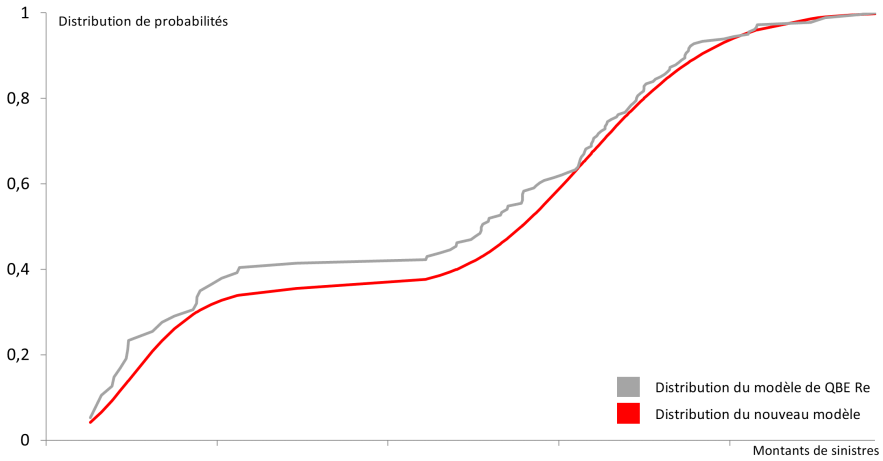

FIGURE 2.2 – Le nouvel estimateur trouve les bonnes bosses dans la distribution mais pas les bons poids

Cela montre que les estimateurs de M, $(r_j)_j$ et θ semblent corrects mais qu'il faut modifier les probabilités $(\alpha_j)_j$.

Optimisation des $(\alpha_j)_j$

Une fois que les estimateurs finaux pour M, $(r_j)_j$ et θ ont été obtenus, ils sont gardés pour faire une nouvelle estimation des $(\alpha_j)_j$.

A partir d'une liste $(\alpha_j)_j$, le critère AICc converge en bouclant sur $z_{i,j}$ (étape Espérance) et $(\alpha_j)_j$ et la distribution estimée est largement améliorée. Voici la nouvelle distribution obtenue avec le même exemple après l'optimisation des α.

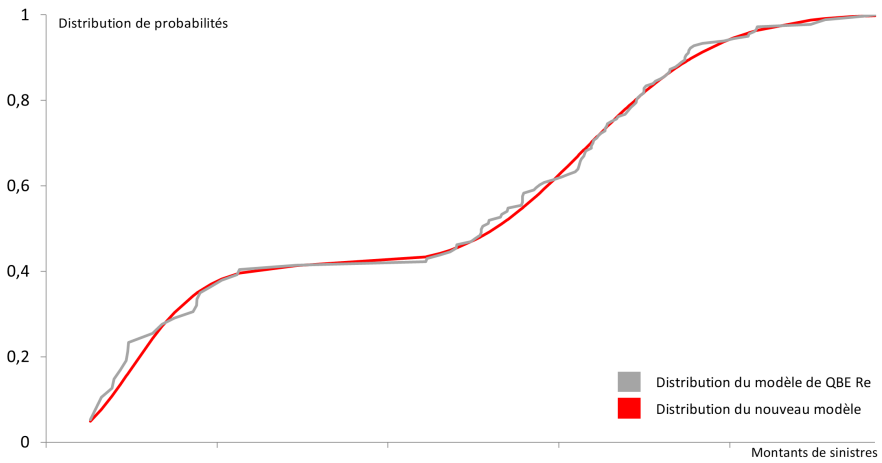

FIGURE 2.3 – L'optimisation des α permet de réajuster la distribution

Pour éviter de trouver un minimum local du critère AICc, dix listes de probabilités α sont créées aléatoirement et dix optimisations sont exécutées indépendamment. A la fin, l'optimisation avec le meilleur AICc est sélectionnée. Si ce meilleur AICc est aussi meilleur que celui du modèle avant l'optimisation des α, ces nouveaux α_j estimés sont retenus.

2.6 Amélioration de l'estimateur

2.6.1 L'algorithme de Newton-Raphson pour le calcul de θ

Dans l'étape de maximisation, il faut calculer $T_{i,j}^{(k+1)}$ et $\theta^{(k+1)}$ mais ils sont interdépendants. Pour estimer la valeur de θ, $\theta^{(k+1)}$ est calculé en utilisant $T_{i,j}^{(k)}$ et en bouclant jusqu'à ce que le résultat se stabilise. L'article de Verbelen et al. (2015) suggère l'utilisation de l'algorithme de Newton-Raphson pour obtenir la valeur de θ. Dans cette section, une version de l'algorithme est adaptée pour le modèle. Pour rappel, il faut résoudre pour θ :

$$T_{i,j}^{(k+1)} := \frac{(t_i^l)^{r_j} e^{-t_i^l/\theta^{(k+1)}} - (t^u)^{r_j} e^{-t^u/\theta^{(k+1)}}}{(\theta^{(k+1)})^{r_j-1}(r_j-1)!(F(t^u;r_j,\theta^{(k+1)}) - F(t_i^l;r_j,\theta^{(k+1)}))}$$

$$\theta^{(k+1)} := \frac{\sum_i x_i - \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} T_{i,j}^{(k+1)}}{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} r_j}$$

L'algorithme de Newton-Raphson va annuler la fonction f définie comme :

$$f(\theta) = \theta - \frac{\sum_i x_i - \sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} T_{i,j}^{(k+1)}}{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} r_j} \tag{2.32}$$

dont la dérivée est :

$$\frac{\partial f(\theta)}{\partial \theta} = 1 + \frac{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} \frac{\partial T_{i,j}^{(k+1)}}{\partial \theta}}{\sum_i \sum_{j=1}^{M} z_{i,j}^{(k+1)} r_j}$$

Il faut alors calculer :

$$\frac{\partial T_{i,j}^{(k+1)}}{\partial \theta} = \frac{\partial}{\partial \theta} \left(\frac{(t_i^l)^{r_j} e^{-t_i^l/\theta^{(k+1)}} - (t^u)^{r_j} e^{-t^u/\theta^{(k+1)}}}{(\theta^{(k+1)})^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta^{(k+1)}) - F(t_i^l; r_j, \theta^{(k+1)}))} \right)$$

En définissant :

$$N_{i,j}(\theta) = (t_i^l)^{r_j} e^{-t_i^l/\theta} - (t^u)^{r_j} e^{-t^u/\theta} \tag{2.33}$$

$$D_{i,j}(\theta) = \theta^{r_j - 1}(r_j - 1)!(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)) \tag{2.34}$$

dont les dérivées (calculées en annexe A) sont :

$$\frac{\partial N_{i,j}(\theta)}{\partial \theta} = \frac{1}{\theta^2} \left((t_i^l)^{r_j+1} e^{-t_i^l/\theta} - (t^u)^{r_j+1} e^{-t_i^l/\theta} \right) \tag{2.35}$$

$$\frac{\partial D_{i,j}(\theta)}{\partial \theta} = \theta^{r_j - 2}(r_j - 1)! \left(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta) + r_j \left(F(t^u; r_j + 1, \theta) - F(t_i^l; r_j + 1, \theta) \right) \right) \tag{2.36}$$

On obtient :

$$\frac{\partial T_{i,j}}{\partial \theta} = \frac{D_{i,j}(\theta) * \frac{\partial N_{i,j}(\theta)}{\partial \theta} - N_{i,j}(\theta) * \frac{\partial D_{i,j}(\theta)}{\partial \theta}}{(D_{i,j}(\theta))^2}$$

$$T_{i,j} = \frac{N_{i,j}(\theta)}{D_{i,j}(\theta)}$$

Pour réaliser l'algorithme de Newton-Raphson, il suffit ensuite de boucler sur les résultats suivants :

$$\frac{\partial f(\theta)}{\partial \theta} = 1 + \frac{\sum_i \sum_{j=1}^{M} z_{i,j} \frac{\partial T_{i,j}}{\partial \theta}}{\sum_i \sum_{j=1}^{M} z_{i,j} r_j}$$

$$f(\theta) = \theta - \frac{\sum_i - \sum_i \sum_{j=1}^{M} z_{i,j} T_{i,j}}{\sum_i \sum_{j=1}^{M} z_{i,j} r_j}$$

$$\theta^{(k+1)} = \theta - \frac{f(\theta)}{\frac{\partial f(\theta)}{\partial \theta}}$$

Ces résultats sont corrects mais inutilisables tels quels. Ils comprennent des factoriels et des puissances avec les paramètres de forme r_j qui peuvent atteindre de très grandes valeurs. Il faut passer en base logarithmique. Une première idée était de calculer les logarithmes de $D_{i,j}(\theta)$, $N_{i,j}(\theta)$, et de leurs dérivées mais ils sont définis avec des sommes et les grandes valeurs ne peuvent pas être évitées. Les $T_{i,j}(\theta)$ et leurs dérivées doivent être complètement réécrits. Voici le résultat (calculs dans l'annexe A) :

$$\begin{aligned} T_{i,j}(\theta) = & \, exp\left(r_j log(t_i^l) - \frac{t_i^l}{\theta} - (r_j - 1)log(\theta) - \sum_{k=1}^{r_j - 1} log(k) - log(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)) \right) \\ & - exp\left(r_j log(t^u) - \frac{t^u}{\theta} - (r_j - 1)log(\theta) - \sum_{k=1}^{r_j - 1} log(k) - log(F(t^u; r_j, \theta) - F(t_i^l; r_j, \theta)) \right) \end{aligned} \tag{2.37}$$

$$\frac{\partial T_{i,j}}{\partial \theta} = A_{i,j}(\theta) - B_{i,j}(\theta) \tag{2.38}$$

avec

$$\begin{aligned} A_{i,j}(\theta) = & \, exp\left((r_j + 1)log(t_i^l) - \frac{t_i^l}{\theta} - 2log(\theta) - log(D_{i,j}(\theta)) \right) \\ & - exp\left((r_j + 1)log(t^u) - \frac{t^u}{\theta} - 2log(\theta) - log(D_{i,j}(\theta)) \right) \end{aligned} \tag{2.39}$$

$$\begin{aligned} B_{i,j}(\theta) = & \, exp\left(log\left(\frac{\partial D_{i,j}(\theta)}{\partial \theta} \right) + r_j log(t_i^l) - \frac{t_i^l}{\theta} - 2log(D_{i,j}(\theta)) \right) \\ & - exp\left(log\left(\frac{\partial D_{i,j}(\theta)}{\partial \theta} \right) + r_j log(t^u) - \frac{t^u}{\theta} - 2log(D_{i,j}(\theta)) \right) \end{aligned} \tag{2.40}$$

Avec ces deux nouvelles expressions pour $T_{i,j}(\theta)$ et sa dérivée, il est possible d'obtenir les valeurs de $f(\theta)$ et $\frac{\partial f(\theta)}{\partial \theta}$.

L'algoritme de Newton-Raphson n'est pas à l'abri de tomber sur un extremum local. Pour éviter cette situation, on commence par boucler sur $T_{i,j}$ et θ jusqu'à être proche de la bonne valeur puis l'algorithme de Newton-Raphson peut améliorer la précision de θ. Malheureusement, les essais ne se sont pas passés comme prévu. L'algorithme ne fonctionne bien que dans le cas d'une mixture d'une seule Erlang. En réalité, le problème vient des $z_{i,j}$ dans les expressions de $f(\theta)$ et $\frac{\partial f(\theta)}{\partial \theta}$. Ils ont été considérés constants mais ils dépendent de θ. Il n'est pas raisonnable de calculer la dérivée de la fonction f en considérant les $z_{i,j}$ comme des fonctions de θ d'un point de vue complexité. Avec seulement une seule Erlang dans la mixture, $z_{i,j} = 1$ pour n'importe quel i ou j et est bien constant. C'est pourquoi l'algorithme réussit dans ce cas.

Le problème est illustré avec quelques exemples. D'abord un échantillon de 200 valeurs est simulé - avec une troncature aléatoire à gauche pour chaque point - d'après une mixture d'une seule Erlang définie par :

$$f(x_i) = f(x_i; r = 30, \theta = 3)$$

Les résultats sont les mêmes avec ou sans l'optimisation de Newton-Raphson.

$$\hat{f}(x_i) = f(x_i; r = 29, \theta = 3,2)$$

Voici le graphique obtenu en comparant le nouvel estimateur avec l'estimateur empirique de QBE Re :

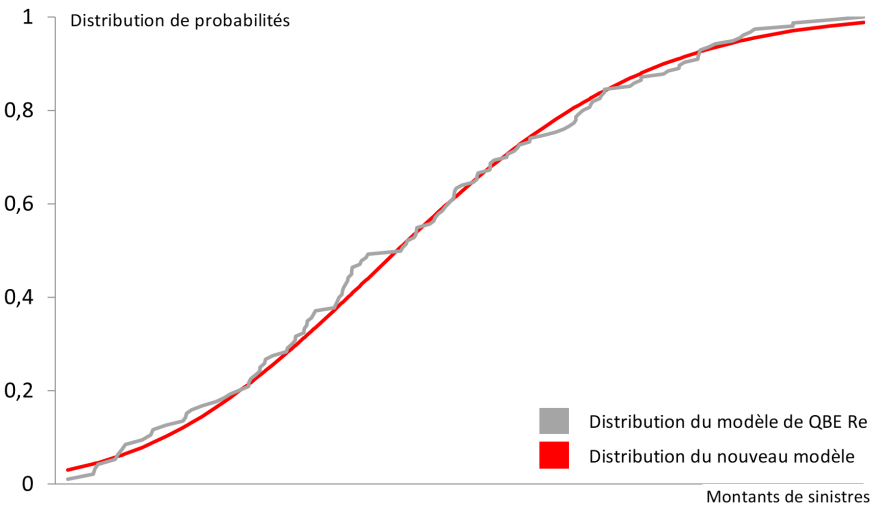

FIGURE 2.4 – Newton-Raphson fonctionne pour une Erlang dans la mixture

Dans le cas d'une seule Erlang dans la mixture, il n'y a pas de différence sur le résultat final.

Maintenant, voici ce qu'il se passe avec une mixture de deux lois d'Erlang. Un ensemble de 200 sinistres avec leurs troncatures aléatoires et indépendantes est simulé suivant la distribution :

$$f(x_i) = 0,5 * f(x_i; r = 10, \theta = 3) + 0,5 * f(x_i; r = 40, \theta = 3)$$

Sans utiliser l'algorithme de Newton-Raphson, l'estimateur donne la formule suivante :

$$\hat{f}(x_i) = 0,45 * f(x_i; r = 8, \theta = 4,12) + 0,55 * f(x_i; r = 29, \theta = 4,12)$$

Et le graphique suivant (le résultat comparé au modèle de l'entreprise) :

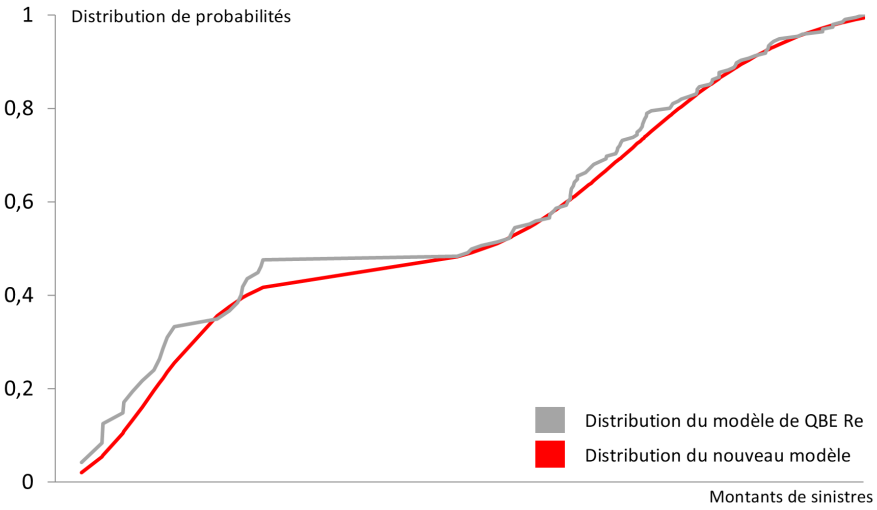

FIGURE 2.5 – Le nouvel estimateur trouve les bonnes bosses de la distribution

Les paramètres estimés ne sont pas proches des originaux mais la distribution finale est proche donc représente les données. C'est ce qui est important pour la tarification. Le graphique n'est pas parfait mais il capte les bonnes bosses dans la distribution.

La version calculée en utilisant l'algorithme de Newton-Raphson est très différente :

$$\hat{f}(x_i) = f(x_i; r = 2, \theta = 66,03)$$

Souvent il n'est pas facile de comparer des distributions avec seulement leurs formules mais on peut déjà dire que le résultat est moins bien. Il n'y a qu'une Erlang dans la distribution et donc qu'une seule bosse alors qu'on sait qu'il y en a normalement deux.

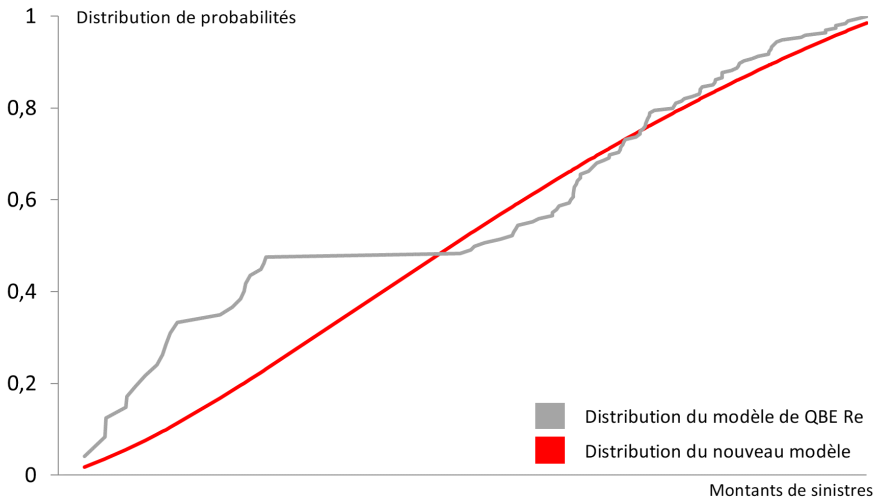

FIGURE 2.6 – L'algorithme de Newton-Raphson ne fonctionne pas pour plusieurs Erlang

Comme attendu, le nouvel estimateur avec Newton-Raphson est beaucoup plus loin de la distribution cible. Il est aussi possible de comparer les critères AICc. Pour le premier estimateur calculé sans Newton-Raphson, $AICc = 981,214$ alors que pour le second, $AICc = 1026,24$. Sachant que ce critère pénalise le nombre de paramètres, le second estimateur est d'autant plus mauvais. Dans le but de comprendre ce qui ne fonctionne pas, le graphique de la fonction à annuler dans l'algorithme de Newton-Raphson est affiché.

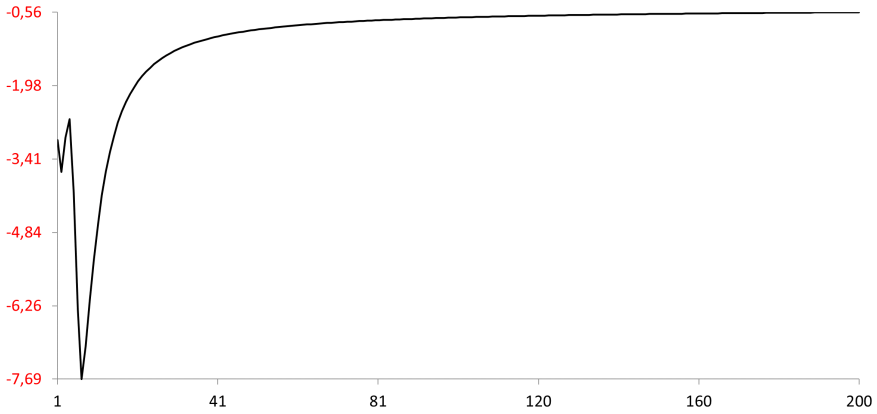

FIGURE 2.7 – La fonction ne s'annule jamais

La fonction ne s'annule jamais, l'algorithme de Newton Raphson est inutile dans ce cas. Après quelques autres tests, cette idée est abandonnée.

2.7 Test de l'estimateur

Dans cette section, le nouvel estimateur final pour le corps de la distribution est testé et comparé avec l'estimateur de QBE Re. Comme les jeux de données sont simulés, les deux estimateurs sont aussi comparés avec la distribution originale.

2.7.1 Le fonctionnement du modèle de marché de QBE Re

Le principe général de l'estimateur actuel de QBE Re a déjà été expliqué mais il va être expliqué en détail avant de faire la comparaison finale.

Cet estimateur donne une distribution empirique dont les probabilités des points sont revues en fonction de leurs troncatures aléatoires à gauche.

Soient $t^1 < ... < t^T < t^u$ les T troncatures aléatoires à gauche possibles et la troncature à droite fixe t^u.

D'abord le modèle calcule les poids w^T pour les sinistres entre t^T et t^u :

$$w^T = \frac{1}{\#\{x_i; x_i \in [t^T, t^u]\}}$$

Les poids sont tels que $\sum w^T = 1$

Ensuite les poids w^{T-1} pour les sinistres $x_i \in [t^{T-1}, t^T]$ sont calculés tels que :

$$w^{T-1} = \frac{1}{\#\{x_i; x_i \geq t^{T-1}, t_i^l \leq t^{T-1}\}}$$

Et on doit réajuster les poids w^T pour tout $x_i \geq t^T$:

$$w^T_{redimensionne} = \frac{\#\{x_i; x_i \geq t^T, t^l_i \leq t^{T-1}\}}{\#\{x_i; x_i \geq t^{T-1}, t^l_i \leq t^{T-1}\}} * w^T$$

De manière à toujours avoir $\sum w^{T-1} + \sum w^T_{redimenssionne} = 1$

En suivant le même modèle, il faut calculer les poids pour les sinistres de troncatures inférieures et toujours réajuster les poids déjà calculés pour les sinistres supérieurs.

Une fois que tous les poids sont calculés, la fonction de répartition empirique F est :

$$F(x_i) = \sum_{k \leq i} w_k \tag{2.41}$$

2.7.2 Les améliorations apportées par le nouvel estimateur

Le nouvel estimateur a certains objectifs en termes d'amélioration de l'estimateur actuel :

— Réduire le nombre de paramètres en passant d'une distribution empirique à une distribution paramétrique
— Lisser la distribution
— Etre au moins aussi précis que l'estimateur actuel
— Augmenter la robustesse du modèle

Réduction du nombre de paramètres et lissage

Comme le nouvel estimateur est paramétrique, il a moins de paramètres que pour une distribution empirique. C'est bien une amélioration de l'estimateur actuel sur ce point. Cet estimateur paramétrique permet aussi de lisser la distribution.

La précision de l'estimateur

La distribution obtenue par le nouvel estimateur, celle du modèle de QBE Re et la distribution empirique calculée à partir du jeu de données entier avant troncature sont comparées.

Avec l'outil de simulation, un échantillon de 200 sinistres suivant la loi mixture d'Erlang avec les paramètres suivants est créé :

$$\begin{cases} M = 2 \\ (\alpha_1; \alpha_2) = (\frac{1}{2}; \frac{1}{2}) \\ (r_1; r_2) = (10; 40) \\ \theta = 3 \end{cases}$$

Avec la troncature, l'échantillon est plus petit. Il perd les sinistres qui sont plus petits que leurs troncatures et ne compte plus que 98 sinistres. Le nouvel estimateur paramétrique trouve la mixture

d'Erlang suivante :

$$\begin{cases} M = 2 \\ (\alpha_1; \alpha_2) = (0,5; 0,5) \\ (r_1; r_2) = (13; 49) \\ \theta = 2,5 \end{cases}$$

Graphiquement, ça donne :

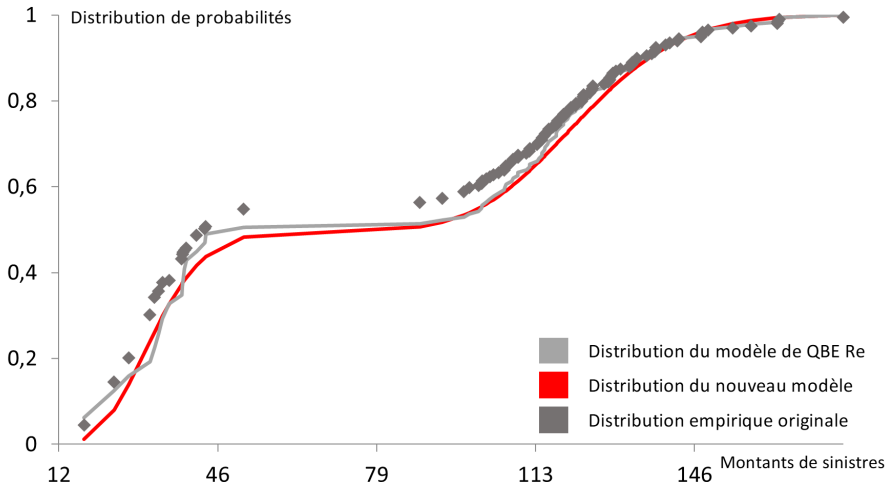

FIGURE 2.8 – Les deux estimateurs sont équivalents

Le nouveau modèle suit le modèle de marché actuel. Dans cet exemple, le nouvel estimateur est au moins aussi précis que l'estimateur empirique. Pour avoir une indication, la distribution empirique réelle du jeu de données (avant troncature) est représentée par la suite de points.

Un estimateur plus robuste

Une faiblesse importante du modèle de marché actuel est son manque de robustesse. Un petit sinistre a un poids très important dans le modèle et une valeur extrême parmi les petits sinistres peut mettre en difficulté le modèle de QBE Re.

Pour montrer cela, le même exemple que précédemment est repris, i.e. simulé à partir de :

$$\begin{cases} M = 2 \\ (\alpha_1; \alpha_2) = (\frac{1}{2}; \frac{1}{2}) \\ (r_1; r_2) = (10; 40) \\ \theta = 3 \end{cases}$$

Dans l'exemple précédent, la plus petite troncature à gauche est 12 et le jeu de données comprend 98 valeurs. Les deux valeurs suivantes sont ajoutées :

— $(x_1; l_1^l) = (11; 8)$
— $(x_2; l_2^l) = (12; 8)$

En ajoutant ces deux nouvelles valeurs avec une troncature inférieure à la plus petite troncature, elles ont un poids de 50% dans l'estimateur de QBE Re. La distribution est alors tirée vers le haut. Le

nouveau modèle donne l'estimateur suivant :

$$\begin{cases} M = 2 \\ (\alpha_1; \alpha_2) = (0,59; 0,41) \\ (r_1; r_2) = (9; 40) \\ \theta = 3,1 \end{cases}$$

Le nouveau graphique est :

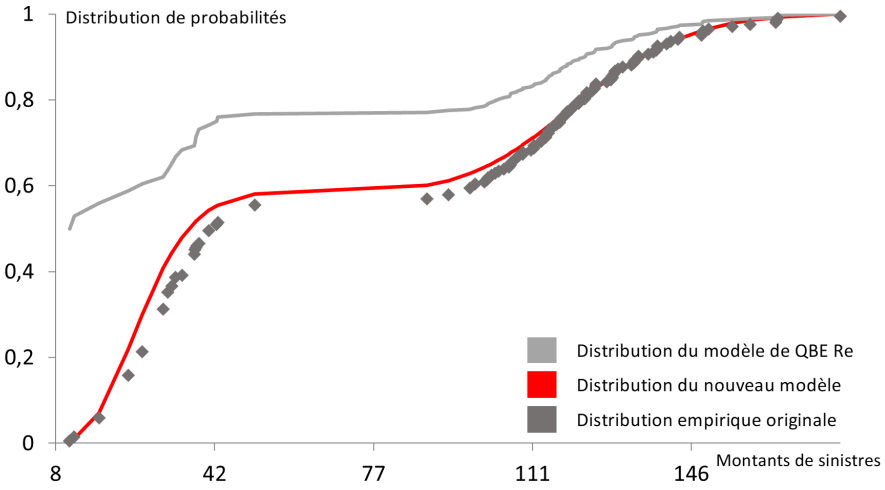

FIGURE 2.9 – L'estimateur empirique est très mauvais dans cette situation

Le modèle de marché actuel est loin de la distribution réelle. Comme prévu, les probabilités de sinistres plus faibles sont surestimées. Le nouveau modèle surestime un peu les probabilités de sinistres les plus faibles mais reste proche de la distribution réelle.

L'amélioration de la robustesse semble donc réussie.

2.8 Conclusion

Le but de ce chapitre était de modéliser le corps d'une distribution de sinistres à partir de sinistres tronqués **aléatoirement** à gauche. L'algorithme décrit par Verbelen et al. (2015) permet d'estimer une mixture de lois d'Erlang dans le cas de troncatures **fixes**. Il a été adapté dans le cas de troncatures **aléatoires** et le résultat est satisfaisant.

Dans le prochain chapitre, la troncature aléatoire à gauche est introduite dans la modélisation de la queue de distribution.

Chapitre 3

Modéliser la queue de distribution

3.1 L'estimateur de l'entreprise à développer

3.1.1 Le fonctionnement général

Ce chapitre se concentre sur la modélisation de la queue de distribution. La distribution est supposée divisée en deux au niveau d'un seuil A. La partie en dessous de ce seuil est le corps et la partie au-dessus est la queue de la distribution.

Par ailleurs, trois distributions sont utilisées en fonction de la nature des données. Le choix de se restreindre à ces trois distributions vient de Albrecher et al. (2017). Il y a :

— La distribution de Pareto
— La distribution exponentielle
— La distribution de Pareto tronquée

Il faut sélectionner la meilleure distribution et estimer ses paramètres à l'aide d'une **étude graphique**.

D'abord, l'excédent moyen de chaque point est affiché pour trouver quelle distribution utiliser. Ce graphique est couramment appelé ***Mean-Excess plot***, c'est comme ça qu'il sera appelé dans la suite du mémoire. Il représente l'excédent moyen au-dessus d'un sinistre en fonction de chaque sinistre de l'ensemble à modéliser. Sa forme suggère quelle est la distribution la plus adaptée aux données.

Ainsi, on cherche une tendance linéaire dans le *Mean-Excess plot*. Son point de départ donne la valeur du seuil A et sa direction indique la meilleure distribution à utiliser. Une tendance linéaire **plate** suggère d'utiliser une distribution **exponentielle**, une **croissante** suggère d'utiliser la distribution de **Pareto** et une **décroissante** suggère d'utiliser une distribution de **Pareto tronquée**. Voici un exemple :

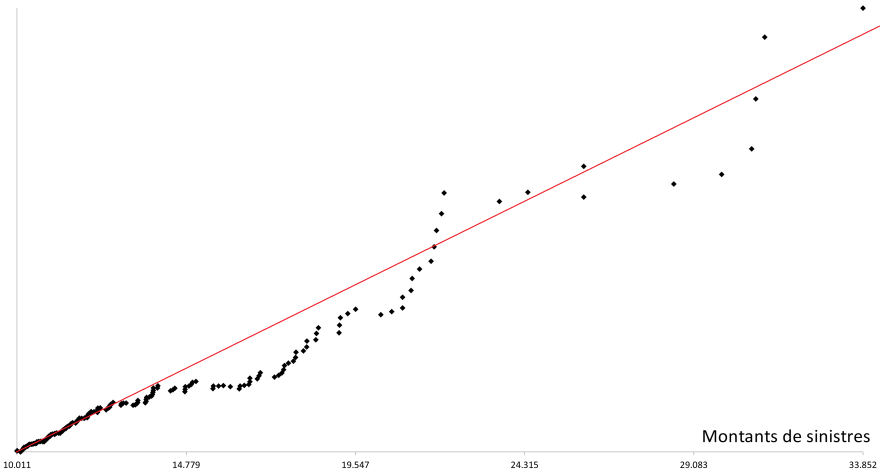

FIGURE 3.1 – Ce *ME plot* suggère une Pareto

Sur ce *Mean-Excess plot*, une tendance linéaire croissante est observée. Cela suggère qu'une loi de Pareto serait adaptée pour décrire les données. De plus cette tendance linéaire commence dès le début du graphique, on peut prendre $A = 10\ 000$. En réalité, ce *Mean-Excess plot* a été réalisé à partir d'un échantillon de 200 simulations suivant une loi de Pareto de paramètres $X_m = 10\ 000$ et $\alpha = 3$. C'est cohérent avec le graphique obtenu.

Une fois qu'une des trois distributions a été choisie et qu'un seuil A a été défini, les paramètres doivent être estimés. Pour une distribution de Pareto, il s'agit du paramètre de forme α. Un *α-plot* est utilisé. Il s'agit d'un estimateur du paramètre α. Pour chaque sinistre de l'ensemble, il estime une valeur pour α en utilisant tous les sinistres supérieurs ou égaux à ce sinistre. Une tendance plate dans ce graphique indique la valeur du paramètre α.

Voici l'exemple d'un graphique *α-plot* (avec un intervalle de confiance à 80%) obtenu à partir de l'échantillon précédent :

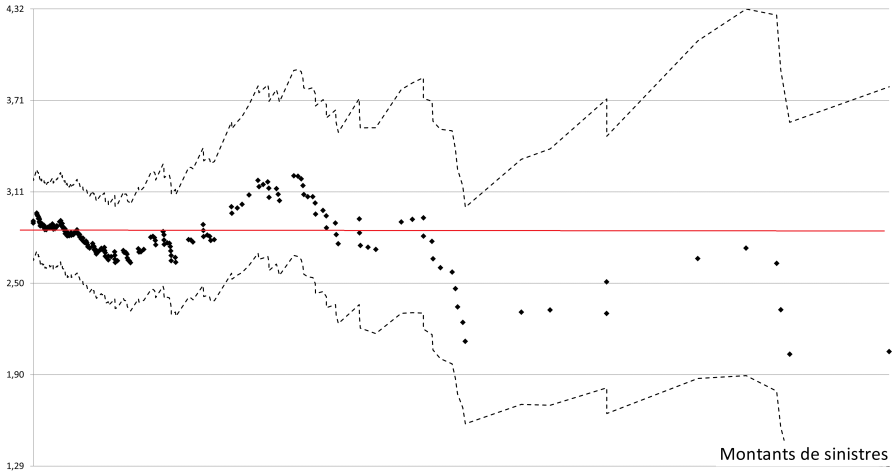

FIGURE 3.2 – L'*α-plot* suggère $\alpha = 2,9$

Cet exemple suggère de choisir $\alpha = 2,9$. La distribution originale a le paramètre $\alpha = 3$.

Dans le cas de la distribution exponentielle, le paramètre de forme λ est estimé avec le même processus que dans le cas de la distribution de Pareto mais c'est un estimateur de λ en fonction des différents seuils A qui est affiché. Le graphique est très similaire au *α-plot*.

Enfin, pour la distribution de Pareto tronquée, il faut estimer le paramètre de forme α mais aussi la troncature supérieure T. Comme ces deux paramètres sont interdépendants, on les estime à l'aide d'un algorithme de Newton-Raphson puis on obtient un *α-plot* et un T-plot que l'on interprète de la même manière que les autres distributions.

3.1.2 Les faiblesses du modèle actuel

La manière de traiter les distributions de queue décrite n'est pas parfaite. Les graphiques sont très volatils et le souscripteur doit deviner les valeurs des paramètres à partir de ces graphiques volatils. Les biais humains peuvent être énormes dans ce processus.

La volatilité provient principalement d'un manque de données. En effet le modèle de queue concerne les grosses pertes, donc plus rares que les autres. Et la réassurance dispose déjà de moins de données que l'assurance. Ceci est accentué par les troncatures inférieures importantes et différentes (franchises) qui suppriment encore plus de données.

Comme pour le corps de la distribution, la queue est modélisée en ne gardant que les sinistres au-dessus de la plus grande troncature inférieure. L'idée pour améliorer le processus est de prendre en compte les différentes troncatures dans les calculs des différents graphiques tels que *α-plot* ou *Mean-Excess plot*. Les parties suivantes décrivent comment procéder pour la distribution de Pareto, la distribution exponentielle et la distribution de Pareto tronquée. A chaque fois, les estimateurs sont présentés dans le cas d'une troncature unique puis adaptés dans le cas d'une troncature aléatoire à l'aide de la note de Masquelein (2022*b*).

3.2 La distribution de Pareto

3.2.1 Définition

La fonction de densité de probabilité de la loi de Pareto est définie comme :

$$f(x) = \alpha \frac{A^\alpha}{x^{\alpha+1}} \tag{3.1}$$

avec :

— $A > 0$, le paramètre d'échelle (troncature inférieure de la distribution)
— $\alpha > 0$, le paramètre d'échelle
— $x \in [A, \infty)$

Pour modéliser cette distribution, il faut trouver un estimateur pour α. La vraisemblance est utilisée. Pour un échantillon $x = [x_1, x_2, ..., x_n]$, $n \in \mathbb{N}$, elle est donnée par :

$$\mathcal{L}(x) = \prod_{i=1}^{n} \frac{\alpha}{A} \left(\frac{x}{A}\right)^{-(\alpha+1)} \tag{3.2}$$

Le calcul de la vraisemblance est dans l'annexe B.

3.2.2 L'estimateur de la loi de Pareto

Le cas d'un jeu de données avec une troncature à gauche unique

La vraisemblance peut être exprimée à l'aide d'une loi de probabilité Gamma.

$$\begin{aligned}\mathcal{L}(\alpha) &= \prod_{i=1}^{n} \frac{\alpha}{A}\left(\frac{x}{A}\right)^{-(\alpha+1)} \\ &\sim \alpha^n e^{-\alpha\sum_{i=1}^{n}(ln(x_i)-ln(A))} \\ &\sim Gamma\left(k=n+1, \theta=\frac{1}{\sum_{i=1}^{n}(ln(x_i)-ln(A))}\right)\end{aligned} \tag{3.3}$$

De cette manière, l'estimateur du maximum de vraisemblance est :

$$e^{\mathcal{L}}_{esperance} := \mathbb{E}_{\mathcal{L}}(\alpha) = k\theta = \frac{n+1}{\sum_{i=1}^{n}(ln(x_i)-ln(A))} \tag{3.4}$$

Pour chaque sinistre x_i dans l'échantillon, on pose $A = x_i$ et calcule l'estimateur du maximum de vraisemblance pour le groupe $\{x_j, \forall j | x_j > A = x_i\}$. En affichant la liste des estimateurs en fonction des différents sinistres, l'α-*plot* est obtenu.

L'ajout de la troncature à gauche aléatoire

Maintenant, comme dans la note de Masquelein (2022*b*), une troncature à gauche aléatoire T_i est introduite pour chaque sinistre x_i dans le jeu de données. Ainsi, la troncature à gauche pour chaque sinistre x_i devient :

$$A_i := max(A, T_i)$$

Et la vraisemblance :

$$\begin{aligned}\mathcal{L}(\alpha) &= \prod_{i=1}^{n} \frac{\alpha}{A_i}\left(\frac{x}{A_i}\right)^{-(\alpha+1)} \\ &\sim \alpha^n e^{-\alpha\sum_{i=1}^{n}(ln(x_i)-ln(A_i))} \\ &\sim Gamma\left(k=n+1, \theta=\frac{1}{\sum_{i=1}^{n}(ln(x_i)-ln(A_i))}\right)\end{aligned} \tag{3.5}$$

Le nouvel estimateur du maximum de vraisemblance :

$$e^{\mathcal{L}}_{esperance} := \mathbb{E}_{\mathcal{L}}(\alpha) = k\theta = \frac{n+1}{\sum_{i=1}^{n}(ln(x_i)-ln(A_i))} \tag{3.6}$$

L'α-*plot* est obtenu de la même manière que dans le cas de la troncature unique.

Un intervalle de confiance pour aider à la prise de décision

Dans les deux cas de troncature, un intervalle de confiance peut être calculé grâce à la loi Gamma.

En définissant c comme la probabilité d'être dans l'intervalle de confiance et $F_{\mathcal{G}}$ la fonction de répartition de la loi Gamma de paramètres k et θ :

$$Borne \quad Inférieure \quad = \quad F_{\mathcal{G}}^{-1}\left(\frac{1-c}{2}\right) \tag{3.7}$$

$$Borne \quad Supérieure \quad = \quad F_{\mathcal{G}}^{-1}\left(\frac{1+c}{2}\right) \tag{3.8}$$

3.2.3 La comparaison des α-*plot* est satisfaisante

Dans cette section, le nouvel estimateur de la loi de Pareto est évalué. Pour faire cela, un jeu de données est simulé sans troncature et son α-*plot* est construit. Ensuite, des troncatures à gauche aléatoires sont rajoutées à ce même jeu de données et un α-*plot* est construit avec le nouvel estimateur. L'objectif est que ces deux α-*plot* soient proches.

D'abord, 200 sinistres sont simulés, suivant la loi de Pareto de paramètres $x_m = 10\,000$ et $\alpha = 3$. A partir de cet échantillon, l'α-*plot* original est obtenu. Ensuite, l'échantillon est tronqué aléatoirement à gauche et un nouvel échantillon de 73 sinistres est obtenu (les 127 autres ayant disparu avec les troncatures). A partir de ce dernier, un α-*plot* est construit avec le nouvel estimateur. En superposant les deux graphiques (avec leurs intervalles de confiance à 80%), cela donne :

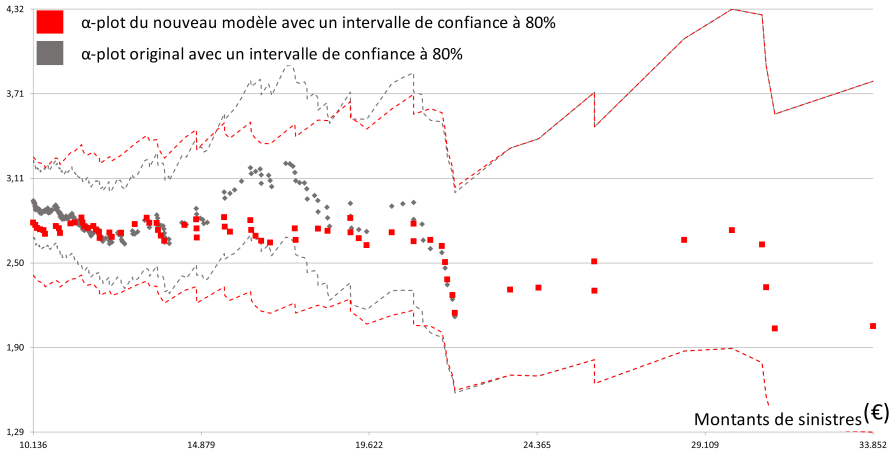

FIGURE 3.3 – Le nouvel estimateur donne une bonne approximation du graphique original

Sur ce graphique, il y a les α-*plot* et leurs intervalles de confiance à 80% pour le nouvel estimateur et l'estimateur de l'échantillon original non tronqué.

On peut noter que les deux graphiques se superposent parfaitement à partir d'environ 23 000. Cela vient du fait que la plus grande troncature basse de l'échantillon tronqué est 23 026. Au-dessus de cette valeur, les deux échantillons sont identiques.

L'objectif est d'obtenir le graphique gris. Il représente l'échantillon original. En pratique, ce n'est pas possible de l'obtenir car les données pour cela sont absentes. Seule la partie au-dessus de la plus grande troncature basse peut être calculée dans le modèle actuel.

Dans cette nouvelle proposition, le graphique du modèle actuel de l'entreprise est conservé et le graphique sous cette troncature est estimé en incluant les troncatures aléatoires dans le calcul. Les informations actuelles sont conservées et on y ajoute seulement de l'information en dessous d'un certain seuil. De plus, les résultats semblent être une bonne approximation du graphique recherché.

3.2.4 Le *Mean-Excess plot*

Sans troncature, le *Mean-Excess plot* peut être défini en fonction des seuils t comme :

$$ME(t) := \mathbb{E}(X - t|X > t) \tag{3.9}$$

Pour la liste de sinistres $[x_1, x_2, ..., x_n]$, cela donne :

$$ME(t) := \frac{\sum_{i=1}^{n}(x_i - t)\mathbb{1}_{\{x_i>t\}}}{\sum_{i=1}^{n}\mathbb{1}_{\{x_i>t\}}} \tag{3.10}$$

Dans le cas d'un échantillon avec une troncature aléatoire pour chaque sinistre, la formule ci-dessus n'est plus utilisable qu'au-dessus de la plus grande troncature basse et le *Mean-Excess plot* sous cette valeur est totalement inconnu. Il faut adapter cela pour prendre en compte les troncatures et obtenir une approximation du graphique sous la plus grande troncature basse.

La méthode vient de la note de Masquelein (2022*b*). En raison de la troncature, il y a quelques lacunes. Pour les remplir, il est possible de faire un processus *as-if* basé sur le quantile :

$$\mathbb{P}(X \leq x_i|X > A_i) = \mathbb{P}(X \leq x_i^{as-if}|X > A) \tag{3.11}$$

avec
— $A_i = max(T_i, A)$
— T_i la troncature aléatoire à gauche du sinistre x_i
— A la troncature basse commune
Le processus *as-if* dans le cadre du *Mean-Excess Plot* pour la distribution de Pareto est :

$$\begin{aligned}\mathbb{P}(X \leq x_i|X > A_i) &= 1 - \left(\frac{x_i}{A_i}\right)^{-\alpha} = 1 - \left(\frac{A\frac{x_i}{A_i}}{A}\right)^{-\alpha} \\ &= \mathbb{P}\left(X \leq A\frac{x_i}{A_i}|X > A\right)\end{aligned} \tag{3.12}$$

Soit :

$$x_i^{as-if} = A\frac{x_i}{A_i} \tag{3.13}$$

L'estimateur du *Mean-Excess plot* devient :

$$\mathbb{E}(X - A|X > A) = \frac{1}{n_A}\sum_{i=1}^{n_A}\left(A\frac{x_i}{A_i} - A\right) \tag{3.14}$$

avec n_A le nombre de sinistres au-dessus du seuil A.

Remarque

Il est important de comprendre que l'estimateur basé sur le processus as-if fait l'hypothèse que la distribution suit une loi de Pareto. Par conséquent, seule une distribution de Pareto peut être validée. Par exemple, une tendance linéaire plate sur ce graphique ne signifierait pas que la distribution est exponentielle car le processus as-if est basé sur la distribution de Pareto. Un tel graphique permettrait seulement d'invalider la loi de Pareto.

Le nouvel estimateur est maintenant comparé en utilisant les deux mêmes échantillons utilisés dans la sous-section précédente. Voici le graphique obtenu :

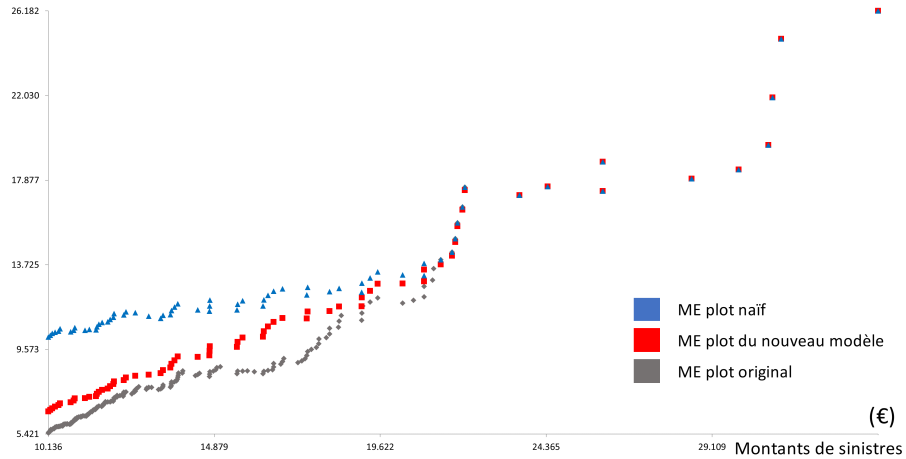

FIGURE 3.4 – L'estimateur donne une bonne approximation du *Mean-Exess plot* original

Sur ce graphique, il y a le *Mean-Excess plot* du jeu de données complet, le nouvel estimateur sur le jeu de données tronqué ainsi que le *Mean-Excess plot* classique calculé sur l'échantillon tronqué.

Comme pour l'α-plot, on peut observer que les graphiques sont identiques au-dessus de 23 000. On pouvait s'y attendre étant donné que la troncature basse maximale est 23 026. Le problème des troncatures intervient sous cette valeur et le modèle actuel ne donne que le tracé du *Mean-Excess plot* au-dessus. Le nouvel estimateur conserve la même tendance que le graphique cible.

3.3 La distribution exponentielle

Cette section explique comment choisir et estimer une loi exponentielle et adapte le processus dans le cas d'une troncature à gauche aléatoire. Le plan sera similaire à celui de la distribution de Pareto.

3.3.1 Définition

Pour estimer une loi exponentielle sur des sinistres, on définit un seuil A et on ne considère que ceux de valeur supérieure à ce seuil. Ainsi, c'est une loi exponentielle tronquée qui est estimée. Comme cette loi est sans mémoire, cela revient à modéliser une loi exponentielle sur les excédents du seuil A.

La fonction de densité de probabilité de la loi exponentielle est donnée par :

$$f(x) = \lambda e^{-\lambda x} \tag{3.15}$$

avec :

— $\lambda > 0$, le paramètre d'échelle
— $x \in [0, \infty)$

et la fonction de répartition est :

$$F(x) = 1 - e^{-\lambda x} \tag{3.16}$$

Autrement dit, la densité de la loi exponentielle tronquée à gauche en A peut être écrite comme :

$$f_A(x) = \frac{f(x)}{1 - F(A)} = \frac{\lambda e^{-\lambda x}}{1 - (1 - e^{-\lambda A})} = \lambda e^{-\lambda(x-A)} \tag{3.17}$$

Pour modéliser cette distribution, il faut estimer le paramètre λ. Pour cela, la vraisemblance peut être utilisée. Pour un ensemble $x = [x_1, x_2, ..., x_n]$, $n \in \mathbb{N}$ et les paramètres de la loi exponentielle tronquée, elle vaut :

$$\mathcal{L}(\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda(x_i - A)} \tag{3.18}$$

Son calcul (immédiat) est présenté dans l'annexe B.

3.3.2 L'estimateur de la loi exponentielle

Le cas d'un jeu de données avec une troncature à gauche unique

Comme pour la loi de Pareto, la vraisemblance pour la loi exponentielle peut être exprimée avec une loi de probabilité Gamma :

$$\begin{aligned} \mathcal{L}(\alpha) &= \prod_{i=1}^{n} \lambda e^{-\lambda(x_i - A)} \\ &\sim \lambda^n e^{-\lambda \sum_{i=1}^{n}(x_i - A)} \\ &\sim Gamma\left(k = n + 1, \theta = \frac{1}{\sum_{i=1}^{n}(x_i - A)}\right) \end{aligned} \tag{3.19}$$

De cette manière, l'estimateur du maximum de vraisemblance est :

$$e^{\mathcal{L}}_{esperance} := \mathbb{E}_{\mathcal{L}}(\lambda) = k\theta = \frac{n+1}{\sum_{i=1}^{n}(x_i - A)} \tag{3.20}$$

Pour chaque sinistre x_i de l'ensemble, le seuil $A = x_i$ est défini et l'estimateur du maximum de vraisemblance est calculé sur le nouvel ensemble $\{x_j, \forall j | x_j > A = x_i\}$. En affichant la liste des

estimateurs en fonction des différents seuils A, un graphique similaire au α-plot de la distribution de Pareto est obtenu mais avec λ. Il est ensuite possible d'estimer sa valeur. Pour plus de lisibilité, l'inverse de λ est affiché en fonction des différents seuils A.

L'ajout de la troncature à gauche aléatoire

Comme pour la distribution de Pareto, l'estimateur du paramètre λ est adapté dans le cas d'une troncature à gauche aléatoire des données d'après le travail de Masquelein (2022*b*). Ainsi, pour un sinistre x_i, la nouvelle troncature est :

$$A_i := max(A, T_i)$$

La vraisemblance devient :

$$\begin{aligned}\mathcal{L}(\alpha) &= \prod_{i=1}^{n} \lambda e^{-\lambda(x_i - A_i)} \\ &\sim \lambda^n e^{-\lambda \sum_{i=1}^{n}(x_i - A_i)} \\ &\sim Gamma\left(k = n+1, \theta = \frac{1}{\sum_{i=1}^{n}(x_i - A_i)}\right)\end{aligned} \tag{3.21}$$

Et l'estimateur du maximum de vraisemblance :

$$e^{\mathcal{L}}_{esperance} := \mathbb{E}_{\mathcal{L}}(\lambda) = k\theta = \frac{n+1}{\sum_{i=1}^{n}(x_i - A_i)} \tag{3.22}$$

Le graphique des estimateurs du paramètre λ est réalisé de la même manière que pour l'ensemble avec une troncature fixe.

Un intervalle de confiance pour aider à la prise de décision

Comme pour l'estimateur du paramètre de la loi de Pareto, un intervalle de confiance est calculé grâce à la loi Gamma. Comme $\frac{1}{\lambda}$ est affiché au lieu de λ, l'intervalle de confiance doit être modifié.

En définissant c la probabilité d'être dans l'intervalle de confiance et $F_{\mathcal{G}}$ la fonction de répartition de la loi Gamma de paramètres k et θ, l'intervalle de confiance est délimité par les bornes suivantes :

$$Borne \quad Inférieure \quad = \quad \frac{1}{F_{\mathcal{G}}^{-1}\left(\frac{1+c}{2}\right)} \tag{3.23}$$

$$Borne \quad Supérieure \quad = \quad \frac{1}{F_{\mathcal{G}}^{-1}\left(\frac{1-c}{2}\right)} \tag{3.24}$$

3.3.3 Le nouvel estimateur du paramètre λ est plutôt bon

Il faut évaluer le nouvel estimateur de λ. Pour cela il est comparé avec un estimateur calculé sur un ensemble de sinistres non tronqués. Au lieu d'afficher λ dans le graphique, $1/\lambda$ est affiché pour plus de sens.

Un échantillon de 200 sinistres de loi exponentielle de paramètre $1/\lambda = 10\,000$ et de troncature fixe $A = 1000$ est simulé puis le graphique des estimateurs est construit. En pratique, cet ensemble n'est pas observé. Ensuite chaque sinistre est tronqué aléatoirement et un nouvel ensemble tronqué aléatoirement de 77 sinistres est obtenu. Il s'agit de ce qu'on aurait en réalité avec les seuils des cédantes. Le nouvel estimateur lui est appliqué.

En superposant les deux graphiques :

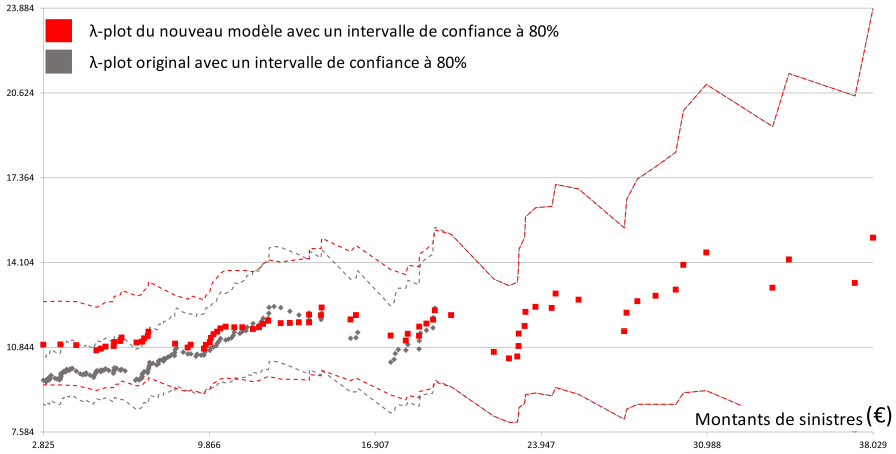

FIGURE 3.5 – Cet estimateur donne une bonne approximation du "λ-*plot*" original

Sur ce graphique, sont représentés le nouvel estimateur de λ sur l'ensemble des données tronquées aléatoirement, l'estimateur classique sur le même ensemble non tronqué et leurs intervalles de confiance respectifs. Les deux estimateurs sont identiques au-dessus d'environ 18 000. Cela vient du fait que la troncature à gauche maximale est 20 000 et qu'au-dessus de ce seuil, les deux ensembles sont égaux. Dans le modèle de QBE Re actuel, on n'est seulement capable d'obtenir la partie de l'estimateur au-dessus de ce seuil.

L'objectif est d'approcher le graphique gris qu'on ne peut pas obtenir directement en pratique. Dans cette nouvelle proposition, l'information du modèle de marché est gardée au-dessus de la troncature à gauche maximale et complète ce graphique en dessous. De l'information est donc rajoutée et elle parait cohérente en comparant avec l'estimateur original. Ce nouvel estimateur serait utilisable en pratique.

3.3.4 Le *Mean-Excess plot*

Sans troncature, le *Mean-Excess plot* ne dépend pas de la distribution. Il est défini dans la partie sur la loi de Pareto (3.2.4).

Il faut l'adapter dans le cas d'un ensemble de sinistres tronqués aléatoirement à gauche. Comme dans le cas de la loi de Pareto, Masquelein (2022*b*) suggère l'utilisation d'un processus *as-if*. Dans le cadre du *Mean-Excess plot* de la loi exponentielle :

$$\begin{aligned}\mathbb{P}(X \leq x_i | X > A_i) &= 1 - e^{-\lambda(x_i - A_i)} = 1 - e^{-\lambda((x_i - A_i + A) - A)} \\ &= \mathbb{P}(X \leq x_i - A_i + A | X > A)\end{aligned} \tag{3.25}$$

Soit

$$x_i^{as-if} = x_i - A_i + A \tag{3.26}$$

L'estimateur du *Mean-Excess plot* est donc :

$$\mathbb{E}(X - A|X > A) = \frac{1}{n_A} \sum_{i=1}^{n_A} (x_i - A_i) \tag{3.27}$$

avec n_A le nombre de sinistres au-dessus du seuil A.

Remarque

Ce Mean-Excess plot ne permet que de valider une distribution exponentielle comme il est réalisé en faisant l'hypothèse d'une loi exponentielle sur les données. Il ne pourra pas suggérer l'utilisation d'une loi de Pareto ou Pareto tronquée.

Cet estimateur est comparé en utilisant les deux mêmes ensembles que pour comparer l'estimateur du paramètre λ. Le graphique suivant est obtenu :

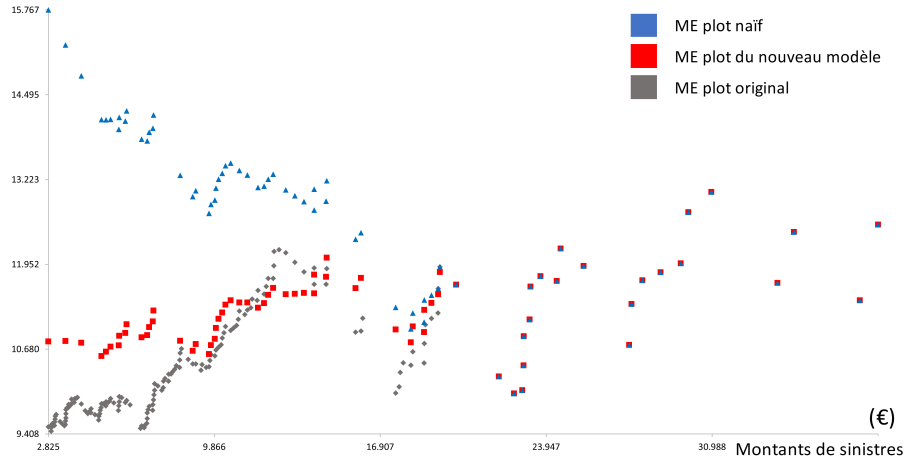

FIGURE 3.6 – Le nouvel estimateur donne une bonne approximation du *Mean-Excess plot* original

Ce graphique présente le *Mean-Excess plot* de l'ensemble original que l'on ne peut pas avoir en pratique par manque de données. Ensuite il y a le *Mean-Excess plot* classique réalisé sur les données tronquées (les données que l'on a en pratique). Enfin, le *Mean-Excess plot* obtenu sur les données tronquées avec le nouvel estimateur est représenté.

Comme dans le cas du graphique des estimateurs de λ, les *Mean-Excess plot* sont identiques au-delà de 18 000. Cela vient aussi de la valeur de la troncature à gauche maximale (20 000). Les deux ensembles sont identiques au-dessus de ce seuil. Cette partie n'est pas l'intérêt ici puisqu'elle est déjà maitrisée.

Lorsque l'on regarde la partie du graphique pour laquelle les ensembles sont différents, on s'aperçoit que le *Mean-Excess plot* classique sur les données tronquées a une tendance linéaire différente du *Mean-Excess plot* original des données non tronquées. En revanche, le nouvel estimateur donne la même tendance linéaire plate caractéristique de la loi exponentielle. Les deux graphiques ne se superposent pas exactement mais l'important ici est la tendance linéaire.

3.4 La distribution de Pareto tronquée

La dernière loi possible pour modéliser la queue de distribution est la loi de Pareto tronquée. On la choisit lorsque le *Mean-Excess plot* a une tendance linéaire décroissante. En pratique, la troncature peut traduire des sinistres avec des valeurs assurées et un montant de remboursement maximal.

Cette partie va expliquer et adapter le processus pour estimer les paramètres et construire le *Mean-Excess plot* en suivant un plan identique aux deux autres distributions. Cette loi est un peu plus délicate à estimer puisqu'il y a deux paramètres interdépendants : le paramètre de forme α et la troncature haute T.

3.4.1 Définition

La loi de Pareto tronquée est une loi de Pareto à laquelle on ajoute une troncature à droite (un maximum). La fonction de densité de probabilité est définie comme :

$$f(x) = \frac{\alpha \frac{A^\alpha}{x^{\alpha+1}}}{F_{A,\alpha}(T)} = \frac{\alpha \frac{A^\alpha}{x^{\alpha+1}}}{1 - \left(\frac{T}{A}\right)^{-\alpha}} \tag{3.28}$$

avec :

— $A > 0$, le paramètre d'échelle (seuil entre le corps et la queue de distribution)
— $\alpha > 0$, le paramètre de forme
— $T > A$, la troncature à droite
— $x \in [A, \infty)$
— $x \mapsto F_{A,\alpha}(x)$, la fonction de répartition de la loi de Pareto avec les paramètres A et α.

Pour modéliser cette distribution, il faut estimer α. Pour cela on utilise la vraisemblance (calculée en annexe B). Pour un ensemble $x = [x_1, x_2, ..., x_n]$, $n \in \mathbb{N}$ et les paramètres de la loi de Pareto tronquée, elle vaut :

$$\mathcal{L}(x) = \prod_{i=1}^{n} \frac{\frac{\alpha}{A}\left(\frac{x_i}{A}\right)^{-(\alpha+1)}}{1 - \left(\frac{T}{A}\right)^{-\alpha}} \tag{3.29}$$

3.4.2 L'estimateur de la loi de Pareto tronquée

Le cas d'un jeu de données avec une troncature à gauche unique

D'après la vraisemblance, la log-vraisemblance est calculée :

$$\begin{aligned} log\,\mathcal{L}(\alpha) &= \sum_{i=1}^{n_A} \left(ln(\alpha) - ln(A) - (\alpha+1)\, ln\left(\frac{x_i}{A}\right) - ln\left(1 - \left(\frac{T}{A}\right)^{-\alpha}\right)\right) \\ &= n_A\, ln(\alpha) - (1+\alpha) \sum_{i=1}^{n_A} ln\left(\frac{x_i}{A}\right) - n_A\, ln\left(1 - \left(\frac{T}{A}\right)^{-\alpha}\right) \end{aligned} \tag{3.30}$$

Et sa dérivée :

$$\frac{\partial log\,\mathcal{L}(\alpha)}{\partial \alpha} = \frac{n_A}{\alpha} - \sum_{i=1}^{n_A} ln\left(\frac{x_i}{A}\right) - n_A \frac{\left(\frac{T}{A}\right)^{-\alpha} ln\left(\frac{T}{A}\right)}{1-\left(\frac{T}{A}\right)^{-\alpha}} \tag{3.31}$$

A cause de la troncature à droite T, il n'est pas facile d'obtenir une expression pour α. Le processus présenté dans le livre de Albrecher et al. (2017) est utilisé.

Ils commencent par définir une première valeur pour la troncature à droite T.

$$T = max_i(x_i) \tag{3.32}$$

Ainsi que pour le paramètre α avec l'estimateur de Hill :

$$Hill(\alpha) := \frac{n_A}{\sum_{i=1}^{n_A} ln\left(\frac{x_i}{A}\right)} \tag{3.33}$$

On peut alors écrire :

$$\begin{aligned} \frac{\partial log\,\mathcal{L}(\alpha)}{\partial \alpha} &= \frac{n_A}{\alpha} - \frac{n_A}{Hill(\alpha)} - n_A \frac{\left(\frac{T}{A}\right)^{-\alpha} ln\left(\frac{T}{A}\right)}{1-\left(\frac{T}{A}\right)^{-\alpha}} \\ &= n_A \left(\frac{1}{\alpha} - \frac{1}{Hill(\alpha)} - \frac{\left(\frac{T}{A}\right)^{-\alpha} ln\left(\frac{T}{A}\right)}{1-\left(\frac{T}{A}\right)^{-\alpha}}\right) \end{aligned} \tag{3.34}$$

Il faut trouver α tel qu'il annule la quantité ci-dessus. Malheureusement il est toujours lié à la troncature à droite T. Ils utilisent alors l'algorithme de Newton-Raphson. La fonction à annuler avec l'algorithme est :

$$f(\alpha) := \frac{1}{\alpha} - \frac{\left(\frac{T}{A}\right)^{-\alpha} ln\left(\frac{T}{A}\right)}{1-\left(\frac{T}{A}\right)^{-\alpha}} - \frac{1}{Hill(\alpha)} \tag{3.35}$$

et sa dérivée est :

$$\begin{aligned} f'(\alpha) &= -\frac{1}{\alpha^2} - \frac{-\left(\frac{T}{A}\right)^{-\alpha} ln^2\left(\frac{T}{A}\right) * \left(1-\left(\frac{T}{A}\right)^{-\alpha}\right) - \left(\frac{T}{A}\right)^{-\alpha} ln^2\left(\frac{T}{A}\right) * \left(\frac{T}{A}\right)^{-\alpha}}{\left(1-\left(\frac{T}{A}\right)^{-\alpha}\right)^2} \\ &= -\frac{1}{\alpha^2} + \frac{\left(\frac{T}{A}\right)^{-\alpha} ln^2\left(\frac{T}{A}\right)}{\left(1-\left(\frac{T}{A}\right)^{-\alpha}\right)^2} \end{aligned} \tag{3.36}$$

Et avec Newton-Raphson :

$$\alpha_{l+1} = \alpha_l - \frac{f(\alpha_l)}{f'(\alpha_l)} \tag{3.37}$$

L'estimateur de la troncature à droite T est donné par :

$$log(T) = max\left(ln(A) + \frac{1}{\alpha} ln\left(1 + \frac{1 - \left(\frac{max_i(x_i)}{A}\right)^{-\alpha}}{\left(\frac{max_i(x_i)}{A}\right)^{-\alpha} - \frac{1}{n_A+1}}\right); ln(max_i(x_i))\right) \tag{3.38}$$

L'estimateur initial utilisé pour α est $Hill(\alpha)$ puis on boucle sur α et T jusqu'à ce qu'ils se stabilisent. En pratique, 3 ou 4 itérations suffisent.

Dans ce processus, il faut quand même faire attention à deux choses. D'abord, il faut respecter $\alpha > 0$. Si ce n'est pas le cas après une itération, on remplace notre estimateur pour α par $\alpha_{l+1} = 0,8 * \alpha_l$.

Ensuite, les quantités dans les fonctions logarithme dans l'expression de T doivent être positives. Si ce n'est pas le cas, la valeur précédente est conservée.

Petite adaptation

Ce procédé fonctionne bien mais **l'incertitude** n'est pas prise en compte. Pour avoir une telle distribution, Masquelein (2022*b*) utilise la *relativité* et l'*incertitude*. Il considère un processus *as-if* pour exprimer les sinistres suivant la loi de Pareto tronquée comme s'ils suivaient la loi de Pareto classique :

$$\begin{aligned} \mathbb{P}(X_T < x) &= \mathbb{P}(X < x^{as-if}) \\ \frac{1 - \left(\frac{x}{A}\right)^{-\alpha_T}}{1 - \left(\frac{T}{A}\right)^{-\alpha_T}} &= 1 - \left(\frac{x^{as-if}}{A}\right)^{-\alpha_T} \\ x^{as-if} &= A\left(\frac{\left(\frac{x}{A}\right)^{-\alpha_T} - \left(\frac{T}{A}\right)^{-\alpha_T}}{1 - \left(\frac{T}{A}\right)^{-\alpha_T}}\right)^{\frac{1}{\alpha_T}} \end{aligned} \tag{3.39}$$

Remarque

Bien sûr, le processus as-if dépend des précédents estimateurs α_T et T. Si on était capable de faire ce processus as-if sans avoir besoin de ces estimateurs, on pourrait trouver directement la vraie distribution.

Ensuite, une approximation acceptable est :

$$\alpha \sim Gamma\left(k = n_A + 1; \theta = \left(\frac{1}{\alpha_T}\sum_{i=1}^{n_A} ln\left(\frac{1 - \left(\frac{T}{A}\right)^{-\alpha_T}}{\left(\frac{x_i}{A}\right)^{-\alpha_T} - \left(\frac{T}{A}\right)^{-\alpha_T}}\right)\right)^{-1}\right) \tag{3.40}$$

Et l'estimateur du maximum de vraisemblance est :

$$\begin{aligned} e^{\mathcal{L}}_{esperance} &= \mathbb{E}_{\mathcal{L}}(\alpha) = k * \theta \\ &= \frac{n_A + 1}{\frac{1}{\alpha_T}\sum_{i=1}^{n_A} ln\left(\frac{1-\left(\frac{T}{A}\right)^{-\alpha_T}}{\left(\frac{x_i}{A}\right)^{-\alpha_T} - \left(\frac{T}{A}\right)^{-\alpha_T}}\right)} \end{aligned} \tag{3.41}$$

Pour finir, un α-*plot* est construit de la même manière que pour la loi de Pareto classique. Pour chaque siniste x_i de l'ensemble, $A = x_i$ est défini et l'estimateur du maximum de vraisemblance est calculé pour l'ensemble $\{x_j, \forall j | x_j > A = x_i\}$. En affichant ces estimateurs en fonctions des différents sinistres (seuils A), l'α-*plot* est obtenu et la valeur de α peut être estimée.

L'ajout de la troncature à gauche aléatoire

Comme pour les autres distributions, les estimateurs sont adaptés dans le cas d'une troncature à gauche aléatoire des données d'après la note de Masquelein (2022*b*). Ainsi, pour un sinistre x_i, la nouvelle troncature est :

$$A_i := max(A, T_i)$$

La vraisemblance devient :

$$\mathcal{L}(x) = \prod_{i=1}^{n} \frac{\frac{\alpha}{A_i}\left(\frac{x_i}{A_i}\right)^{-(\alpha+1)}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}} \tag{3.42}$$

et le procédé précédemment décrit est adapté avec cette nouvelle vraisemblance. Ainsi :

$$\begin{aligned} log\,\mathcal{L}(\alpha) &= \sum_{i=1}^{n_A}\left(ln(\alpha) - ln(A_i) - (\alpha+1)\,ln\left(\frac{x_i}{A_i}\right) - ln\left(1-\left(\frac{T}{A_i}\right)^{-\alpha}\right)\right) \\ &= n_A\,ln(\alpha) - (1+\alpha)\sum_{i=1}^{n_A} ln\left(\frac{x_i}{A_i}\right) - n_A\,ln\left(1-\left(\frac{T}{A_i}\right)^{-\alpha}\right) \end{aligned} \tag{3.43}$$

Puis

$$\frac{\partial log\,\mathcal{L}(\alpha)}{\partial\alpha} = \frac{n_A}{\alpha} - \sum_{i=1}^{n_A} ln\left(\frac{x_i}{A_i}\right) - n_A\frac{\left(\frac{T}{A_i}\right)^{-\alpha} ln\left(\frac{T}{A_i}\right)}{1-\left(\frac{T}{A_i}\right)^{-\alpha}} \tag{3.44}$$

Toujours à cause de la troncature à droite T, obtenir une expression analytique pour α n'est pas faisable. La méthode du livre de *Reinsurance* Albrecher et al. (2017) est utilisée.

Un estimateur initial pour la troncature haute T est encore défini :

$$T = max_i(x_i) \tag{3.45}$$

ensuite avec l'estimateur d'α suivant

$$Hill(\alpha) := \frac{n_A}{\sum_{i=1}^{n_A} ln\left(\frac{x_i}{A_i}\right)} \tag{3.46}$$

Il est possible d'écrire

$$\begin{aligned} \frac{\partial log\,\mathcal{L}(\alpha)}{\partial\alpha} &= \frac{n_A}{\alpha} - \frac{n_A}{Hill(\alpha)} - n_A\frac{\left(\frac{T}{A_i}\right)^{-\alpha} ln\left(\frac{T}{A_i}\right)}{1-\left(\frac{T}{A_i}\right)^{-\alpha}} \\ &= n_A\left(\frac{1}{\alpha} - \frac{1}{Hill(\alpha)} - \frac{\left(\frac{T}{A_i}\right)^{-\alpha} ln\left(\frac{T}{A_i}\right)}{1-\left(\frac{T}{A_i}\right)^{-\alpha}}\right) \end{aligned} \tag{3.47}$$

Pour trouver α tel que la quantité ci-dessus soit nulle, il faut encore utiliser l'algoritme de Newton-Raphson. α et T sont toujours liés. La fonction à annuler dans l'algorithme est :

$$f(\alpha) := \frac{1}{\alpha} - \frac{\left(\frac{T}{A_i}\right)^{-\alpha} ln\left(\frac{T}{A_i}\right)}{1-\left(\frac{T}{A_i}\right)^{-\alpha}} - \frac{1}{Hill(\alpha)} \tag{3.48}$$

et sa dérivée est

$$\begin{aligned} f'(\alpha) &= -\frac{1}{\alpha^2} - \frac{-\left(\frac{T}{A_i}\right)^{-\alpha} ln^2\left(\frac{T}{A_i}\right) * \left(1-\left(\frac{T}{A_i}\right)^{-\alpha}\right) - \left(\frac{T}{A_i}\right)^{-\alpha} ln^2\left(\frac{T}{A_i}\right) * \left(\frac{T}{A_i}\right)^{-\alpha}}{\left(1-\left(\frac{T}{A_i}\right)^{-\alpha}\right)^2} \\ &= -\frac{1}{\alpha^2} + \frac{\left(\frac{T}{A_i}\right)^{-\alpha} ln^2\left(\frac{T}{A_i}\right)}{\left(1-\left(\frac{T}{A_i}\right)^{-\alpha}\right)^2} \end{aligned} \tag{3.49}$$

L'algorithme de Newton-Raphson donne :

$$\alpha_{l+1} = \alpha_l - \frac{f(\alpha_l)}{f'(\alpha_l)} \tag{3.50}$$

Ensuite l'estimateur de la troncature à droite T reste le même :

$$log(T) = max\left(ln(A) + \frac{1}{\alpha} ln\left(1 + \frac{1-\left(\frac{max_i(x_i)}{A}\right)^{-\alpha}}{\left(\frac{max_i(x_i)}{A}\right)^{-\alpha} - \frac{1}{n_A+1}}\right); ln(max_i(x_i))\right) \tag{3.51}$$

Le premier estimateur pour α est $Hill(\alpha)$ puis on boucle sur α et T jusqu'à ce qu'ils se stabilisent. 3 à 4 itérations suffisent.

Comme pour le procédé dans le cas d'une troncature à gauche fixe, il faut faire attention à deux choses. La première est la positivité de α. Si ce n'est pas le cas, l'estimateur pour α est remplacé par $\alpha_{l+1} = 0,8 * \alpha_l$.

La deuxième condition à vérifier est la positivité des quantités dans les logarithmes pour le calcul de $log(T)$. Si ce n'est pas le cas, la valeur précédente est conservée.

Petite adaptation

Cette méthode fonctionne bien mais elle manque aussi d'*incertitude*. Pour avoir une distribution, la *relativité* et l'*incertitude* sont utilisées comme fait Masquelein (2022*b*). Un processus *as-if* est donc considéré pour exprimer les sinistres comme s'ils suivaient une loi de Pareto non tronquée :

$$\begin{aligned} \mathbb{P}(X_T < x | X_T > A_i) &= \mathbb{P}(X < x^{as-if} | X_T > A_i^{as-if}) \\ \frac{1-\left(\frac{x}{A_i}\right)^{-\alpha_T}}{1-\left(\frac{T}{A_i}\right)^{-\alpha_T}} &= 1 - \left(\frac{x^{as-if}}{A_i^{as-if}}\right)^{-\alpha_T} \\ \frac{x^{as-if}}{A_i^{as-if}} &= \left(\frac{\left(\frac{x}{A_i}\right)^{-\alpha_T} - \left(\frac{T}{A_i}\right)^{-\alpha_T}}{1-\left(\frac{T}{A_i}\right)^{-\alpha_T}}\right)^{\frac{1}{\alpha_T}} \end{aligned} \tag{3.52}$$

Remarque

Ce processus as-if dépend des estimateurs précédents pour α_T et T. Si on était capable de faire la même chose sans les utiliser, on pourrait directement obtenir la distribution réelle.

Ensuite une approximation acceptable peut être :

$$\alpha \sim Gamma\left(k = n_A + 1; \theta = \left(\frac{1}{\alpha_T}\sum_{i=1}^{n_A} ln\left(\frac{1-\left(\frac{T}{A_i}\right)^{-\alpha_T}}{\left(\frac{x_i}{A_i}\right)^{-\alpha_T} - \left(\frac{T}{A_i}\right)^{-\alpha_T}}\right)\right)^{-1}\right) \tag{3.53}$$

Et l'estimateur du maximum de vraisemblance est :

$$\begin{aligned} e^{\mathcal{L}}_{esperance} &= \mathbb{E}_{\mathcal{L}}(\alpha) = k * \theta \\ &= \frac{n_A + 1}{\frac{1}{\alpha_T}\sum_{i=1}^{n_A} ln\left(\frac{1-\left(\frac{T}{A_i}\right)^{-\alpha_T}}{\left(\frac{x_i}{A_i}\right)^{-\alpha_T} - \left(\frac{T}{A_i}\right)^{-\alpha_T}}\right)} \end{aligned} \tag{3.54}$$

L'*α-plot* peut alors être construit à partir de cet estimateur.

Un intervalle de confiance pour α

L'intervalle de confiance est calculé comme pour les deux autres distributions grâce à la loi Gamma.

En définissant c la probabilité d'être dans l'intervalle de confiance et $F_{\mathcal{G}}$ la fonction de répartition de la loi Gamma de paramètres k et θ, les bornes de l'intervalle sont données par :

$$Borne \quad Inférieure \quad = \quad F_{\mathcal{G}}^{-1}\left(\frac{1-c}{2}\right) \tag{3.55}$$

$$Borne \quad Supérieure \quad = \quad F_{\mathcal{G}}^{-1}\left(\frac{1+c}{2}\right) \tag{3.56}$$

3.4.3 Le nouvel estimateur donne une bonne approximation de l'*α-plot*

Dans cette partie, le nouvel estimateur est évalué. Pour cela, un ensemble de 200 sinistres est simulé avec une loi de Pareto tronquée de paramètres $x_m = 10\ 000$, $\alpha = 3$ et $T = 30\ 000$ puis son *α-plot* est calculé. Ensuite, une troncature aléatoire est rajoutée et un ensemble de 121 sinistres tronqués est obtenu. Voici le graphique :

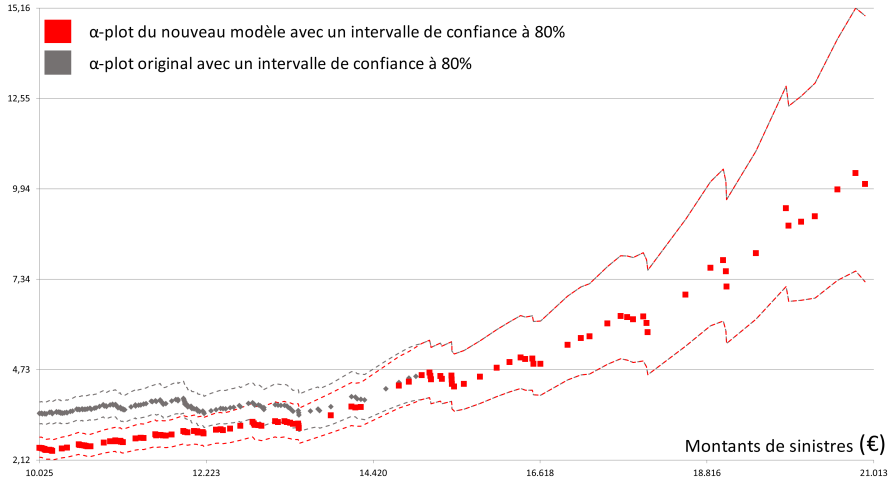

FIGURE 3.7 – Le nouvel estimateur donne une bonne approximation de l'α-*plot* original

Ce graphique représente l'α-*plot* obtenu avec le nouvel estimateur et l'α-*plot* original du jeu de données non tronqué ainsi que leurs intervalles de confiance respectifs.

Au-dessus de 15 000, les graphiques sont identiques, c'est cohérent avec le fait que la troncature à gauche maximale soit 16 118 et qu'au-delà, les deux ensembles de sinistres sont identiques. Le modèle actuel est seulement capable d'avoir cette partie au-dessus de ce seuil. Le défi est de compléter ce graphique en dessous pour essayer d'avoir le graphique gris. L'estimateur présente une déviation mais reste relativement proche du graphique original.

3.4.4 L'estimateur de la troncature à droite T (maximum)

Dans la loi de Pareto tronquée, le paramètre de forme α a été estimé mais il reste la troncature à droite T à estimer. Pour cela, il suffit de reprendre le T estimé pour chaque α lors de la boucle sur α et T.

Ainsi un paramètre T est estimé pour chaque ensemble $\{x_j, \forall j | x_j > A = x_i\}_i$. En utilisant les deux mêmes échantillons que pour comparer les α-plot, le graphique suivant est obtenu pour T :

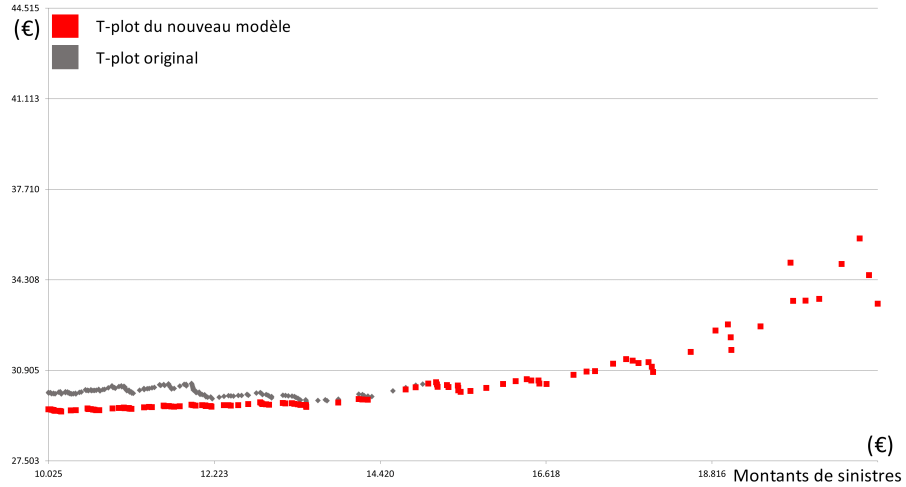

FIGURE 3.8 – L'estimateur donne une bonne approximation du T original

Si l'ensemble n'était pas tronqué, il serait possible d'avoir directement le T original. Dans la réalité, ce n'est pas le cas alors il faut essayer de l'approcher avec le nouvel estimateur. La partie d'intérêt est celle sous la troncature à gauche maximale. Le modèle actuel n'est pas capable de la calculer mais le nouvel estimateur pourrait en trouver une bonne approximation.

3.4.5 Le *Mean-Excess plot* pour valider la distribution

Le *Mean-Excess plot* pour un ensemble non tronqué ne dépend pas de la distribution de l'ensemble. C'est donc le même que celui défini pour la loi de Pareto.

Dans le cas d'un ensemble tronqué aléatoirement à gauche, on ne peut utiliser la formule du *Mean-Excess plot* classique qu'au-dessus de la plus grande troncature à gauche. Maintenant, le calcul doit être adapté pour prendre en compte les troncatures aléatoires et obtenir une approximation du *Mean-Excess plot* en dessous de la troncature à gauche maximale.

Comme pour la loi de Pareto, un processus *as-if* introduit par Masquelein (2022*b*) est utilisé. Dans le cadre du *Mean-Excess plot* pour la loi de Pareto tronquée, il obtient :

$$\begin{aligned}
\mathbb{P}(X_T \leq x_i | X_T > A_i) &= \frac{1-\left(\frac{x_i}{A_i}\right)^{-\alpha}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}} = \frac{\frac{1-\left(\frac{T}{A}\right)^{-\alpha}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}}\left(1-\left(\frac{x_i}{A_i}\right)^{-\alpha}\right)}{1-\left(\frac{T}{A}\right)^{-\alpha}} \\
&= \frac{1-\left(\frac{1}{A} * \left(A\left(1-\frac{1-\left(\frac{T}{A}\right)^{-\alpha}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}}\left(1-\left(\frac{x_i}{A_i}\right)^{-\alpha}\right)\right)^{-\frac{1}{\alpha}}\right)\right)^{-\alpha}}{1-\left(\frac{T}{A}\right)^{-\alpha}} \\
&= \mathbb{P}\left(X \leq A\left(1-\frac{1-\left(\frac{T}{A}\right)^{-\alpha}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}}\left(1-\left(\frac{x_i}{A_i}\right)^{-\alpha}\right)\right)^{-\frac{1}{\alpha}} | X > A\right)
\end{aligned} \tag{3.57}$$

Ensuite

$$x_i^{as-if} = A\left(1 - \frac{1-\left(\frac{T}{A}\right)^{-\alpha}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}}\left(1-\left(\frac{x_i}{A_i}\right)^{-\alpha}\right)\right)^{-\frac{1}{\alpha}} \tag{3.58}$$

Donc l'estimateur du *Mean-Excess plot* sera :

$$\mathbb{E}(X-A|X>A) = \frac{1}{n_A}\sum_{i=1}^{n_A}\left(A\left(1 - \frac{1-\left(\frac{T}{A}\right)^{-\alpha}}{1-\left(\frac{T}{A_i}\right)^{-\alpha}}\left(1-\left(\frac{x_i}{A_i}\right)^{-\alpha}\right)\right)^{-\frac{1}{\alpha}} - A\right) \tag{3.59}$$

avec n_A le nombre de sinistres au-dessus du seuil A.

Remarque

Comme pour les autres distributions, ce Mean-Excess plot estimé ne peut que valider la loi faite en hypothèse. Ici on ne pourra que valider la loi de Pareto tronquée et pas les autres.

Dans le cas de la loi de Pareto tronquée, la difficulté est qu'il faut connaitre α et T dans l'estimateur du *Mean-Excess plot*. Même si le but de ce graphique est de choisir la loi à modéliser, il faut d'abord faire l'hypothèse d'une loi de Pareto tronquée, calculer les estimateurs puis valider (ou pas) l'hypothèse à la fin avec le *Mean-Excess plot*.

Deux *Mean-Excess plot* sont créés. Le premier est a priori. Pour chaque point (troncature à gauche commune), l'α et le T calculés en ce point sont utilisés.

Le second est a posteriori. il est calculé en utilisant l'α et le T estimés finalement par l'utilisateur.

A présent, le nouvel estimateur sur l'ensemble tronqué est comparé avec le véritable *Mean-Excess plot* calculé sur l'ensemble non tronqué. Il en découle deux graphiques, un a priori et un a posteriori :

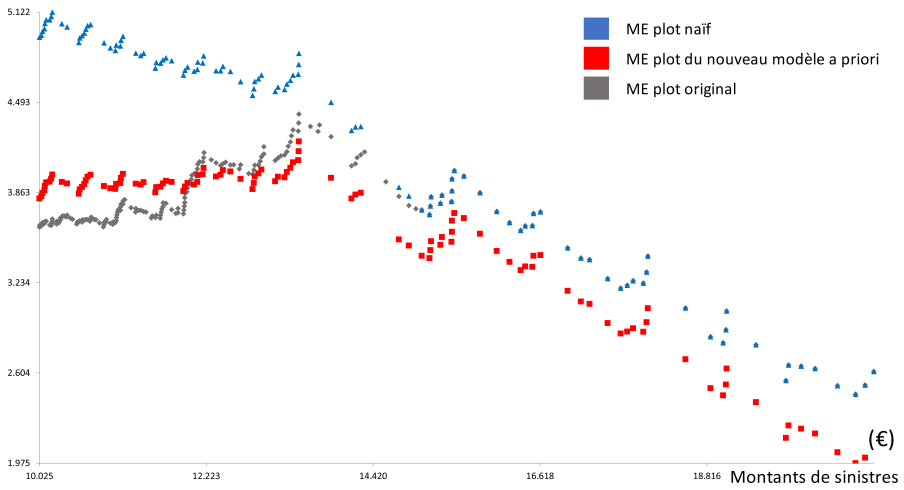

FIGURE 3.9 – L'estimateur a priori trouve la tendance avec un décalage par rapport au *Mean-Excess plot* original

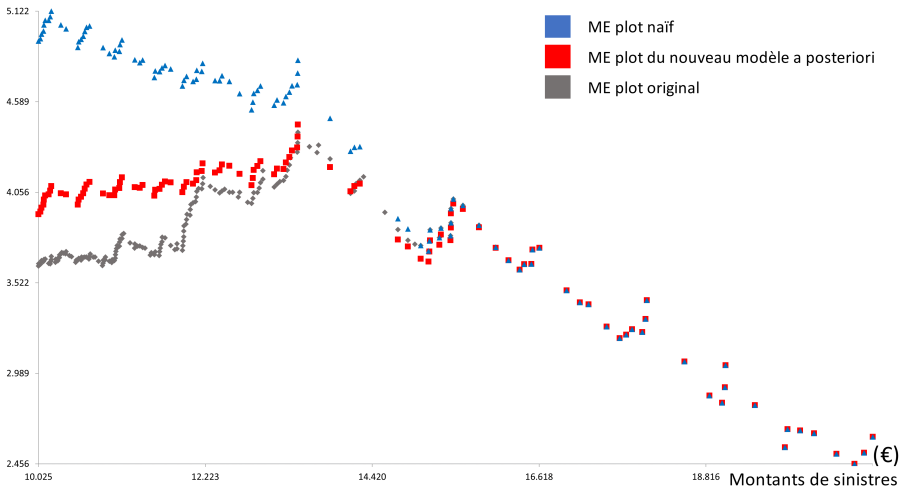

FIGURE 3.10 – L'estimateur a posteriori corrige le décalage

Pour les deux graphiques, il y a les *Mean-Excess plot* originaux calculés à partir de l'ensemble non tronqué. Ce sont ces graphiques que l'on ne peut pas avoir en pratique et qu'on cherche à approcher. Ensuite, il y a les deux *Mean-Excess plot* calculés par les estimateurs a priori et a posteriori. On remarque que celui a priori trouve la même tendance que celui à approcher mais avec un décalage. Celui calculé a posteriori corrige ce décalage. Enfin, le *Mean-Excess plot* classique calculé sur l'ensemble tronqué est représenté comme témoin.

La troncature à gauche maximale est 16 118. Cela veut dire que les deux ensembles sont identiques au-dessus de cette valeur et que les *Mean-Excess plot* devraient être confondus. C'est ce qu'on peut observer sur le second graphique mais pas sur le premier. Même si le nouvel estimateur a priori semble donner la même tendance, il n'est pas bon.

Sur le second graphique, l'estimateur a posteriori est mieux que l'estimateur naif puisqu'il est égale au graphique cible après la troncature à gauche maximale et conserve la même tendance avant.

En pratique, on ne peut pas avoir le graphique original mais on peut avoir les deux autres. On pourrait imaginer construire le graphique bleu, puis le rouge tel qu'il se superpose dessus après la troncature à gauche maximale. Cela pourrait aider à confirmer l'absence de décalage pour le nouvel estimateur.

En résumé, la partie au-dessus de la troncature à gauche maximale n'est pas le centre d'intérêt dans ce travail parce qu'elle est déjà maitrisée. En revanche elle peut donner une indication sur la qualité du nouvel estimateur. L'objectif est d'obtenir le graphique gris (au moins la tendance) avant 16 118. Dans ce but, l'estimateur est satisfaisant.

3.4.6 Tester l'hypothèse d'une troncature dans la loi de Pareto

Lorsque que la troncature à gauche est unique

Il est possible de tester si un ensemble a bien une troncature à droite ou un maximum. Autrement dit, l'idée est de savoir s'il peut être raisonnablement modélisé par une loi de Pareto tronquée. Cette partie s'appuie sur un test présenté dans le livre de Albrecher et al. (2017).

Ce test repose sur la vérification des hypothèses suivantes :

— H_0 **:** X satisfait une **troncature faible (ou aucune) à droite** en t lorsque $\frac{T}{t} \to \beta$
— H_1 **:** X satisfait une **troncature forte à droite** en t lorsque $\frac{T}{t} \to \beta$

La **troncature faible à droite** est définie quand $\frac{X}{t} \to \beta$ si

$$\mathbb{P}\left(\frac{X}{t} > y | X > t\right) \to y^{-\alpha} \tag{3.60}$$

La **troncature forte à droite** est définie quand $\frac{X}{t} \to \beta$ si

$$\mathbb{P}\left(\frac{X}{t} > y | X > t\right) \to \frac{y^{-\alpha} - \beta^{-\alpha}}{1 - \beta^{-\alpha}} \tag{3.61}$$

Comment réaliser le test ?

Ils calculent les quantités suivantes :

$$\bar{R}_{A,n}(\alpha) := \frac{\sum_{i=1}^{n} \left(\frac{x_i}{A}\right)^{-\alpha} \mathbb{1}_{\{x_i > A\}}}{\sum_{i=1}^{n} \mathbb{1}_{\{x_i > A\}}} \tag{3.62}$$

$$L_{A,n}(\alpha) := \frac{\bar{R}_{A,n}(\alpha) - \frac{1}{2}}{1 - \bar{R}_{A,n}(\alpha)} \tag{3.63}$$

L'hypothèse H_0 est **rejetée** avec un niveau q (i.e. la sévérité est significativement tronquée à droite) quand :

$$T_{B,A,n} := \sqrt{12 * \left(\sum_{i=1}^{n} \mathbb{1}_{x_i > A}\right)} L_{A,n}(Hill_{A,n}) < -z_q \tag{3.64}$$

avec

$$\mathbb{P}(\mathcal{N}(0,1) > z_q) = q$$

La ***P-Value*** est donnée par $\Phi\left(\sqrt{12 * (\sum_{i=1}^{n} \mathbb{1}_{x_i > A})} L_{A,n}(Hill_{A,n})\right)$

Une troncature à droite peut être considérée quand :

$$\Phi\left(\sqrt{12 * \left(\sum_{i=1}^{n} \mathbb{1}_{x_i > A}\right)} L_{A,n}(Hill_{A,n})\right) < q \tag{3.65}$$

Ce test est réalisé sur un échantillon simulé d'après une loi de Pareto tronquée (troncature haute). La *p-value*, l'*α-plot* et un seuil à 5% pour analyser la *p-value* sont affichés sur le même graphique :

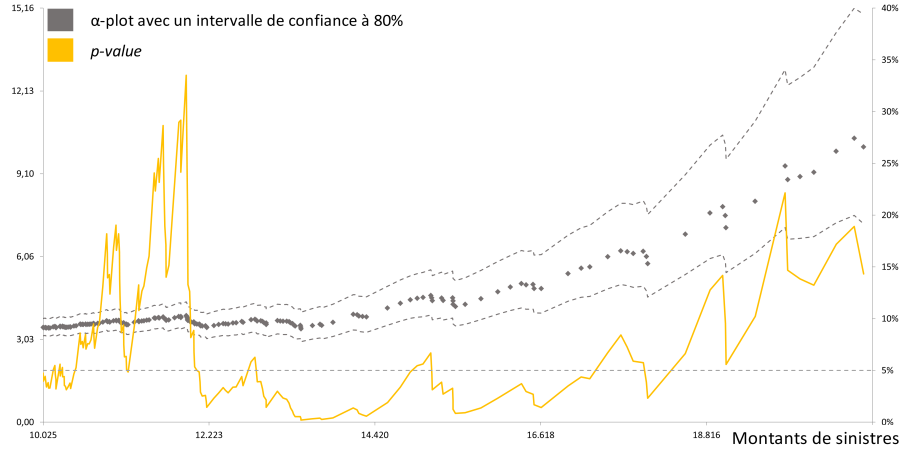

FIGURE 3.11 – *α-plot* et *p-value* sur un échantillon non tronqué aléatoirement à gauche, suivant une loi de Pareto tronquée

Sur ce graphique, il y a plusieurs choses à remarquer. D'abord la *p-value* augmente à mesure qu'on se rapproche de la troncature. Cela est normal. On calcule la *p-value* en un point avec les sinistres supérieurs ou égaux à ce point. Plus on se rapproche de la troncature, moins on a de sinistres et le test a plus de mal à capter la convergence de l'ensemble.

Ensuite, plusieurs pics sont présents pour la *p-value* entre 10 000 et 12 000. Au premier abord, ils semblent anormaux mais ils s'expliquent en regardant l'α-plot. Il y a une densité importante de points au niveau de ces pics. Cette densité importante à cet endroit rend difficile la preuve de convergence de l'ensemble au niveau de la troncature réelle.

L'ajout la troncature aléatoire à gauche

Maintenant, le test est adapté pour l'utiliser dans le cas d'une troncature à gauche aléatoire des données. Ce travail s'appuie sur le papier de Masquelein (2022*b*).

Le seul changement est pour la troncature à gauche de chaque sinistre. Elle devient $A_i = max(A, T_i)$ avec T_i la troncature basse associée au sinistre x_i.

La validation repose sur les mêmes hypothèses à vérifier :

— H_0 **:** X satisfait une **troncature faible à droite (ou aucune)** en t lorsque $\frac{T}{t} \to \beta$
— H_1 **:** X satisfait une **troncature forte à droite** en t lorsque $\frac{T}{t} \to \beta$

Par rapport au cas précédent, le seul changement est $\bar{R}_{A,n}(\alpha)$

$$\bar{R}_{A,n}(\alpha) := \frac{\sum_{i=1}^{n} \left(\frac{x_i}{A_i}\right)^{-\alpha} \mathbb{1}_{\{x_i > A\}}}{\sum_{i=1}^{n} \mathbb{1}_{\{x_i > A\}}} \tag{3.66}$$

La ***P-Value*** est encore donnée par $\Phi\left(\sqrt{12 * (\sum_{i=1}^{n} \mathbb{1}_{x_i > A})} L_{A,n}(Hill_{A,n})\right)$

avec

$$L_{A,n}(\alpha) := \frac{\bar{R}_{A,n}(\alpha) - \frac{1}{2}}{1 - \bar{R}_{A,n}(\alpha)} \tag{3.67}$$

Une troncature à droite peut être considérée quand :

$$\Phi\left(\sqrt{12 * \left(\sum_{i=1}^{n} \mathbb{1}_{x_i > A}\right)} L_{A,n}(Hill_{A,n})\right) < q \tag{3.68}$$

Pour essayer ce test, l'échantillon utilisé précédemment est repris en lui ajoutant une troncature à gauche aléatoire. Le test est effectué et la *p-value*, un seuil à 5% et son *α-plot* sont affichés sur un même graphique :

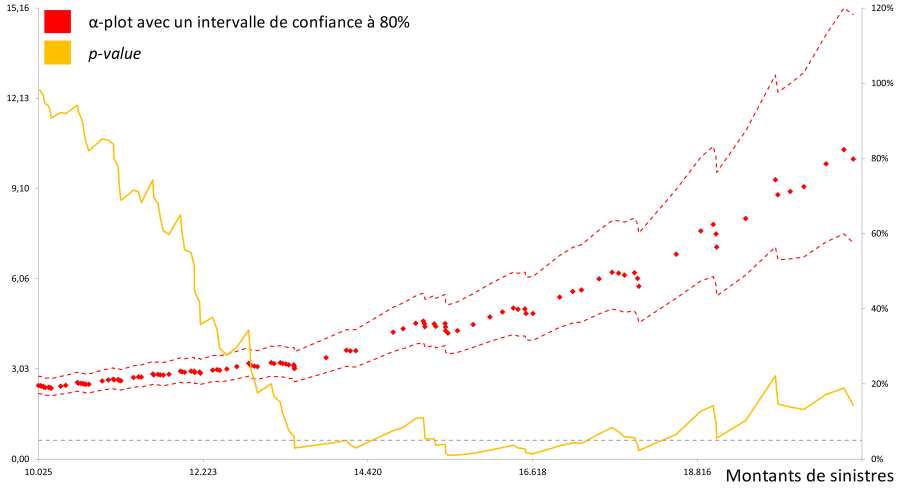

FIGURE 3.12 – *α-plot* et *p-value* sur un échantillon tronqué aléatoirement à gauche, suivant une loi de Pareto tronquée

C'est intéressant de comparer ce graphique avec le précédent. L'échelle est différente mais les *p-value* sont de plus en plus proches lorsque qu'on se rapproche des grandes valeurs. Cela est rassurant puisque les deux échantillons (tronqués et non tronqués) sont de plus en plus en semblables pour les grandes valeurs. Ainsi la *p-value* augmente lorsque l'on se rapproche de la troncature à droite pour les mêmes raisons que pour le graphique précédent.

La différence notable entre les deux graphiques concerne les plus petites valeurs de l'échantillon. Ce sont ces valeurs qui sont le plus impactées par la troncature aléatoire à gauche. Avec toutes ces troncatures à gauche, les sinistres sont réévalués et la convergence vers la troncature à droite n'est plus assez claire pour la *p-value*.

Ce deuxième graphique n'est pas très satisfaisant. La *p-value* prend de grandes valeurs alors que l'ensemble est tronqué. Ce test ne semble pas très pertinent. Cependant la *p-value* atteint aussi des valeurs en dessous du seuil de 5%. Pour réhabiliter le test, il est réalisé sur un ensemble non tronqué à droite. Pour ça, un ensemble de sinistres suivant la loi de Pareto de paramètres $X_m = 10\ 000$ et $\alpha = 3$ est simulé. Une troncature aléatoire à gauche est aussi ajoutée et un ensemble de 73 sinistres est obtenu. Comme pour les autres tests, la *p-value*, un seuil à 5% et l'*α-plot* sont affichés sur le même graphique :

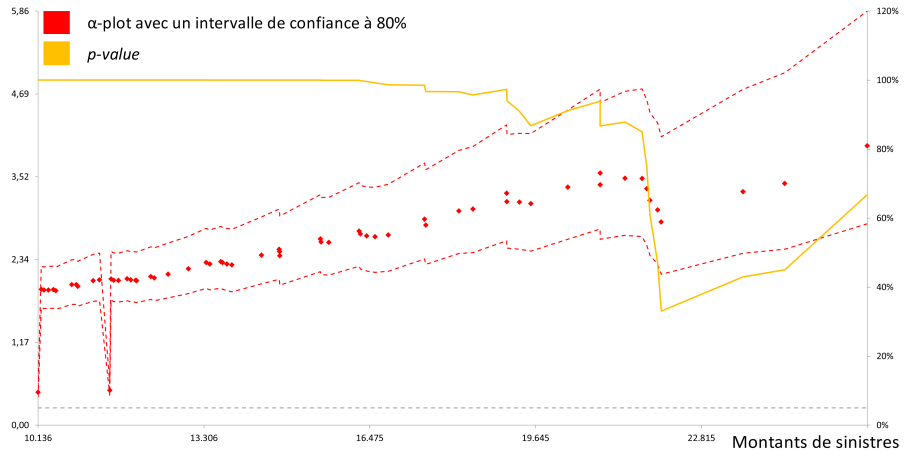

FIGURE 3.13 – *α-plot* et *p-value* sur un échantillon tronqué aléatoirement à gauche, suivant une loi de Pareto

Ce graphique permet de relativiser le graphique précédent. Lorsque l'ensemble n'est pas tronqué à droite, la *p-value* est **toujours** bien au-dessus du seuil de 5%. L'absence de troncature à droite ne fait pas de doute.

Pour finir, le même graphique est affiché pour l'ensemble original (avant la troncature aléatoire à gauche).

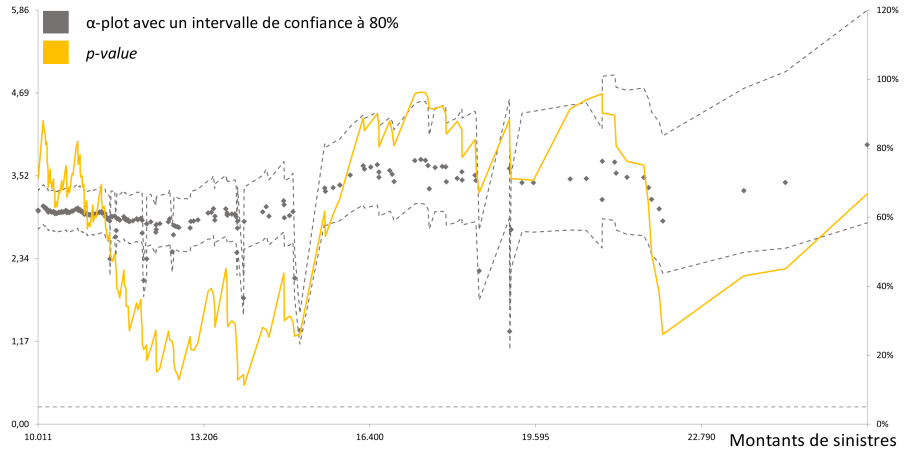

FIGURE 3.14 – *α-plot* et *p-value* sur un échantillon suivant une loi de Pareto

Dans ce cas, la *p-value* est aussi **toujours** largement au-dessus du seuil de 5%. Ces deux exemples montrent qu'il n'y a pas de doute dans le cas d'un échantillon non tronqué à droite. C'est un bon argument pour pouvoir continuer à utiliser ce test.

3.5 Conclusion

A travers ce chapitre, les estimateurs des distributions de Pareto, de Pareto tronquée et exponentielle ont été adaptés dans le cas de données tronquées **aléatoirement** à gauche. Ces trois lois permettent de modéliser tous les profils de queues de distributions. On peut désormais modéliser les queues de distributions à partir de sinistres tronqués aléatoirement à gauche.

Il faut maintenant joindre le corps et la queue des distributions. Encore une fois, il faut prendre en considération les troncatures **aléatoires**. C'est ce qui est fait dans le prochain chapitre.

Chapitre 4

Assembler le corps et la queue de distribution

Une fois que les lois pour le corps et pour la queue de la distribution ont été modélisées, elles doivent être assemblées pour former une seule loi sur tout l'ensemble des sinistres. La distribution finale est une mixture de la loi du corps et de la loi de la queue de distribution. Pour pouvoir créer cette mixture, il ne manque plus que les probabilités d'être dans le corps ou la queue.

Dans ce chapitre, il est décrit comment obtenir ces probabilités. La distribution finale aura ensuite cette forme :

$$f_{Severite}(x) = \mathbb{P}(x \in Corps) * f_{Corps}(x) + \mathbb{P}(x \in Queue) * f_{Queue}(x) \tag{4.1}$$

Sans aucune troncature aléatoire, il serait très facile d'estimer ces probabilités. Par exemple la probabilité qu'un sinistre appartienne au corps de la distribution serait simplement le rapport entre le nombre de sinistres observés dans le corps de la distribution et le nombre de sinistres observés total. Malheureusement, la troncature aléatoire à gauche rend cette estimation plus délicate. Dans ce chapitre, deux méthodes sont envisagées et comparées.

4.1 Une première méthode envisagée

La première solution est assez basique. La sévérité du corps de la distribution est estimée avec toutes les données (du corps et de la queue). La loi de la sévérité du corps de la distribution n'est alors plus tronquée à droite. Elle donne directement la probabilité d'être dans le corps de la distribution au niveau du seuil entre le corps et la queue.

$$\mathbb{P}(X \in Corps) = F_{Corps}(Seuil\,de\,la\,queue\,de\,distribution) \tag{4.2}$$

Graphiquement, cette méthode d'assemblage ressemble à ça :

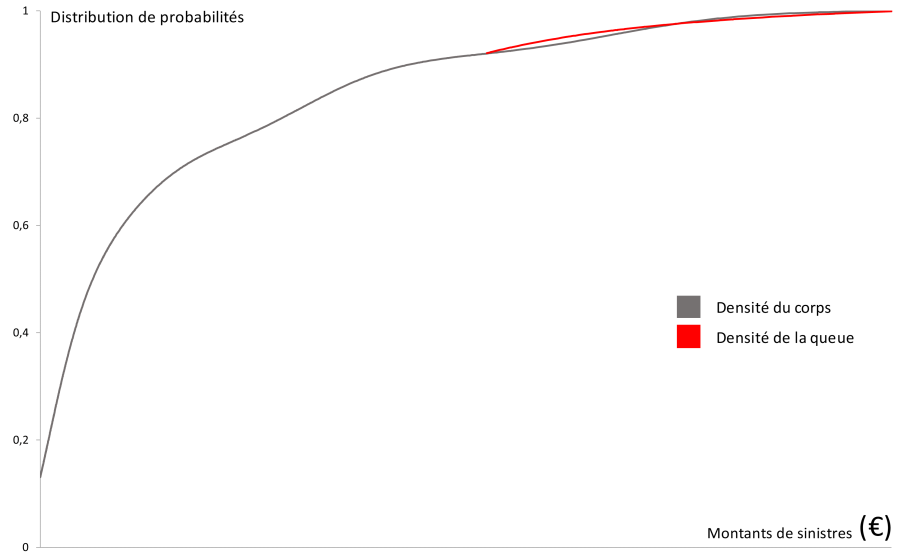

FIGURE 4.1 – Assemblage de la densité du corps de la distribution et de la densité de la queue de distribution

Ce graphique présente la densité du corps de la distribution et, au seuil de la queue de distribution, la densité de la queue de distribution est "collée".

Un gros plan sur le raccord au niveau du seuil de la queue de distribution permet de voir très distinctement les deux distributions.

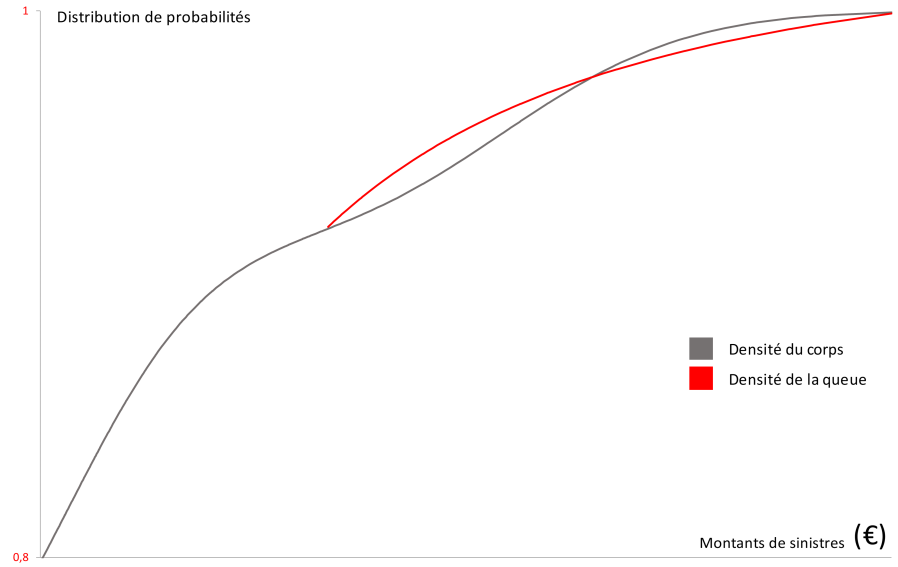

FIGURE 4.2 – On distingue la densité du corps de la distribution et la densité de la queue de distribution

4.2 Une seconde méthode développée

Pour la seconde méthode, les probabilités sont calculées en utilisant un estimateur du maximum de vraisemblance. De cette manière, les troncatures aléatoires à gauche peuvent être prises en compte. Une loi binomiale est utilisée pour donner la probabilité d'être soit dans le corps de la distribution, soit dans la queue.

Soient :

— p_C : La probabilité d'être dans le corps de la distribution
— p_Q : La probabilité d'être dans la queue de la distribution
— (x_i, t_i) : Une observation et sa troncature aléatoire à gauche associée
— $F_C(x) = \mathbb{P}(X < x | X \in Corps)$: La fonction de répartition du corps de la distribution
— $F_Q(x) = \mathbb{P}(X < x | X \in Queue)$: La fonction de répartition de la queue de la distribution

Pour calculer les probabilités p_C et p_Q, les probabilités d'être dans le corps ou la queue sont calculées a posteriori pour chaque sinistre en fonction de son domaine de définition. C'est-à-dire que pour chaque couple (x_i, t_i), $p_Q^i = \mathbb{P}(X_i > T | X_i > t_i)$ avec T le seuil entre le corps et la queue de distribution est calculé.

Si $t_i > T$:

$$\mathbb{P}(X_i > T | X_i > t_i) = 1$$

Ces sinistres ne vont rien apporter dans l'optimisation de la vraisemblance donc seuls ceux tels que $t_i \leq T$ sont gardés.

$$\begin{aligned} p_Q^i = \frac{\mathbb{P}(X_i > T)}{\mathbb{P}(X_i > t_i)} &= \frac{p_Q}{p_Q(1 - F_Q(t_i)) + p_C(1 - F_C(t_i))} \\ &= \frac{p_Q}{p_Q + p_C(1 - F_C(t_i))} \\ &= \frac{p_Q}{p_Q + (1 - p_Q)(1 - F_C(t_i))} \end{aligned} \tag{4.3}$$

et

$$p_C^i = 1 - p_Q^i = 1 - \frac{p_Q}{p_Q + (1 - p_Q)(1 - F_C(t_i))} \tag{4.4}$$

Maintenant la vraisemblance est :

$$\begin{aligned} \mathcal{L}(p_Q) &= \mathbb{P}(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) \\ &= \prod_{i=1}^{n} \mathbb{P}(X_i = x_i) \\ &= \prod_{x_i \in Corps} \mathbb{P}(X_i = x_i) * \prod_{x_i \in Queue} \mathbb{P}(X_i = x_i) \\ &= \prod_{x_i \in Corps} (1 - p_Q^i) * \prod_{x_i \in Queue} p_Q^i \\ &= \prod_{x_i \in Corps} \left(1 - \frac{p_Q}{p_Q + (1 - p_Q)(1 - F_C(t_i))}\right) * \prod_{x_i \in Queue} \frac{p_Q}{p_Q + (1 - p_Q)(1 - F_C(t_i))} \end{aligned} \tag{4.5}$$

Puis la log-vraisemblance est :

$$\begin{aligned} log\,\mathcal{L}(p_Q) &= \sum_{x_i \in Corps} (ln(1-p_Q) + ln(1-F_C(t_i)) - ln(p_Q + (1-p_Q)(1-F_C(t_i)))) \\ &+ \sum_{x_i \in Queue} (ln(p_Q) - ln(p_Q + (1-p_Q)(1-F_C(t_i)))) \\ &= n_{Corps} * ln(1-p_Q) + n_{Queue} * ln(p_Q) + \sum_{x_i \in Corps} ln(1-F_C(t_i)) \\ &- \sum_{i=1}^{n} ln(p_Q + (1-p_Q)(1-F_C(t_i))) \end{aligned} \tag{4.6}$$

Et sa dérivée est :

$$\begin{aligned} \frac{\partial log\,\mathcal{L}(p_Q)}{\partial p_Q} &= -\frac{n_{Corps}}{1-p_Q} + \frac{n_{Queue}}{p_Q} - \sum_{i=1}^{n} \frac{1-(1-F_C(t_i))}{p_Q + (1-p_Q)(1-F_C(t_i))} \\ &= -\frac{n_{Corps}}{1-p_Q} + \frac{n_{Queue}}{p_Q} - \sum_{i=1}^{n} \frac{F_C(t_i)}{p_Q + (1-p_Q)(1-F_C(t_i))} \end{aligned} \tag{4.7}$$

Il faut trouver p_Q qui annule cette fonction. Malheureusement, il n'y a pas de formule fermée pour p_Q. L'algorithme de Newton-Raphson est alors utilisé. Soit f, la fonction à annuler :

$$f(p_Q) = -\frac{n_{Corps}}{1-p_Q} + \frac{n_{Queue}}{p_Q} - \sum_{i=1}^{n} \frac{F_C(t_i)}{p_Q + (1-p_Q)(1-F_C(t_i))} \tag{4.8}$$

$$f'(p_Q) = -\frac{n_{Corps}}{(1-p_Q)^2} - \frac{n_{Queue}}{(p_Q)^2} + \sum_{i=1}^{n} \frac{(F_C(t_i))^2}{(p_Q + (1-p_Q)(1-F_C(t_i)))^2} \tag{4.9}$$

Et avec Newton-Raphson :

$$p_Q^{(l+1)} = p_Q^{(l)} - \frac{f\left(p_Q^{(l)}\right)}{f'\left(p_Q^{(l)}\right)} \tag{4.10}$$

La première idée était de calculer l'estimateur initial pour p_Q tel que $\frac{n_{Queue}}{p_Q} - \frac{n_{Corps}}{1-p_Q} = 0$ soit

$$p_Q^{(0)} = \frac{n_{Queue}}{n_{Corps} + n_{Queue}}$$

Malheureusement, ça ne fonctionnait pas toujours en pratique. Si la valeur initiale est proche d'un minimum local, l'algorithme de Newton-Raphson n'atteint pas toujours la valeur souhaitée. Pour éviter ce problème, il est décidé de calculer la fonction avec 1000 valeurs différentes pour p_Q entre 0 et 1. Ensuite la valeur qui minimise la fonction en valeur absolue est prise comme valeur initiale pour p_Q.

Remarque

Cette valeur initiale est déjà l'estimateur du maximum de vraisemblance avec une précision au millième. Comme on ne cherche pas de précision plus grande, l'algorithme de Newton-Raphson devient inutile en pratique.

4.3 La seconde méthode est préférée

Les deux méthodes sont utilisées sur les données historiques d'un marché de responsabilité civile pour les comparer. La plus petite troncature à gauche est 611 687.

4.3.1 La loi de la queue de distribution

Pour les deux méthodes, la même loi pour la queue de distribution est estimée. C'est une loi de Pareto tronquée avec les paramètres suivants :

— $X_m = 8\ 000\ 000$
— $\alpha = 1,8$
— $T = 15\ 000\ 000$

4.3.2 La loi du corps de la distribution pour la première méthode

Avec la première méthode, la loi mixture d'Erlang avec les paramètres suivants est obtenue :

— $M = 5$
— $\theta = 322\ 699$
—

Erlang	α	r
1	0,845	2
2	0,084	8
3	0,026	11
4	0,030	18
5	0,015	33

Ainsi que les probabilités de jointure suivantes :

$\mathbb{P}(x \in Corps)$	0,984
$\mathbb{P}(x \in Queue)$	0,016

4.3.3 La loi du corps de la distribution pour la seconde méthode

Avec la seconde méthode, la loi mixture d'Erlang avec les paramètres suivants est obtenue :

— $M = 5$
— $\theta = 211\ 489$

Erlang	α	r
1	0,800	3
2	0,123	10
3	0,006	12
4	0,054	21
5	0,017	34

Et les probabilités de jointure suivantes :

$\mathbb{P}(x \in Corps)$	0,968
$\mathbb{P}(x \in Queue)$	0,032

4.3.4 La comparaison graphique

Le graphique suivant affiche les deux distributions finales pour les comparer.

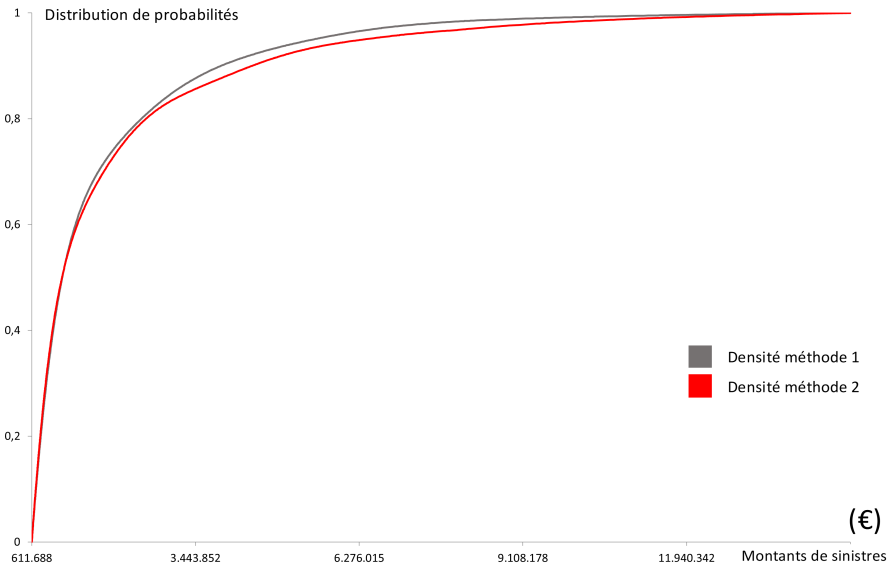

FIGURE 4.3 – La densité finale de la méthode 1 est proche de celle de la méthode 2

Sur ce graphique, les deux méthodes donnent des densités de probabilités plutôt proches. Il n'est pas possible de dire comme ça laquelle est la meilleure.

Cependant la première méthode a un défaut. Le calcul de la loi de probabilité du corps de la distribution peut être sévèrement impacté par des sinitres de la queue de distribution. Pour cette raison, **la seconde méthode** est préférée.

4.4 Conclusion

Dans ce chapitre, deux méthodes pour joindre les corps et queues de distributions ont été développées et comparées. Elles permettent de prendre en compte le caractère **aléatoire** des troncatures à gauche dans les données.

Pour finir, il faut tester ce modèle théorique sur de vraies données.

Chapitre 5

Tester le nouveau modèle sur des marchés réels

Le nouveau modèle semble bien fonctionner mais il reste à le tester et à le comparer avec le modèle de marché de QBE Re. Jusqu'à présent, le travail s'est concentré sur les fonctions de densité. Dans ce chapitre, les prix de contrats calculés avec ce nouveau modèle vont être étudiés. L'intérêt est de voir quel serait l'impact réel d'un changement du modèle sur les tarifications.

Pour cela, des données réelles sont utilisées. Il s'agit des historiques de sinistres pour trois marchés de responsabilités civiles. Dans un souci de confidentialité, leurs noms ne peuvent pas être révélés donc ils sont appelés responsabilité civile 1, 2 et 3. Pour chaque marché, le même seuil entre le corps et la queue de distribution que celui du modèle de QBE Re est sélectionné pour ne pas avoir de biais. Ces trois marchés permettent d'essayer le nouveau modèle avec beaucoup de points et avec peu de points.

5.1 Responsabilité civile 1

5.1.1 La modélisation du corps de la distribution

Le premier exemple est un marché bien connu de QBE Re et il y a beaucoup de données à disposition. Avec la troncature à gauche minimale et le seuil entre le corps et la queue de distribution retenus par le modèle de QBE Re, **1088 points** sont utilisables. Ce n'est pas beaucoup en statistiques mais déjà beaucoup en réassurance. Il n'y aura pas souvent plus de données.

Les paramètres estimés pour la loi mixture d'Erlang sont :

Erlang	α	r	θ
1	0,8712	2	302 270
2	0,0774	8	
3	0,0367	15	
4	0,0147	27	

avec ces troncatures (reprises du modèle de QBE Re) :

Troncature à gauche	Troncature à doite
611 687	9 975 033

Il n'est pas possible de savoir si cet estimateur est correct en regardant seulement les paramètres. La fonction de répartition de la loi mixture d'Erlang estimée et celle de la loi empirique du modèle de marché de QBE Re sont affichées sur un même graphique. Ainsi, il est plus facile de comparer le nouvel estimateur. Le graphique suivant compare donc ces deux lois :

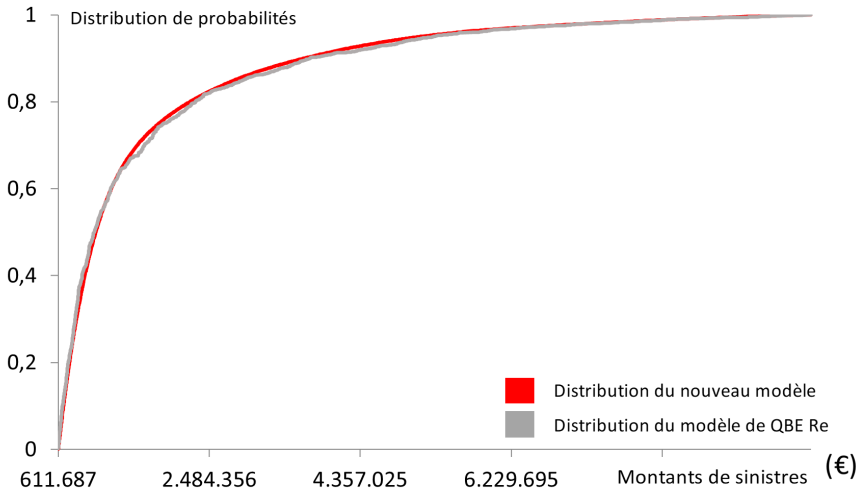

FIGURE 5.1 – La loi estimée concorde avec le modèle empirique de QBE Re pour le corps de la distribution

La loi mixture d'Erlang est une bonne estimation de la loi empirique du modèle de marché de QBE Re. En plus de cela, elle permet de lisser la distribution.

5.1.2 La modélisation de la queue de distribution

Pour commencer, il faut sélectionner une loi parmi la loi de Pareto, la loi exponentielle et la loi de Pareto tronquée. Les *Mean-Excess plot* correspondant à chacune des trois lois sont créés et la loi de Pareto tronquée peut être validée. Son *Mean-Excess plot* a une tendance linéaire décroissante claire et ceux des autres distributions ne les valident pas.

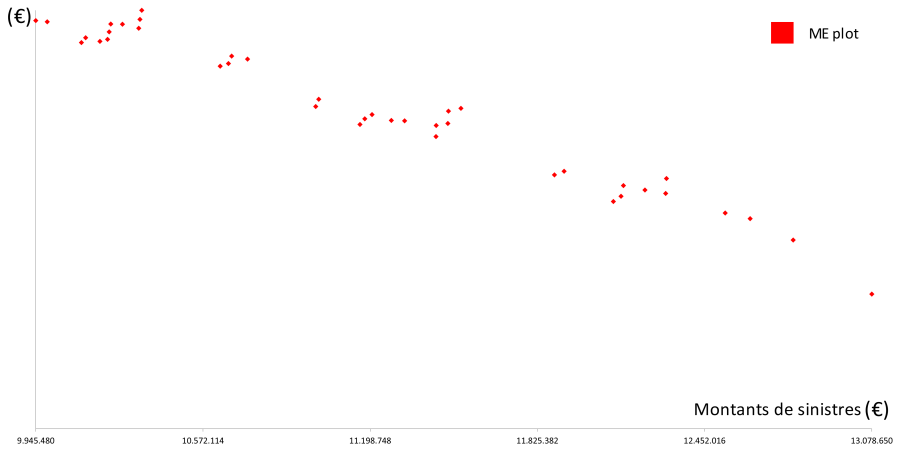

FIGURE 5.2 – Le *Mean-Excess plot* de loi de Pareto tronquée la valide

Les paramètres de la loi de Pareto tronquée doivent être estimés. On commence par s'occuper de la troncature à droite notée T en affichant le T-*plot*. La valeur la plus probable pour T se trouve au niveau du seuil entre le corps et la queue de distribution qui vaut 9 975 033. Le graphique suggère de choisir $T = 15\ 700\ 000$.

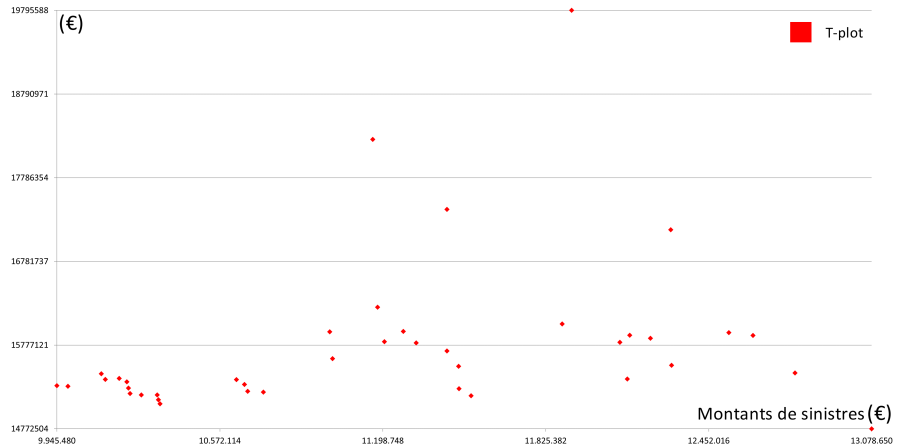

FIGURE 5.3 – Le graphique de T suggère de choisir $T = 15\ 700\ 000$

Pour finir, la valeur du paramètre de forme α est estimée. L'α-*plot* est créé en utilisant la valeur estimée pour T. D'après ce graphique, $\alpha = 4,6$ semble être une bonne estimation.

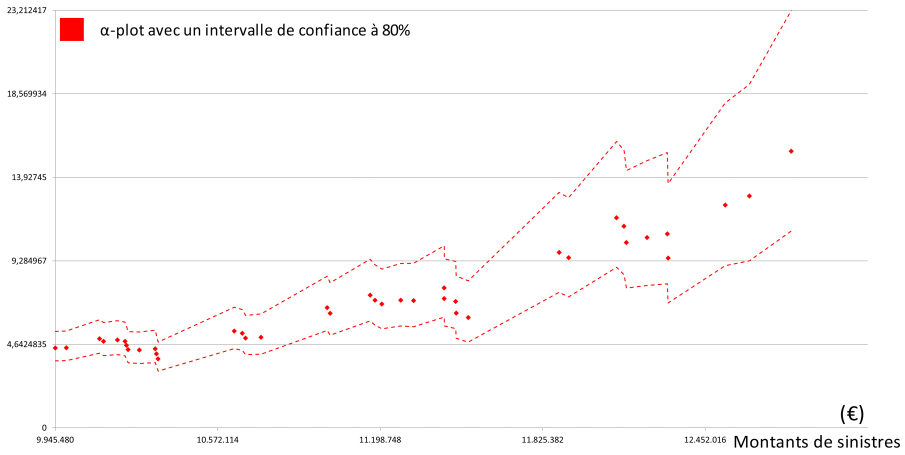

FIGURE 5.4 – L'*α-plot* suggère $\alpha = 4,6$

Pour résumer, la queue de distribution est modélisée avec une loi de Pareto tronquée dont les paramètres estimés sont :

X_m (Troncature à gauche)	α	T (Troncature à droite)
9 975 033	4,6	15 700 000

Pour finir, l'hypothèse d'une troncature à droite des données est testée en affichant la *p-value*. Elle reste principalement en dessous du seuil de 5% ce qui permet de valider l'hypothèse.

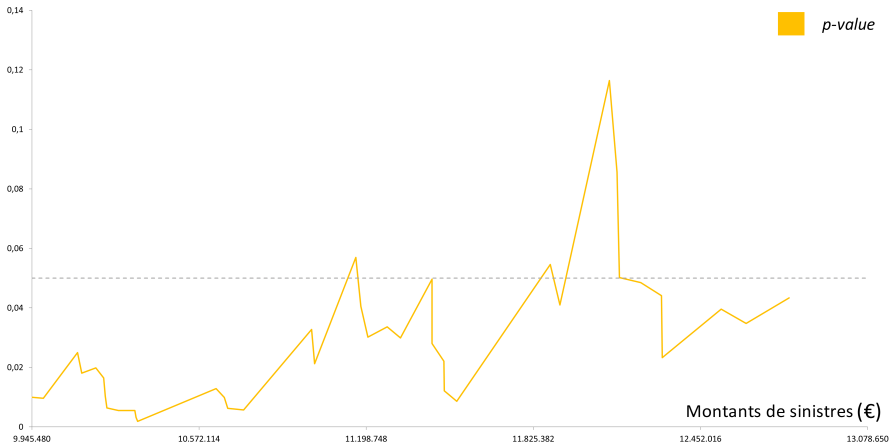

FIGURE 5.5 – La *p-value* valide l'hypothèse d'une troncature à droite dans les données

La distribution pour la queue de distribution retenue pour le modèle de marché de QBE Re était une loi de Pareto généralisée (GPD) avec les paramètres suivants :

μ (localisation)	σ (échelle)	ξ (forme)
9 975 033	2 750 000	-0,5

5.1.3 L'estimation des probabilités de jointure

Une loi pour le corps de la distribution et une loi pour la queue de la distribution ont été modélisées. Il faut estimer les probabilités d'être dans l'une ou l'autre des deux parties pour obtenir une loi finale pour le marché de la responsabilité civile 1.

Il suffit d'exécuter les estimateurs construits et les probabilités suivantes sont obtenues :

Probabilité d'être dans le corps	Probabilité d'être dans la queue
0,9832	0,0168

Le modèle de QBE Re a retenu 0,9815 comme probabilité d'être dans le corps de la distribution. Le nouveau modèle estime donc un tout petit peu moins de sinistres extrêmes. Finalement, cette distribution est obtenue pour le marché complet :

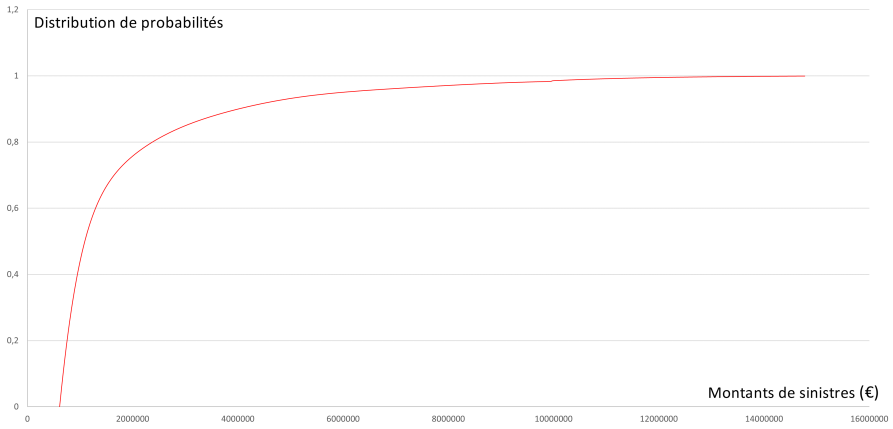

FIGURE 5.6 – La distribution de probabilité de la sévérité pour le marché de responsabilité civile 1

5.1.4 La comparaison des prix obtenus

C'est bien d'avoir une loi qui couvre la sévérité sur tout le marché mais l'objectif de l'entreprise est de calculer des prix. C'est pourquoi les prix obtenus par le nouveau modèle sont comparés avec ceux du modèle actuel de QBE Re.

Des contrats en exécédent de sinistres ont été choisis. Dans ce type de contrat, le réassureur paie la partie qui dépasse la priorité pour chaque sinistre. On pose P, la priorité et x la valeur d'un sinistre. Si x est plus petit que P, le réassureur ne paie rien et si x est plus grand que P, le réassureur paie $x - P$. Cela peut se résumer en : le réassureur paie $max(0, x - P)$ pour chaque sinistre. Souvent il y a une limite au remboursement appelée portée mais on se place dans le cas où il n'y en a pas pour rester simple.

Pour calculer l'espérance de $max(0, x - P)$, un échantillon de sinistres est simulé et la formule est appliquée à chacun. La moyenne donne ensuite l'espérance. Pour comparer, 100 000 sinistres sont simulés avec le nouveau modèle et 100 000 avec le modèle de marché de QBE Re puis les différents traités (contrats) sont appliqués.

Une première comparaison

Pour pouvoir comparer des quantités équivalentes, l'espérance des coûts de sinistres est calculée pour une priorité donnée sachant qu'ils ont atteint cette priorité. Donc l'espérance est divisée par la probabilité d'être plus grand que la priorité. Formellement, cela s'écrit :

$$\begin{aligned}\mathbb{E}(max(X-P,0)|X>P) &= \int_{\mathbb{R}^+} max(x-P,0)f(x|x>P)dx \\ &= \int_{\mathbb{R}^+} max(x-P,0)\frac{f(x)}{1-F(P)}dx \\ &= \mathbb{E}(max(X-P,0)) * \frac{1}{1-F(P)}\end{aligned} \tag{5.1}$$

La probabilité d'atteindre une priorité est le nombre de sinistres supérieurs à cette priorité divisé par le nombre de sinistres total dans l'échantillon. Des priorités allant de 250 000 à 10 000 000 avec un pas de 250 000 ont été choisies. Le graphique suivant donne les prix pour chaque traité avec le nouveau modèle et avec le modèle de QBE Re.

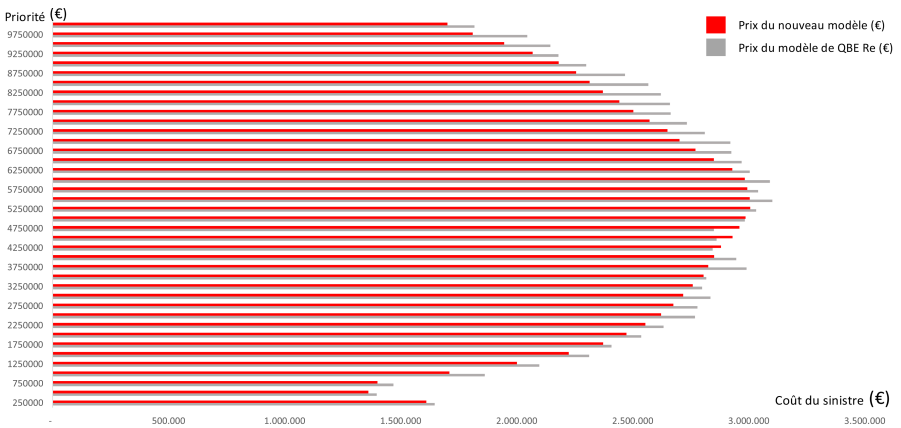

FIGURE 5.7 – Le nouveau modèle estime des prix un petit peu plus faibles que le modèle de QBE Re

Les prix sont très proches mais le nouveau modèle estime des prix un peu plus faibles que ceux du modèle de marché. Pour mieux comparer, on s'intéresse à la différence en pourcentage :

FIGURE 5.8 – La différence de prix reste dans le domaine de l'erreur statistique

Sur ce graphique, la différence est acceptable. Cependant elle est principalement négative. Cela signifie que les prix sont généralement inférieurs à ceux du modèle de marché actuel. Pour affiner la comparaison, la volatilité est réduite. Pour cela, les espérances par contrats sachant que les sinistres sont supérieurs **au même** seuil sont calculées.

La même troncature basse pour les sinistres

Dans cette partie, les prix espérés des sinistres dans chaque contrat sachant que les sinistres sont supérieurs ou égaux à 750 000 sont calculés. Le graphique suivant, de comparaison des prix issus de mon modèle par rapport à ceux issus du modèle de marché de QBE Re, est obtenu.

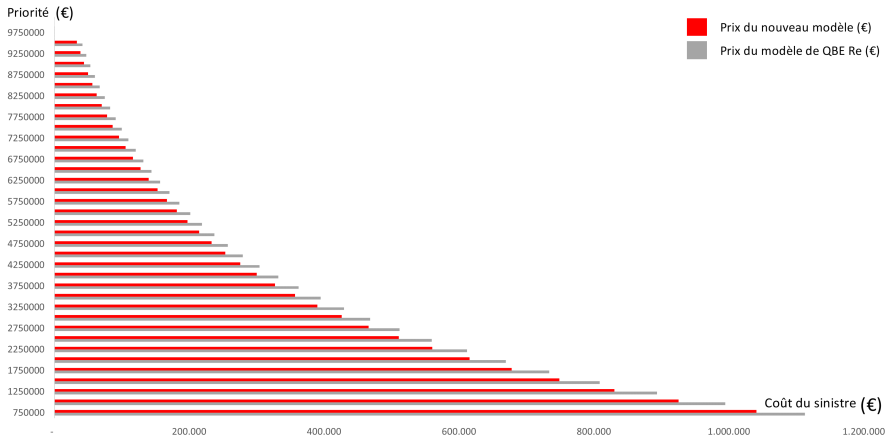

FIGURE 5.9 – Les prix semblent proches

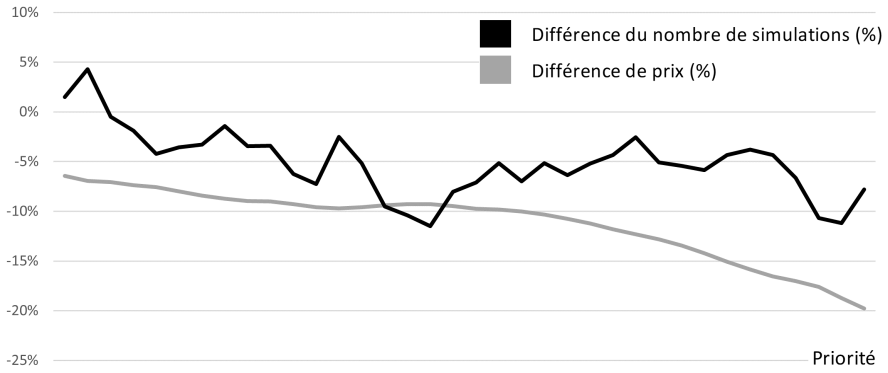

FIGURE 5.10 – La différence de prix reste importante

Avec cette deuxième comparaison, la différence de prix est réduite et lissée. La différence dans le nombre de sinistres simulés pour le calcul de chaque traité (en noir) montre qu'une partie de la différence de prix peut venir directement de l'incertitude statistique. Cependant, il y a toujours une différence notable dans le prix, surtout pour les contrats avec les plus grandes priorités. Il semble que le nouveau modèle sous-estime les plus grands sinistres. Cela peut venir de la loi de la queue de distribution. Pour contrôler ça, la même comparaison est faite mais en utilisant la loi du modèle de marché pour la queue de distribution dans le nouveau modèle.

La même queue de distribution

Ici est réalisée la même comparaison que précédemment mais en remplaçant la queue de distribution du nouveau modèle par celle du modèle de marché de QBE Re. Pour rappel, il s'agit d'une loi de Pareto généralisée de paramètres :

μ (localisation)	σ (échelle)	ξ (forme)
9 975 033	2 750 000	-0,5

Ainsi, le graphique suivant est obtenu :

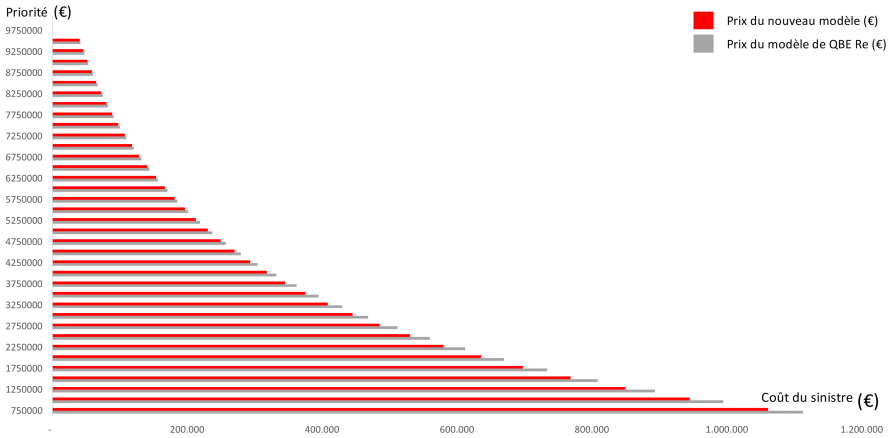

FIGURE 5.11 – Les prix sont plus proches

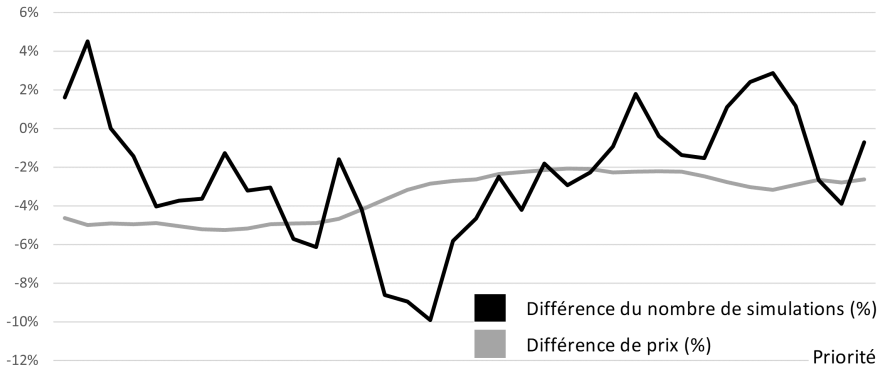

FIGURE 5.12 – La différence de prix est réduite

La différence de prix est déjà plus faible. Cela confirme qu'une partie de la différence vient de la queue de distribution. Cela n'est pas étonnant car cet estimateur est très volatile. Le dernier paramètre qui pourrait entrainer une différence est la probabilité utilisée pour recoller le corps et la queue de distribution. En effet, une probabilité plus faible d'être dans la queue de distribution a été obtenue. Cela peut expliquer une sous-estimation des prix. Une dernière comparaison est effectuée pour vérifier cela.

Les mêmes probabilités de jointure

Dans cette dernière partie, la même comparaison que la précédente est réalisée mais le nouveau modèle utilise les probabilités de jointure du modèle de marché de QBE Re. Il en découle les graphiques suivants :

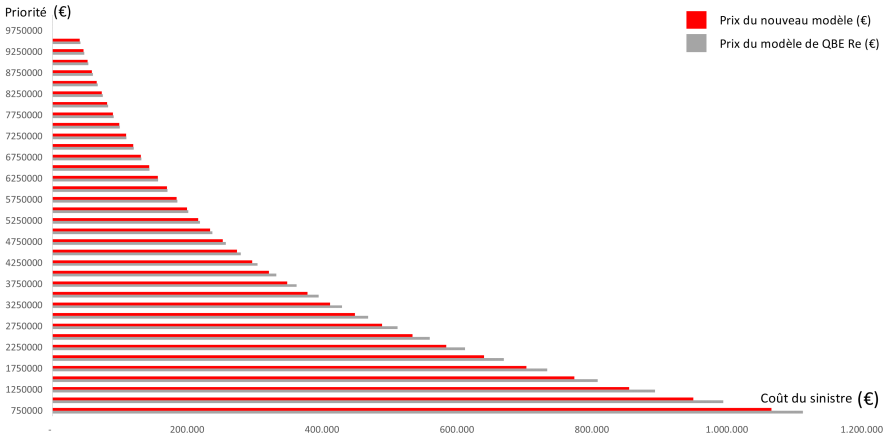

FIGURE 5.13 – Les prix sont plus proches

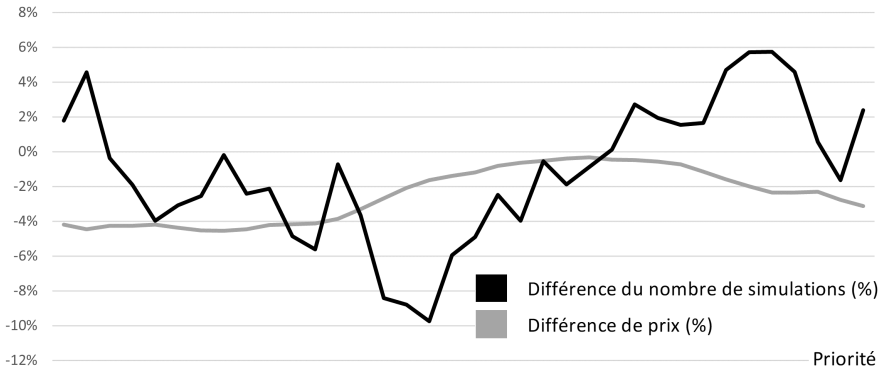

FIGURE 5.14 – La différence de prix est légèrement réduite

La différence de prix est légèrement réduite par rapport au cas précédent. La différence restante ne peut venir que de la loi mixture d'Erlang. En effet, si on regarde la fonction de densité du corps de la distribution, la loi mixture d'Erlang est légèrement au-dessus de la loi empirique du modèle de marché. Cela peut expliquer la différence restante. Cependant, cette différence est faible et pourrait aussi être attribuée en partie à l'erreur statistique.

5.2 Responsabilité civile 2

5.2.1 La modélisation du corps de la distribution

Ce deuxième exemple utilise le même marché de responsabilité civile que l'exemple précédent mais dans un autre pays. Ce devrait être un marché plutôt similaire au précédent mais la troncature à gauche et le seuil entre le corps et la queue de distribution retenus pour le modèle de QBE Re sont très différents de l'exemple précédent. **1334 points** sont quand même utilisables. C'est légèrement plus que pour la responsabilité civile 1.

Le nouvel estimateur trouve les paramètres suivants pour la loi mixture d'Erlang :

Erlang	α	r	θ
1	0,8068	3	307 372
2	0,1588	9	
3	0,0162	14	
4	0,0110	28	
5	0,0073	47	

Avec ces troncatures :

Troncature à gauche	Troncature à droite
1 005 028	18 443 938

Pour ce marché aussi, une comparaison graphique de la fonction de répartition paramétrique et de la fonction de répartition empirique du modèle de QBE Re est affichée. Le nouvel estimateur représente bien la fonction du modèle de QBE Re avec l'avantage de la lisser.

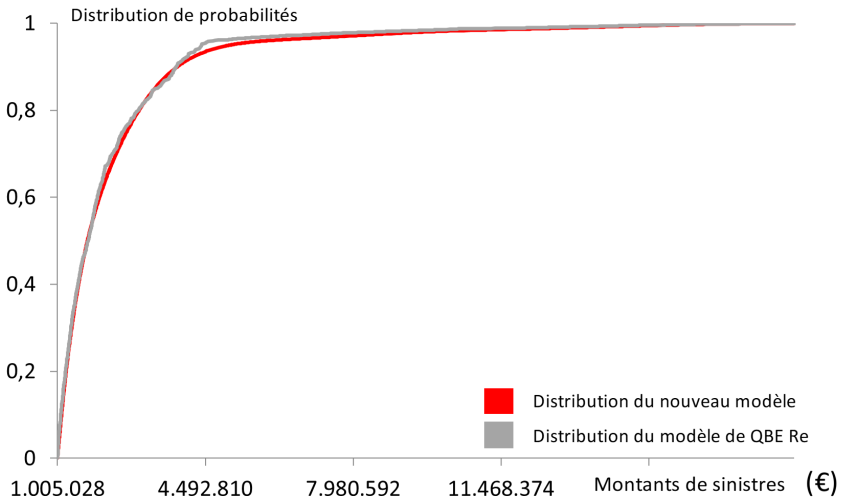

FIGURE 5.15 – La loi mixture d'Erlang est une bonne approximation du modèle de marché actuel

5.2.2 La modélisation de la queue de la distribution

Ici, la queue de la distribution est modélisée. D'après les *Mean-Excess plot*, une loi de Pareto peut être utilisée.

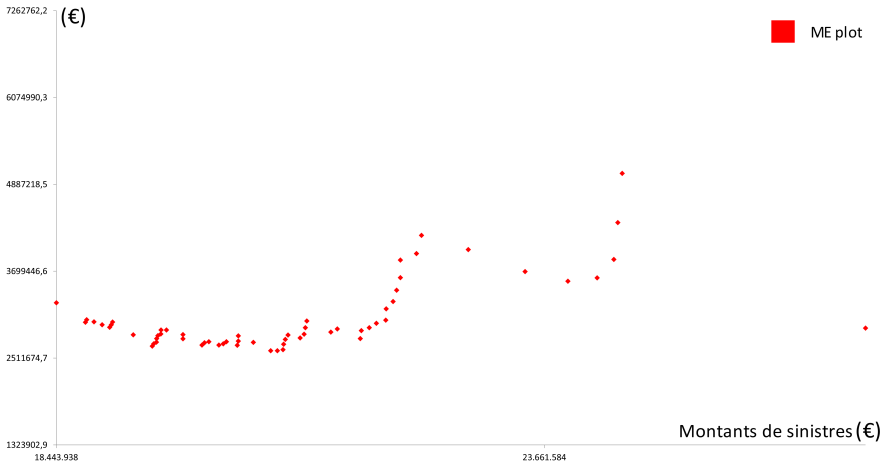

FIGURE 5.16 – Le *Mean-Excess plot* de la loi de Pareto la valide

La tendance linéaire croissante n'est pas parfaite mais les autres *Mean-Excess plot* ont rejeté les deux autres lois sans laisser de doute. Il faut maintenant modéliser la loi de Pareto. Pour cela l'α-*plot* est utilisé afin de trouver une valeur pour le paramètre de forme α.

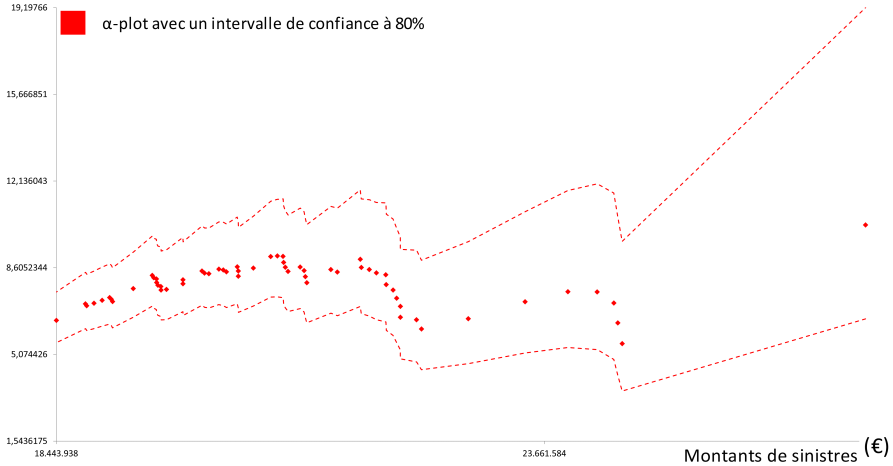

FIGURE 5.17 – L'*α-plot* suggère $\alpha = 6,5$

Pour résumer, une loi de Pareto a été sélectionnée avec les paramètres :

X_m (Troncature à gauche)	α
18 443 938	6,5

Le modèle de marché de QBE Re a retenu une loi très similaire pour la queue de distribution. Il s'agit aussi d'une loi de Pareto avec les paramètres suivants :

X_m (Troncature à gauche)	α
18 443 938	6,7

5.2.3 L'estimation des probabilités de jointure

Après avoir modélisé le corps de la distribution ainsi que la queue de distribution, les probabilités d'être dans l'un ou l'autre doivent être calculées pour avoir une loi sur l'ensemble des sinistres.

Le nouvel estimateur est appliqué et trouve les probabilités suivantes :

Probabilité d'être dans le corps	Probabilité d'être dans la queue
0,9956	0,0044

Le modèle de marché actuel de QBE Re suggère 0,9962 de probabilité d'être dans le corps de la distribution. Cette fois-ci, le nouveau modèle estime un peu moins de probabilité d'être dans le corps de la distribution. Finalement la distribution complète suivante est obtenue pour la sévérité :

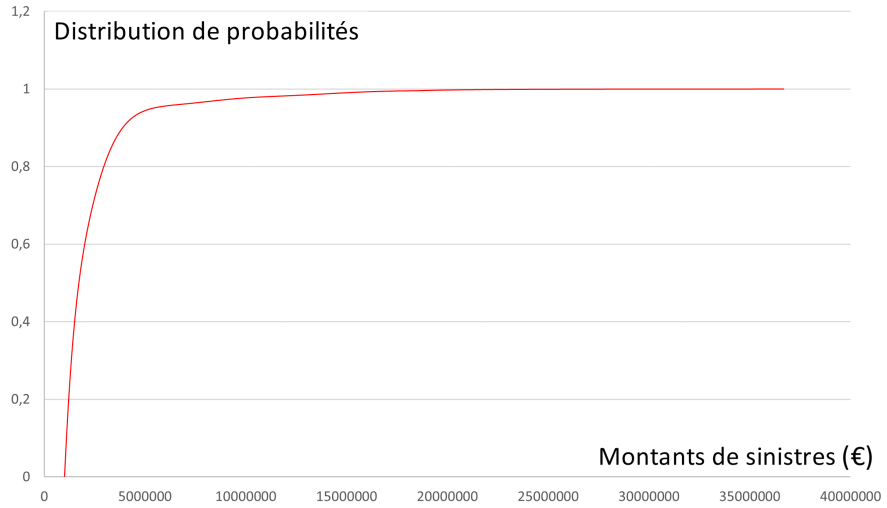

FIGURE 5.18 – La distribution de probabilité de la sévérité pour la responsabilité civile 2 estimée avec le nouveau modèle

5.2.4 La comparaison des prix obtenus

Avec cette loi de probabilité, des montants de sinistres peuvent être simulés pour calculer des prix de contrats. Les prix sont comparés avec ceux obtenus par le modèle de QBE Re pour apprécier l'impact du nouveau modèle sur la tarification.

Comme pour la responsabilité civile 1, des traités en excédent de sinistres sont calculés. C'est-à-dire que l'espérance de $max(0, x - P)$ est calculée. Les mêmes priorités entre 250 000 et 10 000 000 avec un pas de 250 000 sont gardées.

Une première comparaison

Aussi pour toujours comparer des quantités qui correspondent, l'espérance de prix est calculée pour une priorité donnée sachant que le sinistre atteint cette priorité.

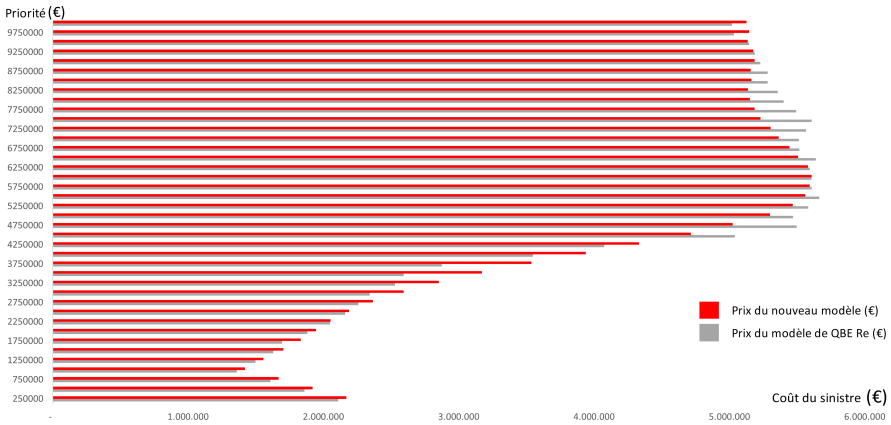

FIGURE 5.19 – Les nouveaux prix estimés restent dans la tendance de ceux du modèle de QBE Re

En analysant ce graphique, on remarque que les prix sont surestimés pour les petites priorités et sous-estimés pour les plus grandes priorités par rapport au modèle de QBE Re. Néanmoins, ils restent globalement proches.

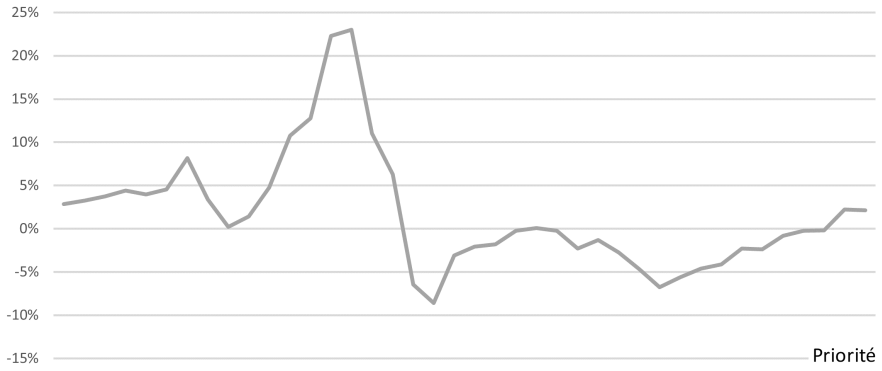

FIGURE 5.20 – La différence de prix reste acceptable du point de vue de l'erreur statistique

Ce graphique met en valeur l'analyse. La différence de prix est positive au début puis négative ensuite. Comme pour l'exemple précédent, la volatilité dans la différence est réduite en comparant les prix par priorité sachant que les sinistres ont atteint le même seuil commun.

La même troncature basse pour les sinistres

Dans cette partie, les prix sachant que les sinistres ont atteint 750 000 sont calculés. Le graphique suivant est obtenu :

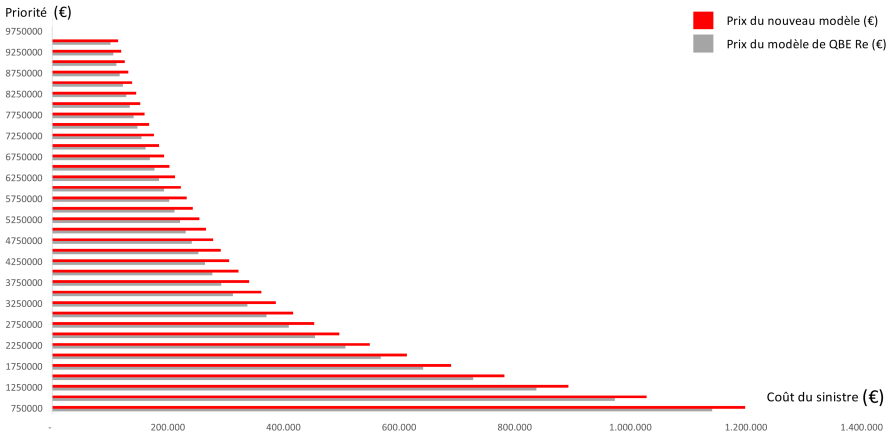

FIGURE 5.21 – Le nouveau modèle estime des prix plus importants

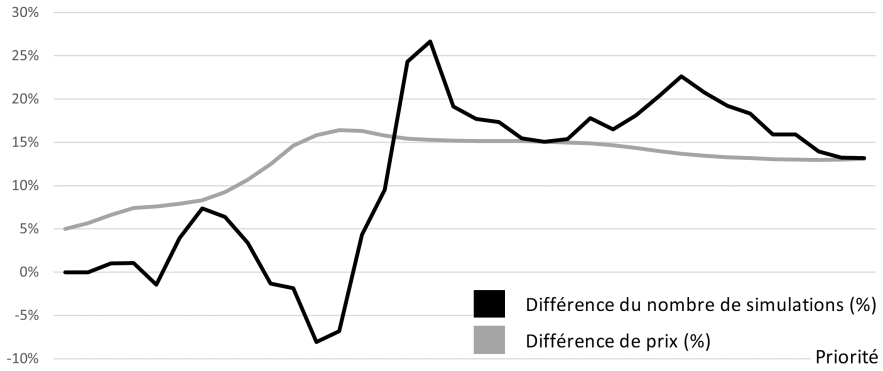

FIGURE 5.22 – La différence de prix reste plus petite que la différence du nombre de sinistres simulés par traité

Avec cette comparaison, la différence de prix est déjà réduite et lissée. La différence dans le nombre de sinistres simulés pour chaque priorité (noir) montre qu'une partie de la différence de prix vient de l'incertitude statistique. Malgré cela, il y a toujours une différence dans la tarification. Le nouveau modèle surestime les prix jusqu'à 15% par rapport au modèle de QBE Re. Pour vérifier si la différence peut venir de la queue de distribution, cette simulation est refaite en prenant la loi du modèle de marché actuel pour la queue de distribution dans le nouveau modèle.

La même queue de distribution

La même comparaison est refaite en utilisant la loi du modèle de marché de QBE Re pour la queue de distribution dans le nouveau modèle. Pour rappel, il s'agit d'une loi de Pareto dont les paramètres sont les suivants :

X_m (Troncature à gauche)	α
18 443 938	6,7

Ainsi les graphiques suivants sont obtenus :

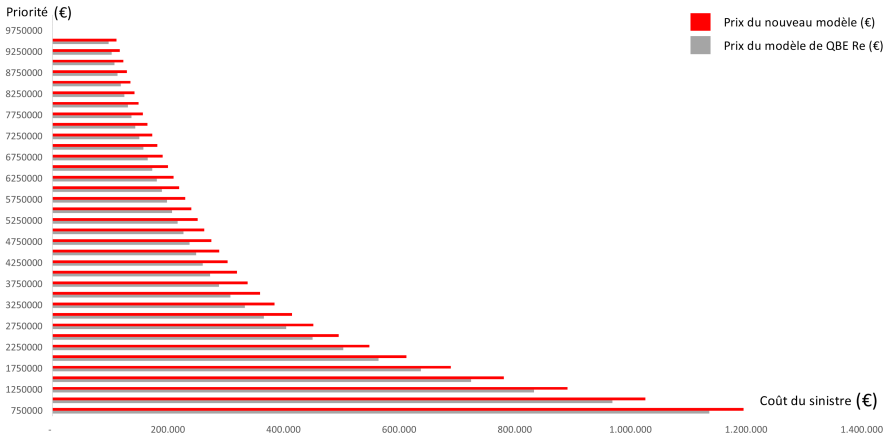

FIGURE 5.23 – Les nouveaux prix estimés ne sont pas plus proches

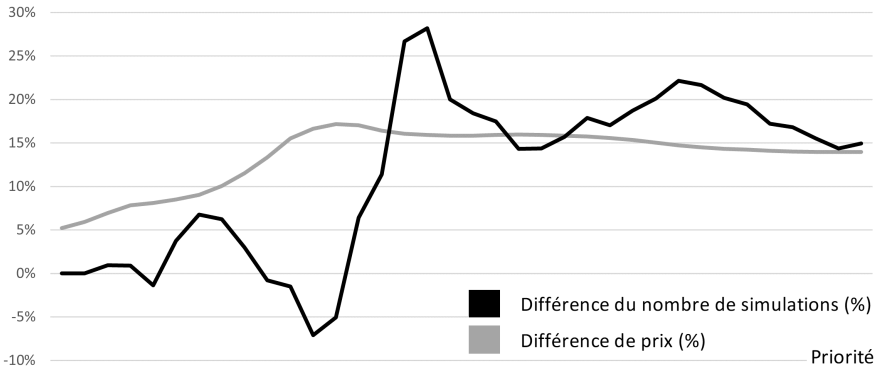

FIGURE 5.24 – La différence de prix reste sensiblement la même

La différence dans les prix n'est pas réduite. Elle est même très légèrement plus grande. Ce n'est pas surprenant étant donné que les deux lois de queues de distribution étaient déjà très similaires.

Dans cette exemple, la différence restante ne peut pas venir non plus des probabilités de jointure puisqu'elles sont déjà très proches. Pour comprendre la différence, on peut regarder la courbe du corps de la distribution. La loi mixture d'Erlang est un peu en dessous de la loi du modèle de marché. Il est donc normal de trouver des prix légèrement plus élevés.

5.3 Responsabilité civile 3

5.3.1 La modélisation du corps de la distribution

Pour finir, le nouveau modèle est testé pour le marché de la responsabilité civile 3. Avec la troncature à gauche et le seuil entre le corps et la queue de distribution retenus par le modèle de QBE Re, seulement **58 points** peuvent être utilisés. C'est intéressant de voir comment se comporte le modèle avec peu de points.

Les paramètres estimés pour la loi mixture d'Erlang sont :

Erlang	α	r	θ
1	0,8306	5	147 392
2	0,1333	13	
3	0,0361	29	

avec ces troncatures :

Troncature à gauche	Troncature à droite
600 132	5 617 229

Comme pour les deux exemples précédents, les fonctions de répartition du nouveau modèle et du modèle de QBE Re sont affichées sur un même graphique.

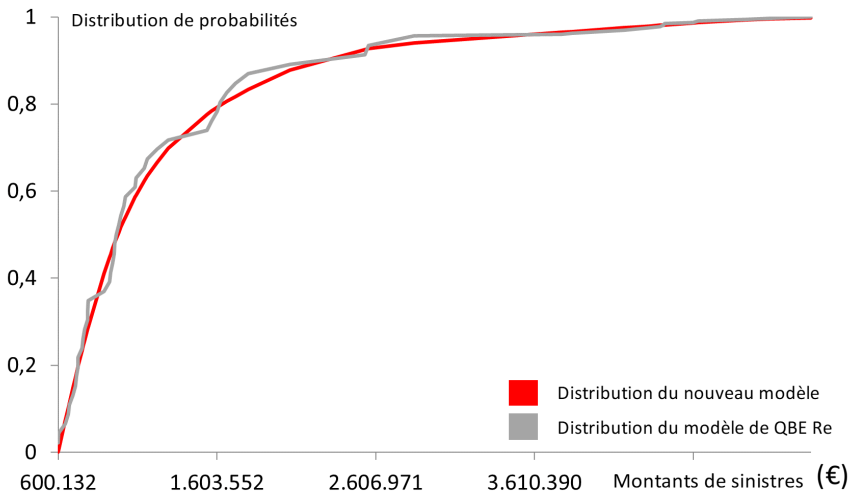

FIGURE 5.25 – La loi mixture d'Erlang est une bonne approximation du modèle empirique de l'entreprise

A première vue, la loi mixture d'Erlang ne colle pas aussi bien que dans les exemples précédents. Cependant, le modèle empirique de QBE Re est réalisé avec seulement 58 points. Cela veut dire que la courbe finale est sévèrement impactée par chaque point de l'ensemble. On peut observer ces impacts sur la courbe grise. Elle fait du surapprentissage des données (*overfitting*). Dans ce cas précis, la robustesse du nouvel estimateur qui évite ce surapprentissage est démontrée. Il permet de trouver une loi potentiellement plus correcte pour représenter les données. Il passe parfaitement au milieu de la fonction empirique sans être lourdement impacté par un unique point.

5.3.2 La modélisation de la queue de la distribution

Le *Mean-Excess plot* suggère de choisir une loi de Pareto tronquée pour représenter la queue de la distribution. Il a une tendance linéaire décroissante et les autres *Mean-Excess plot* invalident les autres lois. La tendance n'est pas facile à observer comme il n'y a que 11 points mais elle est bien là.

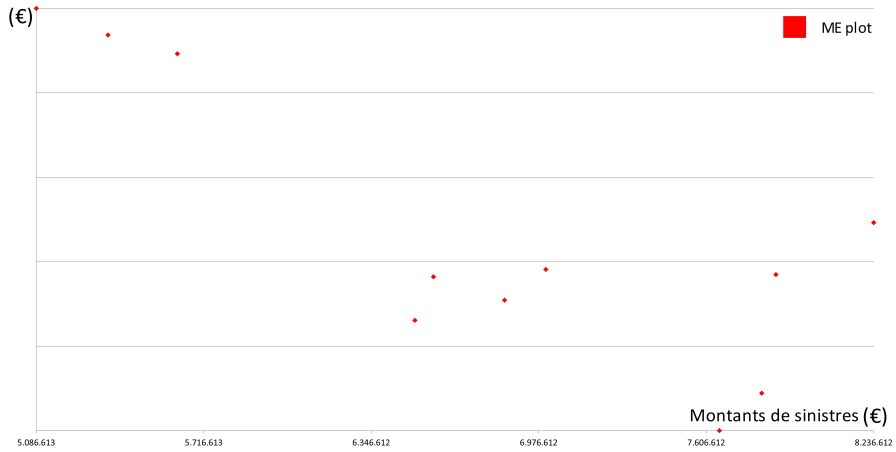

FIGURE 5.26 – Le *Mean-Excess plot* de la loi de Pareto tronquée la valide

Les paramètres de la loi de Pareto tronquée doivent être estimés. On commence par estimer la troncature à droite T. Le T-*plot* est affiché et la valeur potentielle de la troncature T est attendue autour du seuil entre le corps et la queue de distribution (5 617 229). Le graphique suggère $T = 16$ 000 000.

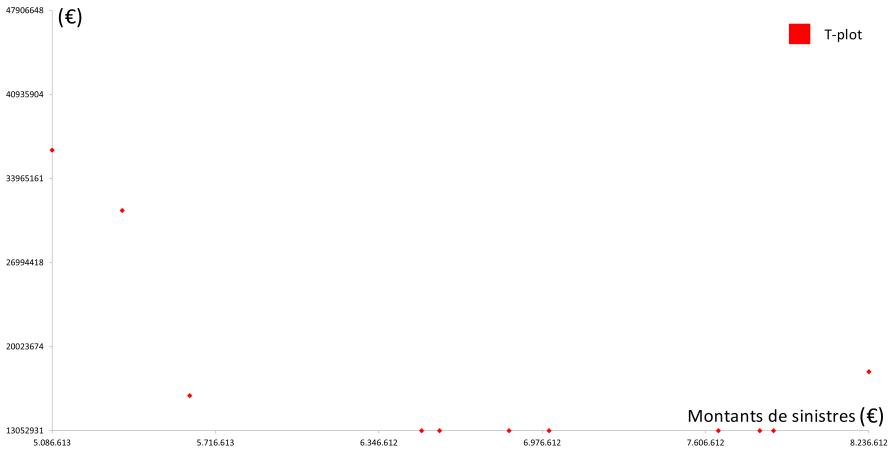

FIGURE 5.27 – Le *T-plot* suggère $T = 16$ 000 000

Le second paramètre à estimer est le paramètre de forme α. L'α-*plot* calculé avec la valeur estimée pour T (comme les deux paramètres sont interdépendants) est tracé. $\alpha = 2,7$ est estimé.

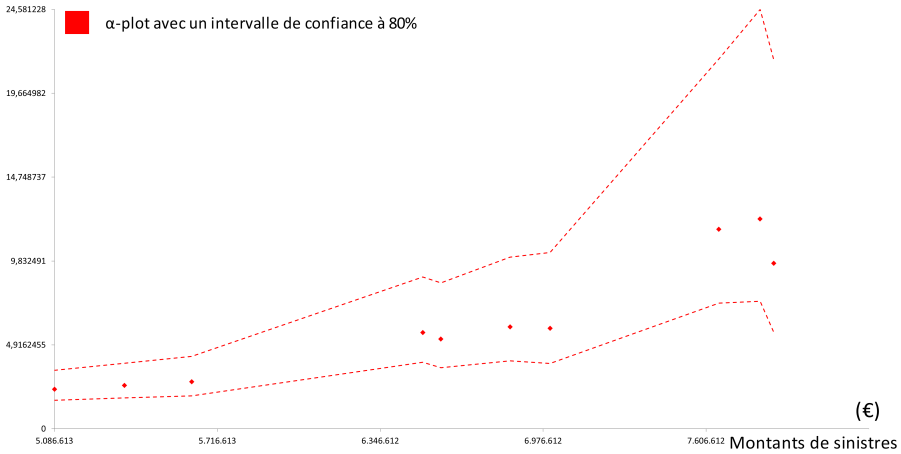

FIGURE 5.28 – L'α-*plot* suggère $\alpha = 2,7$

Pour résumer, la loi de Pareto tronquée est utilisée pour modéliser la queue de distribution. Ses paramètres sont les suivants :

X_m (Troncature à gauche)	α	T (Troncature à droite)
5 617 229	2,7	16 000 000

Pour finir, l'hypothèse d'une troncature des données est testée en affichant la *p-value*. Elle est plutôt élevée pour les dernière valeurs mais, est en dessous du seuil de 5% pour les premières valeurs.

Elle permet de valider l'hypothèse d'une troncature des données.

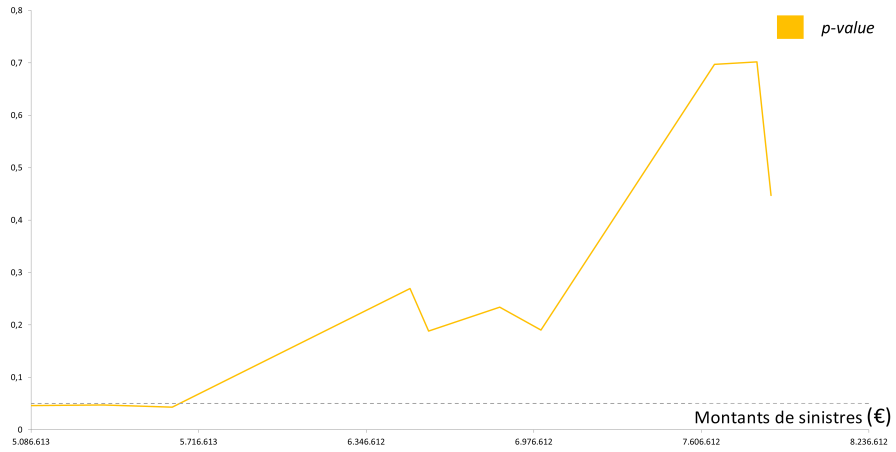

FIGURE 5.29 – La *p-value* valide l'hypothèse d'une troncature à droite

Pour ce marché, le modèle de QBE Re a retenu une distribution de Pareto généralisée (GPD) avec les paramètres suivants :

μ (localisation)	σ (échelle)	ξ (forme)
5 617 229	3 600 000	-0,34

5.3.3 L'estimation des probabilités de jointure

Dans cette partie, les probabilités d'être dans le corps ou dans la queue de la distribution sont estimées pour obtenir une loi sur l'ensemble des sinistres.

Le nouvel estimateur est exécuté et obtient les probabilités suivantes :

Probabilité d'être dans le corps	Probabilité d'être dans la queue
0,9716	0,0284

Le modèle de marché actuel suggère 0,9615 comme probabilité d'être dans le corps de la distribution. Encore une fois, le nouveau modèle estime moins de sinistres dans la queue de distribution. Cette distribution finale est obtenue :

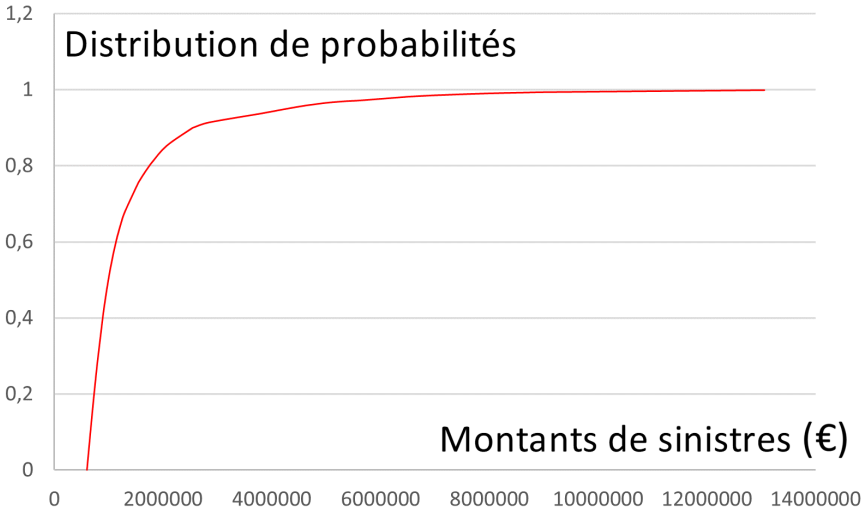

FIGURE 5.30 – La distribution de probabilité de la sévérité pour la responsabilité civile 3

5.3.4 La comparaison des prix obtenus

Comme dans les deux premiers exemples, des prix de traités sont calculés pour mesurer l'impact réel du nouveau modèle.

Il est choisi de tarifer les mêmes contrats que pour les exemples précédents. Ainsi, 100 000 sinistres sont simulés avec le nouveau modèle et 100 000 avec le modèle de l'entreprise puis la formule $max(0, x - P)$ leur est appliquée.

Une première comparaison

Pour la première comparaison, les valeurs espérées de sinistres par priorité sont calculées sachant qu'ils atteignent la priorité.

Des priorités allant de 250 000 à 10 000 000 avec un pas de 250 000 ont été choisies. Le graphique suivant est obtenu :

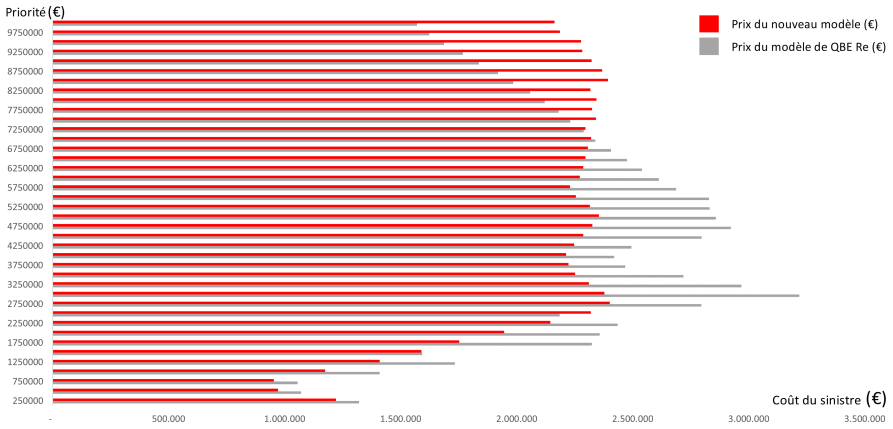

FIGURE 5.31 – Les nouveaux prix sont plutôt différents de ceux du modèle de marché actuel

Par rapport aux deux autres exemples, le résultat ne paraît pas très bien. Pour les premières priorités, les deux modèles donnent des prix similaires mais ensuite ils sont très différents. On peut s'intéresser à la différence en pourcentage.

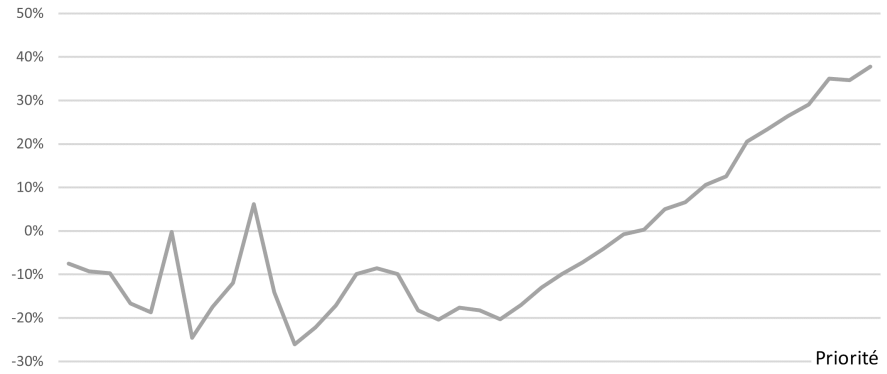

FIGURE 5.32 – La différence de prix est plus grande que ce à quoi on pourrait s'attendre

Sur ce graphique, au début, la différence de prix est acceptable mais ensuite elle augmente très franchement avec une tendance linéaire et atteint presque + 40%. Il faut déterminer d'où vient cette différence. D'abord la volatilité peut être réduite en comparant les sinistres sachant qu'ils ont atteint un seuil commun.

La même troncature basse pour les sinistres

Dans cette partie, les prix des contrats en fonction des différentes priorités sont calculés sachant que les sinistres sont supérieurs ou égaux à 750 000. Cela permet de réduire la volatilité venant de la différence du nombre de sinistres simulés pour chaque priorité. Le graphique suivant est obtenu :

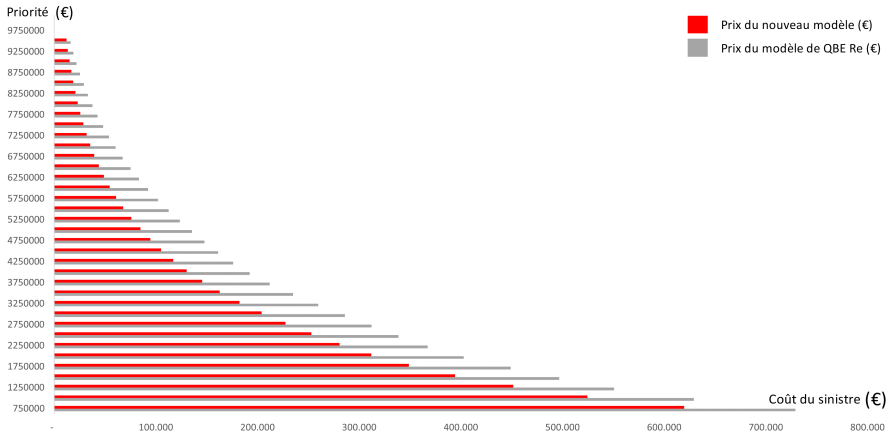

FIGURE 5.33 – Le nouveau modèle estime des prix plus bas que le modèle de marché de QBE Re

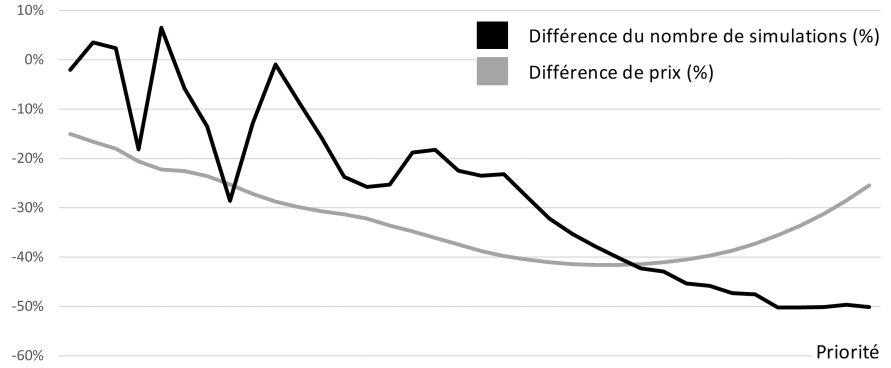

FIGURE 5.34 – La différence de prix est toujours négative

Avec cette comparaison, la différence de prix est lissée mais pas réduite. La différence du nombre de simulations (noir) est importante pour les priorités les plus hautes et montre ainsi que la différence dans la tarification pourrait venir des sinistres extrêmes. La raison la plus probable est que le nouveau modèle simule moins de grands sinistres comme sa probabilité d'être dans la queue de distribution est plus faible. Cependant, on va commencer par vérifier que la différence ne vient pas de la queue de distribution en comparant les deux modèles avec la queue de distribution du modèle de QBE Re.

La même queue de distribution

La même comparaison est refaite en remplaçant la loi de la queue de distribution du nouveau modèle par celle du modèle de QBE Re. Pour rappel, il s'agit d'une loi de Pareto généralisée (GPD) de paramètres suivants :

μ (localisation)	σ (échelle)	ξ (forme)
5 617 229	3 600 000	-0,34

Ainsi les graphiques suivants sont obtenus :

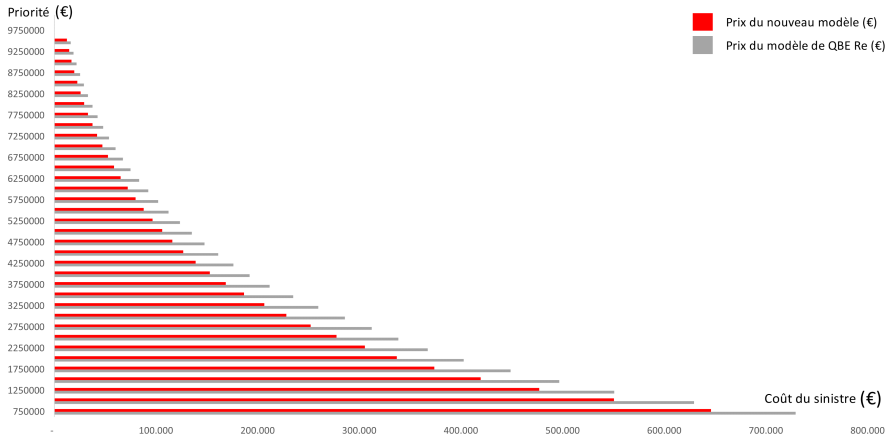

FIGURE 5.35 – Les nouveaux prix estimés sont légèrement plus proches de ceux du modèle de QBE Re

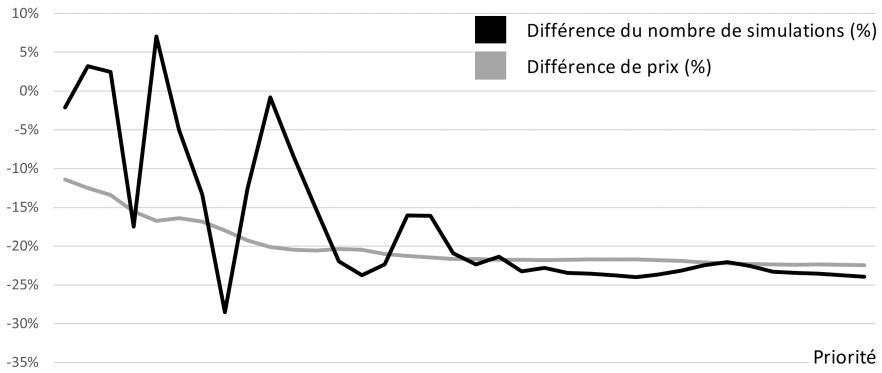

FIGURE 5.36 – La différence de prix est légèrement réduite

La différence de prix est un peu plus faible (en valeur absolue). Cela confirme qu'une part de la différence de prix vient de la queue de distribution. Néanmoins, il y a toujours une différence importante dans les prix et elle vient probablement des probabilités de jointure. C'est ce qui est vérifié avec une dernière comparaison.

Les mêmes probabilités de jointure

Pour finir, la même comparaison que précédemment est refaite en utilisant les probabilités du modèle de QBE Re cette fois-ci. Cette comparaison donne les graphiques suivants :

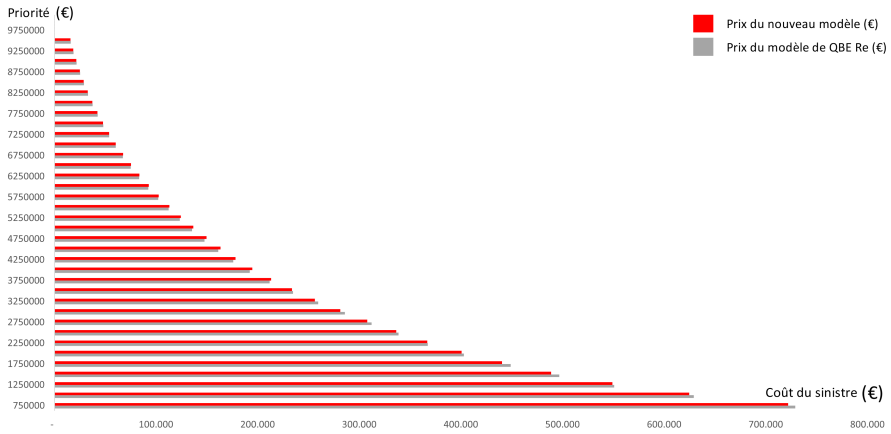

FIGURE 5.37 – Les prix sont presque égaux

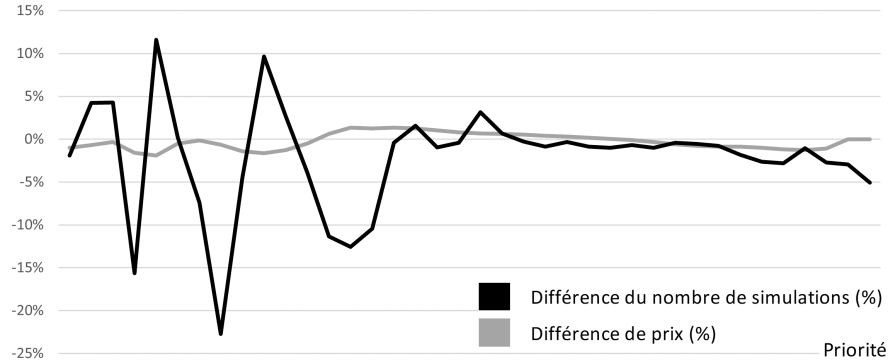

FIGURE 5.38 – La différence de prix reste presque tout le temps égale à zéro

Cette dernière comparaison est excellente. Les prix sont presque égaux et le graphique de la différence le confirme. Cela montre que le nouveau modèle pour le corps de la distribution est très bon même avec très peu de données.

Dans les autres comparaisons, la différence venait principalement des probabilités de jointure. En réalité, il n'est pas facile de dire quel modèle a les meilleures probabilités de jointure et donc les prix les plus justes. Mais il est certain que le nouvel estimateur pour le corps de la distribution n'est pas moins bien que celui du modèle de marché actuel. C'est un point très positif.

Conclusion

L'objectif de ce mémoire était de trouver un modèle de distribution de la sévérité utilisable dans un contexte de forte inflation. Pour cela, il fallait qu'il puisse prendre en compte des données tronquées aléatoirement. Le modèle original se divise en trois grandes parties qui sont la modélisation d'un corps de distribution, la modélisation d'une queue de distribution et l'estimation de probabilités pour joindre ces deux parties. De nouveaux estimateurs pour chacune de ces trois parties ont été proposés.

Au cours des comparaisons, il a pu être observé que les prix obtenus par le nouveau modèle étaient relativement proches de ceux issus du modèle de l'entreprise. Il y a quelques différences mais il n'est pas facile de dire quel modèle obtient le prix le plus juste. Si on ne veut pas changer tout le modèle, il serait quand même possible de changer certaines parties. L'estimateur du corps de la distribution donne des résulats très proches de ceux du modèle de QBE Re. Il serait envisageable de le remplacer et ce serait une amélioration significative car on remplacerait une distribution empirique par une distribution paramétrique. De plus le nouvel estimateur de la queue de distribution conserve l'information du modèle actuel et la complète. On pourrait aussi remplacer cet estimateur sans risque pour le modèle.

Bibliographie

Albrecher, H., Beirlant, J. & Teugels, J. L. (2017), *Reinsurance : actuarial and statistical aspects*, John Wiley & Sons.

Biber, L. (2021), Modélisation de la distribution de la sévérité en réassurance. Mémoire, QBE Re.

Borman, S. (2004), 'The expectation maximization algorithm-a short tutorial', *Submitted for publication* **41**.

Brewer, M. J., Butler, A. & Cooksley, S. L. (2016), 'The relative performance of aic, aicc and bic in the presence of unobserved heterogeneity', *Methods in Ecology and Evolution* **7**(6), 679–692.

Lee, S. C. & Lin, X. S. (2010), 'Modeling and evaluating insurance losses via mixtures of erlang distributions', *North American Actuarial Journal* **14**(1), 107–130.

Masquelein, A. (2022*a*), About market model & truncation. Working paper, QBE Re.

Masquelein, A. (2022*b*), About severity distribution. Working paper, QBE Re.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. & Lin, S. (2015), 'Fitting mixtures of erlangs to censored and truncated data using the em algorithm', *ASTIN Bulletin : The Journal of the IAA* **45**(3), 729–758.

Annexes

Annexe A

Calculs pour l'algorithme de Newton-Raphson

A.1 Les dérivées

Calcul de $\frac{\partial N_{i,j}(\theta)}{\partial \theta}$:

$$\begin{aligned}\frac{\partial N_{i,j}(\theta)}{\partial \theta} &= \frac{(t_i^l)^{r_j+1}e^{-t_i^l/\theta}}{\theta^2} - \frac{(t^u)^{r_j+1}e^{-t_i^l/\theta}}{\theta^2} \\ &= \frac{1}{\theta^2}\left((t_i^l)^{r_j+1}e^{-t_i^l/\theta} - (t^u)^{r_j+1}e^{-t_i^l/\theta}\right)\end{aligned} \tag{A.1}$$

Calcul de $\frac{\partial D_{i,j}(\theta)}{\partial \theta}$:

D'abord

$$\begin{aligned}\frac{\partial F(t^u; r_j, \theta)}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(\int_0^{t^u} \frac{1}{\Gamma(r_j)}\theta^{-r_j}t^{r_j-1}e^{-\frac{t}{\theta}}dt\right) \\ &= \int_0^{t^u} \frac{-r_j}{\Gamma(r_j)}\theta^{-r_j-1}t^{r_j-1}e^{-\frac{t}{\theta}}dt + \int_0^{t^u} \frac{1}{\Gamma(r_j)}\theta^{-r_j-2}t^{r_j}e^{-\frac{t}{\theta}}dt \\ &= \frac{-r_j}{\theta}\int_0^{t^u} \frac{1}{\Gamma(r_j)}\theta^{-r_j}t^{r_j-1}e^{-\frac{t}{\theta}}dt + \frac{r_j}{\theta}\int_0^{t^u} \frac{1}{\Gamma(r_j+1)}\theta^{-r_j+1}t^{r_j}e^{-\frac{t}{\theta}}dt \\ &= \frac{r_j}{\theta}\left(F(t^u; r_j+1, \theta) - F(t^u; r_j, \theta)\right)\end{aligned} \tag{A.2}$$

d'où

$$\begin{aligned}
\frac{\partial D_{i,j}(\theta)}{\partial \theta} &= (r_j-1)\theta^{r_j-2}(r_j-1)!\left(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta)\right)\\
&+\theta^{r_j-1}(r_j-1)!\frac{r_j}{\theta}\left(F(t^u;r_j+1,\theta)-F(t^u;r_j,\theta)\right.\\
&\left.-\frac{r_j}{\theta}\left(F(t_i^l;r_j+1,\theta)-F(t_i^l;r_j,\theta)\right)\right)\\
&=(r_j-1)\theta^{r_j-2}(r_j-1)!\left(\left(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta)\right)((r_j-1)-r_j)\right.\\
&\left.+r_j\left(F(t^u;r_j+1,\theta)-F(t_i^l;r_j+1,\theta)\right)\right)\\
&=\theta^{r_j-2}(r_j-1)!\left(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta)\right.\\
&\left.+r_j\left(F(t^u;r_j+1,\theta)-F(t_i^l;r_j+1,\theta)\right)\right)
\end{aligned} \tag{A.3}$$

et finalement

$$\frac{\partial D_{i,j}(\theta)}{\partial \theta} = \theta^{r_j-2}(r_j-1)!\left(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta)+r_j\left(F(t^u;r_j+1,\theta)-F(t_i^l;r_j+1,\theta)\right)\right) \tag{A.4}$$

A.2 La base logarithmique

Réécriture de $T_{i,j}(\theta)$:

$$\begin{aligned}
T_{i,j}(\theta) &= \frac{(t_i^l)^{r_j}e^{-t_i^l/\theta}-(t^u)^{r_j}e^{-t^u/\theta}}{\theta^{r_j-1}(r_j-1)!(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta))}\\
&= \frac{(t_i^l)^{r_j}e^{-t_i^l/\theta}}{\theta^{r_j-1}(r_j-1)!(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta))}-\frac{(t^u)^{r_j}e^{-t^u/\theta}}{\theta^{r_j-1}(r_j-1)!(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta))}\\
&= exp\left(r_j log(t_i^l)-\frac{t_i^l}{\theta}-(r_j-1)log(\theta)-\sum_{k=1}^{r_j-1}log(k)-log(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta))\right)\\
&- exp\left(r_j log(t^u)-\frac{t^u}{\theta}-(r_j-1)log(\theta)-\sum_{k=1}^{r_j-1}log(k)-log(F(t^u;r_j,\theta)-F(t_i^l;r_j,\theta))\right)
\end{aligned} \tag{A.5}$$

Réécriture de $\frac{\partial T_{i,j}}{\partial \theta}$:

$$\begin{aligned}
\frac{\partial T_{i,j}}{\partial \theta} &= \frac{D_{i,j}(\theta)*\frac{\partial N_{i,j}(\theta)}{\partial \theta}-N_{i,j}(\theta)*\frac{\partial D_{i,j}(\theta)}{\partial \theta}}{(D_{i,j}(\theta))^2}\\
&= \frac{\frac{\partial N_{i,j}(\theta)}{\partial \theta}}{D_{i,j}(\theta)}-\frac{N_{i,j}(\theta)*\frac{\partial D_{i,j}(\theta)}{\partial \theta}}{(D_{i,j}(\theta))^2}\\
&= A_{i,j}(\theta)-B_{i,j}(\theta)
\end{aligned} \tag{A.6}$$

avec

$$\begin{aligned} A_{i,j}(\theta) &= \frac{\frac{(t_i^l)^{r_j+1}e^{-t_i^l/\theta}}{\theta^2} - \frac{(t^u)^{r_j+1}e^{-t_i^l/\theta}}{\theta^2}}{D_{i,j}(\theta)} \\ &= \frac{(t_i^l)^{r_j+1}e^{-t_i^l/\theta}}{\theta^2 D_{i,j}(\theta)} - \frac{(t^u)^{r_j+1}e^{-t_i^l/\theta}}{\theta^2 D_{i,j}(\theta)} \\ &= exp\left((r_j+1)log(t_i^l) - \frac{t_i^l}{\theta} - 2log(\theta) - log(D_{i,j}(\theta))\right) \\ &- exp\left((r_j+1)log(t^u) - \frac{t^u}{\theta} - 2log(\theta) - log(D_{i,j}(\theta))\right) \end{aligned} \tag{A.7}$$

et

$$\begin{aligned} B_{i,j}(\theta) &= \frac{\frac{\partial D_{i,j}(\theta)}{\partial\theta} * \left((t_i^l)^{r_j}e^{-t_i^l/\theta} - (t^u)^{r_j}e^{-t^u/\theta}\right)}{(D_{i,j}(\theta))^2} \\ &= \frac{\frac{\partial D_{i,j}(\theta)}{\partial\theta} * (t_i^l)^{r_j}e^{-t_i^l/\theta}}{(D_{i,j}(\theta))^2} - \frac{\frac{\partial D_{i,j}(\theta)}{\partial\theta} * (t^u)^{r_j}e^{-t^u/\theta}}{(D_{i,j}(\theta))^2} \\ &= exp\left(log\left(\frac{\partial D_{i,j}(\theta)}{\partial\theta}\right) + r_j log(t_i^l) - \frac{t_i^l}{\theta} - 2log(D_{i,j}(\theta))\right) \\ &- exp\left(log\left(\frac{\partial D_{i,j}(\theta)}{\partial\theta}\right) + r_j log(t^u) - \frac{t^u}{\theta} - 2log(D_{i,j}(\theta))\right) \end{aligned} \tag{A.8}$$

Annexe B

Calculs pour la queue de distribution

B.1 Des vraisemblances

B.1.1 La vraisemblance de la loi de Pareto

$$\begin{aligned}\mathcal{L}(x) &= \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \alpha \frac{A^\alpha}{x^{\alpha+1}} \\ &= \prod_{i=1}^{n} \alpha \frac{x^{-(\alpha+1)}}{A^{-\alpha}} \\ &= \prod_{i=1}^{n} \frac{\alpha}{A} \left(\frac{x}{A}\right)^{-(\alpha+1)}\end{aligned} \tag{B.1}$$

B.1.2 La vraisemblance de la loi exponentielle

$$\mathcal{L}(\lambda) = \prod_{i=1}^{n} f_A(x_i) = \prod_{i=1}^{n} \lambda e^{-\lambda(x_i - A)} \tag{B.2}$$

B.1.3 La vraisemblance de la loi de Pareto tronquée

$$\begin{aligned}\mathcal{L}(x) &= \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \frac{\alpha \frac{A^\alpha}{x^{\alpha+1}}}{1 - \left(\frac{T}{A}\right)^{-\alpha}} \\ &= \prod_{i=1}^{n} \frac{\frac{\alpha}{A} \left(\frac{x_i}{A}\right)^{-(\alpha+1)}}{1 - \left(\frac{T}{A}\right)^{-\alpha}}\end{aligned} \tag{B.3}$$