Création d'un outil de couverture de taux d'intérêts en ligne

Auteurs :
Anaïs Ng liet hing
Manon Pieren
Patrick Du Chouchet
Rémi Gauville

Responsables :
Correspondant entreprise : M. Fabrice Hamon
Responsable de formation : M. Franck Vermet

Deuxième année d'actuariat 2015-2016

Résumé

Ce projet a pour objectif le développement d'une application financière grâce au logiciel RShiny. Il a été choisi de travailler sur les courbes de taux et sur le pricing des swaps de taux.
Le but de ce bureau d'étude est aussi de créer une première application sous l'interface RShiny à l'EURIA, elle doit initier une base de travaux d'étudiants des années suivantes sous cette interface pour permettre de donner une meilleure visibilité aux travaux des étudiants de l'école.

Dans un premier temps, nous présenterons l'ensemble des notions de mathématiques financières et d'interpolation nous ayant été utiles pour la création de l'application.
Ensuite, nous détaillerons les méthodes choisies pour la création des courbes de taux zéro-coupon, des facteurs d'actualisation et des taux forward.
Par la suite, nous détaillerons les étapes et les techniques utilisées pour l'implémentation d'un pricer de swaps.
Enfin, nous présenterons le logiciel RShiny et les décisions ayant été prises pour le choix du serveur ainsi que la version finale de notre application.

Remerciements

Avant de détailler le déroulement de notre bureau d'études, nous tenons tout d'abord à adresser nos remerciements à ceux qui nous ont épaulé durant le déroulement de ce projet.

Tout d'abord nous tenons à remercier chaleureusement Monsieur Fabrice Hamon, fondateur de BRITSK, notre tuteur pour ce bureau d'études. Du fait de sa disponibilité et de l'accompagnement qu'il nous a apporté pendant cette année.

Nos remerciements s'adressent également à M. Franck VERMET, directeur des études à l'EURIA et responsable des bureaux d'études qui nous a aidé par ses remarques et conseils lors des soutenances intermédiares et lors des séances de BE. Nous remercions de plus M.Pierre Ailliot, maître de conférence à l'Université de Bretagne Occidentale (UBO), Benjamin Le Boucher,consultant chez Sia Partners, et Maxime Malal, étudiant de master 2, pour leurs conseils et le temps qu'ils nous ont consacré.

Pour finir, nous aimerions remercier Mme Patricia Bellanger-Gruet secrétaire de l'EURIA, toujours présente pour aider les étudiants dans leurs démarches tout au long de l'année.

Table des matières

Notations :

Acronymes	Significations
CC	Coupon Couru
DF	Discount Factor
E6M	EURIBOR 6 mois
IA	Institut des Actuaires
N	Nominal
OAT	Obligation Admissible du Trésor
P	Prix
r	Taux
VA	Valeur Actualisée
ZC	Zéro Coupon

TABLE 1 – Notations pour l'ensemble de ce rapport

Note : Les notations non présentées dans ce tableau seront expliquées au moment de leur utilisation.

Note 2 : Tous les résultats seront arrondis au millième.

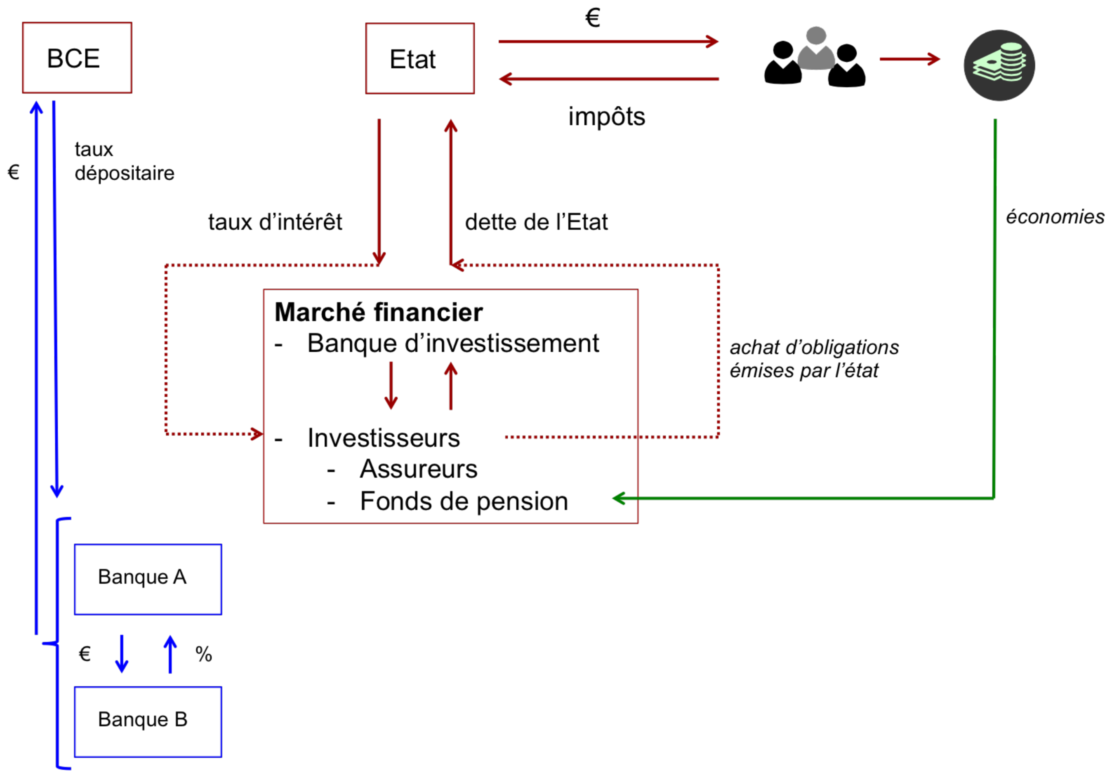


Figure 1 – Contexte économique

Les travaux développés lors de ce bureau d'études s'inscrivent dans le contexte économique actuel décrit par la Figure 1. Il met en valeur la nécessité de construire des courbes de taux pour les différents acteurs économiques.

Première partie

Théorie financière

1 Différentes notions de mathématiques financières

Dans cette section, les notions de mathématiques financières utilisées pour réaliser notre application seront expliquées. Ainsi, lors de la dernière partie de ce rapport, il sera plus simple d'expliquer ce qui est mis en place à travers l'application.

Swap

Un Swap de taux est un contrat d'échange de conditions de taux d'intérêt. Dans sa forme la plus générale, une entité A accepte de payer sur une durée déterminée des flux d'intérêts fixes à l'entité B tandis que B accepte de payer à A des flux d'intérêts variables indexés sur un indice déterminé. Le montant nominal sur lequel porte les intérêts n'est pas échangé.

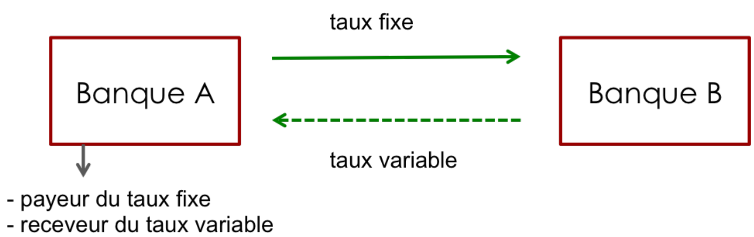


FIGURE 2 – Mécanisme du Swap

Le swap a donc les caractéristiques suivantes :
– Nominal, noté N : capital initial emprunté par l'émetteur de l'obligation divisé par le nombre de titres émis
– Date de valorisation
– Maturité, notée M
– Périodicité de la jambe fixe, notée Pf
– Périodicité de la jambe variable, notée Pv
– Périodicité du taux variable
– Taux fixe du swap, noté t_s qui est coté
– Taux variable, noté t_v
– Conventions de base de calcul

Dans le swap, deux notions sont définies :

La jambe fixe, notée JF - qui est la série des flux à taux fixe, et la jambe variable, notée JV - qui est la série des flux à taux variable :

$$JF = N \cdot \sum_i (r_s \cdot \alpha_i) \cdot DF_i$$

$$JV = N \cdot \sum_i (Fwd_i \cdot \alpha_i) \cdot DF_i$$

avec N : le nominal
r_s : le taux swap coté
α_i : le décompte de jours dans la base choisie
DF_i : le facteur d 'actualisation en $t = t_i$
Fwd_i : le taux forward pour la période $[t_{i-1}; t_i]$, calculé en t_{i-1}

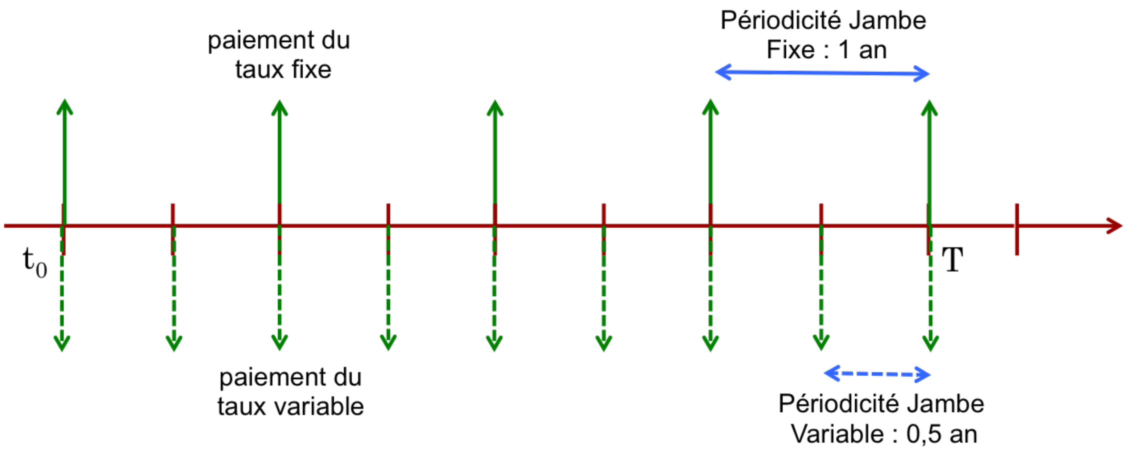


FIGURE 3 – Swap

Obligations

Une obligation est une valeur mobilière représentant une part de dette à long terme d'un État, d'une collectivité locale ou d'une société. Il existe différents types d'obligations, mais le principe général reste le même.

L'achat d'une obligation peut être assimilé à un prêt d'argent à l'émetteur de celle-ci.

Il peut s'agir d'un État, d'une entreprise ou d'une collectivité.

Une obligation est caractérisée par : le montant du prêt ou nominal, le taux d'intérêts, la périodicité du versement des intérêts (ou coupons) et la date d'échéance (ou date de maturité), qui est la date à laquelle le montant nominal doit être remboursé.

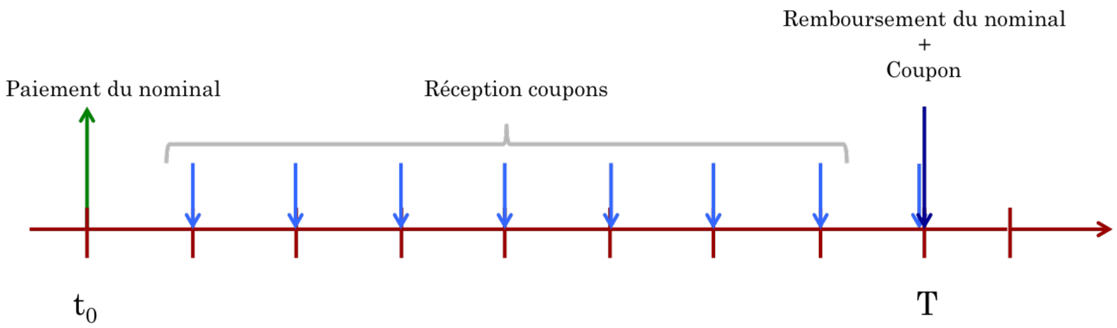


FIGURE 4 – Obligation

De nombreux particuliers choisissent les obligations lorsqu'ils recherchent à la fois une source de revenu régulier et la préservation de leur capital.

Il existe plusieurs agences de notation telles que Moody's, Standard & Poor's (S&P) ou Fitch qui gèrent l'évaluation des obligations. En cas de craintes sur la solvabilité d'un émetteur, la notation accordée par les agences peut s'en ressentir et donc, le taux d'intérêt offert par l'obligation sera généralement bien plus élevé que celui d'une obligation moins risquée.

Il existe un risque de perte relatif si vous revendez une obligation sur le marché avant son échéance. Dans ce cas, on dit que l'on vend l'obligation sur le marché secondaire, et donc, en se soumettant aux conditions de marché. Les variations des taux d'intérêt auront alors un effet sur la valeur de l'actif, le cours de l'obligation peut être plus bas qu'au moment de l'émission. En outre, le montant des intérêts courant jusqu'à l'échéance est aussi perdu.

Coupons

Les coupons correspondent aux intérêts, fixes ou variables, versés périodiquement au détenteur de l'actif par l'émetteur. Ils s'expriment en pourcentage du montant du nominal.

Coupon couru

Le coupon couru est la partie du coupon due au détenteur à l'instant du calcul. Autrement dit, c'est la part du coupon correspondant à la proportion de la période déjà écoulée depuis le dernier paiement de coupon. Ce dernier, noté CC, est donné par la formule suivante :

$$CC = CN \cdot \frac{J_c}{N_a}$$

avec CN : le taux coupon nominal
N_a : le nombre exact de jours dans l'année
J_c : le nombre de jours écoulés depuis le détachement du dernier coupon

Exemple :

Considérons un investisseur achetant une obligation à 3,5% le 10 décembre 2015 et de maturité le 15 novembre 2020. Le prix de marché est de 96,156.
Les coupons sont versés le 21 novembre de chaque année. La période de coupon couru est donc de 26 jours (différence de jours entre le 21 novembre 2015 et le 10 décembre 2015).

$$CC = 0,035 \cdot \frac{26}{365} = 0,249$$

Le coupon couru étant égal à 0,249, l'investisseur doit donc payer :

$$96,156 + 0,249 = 96,405.$$

Discount Factor

Le coefficient d'actualisation (ou Discount Factor), noté DF, est le prix à payer aujourd'hui pour obtenir 1€ à une date future.

$$DF = \frac{1}{(1+r)^T}$$

avec r : le taux d'actualisation
T : la date future

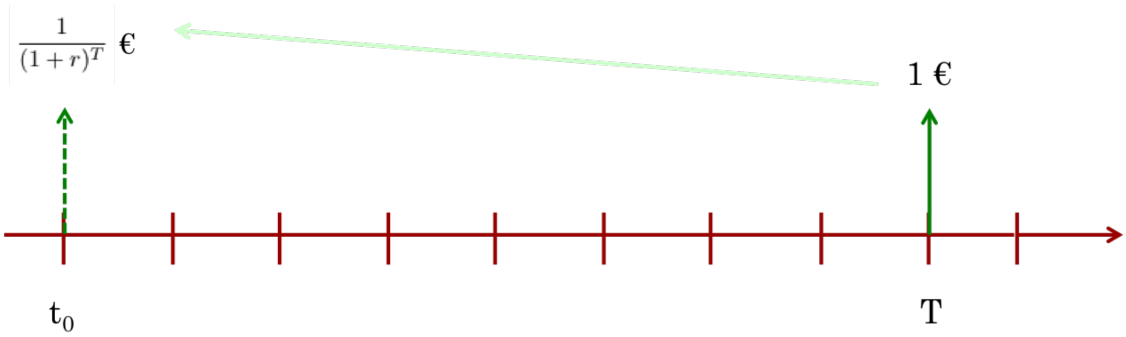


FIGURE 5 – Discount Factor

Taux d'actualisation

Un montant K n'a pas la même valeur aux instants t=0 et à t>0. La formule donnée ci-dessous permet de ramener la valeur du flux à venir à sa valeur actuelle :

$$V_T = V_0 \cdot (1+r)^T$$

avec r : le taux d'actualisation
V_0 : la valeur actuelle du montant
V_T : la valeur du montant à l'instant T
T : le nombre de périodes entre $t = 0$ et $t = T$

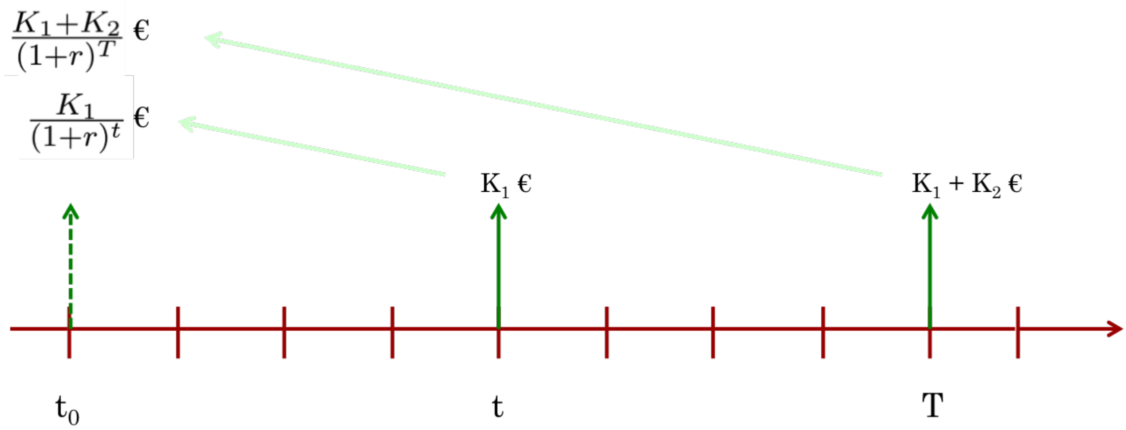


FIGURE 6 – Actualisation

Le prix d'une obligation est la somme des flux actualisés au taux d'actualisation pour une période donnée. Les flux sont composés des coupons périodiques ainsi que du nominal. Ce taux d'actualisation("Yield" ou "Yield to Maturity"), noté r, est défini comme suit :

$$V_T = \frac{c}{1+r} + \frac{c}{(1+r)^2} + \cdots + \frac{c+N}{(1+r)^T}$$

avec c : le coupon
N : le nominal
T : la date de maturité

Il s'agit du taux d'actualisation unique qui va servir pour tous les flux de l'obligation.

Clean price et dirty price

On distingue le cours pied de coupon (Clean Price) du cours plein coupon (Dirty Price). Le Clean Price est défini par le prix d'une obligation diminué du coupon couru (Accrued Interet).
Le Dirty Price correspond à la somme du Clean Price et du coupon couru.
L'obligation est dite au pair lorsque le prix d'émission est égal au nominal.

Exemple :

On se place au 11 mai 2006.

Dates de maturité	**Coupons (en %)**	**DF (en %)**	**Net Present Value (en %)**
01/09/2010	105	85,305	89,570
01/09/2009	5	88,816	4,441
01/09/2008	5	92,337	4,617
01/09/2007	5	95,805	4,790
01/09/2006	5	99,1517	4,958
Dirty Price			98,628
Coupon Couru			3,452
Clean Price			104,924

TABLE 2 – Obligation de maturité 4 ans

Taux zéro coupon

Une obligation dite "zéro coupon" est une obligation dont la totalité des intérêts est versée à la date de maturité.
Une obligation zéro-coupon d'échéance T est un contrat qui garantit à son

propriétaire le paiement du montant nominal à la date T, sans recevoir de coupons intermédiaires. Par exemple, le prix $P(t,T)$ à la date T d'une obligation zéro-coupon de nominal 1€ est $P(T,T)$=1€.

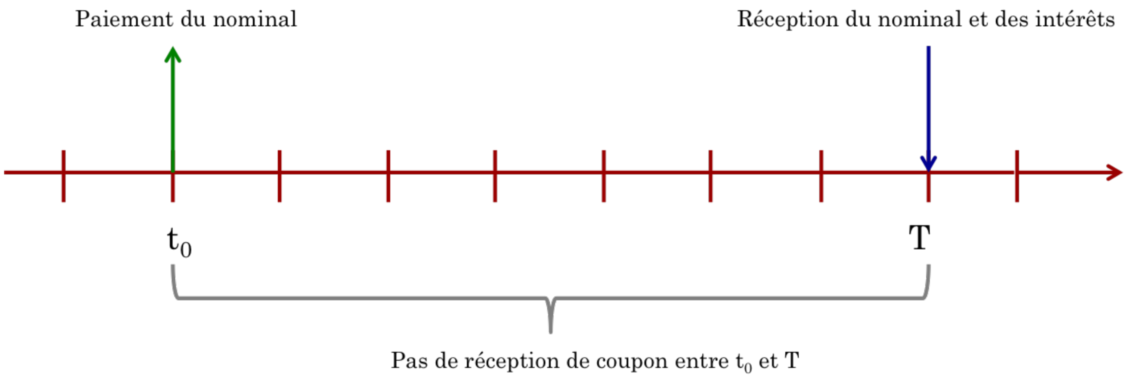


FIGURE 7 – Taux zéro coupon

Exemple :

FIGURE 8 – Prix du ZC

Le prix de l'obligation zéro-coupon, noté P, est donc :
P $= 99,8\% \cdot 1000 = 998$€

Taux spot

Le taux spot est le taux $R(t,T)$ tel que :

$$P(t,T) = e^{-R(t,T)(t,T)}$$

On a donc :

$$R(t,T) = -\frac{\ln P(t,T)}{t-T}$$

$R(t,T)$ est le taux d'intérêt composé continu ou taux spot.

2 Interpolation

Il existe différentes méthodes d'interpolation des courbes des taux sans risque. Dans le cadre de notre bureau d'étude, la méthode de Nelson Siegel Svensson a été utilisée.

Dans cette partie, on expliquera dans un premier temps le modèle Nelson Siegel. Puis, on décrira le modèle Nelson Siegel augmenté (ou Nelson Siegel Svensson). Enfin, on discutera de l'utilisation de ces deux méthodes dans le cadre de notre bureau d'études.

Nelson Siegel :

La méthode de Nelson Siegel, notée NS, suppose que les taux forward instantanés f_τ satisfassent :

$$f_\tau = \beta_0 + \beta_1 \cdot e^{-\tau/\lambda} + \beta_2 \cdot \frac{\tau}{\lambda} \cdot e^{-\tau/\lambda}$$

A noter que les paramètres λ, β_0, β_1 et β_2 sont des paramètres du modèle à estimer, et qu'il existe une relation entre les taux forward instantanés f_τ et le taux $y(\tau)$ telle que :

$$y(\tau) = \frac{\tau}{\lambda} \int_0^\tau f_s ds$$

Ainsi, on obtient :

$$y(\tau) = \beta_0 + \beta_1 \cdot \frac{1 - e^{-\tau/\lambda}}{\tau/\lambda} + \beta_2 \cdot \left(\frac{1 - e^{-\tau/\lambda}}{\tau/\lambda} - e^{-\tau/\lambda} \right)$$

avec λ : le paramètre d'échelle
β_0 : le facteur de niveau
β_1 : le facteur de rotation
β_2 : le facteur de pente

Afin de comprendre les effets des paramètres dans ce modèle, des simulations ont été effectuées.
Pour analyser l'effet de la rotation (ou courbure) β_1, les facteurs β_0, β_2 et λ_1 sont fixés comme suit :

β_0	β_2	λ_1
0,070	0,01	3,33

TABLE 3 – Paramètres fixés pour les simulations de β_1

La courbe a des allures très différentes lorsque β_1 varie :

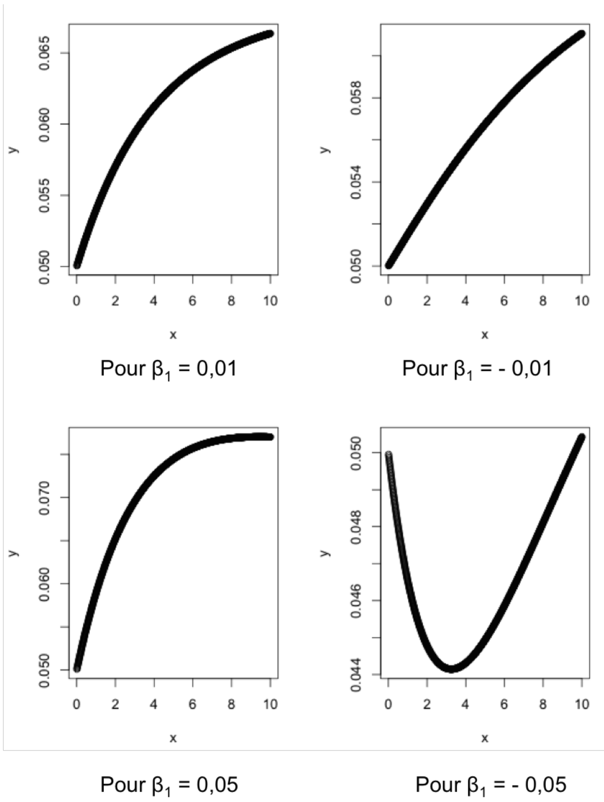


FIGURE 9 – Effet de la rotation dans le modèle NS

Par ailleurs, lorsqu'on fait varier le facteur de pente β_2 pour β_0, β_1 et λ_1 fixés comme suit :

β_0	β_1	λ_1
0,070	$-0,02$	3,33

TABLE 4 – Paramètres fixés pour les simulations de β_2

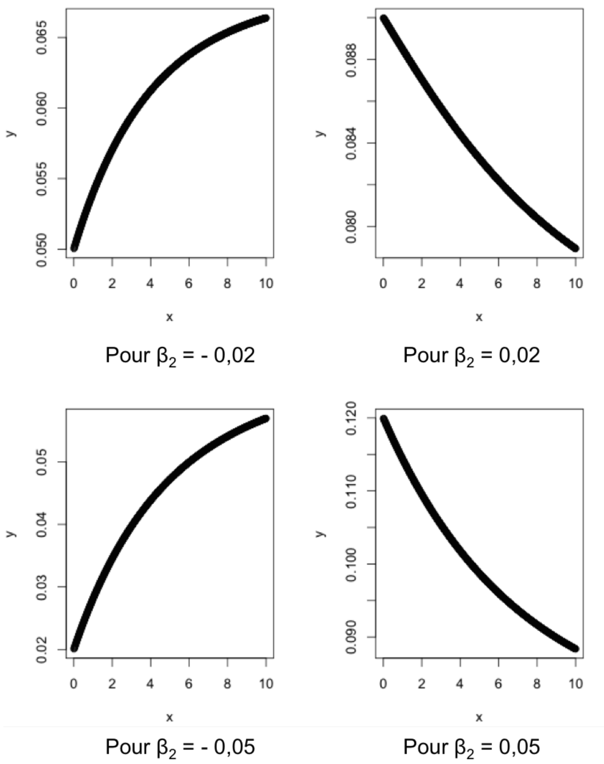


FIGURE 10 – Effet de la pente dans le modèle NS

On obtient une pente positive quand le paramètre β_2 est négatif, et inversement.

Le modèle de Nelson Siegel ne peut pas être adopté dans toutes les circonstances réelles. En effet, la reproduction des "bosses" ou des "creux" de certaines courbes de taux est difficile. Par conséquent, le modèle de Nelson Siegel augmenté propose l'introduction de deux nouveaux paramètres pour plus de précisions.

Nelson Siegel Svensson :

Le modèle de Nelson Siegel Svensson (NSS) propose l'ajout d'un deuxième paramètre d'échelle ainsi qu'un deuxième facteur de courbure au modèle NS :

$$y(\tau) = \beta_0+\beta_1\cdot\frac{1-e^{-\tau/\lambda_1}}{\tau/\lambda_1}+\beta_2\cdot\left(\frac{1-e^{-\tau/\lambda_1}}{\tau/\lambda_1}-e^{-\tau/\lambda_1}\right)+\beta_3\cdot\left(\frac{1-e^{-\tau/\lambda_2}}{\tau/\lambda_2}-e^{-\tau/\lambda_2}\right)$$

avec λ_1 : le premier paramètre d'échelle
λ_2 : le deuxième paramètre d'échelle
β_0 : le facteur de niveau
β_1 : le facteur de rotation
β_2 : le facteur de pente
β_3 : le facteur de courbure supplémentaire

Ainsi, grâce aux deux paramètres supplémentaires, le modèle NSS permet de mieux s'adapter aux différentes maturités à interpoler. Nelson Siegel Svensson possède donc des meilleures qualités d'ajustement aux différentes courbes.

Utilisation de NS et NSS :

Ces deux modèles permettent d'interpoler les taux zéro-coupon, obtenus grâce à une méthode de Bootstrapping ou encore à une méthode matricielle (Ces méthodes seront développées dans la suite du rapport).

Dans le cadre de ce bureau d'études, l'optimisation des paramètres a été faite grâce au modèle de Nelson Siegel. Ces derniers ont ensuite été utilisés dans le modèle de Nelson Siegel Svensson permettant ainsi d'obtenir la courbe voulue (voir partie suivante).

Deuxième partie

Construction de courbes de taux

3 Calcul du taux de rendement actuariel

Le rendement actuariel pour une obligation est le rendement que peut espérer obtenir un investisseur qui conserverait une obligation jusqu'à son terme, en réinvestissant tous les coupons intermédiaires au même taux.
En pratique, ce n'est jamais le cas, puisque les taux évoluent, et que les investisseurs ne gardent pas systématiquement les obligations jusqu'à leur terme, ce taux permet néanmoins d'avoir un aperçu du rendement espéré pour l'obligation considérée.
Le taux actuariel est le taux de rendement qui annule la Valeur Actuelle Nette (VAN). Pour l'obtenir, il s'agit de déterminer r tel que :

$$P_{dirty} = \sum_i \frac{CF_{t_i}}{(1+r)^{t_i}}$$

avec P_{dirty} : le Dirty Price
CF_{t_i} : le Cash Flow obtenu à l'instant t_i

Autrement dit, CF_{t_i} correspond au Cash Flow obtenu à l'instant t_i, c'est à dire au coupon pour une obligation lorsque i < n. La dernière échéance correspond à CF_{t_n} = Coupon + Nominal. Pour obtenir le taux actuariel, il faut donc trouver, la valeur permettant d'annuler la fonction $P_{dirty} - VAN(r)$, avec

$$VAN(r) = \sum_i \frac{CF_{t_i}}{(1+r)^{t_i}}$$

Nous pouvons alors le faire via une optimisation sur R de la fonction $f(r) = P_{dirty} - VAN(r)$

4 Taux zéro-coupon

4.1 Méthode utilisée par l'institut des actuaires

Depuis l'arrêté du 26 décembre 2000 modifiant les articles à 332 est à 334 du code des assurances, de nouvelles exigences ont été mises en place pour les compagnies d'assurance en matière de gestion actif-passif.

Il est donc nécessaire pour les assureurs d'avoir accès à une courbe de taux fiable et ceci de manière actualisée. De plus, Solvabilité II prévoit de mesurer la sensibilité du bilan d'une compagnie d'assurance aux variations des taux d'intérêt.
C'est pourquoi l'institut des actuaires, association professionnelle des actuaires français a considéré qu'il était de son devoir de répondre à ce nouveau besoin des assureurs. En effet, on trouve très facilement de nombreuses courbes de taux dans la gestion actif-passif mais aucune ne peut servir de référence commune aux assureurs.
La courbe de taux de référence publiée par l'institut des actuaires est une courbe destinée aux assureurs pour la valorisation de leur passif, elle est cohérente avec les évaluations de l'actif car elle est calculée à partir de cours de marché .
Un comité scientifique "courbe des taux" a été constitué au sein de l'IA, il regroupe des spécialistes des marchés financiers et de la gestion actif-passif.
Le calcul pratique a été confiée à FININFO, spécialiste européen de l'analyse obligataire.
Le modèle utilisé est celui de Vasicek Fong.

Le comité scientifique a de plus fixé un certain nombre de règles pour l'élaboration de la courbe, afin de garantir sa fiabilité :
− Dans l'échantillon permettant de faire le calcul, il ne peut y avoir que des bons du trésor, des emprunts d'État, et des OAT dont le montant en circulation est supérieur à 8 milliards d'euros.
− Les cours retenus dans les calculs sont des cours de clôture officiels du dernier jour de bourse du trimestre. Ainsi, on peut noter une cohérence avec l'évaluation des actifs du bilan.

Une fois par trimestre, dans les jours suivants sa fin, le comité se réunit à fin d'optimiser les paramètres et valider les calculs. Il est possible, dès les premiers jours du trimestre d'accéder à la courbe sur le site Internet de l'IA.
Cette courbe se présente sous la forme d'un tableau Excel comprenant un taux zéro coupon par mois sur une période de 100 ans. Ce taux est un taux zéro coupon actuariel tronqué à 2 ou à cinq décimales. Il est calculé à partir des facteurs d'actualisation résultant du modèle de Vasicek -Fong c'est-à-dire de la forme $(1 + r)^t$.

Les valeurs actuelles des engagements des assureurs peuvent alors être calculées facilement en utilisant la courbe. Il suffit en effet d'actualiser chacun des flux futurs au taux zéro-coupon préconisé.
Le recours à une courbe zéro coupon pour l'évaluation des passifs des compagnies d'assurances reflète l'évolution en cours pour ce qui est de

l'approche retenue dans le cadre de l'analyse des équilibres actif-passif. L'institut des actuaires soutient cette démarche et souhaite la généraliser en introduisant les scénarios d'évolution des taux à plusieurs paramètres.

Nous allons maintenant détailler la méthodologie utilisée pour le calcul de ces taux.

Dans de nombreux modèles, on représente le taux zéro coupon par un ou plusieurs polynôme, voire des exponentielles de polynômes.

La méthode préconisée par l'institut ne représente pas directement les taux zéro-coupons, mais plutôt les coefficients d'actualisation, notés A, en reprenant la méthode de Vasicek-Fong.

La première idée sur laquelle repose la méthode est un changement de l'échelle de temps :

$$X = 1 - e^{-at}$$

Ainsi, pour t entre 0 et l'infini, X croît de 0 à 1.
Le choix de a devra être optimisé à posteriori.
Ce changement permet de mieux tenir compte dans l'estimation de A du fait qu'il y a peu de flux au delà de 25 ans.

- On utilise deux polynômes $P_1(X)$ et $P_2(X)$ pour représenter A, en raccordant de manière continue en valeur et en dérivée à une date t_c.
- On prend pour P_1 un polynôme de degré 3 et pour P_2 de degré 5.

Posons :

$$X_C = 1 - e^{-at_c}$$

avec $0 < X_C < 1$
On a :

$$A(X) = \left\{ \begin{array}{lcll} P_1(X) & = & a_0 + a_1 \cdot X + a_2 \cdot X_2 & X \leq X_C \\ P_2(X) & = & P_1(X) + (X - X_C)^2(b_0 + b_1 \cdot X + b_2 \cdot X_2 + b_3 \cdot X_3) & X > X_C \end{array} \right.$$

avec (a_i, b_j) : des coefficients à estimer
t_C : un paramètre à choisir

Grâce à cette représentation, on obtient une bonne approximation avec des degrés beaucoup plus faibles que si l'on utilisait un seul polynôme.
En notant r le taux d'actualisation continu associé à A(X),
On a :

$$A(X) = e^{-rt}$$

a) pour $t = X = 0$, le coefficient d'actualisation est égal à 1, donc :
$a_0 = 1$)
b) Quand t tend vers l'infini, et X vers 1, le coefficient d'actualisation tends vers 0, d'où :

$$A(1) = P_2(1) = 0$$

c) Estimation de A(X) :
On a N produits de taux O_n (obligations, BTAN ,BTF,...). Chaque produit O_n est caractérisé à la date t_0 par son prix de marché P'_n et son échéancier $(t_{k,n}, F_{k,n})$.
avec $F_{k,n}$: le flux à la date $t_{k,n}$
k varie de 1 à K_n.
Utilisant les coefficients d'actualisations A(X) et en posant :

$$X(k,n) = 1 - e^{-at_{k,n}}$$

On a pour valeur théorique de O_n :

$$P_n = \sum_{i=1}^{N} F_{k,n} A(X(k,n))$$

avec $k = 1, ..., n$.
Pour un choix de a et de t_c, on estime les coefficients (a_i, b_j) de A(X) en minimisant :

$$\sum_{i=1}^{N} (P'_n - P_n)^2 \qquad (1)$$

sous les contraintes décrites précédemment.

L'avantage de cette méthode est que P_n est linéaire en les coefficients (a_i, b_j) , on peut donc résoudre exactement (1).

Par manque de temps pour développer notre travail sur la méthode de Vasicek-Fong, cette partie n'est pas utilisée dans notre application. Il serait cependant intéressant de parvenir à l'implémenter sous R.

4.2 Méthode bootstrap

Cas obligataire :
Le bootstraping est une procédure permettant de reconstituer une courbe

zéro-coupon pas à pas, c'est-à-dire de proche en proche selon les maturités des obligations étant à disposition.
En pratique, la méthode est la suivante :

Dans un premier temps, on traite les obligations dont les maturités sont inférieures ou égales à un an ($t \leq 1$). En effet, puisqu'aucun coupon n'est versé entre le moment de calcul du taux ZC et l'échéance, il suffit de résoudre une équation simple pour obtenir le taux désiré :

$$r = \left(\frac{nominal}{prix} - 1\right) \cdot \frac{12}{m}$$

avec prix : le prix de marché
m : la maturité (inférieure ou égale à un an)

Puis, dans un second temps, on étudie les titres dont les maturités sont supérieures à un an (t>1).
Tout d'abord, on considère les obligations ayant une maturité de 2 ans ($t = 2$). Grâce à la première étape, le premier flux peut être calculé avec la formule suivante :

$$VA^1_{2y} = \frac{C}{(1 + r_{1y})}$$

avec C : le coupon
r_{1y} : le taux de maturité un an
VA^1_{2y} : la valeur actualisée du premier flux de l'obligation

Connaissant le prix de l'obligation et la valeur actualisée du premier flux, on peut en déduire le second flux.
En effet, comme expliqué précédemment, le prix d'une obligation est la somme des flux actualisés. :

$$VA^2_{2y} = P_{1<t\leq 2} - VA^1_{2y}$$

avec $P_{1<t\leq 2}$: le prix de l'obligation de maturité 2 ans
VA^1_{2y} : la valeur actualisée du premier flux de l'obligation
VA^2_{2y} : la valeur actualisée du second flux de l'obligation

Pour connaître le taux ZC, noté r_{2y}, du second flux, il suffit donc de résoudre l'équation suivante :

$$VA^2_{2y} \cdot (1 + r_{2y})^t$$

avec VA_{2y}^2 : la valeur actualisée du second flux de l'obligation
r_{2y} : le taux ZC à retrouver
t : la maturité

Pour les maturités supérieures ou égales à trois ans, le taux ZC s'obtient en itérant la procédure. Autrement dit, les taux r_{1y} et r_{2y} permettront de trouver VA_{3y}^3 et le taux r_{3y}, ainsi de suite.

Exemple :

Considérons une obligation d'une maturité de 5 ans avec un nominal de 100€ et des coupons annuels de 3,5%. Les flux des obligations actualisés au taux zéro coupon de l'échéance correspondante permettent de donner le prix théorique de l'obligation.

Échéances	**Coupons**	**ZC (en %)**	**Valeurs actualisées**
1	3,50	2,15	3,426
2	3,50	2,64	3,322
3	3,50	3,15	3,189
4	3,50	3,45	3,056
5	103,50	3,63	86,599
Prix théorique de l'obligation			99,593

TABLE 5 – Calculs du prix théorique de l'obligation considérée

Afin de construire notre courbe zéro coupon, prenons comme point de départ des obligations avec différentes maturités comme suit :

Maturités (en année)	**Coupons (en %)**	**Prix de l'obligation**
0,50	0	99,05
0,75	0	98,45
1	0	97,85
2	3,50	101,40
3	4,00	102,20

TABLE 6 – Liste d'obligations de différentes maturités

Les trois premières obligations sont de maturité inférieure (ou égale) à un an : ce sont donc des obligations sans flux intermédiaires. De ce fait, les taux zéro coupon, notés r, peuvent être calculés directement. On obtient alors pour les taux zéro coupon r_{6m}, r_{9m} et r_{1y} des obligations respectives de maturité 6 mois, 9 mois et un an :

$$r_{6m} = \left(\frac{100}{99,05} - 1\right) \cdot \frac{12}{6} = 1,918\%$$

$$r_{9m} = \left(\frac{100}{98,45} - 1\right) \cdot \frac{12}{9} = 2,099\%$$

$$r_{1y} = \left(\frac{100}{97,85} - 1\right) \cdot \frac{12}{12} = 2,197\%$$

Connaissant le taux zéro coupon de maturité un an, noté r_{1y}, le premier flux des deux dernières obligations de maturité supérieure à un an peut être calculé. Sachant que $r_{1y} = 2.1972\%$, on a :

$$VA^1_{2y} = \frac{3,5}{(1 + 2,197\%)} = 3,425$$

Le taux zéro coupon du deuxième flux de ce titre de maturité 2 ans peut être calculé par itération. Le prix de ce titre est, d'après la Table 7, 101,40. La valeur actualisée pour le deuxième flux de cette obligation est par conséquent : $101,40 - 3,425 = 97,975$.

Maturités	**Coupons**	**ZC (en %)**	**Valeurs actualisées**
1	3,50	2.197	3,425
2	103,50	?	97,975
Prix de l'obligation			101,400

Table 7 – Obligation de maturité 2 ans

On cherche maintenant à connaître le taux auquel il faut placer la valeur actualisée du second flux, soit 97,975 €, permettant d'obtenir 103,50 € au bout de deux ans.

$$97,975 \cdot (1 + r)^2 = 103,50$$

On obtient donc $r = 2.781\%$.

Par itération, on obtient pour l'obligation de maturité 3 ans :

Ainsi, les taux zéro coupon sont obtenus avec la méthode du Bootstrapping. La courbe zéro coupon est obtenue grâce à l'interpolation par les

Échéances	Coupons	ZC (en %)	Valeurs actualisées
1	4,00	2,197	3,914
2	4,00	2,781	3,892
3	104,00	3,283	94,394
Prix de l'obligation			102,20

TABLE 8 – Obligation de maturité 3 ans

modèles NS et NSS.

Maturités (en année)	ZC (en %)
0,5	1,918
0,75	2,099
1	2,197
2	2,781
3	3,283

TABLE 9 – Zéro coupon obtenu avec la méthode du Bootstrapping

Cas taux swap :
D'abord il nous faut expliciter notre démarche de valorisation de la jambe variable d'un swap que nous utiliserons ensuite afin d'utiliser l'algorithme du Bootstrap.

Valorisation de la jambe variable :

Notations :
- $T_0 \geq 0$: date de démarrage du swap
- $(T_i)_{i=1,\ldots,N}$: dates de paiement de la jambe fixe
- δ_i : fraction d'année entre deux dates de paiement de la jambe fixe selon la convention du calcul du nombre de jour
- $S(T_0, T_N)$: taux de la jambe fixe constant sur toute la période
- $B^{bor}(t, T_n)$: valeur en t du facteur d'actualisation de maturité T_n, calculé à partir d'une courbe de taux LIBOR (inconnue, aléatoire), pour $T_0 < t \geq$T$_n$
- $B^{bor}(T_0, T)$: valeur en T_0 du facteur d'actualisation de maturité T, calculé à partir d'une courbe de taux LIBOR (connue, déterministe)
- $(S_i)_{i=1,\ldots,M}$: dates de paiement de la jambe flottante ($S_0 = T_0$ et $S_M = T_N$)
- γ_i : fraction d'année entre deux dates de paiement de la jambe variable
- $L(s, S)$: taux variable Libor (exemple Euribor 6M) sur la période

[s,S]
— $L(T_0, s, S)$: valeur forward du taux variable $L(s, S)$
— $\mathbb{Q}_T$: probabilité T-forward neutre $\forall\ T > 0$
— $\mathbb{E}^{\mathbb{P}}$: opérateur espérance sous la probabilité $\mathbb{P}$

Si nous faisons abstraction du risque de crédit, on actualise au même taux que le taux variable, la relation suivante est donc vérifiée :

$$B^{bor}(S_{i-1}, S_i)(1 + \gamma_i \times L(S_{i-1}, S_i)) = 1$$

On a donc :

$$L(S_{i-1}, S_i) = \frac{1-B^{bor}(S_{i-1},S_i)}{\gamma_i B^{bor}(S_{i-1},S_i)} = \frac{B^{bor}(S_{i-1},S_{i-1})-B^{bor}(S_{i-1},S_i)}{\gamma_i \times B^{bor}(S_{i-1},S_i)}$$

$$\Rightarrow L(S_{i-1}, S_i) = \frac{1}{\gamma_i}(\frac{B^{bor}(S_{i-1},S_{i-1})}{B^{bo}(S_{i-1},S_i)} - \frac{B^{bor}(S_{i-1},S_i)}{B^{bor}(S_{i-1},S_i)}$$

Etant donné que nous avons choisi $B^{bor}(., S_i)$ comme numéraire associé à la probabilité $\mathbb{Q}_{S_i}$-forward neutre $\mathbb{Q}_{S_i}$, toute combinaison linéaire d'actifs négociables (dans notre cas ce sont des zéros coupons sans risques) divisés par le numéraire $N^{bor}(., S_i)$ est donc une martingale sous $\mathbb{Q}_{S_i}$. De ce fait nous obtenons :

$$E^{\mathbb{Q}_{S_i}}(L(S_{i-1}, S_i)|S_0) = \frac{1}{\gamma_i} \times E^{\mathbb{Q}_{S_i}}((\frac{B^{bor}(S_{i-1},S_{i-1})}{B^{bor}(S_{i-1},S_i)} - \frac{B^{bor}(S_{i-1},S_i)}{B^{bor}(S_{i-1},S_i)}|S_0)$$

D'où $\forall i = 1, ..., M$

$$\boxed{E^{\mathbb{Q}_{S_i}}(L(S_{i-1}, S_i)|S_0) = \frac{1}{\gamma_i} \times (\frac{B^{bor}(S_0, S_{i-1}) - B^{bor}(S_0, S_i)}{B^{bor}(S_0, S_i)} = L(S_0, S_{i-1}, S_i)}$$

Valorisation de la jambe variable :

$$JV_0 = \sum_{\substack{i=1\\i}}^{M} E^{\mathbb{Q}_{S_i}}(L(S_{i-1}, S_i)|S_0) \times \gamma_i \times B^{bor}(S_0, S_i)$$

$$= \sum_{\substack{i=1\\i}}^{M} (\frac{B^{bor}(S_0,S_{i-1})-B^{bor}(S_0,S_i)}{\gamma_i \times B^{bor}(S_0,S_i)}) \times \gamma_i \times B^{bor}(S_0, S_i)$$

$$= \sum_{\substack{i=1\\i}}^{M} B^{bor}(S_0, S_0) - B^{bor}(S_0, S_M)$$

D'où

$$\boxed{JV_0 = 1 - B^{bor}(T_0, T_N)}$$

Application de l'algorithme du Bootstrap :

Nous prenons un panier de swaps d'un même indice de taux variable de différentes maturité dont la périodicité de la jambe fixe est de 1 an et la périodicité de la jambe variable est la même que celle de l'indice du taux variable.
Pour calculer les zéros coupons à partir des taux swap de différentes maturités trouvés sur le marché, nous appliquons la même méthode du bootstraping explicitée précédemment pour les taux zéros coupons obligataires.

Pour notre exemple nous avons trouvé un panier de swaps de taux sur le marché au 06/06/2006 dont la périodicité de la jambe variable et de la jambe fixe sont de 1 an et le taux variable échangé est l'EURIBOR 6 mois, noté E6M :

Maturités	**Taux swaps (en %)**
06 Août 2006	2,6
06 Août 2007	2,61
06 Août 2008	2,73
06 Août 2009	2,84
06 Août 2010	3,32

Table 10 – Panier Swap

Dans un premier temps, on traite les swaps de taux dont les maturités sont inférieures ou égales à un an ($t \leq 1$). En effet, puisqu'aucun intérêt n'est versé entre le moment de calcul du taux ZC et l'échéance, il suffit de résoudre une équation simple pour obtenir le taux désiré :

$$DF = \left(\frac{1}{1 + t_s.\Delta}\right)$$

$$ZC = \left(\frac{1}{DF}\right)^{\frac{1}{\Delta}} - 1$$

avec t_s : le taux swap coté sur le marché
Δ : le temps jusqu'à la maturité (inférieure ou égale à un an)

Exemple :

$$DF = \left(\frac{1}{1 + 0,026 \cdot \frac{2}{12}} \right)$$

$$\Leftrightarrow DF = 0,996$$

$$ZC = \left(\frac{1}{0,996} \right)^6 - 1$$

$$\Leftrightarrow ZC = 0,0263$$

Puis, dans un second temps, on étudie les titres dont les maturités sont supérieures à un an ($t > 1$).
Nous avons vu précédemment que la jambe variable d'un swap est égale à

$$1 - DF(t0, tn)$$

De plus nous savons que : La jambe variable est égale à la jambe fixe et que la jambe fixe d'un swap se valorise à :

$$tauxswap(t0, tn) * (DF(t0, t1)...DF(t0, tn))$$

Nous avons donc la formule suivante :

$$1 - \frac{Tauxswap}{(1 + ZC_{06Aot2007})^{\frac{14}{12}}} = \frac{Tauxswap}{(1 + ZC_{06Aot2006})^{\frac{2}{12}}} + \frac{Tauxswap}{(1 + ZC_{06Aot2007})^{\frac{14}{12}}}$$

Nous obtenons alors :

$$1 = \frac{Tauxswap}{(1 + ZC_{06Aot2006})^{\frac{2}{12}}} + \frac{1 + Tauxswap}{(1 + ZC_{06Aot2007})^{\frac{14}{12}}}$$

Exemple :

$$1 = \frac{0,0261}{(1 + 0,0263)^{\frac{2}{12}}} + \frac{1 + 0,0261}{(1 + ZC_{06Aot2007})^{\frac{14}{12}}}$$

Par itération, on trouve les résultats suivants :

$$ZC_t = \begin{pmatrix} 0,0263 \\ 0,0456 \\ 0,0382 \\ 0,0362 \\ 0,0406 \end{pmatrix}$$

La courbe des zéros coupons est alors obtenue grâce à l'interpolation des modèles NS et NSS, comme expliquée précédemment.

4.3 Méthode matricielle

Cas obligataire et taux swap :
Afin d'obtenir une courbe de zéros coupons nous reprenons les discount factors déterminés grâce aux calculs effectués ci-dessous selon la méthode choisie.
Puis, nous utilisons la relation suivante entre les facteurs d'actualisation et les taux zéros coupons :

$$DF(t,T) = \left(\frac{1}{1 + ZC(t,T)}\right)^{\Delta_{(t,T)}}$$

avec $DF(t,T)$: le Discount Factor de t à T
$ZC(t,T)$: le taux zéro coupon de t à T
$\Delta(t,T)$: la durée en année entre t et T dans la base choisie

La courbe des zéros coupons est ensuite obtenue par interpolation en utilisant les modèles NS et NSS, via la méthode explicitée ci-avant.

5 Courbe des discount factors

5.1 Méthode Bootstrap

Cas obligataire et taux swap :
Afin d'obtenir une courbe de facteurs d'actualisation nous reprenons les taux zéro coupons déterminés grâce aux calculs effectués ci-dessus selon la méthode choisie.
Puis nous utilisons la relation suivante entre les facteurs d'actualisation et

les taux zéros coupons :

$$DF(t,T) = \left(\frac{1}{1 + ZC(t,T)}\right)^{\Delta_{(t,T)}}$$

avec $DF(t,T)$: le Discount Factor de t à T
$ZC(t,T)$: le taux zéro coupon de t à T
$\Delta(t,T)$: la durée en année entre t et T dans la base choisie

La courbe des Discount factors est obtenue grâce à l'interpolation par les modèles NS et NSS, comme expliqué précédemment.

5.2 Méthode matricielle

Cas obligataire :
La méthode matricielle correspond à une méthode de résolution matricielle de l'algorithme du Bootstrap dans un cas particulier. Nous travaillons toujours selon l'hypothèse qu'il n'y a pas de risque de réinvestissement des coupons. Nous définissons ainsi les matrices et vecteurs suivants :

$$P_t = \begin{pmatrix} P_{t=1} \\ P_{t=2} \\ P_{t=3} \\ \vdots \\ P_{t=T} \end{pmatrix} ; DF_t = \begin{pmatrix} DF_{t=0,t=1} \\ DF_{t=0,t=2} \\ DF_{t=0,t=3} \\ \vdots \\ DF_{t=0,t=T} \end{pmatrix}$$

$$C = (C_{i,j})_{1 \leq i \leq n, 1 \leq j \leq n} = \begin{pmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,n} \\ C_{2,1} & \cdots & \cdots & C_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n,1} & C_{n,2} & \cdots & C_{n,n} \end{pmatrix}$$

avec P_t : le vecteur des prix à l'instant t
DF_t : le vecteur des Discount Factor
$C_{i,j}$: la matrice correspondant aux flux sortant de l'obligation i à la j^{ime} année $\forall$ 1$\leq$ i,j$\leq$n

Nous en déduisons donc la relation suivante :

$$C \cdot DF_t = P_t \Rightarrow DF_t = C^{-1} \cdot P_t$$

Exemple :

Nous avons pris le panier d'obligations à la date du 03/05/2016 suivant :

Dates de maturité	**Coupons (en %)**	**Nominal (N)**	**Prix de l'obligation (P_t)**
03 Mai 2017	0	100	100,60
03 Mai 2018	0	100	102,96
03 Mai 2019	1	100	104
03 Mai 2020	0	100	101
03 Mai 2021	0	100	100,5

TABLE 11 – Panier d'obligations

$$P_t = \begin{pmatrix} 100,60 \\ 102,96 \\ 104 \\ 101 \\ 100,5 \end{pmatrix} ; DF_t = \begin{pmatrix} DF_{t=0,t=1} \\ DF_{t=0,t=2} \\ DF_{t=0,t=3} \\ \vdots \\ DF_{t=0,t=T} \end{pmatrix}$$

$$C = (C_{i,j})_{1\leq i\leq n, 1\leq j\leq n} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Cette méthode présente l'avantage d'une résolution bien plus rapide puisqu'il suffit d'inverser la matrice des flux pour obtenir les facteurs d'actualisation.

$$DF_t = \begin{pmatrix} 1,00 \\ 1,00 \\ 1,01 \\ 0,99 \\ 0,98 \end{pmatrix}$$

Cependant cette méthode a le désavantage que les obligations doivent avoir leurs coupons échéants à la même date. De plus dans le cadre de cette méthode il ne peut y avoir de décalage entre les maturités des obligations supérieures à 1 an. Ces désavantages représentent une grande difficulté en pratique puisqu'il est difficile de trouver un paquet d'obligations assez volumineux (pour pouvoir faire des prédictions) respectant ces conditions.

La courbe des Discount factors est obtenue grâce à l'interpolation par les modèles NS et NSS, comme expliqué précédemment.

Cas taux swap :

Nous prenons un panier de swaps d'un même indice de taux variable de différentes maturité dont la périodicité de la jambe fixe est de 1 an et la périodicité de la jambe variable est celle de l'indice du taux variable.
Pour calculer les discount factors à partir des taux swap de différentes maturités trouvés sur le marché, nous appliquons la même méthode que pour l'algorithme du bootstrap, explicitée précédemment pour les taux zéros coupons.
Nous avons montrer que :

$$1-DF(t=0,t=T) = tauxswap(t=0,t=T)*(DF(t=0,t=1)...DF(t=0,t=T))$$

$$1 = tauxswap(t=0,t=T)*(DF(t=0,t=1)...DF(t=0,t=T-1))+(1+tauxswap(t=0,t=T))*$$

$$P = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} ; DF_t = \begin{pmatrix} DF_{t=0,t=1} \\ DF_{t=0,t=2} \\ DF_{t=0,t=3} \\ \vdots \\ DF_{t=0,t=T} \end{pmatrix}$$

$$C = \begin{pmatrix} tauxswap_{t=0,t=1} & \cdots & tauxswap_{t=0,t=1} & 1+tauxswap_{t=0,t=1} \\ tauxswap_{t=0,t=2} & \cdots & tauxswap_{t=0,t=2} & 1+tauxswap_{t=0,t=2} \\ \vdots & \vdots & \vdots & \vdots \\ tauxswap_{t=0,t=T} & \cdots & tauxswap_{t=0,t=T} & 1+tauxswap_{t=0,t=T} \end{pmatrix}$$

avec P : le nominal des swaps (qui est pris égal à 1)
DF_t : le vecteur des Dicount Factor
$C_{i,j}$: la matrice construite à partir des taux swaps

Nous en déduisons donc la relation suivante :

$$C \cdot DF_t = P \Rightarrow DF_t = C^{-1} \cdot P$$

<u>Exemple :</u>

Pour notre exemple nous avons trouvé un panier de swaps de taux sur le marché au 08/06/2005 dont la périodicité de la jambe variable et de la jambe fixe sont de 1 an et le taux variable échangé est l'EURIBOR 6 mois, noté E6M :

Dates de maturité	**Taux swaps (en %)**
06 Août 2006	2,6
06 Août 2007	2,61
06 Août 2008	2,73
06 Août 2009	2,84
06 Août 2010	3,32

TABLE 12 – Panier

$$P = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} ; DF_t = \begin{pmatrix} DF_{t=0,t=1} \\ DF_{t=0,t=2} \\ DF_{t=0,t=3} \\ \vdots \\ DF_{t=0,t=T} \end{pmatrix}$$

$$C = \begin{pmatrix} 1,026 & 0 & 0 & 0 & 0 \\ 0,026 & 1,026 & 0 & 0 & 0 \\ 0,027 & 0,027 & 1,027 & 0 & 0 \\ 0,028 & 0,028 & 0,028 & 1,028 & 0 \\ 0,032 & 0,032 & 0,032 & 0,032 & 1,032 \end{pmatrix}$$

A nouveau, cette méthode présente l'avantage d'une résolution bien plus rapide puisqu'il suffit d'inverser la matrice des flux pour obtenir les facteurs d'actualisation.

$$DF_t = \begin{pmatrix} 0,97 \\ 0,95 \\ 0,92 \\ 0,89 \\ 0,85 \end{pmatrix}$$

Cependant cette méthode a le désavantage que les taux swaps doivent alors avoir leurs intérêts échéants à la même date. De plus il ne peut y avoir de décalage supérieur à un an entre les maturités des swaps. Ces désavantages représentent une grande difficulté en pratique pour les raisons déjà explicitées.

La courbe des Discount Factors est obtenue grâce à l'interpolation par les modèles NS et NSS, comme expliqué précédemment.

6 Courbe des taux forward

Afin de calculer les taux swaps, une première méthode consiste à utiliser les taux Forwards.
Pour les obtenir, nous pouvons utiliser les taux zéros coupons, ou les discounts factors.
Néanmoins, la méthode pour calculer ces taux zéro-coupons diffère de celle que nous utilisions dans le pricer précédent.
Ce calcul permet de déterminer les taux qui prévaudront sur le marché dans le futur.
Nous allons ci-après vous présenter comment nous pouvons les calculer.

Exemple :

Si le taux zéro coupon 1 an est de 4 % et que celui 2 ans est 5 %, on peut alors déterminer le taux forward 1 an dans 1 an.
C'est à dire que si nous plaçons 100€ sur un an nous obtiendrons :

$$100 \times exp(0,04) = 104,08.$$

Et si nous plaçons 100 € sur deux ans, nous obtiendrons alors :

$$100 \times exp(2 \times 0,05) = 110,52.$$

On s'aperçoit donc, que si nous replaçons les 104,08€ obtenus après un an, nous devons arriver à 110.52€. D'où l'égalité :

$$104,08 \times exp(r) = 110,52$$

avec r est le taux forward 1 an dans un an.

On résout et on trouve 6%.
En itérant ainsi, nous pouvons obtenir les différents taux forwards.

Nous pouvons dès lors trouver une formule générale pour obtenir le taux forward entre deux dates d_1 et d_2 :

$$F_{t_1,t_2} = \left(\frac{(1+r_2)^{d_2}}{(1+r_1)^{d_1}} \right)^{\frac{1}{(d_2-d_1)}} - 1$$

avec $F(t_1, t_2)$: le taux forward entre t_1 et t_2
d_i : le nombre d'années entre la date initiale t_i
r_i : le taux zéro coupon de l'année i

Troisième partie

Construction d'un pricer de swap

Grâce à l'aide de la théorie financière développée ci-dessus, nous allons chercher à déterminer le taux swap correspondant au taux fixe échangé selon les autres caractéristiques de celui-ci.

On rappelle que les caractéristiques du swap sont :
– Nominal, noté N : capital initial emprunté par l'émetteur de l'obligation divisé par le nombre de titres émis
– Date de valorisation
– Maturité, notée M
– Périodicité de jambe fixe, notée Pf
– Périodicité de jambe variable, notée Pv
– Périodicité du taux variable
– Taux fixe d'un swap, noté t_s - qui est coté
– Taux variable, noté t_v
– Conventions de base de calcul

7 Cas général

On rappelle de plus que d'après la définition du swap de taux, celui-ci est composé d'une jambe variable et d'une jambe fixe dont les valeurs actuelles, notées VA_{JF} et VA_{JV}, sont définies comme suit :

$$VA_{JF} = N \cdot \sum_i (t_s \cdot \alpha_i) \cdot DF_i$$

$$VA_{JV} = N \cdot \sum_i (Fwd_i \cdot \alpha_i) \cdot DF_i$$

avec N : le nominal
DF_i : le facteur d 'actualisation en $t = t_i$
Fwd_i : le taux forward pour la période $[t_{i-1}; t_i]$ calculé en t_{i-1}
α_i : le daycount dans la base choisie
ts : le taux swap coté

Or dans un swap, les valeurs actualisées des deux jambes doivent être égales :

$$VA_{JF} = VA_{JV}$$

$$N \cdot \sum_i (t_s \cdot \alpha_i) \cdot DF_i = N \cdot \sum_i (Fwd_i \cdot \alpha_i) \cdot DF_i$$

Nous déduisons donc de l'équation suivante la formule déterminant le taux swap :

$$t_s = \frac{\sum_i (Fwd_i \cdot \alpha_i) \cdot DF_i}{\sum_i (\alpha_i) \cdot DF_i}$$

Selon les périodicités des jambes du swap Pf et Pv, nous calculons les taux zéro coupons, Discount Factor et taux Forward de la manière décrite précédemment avec l'une ou l'autre méthode pour chaque paiement d'intérêts à taux fixe ou variable.

Exemple :

Nous appliquons les démarches effectuées précédemment à la date du 06 Juin 2006 pour l'exemple suivant :
Nous cherchons à pricer un swap de taux ayant les caractéristiques suivantes :

N	**Date de Valorisation**	**M**	**Pf**	**Pv**	**ts**	**tv**
100	06/06/2016	4 ans	6 mois	1 an	?	E6M

Table 13 – Taux du swap ts à trouver

8 Cas particulier

Nous présentons ici un cas particulier assez fréquent dans la pratique permettant de simplifier les calculs.
Nous prenons donc le cas où la périodicité de la jambe variable est égal à la périodicité de l'indice du taux variable et la périodicité de la jambe fixe égal à 1 an.

Nous avons donc la simplification suivante de la jambe variable :

$$VA_{JV} = N \cdot \sum_i (Fwd_i \cdot \alpha_i) \cdot DF_i$$

$$VA_{JV} = N \cdot \sum_i (\frac{1}{\alpha_i} \cdot (\frac{DF_{i-1}}{DF_i} - 1) \cdot \alpha_i) \cdot DF_i$$

$$VA_{JV} = DF_0 - DF_n$$

avec N : le nominal
DF_i : le facteur d 'actualisation en $t = t_i$
Fwd_i : le taux forward pour la période $[t_{i-1}; t_i]$ calculé en t_{i-1}
α_i : le daycount dans la base choisie
ts : le taux swap coté
Nous en déduisons donc la relation suivante de part l'égalité des deux jambes :

$$t_s = \frac{DF_0 - DF_n}{\sum_i (\alpha_i) \cdot DF_i}$$

Nous calculons les Discount Factor de la manière décrite précédemment avec l'une ou l'autre méthode pour chaque paiement d'intérêt de taux fixe.

Exemple :

Nous cherchons à pricer un swap de taux ayant les caractéristiques suivantes :

N	**Date de Valorisation**	**M**	**Pf**	**Pv**	**ts**	**tv**
100	06/06/2016	4 ans	6 mois	6 mois	?	E6M

TABLE 14 – Taux du swap ts à trouver

Quatrième partie

Application sous l'interface Shiny

9 Choix et établissement de R serveur

Le but de notre projet est de rendre accessible aux utilisateurs l'utilisation d'outils financiers en ligne. Pour se faire, nous disposions de diverses possibilités. Nous allons, ci-après, vous décrire le processus qui nous a permis de choisir l'interface offerte par R Shiny, ainsi que les enjeux qui ont motivé ce choix. Nous parlerons également des choix effectués quant à la mise en ligne de notre travail (RServer et ShinyApps.io).

Shiny est un package R disponible en open source, qui permet de créer une interface ergonomique sous forme d'application. L'intérêt est de pouvoir créer une interface utilisable par tous, qui renvoie des graphes et des résultats pouvant être obtenus par l'exécution d'un code R. L'idée est d'automatiser la variation de paramètres de l'algorithme grâce à des curseurs ou des menus déroulants. Il est à noter que RShiny nécessite l'utilisation de l'interface RStudio.

L'un des intérêts de Shiny est qu'il n'est pas nécessaire d'avoir des connaissances en html, css ou Java.

RShiny fonctionne grâce à des fonctions input et des fonctions output.

Pour utiliser Shiny, il est impératif de générer un fichier ui.R qui contient l'interface et un fichier server.R qui contient le traitement. Il n'est pas possible d'utiliser Shiny directement depuis la console R.

Shiny n'est pas le seul à proposer ce service. On retrouve aussi Visual Studio ou encore les langages de programmation. Cependant, R Shiny présente plus d'avantages que ces autres concurrents.

En effet, le code utilisé pour R Shiny est similaire à celui de R, seule l'interface utilisée sous ShinyApps.io présente quelques nouveautés pour nous. Il nous était dès lors plus facile de modéliser un problème financier, étant déjà familiarisé avec R. De plus, le package R Shiny est disponible en libre accès, et les interactions avec l'utilisateur sont possibles.

Nous avions néanmoins envisagé d'autres possibilités, comme Visual Studio, Java, C++, ou encore Python. Cependant, ces alternatives étaient payantes, ou alors demandaient d'apprendre un nouveau langage, ce qui aurait été difficile à mettre en oeuvre dans le temps imparti.

10 Choix du serveur

Une fois arrêté notre choix quant au logiciel de programmation à utiliser, à savoir RShiny, nous nous sommes intéressés à la mise en ligne de notre travail. Pour rendre disponible notre travail sous RSHiny, plusieurs options s'offraient à nouveau à nous, que nous allons présenter ci-après.

10.1 RServer

RShiny propose deux versions de RServer, une version Open source en libre accès (gratuite), et une version pro payante. La version open source du RServer permet l'hébergement public d'applications de faible poids. Elle autorise les utilisateurs à utiliser simultanément l'application, cependant ceux-ci sont classés, ainsi les ordres seront exécutés un à un dans l'unique session R. Le RServer présente également le désavantage de tourner sous linux. Enfin, le code des applications sera en open source également, ce qui laisse la possibilité aux utilisateurs de le modifier.

La version Pro permet le déploiement de tout type d'application, tout en laissant la possibilité d'une utilisation par vingt utilisateurs simultanément. Elle présente également de nombreux avantages en terme de sécurité par rapport à la version gratuite. De plus, en cas de difficulté, la version Pro nous propose un support via mail avec garantie de réponse sous 8H (jours ouvrés).

Cette version semblait donc adaptée à notre projet, exceptés deux inconvénients :
- la nécessite de travailler sous linux (même si, cela ne posait pas de grosse difficulté).
- le prix.

Cependant, la version Pro, était également disponible gratuitement pour les universités proposant un cours sous RShiny. Pour obtenir la version gratuite, il fallait donc envoyer via une adresse académique, un syllabus. Cette partie serait donc gérée par nos tuteurs de bureau d'étude, si cette option était retenue.

10.2 ShinyApps.io

ShinnyApps.io est une plate forme permettant la mise en ligne d'application sous RShiny. ShinyApps.io propose 5 formules (Free, Starter, Basic, Standart, Professional), classées de la moins chère (gratuite) à la plus onéreuse. Un des intérêts de ShinyApps.io est de disposer directement d'un server .io, ce qui évite d'avoir à en créer un, évitant ainsi des difficultés éventuelles sur la mise en place de celui-ci. De plus, il n'y a pas de maintenance du server .io à effectuer. D'autre part, l'utilisation de ShinyApps.io, présente l'avantage d'être évolutive : en débutant avec la version gratuite, nous pouvions travailler rapidement, et éventuellement, envisager une version plus évoluée au fur et à mesure du développement de notre application, ainsi qu'une vision à long terme (lorsque plusieurs applications seraient hébergées, il serait assez facile de passer à la version supérieure si le besoin s'en fait sentir).

La version gratuite permet une utilisation de 25H mensuelle. De plus, nous disposions de contenu disponible quant à l'utilisation de ShinyApps.io. En effet, Maxime MALAL (EURIA promotion 2016), et Benjamin LEBOUCHER (EURIA promotion 2015) avaient tous les deux travaillé sous cette interface.

10.3 Décision pour la mise en pratique

De ce que nous avons exposé ci-dessus, nous pouvons retenir que l'utilisation d'un ServeurPro semblait la plus adaptée à notre projet. En effet, l'utilisation était la plus sûre, la plus complète et permettait d'envisager une poursuite de notre travail par les promotions ultérieures, ce qui était un point important. Nous souhaitons, en effet, pouvoir disposer de diverses applications dans les domaines financiers et actuariels, qui permettent une visibilité de l'école, ainsi que du travail effectué par ses élèves.
La version Pro est donc clairement la plus adaptée à cette volonté de pérennité. Son prix nous oblige à nous tourner vers la version académique qui offre les mêmes garanties. Cependant, nous ne pouvions attendre de recevoir la licence académique pour commencer notre projet, nous avons donc choisi de le débuter via Shinyapps.io.
Il nous serait ensuite possible de faire le transfert sous le R serveur aisément. De plus, comme expliqué plus haut, le travail effectué sous Shinyapps.io par les étudiants des promotions précédentes nous donnait déjà des bases

concrètes dans l'élaboration de notre projet.

11 Fonctionnement de RShiny

Pour utiliser RShiny deux fichiers sont nécessaires, un fichier ui.R, et un fichier server.R.

Le fichier « ui.R » permet de configurer la disposition et l'affichage de l'application. C'est ici qu'est gérée l'ergonomie de l'application et le rendu qui sera visible en ligne. Le fichier «Server.R » quant à lui contient les fonctions utilisées pour l'application, ainsi que les diverses instructions associées. La programmation de celles-ci se fait de la même manière que dans l'utilisation classique de RStudio.

Les deux fichiers ui.R, et server.R, doivent se trouver dans le même fichier au niveau de l'emplacement de l'ordinateur.
Il faut ensuite que l'emplacement de travail de R soit configuré sur le dossier correspondant. On utilise pour se faire la commande :
setwd("C/Emplacement-du-fichier")
Aperçu du fichier Server.R :

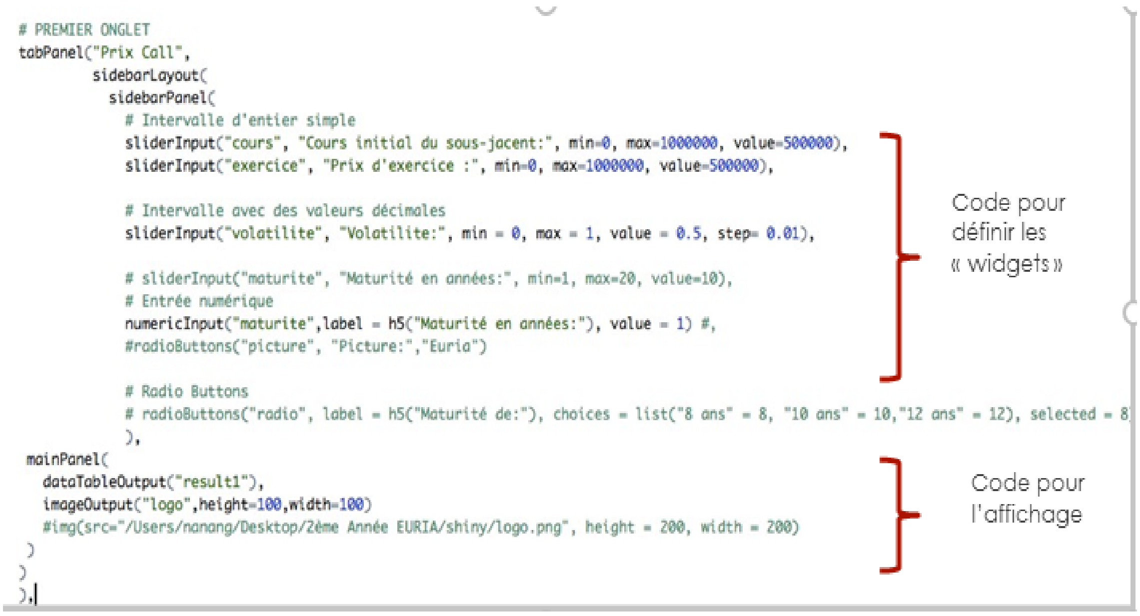


FIGURE 11 – Server.R

Nous allons à présent détailler le fonctionnement de notre application pour un utilisateur.

11.1 Pricing d'obligations

Nous proposons en particulier, à un utilisateur disposant d'un panier d'obligations d'effectuer différents calculs. Celui devra tout d'abord importer sur le site son panier d'obligations. Il disposera de type de formats selon, que les prix sont donnés sous la forme bid/ask (fichier de type 1) ou sous la forme midprice (fichier de type 2). Ces informations sont directement disponibles dans l'onglet « Explications » de notre application. Le format attendu y est également explicité. Il faudra veiller notamment à inscrire une date au format AAAA/MM/JJ qui sera antérieure aux maturités des obligations du panier. C'est à partir de cette date que sera calculée la courbe de taux.
Si l'importation s'est correctement déroulée, l'utilisateur a alors un aperçu de son fichier dans l'onglet calcul pricer, comme présenté ci-dessous :

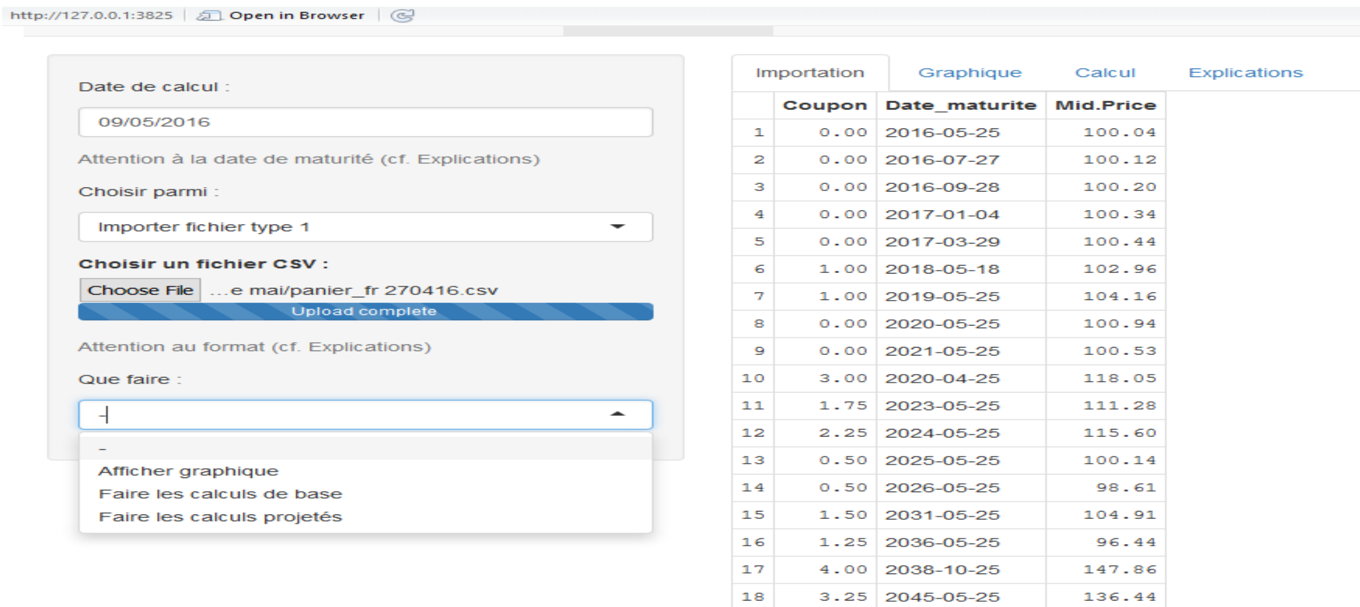
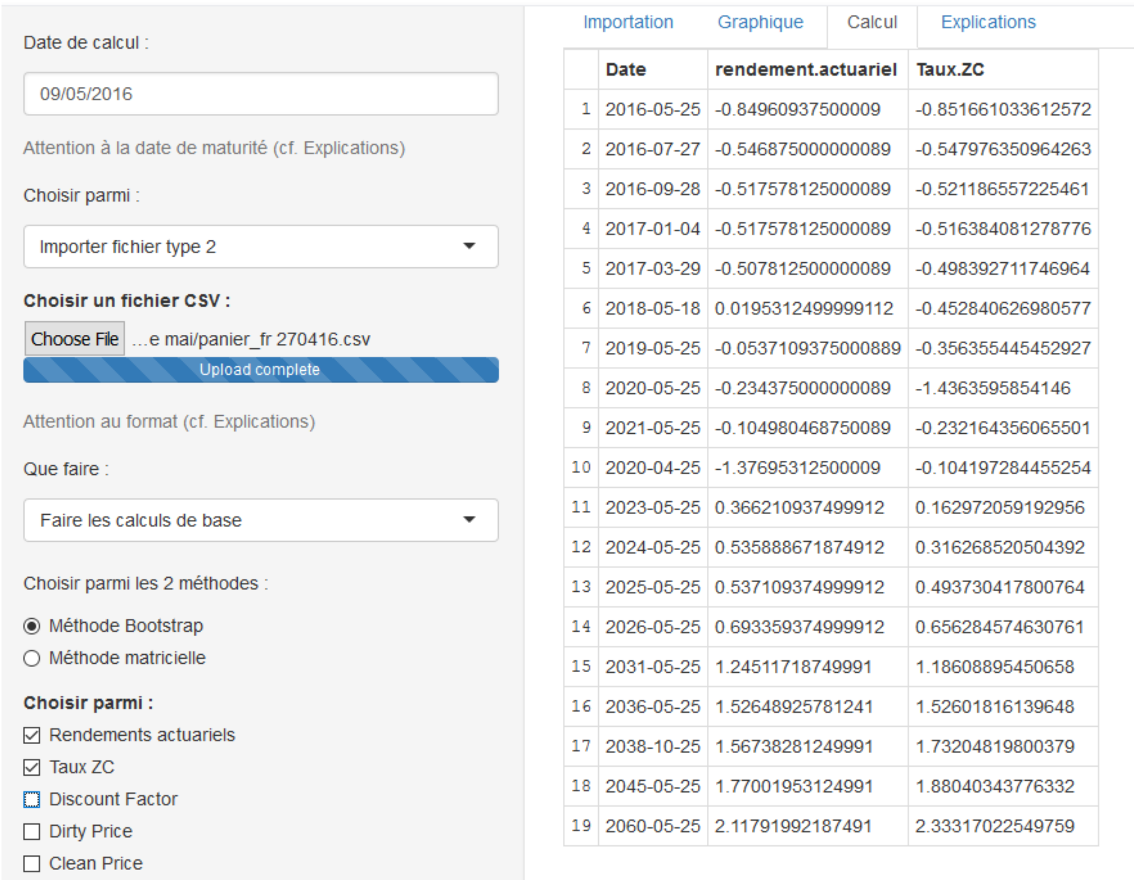


FIGURE 12 – Importation du panier

L'utilisateur dispose alors de différentes options, comme on le voit ci-dessus. Il peut en particulier afficher des graphiques, effectuer certains calculs de base, ou alors effectuer des projections.

Nous proposons ainsi le calcul des taux de rendement actuariels, des taux zéros coupons, des discount factors, du dirty price et du clean price. Ces notions ayant été présentées plus tôt dans notre rapport, nous ne reviendrons pas dessus. Une fois sélectionnés les calculs que l'on souhaite réaliser, les résultats s'affichent dans le second onglet "Calcul" :



	Date	rendement.actuariel	Taux.ZC
1	2016-05-25	-0.84960937500009	-0.851661033612572
2	2016-07-27	-0.546875000000089	-0.547976350964263
3	2016-09-28	-0.517578125000089	-0.521186557225461
4	2017-01-04	-0.517578125000089	-0.516384081278776
5	2017-03-29	-0.507812500000089	-0.498392711746964
6	2018-05-18	0.0195312499999112	-0.452840626980577
7	2019-05-25	-0.0537109375000889	-0.356355445452927
8	2020-05-25	-0.234375000000089	-1.4363595854146
9	2021-05-25	-0.104980468750089	-0.232164356065501
10	2020-04-25	-1.37695312500009	-0.104197284455254
11	2023-05-25	0.366210937499912	0.162972059192956
12	2024-05-25	0.535888671874912	0.316268520504392
13	2025-05-25	0.537109374999912	0.493730417800764
14	2026-05-25	0.693359374999912	0.656284574630761
15	2031-05-25	1.24511718749991	1.18608895450658
16	2036-05-25	1.52648925781241	1.52601816139648
17	2038-10-25	1.56738281249991	1.73204819800379
18	2045-05-25	1.77001953124991	1.88040343776332
19	2060-05-25	2.11791992187491	2.33317022549759

FIGURE 13 – Calculs de base

L'utilisateur peut également choisir d'afficher un graphique, nous lui proposons deux méthodes de construction du graphique : matricielle ou Boostrap. A noter que la méthode matricielle, proposée par l'institut des actuaires, n'a pas encore été implémentée sous R. Nous allons donc effectuer un Boostraping. Nous devons choisir le nombre d'années, ainsi que les graphiques à afficher parmi le rendement actuariel, les taux zéro coupon, et les discount factors. Le résultat s'affiche dans le troisième onglet, par exemple, ici le taux de rendement actuariel :

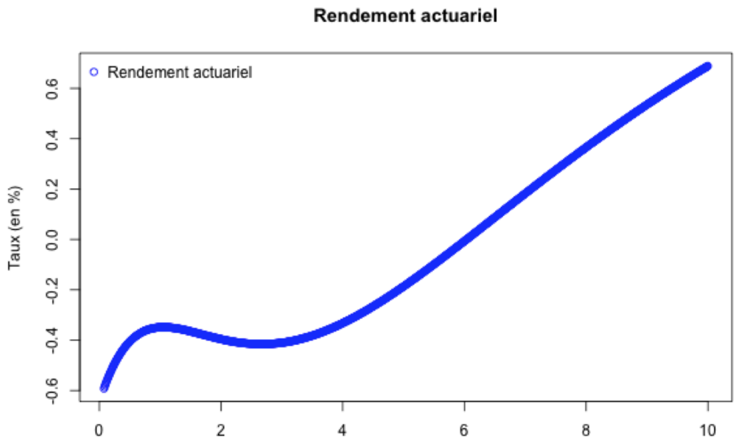
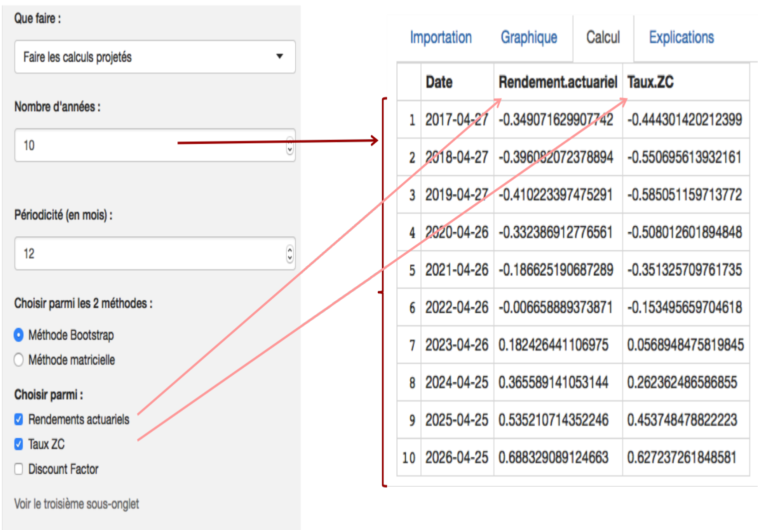


FIGURE 14 – Graphique des rendements actuariels

L'utilisateur peut également obtenir une projection des taux de rendement actuariels, des taux zéro coupon, et des discounts factors. Il lui faut alors indiquer le nombre d'années de projections, et la périodicité choisie en nombre de mois :



	Date	Rendement.actuariel	Taux.ZC
1	2017-04-27	-0.349071629907742	-0.444301420212399
2	2018-04-27	-0.396082072378894	-0.550695613932161
3	2019-04-27	-0.410223397475291	-0.585051159713772
4	2020-04-26	-0.332386912776561	-0.508012601894848
5	2021-04-26	-0.186625190687289	-0.351325709761735
6	2022-04-26	-0.0066588889373871	-0.153495659704618
7	2023-04-26	0.182426441106975	0.0568948475819845
8	2024-04-25	0.365589141053144	0.262362486586855
9	2025-04-25	0.535210714352246	0.453748478822223
10	2026-04-25	0.688329089124663	0.627237261848581

FIGURE 15 – Calculs Projetés

11.2 Pricing de Swap

L'onglet du Pricer Swap, fonctionne de manière assez similaire à celui des obligations. L'utilisateur importe un fichier contenant sur les colonnes maturité et taux swap (conforme aux explications de l'onglet correspondant). Il renseigne la date de calcul et si son importation s'est correctement déroulée, il voit son fichier s'afficher :

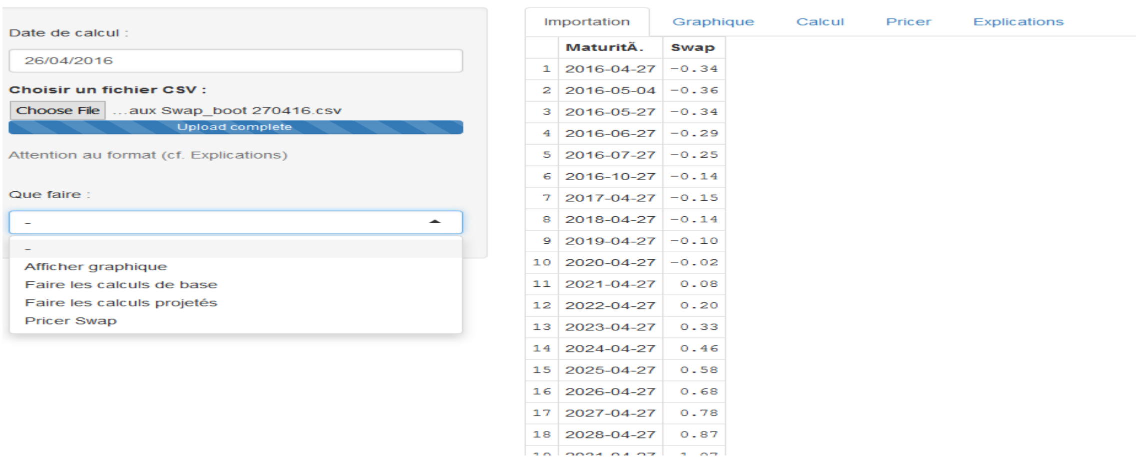


FIGURE 16 – Aperçu du Pricer de Swap

Il peut alors obtenir un calcul des Taux Forwards, des Taux Zéro-coupons, et des Discount Factors, ainsi qu'obtenir les graphiques correspondant à ces calculs. La périodicité à renseigner est celle correspondant à la jambe variable du Swap.

Par exemple ici, un aperçu de la courbe des Taux-Forward avec une périodicité de 3 mois.

Il s'agit ici d'un panier swap d'Avril 2016 de périodicité 3 mois de l'indice du taux variable euribor 3 mois.

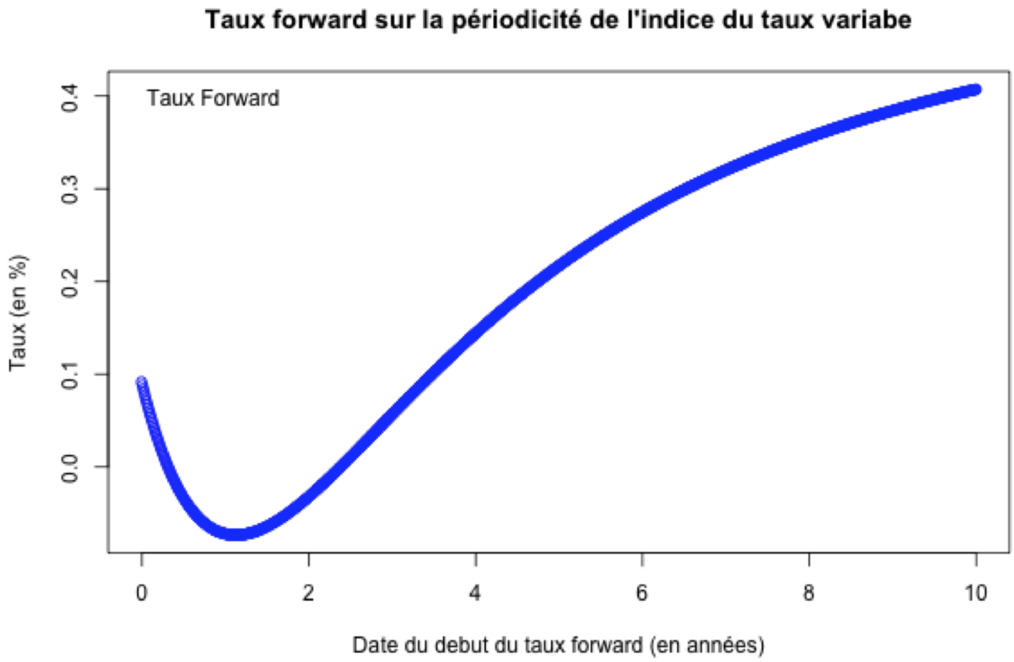


FIGURE 17 – Aperçu du Pricer de Swap

L'utilisateur ne peut cependant pas afficher plusieurs graphiques en même temps. Il peut également pricer un Swap, pour se faire, il doit renseigner la maturité (en année), ainsi que les périodicités (en mois) des deux jambes. Le résultat s'affichera alors dans le quatrième onglet "Pricer".

Conclusion

Notre travail sur les courbes de taux et sur le pricing d'obligations et de swap nous a permis de nous familiariser davantage avec toutes ces notions de mathématiques financières.

De plus, ce bureau d'études a abouti à la création d'une application utilisable par tous à partir de paniers d'obligations et de paniers de taux swap.
Cependant, l'utilisation de l'application est pour le moment limitée à 25 heures actives par mois, il serait donc mieux de poursuivre le développement des applications Shiny sous un hébergeur qui permette une utilisation illimitée.

Il serait intéressant de poursuivre l'étude concernant la méthode utilisée par l'institut des actuaires pour la construction des courbes de taux zéro-coupon. Par ailleurs, nous poursuivrons le développement de l'application durant l'été en ajoutant des fonctionnalités telles qu'un outil de valorisation d'obligations et de renégociation de prêts par exemple.

L'application développée dans le cadre de ce bureau d'études est accessible à l'adresse suivante : https ://euroinstitutdactuariat.shinyapps.io .

Références

Pages Internet

Pour l'utilisation de RShiny :

− https ://www.visualstudio.com/fr-fr/features/application-development-vs.aspx
− http ://forums.cirad.fr/logiciel-R/viewtopic.php?t=7246andstart5
− https ://www.rstudio.com/resources/cheatsheets/
− http ://rstudio.github.io/shinydashboard/getstarted.html
− http ://shiny.rstudio.com/tutorial/lesson1/
− http ://shiny.rstudio.com/tutorial/video

Pour la théorie financière :

− http ://www.thierry-roncalli.com/download/spt.pdf
− http ://www.yats.com/doc/fixed-income-project-fr-ppt.pdf
− http ://www.univ-evry.fr/modules/resources/download/default/m2if/priauley/S2.pdf
− http ://www.therond.fr/wp-content/uploads/cours/Taux.pdf

Articles et mémoires

− PATRICK S.HAGAN, GRAEME WEST Interpolation Methods for Yield Curve Construction. WILMOTT magazine, Mai 2008.

− PATRICK S.HAGAN, GRAEME WEST Methods for constructing a yield curve. Programme in Advanced Mathematics of Finance, School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, South Africa, Juin 2005.

− DAVID GUEZ Impact de la crise de liquidité sur la construction de courbes de taux et la valorisation de swap de taux vanille. Mémoire EURIA, 2014.

− M.BODIN, M.MALAL-BURGUETE, K.PAYEN Construction de la courbe des taux Swap Solvabilité 2. Rapport de bureau d'études EURIA, 2014-2015.

Annexes

Annexe 1 : Le fichier server.R

```
# server.R

#les librairies à charger
library(shiny)
library(zoo)
library(xts)
library(TTR)
library(quantmod)
#source('/Users/anaisng/Desktop/2ème Année EURIA/shiny/helpers.R')
library(png)
library(lubridate) # librarie pour les calculs sur les dates
library(splines)
library(xtable)
library(ggplot2)
library(XML)
library(bitops)
library(RCurl)
library(foreign)


shinyServer(function(input, output){

  #Fonction Midpricing
  #en entrée: coupon Date ask et bid
  Midpricing=function(option){
    mid_Price=c()
    for (i in 1:nrow(option) ){
      if (option[i,3]==0){
        mid_Price[i]=max(option[i,3],option[i,4])
      }
      if (option[i,4]==0){
        mid_Price[i]=max(option[i,3],option[i,4])
      }
      else{
        mid_Price[i]=mean(option[i,3],option[i,4])
      }
    }
    option=option[,c(-3,-4)]
```

```
  op=cbind(option,mid_Price)
  return(op)
}
#en sortie: coupon Date Mid price


#Fonction Dirty price
#en entrée: coupon Date midprice & today
Dirtypricing=function(option,today){
  doublon=as.Date(option[,2],format="%Y-%m-%d")
  today=as.Date(today,"%d/%m/%Y")
  year(doublon)=year(today)
  date_mat_deb=NULL


  for (i in 1:nrow(option)){

    date_mat_deb[i]=(doublon[i]-today)/360
  }
  Dirty_price=option[,3]+option[,1]*abs(date_mat_deb)

  u=as.Date(option[,2])
  u=format(u, format="%Y-%m-%d")

  table=cbind(u,Dirty_price)
  return(table)
}
#en sortie: Date et Dirty price

#Fonction Clean Price réevaluer
#en entrée: coupon maturite et Discount factor & today
Clean_Price_Reevalue=function(z,today){
  n=nrow(z)
  mat=z[,2]
  Clean_Price=NULL

  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z[,3])^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
```

```
    sum((NSS(mat,theta)-z[,3])^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  for (i in 1:n){
    if(mat[i]<=1){
      Clean_Price[i]=(100+z[i,1])*z[i,3]
    }
    else {
      t=floor(as.numeric(mat[i])-0.000001)+1  #au cas ou l"année est égale
      cashflow=rep(as.numeric(z[i,1]),t)
      cashflow[length(cashflow)]=cashflow[length(cashflow)]+100

      matu=c()
      for(j in 0:(t-1)){
        matu=c(as.numeric(mat[i])-j,matu)
      }
      Cp=0
      DFinter=NULL
      for(j in 1:t){
        DFinter=NSS(matu[j],theta$par)
        Cp=Cp+cashflow[j]*DFinter
      }
      Cp=Cp-z[i,1]*NSS(matu[1],theta$par)
      Clean_Price[i]=Cp

    }
  }
  u=mat
  table=cbind(u,Clean_Price)
  return(table)
}
#en sortie: maturite et Clean price reevalue


#Fonction Rendement actuariel
#en entrée: coupon Date et Mid price & today
rendement_actuariel=function(z,today){
  mat=as.numeric(as.Date(z[,2],"%Y-%m-%d")-as.Date(today,"%d/%m/%Y"))/365
  n=nrow(z)
  yield=NULL

  for (i in 1:n){
```

```
    va=NULL
    if(mat[i]<=1){
      f=function(y){
        mid=z[i,3]
        r=(mid-(100+z[i,1])/((1+y)^mat[i]))^2
        return(r)}
      theta=optim(1,f)
      yield[i]=theta$par

    }
    else {
      t=floor(as.numeric(mat[i])-0.000001)+1  #au cas ou l"année est égale
      cashflow=rep(as.numeric(z[i,1]),t)
      cashflow[length(cashflow)]=cashflow[length(cashflow)]+100

      matu=c()
      for(j in 0:(t-1)){
        matu=c(as.numeric(mat[i])-j,matu)
      }

      f=function(y){
        mid=z[i,3]
        c=cashflow/((1+y)^matu)
        r=(mid-sum(c))^2
        return(r)}
      theta=optim(1,f)
      yield[i]=theta$par

    }
  }
  table=cbind(mat,yield)
  return(table)
}
#en sortie: maturite et rendement actuarielle


#Fonction Taux zero coupon obligataire Bootstrap
#en entrée: coupon Date et Mid price & today
ZC=function(obli,today){
  mat=as.numeric(as.Date(obli[,2],"%Y-%m-%d")-as.Date(today,"%d/%m/%Y"))/365

  obli=obli[order(mat),]  #tri croissant des obligations en fonction des maturités
  mat=c(mat[order(mat)])
  obli=cbind(mat,obli)
```

```
n=length(obli[,1])  #Nombre d'obligations
zc=c()
Prix_theorique=c()
DF2=NULL
c=c()
l=0
for(i in 1:n){
  if(obli[i,1]<=0){
    l=l+1
    c=c(c,-i)
  }
}
if (l>0){
  obli=obli[c,]
  mat=mat[c]}

d=length(obli[,1])
for(i in 1:d){
  ob=obli[i,]
  if(ob[1]<=1){
    DF=(as.numeric(ob[4])+ob[2]*abs(ob[1]-1))/(100+as.numeric(ob[2]))
    DF=as.numeric(DF)
    DF2=c(DF2,DF)
    zc[i]=(1/DF)^(1/as.numeric(ob[1]))-1
    Prix_theorique[i]=NA
  }
  if(ob[1]>1){
    t=floor(as.numeric(ob[1])-0.000001)+1  #au cas ou l'année est égale
    cashflow=rep(as.numeric(ob[2]),t)
    cashflow[length(cashflow)]=cashflow[length(cashflow)]+100

    matu=c()
    for(j in 0:(t-1)){
      matu=c(as.numeric(ob[1])-j,matu)
    }

    zcinter=c()
    DF=c()
    for(j in 1:(t-1)){

      k=min(which(mat>=matu[j]))

      if((is.na(zc[k])==FALSE) & (k>1)){
        zcinter=interpol(mat[k-1],mat[k],zc[k-1],zc[k],as.numeric(matu[j]))
```

```
  DF=c(DF,1/(1+zcinter)^matu[j])
}

if(is.na(zc[k])==TRUE){
  mats=mat[1:length(zc)]

  f1=function(theta){
    sum((NS(mats,theta)-zc)^2)
  }

  theta0=c(0,0,0,1)

  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mats,theta)-zc)^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  zcinter=NSS(matu[j],theta\$par)

  DF=c(DF,1/(1+zcinter)^matu[j])
}

if(k==1){
  mats=mat[1:length(zc)]

  f1=function(theta){
    sum((NS(mats,theta)-zc)^2)
  }

  theta0=c(0,0,0,1)

  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mats,theta)-zc)^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  zcinter=NSS(matu[j],theta$par)
```

```
        DF=c(DF,1/(1+zcinter)^matu[j])
      }
    }

    MkP=ob[4]+ob[2]*abs(matu[1]-1)
    DFf=(MkP-sum(cashflow[1:(t-1)]*DF))/(cashflow[t])
    DFf=as.numeric(DFf)
    DF2=c(DF2,DFf)
    zc[i]=((1/DFf)^(1/ob[1])-1)
  }
}
obli=obli[,-1]

tablefinal=cbind(mat,zc,DF2)
return(tablefinal)
}
#en sortie: maturité zero coupon et Discount factor

#Fonction Taux zero coupon obligataire resolution matricielle du bootstrap
#en entrée: coupon Date et Mid price
ZC_matrix=function(z){
  n=nrow(z)
  C=NULL
  for (i in 1:n){
    b=n-i
    a=c(rep(z[i,1],i),rep(0,b))
    a[i]=z[i,1]+100
    C=rbind(C,a)
  }
  DF=solve(C)\%*\%(z[,3])
  zc=((1/DF)^(1/(seq(1,n))))-1

  u=seq(1,n)

  table=cbind(u,zc,DF)
  return(table)
}
#en sortie: maturité zero coupon et Discount factor

#Fonction Taux zero coupon swap Bootstrap
#en entrée: Date et taux swap & today
ZC_swap=function(z,today){
  mat=as.numeric(as.Date(z[,1],"\%Y-\%m-\%d")-as.Date(today,"\%d/\%m/\%Y"))/365
  n=nrow(z)  #Nombre d'obligations
```

```
zc=c()
DF2=NULL
for (i in 1:n){
  if (mat[i]<=1){
    DF=1/(1+z[,2]*mat[i])
    DF=as.numeric(DF)
    DF2=c(DF2,DF)
    zc=c(zc,(1/DF)^(1/mat[i])-1)
  }
  else {
    a=NULL
    t=floor(as.numeric(mat[i])-0.000001)+1  #au cas ou l"année est égale
    matu=c()
    for(j in 0:(t-1)){
      matu=c(as.numeric(mat[i])-j,matu)
    }

    zcinter=c()
    DF=c()
    for(j in 1:(t-1)){

      k=min(which(mat>=matu[j]))

      if((is.na(zc[k])==FALSE) & (k>1)){
        zcinter=interpol(mat[k-1],mat[k],zc[k-1],zc[k],as.numeric(matu[j]))
        DF=c(DF,1/(1+zcinter)^matu[j])
      }

      if(is.na(zc[k])==TRUE){
        mats=mat[1:length(zc)]

        f1=function(theta){
          sum((NS(mats,theta)-zc)^2)
        }

        theta0=c(0,0,0,1)

        theta=optim(theta0,f1)

        f=function(theta){
          sum((NSS(mats,theta)-zc)^2)
        }
        theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
        theta=optim(theta0,f)
```

```
        zcinter=NSS(matu[j],theta$par)

        DF=c(DF,1/(1+zcinter)^matu[j])
      }

      if(k==1){
        mats=mat[1:length(zc)]

        f1=function(theta){
          sum((NS(mats,theta)-zc)^2)
        }

        theta0=c(0,0,0,1)

        theta=optim(theta0,f1)

        f=function(theta){
          sum((NSS(mats,theta)-zc)^2)
        }
        theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
        theta=optim(theta0,f)

        zcinter=NSS(matu[j],theta$par)

        DF=c(DF,1/(1+zcinter)^matu[j])
      }
    }

    a=(1-sum(DF))/(1+z[i,2])
    a=as.numeric(a)
    DF2=c(DF2,a)
    #DF2=cbind(BDF2,a)
    zc=c(zc,(1/DF)^(1/mat[i])-1)
  }
 }
 tablefinal=cbind(mat,zc,DF2)
 return(tablefinal)
}
#en sortie: maturite zero coupon et Discount factor


#Fonction Taux zero coupon swap resolution matricielle du bootstrap
#en entrée: Date et taux swap
```

```
Zc_swap_matrix=function(z){
  n=nrow(z)
  C=NULL
  for (i in 1:n){
    b=n-i
    a=c(rep(z[i,2],i),rep(0,b))
    a[i]=z[i,2]+1
    C=rbind(C,a)
  }
  d=rep(1,n)
  DF=solve(C)%*%(d)
  zc=((1/DF)^(1/(seq(1,n))))-1

  u=seq(1,n)
  table=cbind(u,zc,DF)
  return(table)
}
#en sortie: maturite zero coupon et Discount factor


#Fonction Taux forward (e partir des taux zero coupons de swap)
#en entrée: maturite et zero coupon de swap & today
forward=function(zc,today){
  forward_rate=NULL
  n=nrow(zc)-1
  mat=zc[,1]
  zc[,1]=mat*365+as.Date(today,"%d/%m/%Y")

  for (i in 1:n){
    forward_rate[i]=((1+zc[i+1,2])^(mat[i+1]))/((1+zc[i,2])^(mat[i]))-1
  }
  m=nrow(zc)
  u=zc[-m,1]
  u=format(u, format="%Y-%m-%d")
  v=zc[-1,1]
  v=format(v, format="%Y-%m-%d")
  table=cbind(u,v,forward_rate)
  return(table)
}
#en sortie: Date debut taux Date fin taux et taux forward sur
l'indice des taux swap


#Fonction Taux forward (e partir des taux zero coupons de swap)
```

```
#en entrée: maturite et zero coupon de swap & today
forward2=function(zc,today){
  forward_rate=NULL
  n=nrow(zc)-1
  mat=zc[,1]
  zc[,1]=mat*365+as.Date(today,"%d/%m/%Y")

  for (i in 1:n){
    forward_rate[i]=((1+zc[i+1,2])^(mat[i+1]))/((1+zc[i,2])^(mat[i]))-1
  }

  table=forward_rate
  return(table)
}
#en sortie: Date debut taux Date fin taux et taux forward sur
l'indice des taux swap


#Fonction Discount factor (a partir des taux zero coupons)
#en entrée: Date et zero coupon & today
DF=function(zc,today){
  df=NULL
  n=length(zc[,2])
  mat=as.numeric(as.Date(zc[,1],"%Y-%m-%d")-as.Date(today,"%d/%m/%Y"))/365

  for (i in 1:n){
    df[i]=1/((1+zc[i,2])^(mat[i]))
  }
  table=cbind(mat,df)
  return(table)
}
#en sortie: maturite et Discount factor


#NSS
NSS=function(t,theta){
  rt=theta[1]+theta[2]*(1-exp(-t/theta[5]))/(t/theta[5])
+theta[3]*((1-exp(-t/theta[5]))/(t/theta[5])-exp(-t/theta[5]))
+theta[4]*((1-exp(-t/theta[6]))/(t/theta[6])-exp(-t/theta[6]))
  return(rt)
}


#NS
```

```
NS=function(t,theta){
  rt=theta[1]+theta[2]*(1-exp(-t/theta[4]))/(t/theta[4])
  +theta[3]*((1-exp(-t/theta[4]))/(t/theta[4])-exp(-t/theta[4]))
  return(rt)
}


#Interpolation
#Fonction interpolation linéaire entre deux maturités
interpol=function(t1,t2,r1,r2,t){
  a=(r2-r1)/(t2-t1)
  b=(t2*r1-t1*r2)/(t2-t1)
  return(a*t+b)
}


#Pricer de swap
#en entrée: Date  ZC de swap  Périodicite jambe variable (en année)
Périodicite jambe fixe(en annees)  Maturite & today
Pricing_swap=function(z,Pv,Pf,Matur,today){
  mat=z[,1]
  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z[,2])^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z[,2])^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  n1=(1/Pv)*Matur
  n2=(1/Pf)*Matur

  x1=seq(1:n1)*Pv
  x2=seq(1:n2)*Pf

  zc1=NSS(x1,theta$par)
  df1=1/((1+zc1)^(x1))
  zc2=NSS(x2,theta$par)
  df2=1/((1+zc2)^(x2))
```

```
  x=c(0,x1)
  zc=c(1,zc1)
  forward_rate=NULL
  for (i in 1:n1){
    forward_rate[i]=((1+zc[i+1])^(x[i+1]))/((1+zc[i])^(x[i]))-1
  }

  s1=df1*Pv*forward_rate
  s2=df2*Pf

  s=(sum(s1))/(sum(s2))
  return(s)
}
#en sortie: Taux du swap


#Projection de taux
#en entrée: Date, taux a projeter, maturité, périodicité en année & today
Project=function(z,matur,period,today){
  mat=as.numeric(as.Date(z[,1],"%Y-%m-%d")-as.Date(today,"%d/%m/%Y"))/365

  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z[,2])^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z[,2])^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  x=seq(0,matur,period)

  y=NSS(x,theta$par)

  table=cbind(x,y)

  return(table)
}
#en sortie: temps en année  et  taux projetés
```

```
# ONGLET OBLIGATION
# importation des fichiers obligations --> premier onglet obligation
output$tab_obli <- renderTable({
  if (input$choix==1){
    inFile <- input$fichier
    if (is.null(inFile))
      return(NULL)
    obli=read.csv(inFile$datapath)
  }
  if (input$choix==2){
    inFile <- input$fichier
    if (is.null(inFile))
      return(NULL)
    obli=read.csv(inFile$datapath)
  }
  obli
})

# pour le graphique taux ZC  --> deuxième onglet obligation
output$graph<-renderPlot({
  if (input$choix==1){
    inFile <- input$fichier
    if (is.null(inFile))
      return(NULL)

    z=read.csv(inFile$datapath)
    z=Midpricing(z)
  }
  if (input$choix==2){
    inFile <- input$fichier
    if (is.null(inFile))
      return(NULL)
    z=read.csv(inFile$datapath)
  }

  today=as.Date(input$date,"%d-%m-%Y")

  mat=as.numeric(as.Date(z[,2],"%Y-%m-%d")-as.Date(today,"%d-%m-%Y"))/365

  anneefin=input$anneefin

  x=seq(min(mat),anneefin,length.out=1000)
```

```
c=0

n=length(input$choix_graphique)
if (n>0){
  l1=0
  l2=0
  l3=0

  for (i in 1:n){
    if(input$choix_graphique[i]==1){
      l1=l1+1
    }
    if(input$choix_graphique[i]==2){
      l2=l2+1
    }
    if(input$choix_graphique[i]==3){
      l3=l3+1
    }
  }

  if (l1>0){
    c=c+1

    z2=rendement_actuarielle(z,today)

    #paramétrage de NS puis de NSS en partant du précedent paramétrage
    f1=function(theta){
      sum((NS(mat,theta)-z2[,2])^2)
    }
    theta0=c(0,0,0,1)
    theta=optim(theta0,f1)

    f=function(theta){
      sum((NSS(mat,theta)-z2[,2])^2)
    }
    theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
    theta=optim(theta0,f)

    z3=NSS(x,theta$par)*100

    plot(x,z3,col="blue",xlab='Maturité (en années)',ylab='Taux (en %)',
    main='Rendement actuariel')
    legend("topleft",legend="Rendement actuariel",col="blue",pch=1,bty="n")
```

```
}

if (l2>0){
  c=c+1

  if (input$methode==1){
    z2=ZC(z,today)[,-3]
  }
  if (input$methode==2){
    z2=ZC_matrix(z)[,-3]
  }

  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z2[,2])^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z2[,2])^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)


  z3=NSS(x,theta$par)*100

  if (c>1){
    lines(x,z3,col="red",xlab='Maturité (en années)',ylab='Taux (en %)'
    ,main='Taux Zéro Coupon',type = 'l')
    legend("topleft",legend=c("Rendement actuariel","Taux Zéro Coupon"),
    col=c("blue","red"),pch=c(1,18),bty="n")
  }
  else {
    plot(x,z3,col="red",xlab='Maturité (en années)',ylab='Taux (en %)',
    main='Taux zero coupon',type = 'l')
    legend("topleft",legend="Taux Zéro Coupon",col="red",pch=18,bty="n")
  }
}

if (l3>0){
  c=c+1
```

```
      if (input$methode==1){
        z2=ZC(z,today)[,-2]
      }
      if (input$methode==2){
        z2=ZC_matrix(z)[,-2]
      }


      #paramétrage de NS puis de NSS en partant du précedent paramétrage
      f1=function(theta){
        sum((NS(mat,theta)-z2[,2])^2)
      }
      theta0=c(0,0,0,1)
      theta=optim(theta0,f1)

      f=function(theta){
        sum((NSS(mat,theta)-z2[,2])^2)
      }
      theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
      theta=optim(theta0,f)

      z3=NSS(x,theta$par)

      if (c>1){
        lines(x,z3,col="green",xlab='Maturité (en années)',main='Discount Factor',
        type = 'l')
        legend("topleft",legend=c("Rendement actuariel","Discount Factor"),
        col=c("blue","green"),pch=c(1,18),bty="n")
      }
      else {
        plot(x,z3,col="green",xlab='Maturité (en années)',ylab='Discount Factor',
        type = 'l')
        legend("topleft",legend="Discount Factor",col="green",pch=18,bty="n")
      }
    }

  }

  else{
    return(NULL)
  }

})
```

```
# calculs sur les fichiers obligations importés  --> troisième onglet obligation
output$calcul_obli <- renderTable({
  if (input$choix==1){
    inFile <- input$fichier
    if (is.null(inFile))
      return(NULL)
    z=read.csv(inFile$datapath)
    z=Midpricing(z)
  } # fichier type 1
  if (input$choix==2){
    inFile <- input$fichier
    if (is.null(inFile))
      return(NULL)
    z=read.csv(inFile$datapath)
  } # fichier type 2

  today=as.Date(input$date,"%d-%m-%Y")
  mat=as.numeric(as.Date(z[,2],"%Y-%m-%d")-as.Date(today,"%d-%m-%Y"))/365
  C=NULL

  if (input$DO==3){
    C=NULL
    n=length(input$choix_tableau)
    if (n>0){
      l1=0
      l2=0
      l3=0
      l4=0
      l5=0
      for (i in 1:n){
        if(input$choix_tableau[i]==1){
          l1=l1+1
        }
        if(input$choix_tableau[i]==2){
          l2=l2+1
        }
        if(input$choix_tableau[i]==3){
          l3=l3+1
        }
        if(input$choix_tableau[i]==4){
          l4=l4+1
        }
        if(input$choix_tableau[i]==5){
          l5=l5+1
```

```
    }
  }

  if (l1>0){
    a=rendement_actuarielle(z,today)[,-1]*100
    rendement.actuariel=a
    C=cbind(C,rendement.actuariel)
  }

  if (l2>0){
    if (input$methode2==1){
      a=ZC(z,today)[,c(-1,-3)]*100
      Taux.ZC=a
      C=cbind(C,Taux.ZC)
    }
    if (input$methode2==2){
      a=ZC_matrix(z)[,c(-1,-3)]*100
      Taux.ZC=a
      C=cbind(C,Taux.ZC)
    }
  }

  if (l3>0){
    if (input$methode2==1){
      a=ZC(z,today)[,c(-1,-2)]
      Discount.Factor=a
      C=cbind(C,Discount.Factor)
    }
    if (input$methode2==2){
      a=ZC_matrix(z)[,c(-1,-2)]
      Discount.Factor=a
      C=cbind(C,Discount.Factor)
    }
  }

  if (l4>0){
    a=Dirtypricing(z,today)[,-1]
    Dirty.Price=a
    C=cbind(C,Dirty.Price)
  }

  if (l5>0){
    if (input$methode2==1){
      z2=cbind(z[,1],ZC(z,today)[,-2])
```

```
      }
      if (input$methode2==2){
        z2=cbind(z[,1],ZC_matrix(z)[,-2])
      }
      a=Clean_Price_Reevalue(z2,today)[,-1]
      Clean.Price=a
      C=cbind(C,Clean.Price)
    }
  }
  else{
    return(NULL)
  }

  u=as.Date(z[,2])
  u=format(u, format="%Y-%m-%d")
  Date=u
  C=cbind(Date,C)
  return(C)
} # calculs de base
if (input$DO==4){
  C=NULL
  an=input$annee
  per=input$per/12
  if (an==0 || per==0){
    return(NULL)
  }
  x=seq(per,an,by=per)

  n=length(input$choix_tableau2)
  if (n>0){
    l1=0
    l2=0
    l3=0
    l4=0
    l5=0
    for (i in 1:n){
      if(input$choix_tableau2[i]==1){
        l1=l1+1
      }
      if(input$choix_tableau2[i]==2){
        l2=l2+1
      }
      if(input$choix_tableau2[i]==3){
        l3=l3+1
```

```
  }
}

if (l1>0){
  z2=rendement_actuarielle(z,today)[,-1]
  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z2)^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z2)^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  z3=NSS(x,theta$par)*100
  a=z3
  Rendement.actuariel=a
  C=cbind(C,Rendement.actuariel)
}

if (l2>0){
  if (input$methode3==1){
    z2=ZC(z,today)[,c(-1,-3)]
  }
  if (input$methode3==2){
    z2=ZC_matrix(z)[,c(-1,-3)]
  }
  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z2)^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z2)^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)
```

```
        z3=NSS(x,theta$par)*100
        a=z3
        Taux.ZC=a
        C=cbind(C,Taux.ZC)
      }

      if (l3>0){
        if (input$methode3==1){
          z2=ZC(z,today)[,c(-1,-2)]
        }
        if (input$methode3==2){
          z2=ZC_matrix(z)[,c(-1,-2)]
        }
        #paramétrage de NS puis de NSS en partant du précedent paramétrage
        f1=function(theta){
          sum((NS(mat,theta)-z2)^2)
        }
        theta0=c(0,0,0,1)
        theta=optim(theta0,f1)

        f=function(theta){
          sum((NSS(mat,theta)-z2)^2)
        }
        theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
        theta=optim(theta0,f)

        z3=NSS(x,theta$par)
        a=z3
        Discount.Factor=a
        C=cbind(C,Discount.Factor)
      }
    }
    else{
      return(NULL)
    }
    u=x*365+as.Date(today)
    u=format(u, format="%Y-%m-%d")
    Date=u
    C=cbind(Date,C)
    return(C)
  } # calculs projetés
})
```

```
# ONGLET SWAP
# importation des fichiers swaps --> premier onglet swap
output$tab_swap <- renderTable({
  inFile <- input$fichier_swap
  if (is.null(inFile))
    return(NULL)
  obli=read.csv(inFile$datapath)
  obli
})

# pour le graphique de swap  --> deuxième onglet swap
output$grapheswap<-renderPlot({
  inFile <- input$fichier_swap
  if (is.null(inFile))
    return(NULL)
  z=read.csv(inFile$datapath)

  today=as.Date(input$date_swap,"%d-%m-%Y")

  mat=as.numeric(as.Date(z[,1],"%Y-%m-%d")-as.Date(today,"%d-%m-%Y"))/365

  anneefin=input$anneefinswap

  x=seq(min(mat),anneefin,length.out=1000)

  c=0

  n=length(input$choix_graphique_swap)
  if (n>0){
    l1=0
    l2=0
    l3=0

    for (i in 1:n){
      if(input$choix_graphique_swap[i]==1){
        l1=l1+1
      }
      if(input$choix_graphique_swap[i]==2){
        l2=l2+1
      }
      if(input$choix_graphique_swap[i]==3){
        l3=l3+1
      }
    }
```

```
if (l1>0){
  c=c+1

  period=input$period/12
  x2=seq(period,anneefin+period,by=period)

  if (input$methode_swap==1){
    z2=ZC_swap(z,today)[,-3]
  }
  if (input$methode_swap==2){
    z2=Zc_swap_matrix(z)[,-3]
  }

  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z2[,2])^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z2[,2])^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  z3=NSS(x2,theta$par)
  z3=cbind(x2,z3)

  z4=forward2(z3,today)

  x3=x2[-length(x2)]

  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(x3,theta)-z4)^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(x3,theta)-z4)^2)
  }
```

```
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  z5=NSS(x,theta$par)*100

  plot(x,z5,col="blue",xlab='Date du debut du taux forward (en années)',
  ylab='Taux (en %)',main="Taux forward sur la périodicité de l'indice du taux
  legend("topleft",legend="Taux Forward",col="blue",pch1,bty="n")
}

if (l2>0){
  c=c+1

  if (input$methode_swap==1){
    z2=ZC_swap(z,today)[,-3]
  }
  if (input$methode_swap==2){
    z2=Zc_swap_matrix(z)[,-3]
  }

  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z2[,2])^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z2[,2])^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  z3=NSS(x,theta$par)*100

  if (c>1){
    lines(x,z3,col="red",xlab='Maturité (en années)',ylab='Taux (en %)',
    main='Taux Zéro Coupon',type = 'l')
    legend("topleft",legend=c("Taux Forward","Taux Zéro Coupon"),col=c("blue",
    "red"),pch=c(18,1),bty="n")
  }
  else {
    plot(x,z3,col="red",xlab='Maturité (en années)',ylab='Taux (en %)',
    main='Taux Zéro Coupon',type = 'l')
```

```
      legend("topleft",legend="Taux Zéro Coupon",col="red",pch=1,bty="n")
    }
  }

  if (l3>0){
    c=c+1

    if (input$methode_swap==1){
      z2=ZC_swap(z,today)[,-2]
    }
    if (input$methode_swap==2){
      z2=Zc_swap_matrix(z)[,-2]
    }


    #paramétrage de NS puis de NSS en partant du précedent paramétrage
    f1=function(theta){
      sum((NS(mat,theta)-z2[,2])^2)
    }
    theta0=c(0,0,0,1)
    theta=optim(theta0,f1)

    f=function(theta){
      sum((NSS(mat,theta)-z2[,2])^2)
    }
    theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
    theta=optim(theta0,f)

    z3=NSS(x,theta$par)

    if (c>1){
      lines(x,z3,col="green",xlab='Maturité (en années)',main='Discount Factor',
      type = 'l')
      legend("topleft",legend=c("Taux Forward","Discount Factor"),col=c("blue",
      "green"),pch=c(18,18),bty="n")
    }
    else {
      plot(x,z3,col="green",xlab='Maturité (en années)',ylab='Discount factor',
      type = 'l')
      legend("topleft",legend="Discount Factor",col="green",pch=18,bty="n")
    }
  }

}
```

```
  else{
    return(NULL)
  }
})


# calculs sur les fichiers swap importés  --> troisième onglet swap
output$table_swap <- renderTable({
  inFile <- input$fichier_swap
  if (is.null(inFile))
    return(NULL)

  z=read.csv(inFile$datapath)
  today=as.Date(input$date_swap,"%d-%m-%Y")
  mat=as.numeric(as.Date(z[,1],"%Y-%m-%d")-as.Date(today,"%d-%m-%Y"))/365
  C=NULL

  if (input$DO_swap==3){
    C=NULL
    n=length(input$choix_tableau)
    if (n>0){
      l1=0
      l2=0
      l3=0
      for (i in 1:n){
        if(input$choix_tableau_swap[i]==1){
          l1=l1+1
        }
        if(input$choix_tableau_swap[i]==2){
          l2=l2+1
        }
        if(input$choix_tableau_swap[i]==3){
          l3=l3+1
        }

      }
      if (l1>0){
        if (input$methode_swap2==1){
          z2=ZC_swap(z,today)[,-3]
        }
        if (input$methode_swap2==2){
          z2=Zc_swap_matrix(z,today)[,-3]
        }
```

```
    z3=forward(z2,today)
    a=c("Date de début","Date de fin","Taux Forward")
    b=rbind(a,z3)
    C=cbind(C,b)
  }

  if (l2>0){
    if (input$methode_swap2==1){
      a=ZC_swap(z,today)[,c(-1,-3)]*100
      Taux.ZC=a
      C=cbind(C,Taux.ZC)
    }
    if (input$methode_swap2==2){
      a=Zc_swap_matrix(z)[,c(-1,-3)]*100
      Taux.ZC=a
      C=cbind(C,Taux.ZC)
    }
  }

  if (l3>0){
    if (input$methode_swap2==1){
      a=ZC_swap(z,today)[,c(-1,-2)]
      Discount.Factor=a
      C=cbind(C,Discount.Factor)
    }
    if (input$methode_swap2==2){
      a=Zc_swap_matrix(z)[,c(-1,-2)]
      Discount.Factor=a
      C=cbind(C,Discount.Factor)
    }
  }

}

else{
  return(NULL)
}

u=as.Date(z[,1])
u=format(u, format="%Y-%m-%d")

C=cbind(u,C)
return(C)
```

```
} # calculs de base
if (input$DO_swap==4){
  C=NULL
  n=length(input$choix_tableau2)
  an=input$annee_swap
  per=input$per_swap/12
  if (an==0 || per==0){
    return(NULL)
  }
  x=seq(per,an,by=per)
  if (n>0){
    l1=0
    l2=0
    l3=0
    for (i in 1:n){
      if(input$choix_tableau_swap2[i]==1){
        l1=l1+1
      }
      if(input$choix_tableau_swap2[i]==2){
        l2=l2+1
      }
      if(input$choix_tableau_swap2[i]==3){
        l3=l3+1
      }

    }
    if (l1>0){
      if (input$methode_swap3==1){
        z2=ZC_swap(z,today)[,-3]
      }
      if (input$methode_swap3==2){
        z2=Zc_swap_matrix(z)[,-3]
      }
      mat=z2[,1]
      #paramétrage de NS puis de NSS en partant du précedent paramétrage
      f1=function(theta){
        sum((NS(mat,theta)-z2[,2])^2)
      }
      theta0=c(0,0,0,1)
      theta=optim(theta0,f1)

      f=function(theta){
        sum((NSS(mat,theta)-z2[,2])^2)
      }
```

```
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  z3=NSS(x,theta$par)
  z3=cbind(x,z3)

  z4=forward(z3,today)
  a=c("Date de début","Date de fin","Taux Forward")
  b=rbind(a,z4)
  C=cbind(C,b)
}

if (l2>0){
  if (input$methode_swap3==1){
    z2=ZC_swap(z,today)[,c(-1,-3)]
  }
  if (input$methode_swap3==2){
    z2=Zc_swap_matrix(z)[,c(-1,-3)]
  }
  #paramétrage de NS puis de NSS en partant du précedent paramétrage
  f1=function(theta){
    sum((NS(mat,theta)-z2)^2)
  }
  theta0=c(0,0,0,1)
  theta=optim(theta0,f1)

  f=function(theta){
    sum((NSS(mat,theta)-z2)^2)
  }
  theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
  theta=optim(theta0,f)

  z3=NSS(x,theta$par)*100
  a=z3
  Taux.ZC=a
  C=cbind(C,Taux.ZC)

}

if (l3>0){
  if (input$methode_swap3==1){
    z2=ZC_swap(z,today)[,c(-1,-2)]
  }
  if (input$methode_swap3==2){
```

```
        z2=Zc_swap_matrix(z)[,c(-1,-2)]
      }
      #paramétrage de NS puis de NSS en partant du précedent paramétrage
      f1=function(theta){
        sum((NS(mat,theta)-z2)^2)
      }
      theta0=c(0,0,0,1)
      theta=optim(theta0,f1)

      f=function(theta){
        sum((NSS(mat,theta)-z2)^2)
      }
      theta0=c(theta$par[1],theta$par[2],theta$par[3],0.01,theta$par[4],1)
      theta=optim(theta0,f)

      z3=NSS(x,theta$par)*100
      a=z3
      Discount.Factor=a
      C=cbind(C,Discount.Factor)
    }
  }
  else{
    return(NULL)
  }

  u=x*365+as.Date(today)
  u=format(u, format="%Y-%m-%d")

  C=cbind(u,C)
  return(C)
 } # calculs projetés
})


# Pricing d'un swap --> quatrième onglet swap
output$priceswap <- renderTable({
  inFile <- input$fichier_swap
  if (is.null(inFile))
    return(NULL)
  z=read.csv(inFile$datapath)
  today=as.Date(input$date_swap,"%d-%m-%Y")
  Matur=input$mat_swap
  Pv=input$p_jambe_var
  Pf=input$p_jambe_fixe
```

```
    z2=ZC_swap(z,today)[,-3]
    y=Pricing_swap(z2,Pv,Pf,Matur,today)

    return(y)
  })
})
```

Annexe 2 : Le fichier ui.R

```
  # ui.R

library(shiny)

shinyUI(
  fluidPage(
    # pour créer des onglets
    navbarPage("Application EURIA",
               # PREMIER ONGLET
               tabPanel("Explication",
                        sidebarLayout(
                          sidebarPanel(
 h1("Bienvenue"),
 br(),
 br(),
 br(),
br(),

h5("Application de l'EURIA")
                          ),
mainPanel(  tabsetPanel(type="tabs",
                            tabPanel("EURIA",
                                                  h1("Euro-Institut d'Actuariat"),
                                                  br(),
 h5("Ecole d'actuariat reconnue par l'Institut des Actuaires"),
                                                  br(),
                                                  br(),
                                                  br()),
tabPanel("Bureau d'Etude",
h4("Etudiants en actuariat, nous avons effectué un bureau d'étude en lien avec
Britsk - cabinet de conseil en risque financier - et l'EURIA"),
    br(),
    br(),
    br(),
    h5("Tuteurs: Fabrice Hamon, Franck Vermet"),
    h5("AnaÎs Ng Liet Hing, Manon Pieren, Patrick Du Chouchet, Rémi Gauville"),
    h5("Pour nous contacter :"),
    h5("- anais_ng93@hotmail.com"),
    h5("- pieren.manon@orange.fr"),
    h5("- pduchouchet@gmail.com"),
    h5("- remigogo@hotmail.fr"))
                            )
```

```
                        )
                      )
              ),
# ONGLET PRICER
tabPanel("Obligation Pricer",
    sidebarLayout(
     sidebarPanel(
     dateInput("date",label=h5("Date de calcul :"),value=Sys.Date(),
     format="dd/mm/yyyy"),
helpText("Attention à la date de maturité (cf. Explications)"),

selectInput("choix",label=h5("Choisir parmi :"),choices=list("Importer fichier type
1"=1,"Importer fichier type 2"=2),selected = 1),

fileInput('fichier','Choisir un fichier CSV :',accept=c('text/csv','text
/comma-separated-values,text/plain','.csv')),

helpText("Attention au format (cf. Explications)"),

selectInput("DO",label=h5("Que faire :"),choices=list(" - "=1,"Afficher graphique"
=2,"Faire les calculs de base"=3, "Faire les calculs projetés"=4),selected = 1),

conditionalPanel(
    condition="input.DO==2",

radioButtons("methode",label=h5("Choisir parmi les 2 méthodes :"),choices=
list("Méthode Bootstrap"=1,"Méthode matricielle"=2),selected=1),

numericInput("anneefin", label=h5("Nombre d'années :"),value=10,min=0),

checkboxGroupInput("choix_graphique","Choisir parmi les graphes :",choices=
c("Rendements actuariels"=1,"Taux ZC"=2,"DF"=3),selected=1),

    helpText("Voir le deuxième sous-onglet")
                           ),

    conditionalPanel(
    condition="input.DO==3",

radioButtons("methode2",label=h5("Choisir parmi les 2 méthodes :"),
choices=list("Méthode Bootstrap"=1,"Méthode matricielle"=2),selected=1),

checkboxGroupInput("choix_tableau","Choisir parmi :",choices=
c("Rendements actuariels"=1,"Taux ZC"=2,"Discount Factor"=3,"Dirty Price"=4,
```

```
"Clean Price"=5),selected=1),

helpText("Voir le troisième sous-onglet")
                            ),
conditionalPanel(
condition="input.DO==4",

numericInput("annee", label=h5("Nombre d'années :"),value=0,min=0),
                              br(),

numericInput("per", label=h5("Périodicité (en mois) :"),value=0,min=0),

radioButtons("methode3",label=h5("Choisir parmi les 2 méthodes :"),choices=
list("Méthode Bootstrap"=1,"Méthode matricielle"=2),selected=1),

checkboxGroupInput("choix_tableau2","Choisir parmi :",choices=
c("Rendements actuariels"=1,"Taux ZC"=2,"Discount Factor"=3),selected=1),

helpText("Voir le troisième sous-onglet")
                            )
                          ),
                          mainPanel(

tabsetPanel(type="tabs",

tabPanel("Importation",tableOutput('tab_obli')),

tabPanel("Graphique",plotOutput("graph"),plotOutput("graphes")),

tabPanel("Calcul",tableOutput('calcul_obli')),

tabPanel("Explications",
                                                  br(),

  p("La construction de la courbe de taux se fait via l'importation d'un
  panier d'obligations."),
                                                  br(),
strong(h3("Forme du fichier à importer:")),
                                                  br(),
 p("Il faut obligatoirement que les colonnes aient des titres
 spécifiés entre guillemets."),

    p("Les dates doivent Ãªtre au format AAAA/MM/JJ."),
```

```
     strong( h4("Fichier de type 1:")),

    p("Ce format est à utiliser dans le cas où vos données de prix sont
    sous la forme bid/ask."),

         p("En colonnes: coupon, maturité, bid, ask")


         strong( h4("Fichier de type 2:")),


   p("Ce format est à utiliser dans le cas où vos données de prix sont
   sous la forme midprice."),

     p("En colonnes: coupon, maturité, prix"),
                                 br(),

  em("La date spécifiée dans le calendrier doit être antèrieure aux maturités
  des obligations du panier."),

 em("C'est la date à laquelle sera calculée la courbe de taux ."))
                           )
                         )
                       )
              ),
 # ONGLET SWAP PRICER
              tabPanel("Swap Pricer",
                       sidebarLayout(
                         sidebarPanel(

dateInput("date_swap",label=h5("Date de calcul :"),value=Sys.Date(),
format="dd/mm/yyyy"),

fileInput('fichier_swap','Choisir un fichier CSV :',accept=c('text/csv','text
/comma-separated-values,text/plain','.csv')),

helpText("Attention au format (cf. Explications)"),
                            br(),

selectInput("DO_swap",label=h5("Que faire :"),

choices=
list(" - "=1,
```

```
"Afficher graphique"=2,"Faire les calculs de base"=3,

"Faire les calculs projetés"=4,"Pricer Swap"=5),selected = 1),

conditionalPanel(

condition="input.DO_swap==2",

radioButtons("methode_swap",label=h5("Choisir parmi les 2 méthodes :"),

choices=list("Méthode Bootstrap"=1,"Méthode matricielle"=2),selected=1),

numericInput("anneefinswap", label=h5("Nombre d'années :"),

value=10,min=0),

numericInput("period", label=h5("Périodicite de l'indice du taux variabe

(en mois) :"),value=0,min=0),

checkboxGroupInput("choix_graphique_swap","Choisir parmi :",

choices=c("Taux Forward"=1,"Taux ZC"=2,"Discount Factor"=3),selected=1),

helpText("Voir le deuxième sous-onglet")
                            ),

conditionalPanel(
condition="input.DO_swap==3",

radioButtons("methode_swap2",label=h5("Choisir parmi les 2 méthodes :"),

choices=list("Méthode Bootstrap"=1,"Méthode matricielle"=2),selected=1),

checkboxGroupInput("choix_tableau_swap","Choisir parmi :",

choices=c("Taux Forward"=1,"Taux ZC"=2,"Discount Factor"=3),selected=1),

helpText("Voir le troisième sous-onglet")
                            ),
conditionalPanel(

condition="input.DO_swap==4",
```

```
numericInput("annee_swap", label=h5("Nombre d'années (en années) :"),

value=0,min=0),

br(),

numericInput("per_swap", label=h5("Périodicité (en mois) "),value=0,min=0),

radioButtons("methode_swap3",label=h5("Choisir parmi les 2 méthodes
:"),

choices=list("Méthode Bootstrap"=1,"Méthode matricielle"=2),selected=1),

checkboxGroupInput("choix_tableau_swap2","Choisir parmi :",

choices=c("Taux Forward"=1,"Taux ZC"=2,"Discount Factor"=3),selected=1),

helpText("Voir le troisième sous-onglet")
                            ),

conditionalPanel(

    condition="input.DO_swap==5",

numericInput("mat_swap", label=h5("Maturité (en années) :"),value=0,min=0),

numericInput("p_jambe_fixe", label=h5("Périodicité jambe fixe (en mois)

:"),value=0,min=0),

numericInput("p_jambe_var", label=h5("Périodicité jambe variable (en mois)

:"),value=0,min=0),

helpText("Voir le quatriième sous-onglet")
                            )
                          ),

 mainPanel(
            tabsetPanel(type="tabs",

tabPanel("Importation",tableOutput('tab_swap')),

tabPanel("Graphique",plotOutput('grapheswap')),
```

```
        tabPanel("Calcul",tableOutput("table_swap")),

        tabPanel("Pricer",tableOutput("priceswap")),

        tabPanel("Explications",
 br(),

strong(h3("Forme du fichier à importer :")),

br(),

p("Il faut obligatoirement que les colonnes aient des titres spécifiés

entre guillemets."),

p("Les dates doivent être au format AAAA-MM-DD."),

strong( h4("Fichier adapté à la méthode Bootstrap et à la méthode matricielle

:")),

p("En colonnes:  maturité, taux swap"),

br())
                            )
                          )
                        )
                )
    )
  )
)
```

Annexe 3 : Exemple de panier d'obligations

Coupons	**Maturités**	**Prix**
0.00	2016-05-18	100.0350
0.00	2017-07-13	100.1260
0.00	2016-12-10	100.2640
0.00	2017-04-26	100.5275
0.00	2018-03-16	100.9375
1.00	2019-02-22	104.2225
0.00	2020-04-27	102.6275
0.00	2023-04-27	101.4600
2.00	2024-04-27	112.8125
1.50	2025-04-27	111.0525
1.75	2026-04-27	113.6025
0.50	2027-04-27	103.0000
0.50	2028-04-27	101.9675
6.25	2029-04-27	175.9950
4.75	2030-04-27	167.4250
4.75	2031-04-27	181.8950
2.50	2032-04-27	137.6225

Annexe 4 : Exemple de panier de taux swap

Maturités	**Taux Swap**
2016-04-27	-0.342
2016-05-04	-0.359
2016-05-27	-0.343
2016-06-27	-0.287
2016-07-27	-0.251
2016-10-27	-0.142
2017-04-27	-0.15
2018-04-27	-0.144
2019-04-27	-0.103
2020-04-27	-0.022
2021-04-27	0.082
2022-04-27	0.201
2023-04-27	0.326
2024-04-27	0.455
2025-04-27	0.576
2026-04-27	0.685
2027-04-27	0.785
2028-04-27	0.872
2031-04-27	1.068
2036-04-27	1.219
2041-04-27	1.262
2046-04-27	1.271
2051-04-27	1.259
2056-04-27	1.241
2061-04-27	1.199
2066-04-27	1.174