

Faculté des sciences

Calcul des primes d'invalidité à l'aide de la méthode des états fictifs

Auteur : **Derré Jean-Baptiste ESSEHIN**

Promoteur : **Michel Denuit**

Lecteur : **Jérôme Barbarin**

Année académique 2022-2023

Master en sciences actuarielles, à finalité spécialisée

Table des matières

Table des matières	i
Liste des figures	iii
Abstract	v
Résumé	vi
Remerciements	vii
Introduction	1
1 Modèle à trois états	4
1.1 Introduction	4
1.2 Modélisation	4
1.2.1 présentation du modèle	4
1.2.2 Modèle semi-markovien	5
1.3 Taux instantanés de transition	6
1.3.1 Définition	6
1.3.2 Probabilité de transition	7
1.3.3 Probabilité de séjour	8
1.4 Aspects techniques	8
1.4.1 Primes	8
1.4.2 Réserves mathématiques	8
1.5 Algorithme de Waters	9
1.5.1 Intérêt	9
1.5.2 Algorithme de Waters	10
1.5.3 Primes et réserves	17
1.6 Algorithme de Monté-Carlo	18
1.6.1 Intérêt	18
1.6.2 Approche générale	18
1.6.3 Approche pratique	18
1.6.4 Algorithme	19
1.7 Applications aux taux tests	20

1.7.1	Enoncé	20
1.7.2	Analyse et interprétation	21
1.7.3	Origine de la sous-estimation	22
1.8	Limites	24
2	Modèle multi-états : méthode des états fictifs	25
2.1	Chaîne de Markov à temps discret	25
2.2	Modèle markovien : modèle des états fictifs	27
2.2.1	Intérêt	27
2.2.2	Présentation du modèle	27
2.3	Taux de transition	29
2.3.1	Constance par morceaux	29
2.3.2	Raison scientifique du modèle étudié	30
2.4	Probabilités de transition	30
2.4.1	Probabilités de transition entre deux états successifs	31
2.4.2	Constance par morceaux des taux de transition	32
2.4.3	Probabilités de l'état	34
2.4.3.1	Matrice de transition	34
2.4.3.2	Probabilités de l'Etat	35
2.5	Aspects Techniques	36
2.5.1	Prestations variées	36
2.5.2	Franchise temporelle	37
2.5.3	Primes	37
2.5.4	Réserves	38
2.6	Application aux taux tests	38
2.6.1	Enoncé	38
2.6.2	Analyse et interprétation	39
3	Application de l'approche des états fictifs	42
3.1	Présentation des données	42
3.1.1	Sources	42
3.1.2	Structures des taux	43
3.1.2.1	Taux de sortie d'activité	43
3.1.2.2	Taux de sortie d'invalidité	44
3.2	Probabilités de transition	46
3.3	Primes et réseves	48
3.3.1	Prestations	48
3.3.2	Primes	49
	Conclusion	52
	Bibliographie	54

Appendices	56
.1 Théorème des accroissements finis	57
.1.1 Théorème des accroissements finis	57
.1.2 Théorème des accroissements finis généralisé	57
.2 Intégration numérique : méthode des trapèze	57
.3 Taux de transition test	57
.3.1 taux de transition actif-invalidé	57
.3.2 taux de transition invalide-actif en semaine	58
.3.3 taux de transition invalide-décès en semaine	58
.3.4 taux de transition actif-décès	58
.4 Taux réels de sortie d'invalidité	59
.5 Taux de primes de la méthode des états fictifs	59
.6 Code R	60

Table des figures

1.2.1 <i>Modèle invalidité</i>	5
1.2.2 <i>modèle semi-markovien</i>	6
1.7.1 <i>Primes Monte-carlo</i>	21
1.7.2 <i>Primes Waters en fonction du pas de discrétisation</i>	21
1.7.3 <i>Probabilités de transition par l'algorithme de waters</i>	22
2.1.1 <i>marche aléatoire sur les entiers</i>	26
2.2.1 <i>processus en temps discret</i>	29
2.6.1 <i>Primes Markov en fonction du pas de discrétisation</i>	39
2.6.2 <i>Probabilités de transition par l'algorithme des états fictifs et Waters</i>	39
2.6.3 <i>Probabilités de sortie d'invalidité après une semaine par l'algorithme des états fictifs</i>	40
3.1.1 <i>Taux de transition de sortie d'activité</i>	43
3.1.2 <i>Taux de réactivation en fonction de l'âge et de la durée en invalidité</i>	45
3.1.3 <i>Taux de décès des invalides en fonction de l'âge et de la durée en invalidité</i>	46
3.2.1 <i>Probabilités de transition obtenues par des taux constants par morceaux et des taux continus</i>	47
3.2.2 <i>Comparaison des Primes</i>	48
3.3.1 <i>Primes des états fictifs</i>	50
3.3.2 <i>Réserves des actifs après 10 ans</i>	50
.4.1 <i>Taux de transition de sortie d'invalidité selon la duree</i>	59
.4.2 <i>Taux de transition de sortie d'invalidité selon l'âge</i>	59
.5.1 <i>primes des états fictifs</i>	60

*Ainsi ma parole, qui sort de ma bouche, ne me reviendra pas sans résultat, sans avoir fait
ce qui me plaît, sans avoir accompli sa mission.*
Isaïe 55, 11

À Nathan et Eliora

Abstract

The pricing of premiums and reserves in disability insurance presents a particular complexity due to the possibility of reactivation among disabled individuals. This reactivation depends not only on the age of the insured but also on the duration spent in the disabled state. To account for this influence of the duration of disability on mortality and reactivation rates, it is necessary to transition to a semi-Markovian model, which entails complex calculations involving the resolution of relatively heavy integro-differential equations.

Various algorithms have been developed by actuaries to approximate the disability premium, with Waters' algorithm being a commonly used reference. However, an in-depth analysis has revealed the limitations of this algorithm. Its construction relies solely on numerical analysis reasoning, without incorporating a probabilistic approach, which can lead to unrealistic results.

In order to enhance the pricing in this field, we have adopted a probabilistic approach that retains the Markovian calculations while taking into account the duration spent in the disabled state. This approach is known as the fictitious states model. It has successfully addressed the issues encountered with the Waters algorithm and offers several advantages. It enables us to provide comprehensive insurance products that cater to the specific needs of policyholders in different situations, thereby providing them with appropriate financial protection.

Key words : Pricing, premium, reserve, disability insurance, semi-Markovian model, Markovian model, Waters' algorithm, Monte Carlo simulation, Fictitious States model.

Résumé

La tarification des primes et des réserves en assurance invalidité présente une complexité particulière en raison de la possibilité de réactivation chez les personnes invalides. Cette réactivation dépend non seulement de l'âge de l'assuré, mais également de la durée passée dans l'état invalide. Afin de prendre en compte cette influence de la durée d'invalidité sur les taux de mortalité et de réactivation, il est nécessaire de passer à un modèle semi-markovien, ce qui entraîne des calculs complexes impliquant la résolution de systèmes d'équations intégral-différentielles assez lourds.

Pour approximer la prime en assurance invalidité, les actuaires ont développé différents algorithmes, parmi lesquels l'algorithme de Waters est une référence couramment utilisée. Cependant, une analyse approfondie a révélé les limites de cet algorithme. Sa construction repose uniquement sur des raisonnements d'analyse numérique, sans prendre en compte d'approche probabiliste, ce qui peut conduire à des résultats irréalistes.

Dans le but d'améliorer la tarification dans ce domaine, nous avons adopté une approche probabiliste qui conserve les calculs markoviens tout en tenant compte de la durée passée en état d'invalidité. Il s'agit du modèle des états fictifs. Cette approche a permis de résoudre les problèmes rencontrés avec l'algorithme de Waters et présente plusieurs avantages. Elle nous permet d'offrir des produits d'assurance complets qui répondent aux besoins spécifiques des assurés dans différentes situations, en leur fournissant une protection financière adaptée.

Mots clés : Tarification, prime, réserve, assurance, invalidité, modèle semi-markovien, modèle markovien, algorithme de Waters, simulation de Monté-Carlo, Modèle des états fictifs.

Remerciements

Je remercie spécialement le professeur Michel DENUIT pour la patience, les conseils et la grande disponibilité qu'il a manifesté à mon égard.

Les différents rendez-vous furent très enrichissants pour moi dans la conduite de ce mémoire et bien d'autres.

Je tiens aussi à remercier tout le corps enseignant et le personnel de la LSBA pour la qualité de la formation reçue.

Au terme de ce parcours, je remercie ma famille pour les sacrifices, le soutien et la confiance indéfectibles dans mes choix. Leurs attentions et encouragements m'ont accompagné tout au long de mes études.

J'exprime toute ma gratitude à mes camarades de classe et aux personnes qui m'ont aidé de près ou de loin dans ce travail en particulier à Aboubakar TOUFFOUO et Kouadio-Bla Essyh Leandre EKRA.

Introduction

Contexte général

L'assurance des personnes englobe des situations où les prestations dépendent d'événements incertains liés à la vie, à la situation familiale ou encore à la santé d'un individu, tels que l'incapacité de travail et l'invalidité.

Ce travail se focalise sur la modélisation actuarielle des contrats d'assurance couvrant l'incapacité de travail et l'invalidité. Ces contrats offrent une indemnisation en cas d'arrêt de travail causé par une maladie ou un accident, en complément des prestations fournies par la Sécurité sociale.

La gestion du risque liée à l'incapacité de travail et l'invalidité est complexe et implique des aspects tels que la rédaction des contrats d'assurance, une sélection médicale rigoureuse et un suivi régulier des sinistres. Les différentes clauses limitant la couverture offerte par l'assureur, comme les délais de carence, les plafonds d'indemnisation ou les franchises temporelles, rendent l'analyse plus compliquée mais sont importantes pour lutter contre la sélection adverse et évaluer partiellement le niveau de risque choisi par l'assuré (cours [LACTU2230](#)).

Dans le contexte de la solvabilité des assureurs, il est primordial de calculer les réserves avec prudence, afin d'éviter une éventuelle insolvabilité de l'assureur. Cela requiert des choix judicieux en termes d'hypothèses et de paramètres techniques, ainsi que la garantie de disposer de revenus suffisants pour couvrir ces réserves.

La norme Solvabilité II a été mise en place pour assurer la solvabilité des assureurs vis-à-vis des assurés. Elle propose des directives concernant la prise en compte des principaux risques encourus par les compagnies d'assurance. Cependant, elle ne fournit pas de méthodologie spécifique pour évaluer de manière adéquate les primes d'assurance.

Problématique

Actuellement, la tarification des assurances santé repose sur une approche multi-états. Cette approche considère que tous les aléas liés à l'état de santé de l'assuré peuvent être décrits par sa position dans un ensemble fini d'états. Les primes et les prestations en découlent également.

Dans notre travail, nous examinons un modèle à trois états : "actif", "invalidé" et "décédé". Dans de nombreux cas, il est raisonnable de supposer que le processus décrivant l'état de santé de l'assuré est de type markovien. La théorie de Markov offre des outils qui permettent de simplifier considérablement les calculs.

Cependant, dans certaines situations, il est nécessaire de prendre en compte les durées de séjour. Par exemple, pour l'assurance invalidité, l'observation empirique des taux de transition a révélé la nécessité de considérer la durée de séjour pour une tarification précise. En effet, les taux de réactivation et de décès des personnes invalides dépendent non seulement de l'âge de l'assuré, mais également de la durée passée dans cet état, car les longues périodes d'invalidité ont très peu de chances de se réactiver ([Sian, 2017](#)). L'hypothèse markovienne est donc remise en question, car son utilisation sous-estime les invalidités de longue durée et surestime les invalidités de courte durée. La prise en compte de cet effet temporel nécessite l'utilisation de processus semi-markoviens, mieux adaptés à ce type de situation. Cependant, l'utilisation de ces processus conduit à des équations intégrales-différentielles difficiles à résoudre analytiquement lors du calcul des probabilités de transition.

Objectif

Dans la littérature, différentes méthodes ont été proposées pour prendre en compte l'effet temporel dans l'assurance invalidité. Parmi celles-ci, l'algorithme récursif présenté par [Waters \(1991\)](#) est largement utilisé dans la pratique. Cet algorithme permet d'évaluer directement les probabilités de transition en utilisant une discrétisation des taux de transition. Cependant, la précision des probabilités de transition dépend du pas de discrétisation choisi, et une discrétisation trop grossière peut conduire à une moins grande précision.

Dans notre travail, nous allons d'abord utiliser l'algorithme de Waters pour la tarification en invalidité. Nous comparerons ensuite les résultats obtenus avec ceux issus de simulations de Monte-Carlo, afin de vérifier la performance de l'algorithme de Waters. Nous mettrons en évidence les limites de cet algorithme.

Ensuite, nous développerons le modèle des états fictifs proposé par [E. Pitacco \(1994\)](#). Cette approche consiste à découper la durée de séjour en invalidité en intervalles, puis à construire des sous-états invalides. Cela permet de conserver les calculs markoviens tout

en prenant en compte l'effet temporel. Nous comparerons les résultats de cette approche avec ceux obtenus par l'algorithme de Waters et les simulations de Monté-Carlo.

Enfin, nous exploiterons les avantages offerts par la méthode des états fictifs, tels que la possibilité de proposer des produits mieux adaptés aux assurés. Nous analyserons les résultats obtenus et mettrons en évidence les améliorations apportées par cette méthode par rapport aux approches précédentes.

Chapitre 1

Modèle à trois états

1.1 Introduction

L'étude de l'évolution d'un système pouvant transiter entre différents états possibles est en général réalisée par des mesures sur ce système à une fréquence suffisamment petite pour que ces mesures soient considérées comme continues. Nous avons, dans ce cas, la possibilité de décrire ce type de système par un processus discret qui caractérise la suite des états visités et par un autre processus continu qui représente le temps passé dans chaque état visité([sian, 2017](#)).

De tels systèmes doivent être modélisés de sorte à tenir compte de la trajectoire entière du processus, car une information perdue du passé peut fortement influencer les issues du futur et fausser les prédictions.

Cependant, dans certains systèmes, la seule information sur l'état présent suffit entièrement à prédire les issues futures du système. On parle alors d'un processus sans mémoire et cette propriété est dite de Markov. Cette hypothèse permet de réduire considérablement les calculs lorsqu'elle est vérifiée par un système par rapport à la prise en compte des données entières sur son évolution.

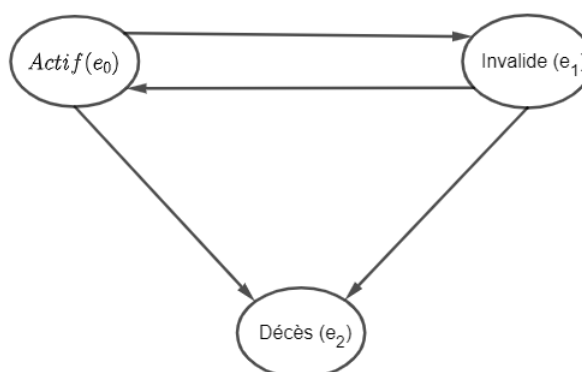
1.2 Modélisation

1.2.1 présentation du modèle

Comme souligné dans l'introduction, la nécessité de prendre en compte le temps passé dans certains états du processus stochastique peut être déterminante pour la suite. Dans notre travail, le modèle étudié comprend trois états "actif" ou "autonome", "invalidé" et "décédé" notés respectivement $\{e_0, e_1, e_2\}$ ou encore $\{a, i, d\}$ tel que vu au cours [LACTU2230](#) (voir figure 1.2.1).

Les flèches entre les différents états indiquent les possibilités de transition à savoir :

- l'invalidation : transition de l'état e_0 vers l'état e_1

FIGURE 1.2.1 – *Modèle invalidité*

- la réactivation : transition de l'état e_1 vers l'état e_0
- le décès en activité : transition de l'état e_0 vers l'état e_2
- le décès en invalidité : transition de l'état e_1 vers l'état e_2

Remarque 1. Nous constatons une différence entre la modélisation de l'invalidité et celle de la dépendance totale, où un individu qui tombe en dépendance ne peut retrouver son état d'autonomie, donc aucune réactivation n'est possible dans le futur. Cette différence peut entraîner de lourdes conséquences au niveau des calculs des primes et réserves.

La majorité des idées de cette partie est inspirée des travaux de [Denuit et Robert \(2007\)](#)

1.2.2 Modèle semi-markovien

La considération de la durée passée en invalidité sur les taux de mortalité et de réactivation nécessite un changement de modèle markovien en modèle semi-markovien. Ce modèle a été introduit pour corriger le défaut du modèle de Markov.

Considérons un contrat souscrit par un assuré d'âge x et prenons le moment de la signature de la police comme origine des temps ($t = 0$). Bien que dans la modélisation, un seul état suffise pour l'invalidité, la durée de séjour dans cet état est généralement scindé en deux dans la pratique. En début d'invalidité, on parle d'incapacité (primaire) basée sur le régime de la sécurité sociale belge. Si cette incapacité dure plus d'un an, elle est qualifiée d'invalidité. Il est nécessaire de souligner cette différence car les règles d'indemnisation diffèrent en pratique entre l'incapacité et l'invalidité.

Le passage en univers semi-markovien stipule de tenir compte de la durée passée dans l'état actuel pour déterminer le futur du processus. Ce qui nous conduit à passer d'un

processus $\{X_t, t \geq 0\}$ décrivant la trajectoire de l'assuré dans l'espace des états $\{e_0, e_1, e_2\}$ à un processus bivarié $\{(X_t, Z_t), t \geq 0\}$ où X_t correspond à l'état occupé à l'instant t , et Z_t à la durée d'occupation de cet état.

Définition 1.2.1. La variable aléatoire Z_t représente le temps passé par l'assuré dans l'état X_t :

$$Z_t = \max\{z \leq t / X_t = X_{t-h} \text{ pour tout } 0 \leq h \leq z\}.$$

Dans notre cas, l'état invalide devient donc semi-markovien. La modélisation de cet état prendra donc en compte l'âge de l'assuré et le temps passé par l'assuré dans cet état. Dans la suite, nous considérons que seul l'état e_1 est semi-markovien et que les états e_0 et e_2 conservent leur caractère markovien. Une approche directe avec des raisonnements probabilistes fondés sur les temps passés dans les différents états visités n'est plus possible, et il nous faudra utiliser des algorithmes numériques pour obtenir les probabilités de transition ou différentes approximations.

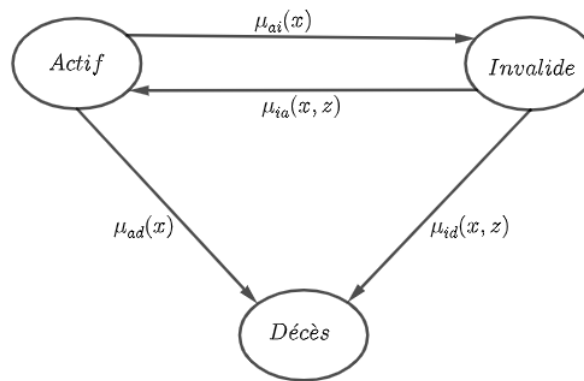


FIGURE 1.2.2 – *modèle semi-markovien*

1.3 Taux instantanés de transition

1.3.1 Définition

Dans le domaine de l'assurance, il est plus pratique de travailler avec des taux de transition car cela permet de simplifier les calculs avec les taux d'intérêts. Afin de bien les définir, on fait l'hypothèse qu'il y a un nombre fini de transitions sur tout intervalle borné et que la limite

$$\lim_{t \rightarrow s} \frac{P[X_t = j / X_s = i]}{t - s}$$

existe et est finie pour tout $i \neq j \in a, i, d$.

Dans la mesure où l'état a est markovien, les taux de transition partant de a sont définis en tenant compte uniquement de l'âge.

Ainsi, le taux de transition de l'état a vers j ($j \in \{i, d\}$) à l'instant t est défini par :

$$\mu_{aj}(x+t) = \lim_{\Delta t \rightarrow 0} \frac{P[X_{t+\Delta t} = j / X_t = a]}{\Delta t} \quad (1.3.1)$$

On définit ainsi les taux suivants :

- $\mu_{ai}(x+t)$: le taux d'invalidation à l'âge $x+t$
- $\mu_{ad}(x+t)$: le taux de décès à l'âge $x+t$ et
- $\mu_{a\bullet}(x+t) = \mu_{ai}(x+t) + \mu_{ad}(x+t)$: le taux de sortie de l'état actif à l'âge $x+t$, toutes causes confondues.

Contrairement à l'état i qui demeure semi-markovien, les taux de transition depuis cet état prennent en compte l'âge de l'assuré $x+t$ et le temps z passé dans cet état.

Ainsi, le taux de transition de l'état i vers j ($j \in \{a, d\}$) à l'instant t est défini par :

$$\mu_{ij}(x+t, z) = \lim_{\Delta t \rightarrow 0} \frac{P[X_{t+\Delta t} = j / X_t = i, Z_t = z]}{\Delta t} \quad (1.3.2)$$

On définit ainsi les taux suivants :

- $\mu_{ia}(x+t, z)$: le taux de réactivation d'un assuré d'âge $x+t$ invalide depuis z années.
- $\mu_{id}(x+t, z)$: le taux de mortalité d'un assuré d'âge $x+t$ invalide depuis z années.
- $\mu_{i\bullet}(x+t, z) = \mu_{ia}(x+t, z) + \mu_{id}(x+t, z)$: le taux de sortie d'invalidité à l'âge $x+t$ dont la durée de l'invalidité en cours est z , toutes causes confondues.

1.3.2 Probabilité de transition

Les probabilités de transition évaluent la probabilité qu'un individu occupant un état donné à un instant donné soit dans un autre état à un instant futur, sans contrôler la trajectoire entre ces deux instants.

L'état a étant markovien, les probabilités de transition depuis a sont définies en considérant uniquement l'âge de l'assuré. On définit donc, pour $j \in \{a, i, d\}$ et $s \leq t$, la probabilité qu'un actif à l'âge $x+s$ le soit également à l'âge $x+t$:

$${}_{t-s}p_{x+s}^{aj} = P[X_t = j | X_s = a] \quad (1.3.3)$$

L'état i étant semi-markovien, les probabilités de transition depuis i sont définies en considérant l'âge de l'assuré et la durée passée dans cet état. On définit donc, pour $j \in \{a, i, d\}$ et $s \leq t$, la probabilité qu'un invalide d'âge $x+s$ entré en invalidité à l'âge $x+s-z$ soit à l'état j à l'âge $x+t$:

$${}_{t-s}p_{x+s,z}^{ij} = P[X_t = j | X_s = i, Z_s = z] \quad (1.3.4)$$

Comme vu au cours [LACTU2230](#), nous admettons que :

$$\begin{aligned} {}_0p_x^{aa} &= 1 \\ {}_0p_x^{ai} &= {}_0p_x^{ad} = 0. \end{aligned} \quad (1.3.5)$$

Ce qui sous-entend que le contrat d'assurance est signé avec un assuré actif.

1.3.3 Probabilité de séjour

Notons ${}_tp_x^{\overline{aa}}$ la probabilité qu'un assuré actif à l'âge x séjourne dans cet état jusqu'à l'âge $x+t$ et ${}_{\tau}p_{x+t,z}^{\overline{ii}}$ la probabilité qu'un invalide âgé de $x+t$ années, entré en invalidité à l'âge $x+t-z$, demeure invalide sans interruption jusqu'à l'âge $x+t+\tau$. on a :

$${}_tp_x^{\overline{aa}} = \exp\left(-\int_0^t \mu_{a\bullet}(x+\tau)d\tau\right) \quad (1.3.6)$$

$${}_{\tau}p_{x+t,z}^{\overline{ii}} = \exp\left(-\int_0^{\tau} \mu_{i\bullet}(x+t+\xi; z+\xi)d\xi\right). \quad (1.3.7)$$

1.4 Aspects techniques

1.4.1 Primes

Généralement, la prime n'est due que si l'assuré est actif, une dispense de paiement de la prime étant prévue pour les invalides. Considérons un contrat d'assurance classique et simple de durée n , c'est-à-dire sans franchise f ni période d'attente w , avec une prestation annuelle en invalidité constante b indépendante de l'âge x et du temps passé en invalidité z . Dans ce cas, la relation d'équivalence entre les prestations des deux parties nous permet d'obtenir le taux de prime annuel constant (c'est-à-dire lissé sur la durée du contrat) suivant [Denuit et Robert \(2007\)](#) :

$$\pi = b \frac{\overline{a}_{x;n}^{ai}}{\overline{a}_{x;n}^{aa}} \quad (1.4.1)$$

$$\text{avec } \overline{a}_{x;n}^{aa} = \int_0^n v(0,t) {}_tp_x^{aa} dt \quad \text{et} \quad \overline{a}_{x;n}^{ai} = \int_0^n v(0,t) {}_tp_x^{ai} dt \quad (1.4.2)$$

1.4.2 Réserves mathématiques

Pour un tel produit, la réserve pour un assuré actif à l'âge $x+t$ vaut

$$\begin{aligned} V_a(t) &= b \int_t^n \left({}_{\tau-t}p_{x+t}^{aa} \mu_{ai}(x+\tau) \int_{\tau}^n {}_{\xi-\tau}p_{x+\tau,0}^{\overline{ii}} v(t,\xi) d\xi \right) d\tau \\ &\quad - \pi \int_t^n {}_{\tau-t}p_{x+t}^{aa} v(t,\tau) d\tau. \end{aligned} \quad (1.4.3)$$

$$= b \overline{a}_{x+t;n-t}^{ai} - \pi \overline{a}_{x+t;n-t}^{aa} \quad (1.4.4)$$

La réserve pour un assuré invalide à l'âge $x + t$ se voit quant à elle rajouter un second argument, à savoir la durée z passée en invalidité. Dans ce cas,

$$\begin{aligned} V_i(t, z) = & b \int_t^n {}_{\tau-t}p_{x+\tau, z}^{\bar{ii}} v(t, \tau) d\tau \\ & + b \int_t^n \left({}_{\tau-t}p_{x+t, z}^{ia} \mu_{ai}(x + \tau) \int_{\tau}^n {}_{\xi-\tau}p_{x+\tau, 0}^{\bar{ii}} v(t, \xi) d\xi \right) d\tau \\ & - \pi \int_t^n {}_{\tau-t}p_{x+t, z}^{ia} v(t, \tau) d\tau. \end{aligned} \quad (1.4.5)$$

Pour des durées d'invalidité assez courtes, nous avons ${}_{\tau-t}p_{x+t}^{aa} \approx {}_{\tau-t}p_{x+t, z}^{ia}$.
Finalement nous avons :

$$V_i(t, z) \approx b \int_t^n {}_{\tau-t}p_{x+t, z}^{\bar{ii}} v(t, \tau) d\tau + V_a(t). \quad (1.4.6)$$

Une manière prudente de calculer la réserve pour un invalide âgé de $x + t$ est d'approximer cette réserve en additionnant la provision pour les sinistres en cours à la réserve pour un actif du même âge. Cette approche est souvent utilisée en pratique pour minimiser le risque de perte de l'assureur car elle sous-entend qu'un assuré invalide reste toujours susceptible d'entrer en invalidité. En d'autres termes, il est possible qu'un assuré sortant de l'état invalide à la date t puisse y retourner à $t + dt$.

1.5 Algorithme de Waters

1.5.1 Intérêt

L'hypothèse de Markov ne tient plus dans l'état e_1 car les fonctions $\mu_{ia} = \mu_{ia}(x, z)$ et $\mu_{id} = \mu_{id}(x, z)$ dépendent non seulement de l'âge de l'assuré mais également du temps passé en invalidité, ce qui aura un impact sur les primes et les réserves. Ainsi, nous sommes dans un modèle semi-markovien.

On définit la valeur actuelle moyenne d'une rente de 1€ par an depuis le début de l'invalidité par :

$$\bar{a}_{x+t, 0}^i = \int_0^\infty \exp \left(- \int_0^s \mu_{i\bullet}(x + t + z, z) dz \right) v(t, t + s) ds. \quad (1.5.1)$$

La prime unique pure de la couverture invalidité sur une durée n peut alors se mettre sous la forme :

$$\int_0^n {}_t p_x^{aa} \mu_{ai}(x + t) v(0, t) \bar{a}_{x+t, 0}^i dt \quad (1.5.2)$$

Cette formulation est simple et facilement compréhensible, mais elle complique le calcul de ${}_t p_x^{aa}$, contrairement aux modèles de dépendance totale où ces probabilités sont des probabilités de séjour dans l'état autonome, ici elles incluent des possibilités de transition

entre l'autonomie et l'invalidité. Par conséquent, il n'existe plus de formule fermée pour calculer cette probabilité, comme c'est le cas avec les modèles markoviens.

Les probabilités de transition peuvent s'obtenir comme solutions d'un système d'équations intégral-différentielles. Cette approche est cependant assez lourde et peu efficace en termes de temps de calcul car elle nécessite la résolution numérique de ce système ([cours LACTU2230](#)). C'est pourquoi les actuaires ont développés plusieurs autres méthodes numériques. L'algorithme de Waters est la méthode la plus connue, mais on peut également citer l'algorithme de Levikson-Frostig-Bshouty ([Denuit et Robert, 2007, p. 257](#)), et aussi la distribution de type phase mais celle-ci ne rend pas compte des invalidités de courte durée de manière appropriée ([Sian, 2017, p. 57](#)).

Dans notre travail, nous étudierons l'algorithme de Waters.

1.5.2 Algorithme de Waters

Approche générale

En 1991, H.R. Waters a proposé un algorithme pour approximer les probabilités de transition semi-markoviennes dans un processus à trois états actif, invalide, décès en utilisant les taux de transition connus.

Le processus est caractérisé par les taux de transition connus entre les états, qui peuvent être exprimés sous forme de matrices de taux de transition. Les probabilités de transition semi-markoviennes sont définies comme la probabilité de passer d'un état à un autre en un certain temps, étant donné que l'on a déjà été dans l'état initial pendant une certaine durée.

Le processus de calcul des probabilités de transition semi-markoviennes pour les systèmes à états multiples peut être décomposé en quatre étapes principales : initialisation, itération, convergence et calcul des probabilités de durée.

1. Dans la première étape, les probabilités de transition actif-invalide et actif-décès sont initialisées à zéro, et les probabilités de transition immédiate actif-actif, qui sont initialisées à 1.
2. Ensuite, les probabilités de transition semi-markoviennes sont mises à jour itérativement en utilisant la formule de Chapman-Kolmogorov. Les nouvelles probabilités de transition sont stockées dans une nouvelle matrice.
3. L'itération se poursuit jusqu'à ce que les probabilités de transition semi-markoviennes convergent vers une valeur stable. La convergence est déterminée en comparant les valeurs des matrices de probabilités de transition semi-markoviennes à chaque étape de l'itération.

4. Enfin, une fois que les probabilités de transition semi-markoviennes sont calculées, il est possible de calculer les probabilités de durée pour chaque état. Celles-ci sont obtenues en intégrant les probabilités de transition semi-markoviennes sur toutes les durées possibles passées dans un état donné avant de passer à un autre état.

L'algorithme de Waters est largement utilisé dans l'analyse actuarielle et l'évaluation des risques. Il peut être appliqué à d'autres processus semi-markoviens à plus de trois états, mais la complexité de l'approche itérative augmente avec le nombre d'états ([Haberman et Pitacco 1998, P171](#)).

Algorithme

A l'aide des taux de transition connus, l'approche de Waters consiste à discrétiser l'échelle de temps avec un pas régulier Δ . La précision de l'algorithme sera d'autant meilleure que Δ est petit, mais le temps de calcul s'en trouvera augmenté (diviser Δ par deux revient approximativement à quadrupler le temps de calcul, ([Denuit et Robert 2007, P251](#))).

Considérons un assuré actif à l'âge x (âge auquel il souscrit la police) et un âge $y \geq x$. Dans l'approche de l'algorithme de Waters, la probabilité que l'assuré soit actif à cet âge sera définie par :

$$p_a(y) = {}_{y-x}p_x^{aa}$$

et d'être invalide à cet âge :

$$p_i(y) = {}_{y-x}p_x^{ai}$$

Nous supposons dorénavant que $y - x$ est un multiple de Δ . En outre, nous aurons besoin de la probabilité d'être invalide à l'âge y , avec une durée d'invalidité comprise entre $(m - 1)\Delta$ et $m\Delta$, notée $p_i(y, m)$ définie explicitement par :

$$p_i(y, m) = P[X_{y-x} = e_1, Z_{y-x} \leq m\Delta | X_0 = e_0] - P[X_{y-x} = e_1, Z_{y-x} \leq (m - 1)\Delta | X_0 = e_0] \quad (1.5.3)$$

$$= \int_{(m-1)\Delta}^{m\Delta} {}_{y-x-s}p_x^{aa} \mu_{ai}(y - s) {}_s\bar{p}_{y-s,0}^{ii} ds \quad (1.5.4)$$

La connaissance des $p_i(y, m)$ permet de retrouver $p_i(y)$ par sommation grâce à la formule

$$p_i(y) = \sum_{m=1}^{\lceil (y-x)/\Delta \rceil} p_i(y, m). \quad (1.5.5)$$

Le calcul des $p_i(y, m)$ suffit donc pour obtenir $p_i(y)$.

Intéressons-nous à présent à la différence

$$\begin{aligned}
p_i(y + \Delta, m + 1) - p_i(y, m) &= \int_{m\Delta}^{(m+1)\Delta} y + \Delta - x - s p_x^{aa} \mu_{ai}(y + \Delta - s) {}_s\bar{p}_{y+\Delta-s,0}^{ii} ds \\
&\quad - \int_{(m-1)\Delta}^{m\Delta} y - x - s p_x^{aa} \mu_{ai}(y - s) {}_s\bar{p}_{y-s,0}^{ii} ds \\
&= \int_{(m-1)\Delta}^{m\Delta} y - x - s p_x^{aa} \mu_{ai}(y - s) ({}_{s+\Delta}\bar{p}_{y-s,0}^{ii} - {}_s\bar{p}_{y-s,0}^{ii}) ds
\end{aligned} \tag{1.5.6}$$

pour autant que $y - x > m\Delta$. Par ailleurs

$$\begin{aligned}
{}_{s+\Delta}\bar{p}_{y-s,0}^{ii} &= {}_s\bar{p}_{y-s,0}^{ii} \times {}_\Delta\bar{p}_{y,s}^{ii} \\
&= {}_s\bar{p}_{y-s,0}^{ii} \exp\left(-\int_0^\Delta \mu_{i\bullet}(y + u; s + u) du\right)
\end{aligned}$$

En remplaçant cette expression dans celle de la différence, on obtient

$$\begin{aligned}
p_i(y + \Delta, m + 1) - p_i(y, m) &= \int_{(m-1)\Delta}^{m\Delta} y - x - s p_x^{aa} \mu_{ai}(y - s) {}_s\bar{p}_{y-s,0}^{ii} \\
&\quad \times \left(\exp\left(-\int_0^\Delta \mu_{i\bullet}(y + u; s + u) du\right) - 1 \right) ds
\end{aligned} \tag{1.5.7}$$

Soient les fonctions f et g définies à partir de (1.5.7) sur $[(m-1)\Delta, m\Delta]$ par

$$f(s) = \exp\left(-\int_0^\Delta \mu_{i\bullet}(y + u; s + u) du\right) - 1 \quad \text{et} \quad g(s) = y - x - s p_x^{aa} \mu_{ai}(y - s) {}_s\bar{p}_{y-s,0}^{ii}$$

La fonction g est positive sur $[(m-1)\Delta, m\Delta]$. Alors d'après le théorème des accroissements finis généralisés (voir annexe.1), il existe $t \in](m-1)\Delta, m\Delta[$ tel que :

$$\begin{aligned}
p_i(y + \Delta, m + 1) - p_i(y, m) &= \left(\exp\left(-\int_0^\Delta \mu_{i\bullet}(y + u; t + u) du\right) - 1 \right) \\
&\quad \times \int_{(m-1)\Delta}^{m\Delta} y - x - s p_x^{aa} \mu_{ai}(y - s) {}_s\bar{p}_{y-s,0}^{ii} ds \\
&= \left(\exp\left(-\int_0^\Delta \mu_{i\bullet}(y + u; t + u) du\right) - 1 \right) p_i(y, m).
\end{aligned} \tag{1.5.8}$$

Ensuite, considérons la fonction h définie sur $[0, \Delta]$ par :

$$h(\varepsilon) = \exp\left(-\int_0^\varepsilon \mu_{i\bullet}(y + u; t + u) du\right)$$

Alors d'après le théorème des accroissements finis (voir annexe .1), il existe $\tau \in]0, \Delta[$ tel que :

$$\begin{aligned}
f(t) &= h(\Delta) - h(0) = \Delta h'(\tau) \\
&= -\Delta \mu_{i\bullet}(y + \tau; t + \tau) h(\tau)
\end{aligned} \tag{1.5.9}$$

En remplaçant ce résultat dans (1.5.8), on aboutit à :

$$p_i(y + \Delta, m + 1) - p_i(y, m) = -\Delta \mu_{i\bullet}(y + \tau; t + \tau) \exp \left(- \int_0^\tau \mu_{i\bullet}(y + u; t + u) du \right) p_i(y, m) \quad (1.5.10)$$

Pour un certain τ entre 0 et Δ . A présent approximations $p_i(y + \Delta, m + 1) - p_i(y, m)$ en prenant la moyenne de cette dernière expression pour $\tau=0$ et $\tau = \Delta$:

$$p_i(y + \Delta, m + 1) - p_i(y, m) \approx \left(-\frac{\Delta}{2} \mu_{i\bullet}(y; t) - \frac{\Delta}{2} \mu_{i\bullet}(y + \Delta; t + \Delta) \exp \left(- \int_0^\Delta \mu_{i\bullet}(y + u; t + u) du \right) \right) \times p_i(y, m)$$

De la relation 1.5.8, il en ressort que t est tel que

$$p_i(y + \Delta, m + 1) = \exp \left(- \int_0^\Delta \mu_{i\bullet}(y + u; t + u) du \right) p_i(y, m) \quad (1.5.11)$$

Rappelons que $t \in](m-1)\Delta, m\Delta[$ un intervalle de longueur Δ . Pour Δ assez petit, on peut facilement approximer t par le milieu de l'intervalle $(m - \frac{1}{2})\Delta$, nous obtenons au final :

$$\begin{aligned} p_i(y + \Delta, m + 1) - p_i(y, m) &\approx -\frac{\Delta}{2} \mu_{i\bullet}(y; t) p_i(y, m) - \frac{\Delta}{2} \mu_{i\bullet}(y + \Delta; t + \Delta) p_i(y + \Delta, t + \Delta) \\ &\approx -\frac{\Delta}{2} \mu_{i\bullet} \left(y; (m - \frac{1}{2})\Delta \right) p_i(y, m) - \frac{\Delta}{2} \mu_{i\bullet} \left(y + \Delta; (m + \frac{1}{2})\Delta \right) \\ &\quad \times p_i(y + \Delta, m + 1) \end{aligned} \quad (1.5.12)$$

Ceci nous donne

$$p_i(y + \Delta, m + 1) \approx \frac{p_i(y, m) \left(1 - \frac{\Delta}{2} \mu_{i\bullet} \left(y; (m - \frac{1}{2})\Delta \right) \right)}{\left(1 + \frac{\Delta}{2} \mu_{i\bullet} \left(y + \Delta; (m + \frac{1}{2})\Delta \right) \right)} \quad (1.5.13)$$

Remarque 2. Il est important de noter que la relation (1.5.13) ne rencontre aucun problème mathématique avec les approximations utilisées. Toutefois, il convient de souligner que cette équation évalue des probabilités qui doivent toujours être comprises entre 0 et 1. Par conséquent, l'expression $\left(1 - \frac{\Delta}{2} \mu_{i\bullet} \left(y; (m - \frac{1}{2})\Delta \right) \right)$ doit être positive pour que cette approche soit réaliste, sinon les résultats pourraient être négatifs pour des probabilités, entraînant des primes d'assurance négatives. De plus, il convient de noter que le signe de cette expression dépendra fortement du pas Δ .

Par ailleurs, les taux de réactivation diminuent au fil du temps et deviennent quasiment constants, suggérant l'existence d'une durée d'invalidité $m_0\Delta$ à partir de laquelle ces différents taux peuvent être considérés comme constants. Pour tout $m > m_0$, Waters (1991)

propose alors de poser les taux de mortalité et de réactivation comme suit, pour tout âge y :

$$\begin{aligned} \mu_{ia}\left(y; \left(m - \frac{1}{2}\right)\Delta\right) &= \mu_{ia}\left(y; \left(m_0 - \frac{1}{2}\right)\Delta\right) \\ \text{et} \\ \mu_{id}\left(y; \left(m - \frac{1}{2}\right)\Delta\right) &= \mu_{id}\left(y; \left(m_0 - \frac{1}{2}\right)\Delta\right) \end{aligned} \quad (1.5.14)$$

Ces égalités sont supposées vérifier pour tout âge y et $m \geq m_0$, et on parle de taux ultimes dans ce cas.

On peut ensuite définir la probabilité $p_i(y; m_0+)$ d'être invalide à l'âge y depuis au moins $m_0\Delta$ par :

$$p_i(y; m_0+) = \sum_{k \geq 1} p_i(y; m_0 + k) \quad (1.5.15)$$

En partant de la relation (1.5.12) et en remplaçant m par $m_0 + k$ avec $k \geq 0$, on a :

$$\begin{aligned} &p_i(y + \Delta, m_0 + k + 1) - p_i(y, m_0 + k) \approx \\ &-\frac{\Delta}{2}\mu_{i\bullet}\left(y; \left(m_0 + k - \frac{1}{2}\right)\Delta\right)p_i(y, m_0 + k) - \frac{\Delta}{2}\mu_{i\bullet}\left(y + \Delta; \left(m_0 + k + \frac{1}{2}\right)\Delta\right)p_i(y + \Delta, m_0 + k + 1) \\ &\approx -\frac{\Delta}{2}\mu_{i\bullet}\left(y; \left(m_0 - \frac{1}{2}\right)\Delta\right)p_i(y, m_0 + k) - \frac{\Delta}{2}\mu_{i\bullet}\left(y + \Delta; \left(m_0 + \frac{1}{2}\right)\Delta\right)p_i(y + \Delta, m_0 + k + 1) \end{aligned}$$

Par sommation, on obtient :

$$\begin{aligned} &\sum_{k \geq 0} p_i(y + \Delta, m_0 + k + 1) - \sum_{k \geq 0} p_i(y, m_0 + k) \approx \\ &-\frac{\Delta}{2}\mu_{i\bullet}\left(y; \left(m_0 - \frac{1}{2}\right)\Delta\right) \sum_{k \geq 0} p_i(y, m_0 + k) - \frac{\Delta}{2}\mu_{i\bullet}\left(y + \Delta; \left(m_0 + \frac{1}{2}\right)\Delta\right) \sum_{k \geq 0} p_i(y + \Delta, m_0 + k + 1) \end{aligned} \quad (1.5.16)$$

En effectuant un changement de variable sur k et par la relation 1.5.15, on aboutit à :

$$\begin{aligned} &p_i(y + \Delta, m_0+) - \{p_i(y, m_0) + p_i(y, m_0+)\} \approx \\ &-\frac{\Delta}{2}\mu_{i\bullet}\left(y; \left(m_0 - \frac{1}{2}\right)\Delta\right) \{p_i(y, m_0) + p_i(y, m_0+)\} - \frac{\Delta}{2}\mu_{i\bullet}\left(y + \Delta; \left(m_0 + \frac{1}{2}\right)\Delta\right) p_i(y + \Delta, m_0+) \end{aligned} \quad (1.5.17)$$

Ce qui nous conduit à la relation de récurrence suivante :

$$p_i(y + \Delta, m_0+) \approx \frac{\left(p_i(y, m_0) + p_i(y, m_0+)\right) \left(1 - \frac{\Delta}{2}\mu_{i\bullet}\left(y; \left(m_0 - \frac{1}{2}\right)\Delta\right)\right)}{\left(1 + \frac{\Delta}{2}\mu_{i\bullet}\left(y + \Delta; \left(m_0 + \frac{1}{2}\right)\Delta\right)\right)} \quad (1.5.18)$$

En procédant de façon similaire à l'équation (1.5.6), nous obtenons l'approximation suivante :

$$\begin{aligned}
& p_a(y + \Delta) - p_a(y) \approx \\
& -\frac{\Delta}{2} \left(p_a(y) \mu_{i\bullet}(y) - p_a(y + \Delta) \mu_{i\bullet}(y + \Delta) \right) + \frac{\Delta}{2} \sum_{m=1}^{m_0} \left(p_i(y, m) \mu_{ia}(y; (m - \frac{1}{2})\Delta) \right. \\
& \quad \left. + p_i(y + \Delta, m) \mu_{ia}(y + \Delta; (m - \frac{1}{2})\Delta) \right) + \frac{\Delta}{2} \left(p_i(y, m_0+) \mu_{ia}(y; (m_0 - \frac{1}{2})\Delta) \right. \\
& \quad \left. + p_i(y + \Delta, m_0+) \mu_{ia}(y + \Delta; (m_0 - \frac{1}{2})\Delta) \right)
\end{aligned} \tag{1.5.19}$$

et

$$\begin{aligned}
& \frac{\Delta}{2} \left(p_a(y) \mu_{ai}(y + \Delta) + p_a(y + \Delta) \mu_{ai}(y + \Delta) \right) \\
& \approx p_a(y + \Delta, 1) \left(1 + \frac{\Delta}{2} \mu_{i\bullet}(y + \Delta; \frac{\Delta}{2}) \right)
\end{aligned} \tag{1.5.20}$$

En effet, pour avoir la probabilité des individus actifs à l'âge y , intuitivement, il suffit de soustraire de la probabilité d'être actif à l'âge x la probabilité de quitter cet état entre x et y auquel il faut ajouter toutes les réactivations avant l'âge y quelque soit la durée de séjour dans l'état invalide.

$$\begin{aligned}
p_a(y) &= - \int_0^{y-x} p_a(y-s) \mu_{a\bullet}(y-s) ds + \int_0^{y-x} \int_0^{y-x} p_{x,z}^{ia} dz ds \\
&\approx - \int_0^{y-x} p_a(y-s) \mu_{a\bullet}(y-s) ds + \int_0^{y-x} \sum_{m=1}^{\lceil (y-x-s)/\Delta \rceil} p_i(y, m) \mu_{ia}(y-s, (m-s)\Delta) ds
\end{aligned} \tag{1.5.21}$$

En effectuant la différence de probabilités entre l'âge y et $y + \Delta$ puis un changement de variable, on obtient :

$$\begin{aligned}
& p_a(y + \Delta) - p_a(y) \approx \\
& - \int_0^{\Delta} p_a(y + \Delta - s) \mu_{a\bullet}(y + \Delta - s) ds \\
& + \int_0^{\Delta} \sum_{m=1}^{\lceil (y+\Delta-x-s)/\Delta \rceil} p_i(y + \Delta - s, m) \mu_{ia}(y + \Delta - s, (m-s)\Delta) ds
\end{aligned} \tag{1.5.22}$$

Approximons la relation (1.5.22) par la méthode des trapèzes (annexe.1) et conformément aux résultats obtenus ci-dessus, on aboutit à la relation (1.5.19).

D'autre part, par le même procédé [Waters \(1991\)](#) établit l'équation (1.5.20). En effet, intuitivement, le terme à gauche de l'Équation (1.5.20) représente la probabilité d'entrer en invalidité pour un âge compris entre y et $y + \Delta$ et celui de droite la probabilité d'être

invalide à l'âge $y + \Delta$, pour une durée inférieure à Δ à laquelle on ajoute la probabilité de quitter l'état invalide pour une durée Δ (Sian, 2017).

Supposons que nous avons les valeurs de $p_a(y)$, $p_i(y, m)$ et $p_i(y, m_0+)$ pour un certain âge y et pour $m = 1, 2, \dots, m_0$. En utilisant les équations (1.5.13) et (1.5.18), nous pouvons alors calculer $p_i(y + \Delta; m)$ pour $m = 2, 3, \dots, m_0$ et $p_i(y + \Delta; m_0+)$. Ensuite, les équations (1.5.19) et (1.5.20) deviennent deux équations linéaires en $p_i(y + \Delta; 1)$ et $p_a(y + \Delta)$ qu'il suffit de résoudre pour obtenir les probabilités.

En partant des conditions initiales $p_a(x) = 1$ et $p_i(x; m) = 0 = p_i(x; m_0)$, nous pouvons ainsi obtenir itérativement $p_a(y)$, $p_i(y, m)$ et $p_i(y, m_0+)$, ainsi que $p_i(y)$ par sommation, pour tout âge $y \geq x$ multiple de Δ . Il est important de noter que les temps de calcul sont fortement impactés par le pas de discrétisation. En effet, diviser le pas Δ par deux multiplie approximativement le temps de calcul par quatre.

Stabilité du modèle

L'algorithme de Waters utilise une approche itérative pour calculer les probabilités de transition semi-markoviennes. La stabilité de l'algorithme est importante pour s'assurer que les probabilités de transition semi-markoviennes évaluées sont fiables.

Le pas de discrétisation est un paramètre important qui peut affecter le réalisme de l'algorithme de Waters. Le pas de discrétisation est la durée maximale pendant laquelle un individu peut rester dans un état avant de passer à un autre état. Si le pas de discrétisation est trop grand, les probabilités de transition semi-markoviennes évaluées peuvent être imprécises et surtout irréalistes. En général, plus le pas de discrétisation est petit, plus les résultats de l'algorithme de Waters sont précis et stables.

Le choix optimal du pas de discrétisation dépend de la nature du processus semi-markovien et de la précision souhaitée pour les résultats. En pratique, il est souvent nécessaire d'essayer différents pas de discrétisation pour trouver celui qui donne les résultats les plus précis avec le nombre d'itérations minimal.

En résumé, le pas de discrétisation est un paramètre important qui peut affecter la stabilité et le réalisme de l'algorithme de Waters. Un pas de discrétisation plus petit peut conduire à des résultats plus précis, mais peut également nécessiter plus d'itérations.

1.5.3 Primes et réserves

Rappelons que

$$\pi = b \frac{\bar{a}_{x;n|}^{ai}}{\bar{a}_{x;n|}^{aa}} \quad (1.5.23)$$

$$\text{avec } \bar{a}_{x;n|}^{aa} = \int_0^n v(0,t) {}_t p_x^{aa} dt \quad \text{et} \quad \bar{a}_{x;n|}^{ai} = \int_0^n v(0,t) {}_t p_x^{ai} dt \quad (1.5.24)$$

Supposons que n est un multiple de Δ . En appliquant la méthode des trapèzes pour les intégrations numériques (en annexe.1) aux équations (1.5.24) et également aux probabilités de transition de l'algorithme de Waters, on obtient des approximations de $\bar{a}_{x;n|}^{aa}$ et $\bar{a}_{x;n|}^{ai}$, ce qui nous permet d'évaluer la prime de Waters par :

$$\pi \approx b \frac{\sum_{j=0}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} (v(0, j\Delta) p_i(x + j\Delta) + v(0, (j+1)\Delta) p_i(x + (j+1)\Delta))}{\sum_{j=0}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} (v(0, j\Delta) p_a(x + j\Delta) + v(0, (j+1)\Delta) p_a(x + (j+1)\Delta))} \quad (1.5.25)$$

Par le même procédé, en utilisation la relation (1.4.4), la réserve d'un assuré actif à l'âge $x + t$ est approximée par :

$$\begin{aligned} V_a(t) \approx & b \sum_{j=\lfloor \frac{t}{\Delta} \rfloor}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} (v(0, j\Delta) p_i(x + j\Delta) + v(0, (j+1)\Delta) p_i(x + (j+1)\Delta)) \\ & - \pi \sum_{j=\lfloor \frac{t}{\Delta} \rfloor}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} (v(0, j\Delta) p_a(x + j\Delta) + v(0, (j+1)\Delta) p_a(x + (j+1)\Delta)) \end{aligned} \quad (1.5.26)$$

En pratique, pour l'approche de Waters, nous allons utiliser un pas de discrétisation d'une semaine. Pour l'autre méthode, nous utiliserons un pas de discrétisation d'un jour pour les taux de transition et d'une semaine pour les probabilités de transition qui en découlent afin d'assurer la comparabilité avec l'approche de Waters. Pour le calcul des prix de rente et du taux de prime, nous utiliserons des formules identiques à celles de Waters.

Nous avons développé une fonction sur R qui permet de mettre en œuvre l'approche de [Waters \(1991\)](#). Cette fonction prend en entrée l'âge de l'assuré, le taux de prestation, l'âge limite pour le contrat et le pas de discrétisation. Les codes sources de cette fonction sont donnés en annexe .6.

Afin de s'assurer que les résultats de l'algorithme de Waters peuvent être considérés comme fiables, nous avons choisi de les comparer aux résultats d'un algorithme classique : l'approche de Monté-Carlo.

1.6 Algorithme de Monté-Carlo

1.6.1 Intérêt

L'algorithme de Monté-Carlo présente plusieurs avantages dans le calcul des primes d'assurance en invalidité dans un système à 3 états autonome, invalide, décès avec des taux de transition connus. Il permet une estimation précise des primes en simulant de nombreuses trajectoires stochastiques, il est flexible pour s'adapter à différentes situations d'assurance, rapide à implémenter et peut aider à comprendre les tendances et les comportements du système d'assurance grâce à la visualisation des résultats. En somme, c'est un outil puissant pour le calcul des primes d'assurance en prévoyance ([Arthur Charpentier, 2014](#)).

1.6.2 Approche générale

Pour calculer les primes d'assurance en prévoyance dans un système à trois états (autonome, invalide, décès) avec des taux de transition connus, il est nécessaire de suivre les étapes suivantes :

1. Identifier les variables d'entrée telles que l'âge de l'assuré, le montant de la couverture, la durée de la période d'assurance et les taux de transition entre les états.
2. Générer des nombres pseudo-aléatoires pour simuler les événements aléatoires du système.
3. Effectuer la simulation des durées passées dans les différents états (autonome, invalide ou décédé) à la fin de la période d'assurance pour chaque nombre pseudo-aléatoire, en fonction des taux de transition.
4. Déterminer pour chaque simulation la prime d'assurance en invalidité en fonction de la trajectoire de l'assuré et de la couverture d'assurance. Ensuite, il convient de cumuler les primes pour obtenir une estimation statistique du montant total de primes à payer pour l'ensemble des assurés, puis prendre sa moyenne pour le taux de prime à considérer.
5. Répéter le processus de simulation un grand nombre de fois pour obtenir une estimation statistique précise du montant total de primes à payer.

1.6.3 Approche pratique

Cette partie est essentiellement basée sur les algorithmes de simulation vus au cours [LACTU2230](#) au chapitre «modèles à sorties multiples».

Considérons un contrat d'invalidité simple et classique de durée n . Un assuré d'âge x qui souscrit à ce contrat doit payer des primes tant qu'il est autonome et recevra des prestations tant qu'il est invalide. Le taux de prime sera évalué en fonction du taux de prestation b définie par le principe actuariel.

La méthode de Monté-Carlo consistera donc à :

1. Simuler la trajectoire d'un assuré depuis l'âge x de signature du contrat jusqu'à l'âge $x + n$ ou son décès et ensuite évaluer le taux de prime.
2. Evaluer la valeur actuelle des primes futures pour 1€ de taux de prestation $\bar{a}_{x;n}^{aa}$.
3. Evaluer la valeur actuelle des prestations futures pour 1€ de taux de prestation $\bar{a}_{x;n}^{ai}$.
4. Calculer le taux de prime pour cet assuré (voir formule (1.5.23))
5. Répéter le processus de simulation pour un grand nombre d'assurés (par exemple 10 000 fois) avec les mêmes paramètres.
6. Estimer le taux de prime final en considérant la moyenne arithmétique de l'ensemble des simulations.

1.6.4 Algorithme

Valeur actuelle des primes

Considérons un assuré d'âge y ($y \geq x$) se trouvant dans l'état autonome a . Soient les variables suivantes :

- aa : la valeur actuelle d'une rente en autonomie à cet âge,
- v : le facteur d'actualisation,
- x : l'âge initial de l'assuré à sa signature du contrat.

L'objectif est de simuler T_a , son temps passé en autonomie et J_a , son état de sortie.

Le processus est décrit selon l'algorithme suivant :

- Soient u_i et u_d des réalisations indépendantes de variables aléatoires U_i et U_d telles que

$$U_j \sim \text{Uniform}(0, 1), \quad j \in \{i, d\}.$$

- Simuler des réalisations de $T_0^{(i)}$ et $T_0^{(d)}$ en résolvant les équations suivantes d'inconnues t

$$u_j = \exp\left(-\int_0^t \mu_{aj}(y + \varepsilon)d\varepsilon\right) \quad \text{pour } j = i, d.$$

- Calculer $T_a^{(j^*)} = \min\{T_0^{(i)}, T_0^{(d)}\}$, puis définir
 - $T_a = T_a^{(j^*)}$: le temps mis en autonomie,
 - $J_a = j^*$: l'état de sortie de l'autonomie ($j^* \in \{i, d\}$),
 - $y = y + T_a$: l'âge de l'assuré à la sortie de l'autonomie,
 - $aa = aa + v^{y-x} \cdot \int_0^{T_a} v^\varepsilon d\varepsilon$: nouvelle valeur actuelle des primes à l'âge de sortie.

Cet algorithme est facilement adaptable pour un assuré se trouvant en invalidité.

Valeur actuelle des prestations

En procédant de façon similaire, considérons un assuré d'âge y ($y \geq x$) entrant en invalidité à cet âge.

Soient les variables suivantes :

- ai : la valeur actuelle d'une rente en invalidité à cet âge,
- v : le facteur d'actualisation,
- x : l'âge initial de l'assuré à sa signature du contrat.

L'objectif est de simuler T_i , son temps passé en invalidité et J_i , son état de sortie.

Le processus est décrit selon l'algorithme suivant :

- Soient u_a et u_d des réalisations indépendantes de variables aléatoires U_a et U_d telles que

$$U_j \sim \text{Uniform}(0, 1), \quad j \in \{a, d\}.$$

- Simuler des réalisations de $T_i^{(a)}$ et $T_i^{(d)}$ en résolvant les équations suivantes d'inconnues t

$$u_j = \exp \left(- \int_0^t \mu_{ij}(y + \varepsilon, \varepsilon) d\varepsilon \right) \quad \text{pour } j = a, d.$$

- Calculer $T_i^{(j^*)} = \min\{T_i^{(a)}, T_i^{(d)}\}$, puis définir
 - $T_i = T_i^{(j^*)}$: le temps mis en invalidité,
 - $J_i = j^*$: l'état de sortie de l'invalidité ($j^* \in \{a, d\}$),
 - $y = y + T_i$: l'âge de l'assuré à la sortie de l'invalidité,
 - $ai = ai + v^{y-x} \cdot \int_0^{T_i} v^\varepsilon d\varepsilon$: nouvelle valeur actuelle des primes à l'âge de sortie.

Par suite, il faut répéter ces deux procédures jusqu'à l'âge $x + n$ ou au décès de l'assuré afin d'avoir les valeurs actuelles $\bar{a}_{x;n}^{aa}$ et $\bar{a}_{x;n}^{ai}$, puis déduire la prime de cet assuré.

1.7 Applications aux taux tests

1.7.1 Enoncé

Dans cette section, nous allons mettre en pratique les algorithmes de Waters et Monté-Carlo décrits précédemment en utilisant des taux de transition simples (voir annexe .3) basés sur des exercices vus en classe. L'objectif ici est de vérifier la convergence des deux méthodes et de choisir le pas de discrétisation optimal Δ pour Waters.

A partir des codes R, nous avons calculé les différentes primes d'un contrat d'invalidité classique de durée dix(10) ans pour un taux de prestation de 12.000 et un taux d'actualisation fixé à 3%. Nous avons présenté des primes en fonction des paramètres choisis.

1.7.2 Analyse et interprétation

MONTE -CARLO	Age			
	25	30	35	40
lower	136,597	150,546	158,239	177,517
prime	138,553	152,614	160,381	179,829
upper	140,511	154,684	162,525	182,144

FIGURE 1.7.1 – *Primes Monte-carlo*

WATERS		pas de discretisation		
		1/12	1/52	1/360
Age	25	133,630	128,421	128,516
	30	149,256	144,672	144,810
	35	156,361	152,459	152,599
	40	171,267	168,363	168,532

FIGURE 1.7.2 – *Primes Waters en fonction du pas de discrétisation*

Le tableau 1.7.1 présente les primes obtenues en utilisant la méthode de Monte-Carlo avec un intervalle de confiance de $\alpha = 10\%$, tandis que le tableau 1.7.2 présente les primes obtenues en utilisant la méthode de Waters avec différents pas de discrétisation.

En examinant les différentes valeurs, nous constatons que les primes obtenues par les deux méthodes, bien qu'elles diffèrent, sont proches. Les primes augmentent en fonction de l'âge, qui représente ici le facteur principal de risque. Nous avons observé que l'algorithme de Waters est stable à partir d'un pas de discrétisation $\Delta = 1/52$ (1 semaine). Cependant, pour des pas de discrétisation plus grands, les primes générées par Waters ne suivent aucune logique mathématique et diminuent jusqu'à atteindre des valeurs négatives. Par exemple, pour $\Delta = 1/3$, la prime obtenue est de -433.45 , comme indiqué dans la relation (1.5.13) soulignée à la remarque 2. En essayant de contourner ce problème, nous avons utilisé l'équation avec intégrale (1.5.11) mais le problème demeure toujours causé par l'inversion matricielle des relations (1.5.21) et (1.5.22).

Sur la base de ces résultats, nous avons déduit que le pas de discrétisation optimal pour assurer la stabilité de l'algorithme de Waters est d'une semaine.

En utilisant un pas de discrétisation d'une semaine avec l'algorithme de Waters, nous avons appliqué ce dernier à un contrat pour un assuré de 30 ans, ce qui nous a permis

d'obtenir les résultats présentés dans la figure 1.7.3 pour les probabilités de transition.

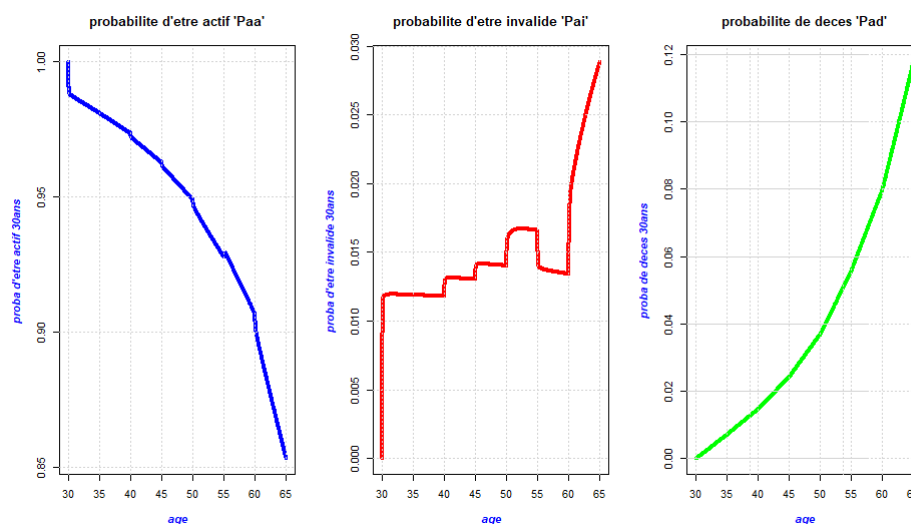


FIGURE 1.7.3 – Probabilités de transition par l'algorithme de waters

La forme des probabilités d'être actif semble naturelle, car elle décroît avec l'âge. Celle des probabilités de transition vers l'invalidité présente une allure légèrement complexe mais facile à comprendre. En effet, la courbe montre une croissance des probabilités d'invalidité de 30 à 55 ans environ, suivie d'une diminution jusqu'à 60 ans avant de suivre une tendance normale due au vieillissement. La décroissance des probabilités entre 55 et 60 ans semble loin d'être naturelle, elle peut être due à nos taux tests utilisés ou à un autre phénomène qu'on pourra expliquer dans le cadre des taux réels dans la partie application.

Nous notons que la non-régularité des courbes est due aux taux tests très discontinus. En effet, nos taux tests sont des fonctions en escaliers, ce qui agit directement sur l'allure des courbes de probabilités. Nous reviendrons sur ces analyses dans la partie application avec des taux réels du *Continuous Mortality Investigation Bureau*¹ (**CMIB**) qui sont continus.

1.7.3 Origine de la sous-estimation

Théorique

La sous-estimation des primes par l'algorithme de Waters peut être justifiée en examinant les différentes approximations utilisées dans sa construction. Voici deux raisons principales qui expliquent cette sous-estimation :

1. <https://www.actuaries.org.uk/>

1. L'algorithme repose sur l'utilisation de deux théorèmes d'analyse numérique pour établir des égalités et introduire certaines variables. Cependant, ces égalités sont ensuite approximées en utilisant des valeurs de milieu d'intervalle. Ces approximations numériques peuvent introduire des erreurs et conduire à des sous-estimations ou des sur-évaluations des primes.
2. La relation récursive utilisée dans l'algorithme pour calculer les probabilités de transition (conformément à l'équation (1.5.13)) dépend fortement du pas de discrétisation Δ . Pour obtenir des résultats significatifs, il est important que Δ soit choisi de manière appropriée et suffisamment petit. Si Δ est choisi de manière trop grande, cela peut entraîner des probabilités complètement irréalistes, ce qui se traduirait par des primes également irréalistes, voire négatives.

En résumé, les approximations numériques utilisées dans l'algorithme de Waters ainsi que le choix adéquat du pas de discrétisation sont des facteurs clés qui peuvent contribuer à la sous-estimation des primes.

Pratique

Dans cette sous-section, nous allons examiner de manière pratique l'origine de la sous-estimation des primes dans l'algorithme de Waters. Pour ce faire, nous allons effectuer une évaluation manuelle de la prime exacte sur une période d'un an.

En utilisant l'hypothèse de "no differential mortality" vue dans le cours [LACTU2230](#) et appliquée aux taux tests constants par morceaux, nous obtenons la prime exacte sur un contrat d'une durée d'un an pour un assuré âgé de 25 ans :

$$\pi_{exacte} = \int_0^1 {}_t p_x^{ai} v(0, t) dt = 122.8851$$

En évaluant cette prime avec l'algorithme de Waters, nous obtenons $\pi_{Waters} = 111.75102$.

La différence entre ces deux primes est de 11.13408, ce qui est proche de la différence observée des primes sur le contrat d'une durée de dix ans, qui est de 10.037. Cette dernière tient compte des différentes probabilités de transition liées à l'âge et du temps de cotisation.

Nous constatons donc que la perte de prime causée par l'algorithme de Waters est due au fait qu'il sous-évalue les probabilités de transition ${}_t p_x^{ai}$ et surestime probablement les ${}_t p_x^{aa}$.

Nous remarquons également une différence entre les primes évaluées sur une durée d'un an et celles évaluées sur dix ans. Cela s'explique par le fait que le risque d'invalidité évolue avec l'âge et que les taux de primes sur les contrats de dix ans sont des primes lissées.

1.8 Limites

L'algorithme de Waters est considéré comme une référence dans la pratique, car il permet de résoudre des équations intégrales-différentielles et donne des résultats acceptables. Cependant, certaines limites peuvent affecter les grands portefeuilles d'assurance en termes de primes et de réserves :

- L'algorithme manque de transparence et de raisonnement probabiliste. Pour certains pas de discrétisation, il peut donner des valeurs complètement irréalistes (par exemple, des primes négatives dans notre cas conformément à la remarque sur l'algorithme de Waters).
- Il est principalement basé sur des raisonnements analytiques en discrétisant des fonctions avec des approches purement numériques (par exemple, le théorème des accroissements finis, la valeur moyenne, etc.).
- La précision de l'algorithme dépend de la taille du pas de discrétisation Δ , plus ce dernier est petit, plus la précision est grande, mais cela augmente considérablement le temps de calcul (environ quatre fois plus de temps de calcul si Δ est divisé par deux).
- Les probabilités ${}_t p_{x,z}^{ia}$ et ${}_t p_{x,z}^{id}$ de sortie d'invalidité ne sont pas fournies par l'algorithme. Cela rendra complexe, voire impossible, la tarification des produits d'assurance moins simples qui ont des prestations liées à la sortie de l'état d'invalidité.

Chapitre 2

Modèle multi-états : méthode des états fictifs

Introduction

Ce chapitre est consacré à la théorie générale des opérations d'assurance de personnes dans le cadre markovien. Les processus de Markov fournissent un cadre de travail souple et efficace, autorisant de nombreux développements actuariels.

Malheureusement, l'hypothèse de Markov ne permet pas de prendre en compte l'influence du temps passé dans un état sur le futur du processus, ce qui s'avère très nécessaire pour l'état invalide dans notre cas. Le but de cette section est de trouver une solution pour conserver les calculs markoviens de l'état i tout en prenant en compte l'âge de l'assuré et son temps passé en invalidité.

Cette approche consiste à passer à une échelle de temps discret en substituant le processus semi-markovien par une chaîne de Markov (processus de Markov à temps discret) avec des états artificiels. Notre objectif est de mettre en pratique l'approche théorique développée par [E. Pitacco \(1994\)](#). Pour ce faire, nous commençons par rappeler quelques notions sur les chaînes de Markov nécessaires pour la suite.

2.1 Chaîne de Markov à temps discret

Définition 2.1.1. On considère un processus stochastique en temps discret $\{X_n, n = 0, 1, 2, \dots\}$ dont l'espace d'états χ (l'ensemble de tous les états possibles) est dénombrable. On va habituellement numéroter les états par des entiers non négatifs, et ainsi supposer que $\chi = \{0, 1, \dots, r\}$ (fini) ou $\chi = \{0, 1, \dots, r\}$ (infini). Ce processus est une chaîne de Markov si

$$\mathbb{P}[X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0] = \mathbb{P}[X_{n+1} = j | X_n = i] = P_{i,j}.$$

En d'autres mots, la loi de probabilité du prochain état X_{n+1} conditionnellement à l'histoire passée ne dépend que de l'état actuel X_n . Cette propriété d'"oubli du passé" constitue la définition classique des chaînes de Markov, c'est un processus sans mémoire. Les $P_{i,j}$ sont les probabilités de transition de la chaîne. La chaîne de Markov est dite homogène si les probabilités ne dépendent pas de n .

Théorème 2.1.2. Soit $\{X_n, n = 0, 1, 2, \dots\}$ une chaîne de Markov, alors les probabilités de transition $P_{i,j}$ satisfont :

$$P_{i,j} \geq 0 \quad \forall i, j : \sum_{j=1}^{\infty} P_{i,j} = 1 \quad \forall i. \quad (2.1.1)$$

Ils sont les éléments de la matrice de transition P définie par :

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & \cdots \\ P_{1,0} & P_{1,1} & P_{1,2} & \cdots \\ P_{2,0} & P_{2,1} & P_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

NB : Le produit de deux matrices de transition est une matrice de transition.

Exemple : la marche aléatoire sur les entiers (voir figure 2.1.1) :

$$P_{i,i+1} = p = 1 - P_{i,i-1}, \quad \text{pour } i \in \mathbb{Z}. \quad (2.1.2)$$

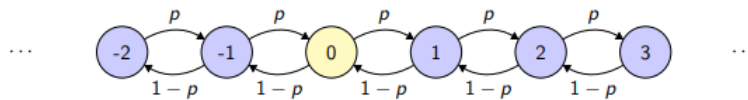


FIGURE 2.1.1 – marche aléatoire sur les entiers

Théorème 2.1.3. Equation de Chapman-Kolmogorov

Soit une chaîne de Markov homogène,

$$P_{i,j}^{(n)} = \mathbb{P}[X_n = j | X_0 = i],$$

la probabilité de passer de i à j en exactement n étapes. L'équation de Chapman-Kolmogorov est donnée par :

$$P_{i,j}^{(n+m)} = \sum_{k=0}^{\infty} \mathbb{P}[X_{n+m} = j | X_n = k] \mathbb{P}[X_n = k | X_0 = i] = \sum_{k=0}^{\infty} P_{i,k}^{(n)} P_{k,j}^{(m)}$$

En notation matricielle, si $P^{(n)}$ est la matrice contenant les $P_{i,j}^{(n)}$, cela donne

$$P^{(n+m)} = P^{(n)} \times P^{(m)}.$$

En particulier, $P^{(2)} = P \times P$ et par induction sur n , on a $P^n = P^{(n-1)} \times P = P^n$.

Similaire aux matrices donnant les nombres de chemins de longueur n entre chaque paire de sommets dans un graphe.

2.2 Modèle markovien : modèle des états fictifs

2.2.1 Intérêt

Comme souligné plus haut, dans notre modèle étudié, l'état invalide est semi-markovien, ce qui conduit préalablement à la résolution de systèmes intégraux-différentiels assez complexes pour les calculs des primes et réserves des assurés. Afin de simplifier les calculs, en 1991, Waters a proposé un algorithme permettant d'approximer les probabilités de transition semi-markoviennes à l'aide des taux de transition connus. Cet algorithme nous a permis de résoudre une grande partie du problème, mais il présente des limites : l'approche est très numérique, le temps de calcul quadruple dès que le pas est scindé en deux, les probabilités de transition partant de l'état invalide ne sont pas prises en compte dans l'algorithme, etc. De ce fait, il faut donc trouver une meilleure approche de calcul afin de résoudre les limites rencontrées avec l'algorithme de Waters.

Cependant, sachant tous les avantages fournis par un processus markovien, il est plus judicieux de trouver une méthode pour conserver cette hypothèse pour nos calculs. Notre approche étudiée dans ce mémoire consiste à passer à une échelle de temps discret en substituant le processus semi-markovien en une chaîne de Markov (processus de Markov à temps discret) avec des états artificiels par décomposition de l'état invalide i ([E. Pitacco, 1994](#)).

2.2.2 Présentation du modèle

Une approche unifiée des problèmes actuariels concernant un certain nombre de couvertures d'assurance dans le domaine des assurances de la personne (assurance-vie, couvertures d'invalidité, produits de soins à long terme, etc.) peut être réalisée grâce aux mathématiques des processus stochastiques de Markov, à la fois dans un contexte de temps continu et de temps discret. Les modèles qui en résultent sont généralement appelés modèles de Markov multi-états.

Par ailleurs, l'évolution d'un risque (par exemple, un assuré) peut être décrite en termes de présence du risque lui-même, à chaque instant, dans un certain état appartenant à un ensemble donné d'états, ou espace d'états. Formellement, nous désignons par $\mathcal{S}$ l'espace d'états. Nous supposons que $\mathcal{S}$ est un ensemble fini. Dans le cas d'une assurance invalidité, l'espace d'états (le plus simple) est le suivant (voir figure 1.2.1).

$$\mathcal{S} = \{a, i, d\}. \quad (2.2.1)$$

Considérons l'état i de notre modèle multi-état. Nous allons décomposer cet état en un ensemble de m états de Markov $\mathcal{S}'_i = \{i_0, i_1, i_2, \dots, i_m\}$, afin de discrétiser la durée effective semi-markovienne en temps discret. Nous conservons la même idée de discrétisation du temps (Δ étant un multiple de $1/52$), mais nous substituons le processus semi-markovien par une chaîne de Markov en créant des états fictifs en cascade (successions d'états artificiels).

$$i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow \dots i_{m_0}$$

- i_m correspond à une invalidité d'une durée totale de (au moins) $m\Delta$
- i_{m_0} correspond à une invalidité permanente, sans possibilité de revenir à l'état a .

Nous passons donc d'un processus semi-Markov à temps continu $X_t, t \geq 0$ décrivant la trajectoire à travers $\{a, i, d\}$ à une chaîne de Markov à temps discret $\{a, i_0, i_1, \dots, i_{m_0}, d\}$ avec un espace d'états $\{Y_j, j = 0, 1, \dots\}$ et des transitions possibles :

$$\begin{aligned} a &\rightarrow i_0, a \text{ ou } d \\ i_m &\rightarrow a, i_{m+1} \text{ ou } d \text{ pour } m = 0, 1, \dots, m_0 - 1 \\ i_{m_0} &\rightarrow i_{m_0} \text{ ou } d \end{aligned}$$

En effet, si Δ représente une semaine, on suppose implicitement que pendant une semaine notre assuré ne peut avoir qu'une seule invalidité. Ainsi, lors de la première semaine d'invalidité, l'assuré se trouve dans l'état i_0 , et les seules transitions possibles sont : retourner à l'état actif (a), décédé (d) ou passer à l'état suivant i_1 en raison d'un changement de taux de réactivation qui varie également en fonction du temps passé en invalidité (hypothèse semi-markovienne). Dans le cas où l'assuré continue de séjourner en invalidité, ce processus continuera jusqu'à atteindre le point de non-retour possible en activité i_{m_0} . Dans ce cas, la seule possibilité de sortie de cet état est le décès.

La figure 2.2.1, inspirée de [Ermano Pitacco \(1994\)](#), illustre bien le principe expliqué.

Cette différenciation des états d'invalidité a été initialement proposée par [Amsler \(1968\)](#). Une procédure de tarification basée sur un tel modèle est utilisée par les assureurs néerlandais ; voir [Gregorius \(1993\)](#), [Wolthuis \(1993\)](#). L'espace d'état $\mathcal{S}' = \{a, i_0, i_1, i_2, \dots, i_{m_0}, d\}$ résultant permet de tenir compte des effets de la durée de l'invalidité sur la récupération et la mortalité des personnes invalides. Ce principe de subdivision du processus semi-markovien en plusieurs états markoviens oblige la chaîne de Markov, processus sans mémoire à la base, à mémoriser la durée d'invalidité de l'assuré.

Remarque 3. *Cette subdivision en plusieurs états markoviens permet de prendre en compte l'information sur la durée de l'invalidité de l'assuré, ce qui n'est pas*

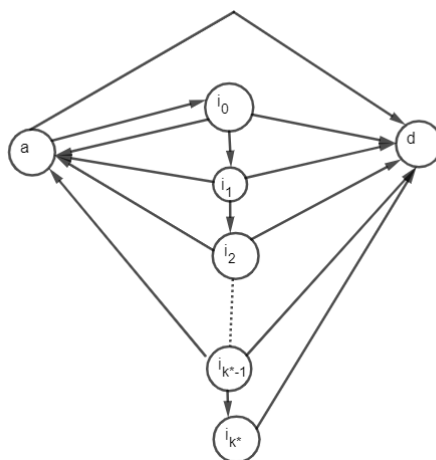


FIGURE 2.2.1 – processus en temps discret

possible dans une chaîne de Markov simple à espace d'état fini. Cela permet notamment d'ajuster les probabilités de transition en fonction de la durée d'invalidité, afin de refléter les effets sur la récupération et la mortalité des personnes invalides. Cela permet également de mieux modéliser les flux de trésorerie de l'assurance invalidité, ce qui est important pour la tarification et la gestion des risques pour les assureurs.

2.3 Taux de transition

2.3.1 Constance par morceaux

La raison de cette subdivision de l'état d'invalidité est que, comme on le sait, pour différentes durées depuis le début de l'invalidité, les taux de réactivation et de mortalité observés diffèrent ; c'est-à-dire que la subdivision nous permet de considérer des taux et des probabilités sélectionnés sans introduire formellement un processus semi-Markov (ce qui entraîne des difficultés majeures).

Lorsque le pas de subdivision Δ est assez petit, les taux de transition préalablement continus peuvent être supposés constants par morceaux (Denuit et Robert 2007, p.171), c'est-à-dire, pour un âge y donné,

$$\mu_{ij}(y + m\Delta + \varepsilon) = \mu_{ij}(y + m\Delta) \quad \forall \varepsilon \in (m\Delta; (m+1)\Delta) \quad \text{et} \quad m = 0, 1, 2, \dots \quad (2.3.1)$$

$\mu_{ij}(y + m\Delta)$ est ainsi le taux de transition relatif à l'intervalle de temps $(m\Delta; (m+1)\Delta)$.

Le modèle requiert les taux de transition suivants :

- $\mu_{ad}(y)$: taux de mortalité d'un assuré actif d'âge y ,
- $\mu_{ai_0}(y)$: taux de passage actif-invalidé d'un assuré d'âge y ,
- $\mu_{i_ma}(y)$: taux de réactivation d'un assuré d'âge y invalide de durée $m\Delta$,
- $\mu_{i_md}(y)$: taux de mortalité d'un assuré d'âge y invalide de durée $m\Delta$,
avec $m = 0, 1, \dots$

2.3.2 Raison scientifique du modèle étudié

En ce qui concerne la réactivation de l'assuré, l'expérience de morbidité suggère pour tout y donné (E. Pitacco 2014) :

$$\mu_{i_0a}(y) > \mu_{i_1a}(y) > \dots > \mu_{i_ma}(y) \quad (2.3.2)$$

c'est-à-dire, le taux de réactivation devient de plus en plus faible avec le temps passé en invalidité. On peut donc supposer qu'il existe un moment $m_0\Delta$ où ce taux est tellement faible qu'il est supposé nul à partir de ce moment. Par la suite, on considère donc que :

$$\begin{aligned} \mu_{i_ma}(y) &> 0, & \text{pour } m = 0, 1, \dots, (m_0 - 1) \\ \mu_{i_ma}(y) &= 0 & \text{pour } m \geq m_0 \end{aligned} \quad (2.3.3)$$

L'état i_{m_0} considéré est une hypothèse prudente, en effet ayant assez durée en invalidité, la possibilité d'être à nouveau actif, même si elle n'est pas totalement nul devient très faible donc approximativement nul, cet état est donc considéré comme une invalidité permanente par prudence.

Se basant sur la relation (2.3.3), on justifie donc scientifiquement le modèle étudié. Et on remarque dans cette chaîne de Markov :

- L'état d est absorbant,
- L'état m_0 est semi-absorbant,
- Les états $\{a, i_0, i_1, \dots, i_{m_0-1}\}$ sont transitoires.

2.4 Probabilités de transition

Le modèle probabiliste que nous étudions suppose que les probabilités de transition pour un assuré actif d'âge x à l'émission de la police vers n'importe quel âge y ($y \geq x$) ne dépendent que de l'état actuel de l'assuré à cet âge, et sont donc indépendantes de l'historique de sa santé avant cet âge (processus sans mémoire).

Des modèles plus complexes peuvent être construits en prenant en compte l'historique de santé de l'assuré (son état de santé passé, son patrimoine génétique, etc.), mais dans ce cas l'état actif a ne serait plus markovien. Cependant, cela ne sera pas étudié dans notre travail et pourra faire l'objet d'une autre recherche.

Pour effectuer les calculs de primes et de réserves, nous avons besoin des probabilités de transition d'un état à l'autre, ce qui implique implicitement les taux de transition (voir l'équation 1.4.1). Cependant, le calcul des probabilités de transition ${}_t p_x^{aa}$ et ${}_t p_x^{ai}$ peut être complexe, ce qui conduit à l'utilisation de méthodes approximatives pour simplifier leur résolution.

Afin de calculer les probabilités de transition sur une période de temps arbitraire t (qui est supposée être un multiple de Δ dans notre méthode de subdivision), il nous faut connaître les probabilités de transition à chaque intervalle de temps $((m-1)\Delta; m\Delta)$ $m = 1, 2, \dots, \frac{t}{\Delta}$.

2.4.1 Probabilités de transition entre deux états successifs

Rappelons que dans notre modèle, nous avons émis l'hypothèse selon laquelle au plus une transition est possible dans un intervalle de temps de longueur Δ . Ce qui a pour conséquence, les probabilités de transition de l'état a vers a et entre deux états fictifs d'invalidité sur un intervalle $(m\Delta, (m+1)\Delta)$ $m = 1, 2, \dots$ sont assimilables à des probabilités de séjour sur cet intervalle.

Pour ce faire, nous commençons par définir les probabilités initiales de transition de notre chaîne de Markov avec le pas de temps Δ .

Pour $\{Y_j, j = 0, 1, \dots\}$ à l'âge y atteint par l'assuré, les probabilités de transition depuis l'état a sont :

$$\begin{aligned} P_{a,i_0} &= P[Y_{j+1} = i_0 | Y_j = a] = {}_\Delta p_y^{ai} \approx \int_0^\Delta {}_t p_y^{\bar{a}\bar{a}} \mu_{ai}(y+t) {}_{\Delta-t} p_{y+t,0}^{\bar{i}\bar{i}} dt \\ P_{a,a} &= {}_\Delta p_y^{aa} \approx {}_\Delta p_y^{\bar{a}\bar{a}} = \exp \left(- \int_0^\Delta \mu_{ai}(y+t) + \mu_{ad}(y+t) dt \right) \\ P_{a,d} &= {}_\Delta p_y^{ad} \approx \int_0^\Delta {}_t p_y^{\bar{a}\bar{a}} \mu_{ad}(y+t) dt \end{aligned} \quad (2.4.1)$$

Ensuite, celles partant du premier état fictif i_0 sont :

$$\begin{aligned} P_{i_0,a} &= {}_\Delta p_{y,0}^{ia} \approx \int_0^\Delta {}_t p_{y,0}^{\bar{i}\bar{i}} \mu_{ia}(y+t,0) {}_{\Delta-t} p_{y+t}^{\bar{a}\bar{a}} dt \\ P_{i_0,i_1} &= {}_\Delta p_{y,0}^{ii} \approx {}_\Delta P_{y,0}^{\bar{i}\bar{i}} = \exp \left(- \int_0^\Delta \mu_{ia}(y+t,0) + \mu_{id}(y+t,0) dt \right) \\ P_{i_0,d} &= {}_\Delta p_{y,0}^{id} \approx \int_0^\Delta {}_t p_{y,0}^{\bar{i}\bar{i}} \mu_{id}(y+t,0) dt \end{aligned} \quad (2.4.2)$$

Pour les états transitoires en invalidité $m = 1, 2, \dots, m_0 - 1$, on a :

$$\begin{aligned} P_{i_m, a} &= \Delta p_{y, m\Delta}^{ia} \approx \int_0^\Delta {}_t p_{y, m\Delta}^{\bar{i}\bar{i}} \mu_{ia}(y+t, m\Delta) {}_{\Delta-t} p_{y+t}^{\bar{a}\bar{a}} dt \\ P_{i_m, i_{m+1}} &= \Delta p_{y, m\Delta}^{ii} \approx {}_\Delta p_{y, m\Delta}^{\bar{i}\bar{i}} = \exp \left(- \int_0^\Delta \mu_{ia}(y+t, m\Delta) + \mu_{id}(y+t, m\Delta) dt \right) \\ P_{i_m, d} &= \Delta p_{y, m\Delta}^{id} \approx \int_0^\Delta {}_t p_{y, m\Delta}^{\bar{i}\bar{i}} \mu_{id}(y+t, m\Delta) dt \end{aligned} \quad (2.4.3)$$

Enfin, pour l'état semi-absorbant (invalidité permanente), on a :

$$\begin{aligned} P_{i_{m_0}, a} &= {}_s \Delta p_{y, m_0\Delta}^{ia} \approx 0 \quad \text{pour } s = 1, 2, \dots \\ P_{i_{m_0}, i_{m_0}} &= \Delta p_{y, m_0\Delta}^{ii} \approx {}_\Delta p_{y, m_0\Delta}^{\bar{i}\bar{i}} = \exp \left(- \int_0^\Delta \mu_{ia}(y+t, m_0\Delta) + \mu_{id}(y+t, m_0\Delta) dt \right) \\ P_{i_{m_0}, d} &= \Delta p_{y, m_0\Delta}^{id} \approx \int_0^\Delta {}_t p_{y, m_0\Delta}^{\bar{i}\bar{i}} \mu_{id}(y+t, m_0\Delta) dt \end{aligned} \quad (2.4.4)$$

En effet, le pas Δ étant assez petit, nous supposons que l'assuré ne peut effectuer au plus qu'une seule transition dans cette période. Ainsi, les probabilités de changement réel d'état sont considérées comme des probabilités d'une seule transition de i vers j et celles de transition entre états fictifs sont assimilées à des probabilités de séjour. L'état i_{m_0} étant considéré comme une invalidité permanente, la probabilité de réactivation est sensiblement nulle et la probabilité $P_{i_{m_0}, i_{m_0}}$ est approximativement égale à une probabilité de séjour.

Remarque 4. *L'hypothèse selon laquelle au plus une transition est possible dans un intervalle de temps de longueur Δ assure que les probabilités de séjour après la première transition ${}_{\Delta-t} p_{y+t}^{\bar{a}\bar{a}}, {}_{\Delta-t} p_{y+t, 0}^{\bar{i}\bar{i}}$ sont sensiblement égales à 1. Par conséquent, les probabilités ci-dessus sont bien définies, c'est-à-dire :*

$$P_{a, a} + P_{a, i_0} + P_{a, d} = 1 \quad \text{et} \quad P_{i_m, a} + P_{i_m, i_{m+1}} + P_{i_m, d} = 1$$

Toutefois, pour se fixer les idées, nous évaluerons l'écart d'erreurs des probabilités dans la partie application.

2.4.2 Constance par morceaux des taux de transition

Dans cette partie, avec le pas Δ assez faible, et dans le but d'alléger les calculs, nous pouvons supposer que les taux de transition sont constants sur chaque intervalle de temps $(s\Delta, (s+1)\Delta)$ $s = 1, 2, \dots$ comme dans la première partie de la section précédente. Cette hypothèse supplémentaire nous permet de simplifier l'ensemble de calculs des intégrales.

Ainsi, les probabilités de transition depuis l'état a de la relation (2.4.1) deviennent :

$$\begin{aligned} P_{a,a} &\approx \exp\left(-\Delta(\mu_{ai}(y) + \mu_{ad}(y))\right) \\ P_{a,d} &\approx \mu_{ad}(y) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ai}(y) + \mu_{ad}(y))\right)}{\mu_{ai}(y) + \mu_{ad}(y)} \\ P_{a,i_0} &\approx \mu_{ai}(y) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ai}(y) + \mu_{ad}(y))\right)}{\mu_{ai}(y) + \mu_{ad}(y)} \end{aligned} \quad (2.4.5)$$

Ensuite, celles partant de i_0 de la relation (2.4.2) sont :

$$\begin{aligned} P_{i_0,i_1} &\approx \exp\left(-\Delta(\mu_{ia}(y, 0) + \mu_{id}(y, 0))\right) \\ P_{i_0,d} &\approx \mu_{id}(y, 0) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ia}(y, 0) + \mu_{id}(y, 0))\right)}{\mu_{ia}(y, 0) + \mu_{id}(y, 0)} \\ P_{i_0,a} &\approx \mu_{ia}(y, 0) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ia}(y, 0) + \mu_{id}(y, 0))\right)}{\mu_{ia}(y, 0) + \mu_{id}(y, 0)} \end{aligned} \quad (2.4.6)$$

Pour les états transitoires en invalidité $m = 1, 2, \dots, m_0 - 1$ de (2.4.3), on a :

$$\begin{aligned} P_{i_m,i_{m+1}} &\approx \exp\left(-(\mu_{ia}(y, m\Delta) + \mu_{id}(y, m\Delta))\right) \\ P_{i_m,d} &\approx \mu_{id}(y, m\Delta) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ia}(y, m\Delta) + \mu_{id}(y, m\Delta))\right)}{\mu_{ia}(y, m\Delta) + \mu_{id}(y, m\Delta)} \\ P_{i_m,a} &\approx \mu_{ia}(y, m\Delta) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ia}(y, m\Delta) + \mu_{id}(y, m\Delta))\right)}{\mu_{ia}(y, m\Delta) + \mu_{id}(y, m\Delta)} \end{aligned} \quad (2.4.7)$$

L'invalidité permanente de la relation (2.4.4) devient :

$$\begin{aligned} P_{i_{m_0},a} &\approx s\Delta P_{y,m_0\Delta}^{ia} \approx 0 \quad \text{pour } s = 1, 2, \dots \\ P_{i_{m_0},i_{m_0}} &\approx \exp\left(-(\mu_{ia}(y, m_0\Delta) + \mu_{id}(y, m_0\Delta))\right) \\ P_{i_{m_0},d} &\approx \mu_{id}(y, m_0\Delta) \cdot \frac{1 - \exp\left(-\Delta(\mu_{ia}(y, m_0\Delta) + \mu_{id}(y, m_0\Delta))\right)}{\mu_{ia}(y, m_0\Delta) + \mu_{id}(y, m_0\Delta)} \end{aligned} \quad (2.4.8)$$

En se basant sur la remarque précédente, nous pouvons utiliser des approches prudentes pour le calcul de certaines probabilités dans la pratique :

$$\begin{aligned} P_{a,i_0} &\approx 1 - P_{a,a} - P_{a,d} \\ P_{i_0,a} &\approx 1 - P_{i_0,i_1} - P_{i_0,d} \\ P_{i_m,a} &\approx 1 - P_{i_m,i_{m+1}} - P_{i_m,d} \\ P_{i_{m_0},d} &\approx 1 - P_{i_{m_0},i_{m_0}} \end{aligned}$$

Ces approches peuvent varier en fonction du produit en vu de rester prudentes. Toutefois, pour toujours se fixer les idées, nous évaluerons l'écart d'erreurs des probabilités dans la partie application.

Les calculs des sections précédentes suggèrent que l'on connaît l'état i occupé par l'assuré pour la durée $m\Delta$ et l'on cherche la probabilité d'occupation de l'état j à $(m+1)\Delta$. Ces calculs effectués de manière récursive sont très importants mais peuvent avoir des inconvénients sur les temps de calcul. Et en pratique, l'assureur n'a pas besoin de connaître l'état occupé sur chaque intervalle de temps $(s\Delta, (s+1)\Delta)$ $s = 1, 2, \dots$ par chaque assuré, le temps de calcul des primes et réserves serait interminable. Il est donc indispensable de pouvoir calculer directement la probabilité de passage d'un état i à j quelque soit la durée entre ces deux états.

2.4.3 Probabilités de l'état

2.4.3.1 Matrice de transition

Dans cette section, nous allons chercher la probabilité que l'assuré d'âge y se trouve dans l'un des états $a, i_0, i_1, \dots, i_{m_0}, d$ pendant la durée $j\Delta$ en passant par la matrice de transition $P(y)$ de notre modèle.

Cette matrice $P(y)$ de dimension $(m_0 + 3) \times (m_0 + 3)$ est définie par la relation 2.4.9 suivante :

$$P(y) = \begin{pmatrix} P_{a,a} & P_{a,i_0} & 0 & 0 & 0 & \dots & 0 & P_{a,d} \\ P_{i_0,a} & 0 & P_{i_0,i_1} & 0 & 0 & \dots & 0 & P_{i_0,d} \\ P_{i_1,a} & 0 & 0 & P_{i_1,i_2} & 0 & \dots & 0 & P_{i_1,d} \\ P_{i_2,a} & 0 & 0 & 0 & P_{i_2,i_3} & \dots & 0 & P_{i_2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{i_{m_0-1},a} & 0 & 0 & 0 & 0 & \dots & P_{i_{m_0-1},i_{m_0}} & P_{i_{m_0-1},d} \\ 0 & 0 & 0 & 0 & 0 & \dots & P_{i_{m_0},i_{m_0}} & P_{i_{m_0},d} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \quad (2.4.9)$$

Elle nous donne les états du départ en ligne et ceux d'arrivée en colonne. Elle contient l'ensemble des transitions en une période de temps Δ de l'état occupé i en ligne vers l'état occupé j en colonne. On remarque également qu'elle contient assez de 0, cela se justifie par le fait du nombre élevé de transitions non admis dans notre modèle.

Notons que du fait de la non-homogénéité des taux de transition, notre matrice n'est donc pas homogène sur une période $s > \Delta$. Cependant avec l'hypothèse de constante par morceaux, elle demeure homogène sur chaque intervalle de longueur Δ .

Et les équations de Chapman-Kolmogorov nous permettent d'écrire que :

$$P(y, y + \Delta) = P(y) \times P(y + \Delta) \quad (2.4.10)$$

2.4.3.2 Probabilités de l'Etat

Soit Y_j la variable aléatoire de l'état occupé par l'assuré pendant la durée $j\Delta$ et $e \in \{a, i_0, i_1, \dots, i_{m_0}, d\}$. Le contrat étant souscrit à l'état actif a et en appliquant l'équation de Chapman-Kolmogorov sur chaque intervalle de temps de longueur Δ , on obtient le vecteur de probabilités de e parcourant $\mathcal{S}' = \{a, i_0, i_1, \dots, i_{m_0}, d\}$ par l'équation :

$$P[Y_j = e] = (1 \ 0 \ 0 \ \dots \ 0)P(y) \times P(y + \Delta) \times \dots \times P(y + j\Delta) \quad (2.4.11)$$

Les probabilités définies ci-dessus portent sur l'état d'occupation de l'assuré, ce sont bien entendu des probabilités conditionnées qu'à l'âge y de l'ouverture du contrat, l'assuré est actif.

Si l'on note ce vecteur de probabilités L , alors la relation (2.4.11) fournie explicitement les probabilités de l'état d'occupation :

$$\begin{aligned} {}_j\Delta p_y^{aa} &= L_1 \\ {}_j\Delta p_y^{ai_m} &= L_{m+2} \quad m = 0, 1, \dots, m_0 - 1 \\ {}_j\Delta p_y^{ai_{m_0}} &= L_{m_0+2} \quad m \geq m_0 \\ {}_j\Delta p_y^{ad} &= L_{m_0+3} \end{aligned} \quad (2.4.12)$$

Il est important de souligner que l'un des avantages majeurs de cette méthode est la connaissance des ${}_j\Delta p_y^{ai_m}$. Cela permet une intégration facile des franchises temporelles dans les produits d'assurance.

De cette relation, on déduit que sur une période de longueur $j\Delta$, la probabilité de passage en invalidité est :

$${}_j\Delta p_y^{ai} = \sum_{m=0}^{m_0} {}_j\Delta p_y^{ai_m} \quad (2.4.13)$$

La connaissance de ces différentes probabilités nous permettra de calculer les primes et réserves.

Par ailleurs, le vecteur de probabilités L nous a fourni les probabilités de transition partant uniquement de l'état a . Il convient de noter que l'un des avantages de cette méthode est que nous pouvons obtenir toutes les probabilités de transition, y compris celles depuis l'état invalide, en utilisant le produit des matrices de transition. Cette possibilité n'est pas offerte par l'algorithme de Waters.

Notons :

$$M^k = P(y) \times P(y + \Delta) \times \dots \times P(y + k\Delta) \quad (2.4.14)$$

avec $M^0 = P(y)$

En plus des probabilités obtenues par le vecteur L , la matrice M^k nous fournit les probabilités de sortie d'invalidité ${}_k\Delta p_{y,m\Delta}^{i_m^a}$ et ${}_k\Delta p_{y,m\Delta}^{i_m^d}$ représentant respectivement la probabilité de passage de l'état invalide à l'état actif et celle de l'état invalide au décès pendant la m -ième semaine d'invalidité :

$$\begin{aligned} {}_k\Delta p_{y,m\Delta}^{i_m^a} &= M_{m+1,1}^k \quad m = 0, 1, \dots, m_0 \\ {}_k\Delta p_{y,m\Delta}^{i_m^d} &= M_{m+1,m_0+3}^k \quad m = 0, 1, \dots, m_0 \end{aligned} \quad (2.4.15)$$

et

$$\begin{aligned} {}_k\Delta p_{y,m\Delta}^{i_m^a} &= 0 \quad m > m_0 \\ {}_k\Delta p_{y,m\Delta}^{i_m^d} &= M_{m_0+2,m_0+3}^k \quad m > m_0 \end{aligned} \quad (2.4.16)$$

Avec ces informations supplémentaires, l'assureur peut présenter des produits plus variés et plus complets aux assurés avec des prestations et franchises adaptées.

2.5 Aspects Techniques

2.5.1 Prestations variées

Le modèle des états fictifs rend les prestations plus explicites. En effet, si l'une des parties souhaite des montants de prestations en fonction de la durée d'invalidité et/ou de l'âge de l'assuré, l'état invalidité étant subdivisé en plusieurs sous-états fictifs, nous pouvons fixer les montants de prestations de chacun de ces états fictifs. Ainsi, les prestations en fonction de la durée de l'invalidité deviennent équivalentes aux prestations en fonction des états fictifs. Donc, on associe aisément des prestations dans le modèle à trois états à des prestations différenciées dans notre cascade d'états invalides artificiels.

Un exemple concret d'un tel produit complet serait une offre comprenant les prestations suivantes :

1. Prestation liée aux sorties de l'état actif : cette prestation couvre les cas d'invalidité et de décès de l'assuré lorsqu'il cesse son activité professionnelle.
2. Prestation liée au séjour en invalidité : cette prestation prend en charge les dépenses liées au séjour prolongé de l'assuré en situation d'invalidité, notamment les frais médicaux, les soins et les traitements nécessaires.
3. Prestation liée aux sorties de l'état invalide : cette prestation couvre la période où l'assuré sort de l'état invalide pour revenir à une activité professionnelle ou en cas de décès.

En combinant ces différentes prestations, l'assureur peut offrir un produit d'assurance complet qui répond aux besoins spécifiques des assurés dans différentes situations, en leur fournissant une protection financière adéquate.

2.5.2 Franchise temporelle

Dans le contexte des prestations d'assurance, la durée de la franchise est traitée en remplaçant les paiements effectifs par des prestations d'un montant nul sur une période correspondante. Cela est réalisé en utilisant des états fictifs qui représentent la durée de la franchise.

Plus précisément, la période de franchise f est substituée par un certain nombre d'états fictifs dans le calcul des primes d'assurance. Pendant cette période, aucune prestation réelle n'est versée. Au lieu de cela, des prestations de montants nuls sont considérées, ce qui signifie que l'assuré ne reçoit aucun paiement pendant la durée de la franchise.

Ces états fictifs sont inclus dans le calcul des primes pour refléter la période de franchise et ajuster le coût global de l'assurance en conséquence. Ainsi, les primes d'assurance sont déterminées en tenant compte à la fois des prestations réelles et des prestations fictives de montants nuls correspondant à la période de franchise.

Cette approche permet de prendre en considération l'impact de la franchise sur les primes d'assurance, en ajustant les paiements effectifs par des prestations nulles pendant la durée spécifiée, représentée par les états fictifs.

2.5.3 Primes

Grâce à la méthode des états fictifs, on peut obtenir les approximations de $\bar{a}_{x;n|}^{aa}$ et $\bar{a}_{x;n|}^{ai}$ (en supposant que n est un multiple de Δ). Ainsi dans le but de conserver les mêmes approches de calcul que celui de Waters, nous décomposons l'intervalle $(0; n)$ par des intervalles de pas Δ ; et en approchant les intégrales avec la méthode des trapèzes (annexe 5), on obtient :

$$\begin{aligned}
 \bar{a}_{x;n|}^{aa} &= \int_0^n {}_t p_x^{aa} v^t dt = \sum_{j=0}^{\frac{n}{\Delta}-1} \int_{j\Delta}^{(j+1)\Delta} {}_t p_x^{aa} v^t dt \\
 &= \sum_{j=0}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_{j\Delta} p_x^{aa} v^{j\Delta} + {}_{(j+1)\Delta} p_x^{aa} v^{(j+1)\Delta} \right\} \\
 \bar{a}_{x;n|}^{ai} &= \sum_{j=0}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_{j\Delta} p_x^{ai} v^{j\Delta} + {}_{(j+1)\Delta} p_x^{ai} v^{(j+1)\Delta} \right\}.
 \end{aligned} \tag{2.5.1}$$

Les relations (2.5.1) nous permettent d'obtenir la prime d'un assuré actif :

$$\pi \approx b \frac{\sum_{j=0}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_j\Delta p_x^{ai} v^{j\Delta} + {}_{(j+1)\Delta} p_x^{ai} v^{(j+1)\Delta} \right\}}{\sum_{j=0}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_j\Delta p_x^{aa} v^{j\Delta} + {}_{(j+1)\Delta} p_x^{aa} v^{(j+1)\Delta} \right\}} \quad (2.5.2)$$

Les probabilités de transition sont données par les relations (2.4.13).

2.5.4 Réserves

Soit à déterminer les réserves en activité et invalidité d'un contrat classique passé de t (supposé multiple de Δ) années.

La réserve active est directement calculée sans complexe par :

$$\begin{aligned} V_a(t) = & b \sum_{j=\frac{t}{\Delta}}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_{j\Delta-t} p_{x+t}^{ai} v^{j\Delta-t} + {}_{(j+1)\Delta-t} p_{x+t}^{ai} v^{(j+1)\Delta-t} \right\} \\ & - \pi \sum_{j=\frac{t}{\Delta}}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_{j\Delta-t} p_{x+t}^{aa} v^{j\Delta-t} + {}_{(j+1)\Delta-t} p_{x+t}^{aa} v^{(j+1)\Delta-t} \right\} \end{aligned} \quad (2.5.3)$$

Par ailleurs, pour un assuré invalide de durée z , la réserve invalide est approximée de manière prudente par :

$$V_i(t, z) = b \sum_{j=\frac{t}{\Delta}}^{\frac{n}{\Delta}-1} \frac{\Delta}{2} \left\{ {}_{j\Delta-t} p_{x+t,z}^{\bar{ii}} v^{j\Delta-t} + {}_{(j+1)\Delta-t} p_{x+t,z}^{\bar{ii}} v^{(j+1)\Delta-t} \right\} + V_a(t) \quad (2.5.4)$$

La justification de cette approche est à la section 1.4 du chapitre 1.

La probabilité de séjour est explicitement connue avec la connaissance des taux de transition.

2.6 Application aux taux tests

2.6.1 Enoncé

Dans cette partie, nous allons appliquer la méthode décrite ci-dessus à des taux de transition simples comme dans la partie de l'algorithme de Waters. L'objectif est de vérifier la convergence des méthodes. La précision des calculs sera fonction du pas de discrétisation, plus le pas Δ est faible plus le calcul sera précis.

A partir des codes R, nous avons calculé les différentes primes d'un contrat d'invalidité classique de durée dix(10) ans pour un taux de prestations de 12.000 et un taux d'actualisation fixé à 3% comme dans le cas de l'algorithme de Waters.

2.6.2 Analyse et interprétation

Nous avons présenté des primes en fonction des paramètres choisis.

ETATS FICTIFS		pas de discretisation		
		1/12	1/52	1/180
Age	25	303,042	161,597	139,412
	30	339,685	181,815	154,963
	35	356,061	191,273	163,197
	40	390,376	210,811	180,127

FIGURE 2.6.1 – *Primes Markov en fonction du pas de discrétisation*

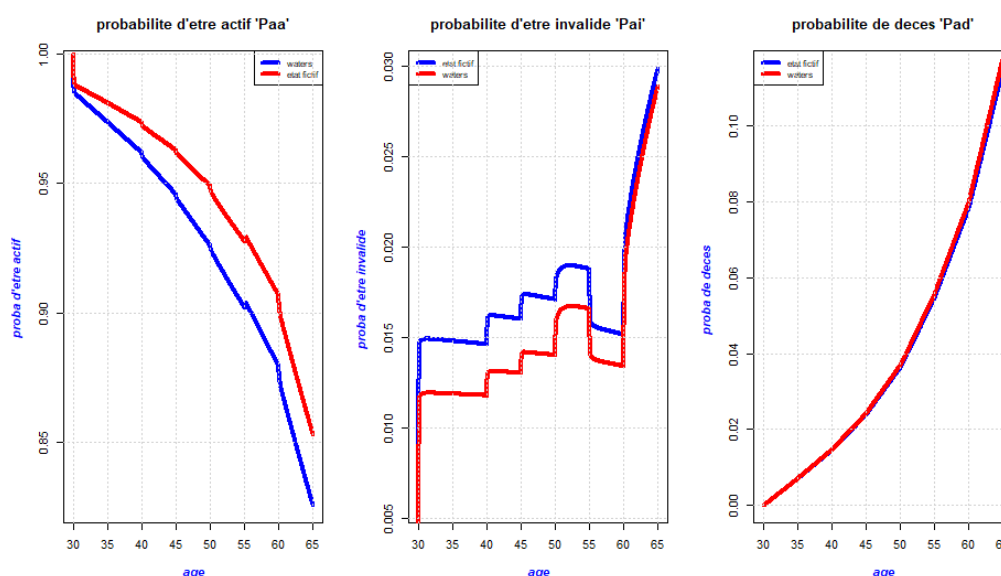


FIGURE 2.6.2 – *Probabilités de transition par l'algorithme des états fictifs et Waters*

En analysant les valeurs du tableau 2.6.1, on observe que la prime obtenue à l'aide de l'approche des états fictifs diminue avec la diminution du pas de temps. Les primes augmentent également avec l'âge, convergeant vers la prime calculée par la méthode de

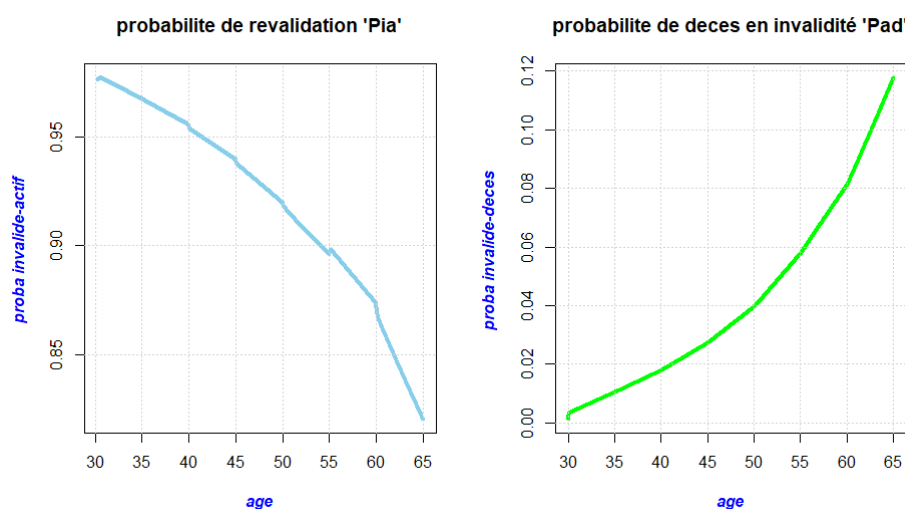


FIGURE 2.6.3 – Probabilités de sortie d'invalidité après une semaine par l'algorithme des états fictifs

Monté-Carlo (voir tableau 1.7.1).

En prenant la prime de Monté-Carlo comme référence, on peut conclure que les trois méthodes fournissent des valeurs proches, cependant la méthode de Waters sous-estime la prime tandis que la méthode des états fictifs la surestime de manière prudente. Les primes obtenues par cette approche sont toujours supérieures aux autres mais pour un pas de temps très faible (2 jours¹), les primes obtenues appartiennent à l'intervalle de confiance de celle de Monté-Carlo avec des écarts faibles comparés à ceux de Waters.

En utilisant un pas de discrétisation d'une semaine avec l'algorithme de Markov (algorithme des états fictifs), nous avons appliqué cette méthode à un contrat pour un assuré de 30 ans. Les résultats obtenus sont présentés dans les figures 2.6.2 et 2.6.3, qui montrent respectivement les probabilités de transition de l'état actif et les probabilités de sortie d'invalidité.

Les probabilités de transition de l'état actif, calculées à l'aide de la méthode de Markov, présentent une tendance similaire à celles obtenues avec la méthode de Waters. Cependant, les valeurs des probabilités de rester en activité sont généralement plus faibles, tandis que les probabilités de transition de l'état actif vers l'état invalide sont plus élevées. Cela a pour effet d'augmenter la prime calculée par la méthode de Markov par rapport à celle obtenue avec la méthode de Waters. Ainsi, la méthode de Markov se distingue par un caractère

1. Un pas de temps de deux jours est utilisé dans cette étude pour assurer un temps de calcul raisonnable

plus prudent dans l'évaluation des risques.

Les probabilités de sortie d'invalidité après une semaine (actif $\Delta p_{y,\Delta}^{ia}$ et décès $\Delta p_{y,\Delta}^{id}$) dans la figure 2.6.3 présentent des tendances similaires à celles des probabilités d'être actif Paa et de décès Pad . Cela confirme l'approximation selon laquelle, pour de courtes durées d'invalidité, ${}_{\tau-t}p_{x+t}^{aa} \approx {}_{\tau-t}p_{x+t,z}^{ia}$, ce qui justifie l'approximation utilisée dans le calcul de la réserve invalide (équation 1.4.6).

Chapitre 3

Application de l'approche des états fictifs

3.1 Présentation des données

3.1.1 Sources

Les données utilisées dans notre travail ont été obtenues du *Continuous Mortality Investigation Bureau*¹ (**CMIB**), une organisation de l'Institute and Faculty of Actuaries du Royaume-Uni. Elles sont essentiellement composées de taux de transition du modèle multi-états (actif-invalidé-décédé).

Les sources des observations utilisées pour leur calcul proviennent des données collectées annuellement auprès des compagnies d'assurance qui acceptent de les fournir. En contrepartie, elles ont la possibilité d'utiliser les résultats pour la tarification de leurs produits. Ces observations servent également de base de données pour les recherches académiques, ce qui est le cas dans notre étude.

Trois grandes périodes d'expérimentation ont été utilisées pour traiter les données collectées : 1975-1978, 1991-1998 et 2007-2016. Bien évidemment, l'expérience de 2007-2016 repose sur un volume de données plus important que les précédentes. Cependant, les tendances générales des taux de transition sont restées globalement les mêmes. Néanmoins, on observe une amélioration des taux de sortie de l'état invalide pour de petits délais de carence et une détérioration significative pour de longs délais.

Les taux de transition utilisés dans ce travail sont estimés par le CMIB à partir des données de la période 2007 – 2016. Ils ont été calculés en utilisant des méthodes découlant du maximum de vraisemblance, ils tiennent compte des paramètres tels que l'âge, le sexe de l'assuré, sa catégorie professionnelle (cols blancs, cols bleus, ...) et la période d'attente

1. <https://www.actuaries.org.uk/>

(délai de carence). Des détails sur la méthodologie sont disponibles dans le [CMI \(2010\)](#).

La méthode de calcul des taux de transition a été explicitée dans le [CMIR 12 \(1991\)](#), et les différentes formules qui ont été utilisées pour la création de ces taux sont basées sur l'[IPM 1991-98](#) et le [SM1975 – 78](#). Un outil Excel a été développé pour générer les différents taux de la dernière expérience, en se basant sur les documents [CMIR 12 \(1991\)](#), [CMI Working Papers 5](#) et [47](#).

À partir de ces formules, nous avons généré les taux de transition pour les hommes de la catégorie des cols blancs, afin de les comparer à ceux obtenus à l'aide de l'outil Excel et les résultats sont similaires. Les codes R utilisés pour cette génération sont donnés en annexe .6.

3.1.2 Structures des taux

À partir du code ayant permis de régénérer les différents taux, nous allons présenter l'allure de ces taux en fonction des paramètres choisis.

3.1.2.1 Taux de sortie d'activité

Pour des individus d'âge compris entre 20 et 60 ans, nous avons les taux de décès et d'invalidité des actifs suivants :

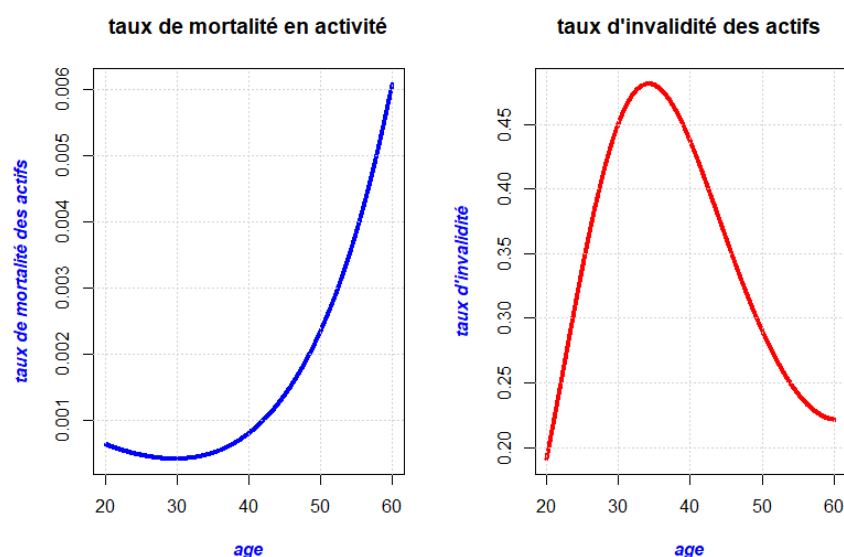


FIGURE 3.1.1 – *Taux de transition de sortie d'activité*

Nous pouvons remarquer que :

- Pour les taux de transition de l'état actif vers l'état décès, leur augmentation avec l'âge s'explique par le processus naturel du vieillissement. À mesure que les individus avancent en âge, ils deviennent plus vulnérables aux maladies, affections et risques liés à la santé, ce qui entraîne une augmentation de la probabilité de décès.
- En ce qui concerne les taux de transition de l'état actif vers l'état invalide, leur allure complexe peut être attribuée à différents facteurs :
 - La première partie de la courbe, marquée par une augmentation de l'entrée en invalidité entre environ 20 et 35 ans, peut être expliquée par des raisons telles que la propension des jeunes à prendre des risques. Les jeunes adultes ont tendance à être plus audacieux, à rechercher des sensations fortes et à s'engager dans des emplois ou des activités plus risqués, ce qui peut accroître le risque d'invalidité.
 - La deuxième partie de la courbe, caractérisée par une diminution des taux de transition à partir de cet âge, peut être attribuée à des facteurs tels que la prise de conscience, la maturité et la stabilité dans le travail. À mesure que les individus mûrissent, ils tendent à adopter des comportements plus prudents, à prendre des décisions plus réfléchies et à rechercher des emplois plus stables et moins risqués, ce qui réduit la probabilité d'invalidité.

Il est important de noter que ces explications sont générales et peuvent varier en fonction des contextes spécifiques et des caractéristiques des populations étudiées.

- La variation de ces taux est exclusivement due à l'âge de l'assuré, ce qui indique la propriété markovienne de l'état actif.

3.1.2.2 Taux de sortie d'invalidité

Ces taux peuvent être analysés sous deux angles : la sortie d'invalidité en fonction du temps mis et de l'âge².

Taux de réactivation

À partir de ce graphique, nous pouvons tirer les conclusions suivantes :

- **durée** : Les taux de réactivation sont très élevés au début de l'invalidité, ce qui indique une forte probabilité de récupérer la santé. Cependant, ces taux diminuent considérablement avec le temps passé en invalidité et deviennent presque nuls au-delà de 2 ans. Ainsi, il est quasiment impossible pour un assuré invalide depuis 2 ans de retrouver son état actif.
- **âge** : La courbe des taux présente une croissance pour les jeunes âgés d'environ 20 à 30 ans, puis une décroissance pour les autres tranches d'âge. Cette observation suggère que les taux de réactivation varient en fonction de l'âge de l'assuré, avec des valeurs plus élevées pour les personnes plus jeunes.

2. voir Annexe .4 pour des vues en 2D

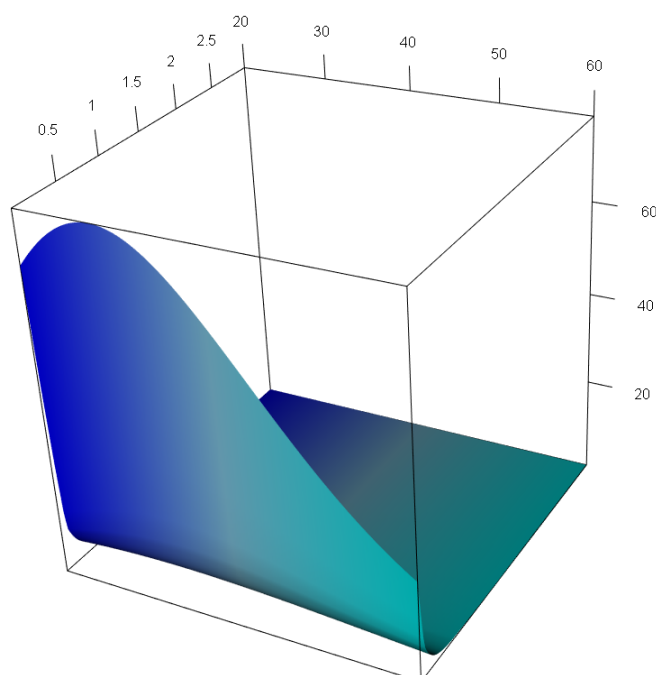


FIGURE 3.1.2 – *Taux de réactivation en fonction de l'âge et de la durée en invalidité*

En résumé, le graphique met en évidence que les taux de réactivation sont fortement influencés par la durée d'invalidité et l'âge de l'assuré. Les taux sont élevés au début de l'invalidité, mais diminuent progressivement avec le temps et varient selon l'âge.

Taux de décès des invalides

Ces courbes nous montrent que :

- **durée** : En début d'invalidité, les taux de mortalité augmentent en fonction du temps passé en invalidité (courbe 1 semaine < 1 mois < 6 mois) avant de diminuer à nouveau (courbe 2 ans < 6 mois). Cette diminution peut s'expliquer par le fait que l'assuré invalide s'adapte progressivement à son état en aménageant ses conditions de vie, ce qui permet à son organisme d'accepter son nouvel état.
- **âge** : Pour chaque durée passée en invalidité, la courbe de mortalité des invalides présente une tendance croissante, indépendamment de cette durée. Cela correspond à la croissance naturelle des taux de mortalité en fonction de l'âge. Cette croissance est indépendante de l'état occupé par l'assuré ainsi que du temps passé dans cet état. Elle dépend uniquement de l'âge.

La courbe de mortalité des invalides présente une augmentation au cours des trois premiers mois environ, puis diminue par la suite. La hausse initiale de la mortalité en début

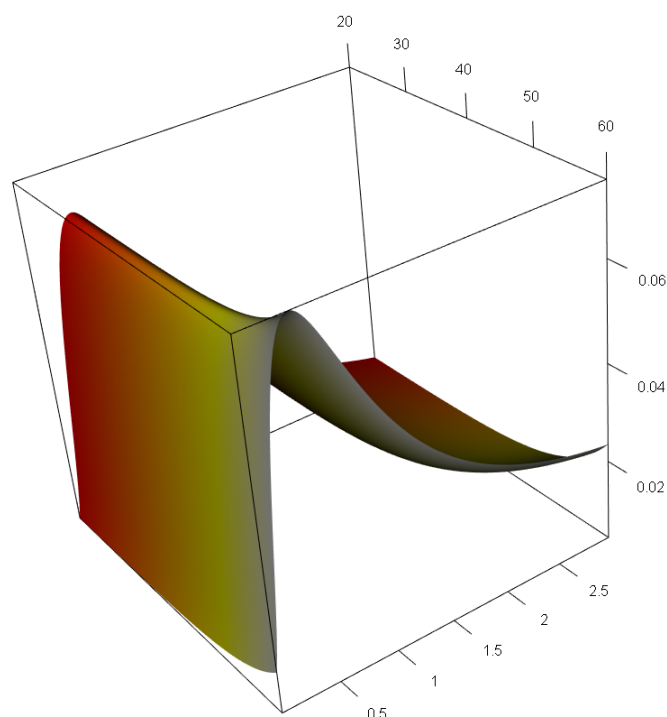


FIGURE 3.1.3 – *Taux de décès des invalides en fonction de l'âge et de la durée en invalidité*

d'invalidité peut être attribuée au changement brutal d'état subi par l'assuré. L'intervention médicale l'aide à survivre lors de la transition vers cet état, mais avec le temps, la détérioration de sa santé prend le dessus. Parallèlement, il est confronté à la réalité de sa situation, ce qui peut être difficile à accepter.

En résumé, ces courbes mettent en évidence la tendance croissante de la mortalité des invalides en fonction de l'âge, indépendamment de la durée d'invalidité. De plus, elles montrent l'impact du temps passé en invalidité sur les taux de mortalité, avec une augmentation initiale suivie d'une diminution progressive.

Ces observations confirment que l'hypothèse markovienne pour l'état actif et semi-markovienne pour l'état invalide sont conformes à notre base de données.

3.2 Probabilités de transition

Dans cette section, nous examinons l'effet de l'hypothèse selon laquelle les taux de transition sont considérés comme constants par morceaux dans un intervalle de temps discrétisé. Nous utilisons la méthode des états fictifs avec un pas de temps de Δ égal à une

semaine. Pour cela, nous avons mis en place deux fonctions. La première fonction permet de remplir la matrice de transition pour chaque âge donné en utilisant des taux de transition constants par morceaux. La deuxième fonction calcule les probabilités réelles en utilisant les taux de transition réels. Le code source en langage R pour ces fonctions est fourni en annexe.

En appliquant ces fonctions aux taux de transition fournis par ©CMIB, nous obtenons les probabilités présentées dans la figure 3.2.1 et les primes pour le même produit d'assurance classique décrit dans le chapitre précédent (prestation constante de $b = 12000$ et durée de $z = 10$ ans) comme indiqué dans le tableau 3.2.2.

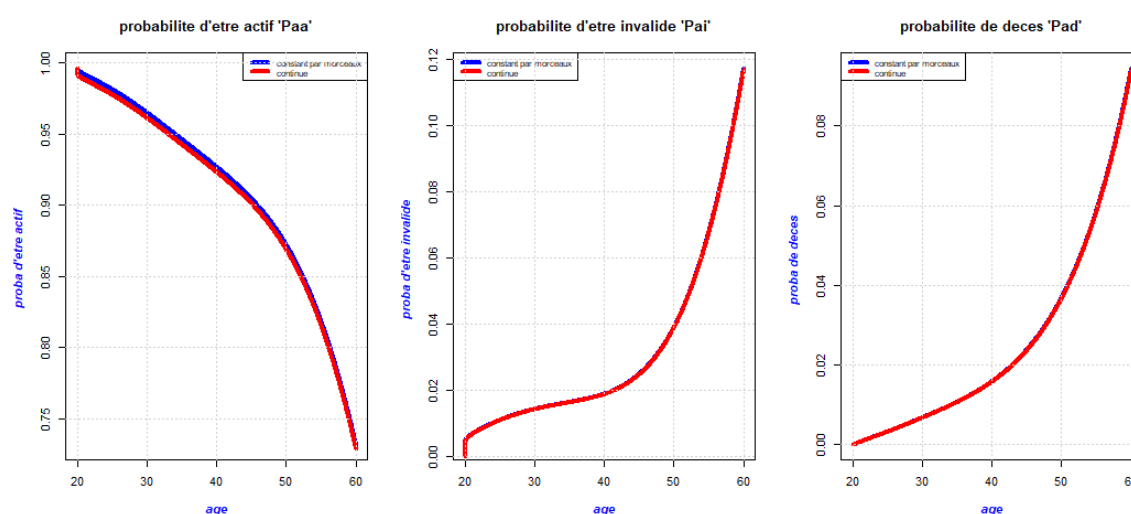


FIGURE 3.2.1 – Probabilités de transition obtenues par des taux constants par morceaux et des taux continus

L'analyse des résultats révèle une similitude remarquable entre les différentes courbes obtenues, qui se chevauchent presque partout. L'hypothèse selon laquelle les taux de transition sont constants par morceaux sur de petits intervalles de temps n'a aucun impact négatif sur les primes. De plus, cette hypothèse s'avère avantageuse à plusieurs égards. Elle se révèle prudente en termes de montant des primes, simplifie les codes de calcul et accélère le temps de calcul. Par conséquent, nous maintiendrons cette hypothèse pour la suite de notre analyse.

De plus, étant donné que la comparaison entre les probabilités de transition réelles et celles approximées par des fonctions constantes par morceaux concorde pratiquement en tous points, il n'est donc plus nécessaire d'examiner les approches prudentes des probabilités mentionnées à la section 2.4 du chapitre précédent.

ETATS		Primes: pas=1/52	
FICTIFS		taux constant	taux reels
		par morceaux	
Age	25	156,945	156,711
	30	181,890	181,609
	35	212,370	212,018
	40	275,883	275,384

FIGURE 3.2.2 – *Comparaison des Primes*

3.3 Primes et réseves

Cette section étudie l'effet de la période de franchise et du type de prestations sur les primes et les réserves de divers produits d'assurance. Les polices d'assurance étudiées sont souscrites par des assurés actifs à partir de l'âge de 20 ans, et la période de retraite est fixée à 60 ans. Le taux d'actualisation est constant et fixé à 3%. Le pas de discrétisation utilisé est d'une semaine pour un temps de calcul rapide dans notre étude. Toutefois, pour une utilisation pratique, nous conseillons aux utilisateurs un pas maximal de deux(2) jours.

3.3.1 Prestations

Prestation constante

Dans cette sous-section, nous examinons des produits d'assurance invalidité classiques avec un taux de prestation constant de montant $b = 12000$ indépendamment de l'âge et de la durée d'invalidité. Cependant, les prestations sont limitées à une période de 3 ans en cas d'invalidité :

- **produit 1** : prestation constante sans franchise,
- **produit 2** : prestation constante avec une franchise de 1 mois.

Prestation variée

Ici, nous examinons des produits d'assurance invalidité avec des prestations variables en fonction de l'âge et de la durée d'invalidité, définies comme suit :

$$b(x, z) = \begin{cases} 9000(1 - \frac{z}{3}) & \text{si } 20 \leq x < 35 \text{ et } z \leq 3, \\ 12000(1 - \frac{z}{3}) & \text{si } 35 \leq x < 45 \text{ et } z \leq 3, \\ 15000(1 - \frac{z}{3}) & \text{si } 45 \leq x < 60 \text{ et } z \leq 3, \\ 0 & \text{si } z > 3. \end{cases} \quad (3.3.1)$$

Il est observé que cette prestation est continue dans chaque tranche d'âge et diminue progressivement jusqu'à son épuisement pour une durée d'invalidité dépassant 3 ans, fonctionnant ainsi comme une assurance de Solde Restant Dû (SRD). Nous définissons ci-dessous les produits suivants :

- **produit 3** : prestation variée sans franchise,
- **produit 4** : prestation variée avec une franchise de 1 mois.
- **produit 5** : produit 2 + capital décès des actifs $C = 50000$,
- **produit 6** : produit 4 + capital décès des actifs $C = 50000$.

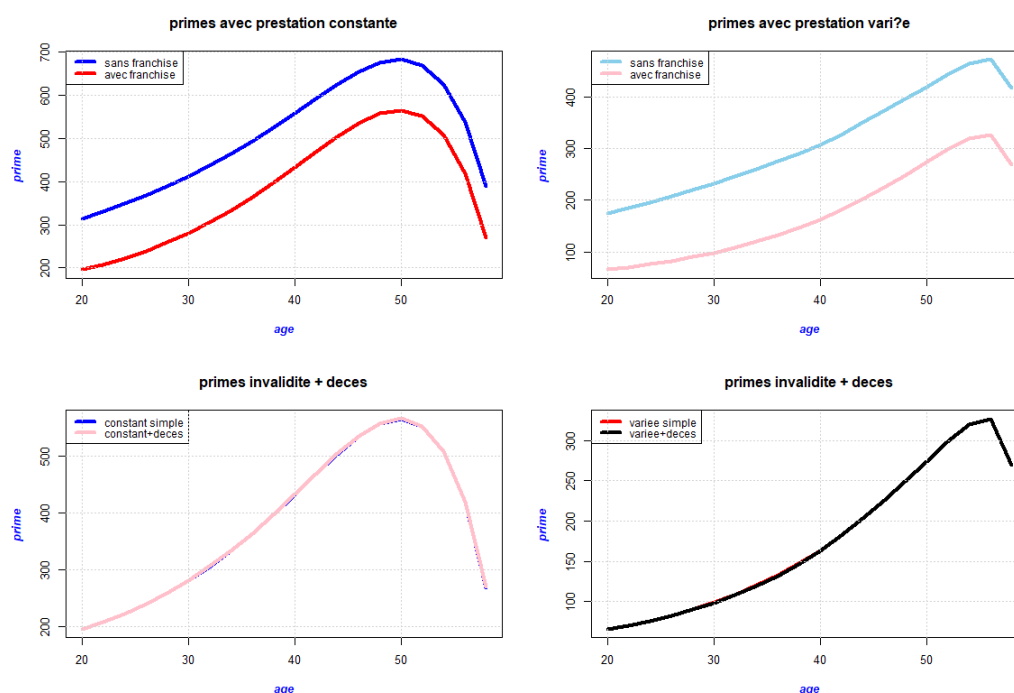
Rappelons qu'un des avantages de cette méthode est la possibilité d'inclure une franchise f dans le calcul des primes en considérant des prestations d'un montant nul sur un certain nombre d'états fictifs équivalents à cette durée.

3.3.2 Primes

Le résultat des primes³ des différents produits est présenté sur les figures ci-dessous, chaque taux de prime étant lissé sur sa période de couverture figure (3.3.1) :

1. Les taux de primes des assurés suivent une tendance similaire pour tous les produits. Ils augmentent jusqu'à atteindre un niveau de risque maximal, puis diminuent. Cette évolution peut s'expliquer de la manière suivante :
 - La partie croissante est due à l'augmentation du risque causée par l'augmentation des probabilités de transition vers l'invalidité et la réduction du temps de cotisation.
 - La partie décroissante s'explique par le fait que même si les probabilités de transition continuent d'augmenter vers l'invalidité, le contrat approche de son échéance et il n'y a pas de prestations à la fin du contrat, quelque soit l'état de l'assuré à cette date. Ainsi, le risque tend également à diminuer.
2. L'introduction d'une franchise d'un mois a un impact significatif sur les primes en les réduisant considérablement. Cela s'explique par les taux de réactivation élevés observés au début de l'invalidité. En effet, la majorité des personnes invalides retrouvent

3. un tableau de valeurs des primes en annexe .5

FIGURE 3.3.1 – *Primes des états fictifs*

ETAT FICTIF		AGE INITIAL ASSURE				
		20	25	30	35	40
RESERVE ACTIF APRES 10 ANS	produit 1	1830,881	1957,321	2023,809	1812,772	1005,828
	produit 2	1563,511	1920,373	2111,929	1915,730	1064,896
	produit 3	1059,365	1110,169	1053,676	1054,816	899,000
	produit 4	607,842	763,674	896,412	987,329	886,889

FIGURE 3.3.2 – *Réserves des actifs après 10 ans*

leur état actif dans de courtes périodes, ce qui conduit à une faible population à indemniser après un mois d'invalidité.

3. Les primes des prestations à valeurs constantes sont plus élevées. Comme mentionné précédemment, les prestations variables diminuent en fonction de la durée de l'invalidité jusqu'à atteindre zéro. Cela génère donc un coût plus faible pour l'assureur, d'où des primes plus élevées pour les prestations à valeurs constantes.
4. Les primes des produits combinant invalidité et décès sont sensiblement égales à celles des produits d'invalidité seuls. Cela s'explique par les très faibles probabilités

de passage des actifs au décès et peut servir à justifier également la modélisation à deux états $\{e_0, e_1\}$ pour les assurés moins âgés considérant les probabilités de décès négligeables comme vu au cours [LACTU2230](#).

5. Les réserves présentent des observations similaires à celles des primes, et elles peuvent être expliquées de la même manière.

En résumé, les caractéristiques des produits d'assurance invalidité, telles que les taux de primes, les franchises et les types de prestations, influencent les coûts pour l'assureur et les risques pour l'assuré, ce qui se reflète dans les modèles d'évolution des primes et des réserves.

Conclusion

La tarification en assurance invalidité est un domaine complexe où l'utilisation de formules exactes fermées est presque impossible en raison de la nature semi-markovienne de l'état invalide. Différents algorithmes ont été développés dans la littérature pour approximer l'évaluation des primes en invalidité, parmi lesquels l'algorithme de Waters est le plus connu.

L'implémentation de cet algorithme sur des données nous a permis d'obtenir des résultats intéressants en les comparant à ceux obtenus par simulation de Monté-Carlo bien que sous-estimant légèrement ses primes. Cependant, une analyse approfondie a révélé des limitations significatives dans son utilisation.

En effet, l'algorithme de Waters souffre d'un manque de transparence et d'un manque de raisonnement analytique basé sur des approches purement numériques. Sa précision dépend du pas de discrétisation et peut même conduire à des valeurs de primes négatives, ce qui ne suit aucune logique mathématique. De plus, il ne tient pas compte des probabilités de sortie d'invalidité, ce qui limite les produits proposés aux assurés.

Afin d'améliorer la méthode de tarification dans ce domaine et de résoudre les limitations de l'algorithme de Waters, nous avons implémenté la méthode proposée par [E. Pitacco \(1994\)](#). Cette méthode permet de conserver les calculs markoviens dans l'état invalide en substituant le processus semi-markovien par une chaîne de Markov avec des états fictifs en cascade. Cet algorithme, basé sur un raisonnement logique mathématique et probabiliste, nous a permis de résoudre tous les problèmes rencontrés, avec des résultats très intéressants en termes de calcul des primes et des réserves. De plus, il nous fournit l'ensemble des probabilités de transition, ce qui nous permet de proposer des produits plus variés et complets aux assurés, avec des prestations et des franchises temporelles adaptées.

Le modèle des états fictifs étudié s'est avéré prometteur, mais il présente une lenteur de calcul pour des pas très petits (1 jour). Un travail complémentaire consisterait à le comparer à l'approche assurance des dommages (fréquence-coût) proposée par [Denuit](#) dans le cours de [LACTU2230](#).

Le modèle étudié dans notre travail est un modèle supposant l'état actif markovien,

l'émission de la police ne dépend que de son état actuel. Des modèles plus complexes peuvent être construits en prenant en compte l'historique de santé de l'assuré (son état de santé passé, son patrimoine génétique, etc.), mais dans ce cas l'état actif a ne serait plus markovien. Cela pourra faire l'objet d'une autre recherche.

Bibliographie

Amsler, M.H. (1968) «Les chaînes de Markov des assurances vie, invalidité et maladie», T18ICA, München, 5, 731-746.

Amsler, M.H. (1988) «Sur la modélisation de risques vie par les chaînes de Markov», T23ICA, Helsinki, 3, 1-17.

Charpentier, A. et C. Dutang. 2012, *L'actuariat avec R* CRC.
<http://creativecommons.org/by-sa/3.0/fr/>.

CMI. 1991, «The analysis of permanent health insurance data», *Continuous Mortality Investigation Report 12*, n^o 12.

CMI. 2010, «An overview of the graduations of sickness inception and termination rates for the cmi individual income protection experience for 1991 – 1998 of males, occupation class 1», *Continuous Investigation Working Paper 48*, n^o 48.

CMI. 2020, «An overview of the graduations of sickness inception and termination rates for the cmi individual income protection experience for 2007 – 2016 of males, occupation class 1», *Continuous Investigation Working Paper 48*, n^o 48.

Denuit, M. et C. Robert. 2007, *Actuariat des Assurances de Personnes : modélisation, tarification et provisionnement*, Economica.

Denuit, M. 2021, «Health Insurance», Notes du Cours LACTU2230, Université Catholique de Louvain.

Gregorius, F. 1993, «Disability Insurance in the Netherlands», *Insurance : Mathematics and Economics*, vol. 13, n^o 2.

Haberman, S. et E. Pitacco. 1998, *Actuarial models for disability insurance*, CRC Press.

Pitacco, E. 1994, «Actuarial models for pricing disability benefits : Towards a unifying approach», *Insurance : Mathematics and Economics*, n^o 16, p.39-62.

Pitacco, E. 2014, «Health Insurance», *Basic Actuarial Models*. Springer.

Sian, C. 2017, «Approximation du Modèle sem-Markovien Actif-Invalide-Décédé», mémoire, Université Catholique de Louvain.

Waters, H.R. 1991, «The analysis of permanent health insurance data», *Continuous Mortality Investigation Report 12*, n^o 12, p.79-83. Part D.

Wolthius, H. (1994) «Life Insurance Mathematics (The Markovian model)», CAIRE Education Series, No.2, Bruxelles.

Appendices

.1 Théorème des accroissements finis

.1.1 Théorème des accroissements finis

Pour toute fonction réelle d'une variable réelle $f : [a, b] \longrightarrow \mathbb{R}$ (a et b réels tels que $a < b$), supposée continue sur l'intervalle fermé $[a, b]$ et dérivable sur l'intervalle ouvert $]a, b[$, il existe un réel c dans $]a, b[$ vérifiant :

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

.1.2 Théorème des accroissements finis généralisé

Pour toutes fonctions d'une variable réelle f et g continues sur le segment $[a, b]$, avec $a < b$, g gardant un signe constant sur $[a, b]$, il existe un réel c de $]a, b[$ tel que

$$\int_a^b f(x)g(x)dx = f(c) \int_a^b g(x)dx$$

.2 Intégration numérique : méthode des trapèze

Soit f fonction réelle positive $[a, b]$ (a et b réels tels que $a < b$), supposée continue sur l'intervalle fermé $[a, b]$, alors on a :

$$\int_a^b f(x)dx \approx \frac{b-a}{2} * (f(a) + f(b))$$

.3 Taux de transition test

.3.1 taux de transition actif-invalide

age	20<x<25	25<x<30	30<x<35	35<x<40	40<x<45	45<x<50
taux	0.20	0.34	0.45	0.48	0.43	0.36

age	50<x<55	55<x<60	60<x
taux	0.29	0.24	0.22

.3.2 taux de transition invalide-actif en semaine

$\frac{\text{age} \Rightarrow}{\text{durée} \Downarrow}$	$20 < x < 30$	$30 < x < 40$	$40 < x < 50$	$50 < x < 60$	$60 < x$
$0 < z < 12$	7.1	9.87	9.52	7.15	4.72
$12 < z < 24$	3.38	4.47	4.10	2.92	1.83
$24 < z < 36$	1.8	2.28	2	1.36	0.82
$36 < z < 52$	0.77	0.91	0.74	0.47	0.26
$52 < z$	0.42	0.45	0.34	0.19	0.1

.3.3 taux de transition invalide-décès en semaine

$\frac{\text{age} \Rightarrow}{\text{durée} \Downarrow}$	$20 < x < 30$	$30 < x < 40$	$40 < x < 50$	$50 < x < 60$	$60 < x$
$0 < z < 12$	0.04	0.04	0.04	0.04	0.05
$12 < z < 24$	0.086	0.086	0.087	0.088	0.098
$24 < z < 36$	0.082	0.082	0.082	0.083	0.094
$36 < z < 52$	0.07	0.07	0.071	0.073	0.083
$52 < z$	0.057	0.057	0.057	0.059	0.07

.3.4 taux de transition actif-décès

age	20<x<25	25<x<30	30<x<35	35<x<40	40<x<45	45<x<50
taux	0.00065	0.00048	0.00042	0.00051	0.00081	0.0014
age	50<x<55	55<x<60	60<x			
taux	0.0024	0.0039	0.0061			

.4 Taux réels de sortie d'invalidité

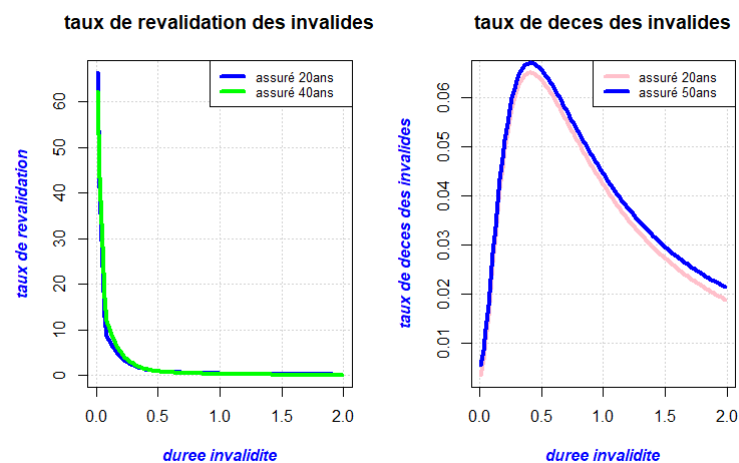


FIGURE 4.1 – *Taux de transition de sortie d'invalidité selon la durée*

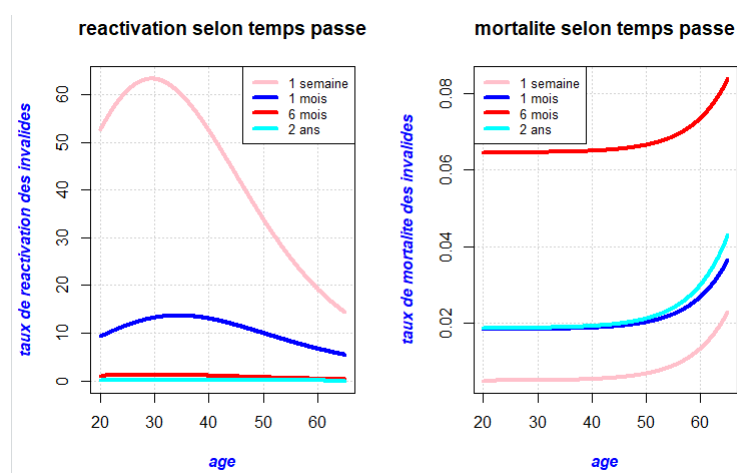


FIGURE 4.2 – *Taux de transition de sortie d'invalidité selon l'âge*

.5 Taux de primes de la méthode des états fictifs

ETATS		AGE			
FICTIFS		25	35	45	55
PRIMES	produit 1	358,419	478,353	640,720	587,635
	produit 2	230,899	348,570	520,159	469,624
	produit 3	201,565	269,590	364,068	472,929
	produit 4	78,888	125,682	214,116	326,328
	produit 5	231,128	348,911	520,737	470,098
	produit 6	79,117	126,024	214,694	326,802

FIGURE .5.1 – *primes des états fictifs*

.6 Code R

```

1 ##### PARTIE DES TAUX TESTS #####
2
3 ##### enregistrement des taux #####
4
5 ## ## ## ## ## taux d'entree en invalidite ## ## ## ## ##
6 sickentry<-function(y){
7   taux=function(x){
8     ifelse(60<=x,0.22,ifelse(55<=x,0.24,ifelse(50<=x,0.29,ifelse
9       (45<=x,0.36,
10      ifelse(40<=x,0.33,ifelse(35<=x,0.3,ifelse(30<=x,0.3,
11      ifelse(25<=x,0.24,ifelse(20<=x,0.10,0))))))))))
12   }
13   taux(y)
14 }
15
16 ##### mortalite des actifs #####
17 mortalentry<-function(age){
18   taux=function(x){
19     ifelse(60<=x,0.0061,ifelse(55<=x,0.0039,ifelse(50<=x,0.0024,
20       ifelse
21       (45<=x,0.0014,
22       ifelse(40<=x,0.00081,ifelse(35<=x,0.00051,ifelse(30<=x,0.00042,
23       ifelse(25<=x,0.00048,ifelse(20<=x,0.00065,0))))))))))
24   }
25   taux(age)
26 }
27

```

```

28 ##### taux de revalidations #####
29 recovery<-function(age,duree){
30   taux=function(x,z){
31     ifelse(60<=x,ifelse(2<=z,0.01,ifelse(1<=z,0.1,ifelse(0.75<=z
32       ,0.26,
33       ifelse(0.5<=z,0.82,ifelse(0.25<=z,4.83,10.72))))),
34     ifelse(50<=x,ifelse(1<=z,1.19,ifelse(0.75<=z,1.27,
35     ifelse(0.5<=z,1.36,ifelse(0.25<=z,4.92,17.15))))),
36     ifelse(40<=x,ifelse(1<=z,1.34,ifelse(0.75<=z,1.74,
37     ifelse(0.5<=z,2,ifelse(0.25<=z,6.1,24.52))))),
38     ifelse(30<=x,ifelse(1<=z,1.35,ifelse(0.75<=z,1.61,
39     ifelse(0.5<=z,2.28,ifelse(0.25<=z,6.47,24.87))))),
40     ifelse(20<=x,ifelse(1<=z,1.42,
41     ifelse(0.75<=z,1.77,ifelse(0.5<=z,2.80,ifelse(0.25<=z
42       ,6.38,25.1))))),0
43
44   )))))
45 }
46 taux(age,duree)
47 }
48
49 # ## ## ### ## ## ## ## taux de deces des invalides ## #####
50 mortalsickness<-function(age,duree){
51   taux=function(x,z){
52     ifelse(60<=x,ifelse(1<=z,0.07,ifelse(0.75<=z,0.083,ifelse(0.5<=
53       z,0.094,
54       ifelse(0.25<=z,0.098,0.098))))),
55     ifelse(50<=x,ifelse(1<=z,0.059,ifelse(0.75<=z,0.73,
56     ifelse(0.5<=z,0.083,ifelse(0.25<=z,0.088,0.086))))),
57     ifelse(40<=x,ifelse(1<=z,0.057,ifelse(0.75<=z,0.07,
58     ifelse(0.5<=z,0.082,ifelse(0.25<=z,0.086,0.086))))),
59     ifelse(30<=x,ifelse(1<=z,0.057,ifelse(0.75<=z,0.07,
60     ifelse(0.5<=z,0.082,ifelse(0.25<=z,0.086,0.086))))),
61     ifelse(20<=x,ifelse(1<=z,0.057,
62     ifelse(0.75<=z,0.07,ifelse(0.5<=z,0.082,
63     ifelse(0.25<=z,0.086,0.086))))),0
64
65   )))))
66 }
67 taux(age,duree)
68 }
69

```

```

70 ##### ALGORITHMES DE WATERS
   #####
71
72 ##### WATERS AVEC m0
   #####
73
74 waters_m0<-function(age,delta0,time.cont=10,tx=0.03,pres=12000){
75
76   agemax=age+time.cont
77   v<-1/(1+tx)
78   N<- floor((agemax-age)/delta0)
79   ##### initialisation des probas
   #####
80   pS<- matrix(0,N+1,N+1)
81   pH<- rep(0,N+1)
82   pD<- rep(0,N+1)
83   pH[1]=1
84   ##### initialisation des taux
   #####
85   re<- matrix(0,N+1,N)
86   ne<- matrix(0,N+1,N)
87   se<- rep(0,N+1)
88   me<- rep(0,N+1)
89   ##### calculs des taux utiles
   #####
90   dt<-(seq(1,N)-0.5)*delta0
91   dg<-seq(age,agemax,delta0)
92   re <- sapply(dt, function(x) {mapply(recovery, dg, x)})
93   ne <- sapply(dt, function(x) {mapply(mortalsickness, dg, x)})
94   me <- sapply(dg, mortalentry)
95   se <- sapply(dg, sickentry)
96   ##### calcul des probas
   #####
97   for(j in 1:(length(dg)-1)){
98     ##### resolution systeme d'equation
       #####
99     b<-c()
100     A<-matrix(nrow=2,ncol=2)
101     sm<-0
102     m0=2/delta0+1
103     #####
104     for (k in seq(1,m0-1)){
105       pS[j+1,k+1]<-(pS[j,k]*abs(1- (delta0/2)*(re[j,k] + ne[j,k])))
106       /
       (1 + (delta0/2)*(re[j+1,k+1] + ne[j+1,k+1]))

```

```

107 }
108 pS[j+1,m0+1]<-((pS[j,m0] + pS[j,m0+1])*(1 - (delta0/2)
109 *(re[j,m0] + ne[j,m0])))/(1 + (delta0/2) *
110 (re[j+1,m0] + ne[j+1,m0]))
111 for (k in seq(2,m0)){
112   sm <- sm + pS[j,k] * re[j,k] + pS[j+1,k]* re[j+1,k]
113 }
114 A[1,1]<- -(delta0/2)*(se[j+1])
115 A[2,1]<- 1 + (delta0/2)*(se[j+1] + me[j+1])
116 A[1,2]<- 1 + (delta0/2)*(re[j+1,1]+ne[j+1,1])
117 A[2,2]<- -(delta0/2)*re[j+1,1]
118 b[1]<- pH[j]*(delta0/2)*se[j]
119 b[2]<- pH[j]*(1 - (delta0/2)*(se[j]+me[j])) + (delta0/2)*sm
120 + (delta0/2)*pS[j,1]*re[j,1] +
121 (delta0/2)*(pS[j,m0+1]*re[j,m0]+pS[j+1,m0+1]*re[j+1,m0])
122 {if(det(A)==0){
123   pH[j+1]=pH[j+1]*0.9
124   pS[j+1,1]= (b[1]-A[1,1]*pH[j+1])/A[2,1]
125 }
126 else{
127   pH[j+1]<-solve(A,b)[1]
128   pS[j+1,1]<- solve(A,b)[2]
129 }
130 }
131 }
132 pD[j+1]<- 1-pH[j+1]-sum(pS[j+1,])
133 }
134 pinv<-apply(pS,1,sum)
135 res<-list(pH,pinv,pD)
136 names(res)<-c("prob_actif","prob_invalide","prob_decès_actif")
137 pai<-res$prob_invalide
138 paa<-res$prob_actif
139
140 vectaux<-sapply( seq(0, (agemax-age),delta0), function(x){v~{x}}
141 )
142
143 v1=(delta0/2)*( pai[1:(length(pai)-1)] * vectaux[1:(length(
144 vectaux)-1)] +
145 pai[2:length(pai)] * vectaux[2:length(vectaux)] )
146 v2=(delta0/2)*( paa[1:(length(paa)-1)] * vectaux[1:(length(
147 vectaux)-1)] +
148 paa[2:length(paa)] * vectaux[2:length(vectaux)] )
149 ai0=sum(v1)
150 aa0=sum(v2)
151 ai0<-as.numeric(ai0)

```

```

149 aa0<-as.numeric(aa0)
150 prim<-pres*(ai0/aa0)
151
152 #return(prim)
153 list(prime=prim, paa = paa, pai = pai)
154 }#fin waters
155
156 ##### WATERS AVEC Integrale
157 #####
158 waters_integ<-function(age,delta0,time.cont=10, tx=0.03,pres=12000)
159 {
160 #####
161 #####
162 agemax=age+time.cont
163 v<-1/(1+tx)
164 N<- floor((agemax-age)/delta0)
165
166 ##### initialisation des probas
167 #####
168 pS<- matrix(0,N+1,N+1)
169 pH<- rep(0,N+1)
170 pD<- rep(0,N+1)
171 pH[1]=1
172 ##### initialisation des taux
173 #####
174 re<- matrix(0,N+1,N)
175 ne<- matrix(0,N+1,N)
176 se<- rep(0,N+1)
177 me<- rep(0,N+1)
178 ##### calculs des taux utiles
179 #####
180 dt<-(seq(1,N)-0.5)*delta0
181 dg<-seq(age,agemax,delta0)
182 re <- sapply(dt, function(x) {mapply(recovery, dg, x)})
183 ne <- sapply(dt, function(x) {mapply(mortalsickness, dg, x)})
184 me <- sapply(dg, mortalentry)
185 se <- sapply(dg, sickentry)
186 ##### calcul des probas
187 #####
188 for(j in 1:(length(dg)-1)){
189 ##### resolution systeme d'equation
190 #####
191 b<-c()

```

```

187 A<-matrix(nrow=2,ncol=2)
188 sm<-0
189
190 y=age+(j-1)*delta0
191 #####
192 for (m in 1:N) {
193   t=(m-1/2)*delta0
194   ##### Fonction integrale
195   #####
196   mu_i=function(u){recovery(y+u,t+u)+mortalsickness(y+u,t+u)}
197   int_mui=integrate(mu_i,lower = 0,upper = delta0)$value
198   ##### probabilite d'invalidite avec integrale
199   #####
200   pS[j+1,m+1]=exp(-int_mui)*pS[j,m]
201 }
202
203 for (m in 2:N) {
204   sm <- sm + pS[j,m] *re[j,m] + pS[j+1,m]*re[j+1,m]
205 }
206
207 A[1,1]<- -(delta0/2)*(se[j+1])
208 A[2,1]<- 1 + (delta0/2)*(se[j+1] + me[j+1])
209 A[1,2]<- 1 + (delta0/2)*(re[j+1,1]+ne[j+1,1])
210 A[2,2]<- -(delta0/2)*re[j+1,1]
211
212 b[1]<- pH[j]*(delta0/2)*se[j]
213
214 b[2]<- pH[j]*(1 - (delta0/2)*(se[j]+me[j])) +
215 (delta0/2)*sm + (delta0/2)*pS[j,1]*re[j,1]
216
217 {if(det(A)==0){
218   pH[j+1]=pH[j+1]*0.9
219   pS[j+1,1]= (b[1]-A[1,1]*pH[j+1])/A[2,1]
220 }
221 else{
222   pH[j+1]<-solve(A,b)[1]
223   pS[j+1,1]<- solve(A,b)[2]
224 }
225 }
226
227 pD[j+1]<- 1-pH[j+1]-sum(pS[j+1,])
228
229 pinv<-apply(pS,1,sum)
230 res<-list(pH,pinv,pD)
231 names(res)<-c("prob_actif","prob_invalide","prob_decès_actif")

```

```

230 pai<-res$prob_invalide
231 paa<-res$prob_actif
232 (pS[,1])
233 vectaux<-sapply( seq(0, (agemax-age),delta0), function(x){v^{x}}
    )
234
235 v1=(delta0/2)*( pai[1:(length(pai)-1)] * vectaux[1:(length(
    vectaux)-1)] +
236 pai[2:length(pai)] * vectaux[2:length(vectaux)] )
237 v2=(delta0/2)*( paa[1:(length(paa)-1)] * vectaux[1:(length(
    vectaux)-1)] +
238 paa[2:length(paa)] * vectaux[2:length(vectaux)] )
239 ai0=sum(v1)
240 aa0=sum(v2)
241 ai0<-as.numeric(ai0)
242 aa0<-as.numeric(aa0)
243 prim<-pres*(ai0/aa0)
244
245 return(prim)
246 list(prime=prim, paa = paa, pai = pai)
247 }#fin waters
248
249
250
251 ##### ALGORITHMES DE MONTE-CARLO
    #####
252
253 ##### fonction de choix aleatoire portant sur l'etat "e0"
    #####
254
255 auton0=function(x,time.cont=10,aa,age_ini,v){ #cours sortie
    multiple p21/60
256
257
258 agemax=age_ini+time.cont
259
260 x0=age_ini
261
262
263 #####sortie par invalidite
264
265 u1<-runif(1)#simulation de sa realisation
266
267 # Definition de la fonction a integrer
268 f1=function(t){sickentry(x+t)+mortalentry(x+t)}

```

```

269
270 # Definition de la borne superieure de l'integrale comme variable
    d'entree
271 g1=function(a,b) {exp(-integrate(f1, lower = a, upper = b)$value)
    }
272
273 step0=(agemax-x0)/100
274
275 prod1=g1(0,step0)
276 {if(u1>=prod1){T1=
277     min(agemax-x, uniroot(function(t){g1(0,t)-u1},c(0,step0+10))$
        root)}
278     else{ j=step0
279
280         while(u1<prod1){
281             prod0=prod1
282             prod1=prod1*g1(j,j+step0)
283             j=j+step0
284         }
285         T1=min(agemax-x,uniroot(function(t){prod0*g1(j,t)-u1},c(j,j
            +10))$root)
286
287
288     }}
289
290 u2<-runif(1)#simulation de sortie
291
292 sortie=ifelse(T1==agemax-x,"e1",ifelse(u2<=mortalentry(x+T1),"e2"
    ,"e1"))
293
294 #valeur actuelle et age a la sortie de l'autonomie
295
296 vt=function(s){v^s}
297 aa=aa+(integrate(vt,0,T1)$value)*(vt(x-x0)) #ACTUALISATION SIMPLE
298 x=x+T1
299 list(x=x,choix=sortie,temps=T1,aa=aa)
300 }
301 #xx=auton0(30,1,30,40,1/1.03)
302
303 ##### fonction choix aleatoire portant sur l'etat "e1"
    #####
304 inval0=function(x,time.cont=10,ai,age_ini,v){
305
306     agemax=age_ini+time.cont
307     x0=age_ini

```

```

308
309
310 #####sortie par invalidite
311
312 u1<-runif(1)#simulation de sa realisation
313
314 # Definition de la fonction a integrer
315 f1=function(t){recovery(x+t,t)+mortalsickness(x+t,t)}
316
317 # Definition de la borne superieure de l'integrale comme variable
    d'entree
318 g1=function(a,b) {exp(-integrate(f1, lower = a, upper = b)$value)
    }
319
320 step0=(agemax-x0)/100
321
322 prod1=g1(0,step0)
323 {if(u1>=prod1){T1=
324     min(agemax-x, uniroot(function(t){g1(0,t)-u1},c(0,step0+10))$
        root)}
325     else{ j=step0
326
327         while(u1<prod1){
328             prod0=prod1
329             prod1=prod1*g1(j,j+step0)
330             j=j+step0
331         }
332         T1=min(agemax-x,uniroot(function(t){prod0*g1(j,t)-u1},c(j,j
            +10))$root)
333
334
335     }}
336
337 u2<-runif(1)#simulation de sortie
338
339 sortie=ifelse(T1==agemax-x,"e0",ifelse(u2<=mortalentry(x+T1),"e2"
    ,"e0"))
340
341 #valeur actuelle et age a la sortie de l'autonomie
342
343 vt=function(s){v^s}
344 ai=ai+(integrate(vt,0,T1)$value)*(vt(x-x0)) #ACTUALISATION SIMPLE
345 x=x+T1
346
347 list(x=x,choix=sortie,temps=T1,ai=ai)

```

```

348 }
349
350
351
352
353 ##### function de Monte Carlo #####
354
355 monte0=function(age,time.cont=10,simul=10000,tx=0.03,prest=12000){
356
357   carl=function(age,time.cont=10,tx=0.03){
358
359     agemax=age+time.cont
360     v=1/(1+tx)
361     aa=1
362     ai=0
363     choix="e0"
364     x=age
365     ini=auton0(x,time.cont,aa,age,v)
366     x=ini$x
367     choix=ini$choix
368     aa=ini$aa
369
370     #repeat{
371       while(choix != "e2" & x<agemax){
372         switch (choix,
373           "e1" = {
374             inv=invale0(x,time.cont ,ai,age,v)
375             choix=inv$choix
376             x =inv$x
377             ai=inv$ai
378           },
379           "e0"={
380             aut=auton0(x,time.cont ,aa ,age,v)
381             choix =aut$choix
382             x =aut$x
383             aa =aut$aa
384           }
385         )
386
387       }
388
389       list(aa=aa,ai=ai)
390     }
391
392     dg<-rep(age,simul)

```

```

393 # Application de la fonction carl a chaque element de age avec
      sapply
394 resultats<-sapply(dg,carl)
395
396 # Extraction des vecteurs aa et ai
397 aa <-as.numeric(resultats[1,])
398 ai <- as.numeric(resultats[2,])
399 intaa<-t.test(aa, conf.level = 0.8)$conf.int
400 intai<-t.test(ai, conf.level = 0.8)$conf.int
401
402 maa=mean(aa)
403 mai=mean(ai)
404
405 prim<-(mai/maa)*prest
406 primlower=(intai[1]/intaa[2])*prest
407 primupper=(intai[2]/intaa[1])*pres
408
409 list(prime=prim,primemin=primlower,primemax=primupper)
410 }
411
412
413
414 ##### METHODE DES ETATS FICTIFS
      #####
415
416
417 ##### Matrice de Transition #####
418 Trans=function(age,delta0){
419
420     Kstar=5/delta0## duree d'invalidite permanente
421
422     P= matrix(0,Kstar+3,Kstar+3)#matrice de transition
423
424     y=age
425     P[1,1]= exp(-delta0*(sickentry(y)+mortalentry(y)))
426     P[1,Kstar+3]= mortalentry(y)*
427     (1- exp(-delta0*(sickentry(y)+mortalentry(y))))/(sickentry(y)+
      mortalentry(y))
428     P[1,2]=1-P[1,1]-P[1,Kstar+3]
429
430     for (k in 2:(Kstar+1)) {
431         P[k,k+1]= exp(-delta0*
432         (recovery(y,(k-2)*delta0)+mortalsickness(y,(k-2)*delta0)))
433         P[k,Kstar+3]=

```

```

434     mortalsickness(y,(k-2)*delta0)*(1-exp(-delta0*(recovery(y,(k
435         -2)*
436         delta0)+mortalsickness(y,(k-2)*delta0))))/
437     (mortalsickness(y,(k-2)*delta0)+recovery(y,(k-2)*delta0))
438     P[k,1]=1-P[k,k+1]-P[k,Kstar+3]
439 }
440
441 P[(Kstar+2),(Kstar+2)]=
442 exp(-delta0*(recovery(y,Kstar*delta0)+mortalsickness(y,Kstar*
443     delta0)))
444 P[(Kstar+2),(Kstar+3)]=1-P[(Kstar+2),(Kstar+2)]
445 P[(Kstar+3),(Kstar+3)]=1
446
447 return(P)
448 }
449 (markov0(30,1/360))
450 ##### algorithme de Markov #####
451 markov0=function(xage,delta0, tx=.03,time.cont=10,pres=12000){
452     agemax=xage+time.cont
453     v=1/(1+tx)
454
455     Kstar=5/delta0 ## duree d'invalidite permanente
456     M=diag(Kstar+3)
457     N<- floor((agemax-xage)/delta0)
458     Paa<- matrix(0,2,(N))
459     Pai<- matrix(0,2,(N))
460     ##### calcul des probas de transition
461     M=M%%Trans(xage,delta0)
462     I=1
463     while(I<=(N)){
464
465         Paa[1,I]<-M[1,1]
466         som<-0
467         for (j in 2:(Kstar+2)) {
468             som=som+M[1,j]
469         }
470         Pai[1,I]=som
471
472         M=M%%Trans(xage+I*delta0,delta0)
473         Paa[2,I]=M[1,1]
474         som2=0
475         for (j in 2:(Kstar+2)) {
476             som2=som2+M[1,j]
477         }
478     }

```

```

477     Pai[2,I]=som2
478
479     I=I+1
480 }
481
482 ##### calcul des primes
483 vectaux<-sapply(seq(0, (agemax-xage),delta0), function(x){exp(x
      *log(v))} )
484 v1=(delta0/2)*( Pai[1,1:(N)] * vectaux[1:(length(vectaux)-1)] +
485 Pai[2,1:(N)] * vectaux[2:length(vectaux)] )
486 v2=(delta0/2)*( Paa[1,1:(N)] * vectaux[1:(length(vectaux)-1)] +
487 Paa[2,1:(N)] * vectaux[2:length(vectaux)] )
488
489 ai0=sum(v1)
490 aa0=sum(v2)
491 priM=pres*ai0/aa0
492
493 # return(priM)
494 list(prime=priM , paaM = Paa[1,],paiM = Pai[1,])
495
496 } #fin Markov
497
498 dg<-seq(30,40,1/52)
499 wterproba=waters_m0(30,1/52)
500 length(wterproba$paa)
501 par(mfrow=c(1,3))
502
503 plot(dg,1-wterproba$pai-wterproba$paa,xlab =
504 "age",col="blue",font.lab=4,col.lab="blue",grid(),
505 ylab = "proba d'etre actif 30ans ",
506 type="l",lwd=4, main="probabilite d'etre actif paa")
507 #lines(xlabs,primM[1,],type = "l",col="red",lwd=4,font.lab=4,col.
      lab="blue")
508 lines(xlabs,mcarlo[1,],type = "l",col="green",lwd=4,font.lab=4,
      col.lab="blue")
509 legend("topright", legend=c("waters", "montecarlo"),
510 col=c("blue","green" ), lty=1:1, cex=0.8,lwd=4)
511 grid()
512
513
514 markovproba=markov0(30,1/360)
515 (wterproba$paa)
516
517 ##### PRINCIPAL
      #####

```

```

518
519
520
521 pkg <- c("mapfit", "expm", "PolynomF","rbenchmark")
522 inst <- pkg %in% installed.packages()
523 if(length(pkg[!inst]) > 0) install.packages(pkg[!inst])
524 lapply(pkg, require, character.only=TRUE)
525 #install.packages("mapfit")
526 #install.packages("rgl") # installer la librairie
527 library(mapfit)
528 library(rgl) # lancer la librairie
529 library(Matrix)
530 library(expm)
531 library(rbenchmark)
532 library(PolynomF)
533 library(readxl)
534 library(base)
535 library(stats)
536 library(dplyr)
537 library(graphics)
538 library(magrittr)
539 library(robustbase)
540 library(questionr)#intervalle de confiance
541 library(openxlsx)#extraction excel
542
543 set.seed(245869)
544 dg=c(25,30,35,40,45,50,55)
545 pas=c(1/12,1/52,1/360)
546
547 system.time(watest<-sapply(pas, function(x) {mapply(waters_m0, dg
548 , x)}))
549 system.time(watestint<-sapply(pas, function(x) {mapply(waters_
550 integ, dg, x)}))
551
552 system.time(monetest<-sapply(dg, monte0))
553
554 system.time(mktest<-sapply(pas, function(x) {mapply(markov0, dg,
555 x)}))
556
557 ##### PARTIE DES TAUX REELS
558 #####

```

```

559 #####taux d'entree en invaliditee
560 #####
561 invalidentry<-function(age,DP=1){
562 ##### fonction principale
563 #####
564 sigmarates<-function(age,DP){
565     t<-(age-40)/25
566     c0<-1
567     c1<- t
568     c2<- 2*t**2-1
569     c3<- 4*t**3-3*t
570     if(DP==1){
571         b0<- -1.416038
572         b1<- 0.238522
573         b2<- -0.588151
574         b3<- 0.333549
575     }
576     else{
577         if(DP==4){
578             b0<- -3.200943
579             b1<- 0.474149
580             b2<-0
581             b3<-0
582         }
583         else{
584             if(DP==13){
585                 b0<- -2.008366
586                 b1<- -0.614523
587                 b2<-0
588                 b3<-0
589             }
590             else{
591                 if(DP==26){
592                     b0<- -2.786287
593                     b1<- -0.216186
594                     b2<-0
595                     b3<-0
596                 }
597                 else{
598                     if(DP==52){
599                         b0<- -3.816687
600                         b1<- 0.507200
601                         b2<-0
602                         b3<-0

```

```

602         }
603         else{
604             stop("Delais de carence different de 1,4,13,26 et
605                 52")
606         }
607     }
608 }
609 }
610     exp(b0*c0 + b1*c1 + b2*c2 +b3*c3)
611 }
612     sigmarates(age,DP)
613 }
614
615 #####taux de deces des invalides
616 #####
617
618 decesinval<-function(age,duree){
619     ##### fonction de calcul
620     #####
621     nurates<-function(age,duree){
622         a<-0.188906
623         b<-1.081708
624         c<-0.132474
625         r<-0.257331
626         s<-0.149466
627         q<-0.744739
628         if(duree<=5){
629             X<-age-50
630             Z<-duree
631         }
632         else{
633             X<-age-55+duree
634             Z<-5
635         }
636         ((a*exp(-b/(Z+c)))/((Z+c)**2) + (r/100)* exp(s*(X+Z)))*q
637     }
638     #####fonction mere
639     #####
640     nurates(age,duree)
641 }
642
643 ##### taux de revalidations
644 #####

```

```

642 revalide<-function(age, duree, DP=1){
643     ##### fonction de calculs
644     #####
645     revalid<-function(age, duree, DP=1){
646         ## ## ## ## declaration des constantes et param'etres
647         generaux #####
648         a1<- -3.080944
649         a2<- -6.419924
650         a3<-20.048953
651         a4<- -0.113352
652         b1<-0.195291
653         b2<-0.108662
654         h0<-0.198289
655         h1<- -0.724805
656         h2<-0.047682
657         r1<-0.622543
658         r2<-1.197880
659         r3<-1.830356
660         vus<-function(x,h){
661             if(h <= 5){
662                 X<-x-50
663                 Z<-h
664             }
665             else{
666                 X<-x-55+h
667                 Z<-5
668             }
669             return(c(X,Z))
670         }
671         Z<-vus(age,duree)[2]
672         X<-vus(age,duree)[1]
673         w<-(365/7)*Z
674         ## ## ## ## declaration des fonctions utiles ## ## #
675         tt<-function(omega){
676             k<-0.016000
677             res<-omega/(1+k*omega)
678             res
679         }
680         sus<-function(DP){
681             if(DP==1){
682                 res<-3.036467
683             }
684             else{
685                 if(DP==4){
686                     res<-3.316474

```

```

685     }
686     else{
687         if(DP==13){
688             res<-3.025743
689         }
690         else{
691             if(DP==26){
692                 res<-2.856549
693             }
694             else{
695                 if(DP==52){
696                     res<-2.511347
697                 }
698                 else{
699                     stop("delai de carence incorrect")
700                 }
701             }
702         }
703     }
704 }
705 res
706 }
707 fus<-function(x,z){
708     res<-a1*(x/100) + a2*(x/100)**2 + a3*(x/100)**3 + a4*(x/
709         100)*tt(z)
710     res
711 }
712 gus<-function(omega,z){
713     if(omega<=26){
714         res<- -b1*tt(z)
715     }
716     else{
717         res<- -b1*tt(26)-b2*(tt(z)-tt(26))
718     }
719     res
720 }
721 hus<-function(omega,x,z){
722     res<-0
723     if(omega < 4){
724         res<-(tt(4)-tt(z))*(h0 + h1*(x/100) + h2*tt(z))
725     }
726     res
727 }
728 qus<-function(omega,DP){
729     res<-0

```

```

729     if(DP==4 && omega >= 8 && omega < 16){
730         res<--r1*(16-omega)/8
731     }
732     res
733 }
734 rus<-function(omega,DP){
735     res<-0
736     if(DP==4 && omega >= 4 && omega < 8){
737         res<--r2*(8-omega)/(4-r1)
738     }
739     else{
740         if(DP==13 && omega >= 13 && omega < 17){
741             res<--r3*(17-omega)/4
742         }
743     }
744     res
745 }
746 ## ## ## calcul, corps de l'algorithme ## ## ## ## ## ##
747 ss<-sus(DP)
748 ff<-fus(X,w)
749 gg<-gus(w,w)
750 hh<-hus(w,X,w)
751 qq<-qus(w,DP)
752 rr<-rus(w,DP)
753 exp(ss+ff+gg+hh+qq+rr)
754 }
755 #####fonction mere
756 #####
757 revalid(age,duree)
758 }
759 ##### mortalitee des actifs
760 #####
761 decesact<-function(age){
762     ##### fonction de calcul Danish model
763     #####
764     deces<-function(x){
765         a <- -0.00465192
766         b <- -0.00452546
767         c <- -3.985723
768         d <- 3.185063
769         xprime <- (x - 70)/50
770         a + b*xprime + exp(c + d*xprime)
771     }
772     deces(age)

```

```

771 }
772
773 ##### Representation des taux
774 #####
775
776 ##### initialisation des taux
777 #####
778 age=20
779 dis= 1/52
780 tx = .03
781 agemax=60
782 N<- floor((agemax-age)/dis)
783 re<- matrix(0,N+1,N)
784 ne<- matrix(0,N+1,N)
785 se<- rep(0,N+1)
786 me<- rep(0,N+1)
787 ##### calculs des taux utiles
788 #####
789 age=20
790 dis= 1/52
791 tx = .03
792 agemax=60
793 N<- floor((agemax-age)/dis)
794 dt<-(seq(1,N)-0.5)*dis
795 N1<- floor((3)/dis) #si invalidite permanente apres deux ans
796 dt1<-(seq(1,N1)-0.5)*dis
797 dg<-seq(age,agemax,dis)
798 re1 <- sapply(dt1, function(x) {mapply(revalide, 20, x)})
799 re2 <- sapply(dt1, function(x) {mapply(revalide, 40, x)})
800 ne1 <- sapply(dt1, function(x) {mapply(decesinval, 20, x)})
801 ne2 <- sapply(dt1, function(x) {mapply(decesinval, 50, x)})
802 re <- sapply(dt1, function(x) {mapply(revalide, dg, x)})
803 ne <- sapply(dt1, function(x) {mapply(decesinval, dg, x)})
804 me <- sapply(dg, decsesact)
805 se <- sapply(dg, invalidentry)
806
807 par(mfrow=c(1,2))
808
809 plot(dg,me,xlab = "age",ylab = "taux de mortalite des actifs",
      font.lab=4,
      col.lab="blue",type="l",lwd=4,col="blue", main="taux de mortalite
      en activite")
810 grid()

```

```

810 plot(dg,se,xlab = "age",ylab = "taux d'invalidite",type="l",col="
      red",
811 font.lab=4,col.lab="blue",lwd=4, main="taux d'invalidite des
      actifs")
812 grid()
813
814 par(mfrow=c(1,1))
815 # Definir la palette de couleurs en degrade
816 palette <- colorRampPalette(c("blue", "skyblue","cyan"))
817
818 # Generer les couleurs en degrade
819 colors <- palette(length(dg))
820 # Creation du graphique de surface
821 persp3d(dg, dt1, re, xlab = "Age", ylab = "Duree en invalidite",
822 zlab = "Taux de sortie",col=colors,
823 main = "Taux de reactivation en fonction de l'age et de la duree
      en invalidite",
824 theta = 30, phi = 30)
825
826 grid()
827
828 colors1 <- heat.colors(length(dg))
829 persp3d(dg, dt1, ne, xlab = "Age",
830 ylab = "Duree en invalidite", zlab = "Taux de sortie",col=colors1
      ,
831 main = "Taux de deces des invalides en
832 fonction de l'age et de la duree en invalidite",
833 theta = 30, phi = 30)
834
835
836 plot(dt1,re1,xlab = "age",ylab = "taux de revalidation",
837 col="blue",type="l",lwd=2, main="taux de revalidation des actifs"
      )
838 lines(dt1,re2,type = "l",col="green",lwd=2)
839 legend("topright", legend=c("assure de 20ans", "assure de 40ans")
      ,
840 col=c("blue","green" ), lty=1:1, cex=0.8)
841 plot(dt1,ne1,xlab = "age",ylab = "taux de deces des invalides",
842 col="pink",type="l",lwd=2, main="taux de deces des invalides")
843 lines(dt1,ne2,type = "l",col="blue",lwd=2)
844 legend("topright", legend=c("assure de 20ans", "assure de 50ans")
      ,
845 col=c("pink","blue" ), lty=1:1, cex=0.8)
846
847

```

```

##### Probabilitees de transition
#####

##### Matrice de Transition avec constance par morceaux
#####

Trans=function(age,delta0,Kstar){

  #Kstar=3/delta0## duree d'invalidite permanente

  P= matrix(0,(Kstar+3),(Kstar+3))#matrice de transition

  y=age
  P[1,1]= exp(-delta0*(invalidentry(y)+decesact(y)))
  P[1,(Kstar+3)]= decesact(y)*(1- exp(-delta0*(invalidentry(y)+
    decesact(y))))
  /(invalidentry(y)+decesact(y))
  P[1,2]=1-P[1,1]-P[1,(Kstar+3)]

  for (k in 2:(Kstar+1)) {
    P[k,(k+1)]= exp(-delta0*
      (revalide(y,(k-2)*delta0)+decesinval(y,(k-2)*delta0)))
    P[k,(Kstar+3)]= decesinval(y,(k-2)*delta0)*
      (1-exp(-delta0*(revalide(y,(k-2)*delta0)
        +decesinval(y,(k-2)*delta0))))/
      (decesinval(y,(k-2)*delta0)+revalide(y,(k-2)*delta0))
    P[k,1]=revalide(y,(k-2)*delta0)*exp(-delta0*(invalidentry(y)+
      decesact(y)))*
      (1-exp(-delta0*(revalide(y,(k-2)*
        delta0)+decesinval(y,(k-2)*delta0)-(invalidentry(y)+decesact(
          y))))))
    /(decesinval(y,(k-2)*delta0)+
      revalide(y,(k-2)*delta0)-(invalidentry(y)+decesact(y)))
  }

  P[(Kstar+2),(Kstar+2)]=
  exp(-delta0*(revalide(y,Kstar*delta0)+decesinval(y,Kstar*delta0
    )))
  P[(Kstar+2),(Kstar+3)]=1-P[(Kstar+2),(Kstar+2)]
  P[(Kstar+3),(Kstar+3)]=1

  return(P)
}

```

```

886 ##### Calcul des probabilites constantes par morceaux
887 #####
887 markov_const=function(xage,delta0,time.cont=40){
888   agemax=xage+time.cont
889   #v=1/(1+tx)
890
891   Kstar=3/delta0 ## duree d'invalidite permanente
892   M=diag(Kstar+3)
893   N<- floor((agemax-xage)/delta0)
894   Paa<- rep(0,(N+1))
895   Pai<- rep(0,(N+1))
896   Pad<- rep(0,(N+1))
897   Pila<- rep(0,(N+1))
898   Pild<- rep(0,(N+1))
899   #Pili1<- rep(0,(N+1))
900   ##### calcul des probas de transition
901   M=Trans(xage,delta0,Kstar)
902   Paa[1]<-M[1,1]
903   Pai[1]<-M[1,2]
904   Pad[1]<-M[1,(Kstar+3)]
905   Pila[1]<-M[3,1]
906   Pild[1]<-M[3,(Kstar+3)]
907
908
909   I<-2
910   xage<-xage+delta0
911   while(I<=(N+1)){
912     M<-M%%Trans(xage,delta0,Kstar)
913     Paa[I]<-M[1,1]
914     Pai[I]<-sum(M[1,2:(Kstar+2)])
915     Pad[I]<-M[1,(Kstar+3)]
916     Pila[I]<-M[3,1]
917     Pild[I]<-M[3,(Kstar+3)]
918     I=I+1
919     xage=xage+delta0
920   }
921
922   # return(priM)
923   list(paaM = Paa, paiM = Pai, padM = Pad, piaM=Pila, pidM=Pild)
924
925 } #fin Markovconst
926
927 #kkk=markov_const(40,1/52)
928
929

```

```

930 ##### Fonctions utiles pour les calculs des probas
      reelles #####
931
932 #####fonction de sejour en activite
933 sejact=function(x,z){
934     maid=function(t){invalidentry(x+t)+decesact(x+t)}
935     SeAc=exp(-integrate(maid,0,z)$value)
936 }
937
938 #####fonction de sejour en invalidite
939 sejinv=function(x,s,z){
940     miad=function(t){revalide(x+t,s)+decesinval(x+t,s)}
941     SeIn=exp(-integrate(miad,0,z)$value)
942 }
943
944 #####fonction de passage actif-deces
945 actdec=function(x,z){
946     mad=function(t){exp(-t*(invalidentry(x)+decesact(x)))*decesact(
947         x+t)}
948     psd=integrate(mad,0,z)$value
949     #(actdec(20,1/52))
950
951 #####fonction de passage actif-invalidite
952 actinv=function(x,z){
953     mai=function(t){exp(-t*(invalidentry(x)+decesact(x)))*decesact(
954         x+t)*
955         exp(-(z-t)*(revalide(x+t,0)+decesinval(x+t,0)))}
956     psai=integrate(mai,0,z)$value
957 }
958
959 #####fonction de passage invalide-actif
960 invact=function(x,s,z){
961     mia=function(t){exp(-t*(revalide(x,s)+decesinval(x,s)))*
962         revalide(x+t,s)*
963         exp(-(z-t)*(invalidentry(x+t)+decesact(x+t)))}
964     psia=integrate(mia,0,z)$value
965 }
966
967 #####fonction de passage invalide-deces
968 invdec=function(x,s,z){
969     mid=function(t){exp(-t*(revalide(x,s)+decesinval(x,s)))*
970         decesinval(x+t,s)}
971     psid=integrate(mid,0,z)$value
972 }

```

```

970 ##### Matrice de Transition reelle
971 #####
972
973
974 realtrans=function(age,delta0,Kstar){
975
976   #Kstar=3/delta0## duree d'invalidite permanente
977
978   P= matrix(0,(Kstar+3),(Kstar+3))#matrice de transition
979
980   y=age
981   P[1,1]= sejact(y,delta0)
982   P[1,(Kstar+3)]= actdec(y,delta0)
983   P[1,2]=actinv(y,delta0)
984
985   for (k in 2:(Kstar+1)) {
986     P[k,(k+1)]= sejin(y,(k-2)*delta0,delta0)
987
988     P[k,(Kstar+3)]=invdec(y,(k-2)*delta0,delta0)
989     P[k,1]=invact(y,(k-2)*delta0,delta0)
990   }
991
992   P[(Kstar+2),(Kstar+2)]= sejin(y,Kstar*delta0,delta0)
993   P[(Kstar+2),(Kstar+3)]=invdec(y,Kstar*delta0,delta0)
994   P[(Kstar+3),(Kstar+3)]=1
995
996   return(P)
997 }
998
999
1000 ##### Calcul des probabilites relles
1001 #####
1002 markov_real=function(xage,delta0,time.cont=40){
1003   agemax=xage+time.cont
1004   #v=1/(1+tx)
1005
1006   Kstar=3/delta0 ## duree d'invalidite permanente
1007   M=diag(Kstar+3)
1008   N<- floor((agemax-xage)/delta0)
1009   Paa<- rep(0,(N+1))
1010   Pai<- rep(0,(N+1))
1011   Pad<- rep(0,(N+1))
1012   Pila<- rep(0,(N+1))
1013   Pild<- rep(0,(N+1))

```

```

1013 #Pi1i1<- rep(0,(N+1))
1014 ##### calcul des probas de transition
1015 M=realtrans(xage,delta0,Kstar)
1016 Paa[1]<-M[1,1]
1017 Pai[1]<-M[1,2]
1018 Pad[1]<-M[1,(Kstar+3)]
1019 Pi1a[1]<-M[3,1]
1020 Pi1d[1]<-M[3,(Kstar+3)]
1021
1022 I<-2
1023 xage<-xage+delta0
1024 while(I<=(N+1)){
1025     M<-M%%Trans(xage,delta0,Kstar)
1026     Paa[I]<-M[1,1]
1027     Pai[I]<-sum(M[1,2:(Kstar+2)])
1028     Pad[I]<-M[1,(Kstar+3)]
1029     Pi1a[I]<-M[3,1]
1030     Pi1d[I]<-M[3,(Kstar+3)]
1031     I=I+1
1032     xage=xage+delta0
1033 }
1034
1035 # return(priM)
1036 list(paaM = Paa, paiM = Pai, padM = Pad, piaM=Pi1a, pidM=Pi1d)
1037
1038 } #fin Markov
1039
1040
1041 probaconst=markov_const(20,1/52)
1042 probareal=markov_real(20,1/52)
1043
1044 dg<-seq(20,60,1/52)
1045
1046 par(mfrow=c(1,3))
1047
1048 plot(dg,probaconst$paaM,xlab = "age",col="blue",font.lab=4,col.
1049     lab="blue",
1049 ylab = "proba d'etre actif ",
1050 type="l",lwd=3, main="probabilite d'etre actif 'Paa'")
1051 lines(dg,probareal$paaM,type = "l",col="red",lwd=3,font.lab=4,col.
1052     .lab="blue")
1053 legend("topright", legend=c("constant par morceaux","continue"),
1054     col=c("blue","red" ), lty=1:1, cex=0.8,lwd=4)
1055
1056 grid()

```

```

1056
1057 plot(dg,probaconst$paiM,xlab = "age",col="red",font.lab=4,col.lab
      ="blue",grid(),
1058 ylab = "proba d'etre invalide ",
1059 type="l",lwd=3, main="probabilite d'etre invalide 'Pai'")
1060 lines(dg,probareal$paiM,type = "l",
1061 col="skyblue",lwd=3,font.lab=4,col.lab="blue")
1062 legend("topleft", legend=c("constant par morceaux","continue"),
1063 col=c("red","skyblue" ), lty=1:1, cex=0.8,lwd=4)
1064
1065 grid()
1066
1067 plot(dg,probaconst$padM,xlab =
1068 "age",col="black",font.lab=4,col.lab="blue",grid(),
1069 ylab = "proba de deces ",type="l",lwd=3,
1070 main="probabilite de deces 'Pad'")
1071 lines(dg,probareal$padM,type = "l",col="red",lwd=3,font.lab=4,col
      .lab="blue")
1072 legend("topleft", legend=c("constant par morceaux","continue"),
1073 col=c("black","red" ), lty=1:1, cex=0.8,lwd=4)
1074
1075 grid()
1076 par(mfrow=c(1,2))
1077
1078 plot(dg[15:length(dg)],probaconst$piaM[15:length(dg)],
1079 xlab = "age",col="skyblue",font.lab=4,col.lab="blue",grid(),
1080 ylab = "proba invalide-actif ",type="l",
1081 lwd=4, main="probabilite de revalidation 'Pia'")
1082
1083 grid()
1084
1085 plot(dg,probamarkov30$pidM,xlab = "age",
1086 col="green",font.lab=4,col.lab="blue",grid(),
1087 ylab = "proba invalide-deces ",
1088 type="l",lwd=4, main="probabilite de deces en invalidite 'Pad'")
1089
1090 grid()
1091
1092
1093
1094 ##### Calcul des primes constances et reeles
      #####
1095 primarkov_const=function(xage,delta0, tx=.03,time.cont=10,pres
      =12000){
1096   agemax=xage+time.cont

```

```

1097     v=1/(1+tx)
1098
1099     ###calcul des probabilites
1100     proba<-markov_const(xage,delta0,time.cont=10)
1101
1102     Paa<- as.numeric(proba$paaM)
1103     Pai<- as.numeric( proba$paiM)
1104     Pad<- as.numeric(proba$padM)
1105
1106     N<- floor((agemax-xage)/delta0)
1107
1108     ##### calcul des primes
1109     vectaux<-sapply(seq(0, (agemax-xage),delta0), function(x){exp(x
1110         *log(v))} )
1111     v1=(delta0/2)*( Pai[1:(N)] * vectaux[1:(length(vectaux)-1)] +
1112     Pai[2:(N+1)] * vectaux[2:length(vectaux)] )
1113     v2=(delta0/2)*( Paa[1:(N)] * vectaux[1:(length(vectaux)-1)] +
1114     Paa[2:(N+1)] * vectaux[2:length(vectaux)] )
1115
1116     ai0=sum(v1)
1117     aa0=sum(v2)
1118     priM=pres*ai0/aa0
1119
1120     list(prime=priM)# , paaM = Paa,paiM = Pai)
1121
1122 } #fin Markov
1123
1124 primarkov_real=function(xage,delta0, tx=.03,time.cont=10,pres
1125     =12000){
1126     agemax=xage+time.cont
1127     v=1/(1+tx)
1128
1129     ###calcul des probabilites
1130     proba<-markov_real(xage,delta0,time.cont=10)
1131
1132     Paa<- as.numeric(proba$paaM)
1133     Pai<- as.numeric( proba$paiM)
1134     Pad<- as.numeric(proba$padM)
1135
1136     N<- floor((agemax-xage)/delta0)
1137
1138     ##### calcul des primes
1139     vectaux<-sapply(seq(0, (agemax-xage),delta0), function(x){exp(x
1140         *log(v))} )

```

```

1139     v1=(delta0/2)*( Pai[1:(N)] * vectaux[1:(length(vectaux)-1)] +
1140     Pai[2:(N+1)] * vectaux[2:length(vectaux)] )
1141     v2=(delta0/2)*( Paa[1:(N)] * vectaux[1:(length(vectaux)-1)] +
1142     Paa[2:(N+1)] * vectaux[2:length(vectaux)] )
1143
1144     ai0=sum(v1)
1145     aa0=sum(v2)
1146     priM=pres*ai0/aa0
1147
1148
1149     list(prime=priM)# , paaM = Paa,paiM = Pai)
1150
1151 } #fin Markov
1152
1153
1154
1155
1156 dg=c(25,30,35,40)
1157 mkovconst<-function(x){primarkov_const(x,1/52)}
1158 system.time(mkconst<-sapply(dg,mkovconst))
1159
1160 mkovreal<-function(x){primarkov_real(x,1/52)}
1161 system.time(mkreal<-sapply(dg,mkovreal))
1162
1163
1164
1165
1166 ##### APPLICATIONS AUX PRODUITS D'ASSURANCE
1167 #####
1168
1169 ##### Matrice de Transition finale
1170 #####
1171 Trans=function(age,delta0,Kstar){
1172
1173     #Kstar=3/delta0## duree d'invalidite permanente
1174
1175     P= matrix(0,(Kstar+3),(Kstar+3))#matrice de transition
1176
1177     y=age
1178     P[1,1]= exp(-delta0*(invalidentry(y)+decesact(y)))
1179     P[1,(Kstar+3)]= decesact(y)*(1- exp(-delta0*(invalidentry(y)+
1180     decesact(y))))/
1181     (invalidentry(y)+decesact(y))
1182     P[1,2]=1-P[1,1]-P[1,(Kstar+3)]

```

```

1181   for (k in 2:(Kstar+1)) {
1182     P[k,(k+1)]= exp(-delta0*
1183       (revalide(y,(k-2)*delta0)+decesinval(y,(k-2)*delta0)))
1184     P[k,(Kstar+3)]= decesinval(y,(k-2)*delta0)*
1185       (1-exp(-delta0*(revalide(y,(k-2)*delta0)
1186         +decesinval(y,(k-2)*delta0))))/(decesinval(y,(k-2)*
1187         delta0)+revalide(y,(k-2)*delta0))
1188     P[k,1]=revalide(y,(k-2)*delta0)*exp(-delta0*(invalidentry(y)+
1189       decesact(y)))*
1190       (1-exp(-delta0*(revalide(y,(k-2)*delta0)+decesinval(y,(k-2)*
1191         delta0)-
1192         (invalidentry(y)+decesact(y)))))/
1193       (decesinval(y,(k-2)*delta0)+revalide(y,(k-2)*delta0)
1194       -(invalidentry(y)+decesact(y)))
1195   }
1196
1197   P[(Kstar+2),(Kstar+2)]=
1198   exp(-delta0*(revalide(y,Kstar*delta0)+decesinval(y,Kstar*delta0
1199     )))
1200   P[(Kstar+2),(Kstar+3)]=1-P[(Kstar+2),(Kstar+2)]
1201   P[(Kstar+3),(Kstar+3)]=1
1202
1203   return(P)
1204 }
1205
1206 ##### Calcul des probabilites jusqu'a la retraite
1207 #####
1208 probatransition=function(xage,delta0,agemax=60){
1209
1210   Kstar=3/delta0 ## duree d'invalidite permanente
1211   M=diag(Kstar+3)
1212   N<- floor((agemax-xage)/delta0)
1213   Paa<- rep(0,(N+1))
1214   Pai<- matrix(0,(N+1),(Kstar+1))
1215   Pad<- rep(0,(N+1))
1216   Pila<- rep(0,(N+1))
1217   Pild<- rep(0,(N+1))
1218   #Pili1<- rep(0,(N+1))
1219   ##### calcul des probas de transition
1220   M=Trans(xage,delta0,Kstar)
1221   Paa[1]<-M[1,1]
1222   Pai[1,1]<-M[1,2]
1223   Pad[1]<-M[1,(Kstar+3)]
1224   Pila[1]<-M[3,1]

```

```

1222     Pild[1] <- M[3, (Kstar+3)]
1223
1224
1225     I <- 2
1226     xage <- xage + delta0
1227     while (I <= (N+1)) {
1228         M <- M %% Trans(xage, delta0, Kstar)
1229         Paa[I] <- M[1, 1]
1230         Pad[I] <- M[1, (Kstar+3)]
1231         Pila[I] <- M[3, 1]
1232         Pild[I] <- M[3, (Kstar+3)]
1233         for (j in 1:min(I, (Kstar+1))) {
1234             Pai[I, j] <- M[1, (j+1)]
1235         }
1236         #Pai[I] <- sum(M[1, 2:(Kstar+2)])
1237         I = I + 1
1238         xage = xage + delta0
1239     }
1240
1241     # return(priM)
1242     list(paaM = Paa, paiM = Pai, padM = Pad, piaM = Pila, pidM = Pild)
1243
1244 } #fin Markovconst
1245
1246
1247 ##### Prestation variee #####
1248 prestvariee = function(age, duree) {
1249     prestv = function(x, z) {
1250         ifelse(x > 65, 0,
1251             ifelse(x >= 45, ifelse(z <= 3, 15000 * (1 - z/3), 0),
1252                 ifelse(x > 35, ifelse(z <= 3, 12000 * (1 - z/3), 0),
1253                     ifelse(x >= 20, ifelse(z <= 3, 9000 * (1 - z/3), 0),
1254                         0))))
1255     }
1256     prestv(age, duree)
1257 }
1258
1259
1260 ##### Calcul des primes #####
1261
1262
1263 primarkov_final = function(xage, delta0, tx = .03, agemax = 60) {
1264     v = 1 / (1 + tx)
1265

```

```

1266 #####calcul des probabilites
1267 #####
1268 proba<-probatransition(xage,delta0)
1269
1269 Paa<- as.numeric(proba$paaM)
1270 Pai<- as.numeric( proba$paiM)
1271 Pad<- as.numeric(proba$padM)
1272
1273 N<- floor((agemax-xage)/delta0)
1274 m0<-3 ## duree d'invaliditee permanente
1275 Kstar=m0/delta0
1276 tempsfr=1/12
1277 franchise <-tempsfr/delta0 ## duree de la franchise temporelle
1278
1279 ##### vecteur d'actualisation
1280 #####
1280 vectaux<-sapply(seq(0, (agemax-xage),delta0), function(x){exp(x
1281 *log(v))} )
1282 decestaux<-sapply(seq(0, (agemax-xage),delta0), decesact )
1283
1283 vaa=(delta0/2)*( Paa[1:(N)] * vectaux[1:(length(vectaux)-1)] +
1284 Paa[2:(N+1)] * vectaux[2:length(vectaux)] )
1285 vad=(delta0/2)*( Pad[1:(N)] * vectaux[1:(length(vectaux)-1)]*
1286 decestaux[1:(N)]
1287 +
1287 Pad[2:(N+1)] * vectaux[2:length(vectaux)]*
1288 decestaux[2:(N+1)] )
1289 aa=sum(vaa) ##valeur actuelle des actifs pour 1 euro
1290 ad=sum(vad) ##valeur actuelle des deces pour 1 euro
1291
1292 prim_deces=20000*ad/aa
1293 ##### Prestation constante
1294 #####
1295 ##### sans franchise #####
1296 prest1=12000*matrix(1,(N+1),(Kstar+1))
1297 prest1_proba=prest1*Pai
1298 vecprest1_proba=rowSums(prest1_proba)
1299
1300 prestvai1=(delta0/2)*( vecprest1_proba[1:(N)] *
1301 vectaux[1:(length(vectaux)-1)] +
1302 vecprest1_proba[2:(N+1)] * vectaux[2:length(vectaux)] )
1303 product1_prestai=sum(prestvai1)
1304
1305 prim1=product1_prestai/aa

```

```

1306
1307
1308 ##### avec franchise #####
1309 prest2_proba=prest1_proba
1310 prest2_proba[,1:franchise]=0 ##application de la franchise
1311 vecprest2_proba=rowSums(prest2_proba)
1312
1313 prestvai2=(delta0/2)*( vecprest2_proba[1:(N)] *
1314 vectaux[1:(length(vectaux)-1)] +
1315 vecprest2_proba[2:(N+1)] *
1316 vectaux[2:length(vectaux)] )
1317 product2_prestai=sum(prestvai2)
1318
1319 prim2=product2_prestai/aa
1320
1321 ##### Prestation variee
1322 #####
1323 dt<-seq(0,m0,delta0)
1324 dg<-seq(xage,agemax,delta0)
1325
1326 ##### sans franchise #####
1327 prest3=matrix(0,(N+1),(Kstar+1))
1328 prest3=sapply(dt, function(x) {mapply(prestvariee, dg, x)})
1329 prest3_proba=prest3*Pai
1330 vecprest3_proba=rowSums(prest3_proba)
1331
1332 prestvai3=(delta0/2)*( vecprest3_proba[1:(N)] *
1333 vectaux[1:(length(vectaux)-1)] +
1334 vecprest3_proba[2:(N+1)] *
1335 vectaux[2:length(vectaux)] )
1336 product3_prestai=sum(prestvai3)
1337
1338 prim3=product3_prestai/aa
1339
1340 ##### avec franchise #####
1341 prest4_proba=prest3_proba
1342 prest4_proba[,1:franchise]=0 ##application de la franchise
1343 vecprest4_proba=rowSums(prest4_proba)
1344
1345 prestvai4=(delta0/2)*( vecprest4_proba[1:(N)] *
1346 vectaux[1:(length(vectaux)-1)] +
1347 vecprest4_proba[2:(N+1)] *
1348 vectaux[2:length(vectaux)] )
1349 product4_prestai=sum(prestvai4)

```

```

1350
1351     prim4=product4_prestai/aa
1352
1353     prim5=prim2+prim_deces
1354     prim6=prim4+prim_deces
1355
1356     list(prim1=prim1,prim2=prim2,prim3=prim3,prim4=prim4,prim5=
          prim5,prim6=prim6)
1357
1358 } #fin Markov
1359
1360 dgf=seq(20,58,2)
1361 primf=function(x){primarkov_final(x,1/52)}
1362 vec_prime=sapply(dgf,primf)
1363
1364 #dgff=seq(25,55,10)
1365 #tab_prim=sapply(dgff,primf)
1366 #write.table(tab_prim,file = "C:/Users/Yakri/
1367 #OneDrive - UCL/memoire/codeR/thesis/excel/tab_prim.xls")
1368
1369 par(mfrow=c(2,2))
1370
1371 plot(dgf,vec_prime[1,],xlab = "age",col="blue",font.lab=4,col.lab
      ="blue",
1372      ylim = c(min(as.numeric(vec_prime[2,]),na.rm = TRUE),
1373               max(as.numeric(vec_prime[1,]),na.rm = TRUE)),
1374      ylab = "prime ",type="l",lwd=4, main="primes avec prestation
            constante")
1375      lines(dgf,vec_prime[2,],type = "l",col="red",lwd=4,font.lab=4,col
            .lab="blue")
1376      legend("topleft", legend=c("sans franchise","avec franchise"),
1377            col=c("blue","red" ), lty=1:1, cex=0.9,lwd=4)
1378
1379 grid()
1380
1381 plot(dgf,vec_prime[3,],xlab = "age",col="skyblue",font.lab=4,col.
      lab="blue",
1382      ylim = c(min(as.numeric(vec_prime[4,]),na.rm = TRUE),
1383               max(as.numeric(vec_prime[3,]),na.rm = TRUE)),
1384      ylab = "prime ",type="l",lwd=4, main="primes avec prestation vari
            ?e")
1385      lines(dgf,vec_prime[4,],type = "l",col="pink",lwd=4,font.lab=4,
            col.lab="blue")
1386      legend("topleft", legend=c("sans franchise","avec franchise"),
1387            col=c("skyblue","pink" ), lty=1:1, cex=0.9,lwd=4)

```

```

1388
1389 grid()
1390
1391 plot(dgf,vec_prime[2,],xlab = "age",col="blue",font.lab=4,col.lab
    ="blue",
1392 ylab = "prime ",type="l",lwd=4, main="primes invalidite + deces")
1393 lines(dgf,vec_prime[5,],type = "l",col="pink",lwd=4,font.lab=4,
    col.lab="blue")
1394 #lines(dgf,vec_prime[5,],type = "l",col="red",lwd=4,font.lab=4,
    col.lab="blue")
1395 #lines(dgf,vec_prime[6,],type = "l",col="black",lwd=4,font.lab=4,
    col.lab="blue")
1396 legend("topleft", legend=c("constant simple","constant+deces"),
1397 col=c("blue","pink" ), lty=1:1, cex=0.9,lwd=4)
1398
1399 grid()
1400
1401 plot(dgf,vec_prime[6,],xlab = "age",col="red",font.lab=4,col.lab=
    "blue",
1402 ylab = "prime ",type="l",lwd=4, main="primes invalidite + deces")
1403 lines(dgf,vec_prime[4,],type = "l",col="black",lwd=4,font.lab=4,
    col.lab="blue")
1404 legend("topleft", legend=c("variee simple","variee+deces"),
1405 col=c("red","black" ), lty=1:1, cex=0.9,lwd=4)
1406
1407 grid()
1408
1409 ##### Calcul des reserves
    #####
1410
1411 resmarkov_final=function(xage,delta0,tpas, tx=.03,agemax=60){
1412   v=1/(1+tx)
1413   ##### calcul des primes
    #####
1414   prim<-primarkov_final(xage,delta0)
1415   prim1=prim$prim1
1416   prim2=prim$prim2
1417   prim3=prim$prim3
1418   prim4=prim$prim4
1419   prim5=prim$prim5
1420   prim6=prim$prim6
1421   ##### calcul des probabilites
    #####
1422   xage=xage+tpas
1423   proba<-probatransition(xage,delta0)

```

```

1424
1425 Paa<- as.numeric(proba$paaM)
1426 Pai<- as.numeric( proba$paiM)
1427 Pad<- as.numeric(proba$padM)
1428
1429 N<- floor((agemax-xage)/delta0)
1430 m0<-3    ## duree d'invaliditee permanente
1431 Kstar=m0/delta0
1432 tempsfr=1/12
1433 franchise <-tempsfr/delta0 ## duree de la franchise temporelle
1434
1435 ##### vecteur d'actualisation
1436 #####
1437 vectaux<-sapply(seq(0, (agemax-xage),delta0), function(x){exp(x
1438 *log(v))} )
1439 decestaux<-sapply(seq(0, (agemax-xage),delta0), decesact )
1440
1441 vaa=(delta0/2)*( Paa[1:(N)] * vectaux[1:(length(vectaux)-1)] +
1442 Paa[2:(N+1)] * vectaux[2:length(vectaux)] )
1443 vad=(delta0/2)*( Pad[1:(N)] * vectaux[1:(length(vectaux)-1)]*
1444 decestaux[1:(N)]+
1445 Pad[2:(N+1)] *
1446 vectaux[2:length(vectaux)]*decestaux[2:(N+1)] )
1447 aa=sum(vaa) ##valeur actuelle des actifs pour 1 euro
1448 ad=sum(vad) ##valeur actuelle des deces pour 1 euro
1449
1450 res_deces=20000*ad-(prim5-prim2)*aa
1451 ##### Prestation constante
1452 #####
1453 ##### sans franchise #####
1454 prest1=12000*matrix(1,(N+1),(Kstar+1))
1455 prest1_proba=prest1*Pai
1456 vecprest1_proba=rowSums(prest1_proba)
1457
1458 prestvai1=(delta0/2)*( vecprest1_proba[1:(N)] *
1459 vectaux[1:(length(vectaux)-1)] +
1460 vecprest1_proba[2:(N+1)] *
1461 vectaux[2:length(vectaux)] )
1462 product1_prestai=sum(prestvai1)
1463
1464 res1=product1_prestai-prim1*aa
1465
1466 ##### avec franchise #####

```

```

1466 prest2_proba=prest1_proba
1467 prest2_proba[,1:franchise]=0 ##application de la franchise
1468 vecprest2_proba=rowSums(prest2_proba)
1469
1470 prestvai2=(delta0/2)*( vecprest2_proba[1:(N)] *
1471 vectaux[1:(length(vectaux)-1)] +
1472 vecprest2_proba[2:(N+1)] *
1473 vectaux[2:length(vectaux)] )
1474 product2_prestai=sum(prestvai2)
1475
1476 res2=product2_prestai-prim2*aa
1477
1478 ##### Prestation variee
1479 #####
1480 dt<-seq(0,m0,delta0)
1481 dg<-seq(xage,agemax,delta0)
1482
1483 ##### sans franchise #####
1484 prest3=matrix(0,(N+1),(Kstar+1))
1485 prest3=sapply(dt, function(x) {mapply(prestvariee, dg, x)})
1486 prest3_proba=prest3*Pai
1487 vecprest3_proba=rowSums(prest3_proba)
1488
1489 prestvai3=(delta0/2)*( vecprest3_proba[1:(N)] *
1490 vectaux[1:(length(vectaux)-1)] +
1491 vecprest3_proba[2:(N+1)] *
1492 vectaux[2:length(vectaux)] )
1493 product3_prestai=sum(prestvai3)
1494
1495 res3=product3_prestai-prim3*aa
1496
1497 ##### avec franchise #####
1498 prest4_proba=prest3_proba
1499 prest4_proba[,1:franchise]=0 ##application de la franchise
1500 vecprest4_proba=rowSums(prest4_proba)
1501
1502 prestvai4=(delta0/2)*( vecprest4_proba[1:(N)] *
1503 vectaux[1:(length(vectaux)-1)] +
1504 vecprest4_proba[2:(N+1)] *
1505 vectaux[2:length(vectaux)] )
1506 product4_prestai=sum(prestvai4)
1507
1508 res4=product4_prestai-prim4*aa
1509

```

```
1510     res5=res2+res_deces
1511     res6=res4+res_deces
1512
1513     list(res1=res1, res2=res2,res3=res3,res4=res4,res5=res5,res6=
1514           res6)
1515 } #fin Markov
1516
1517 reservm=function(x){resmarkov_final(x,1/52,10)}
1518 dgff=seq(20,40,5)
1519 tab_resv=sapply(dgff,reservm)
1520 write.table(tab_resv,file = "C:/Users/Yakri/
1521 OneDrive - UCL/memoire/codeR/thesis/excel/tab_resvc.xls")
```

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences

Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/sc