

Destinie2

Généré par Doxygen 1.8.12

3 mai 2019

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Vue d'ensemble</b>	<b>3</b>
2.1	Le package Destinie . . . . .	3
2.2	Options et paramètres de simulation . . . . .	6
2.2.1	Hypothèses démographiques . . . . .	6
2.2.2	Hypothèses économiques . . . . .	6
2.2.3	Hypothèses règlementaires et autres . . . . .	6
2.3	Choix de la législation . . . . .	6
2.3.1	Réforme de 1993 . . . . .	6
2.3.2	Réforme de 2003 . . . . .	7
2.3.3	Réforme de 2010 . . . . .	7
2.3.4	Accélération du calendrier fixé en 2010 suite au projet de loi de financement de la sécurité sociale pour 2012 . . . . .	8
2.3.5	Décret de juillet 2012 : extension du dispositif carrières longues . . . . .	8
2.3.6	Réforme de 2013 . . . . .	9
2.3.7	Accord national interprofessionnel relatif aux retraites complémentaires Agirc-Arrco du 30 octobre 2015 . . . . .	9
<b>3</b>	<b>Index des classes</b>	<b>9</b>
3.1	Liste des classes . . . . .	9
<b>4</b>	<b>Index des fichiers</b>	<b>12</b>
4.1	Liste des fichiers . . . . .	12

<b>5</b>	<b>Documentation des classes</b>	<b>13</b>
5.1	Référence de la structure CiblesDemo . . . . .	13
5.1.1	Description détaillée . . . . .	15
5.2	Référence de la structure CiblesTrans . . . . .	15
5.3	Référence de la structure Cotisations . . . . .	18
5.3.1	Description détaillée . . . . .	19
5.4	Référence de la structure CotisationsTRI . . . . .	19
5.5	Référence de la structure CStructSexeAge . . . . .	20
5.5.1	Description détaillée . . . . .	20
5.6	Référence de la classe DroitsRetr . . . . .	20
5.6.1	Description détaillée . . . . .	25
5.6.2	Documentation des fonctions membres . . . . .	26
5.7	Référence de la structure Ech . . . . .	30
5.8	Référence de la structure Emp . . . . .	31
5.9	Référence de la structure EqSalaires . . . . .	31
5.9.1	Description détaillée . . . . .	32
5.10	Référence de la structure EqSante . . . . .	32
5.10.1	Description détaillée . . . . .	33
5.11	Référence de la structure EqTrans . . . . .	33
5.12	Référence de la structure Fam . . . . .	35
5.13	Référence de la structure FinEtudeMoy . . . . .	36
5.13.1	Description détaillée . . . . .	36
5.14	Référence de la structure Indic_annee . . . . .	36
5.15	Référence de la structure Indic_gen . . . . .	38
5.16	Référence de la structure Indic_gen_age . . . . .	39
5.17	Référence de la structure Indicateur . . . . .	40
5.18	Référence de la classe Indiv . . . . .	40
5.18.1	Description détaillée . . . . .	42
5.18.2	Documentation des constructeurs et destructeur . . . . .	42
5.18.3	Documentation des fonctions membres . . . . .	43

5.19	Référence de la classe Leg	44
5.19.1	Description détaillée	45
5.19.2	Documentation des constructeurs et destructeur	46
5.20	Référence de la structure Macro	46
5.20.1	Description détaillée	56
5.20.2	Documentation des données membres	56
5.21	Référence de la structure Mortadiff_dip_F	56
5.21.1	Description détaillée	57
5.22	Référence de la structure Mortadiff_dip_H	57
5.22.1	Description détaillée	57
5.23	Référence de la structure Mortalite_diff	58
5.23.1	Description détaillée	58
5.24	Référence de la structure Moyenne	58
5.25	Référence de la structure Opt	59
5.26	Référence de la structure Option_optim	59
5.27	Référence de la structure Options	59
5.27.1	Description détaillée	61
5.28	Référence de la structure Options_salaires	61
5.29	Référence de la structure OptionsTRI	62
5.30	Référence de la structure ordre_findet	62
5.31	Référence de la structure Paire	62
5.32	Référence de la structure quantile	63
5.33	Référence de la structure Ratio	63
5.34	Référence de la classe Retraite	63
5.34.1	Description détaillée	65
5.34.2	Documentation des fonctions membres	66
5.35	Référence de la structure Retraite_comp	69
5.36	Référence de la classe Reversion	70
5.36.1	Documentation des constructeurs et destructeur	70
5.37	Référence de la structure Salaire	71
5.37.1	Description détaillée	71
5.37.2	Documentation des fonctions membres	72
5.38	Référence de la structure Simulation	72
5.39	Référence de la structure Somme	73
5.40	Référence de la structure Taux	73
5.41	Référence de la structure Transitions	74
5.41.1	Description détaillée	75
5.41.2	Documentation des fonctions membres	75
5.41.3	Documentation des données membres	75

<b>6</b>	<b>Documentation des fichiers</b>	<b>76</b>
6.1	Référence du fichier src/Constantes.h . . . . .	76
6.1.1	Documentation du type de l'énumération . . . . .	77
6.1.2	Documentation des variables . . . . .	81
6.2	Référence du fichier src/Cotisations.h . . . . .	83
6.2.1	Documentation des fonctions . . . . .	84
6.3	Référence du fichier src/CotisationsTRI.h . . . . .	86
6.3.1	Description détaillée . . . . .	86
6.4	Référence du fichier src/Destinie.h . . . . .	86
6.4.1	Description détaillée . . . . .	86
6.4.2	Documentation des fonctions . . . . .	87
6.5	Référence du fichier src/DroitsRetr.h . . . . .	87
6.6	Référence du fichier src/Indicateurs_annee_COR.h . . . . .	88
6.7	Référence du fichier src/Indicateurs_gen.h . . . . .	88
6.8	Référence du fichier src/Indicateurs_gen_age.h . . . . .	89
6.8.1	Documentation des fonctions . . . . .	89
6.9	Référence du fichier src/Indiv.h . . . . .	89
6.9.1	Documentation des fonctions . . . . .	90
6.10	Référence du fichier src/Legislation.h . . . . .	90
6.11	Référence du fichier src/Migration.h . . . . .	90
6.11.1	Documentation des fonctions . . . . .	90
6.12	Référence du fichier src/Mortalite.h . . . . .	91
6.12.1	Documentation des fonctions . . . . .	92
6.13	Référence du fichier src/Naissance.h . . . . .	92
6.13.1	Documentation des fonctions . . . . .	93
6.14	Référence du fichier src/OutilsBase.h . . . . .	93
6.14.1	Description détaillée . . . . .	95
6.14.2	Documentation des macros . . . . .	95
6.14.3	Documentation des fonctions . . . . .	95
6.15	Référence du fichier src/OutilsComp.h . . . . .	98

6.15.1 Description détaillée . . . . .	98
6.16 Référence du fichier src/Retraite.h . . . . .	99
6.17 Référence du fichier src/Reversion.h . . . . .	99
6.18 Référence du fichier src/Salaires.h . . . . .	99
6.19 Référence du fichier src/Sante.h . . . . .	100
6.19.1 Description détaillée . . . . .	100
6.19.2 Documentation des fonctions . . . . .	100
6.20 Référence du fichier src/Separations.h . . . . .	100
6.20.1 Documentation des fonctions . . . . .	101
6.21 Référence du fichier src/Simulation.h . . . . .	102
6.21.1 Description détaillée . . . . .	103
6.21.2 Documentation des fonctions . . . . .	103
6.21.3 Documentation des variables . . . . .	103
6.22 Référence du fichier src/Statistiques.h . . . . .	104
6.23 Référence du fichier src/Transitions.h . . . . .	104
6.23.1 Description détaillée . . . . .	105
6.23.2 Documentation du type de l'énumération . . . . .	105
<b>7 Guide de prise en main du modèle Destinie</b>	<b>105</b>
7.1 Lancer une projection avec le modèle Destinie . . . . .	105
7.1.1 Installation minimale . . . . .	106
7.1.2 Réaliser une première simulation . . . . .	106
7.2 Pour obtenir/utiliser les sources . . . . .	109
<b>Références bibliographiques</b>	<b>110</b>

## 1 Introduction

Le modèle Destinie 2 (modèle Démographique Économique et Social de Trajectoires INdividuelles sImulÉs, version 2) est un modèle de microsimulation dynamique, développé et géré par l'Insee, dont l'objectif principal est la projection à long terme des retraites. Il succède à Destinie 1, premier modèle de microsimulation élaboré sur ce domaine en France, depuis 2010.

## Utilisation

Ce modèle est mobilisé lors des exercices de projection de long terme conduits par le Conseil d'Orientation des Retraites (COR) ou la Commission européenne, mais aussi dans le cadre d'études afin d'éclairer le débat économique et social dans le domaine de la protection sociale (retraites, mais également dépendance ou encore dépenses de santé). En effet, la construction d'un échantillon démographique, couplée à la constitution des liens familiaux et des trajectoires professionnelles, offre au modèle une perspective plus généraliste, qui s'étend au-delà des retraites stricto sensu. Sa dimension « ménages » le distingue d'autres modèles de microsimulation. Par ailleurs, la taille modeste de l'échantillon permet une simulation rapide et ainsi la réalisation de nombreuses variantes, au prix d'une plus faible précision des résultats.

## Méthodologie

Le modèle s'appuie sur un échantillon représentatif de la population française au 1er janvier 2010, construit à partir de l'enquête Patrimoine 2009-2010. Composé d'environ 62 000 individus, cet échantillon initial contient des informations sur les liens familiaux ainsi que l'historique des positions occupées sur le marché du travail. La simulation est effectuée au niveau individuel. Le modèle renouvelle la population initiale en simulant des naissances, des décès et des flux migratoires. Il projette ensuite les carrières professionnelles, les revenus d'activité et le départ à la retraite de chaque individu. Les personnes en emploi sont réparties en trois grands groupes (les salariés du secteur privé, les titulaires de la fonction publique et les indépendants), comme les retraités (en autant de régimes de base). Ce modèle simplifie un certain nombre d'éléments législatifs (par exemple, les indépendants sont simulés en étant tous soumis aux mêmes règles de calcul des droits). Destinie 2 permet de reconstituer l'ensemble de la trajectoire professionnelle d'un individu (statuts d'activité et revenus) et simule la liquidation de la retraite selon les règles en vigueur pour chaque régime.

## Trois modules principaux

Le modèle comprend trois modules distincts, écrits en C++ et transcrits en R : le module générateur des biographies démographiques, le module générateur des trajectoires professionnelles et le module « retraite ». La construction des trajectoires démographiques s'appuie sur les projections démographiques de l'Insee publiées en 2016, couvrant le champ France entière sur la période 2013-2070. Le nombre de décès, de naissances et les flux migratoires sont ainsi calés chaque année de la projection. Les parcours professionnels des individus sont observés jusqu'en 2009, année de base. À compter de 2010, leurs carrières (statuts d'activité et revenus) sont projetées en respectant des contraintes de calage fondées sur des hypothèses macroéconomiques. Ces hypothèses portent sur les gains de productivité du travail, le taux de chômage, tous deux repris des publications du COR, et le taux d'activité, issu des projections de population active de l'Insee. Le module « retraite » simule le départ à la retraite et calcule le montant des droits associés. Ces droits incluent les droits directs et la pension de réversion des régimes de base (le régime général de la Sécurité sociale, le Service des Retraites de l'État (SRE), la Caisse nationale de retraites des agents des collectivités locales (CNRACL), la Sécurité Sociale pour les Indépendants (SSI) et de certains régimes complémentaires (Association pour le régime de retraite complémentaire des salariés (Arrco), Association générale des institutions de retraite complémentaire des cadres (Agirc)). Outre la réversion, la dimension « ménages » permet de simuler l'attribution de l'Allocation de Solidarité aux Personnes Âgées (Aspa).

## Plusieurs options de simulation

Le modèle permet une simulation à la carte via un vaste choix d'options. Certaines options permettent d'évaluer la robustesse des résultats obtenus, d'autres ont été rajoutées à l'occasion d'études spécifiques. Les options incluent :

- le scénario démographique de l'Insee utilisé en projection relatif à la mortalité, à la fécondité ou encore au solde migratoire.
- la législation à appliquer : soit la dernière législation intégrée, soit une précédente législation.
- le type de mortalité : prise en compte ou non d'une mortalité différenciée selon le niveau de diplôme.

Les autres options présentes dans le modèle sont documentées sur le dépôt en ligne.

## 2 Vue d'ensemble

Le programme est un package R écrit en C++. Il peut être directement utilisé à partir d'une console en R, pour effectuer des simulations à partir d'un échantillon et de paramètres importés sous R. En revanche, le code source du package est écrit en C++.

Structure des répertoires du package :

**src** contient les fichiers sources du package écrits en C++

**R** contient les fichiers sources du package écrits en R

**parametres** contient les paramètres démographiques et macroéconomiques utilisés lors de la projection

Le sous-répertoire `Projections_COR_2018` comprend les 5 scénarios économiques du rapport de juin 2018 du COR (voir la table 1).

**TABLE 1 Scénarios économiques du COR**

Gain de productivité du travail (en %)	Taux de chômage de long terme (en %)
1,8	4,5
1,8	7
1,5	7
1,3	7
1	7
1	10

**data** contient un échantillon test **non représentatif** de la population mais que l'on peut utiliser pour prendre en main le modèle de microsimulation mais aussi des jeux de paramètres du modèle préchargés sous forme d'environnement.

**data\_raw** contient les scripts R ayant permis d'obtenir les environnements de paramètres et l'échantillon non représentatif. Ces fichiers peuvent être réutilisés pour construire d'autres jeux d'hypothèses.

**demo** contient le script `simulation.R` permettant de lancer une première simulation avec le modèle.

**documentation** contient la documentation du package, des conseils pratiques, la liste des documents de travail de l'Insee publiés à partir du modèle Destinie 2 et la liste des contributeurs du modèle.

Le fichier `documentation.R` permet de réaliser cette documentation en créant une documentation en latex, plus la documentation du package en ligne.

### 2.1 Le package Destinie

Le programme est principalement structuré autour de cinq classes d'objets et de structures (voir la figure 1).

**La classe Simulation** contient l'ensemble des paramètres de la simulation ainsi que l'échantillon issu du générateur de biographies utilisé sous la forme d'un tableau d'objets `Indiv`.

**La classe Indiv** contient l'ensemble des informations relatives à un individu donné.

**La classe Retraite** contient les variables sur les pensions (de droit direct ou dérivé) pour un individu et une année donnés.

**La classe DroitsRetr** contient l'ensemble des variables concernant la liquidation des droits directs.

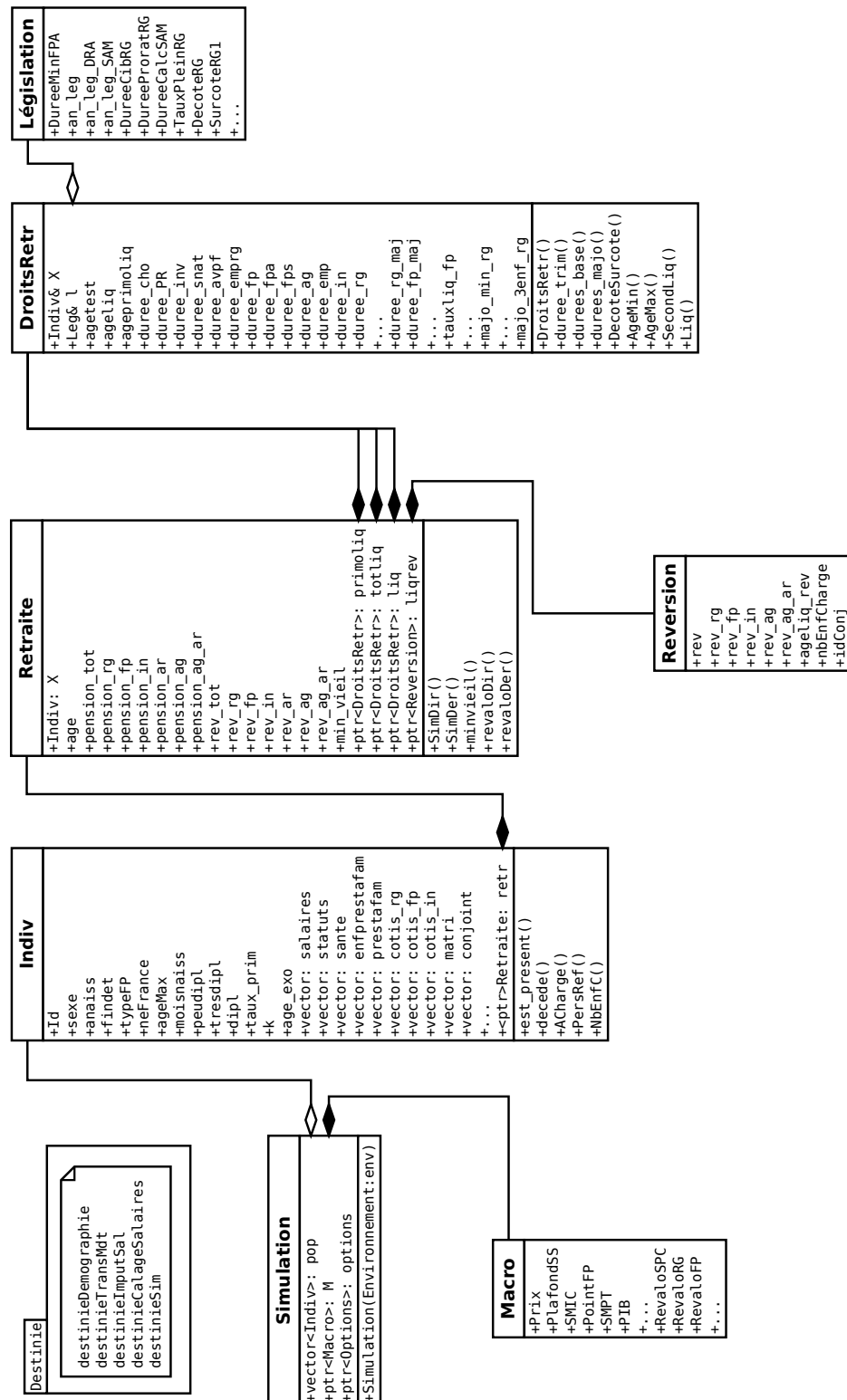


**La classe Legislation** contient l'ensemble des paramètres législatifs pour une date de législation, une année et un individu donnés.

**La structure Macro** contient l'ensemble des paramètres macroéconomiques (PIB, inflation,...) mais aussi des paramètres plus spécifiques aux retraites (taux de cotisation, valeur des points dans les complémentaires ou encore montant du minimum vieillesse).

De plus, les fichiers `Demographie.cpp` et `Destinie.h` contiennent les fonctions que l'on peut directement appeler depuis R, ainsi que les constantes du modèles. Le fichier `OutilsBase.h` contient des fonctions utilitaires.

FIGURE 1 Structure générale du package



## 2.2 Options et paramètres de simulation

Un vaste choix d'options et de paramètres permet une simulation à la carte. Ce choix porte sur les hypothèses démographiques, économiques et réglementaires.

### 2.2.1 Hypothèses démographiques

Les scénarios démographiques se distinguent par leurs variantes sur trois composantes que sont la natalité, la mortalité et le solde migratoire. Pour chaque composante, trois variantes existent : centrale (modalité *Cent*), basse (modalité *Bas*), haute (modalité *Haut*). Il existe autant de scénarios que de croisements possibles entre ces variantes, soit 27. Les scénarios s'appuient sur les projections démographiques de l'Insee de 2016, couvrant la période 2013-2070.

### 2.2.2 Hypothèses économiques

Le cadre macroéconomique est défini selon deux axes : le scénario de chômage et de productivité du travail. Trois scénarios de taux de chômage de long terme sont proposés : 4,5 %, 7 %, 10 %. Quant à la productivité, elle évolue selon quatre variantes de croissance tendancielle : 1,0 %, 1,3 %, 1,5 %, 1,8 %.

### 2.2.3 Hypothèses réglementaires et autres

Avant d'effectuer une simulation, certains paramètres auxiliaires doivent être renseignés dans l'objet *options*. Il s'agit :

- de la dernière année de législation (variable *anLeg*)
- du mode de comportement de liquidation <sup>1</sup>
- du pas de test de liquidation entre l'âge d'ouverture des droits et l'âge d'annulation de la décote (variable *pas1*)
- du pas de test de liquidation avant l'âge d'ouverture des droits et après l'âge d'annulation de la décote (variable *pas2*)
- de l'horizon de projection (variable *AN\_MAX*)
- du champ de la projection : France entière (modalité "FE") ou métropolitaine (modalité "FM")

D'autres options existent et sont détaillées dans la partie 5.27. Parmi celles-ci, toutes les options booléennes valent 0 par défaut.

## 2.3 Choix de la législation

La liquidation des droits se fait selon la législation en vigueur à une date choisie par l'utilisateur, stockée dans la variable *anLeg* de l'objet *options*. La législation est alors figée à celle prévalant à cette date.

### 2.3.1 Réforme de 1993

- Concerne uniquement les salariés du secteur privé et les régimes alignés ;
- Allongement progressif de la durée d'assurance requise pour accéder au taux plein (modification progressive pour les générations 1934-1943 à raison d'un trimestre supplémentaire par génération) ;
- Calcul du salaire de référence sur les 25 meilleures années au lieu de 10 (modification progressive pour les générations 1934-1947 à raison d'une année supplémentaire par génération) ;
- Les pensions et les salaires portés au compte sont indexés sur les prix, et non plus sur les salaires (principe déjà en vigueur depuis 1987) ;

**Circulaire n°103/93 du 30 décembre 1993 Caisse nationale d'assurance vieillesse**

1. Bien que plusieurs modèles soient proposés, la plupart n'ont pas été expertisés depuis un certain temps. Le choix le plus prudent est celui du départ au taux plein (modalité "tp").

TABLE 2 Détail par génération de la période transitoire 1993

Année de naissance	Nombre d'années retenues pour le calcul du SAM	Trimestres d'assurance pour obtenir le taux plein
1933 et avant	10	150
1934	11	151
1935	12	152
1936	13	153
1937	14	154
1938	15	155
1939	16	156
1940	17	157
1941	18	158
1942	19	159
1943	20	160
1944	21	160
1945	22	160
1946	23	160
1947	24	160
1948 et au-delà	25	160

### 2.3.2 Réforme de 2003

#### 2.3.2.1 Le régime général

- Allongement progressif de la durée d'assurance requise pour accéder au taux plein à partir de la génération 1949 (passage de 160 trimestres à 164)
- Allongement de la durée intervenant dans le coefficient de proratisation, censée à terme évoluer comme la durée cible d'assurance
- Réduction de la décote à partir de la génération 1944, de 0,5 point par an pour atteindre 5% par annuité manquante pour les générations nées après 1952 (la décote n'est appliquée que si la liquidation a lieu avant 65 ans)
- Surcote de 3 % par année supplémentaire travaillée après le 1er janvier 2004 ; la surcote a été renforcée par le plan « emploi des seniors » au printemps 2005
- Modification du mode de calcul du minimum contributif
- Possibilité de départ anticipé pour carrière longue : depuis le 1er janvier 2004, les assurés qui ont commencé à travailler très jeunes (avant 18 ans), et qui ont eu une longue carrière peuvent partir à la retraite avant 60 ans (la condition de début de carrière est un obstacle pour les assurés nés à partir de 1953 qui ont connu une scolarité obligatoire jusqu'à 16 ans).

#### 2.3.2.2 Fonction publique

- Augmentation de la durée d'assurance à partir de 2004 pour rejoindre celle requise dans le régime général et ensuite progresser en parallèle à partir de 2006
- Décote de 0,5 point par annuité manquante, augmentée de 0,5 point à chaque génération pour atteindre 5% en 2015 (avec hausse progressive de l'âge pivot à partir duquel la décote s'annule)
- Surcote de 3 % par année supplémentaire travaillée, comme dans le secteur privé
- Modification du mode de calcul du minimum garanti
- Modification des droits familiaux et conjugaux

### 2.3.3 Réforme de 2010

- Relèvement progressif de l'âge légal de départ à la retraite pour atteindre **62 ans** en 2018. Cette évolution concerne tous les salariés, du public comme du privé ainsi que les régimes spéciaux, mais avec des calendriers de mise en œuvre différents,

TABLE 3 Détail par génération de la période transitoire 2003

Année de naissance	Nombre d'années retenues pour le calcul du SAM	Trimestres d'assurance pour obtenir le taux plein	Minoration du taux par nombre de trimestre manquant	Trimestres d'assurance maximum retenus pour le calcul RG
1933 et avant	10	150		
1934	11	151		
1935	12	152		
1936	13	153		
1937	14	154		
1938	15	155	-1,2500	150
1939	16	156		
1940	17	157		
1941	18	158		
1942	19	159		
1943	20			
1944	21		-1,1875	152
1945	22	160	-1,1250	154
1946	23		-1,0625	156
1947	24		-1,0000	158
1948			-0,9375	160
1949		161	-0,8750	161
1950	25	162	-0,8125	162
1951		163	-0,7500	163
1952		164	-0,6875	164

- L'âge à partir duquel il est permis à un assuré, n'ayant pas la durée de cotisation requise, de bénéficier tout de même d'une retraite à taux plein, passe progressivement de **65 à 67 ans**,
- Le dispositif des "**carrières longues**" est modifié, les salariés ayant commencé avant 18 ans peuvent partir à la retraite au plus tôt, sous réserve d'avoir la durée de cotisation requise pour leur génération, plus 2 ans.
- Pour les salariés qui, du fait d'une situation d'usure professionnelle, ont une incapacité physique supérieure ou égale à 20 %, l'âge légal de départ à la retraite reste fixé à 60 ans et aucune décote ne leur est appliquée. *Dispositif non intégré dans le modèle.*
- Les jeunes en chômage non indemnisé peuvent valider jusqu'à 6 trimestres (au lieu de 4).
- pour les femmes, l'indemnité journalière perçue pendant le congé maternité entrera dans le salaire de référence sur lequel sera calculée la pension de retraite. *Dispositif non intégré dans le modèle.*
- de nouvelles recettes financières sont instaurées, comme la hausse de la tranche la plus élevée de l'impôt sur le revenu (41 % au lieu de 40 %), l'augmentation des taxes sur les stock-options et les retraites-chapeaux, le relèvement des prélèvements forfaitaires sur les revenus du capital et des taxes sur les dividendes perçus par les actionnaires. *Dispositif non intégré dans le modèle.*
- l'objectif assigné au fonds de réserve des retraites est modifié : ses réserves (36,2 milliards en 2010) seront, à partir de 2011, ponctionnées annuellement (2,1 milliards) au profit de la Caisse d'amortissement de la dette sociale (Cades). *Dispositif non intégré dans le modèle.*

#### 2.3.4 Accélération du calendrier fixé en 2010 suite au projet de loi de financement de la sécurité sociale pour 2012

La loi du 21 décembre 2011 relative au financement de la sécurité sociale en 2012 prévoit une accélération de la réforme de 2010 : l'âge légal de départ et l'âge d'annulation de la décote passent à 62 et 67 ans respectivement, dès 2017 au lieu de 2018.

#### 2.3.5 Décret de juillet 2012 : extension du dispositif carrières longues

Le décret du 2 juillet 2012 étend le dispositif de départs anticipés pour carrière longues, à compter de novembre 2012, aux personnes qui ont commencé à travailler avant 20 ans, et en supprime la condition relative à la durée d'assurance totale (la condition relative à la durée d'assurance cotisée est maintenue mais aménagée).

## 2.3.6 Réforme de 2013

- Augmentation de la durée de cotisation d'un trimestre tous les 3 ans à partir de la génération née en 1958 pour atteindre 43 ans pour la génération 1973.
- Les périodes d'apprentissage, la formation, le chômage et la maternité sont mieux pris en compte. *Dispositif non intégré dans le modèle.*
- Un trimestre est considéré comme validé dès que le salarié a cotisé 200 heures au Smic. Ce seuil passe à 150 heures au bénéfice des personnes à temps très partiel, avec un plafond fixé à 1,5 Smic. *Dispositif non intégré dans le modèle.*
- Pour les salariés du public comme du privé, les cotisations patronales et salariales sont augmentées de 0,15 point chacune dès le 1er janvier 2014, puis de 0,05 point en 2015, 2016 et 2017.
- Mise en place d'un compte pénibilité. *Dispositif non intégré dans le modèle.*
- Le bonus pour les parents de trois enfants est fiscalisé. *Dispositif non intégré dans le modèle.*
- La revalorisation des pensions est décalée de six mois chaque année.
- Mise en place de la Lura (Liquidation unique des régimes alignés) à compter du 1er juillet 2017. Pour les individus ayant été affiliés à au moins deux des trois régimes alignés (régime général, régime social des indépendants et MSA-salariés), et dont la liquidation intervient à partir du 1er juillet 2017, la pension est calculée de façon unifiée. Le salaire de référence est calculée sur la base des 25 meilleures années (tous régimes confondus), et la durée validée par année est bornée à quatre trimestres (par exemple, pour un individu ayant validé quatre trimestres au régime général et deux trimestres au RSI au cours d'une année, seuls quatre trimestres sont pris en compte dans la durée d'assurance). Le versement de la pension, due au titre des droits ouverts dans les trois régimes alignés, est assuré par le dernier régime d'affiliation. Sur ce point, Destinie 2 s'écarte de la législation : la pension est proratisée en fonction des durées validées dans chaque régime et les proratas sont comptés à la charge des régimes concernés. Cet écart permet d'évaluer les dépenses effectivement engagées par les régimes.

## 2.3.7 Accord national interprofessionnel relatif aux retraites complémentaires Agirc-Arrco du 30 octobre 2015

- En 2016, 2017 et 2018, indexation de la valeur du service des points Agirc et Arrco sur l'inflation moins un point.
- En 2016, 2017 et 2018, indexation de la valeur d'achat des point Agirc et Arrco sur l'évolution du SMPT augmentée de deux points.
- Extension de la cotisation AGFF sur la tranche B à la tranche C à compter du 1er janvier 2016.
- Pour les individus liquidant leur retraite de base au taux plein, application d'un coefficient de minoration de 10% sur la pension complémentaire, et ce, pour une durée de trois ans dans la limite de l'âge d'annulation de la décote. Ce coefficient ne s'applique pas aux individus qui liquident un an au moins après avoir rempli les conditions du taux plein. Pour les individus exonérés de la CSG ou soumis à un taux réduit, le coefficient ne s'applique pas ou est réduit.
- Pour les individus remplissant les conditions du taux plein dans les régimes de base mais reportant leur liquidation de la retraite complémentaire d'au moins deux années, application, pour une durée d'un an, d'un coefficient majorant :
  - de 10 % si la liquidation a lieu au moins deux ans après
  - de 20 % si la liquidation a lieu au moins trois ans après
  - de 30 % si la liquidation a lieu au moins quatre ans après

## 3 Index des classes

## 3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

**CiblesDemo**

**CiblesDemo** contient les cibles démographiques de naissances et de solde migratoire

**13**

<b>CiblesTrans</b>	
<b>CiblesTrans</b> contient les cibles de calage sur le marché du travail	15
<b>Cotisations</b>	
La classe <b>Cotisations</b> contient, entre autres choses, le salaire et les cotisations retraites à une date donnée	18
<b>CotisationsTRI</b>	
La structure <b>CotisationsTRI</b> contient la fonction <b>CotisationsTRI</b> qui calcule l'intégralité des cotisations d'un individu	19
<b>CStructSexeAge</b>	
La structure <b>CStructSexeAge</b> contient la série des coefficients correcteurs utilisés lors de la simulation des salaires, pour les femmes et les hommes	20
<b>DroitsRetr</b>	
Droits directs par régime	20
<b>Ech</b>	
<b>Ech</b> contient les caractéristiques invariantes au cours du temps des individus	30
<b>Emp</b>	
<b>Emp</b> contient les statuts sur le marché du travail et les revenus d'activité	31
<b>EqSalaires</b>	
La structure <b>EqSalaires</b> contient les coefficients des variables explicatives des équations de salaires, pour un sexe et un secteur donnés	31
<b>EqSante</b>	
<b>EqSante</b> contient les coefficients pour les équations de santé (adl, iadl, gali)	32
<b>EqTrans</b>	
<b>EqTrans</b> contient les équations de transitions sur le marché du travail	33
<b>Fam</b>	
<b>Fam</b> contient la chronique des événements familiaux depuis 2009	35
<b>FinEtudeMoy</b>	
<b>FinEtudeMoy</b> désigne l'âge moyen de fin d'études par sexe et par génération	36
<b>Indic_annee</b>	
La structure <b>Indic_annee</b> contient des agrégats sur un régime de retraite au cours d'une année, tels que les effectifs de cotisants, de retraités, de liquidants, les montants de cotisations, de dépenses, la pension moyenne, l'âge moyen de liquidation..	36
<b>Indic_gen</b>	
Contient des indicateurs par génération : les différents motifs de liquidation, les moyennes d'âge à la liquidation, de durée validée, taux de remplacement..	38
<b>Indic_gen_age</b>	
Cette structure contient les principaux indicateurs calculés par génération et par âge	39
<b>Indicateur</b>	
Structure à l'origine des structures <b>Moyenne</b> , <b>Ratio</b> , <b>Somme</b> et <b>Taux</b>	40
<b>Indiv</b>	
Classe contenant l'ensemble des informations sur un individu	40
<b>Leg</b>	
Contient les valeurs des paramètres législatifs pour un individu, un âge et une date de législation donnés	44

**Macro**

**Macro** contient l'ensemble des paramètres macroéconomiques et des paramètres spécifiques aux retraites 46

**Mortadiff\_dip\_F**

Quotients de mortalité par âge et niveau de diplôme des femmes observés sur la période 2009-2013 56

**Mortadiff\_dip\_H**

Quotients de mortalité par âge et niveau de diplôme des hommes observés sur la période 2009-2013 57

**Mortalite\_diff**

**Mortalite\_diff** permet de réaliser l'ancienne mortalité différentielle à la [dubois2015taux] 58

**Moyenne**

Structure permettant de manipuler des moyennes 58

**Opt**

Structure utilisée pour coder destinieFinCar 59

**Option\_optim**

Structure utilisée pour coder destinieSimOptim 59

**Options**

**Options** contient l'ensemble des options de simulation 59

**Options\_salaires**

La structure **Options\_salaires** contient un attribut booléen, valant 1 pour une simulation des salaires sans aléa, et 0 sinon 61

**OptionsTRI**

La structure **OptionsTRI** contient les 3 options pour le calcul du taux de rendement interne 62

**ordre\_findet**

Compare les individus en fonction de leur âge de fin d'étude 62

**Paire**

**Paire** contient les identifiants et leurs probabilités ainsi que la structure d'ordre associée 62

**quantile**

Structure permettant de manipuler des quantiles 63

**Ratio**

Structure permettant de manipuler des ratios 63

**Retraite**

Pensions de droits directs et dérivés ainsi que le minimum vieillesse à l'âge courant 63

**Retraite\_comp**

Contient les éléments permettant de calculer les retraites complémentaires 69

**Reversion**

Calcul de la liquidation d'une reversion 70

**Salaire**

La structure **Salaire** contient les équations et les fonctions permettant l'imputation des revenus d'activité 71



**Simulation**

La structure **Simulation** contient l'ensemble des paramètres, options et séries macroéconomique ou de paramètres législatifs de la simulation, ainsi que le vecteur d'individus pop qui contient l'ensemble des individus 72

**Somme**

Structure permettant de manipuler des sommes 73

**Taux**

Structure permettant de manipuler des taux 73

**Transitions**

La structure **Transitions** contient tous les éléments utilisés dans la fonction destinieTransMdt 74

## 4 Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

src/ <b>Constantes.h</b>	Fichier contenant les constantes générales	76
src/ <b>Cotisations.h</b>	Fichier contenant les outils permettant le calcul des cotisations prélevées sur les revenus d'activité et la pension de retraite	83
src/ <b>CotisationsTRI.h</b>	Code pour le calcul du taux de rendement interne	86
src/ <b>Destinie.h</b>	Contient les fonctions <b>calage_salaires</b> , <b>destinieSim</b> et <b>destinieSimIndiv</b> pouvant être appelées depuis R	86
src/ <b>DroitsRetr.h</b>	Calcul des droits à la retraite (cf. <b>DroitsRetr</b> )	87
src/ <b>Indicateurs_annee_COR.h</b>	Ce fichier permet de calculer des agrégats annuels par régime et par sexe, ainsi que la chronique annuelle des cotisations et du salaire net par individus. Les résultats, stockés dans les tables Indicateurs_an, cotisations et salairenet, sont exportés dans R	88
src/ <b>Indicateurs_gen.h</b>	Permet d'exporter la structure éponyme avec notamment les motifs de liquidation de la pension	88
src/ <b>Indicateurs_gen_age.h</b>	Contient la structure éponyme qui contient des indicateurs agrégés par génération et par âge	89
src/ <b>Indiv.h</b>	Contient la définition de la classe <b>Indiv</b>	89
src/ <b>Legislation.h</b>	Contient la classe <b>Leg</b> qui définit l'ensemble des paramètres législatifs pour un individu à un âge et une date de législation donnés	90
src/ <b>Migration.h</b>	Migration détermine les individus qui émigrent et crée de nouveaux individus immigrants en respectant les cibles de solde migratoire par tranche d'âge	90

src/ <a href="#">Mortalite.h</a>	Ce fichier contient la fonction <a href="#">mortalite</a> qui simule les décès en projection	91
src/ <a href="#">Naissance.h</a>	Contient la fonction <a href="#">naissance</a> qui simule les naissances	92
src/ <a href="#">OutilsBase.h</a>	Contient une bibliothèque de fonctions utilitaires	93
src/ <a href="#">OutilsComp.h</a>		98
src/ <a href="#">Retraite.h</a>	Contient la définition de la classe <a href="#">Retraite</a> et l'implémentation de quelques fonctions du module retraite	99
src/ <a href="#">Reversion.h</a>	Simule la liquidation de la pension de réversion	99
src/ <a href="#">Salaires.h</a>	Fichier contenant les outils permettant de simuler les salaires, à la fois en rétrospectif et en projection	99
src/ <a href="#">Sante.h</a>	Imputation d'un état de sante pour les plus de 50 ans (adl, iadl, gali)	100
src/ <a href="#">Separations.h</a>	Contient les fonctions simulant les mises en couple et les séparations	100
src/ <a href="#">Simulation.h</a>	Fichier contenant la structure <a href="#">Simulation</a> , qui importe les paramètres et l'échantillon depuis les objets en R et les rend accessibles en C avec des objets globaux (accessibles dans tout le programme)	102
src/ <a href="#">Statistiques.h</a>	Statistiques contient des structures qui facilitent la manipulation des indicateurs suivants : moyenne, somme, taux, ratio, quantile. Ces structures sont notamment utilisées dans <a href="#">Indicateurs_annee_COR.h</a>	104
src/ <a href="#">Transitions.h</a>	Fichier contenant la structure <a href="#">Transitions</a>	104

## 5 Documentation des classes

### 5.1 Référence de la structure CiblesDemo

[CiblesDemo](#) contient les cibles démographiques de naissances et de solde migratoire.

```
#include <Simulation.h>
```

#### Fonctions membres publiques

- NumericVector [\\_](#) (Naissances)  
*Cibles de naissances.*
- NumericVector [\\_](#) (Soldemig)  
*Solde migratoire global.*
- NumericVector [\\_](#) (Soldemig\_G)

- *Solde migratoire global des garçons.*  
NumericVector \_ (Soldemig\_G\_1\_5)
- *Proportion de migrants garçons dont l'âge est compris entre 1 et 5 ans.*  
NumericVector \_ (Soldemig\_G\_6\_11)
- *Proportion de migrants garçons dont l'âge est compris entre 1 et 11 ans.*  
NumericVector \_ (Soldemig\_G\_12\_17)
- *Proportion de migrants garçons dont l'âge est compris entre 1 et 17 ans.*  
NumericVector \_ (Soldemig\_Fi)
- *Solde migratoire global de filles.*  
NumericVector \_ (Soldemig\_F\_1\_5)
- *Proportion de migrantes filles dont l'âge est compris entre 1 et 5 ans.*  
NumericVector \_ (Soldemig\_F\_6\_11)
- *Proportion de migrantes filles dont l'âge est compris entre 1 et 11 ans.*  
NumericVector \_ (Soldemig\_F\_12\_17)
- *Proportion de migrantes filles dont l'âge est compris entre 1 et 17 ans.*  
NumericVector \_ (Soldemig\_H)
- *Solde migratoire global d'hommes adultes d'âge compris entre 27 et 60 ans.*  
NumericVector \_ (Soldemig\_H\_27\_30)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 30 ans.*  
NumericVector \_ (Soldemig\_H\_31\_35)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 35 ans.*  
NumericVector \_ (Soldemig\_H\_36\_40)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 40 ans.*  
NumericVector \_ (Soldemig\_H\_41\_45)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 45 ans.*  
NumericVector \_ (Soldemig\_H\_46\_50)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 50 ans.*  
NumericVector \_ (Soldemig\_H\_51\_55)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 55 ans.*  
NumericVector \_ (Soldemig\_H\_56\_60)
- *Proportion d'hommes migrants dont l'âge est compris entre 27 et 60 ans.*  
NumericVector \_ (Soldemig\_Fe)
- *Solde migratoire global des femmes adultes d'âge compris entre 25 et 59 ans.*  
NumericVector \_ (Soldemig\_F\_25\_29)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 29 ans.*  
NumericVector \_ (Soldemig\_F\_30\_34)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 34 ans.*  
NumericVector \_ (Soldemig\_F\_35\_39)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 39 ans.*  
NumericVector \_ (Soldemig\_F\_40\_44)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 44 ans.*  
NumericVector \_ (Soldemig\_F\_45\_49)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 49 ans.*  
NumericVector \_ (Soldemig\_F\_50\_54)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 54 ans.*  
NumericVector \_ (Soldemig\_F\_55\_59)
- *Proportion de femmes migrantes dont l'âge est compris entre 25 et 59 ans.*  
NumericVector \_ (proba\_H\_18)
- *Probabilité d'un homme âgé de 18 ans d'émigrer.*  
NumericVector \_ (proba\_H\_19)
- *Probabilité d'un homme âgé de 19 ans d'émigrer.*  
NumericVector \_ (proba\_H\_20)
- *Probabilité d'un homme âgé de 20 ans d'émigrer.*  
NumericVector \_ (proba\_H\_21)
- *Probabilité d'un homme âgé de 21 ans d'émigrer.*  
NumericVector \_ (proba\_H\_22)
- *Probabilité d'un homme âgé de 22 ans d'émigrer.*  
NumericVector \_ (proba\_H\_23)
- *Probabilité d'un homme âgé de 23 ans d'émigrer.*  
NumericVector \_ (proba\_H\_24)
- *Probabilité d'un homme âgé de 24 ans d'émigrer.*  
NumericVector \_ (proba\_H\_25)
- *Probabilité d'un homme âgé de 25 ans d'émigrer.*  
NumericVector \_ (proba\_H\_26)
- *Probabilité d'un homme âgé de 26 ans d'émigrer.*  
NumericVector \_ (proba\_F\_18)
- *Probabilité d'une femme âgée de 18 ans d'émigrer.*  
NumericVector \_ (proba\_F\_19)
- *Probabilité d'une femme âgée de 19 ans d'émigrer.*  
NumericVector \_ (proba\_F\_20)

- *Probabilité d'une femme âgée de 20 ans d'émigrer.*  
NumericVector\_(proba\_F\_21)
- *Probabilité d'une femme âgée de 21 ans d'émigrer.*  
NumericVector\_(proba\_F\_22)
- *Probabilité d'une femme âgée de 22 ans d'émigrer.*  
NumericVector\_(proba\_F\_23)
- *Probabilité d'une femme âgée de 23 ans d'émigrer.*  
NumericVector\_(proba\_F\_24)
- *Probabilité d'une femme âgée de 24 ans d'émigrer.*  
NumericVector\_(proba\_F\_25)
- *Probabilité d'une femme âgée de 25 ans d'émigrer.*  
NumericVector\_(nbre\_emig\_f)
- *Cible totale de femmes émigrant entre 18 et 25 ans.*  
NumericVector\_(nbre\_emig\_h)
- *Cible totale de hommes émigrant entre 18 et 26 ans.*  
NumericVector\_(emigrant\_F\_0)
- *Cible totale de filles émigrant avant l'âge d'un an.*  
NumericVector\_(emigrant\_H\_0)
- *Cible totale de garçons émigrant avant l'âge d'un an.*

### 5.1.1 Description détaillée

La structure [CiblesDemo](#) contient, pour chaque année de projection, le nombre de naissances, ainsi que le solde migratoire ventilé par sexe et par âge. Attention : le solde migratoire global cible ne correspond pas à la somme des soldes migratoires déclinés par âge et sexe, car les soldes migratoires négatifs au-delà de 60 ans ne sont pas simulés.

Après 2013, ces cibles par âge fin sont issues de **[blanpainprojections]**. En revanche, pour les années 2010 à 2012 incluses, elles ont été fournies aux fins de ces exercices de modélisation par la division Enquêtes et études démographiques de l'Insee, avec l'avertissement que certaines peuvent être fragiles. Pour cette raison, elles ne sont pas publiées. Ces valeurs sont utilisées uniquement pour compléter des séries, et ne doivent l'être qu'à ce titre.

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/Simulation.h](#)

## 5.2 Référence de la structure CiblesTrans

[CiblesTrans](#) contient les cibles de calage sur le marché du travail.

```
#include <Transitions.h>
```

### Fonctions membres publiques

- NumericVector\_(CHO\_H1519)  
*Cible de taux de chômage en projection chez les hommes de 15 à 19 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector\_(CHO\_H2024)  
*Cible de taux de chômage en projection chez les hommes de 20 à 24 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector\_(CHO\_H2529)  
*Cible de taux de chômage en projection chez les hommes de 25 à 29 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*

- NumericVector \_ (CHO\_H3034)  
*Cible de taux de chômage en projection chez les hommes de 30 à 34 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_H3539)  
*Cible de taux de chômage en projection chez les hommes de 35 à 39 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_H4044)  
*Cible de taux de chômage en projection chez les hommes de 40 ans à 44 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_H4549)  
*Cible de taux de chômage en projection chez les hommes de 45 à 49 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_H5054)  
*Cible de taux de chômage en projection chez les hommes de 50 à 54 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F1519)  
*Cible de taux de chômage en projection chez les femmes de 15 à 19 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F2024)  
*Cible de taux de chômage en projection chez les femmes de 20 à 24 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F2529)  
*Cible de taux de chômage en projection chez les femmes de 25 à 29 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F3034)  
*Cible de taux de chômage en projection chez les femmes de 30 à 34 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F3539)  
*Cible de taux de chômage en projection chez les femmes de 35 à 39 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F4044)  
*Cible de taux de chômage en projection chez les femmes de 40 à 44 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F4549)  
*Cible de taux de chômage en projection chez les femmes de 45 à 49 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (CHO\_F5054)  
*Cible de taux de chômage en projection chez les femmes de 50 à 54 ans*  
*Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage ; estimés sur l'Enquête Emploi en Continu pour les années 2010-2011-2012*  
*Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H1519)

- Cible de taux d'inactivité en projection chez les hommes de 15 à 19 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H2024)
  - Cible de taux d'inactivité en projection chez les hommes de 20 à 24 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H2529)
  - Cible de taux d'inactivité en projection chez les hommes de 25 à 29 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H3034)
  - Cible de taux d'inactivité en projection chez les hommes de 30 à 34 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H3539)
  - Cible de taux d'inactivité en projection chez les hommes de 35 à 39 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H4044)
  - Cible de taux d'inactivité en projection chez les hommes de 40 à 44 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H4549)
  - Cible de taux d'inactivité en projection chez les hommes de 45 à 49 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_H5054)
  - Cible de taux d'inactivité en projection chez les hommes de 50 à 54 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F1519)
  - Cible de taux d'inactivité en projection chez les femmes de 15 à 19 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F2024)
  - Cible de taux d'inactivité en projection chez les femmes de 20 à 24 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F2529)
  - Cible de taux d'inactivité en projection chez les femmes de 25 à 29 ans*
  - Source : cible proposée par le COR, avec plusieurs variantes de taux de chômage*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F3034)
  - Cible de taux d'inactivité en projection chez les femmes de 30 à 34 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F3539)
  - Cible de taux d'inactivité en projection chez les femmes de 35 à 39 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F4044)
  - Cible de taux d'inactivité en projection chez les femmes de 40 à 44 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F4549)
  - Cible de taux d'inactivité en projection chez les femmes de 45 à 49 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (INA\_F5054)
  - Cible de taux d'inactivité en projection chez les femmes de 50 à 54 ans*
  - Source : cible proposée par le COR*
  - Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (PART\_FP)
  - Part de la Fonction Publique dans l'emploi total*
  - Source : cible proposée par le COR*

- *Feuille : Param\_transitions/ choX%.*
- NumericVector \_ (PART\_FPE)
- *Part de la Fonction Publique d'État dans la Fonction Publique*
- *Source : cible proposée par le COR*
- *Feuille : Param\_transitions/ choX%.*

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Transitions.h](#)

### 5.3 Référence de la structure Cotisations

La classe [Cotisations](#) contient, entre autres choses, le salaire et les cotisations retraites à une date donnée.

```
#include <Cotisations.h>
```

#### Fonctions membres publiques

- **Cotisations** ([Indiv](#) &X, int t)
- NumericVector \_ (Date)
- NumericVector \_ (CotEmp\_ss\_plaf)
- NumericVector \_ (CotEmpMaladie\_ss\_plaf)
- NumericVector \_ (CotEmpMaladie\_sal\_tot)
- NumericVector \_ (CotEmpMaladie\_alleg\_txmin)
- NumericVector \_ (CotEmpMaladie\_alleg\_txmax)
- NumericVector \_ (CotEmpMaladie\_alleg\_salmin)
- NumericVector \_ (CotEmpMaladie\_alleg\_salmax)
- NumericVector \_ (CotEmpRetr\_ss\_plaf)
- NumericVector \_ (CotEmpRetr\_sal\_tot)
- NumericVector \_ (CotEmpRetr\_alleg\_txmin)
- NumericVector \_ (CotEmpRetr\_alleg\_txmax)
- NumericVector \_ (CotEmpRetr\_alleg\_sal\_min)
- NumericVector \_ (CotEmpRetr\_alleg\_salmax)
- NumericVector \_ (CotEmpAutonomie)
- NumericVector \_ (CotEmpFamille\_ss\_plaf)
- NumericVector \_ (CotEmpFamille\_sal\_tot)
- NumericVector \_ (CotEmpFamille\_taux1)
- NumericVector \_ (CotEmpFamille\_taux2)
- NumericVector \_ (CotEmpFamille\_taux3)
- NumericVector \_ (CotEmpFamille\_seuil1)
- NumericVector \_ (CotEmpFamille\_seuil2)
- NumericVector \_ (CotEmpFamille\_txmin)
- NumericVector \_ (CotEmpFamille\_txmax)
- NumericVector \_ (CotEmpFamille\_alleg\_sal\_min)
- NumericVector \_ (CotEmpFamille\_alleg\_salmax)
- NumericVector \_ (CotEmpATMP\_sal\_tot)
- NumericVector \_ (CotEmpAssCho\_taux\_trA)
- NumericVector \_ (CotEmpAssCho\_taux\_trB)
- NumericVector \_ (taux\_part\_du\_revenu)
- NumericVector \_ (taux\_part\_de\_conso)
- NumericVector \_ (csg\_FSV\_part\_minvieil)
- NumericVector \_ (taux\_csg\_sal\_FSV)
- NumericVector \_ (taux\_csg\_retr\_FSV)
- NumericVector \_ (Transferts\_Unedic\_Agirc\_Arrco)
- NumericVector \_ (taux\_chomage\_A)
- NumericVector \_ (taux\_chomage\_C)
- NumericVector \_ (CotPatASF\_taux\_trA)
- NumericVector \_ (CotSalASF\_taux\_trA)
- NumericVector \_ (CotPatASF\_taux\_trB)
- NumericVector \_ (CotSalASF\_taux\_trB)
- NumericVector \_ (CotPatAGFF\_taux\_tr1)
- NumericVector \_ (CotSalAGFF\_taux\_tr1)
- NumericVector \_ (CotPatAGFF\_taux\_tr2)
- NumericVector \_ (CotSalAGFF\_taux\_tr2)
- NumericVector \_ (CotPatAGFF\_taux\_trB)
- NumericVector \_ (CotSalAGFF\_taux\_trB)

## Attributs publics

- int **Id**
- int **annee**
- int **age**
- int **statut**
- double **partavtprimo**
- double **salaire**
- double **salaire\_net**
- double **cotis\_retr**
- double **cotis\_retr\_patr**

## 5.3.1 Description détaillée

La structure [Cotisations](#) contient les différents taux de cotisations (maladie et retraite) par année.

La classe [Cotisations](#) contient les éléments suivants :

- attributs :
  - l'année
  - l'identifiant de l'individu
  - l'âge de l'individu au cours de l'année
  - le statut de l'individu au cours l'année
  - la fraction de l'année précédant la première liquidation
  - le salaire de l'individu au cours de l'année
  - le salaire net de l'individu au cours l'année
  - le montant des cotisations retraite (part salariale) prélevées sur les rémunérations de l'individu au cours de l'année
  - le montant des cotisations retraite (part employeur) prélevées sur les rémunérations de l'individu au cours de l'année
- méthodes : un constructeur

La documentation de cette structure a été générée à partir des fichiers suivants :

- src/[Cotisations.h](#)
- src/[CotisationsTRI.h](#)

## 5.4 Référence de la structure CotisationsTRI

La structure [CotisationsTRI](#) contient la fonction [CotisationsTRI](#) qui calcule l'intégralité des cotisations d'un individu.

```
#include <CotisationsTRI.h>
```

## Fonctions membres publiques

- **CotisationsTRI** ([Indiv](#) &X, int age, [Macro](#) &s, [Cotisations](#) &c, [OptionsTRI](#) &opt)



## Attributs publics

```

— Indiv & X
— Macro & s
— Cotisations & c
— OptionsTRI & options_tri
— double cot_patr_Retr_rg = 0
— double cot_patr_Retr_fpe = 0
— double cot_patr_Retr_fpt = 0
— double cot_patr_Retr_ind = 0
— double cot_patr_Retr_alleg = 0
— double cot_patr_arrco = 0
— double cot_patr_agirc = 0
— double cot_patr_ASF = 0
— double cot_patr_AGFF = 0
— double cot_sal_Retr_arrco = 0
— double cot_sal_Retr_agirc = 0
— double cot_sal_Retr_rg = 0
— double cot_sal_Retr_AGFF = 0
— double cot_sal_Retr_ASF = 0
— double cot_sal_Retr_fp = 0
— double cot_sal_Retr_ind = 0
— double csg_sal_fsv = 0
— double contrib_retr_actif = 0
— double transferts_Unedic_Agirc_Arrco = 0
— double cot_rg = 0
— double cot_tot = 0

```

La documentation de cette structure a été générée à partir du fichier suivant :

— src/[CotisationsTRI.h](#)

## 5.5 Référence de la structure CStructSexeAge

La structure [CStructSexeAge](#) contient la série des coefficients correcteurs utilisés lors de la simulation des salaires, pour les femmes et les hommes.

```
#include <Salaires.h>
```

## Fonctions membres publiques

```

— NumericVector _ (CORREC_HOMMES)
  Coefficient correctif appliqué au revenu d'activité des hommes.
— NumericVector _ (CORREC_FEMMES)
  Coefficient correctif appliqué au revenu d'activité des femmes.

```

## 5.5.1 Description détaillée

Voir [\[aubert2012disparites\]](#) , page 34, pour plus de détails.

La documentation de cette structure a été générée à partir du fichier suivant :

— src/[Salaires.h](#)

## 5.6 Référence de la classe DroitsRetr

Droits directs par régime.

## Fonctions membres publiques

- **DroitsRetr** (const **Indiv** &**X**, **Leg** &leg, double f\_agetest)  
*Contruit un nouvel objet DroitsRetr pour un individu, une législation et un âge test donnés.*
- double **duree\_trim** (const vector< int > &statuts\_cpt, int anneLimite=**options**->AN\_MAX)  
*Calcule la durée en trimestres passée dans un ou plusieurs statuts.*
- void **durees\_base** ()  
*Calcul des durées de base (durées cotisations et AVPF)*
- void **durees\_majo** ()  
*Calcul des durées majorées.*
- void **DecoteSurcote** ()  
*Calcule les durées de décote et surcote et les taux de liquidation.*
- int **AgeMin** ()  
*Indique si l'individu a atteint l'âge minimal d'ouverture des droits.*
- double **AgeMax** ()  
*Retourne l'âge de mise à la retraite d'office de l'individu.*
- void **SecondLiq** ()  
*Calcule les pensions hors FP lors de la seconde liquidation (appel de liq\_privée et liq\_public)*
- void **Liq** ()  
*Calcule les pensions aux différents régimes (appel de liq\_privée et liq\_public)*

## Méthodes de calculs intermédiaires

- double **calc\_sam** (double \*spc\_begin, double \*spc\_end)  
*calcul du SAM à partir des salaires portés au compte.*
- double **calc\_sam** (double \*spc\_begin, double \*spc\_end, int duree\_calc)  
*calcul du SAM à partir des salaires portés au compte pour les n meilleures années*
- void **SalBase** (int AnneeRefAnticip=9999)  
*Calcul des SAM et du salaire de référence pour la FP.*
- void **Points** (int AnneeRefAnticip=9999)  
*Calcul des décomptes de points ARRCO et AGIRC.*
- void **MinCont** (int AnneeRefAnticip=9999)  
*calcul du minimum contributif au RG et au RSI (appelée par AppliqueBonif)*
- double **MinGaranti** ()  
*calcul du minimum garanti (appelée par AppliqueBonif)*
- void **AppliqueMin** ()  
*Application du minimum contributif et du minimum garanti.*
- void **AppliqueBonif** ()  
*Application des majorations de pensions pour enfants à l'ensemble des pensions de droit direct.*
- void **LiqPrive** (int AnneeRefAnticip=9999)  
*Calcul de la pension au RG, au RSI, à l'Arrco et à l'Agirc (appelée par Liq ou SecondLiq)*
- void **LiqPublic** ()  
*Calcul de la pension FP (appelée par Liq)*

## Attributs publics

- const **Indiv** & **X**  
*Référence à l'individu correspondant aux droits calculés.*
- **Leg** & **l**  
*Référence à l'objet Leg utilisé pour le calcul des droits.*
- double **agetest**  
*Âge fin de mois testé*
- double **agefin\_totliq** = 0  
*Âge fin de mois de liquidation totale.*
- double **agefin\_primoliq** = 0  
*Âge fin de mois de primo-liquidation.*
- int **ageliq** = 0  
*Âge à la liquidation de la totalité des droits (âge entier au 31/12 de l'année)*
- int **ageprimoliq** = 0  
*Âge à la primo-liquidation des droits (âge entier au 31/12 de l'année)*
- double **duree\_cho** = 0  
*Durée cumulée au chômage (au RG)*
- double **duree\_PR** = 0  
*Durée cumulée en pré-retraite.*
- double **duree\_inv** = 0  
*Durée cumulée en invalidité*
- double **duree\_snat** = 0

- *Durée cumulée service national.*  
double `duree_avpf` = 0
- *Durée cumulée passée en AVPF.*  
double `duree_emprg` = 0
- *Durée cumulée avec cotisation au RG.*  
double `duree_fp` = 0
- *Durée cumulée à la fonction publique.*  
double `duree_fpa` = 0
- *Durée cumulée à la fonction publique (catégorie active)*  
double `duree_fps` = 0
- *Durée cumulée à la fonction publique (catégorie sédentaire)*  
double `duree_ag` = 0
- *Durée cotisée avec affiliation à l'Agirc.*  
double `duree_emp` = 0
- *Durée totale en emploi.*  
double `duree_in` = 0
- *Durée cotisée au régime des indépendants.*  
double `duree_rg` = 0
- *Durée cotisée au régime général.*  
double `duree_ag_ar` = 0
- *Durée d'affiliation au régime complémentaire des salariés du privé unifié*  
double `duree_ar` = 0
- *Durée d'affiliation à l'Arrco.*  
double `duree_tot` = 0
- *Durée totale cotisée.*  
double `dureecotmin_tot` = 0
- *Durée totale cotisée, servant pour le calcul des minimum contributif et garanti.*  
double `dureecotdra_tot` = 0
- *Durée totale cotisée, servant pour le calcul de l'éligibilité à la retraite anticipée pour carrière longue.*  
double `duree_rg_maj` = 0
- *Durée RG incluant AVPF et MDA (selon options)*  
double `duree_fp_maj` = 0
- *Durée FP incluant MDA éventuelle (selon options)*  
double `duree_in_maj` = 0
- *Durée indépendant incluant MDA éventuelle (selon options)*  
double `duree_tot_maj` = 0
- *Durée totale incluant AVPF et MDA (selon options)*  
double `durdecote_fp` = 0
- *Durée (en année) déterminant la surcote à la fp (durée = 0 pour un départ sans surcote)*  
double `dursurcote_fp` = 0
- *Durée (en année) déterminant la décote à la fp (durée = 0 pour un départ sans décote)*  
double `durdecote_rg` = 0
- *Durée (en année) déterminant la décote au rg et pour les indépendants (durée = 0 pour un départ sans décote)*  
double `dursurcote_rg` = 0
- *Durée (en année) déterminant la surcote au rg et pour les indépendants (durée = 0 pour un départ sans surcote)*  
double `tauxliq_fp` = 0
- *taux de liquidation au RG (=1 en cas de départ sans décote ni surcote) et au régime des indépendants*  
double `tauxliq_rg` = 0
- *taux de liquidation à la FP (=1 en cas de départ sans décote ni surcote)*  
double `tauxliq_ar` = 0
- *taux de liquidation à l'ARRCO et à l'AGIRC (=1 en cas de départ sans décote ni surcote)*  
double `majo_min_rg` = 0
- *montant de la majoration de pension au RG liée à l'application du minimum contributif (montant différentiel inclus dans \$pension\_rg)*  
double `majo_min_in` = 0
- *montant de la majoration de pension chez les indépendants liée à l'application du minimum contributif (montant différentiel inclus dans \$pension\_rg)*  
double `majo_min_fp` = 0
- *montant de la majoration de pension à la FP liée à l'application du minimum garanti (montant différentiel inclus dans \$pension\_rg)*  
double `majo_3enf_rg` = 0
- *montant de la majoration de pension totale (droits directs + réversion) pour 3 enfants au RG*  
double `majo_3enf_ar` = 0
- *montant de la majoration de pension totale (droits directs + réversion) pour 3 enfants à l'ARRCO*  
double `majo_3enf_ag` = 0
- *montant de la majoration de pension totale (droits directs + réversion) pour 3 enfants à l'AGIRC*  
double `majo_3enf_in` = 0
- *montant de la majoration de pension totale (droits directs + réversion) pour 3 enfants chez les indépendants*  
double `majo_3enf_fp` = 0

- *montant de la majoration de pension totale (droits directs + réversion) pour 3 enfants à la FP*  
double `majo_3enf_ag_ar` = 0
- *montant de la majoration de pension totale (droits directs + réversion) pour 3 enfants à l'ARRCO*  
double `min_cont` = 0
- *minimum contributif au RG*  
double `min_cont_in` = 0
- *minimum contributif au RSI*  
double `min_garanti` = 0
- *minimum garanti*  
double `sr_fp` = 0
- *saire de référence pour le calcul de la pension FP*  
double `sam_rg` = 0
- *SAM servant au calcul de la pension RG.*  
double `sam_in` = 0
- *SAM pour le calcul de la pension d'indépendant.*  
double `sam_rgin` = 0
- *SAM calculé sur l'ensemble des revenus d'activité annuels au RG et chez les indépendants.*  
double `sam_uni` = 0
- *SAM calculé sur l'ensemble des revenus d'activité (tous régimes), selon les règles du RG.*  
double `points_arrco` = 0
- *cumul points ARRCO*  
double `points_agirc` = 0
- *cumul points AGIRC*  
double `ntp_FP` = 0
- *nombre de points ARRCO "fictifs", calculé sur la base des salaires des fonctionnaires (ne seront pris en compte que si le fonctionnaire n'atteint pas la condition de fidélité)*  
double `points_agirc_arrco` = 0
- *cumul points régime unifié AGIRC-ARRCO (cf. accord de 2015)*  
double `maj_points_arrco` = 0
- *cumul des points ARRCO dus pour majoration au titre des 3 enfants*  
double `maj_points_agirc` = 0
- *cumul des points AGIRC dus pour majoration au titre des 3 enfants*  
double `maj_points_agirc_arrco` = 0
- *cumul des points régime unifié AGIRC-ARRCO pour la majoration au titre des 3 enfants*  
double `VFU_rg` = 0
- *montant du versement forfaitaire unique au RG*  
double `VFU_ar` = 0
- *montant du versement forfaitaire unique à l'Arrco*  
double `VFU_ag` = 0
- *montant du versement forfaitaire unique à l'Agirc*  
double `pliq` = 0
- *pension totale à la liquidation*  
double `pension_fp` = 0
- *pension fonction publique*  
double `pension_rg` = 0
- *pension régime général*  
double `pension_ar` = 0
- *pension arrco*  
double `pension_ag` = 0
- *pension agirc*  
double `pension_ag_ar` = 0
- *pension des complémentaires unifiées*  
double `pension_in` = 0
- *pension d'indépendant*  
double `pension` = 0
- *pension directe totale (somme des précédentes)*  
double `taux_prorat_rg` = 0
- *taux de proratisation au RG*  
double `taux_prorat_in` = 0
- *taux de proratisation au RSI*  
double `taux_prorat_fp` = 0
- *taux de proratisation FP*  
double `coeffTemp` = 1.0
- *coefficient temporaire Agirc Arrco*  
double `ageAnnulCoeffTemp` = 0.0
- *âge d'annulation du coefficient temporaire Agirc Arrco.*  
double `tp` = 0
- *indicatrice taux plein*  
bool `primoliq` = false
- *Indicatrice primo-liquidation.*

- bool `liq` = false  
*Indicatrice liquidation totale.*
- bool `indic_mc` = false  
*Indicatrice minimum contributif au RG.*
- bool `indic_mg` = false  
*Indicatrice minimum garanti.*
- bool `indic_mc_in` = false  
*Indicatrice minimum contributif au RSI.*
- bool `dar` = false  
*Indicatrice départ anticipé pour carrière longue.*
- int `type_liq` = `liq_non`  
*Motif de liquidation (par défaut, l'individu n'a pas liquidé)*
- int `t` = 0  
*année testée*
- int `age` = 0  
*age au 31/12 testé*
- double `datetest` = 0  
*date testée*

#### Méthodes renvoyant des informations supplémentaires

- const vector< int > `statuts_cho` = {`S_CHO`, `S_CHOBIT`, `S_CHONONBIT`}
- const vector< int > `statuts_pr` = {`S_PR`}
- const vector< int > `statuts_inv` = {`S_INVALID`, `S_INVALIDRG`}
- const vector< int > `statuts_SNat` = {`S_SNAT`}
- const vector< int > `statuts_Avpf` = {`S_AVPF`}
- const vector< int > `statuts_emprg` = {`S_NC`, `S_NONTIT`, `S_CAD`}
- const vector< int > `statuts_fpa` = {`S_FPAE`, `S_FPAAE`, `S_FPATH`, `S_FPAATH`}
- const vector< int > `statuts_fps` = {`S_FPSE`, `S_FPSTH`, `S_INVALIDFP`}
- const vector< int > `statuts_cad` = {`S_CAD`}
- const vector< int > `statuts_ind` = {`S_IND`}
- const vector< int > `statuts_ind_assimil` = {`S_IND`, `S_INVALIDIND`}
- double `dernier_salaire_brut` ()  
*Renvoie le dernier salaire brut.*
- double `age_dernier_emploi` ()  
*Renvoie l'age au dernier emploi.*
- double `pension_age` (int `age`, int regime)  
*Renvoie la pension perçue à un âge et à un régime donnés (tout régime par défaut)*
- double `duree_maj` (int regime)  
*Renvoie la durée majorée dans le régime.*
- double `duree` (int regime)  
*Renvoie la durée, hors majorations, dans le régime.*
- double `partavtliq` (int `age`)  
*Renvoie la part de l'année précédant la liquidation totale en fonction de l'âge.*
- double `partavtprimo` (int `age`)  
*Renvoie la part de l'année précédant la primo-liquidation en fonction de l'âge.*
- double `pension_prorat` (double `pension`, double `daterevalo`, double `RevaloReg`, double `partavtliq`=0)  
*Renvoie la pension proratisée en fonction de la date de revalorisation.*
- double `pension_prorat` (int `age`, int regime)  
*Renvoie la pension proratisée par âge et par régime.*
- double `duree_emp_reg` (int regime)  
*Renvoie la durée en emploi dans le régime.*
- double `VFU` (int regime)  
*Renvoie le montant du VFU dans le régime.*
- double `min_reg` (int regime)  
*Renvoie, pour un régime donné, une indicatrice valant 1 en cas de bénéfice du minimum de pension, et 0 sinon.*
- double `majo_3enf` (int regime)  
*Renvoie, pour un régime donné, le montant de la majoration accordée au titre des enfants. Lorsque regime=REG←→\_TOT, comprend également le montant des minima de pensions.*
- double `majo_min` (int regime)  
*Renvoie, pour un régime donné, le montant de la majoration accordée au titre du minimum de pension.*
- double `taux_liq` (int regime)  
*Renvoie, pour un régime donné, le coefficient de décote/surcote.*
- double `taux_prorat` (int regime)  
*Renvoie, pour un régime donné, le coefficient de proratisation.*
- double `pensionliq` (int regime)  
*Renvoie, pour un régime donné, la pension annualisée à la liquidation.*
- double `age_liq` (int regime)  
*Renvoie, pour un régime donné, l'âge au 31/12 de l'année de liquidation.*

- double [derSalNet](#) ()  
*Renvoie le salaire brut moyen des cinq années précédant la liquidation.*
- int [ageFinEmp](#) ()  
*Renvoie l'âge au dernier emploi.*
- int [ageFinAct](#) ()  
*Renvoie l'âge au dernier statut d'activité*
- void [coeffTempo](#) ()  
*Calcul du coefficient temporaire de décote ou surcote pour les régimes complémentaires suite à l'accord Agirc-Arcco d'octobre 2015.*

### 5.6.1 Description détaillée

Cette classe détermine les droits directs à la liquidation pour un individu, une législation et un âge de liquidation donnés.

L'objet est créé dans la fonction SimDir de la classe [Retraite](#) et utilisé, entre autres, par les fonctions TestLiq et TestSecondLiq du fichier [OutilsComp.h](#). Cette classe stocke les éléments constitutifs des droits directs (montant à la liquidation, majorations, durée validée, coefficient de décote/surcote, ...).

Un objet de cette classe est créé pour chaque individu à chaque âge testé. Un objet (ou deux si la liquidation a lieu en deux étapes) est ensuite conservé en cas de liquidation.

#### Calcul des pensions de droits directs des régimes de base (régimes en annuités)

Les pensions de droit direct des régimes de base sont calculées pour trois régimes : le RG, la SSI et la FP (SRE et CNRACL). La pension à la liquidation est le produit de trois termes : le taux de liquidation, le coefficient de proratisation et le salaire de référence (salaire annuel moyen, SAM, pour le RSI et le RG, dernier salaire pour la FP).

$$\text{Pension} = \text{SAM} \times \text{tauxliq} \times \text{taux\_prorat}$$

- SAM = [Salaire](#) annuel moyen = moyenne des salaires perçus pendant les meilleures années d'activité (10 à 25 meilleures années selon l'année de naissance), dans la limite du PSS, pour les salaires reportés à partir de 2005.
- tauxliq = [Taux](#) de liquidation = fixé en fonction de la durée d'assurance, tous régimes confondus. Le taux maximal, dit « taux plein », est de 50% (75% pour la fonction publique).
- taux\_prorat = Coefficient de proratisation. C'est le rapport entre le nombre de trimestres d'assurance (cotisés ou non) et la durée de référence de 150 à 164 trimestres, en fonction de l'année de naissance (durée d'assurance requise à partir de la génération 1948).

Les durées d'assurance sont calculées par la fonction durees\_base, et les durées majorées par durees\_majo. Les taux de liquidation sont calculés par la fonction DecoteSurcote. Le salaire de référence est calculé par la fonction SalBase.

Cette formule de base est en outre modifiée par le minimum contributif (ou garanti pour la FP) et majorée d'une bonification pour les mères de 3 enfants.

Il existe des minima de pension (minimum contributif au RG et au RSI, minimum garanti à la FP). Le montant du minimum est calculé au prorata de la durée validée (et peut donc être très faible) : le dispositif ne relève donc pas {stricto sensu} d'une logique de revenu minimum. La comparaison pension/minimum de pension est faite à la liquidation par la fonction AppliqueMin.

## Calcul des pensions de droits directs des régimes complémentaires (régimes en points)

A la liquidation, la pension est calculée comme le produit suivant : [Taux](#) de liquidation \* nombre de points \* valeur de service du point

Le taux de liquidation dépend de

- l'âge à la liquidation
- de la durée validée dans les régimes de base Un taux d'abattement, équivalent à la décote à l'AGIRC ARCO, est ensuite appliqué.

Le nombre de points dépend de la carrière. Au cours de celle-ci, l'acquisition de points se fait via les cotisations selon la formule : Nombre de points = (Revenu salarial \* taux de cotisation) / valeur d'achat du point.

Un taux d'appel fixe la part de cotisations non génératrices de droits.

Les droits à la liquidation sont calculés en quatre étapes :

- calcul des durées de base ([durees\\_base](#))
- calcul des durées majorées ([durees\\_majo](#)). Cette étape détermine l'âge d'ouverture des droits.
- calcul de la décote ou de la surcote ([DecoteSurcote](#)). Permet le calcul des taux de liquidation.
- calcul des pensions à liquidation ([Liq](#) et [SecondLiq](#)). Ces dernières étapes appellent les méthodes suivantes :
  - [SalBase](#) : Calcul du SAM
  - [Points](#) : Calcul des points à l'AGIRC et à l'ARRCO
  - [LiqPublic](#) : Calcul de la pension FP
  - [LiqPrive](#) : Calcul de la pension au RG, au RSI, ainsi qu'à l'Arrco et à l'Agirc
  - [AppliqueMin](#) : Application des minima de pensions
  - [AppliqueBonif](#) : Application de la bonification pour enfants

Exemple d'utilisation :

```
Indiv& X = M->pop[12];
Leg l = Leg(X, 60, min(X.anaiss+60,options->anLeg));

DroitsRetr dr = DroitsRetr(X,l,60+3/12.0);
if(dr.AgeMin()) {
    dr.liq();
    Rcout << "L'individu 12 peut liquider avec une pension : "
    Rcout << dr.pension;
}
else {
    Rcout << "L'individu 12 n'a pas atteint l'âge min.pour liquider";
}
```

## 5.6.2 Documentation des fonctions membres

### 5.6.2.1 AgeMax()

```
double AgeMax ( )
```

Nécessite que [durees\\_base](#), [durees\\_majo](#) et [DecoteSurcote](#) aient été appelées au préalable.

Remarque : ne modifie aucune variable.

### 5.6.2.2 AgeMin()

```
int AgeMin ( )
```

#### Renvoie

retourne true si l'individu a atteint l'âge minimal d'ouverture des droits, false sinon.

Cette méthode indique si l'individu a atteint l'âge minimal d'ouverture des droits. En particulier, elle recalcule l'âge minimal d'ouverture des droits (RG et FP) en cas de départ anticipé pour carrière longue (sauf si option.NoRetr↔AntCarrLongues).

Remarque : fait appel aux méthodes [durees\\_base](#), [durees\\_majo](#) et [DecoteSurcote](#)

### 5.6.2.3 AppliqueBonif()

```
void AppliqueBonif ( )
```

Cette fonction applique les majorations de pensions pour enfants à l'ensemble des pensions, de droit direct ou dérivé.

Comme cette fonction n'est appelée que juste après le calcul des avantages principaux de droit direct OU dérivé, on utilise les variables \$agefin\_primoliq, \$agefin\_totliq et \$agerevliq pour savoir à quelles pensions la bonification doit être appliquée (et, a contrario, auxquelles elle l'a déjà été au cours d'étapes précédentes).

Remarque : on utilise notamment le fait que, pour les droits directs, la fonction est appelée APRES le calcul du montant de pension (hors bonification) mais AVANT la définition des variables d'âge fin.

Remarque : ne sont pas encore programmées ici :

- la modification du taux de bonification à l'Agirc et à l'Arrco pour les points acquis après 2012
- la proratisation de la limite de 1000 euros par an pour la bonification Agirc et Arrco après 2012

Fonction AppliqueBonif

Cette fonction applique les majorations de pensions pour enfants à l'ensemble des pensions, de droit direct ou dérivé.

Comme cette fonction n'est appelée que juste après le calcul des avantages principaux de droit direct OU dérivé, on utilise les variables \$agefin\_primoliq, \$agefin\_totliq et \$agerevliq pour savoir à quelles pensions la bonification doit être appliquée (et, a contrario, auxquelles elle l'a déjà été au cours d'étapes précédentes).

Remarque : on utilise notamment le fait que, pour les droits directs, la fonction est appelée APRES le calcul du montant de pension (hors bonification) mais AVANT la définition des variables d'âge fin.

### 5.6.2.4 AppliqueMin()

```
void AppliqueMin ( )
```

Fonction appelée par liq ou secondLiq. Cette fonction applique les dispositifs de minimum contributif et de minimum garanti à l'ensemble des pensions, le calcul des avantages principaux de droits directs hors min ayant été réalisé au préalable.

L'écèlement des minima est programmé selon une règle tous régimes, prise dans la LFSS 2009 pour une application à partir de 2012. PLFSS 2009 : la surcote s'apprécie après le calcul du minimum contributif et garanti



#### 5.6.2.5 `calc_sam()` [1/2]

```
double calc_sam (
    double * spc_begin,
    double * spc_end )
```

Calcul du salaire annuel moyen à partir des salaires versés au compte. Le calcul prend en compte l'ensemble des salaires, contrairement à [calc\\_sam](#)

#### 5.6.2.6 `calc_sam()` [2/2]

```
double calc_sam (
    double * spc_begin,
    double * spc_end,
    int duree_calc )
```

Calcul du salaire annuel moyen à partir des salaires versés au compte. Le calcul ne prend en compte que les n meilleures années, contrairement à [calc\\_sam](#)

#### 5.6.2.7 `coeffTempo()`

```
void coeffTempo ( )
```

#### 5.6.2.8 `DecoteSurcote()`

```
void DecoteSurcote ( )
```

Calcule les durées de décote et surcote et les taux de liquidation (les durées de décote et surcote sont exprimées en année, mais peuvent prendre des valeurs arrondies à 0.25 près, c'est-à-dire au trimestre près)

Cette fonction nécessite d'avoir lancé auparavant les fonctions `durees_majo` et `durees_base`

Cette méthode renseigne les variables suivantes :

- `dursurcote`, [durdecote\\_fp](#), [dursurcote\\_fp](#), [durdecote\\_rg](#)
- [tauxliq\\_rg](#), [tauxliq\\_fp](#), [tauxliq\\_ar](#)

Le calcul tient compte de plus :

- de l'annulation de la décote en cas d'invalidité
- de l'annulation de la décote en cas de liquidation au titre de l'inaptitude HORS INVALIDITE (proba etat inapt = f(X.k) )
- du taux de liquidation (coefficient d'abattement) dans les régimes complémentaires ARRCO et AGIRC (suppression de la décote si taux plein au RG)

#### 5.6.2.9 `duree_trim()`

```
Méthode duree_trim (
    const vector< int > & stats_cpt,
    int anneLimite = options->AN\_MAX )
```

#### Paramètres

<code>stats_cpt</code>	: vecteur des statuts comptabilisés
------------------------	-------------------------------------

**Renvoie**

durée en années (multiple de 0.25)

Complément de la fonction [durees\\_base](#), en fonction de l'âge de test. Calcule les durées passées par l'individu X dans un ou plusieurs états, jusqu'à l'âge test. Les durées pour l'année civile de l'âge en argument sont arrondies au nombre de trimestres civils jusqu'au 1er du mois suivant cet âge.

**5.6.2.10 durees\_base()**

```
Méthode durees_base ( )
```

Calcule les durées cotisées dans les différents régimes et la durée AVPF pour l'individu X.

Cette méthode calcule : [duree\\_cho](#) , [duree\\_PR](#) , [duree\\_inv](#) , [duree\\_snat](#) , [duree\\_avpf](#) , [duree\\_emprg](#) , [duree\\_fpa](#) , [duree\\_fps](#) , [duree\\_ag](#) , [duree\\_emp](#) et [duree\\_in](#).

**5.6.2.11 durees\_majo()**

```
void durees_majo ( )
```

Repart des durées calculées par [durees\\_base](#), et calcule les durées majorées ([duree\\_rg\\_maj](#) , [duree\\_fp\\_maj](#) , [duree\\_in\\_maj](#) , [duree\\_tot\\_maj](#)) prenant en compte la Mda selon les options (`option.NoMDA`).

**5.6.2.12 Liq()**

```
void Liq ( )
```

Simule la liquidation à l'âge courant. Cette fonction n'a pas de valeur de retour. Elle modifie les variables `pension_rg`, `pension_fp` ... et `pension` et `ageliq`.

Remarque (09/12/2011) : Pour permettre la programmation de la liquidation en deux temps des polyaffiliés public/-privé, deux sous-modules `LiqPrive` et `LiqPublic` ont été créés, reprennant la partie de `Liq` correspondante. Ces deux fonctions calculent l'avantage principal de droit direct, HORS MINIMUM et hors bonification. Il faut donc ensuite appeler les fonctions qui appliquent ces deux types de majoration.

**5.6.2.13 MinCont()**

```
void MinCont (
    int AnneeRefAnticip = 9999 )
```

Calcul du montant attribuable au titre du minimum contributif. Suppose que les durées de cotisation sont connues, nécessite donc des appels préalables de `DurBase` et `DurMajo` (source, Doc N°4, séance COR du 27 juin 2006)

**5.6.2.14 MinGaranti()**

```
double MinGaranti ( )
```

Calcul du montant attribuable au titre du minimum garanti : période transitoire entre 2004 et 2013 avec application progressive de la réforme. Suppose que `duree_fp` ait été renseignée, i.e. un appel préalable de [durees\\_base](#).

Fonction `MinGaranti` Calcul du montant attribuable au titre du minimum garanti : période transitoire entre 2004 et 2013 avec application progressive de la réforme. Suppose que `duree_fp` ait été renseignée, i.e. un appel préalable de `DurBase`.

### 5.6.2.15 Points()

```
void Points (
    int AnneeRefAnticip = 9999 )
```

Calcule les nombres de points accumulés dans les régimes ARRCO et AGIRC et stocke les résultats dans les variables [points\\_arrco](#), [points\\_agirc](#).

La fonction intègre le calcul des points gratuits attribués au titre des périodes de chômage et de préretraite, ainsi que les points gratuits de GMP (garantie minimale de points) à l'Agirc

### 5.6.2.16 SalBase()

```
void SalBase (
    int AnneeRefAnticip = 9999 )
```

Calcule les salaires servant de base au calcul des pensions de base, ie. les SAM pour le RG et le RSI et le salaire de référence pour la FP. Les résultats sont stockés dans [sam\\_rg](#), [sam\\_in](#) et [sr\\_fp](#). De plus, sont mis à jour les totaux [samuni\\_rgin](#) et [samuni](#).

Cette fonction prend en compte pour les législations 1993 et postérieures (I.Legsam), les changements sur le nombre d'années pris en compte dans le calcul du SAM.

Pour les législations postérieures à 2004, les polyaffiliés voient leur calcul de durée de prise en compte du {sam} modifiée.

Pour la fonction publique, on tient compte de l'option FPCarTot. Si cette option est utilisée, on calcule le spc. comme pour le privé (en retenant le nombre d'année en vigueur pour le {rg}).

— SpcToutesAnnees : on ne retient plus les 25 meilleurs années mais toutes les années.

Prise en compte de l'AVPF via les options NoAVPF et NoSpcAVPF.

Possibilité de calculer un sam unifié RG-IN (option SAMRginUnique), RG-IN-FP (option SAMUnique), ou un sam distinct par régime (option SAMSepare).

### 5.6.2.17 SecondLiq()

```
void SecondLiq ( )
```

Calcule les pensions hors FP lors de la seconde liquidation (appel de liq\_privée et liq\_public) Permet la liquidation finale de tous les droits pour les personnes qui liquident en 2 temps (cas des polyaffiliés public-privé, dont le niveau de pension FP (seule) suffit pour déterminer leur âge de départ à la retraite, mais qui n'ont pas encore atteint l'âge d'ouverture des droits dans le privé au moment de la liquidation dans le public)

Par hypothèse, on considère que la liquidation finale a lieu dès que la possibilité de partir au taux plein dans le privé est atteinte.

La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/DroitsRetr.h](#)
- [src/DroitsRetr.cpp](#)

## 5.7 Référence de la structure Ech

[Ech](#) contient les caractéristiques invariantes au cours du temps des individus.

```
#include <Simulation.h>
```

## Fonctions membres publiques

- IntegerVector \_ (Id)  
*Identifiant de l'individu.*
- IntegerVector \_ (sexe)  
*Sexe de l'individu.*
- IntegerVector \_ (anaiss)  
*Année de naissance de l'individu.*
- IntegerVector \_ (moisnaiss)  
*Mois de naissance de l'individu (identifiant modulo 12)*
- IntegerVector \_ (findet)  
*Âge de fin d'étude de l'individu.*
- IntegerVector \_ (neFrance)  
*Indicatrice du fait d'être né en France.*
- IntegerVector \_ (emigrant)  
*Indicatrice du fait d'avoir émigré*
- IntegerVector \_ (typeFP)  
*Type de fonction publique : FPE ou FPTH.*
- IntegerVector \_ (peudip)  
*Indicatrice d'être peu diplômé pour l'imputation de l'état de santé*
- IntegerVector \_ (tresdip)  
*Indicatrice d'être très diplômé pour l'imputation de l'état de santé*
- IntegerVector \_ (dipl)  
*Niveau de diplôme en cinq catégories.*
- NumericVector \_ (taux\_prim)  
*Taux de prime (pour les fonctionnaires)*
- NumericVector \_ (k)  
*Paramètre pour l'option de départ uinst (utilité instantannée cible)*

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Simulation.h](#)

## 5.8 Référence de la structure Emp

[Emp](#) contient les statuts sur le marché du travail et les revenus d'activité

```
#include <Simulation.h>
```

## Fonctions membres publiques

- IntegerVector \_ (Id)  
*Identifiant de l'individu.*
- IntegerVector \_ (age)  
*Âge.*
- IntegerVector \_ (statut)  
*Statut sur le marché du travail.*
- NumericVector \_ (salaire)  
*Revenu annuel d'activité*

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Simulation.h](#)

## 5.9 Référence de la structure EqSalaires

La structure [EqSalaires](#) contient les coefficients des variables explicatives des équations de salaires, pour un sexe et un secteur donnés.

```
#include <Salaires.h>
```

## Fonctions membres publiques

- double \_ (Intercept)
- double \_ (E\_1er)
- double \_ (E\_0)
- double \_ (E)
- double \_ (E2)
- double \_ (FR\_E)
- double \_ (FR\_E2)
- double \_ (FR\_E\_0)
- double \_ (FR\_D2)
- double \_ (D)
- double \_ (D2)
- double \_ (FR\_D)
- double \_ (D\_0)
- double \_ (FR\_D\_0)
- double \_ (FR)
- double \_ (log\_fp)
- double \_ (Sig2pente\_indiv)
- double \_ (corr\_indiv)
- double \_ (Sig2\_indiv)
- double \_ (Sig2\_res)

### 5.9.1 Description détaillée

Pour la signification des libellés des coefficients, voir [\[aubert2012disparites\]](#) , pp 30-35.

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Salaires.h](#)

## 5.10 Référence de la structure EqSante

[EqSante](#) contient les coefficients pour les équations de santé (adl, iadl, gali).

```
#include <Simulation.h>
```

## Fonctions membres publiques

- double \_ (Intercept)  
*Coefficient devant la constante.*
- double \_ (couple)  
*Coefficient devant le fait d'être en couple.*
- double \_ (enf4plus)  
*Coefficient devant le fait d'avoir 4 enfants ou plus.*
- double \_ (enf01)  
*Coefficient devant le fait d'avoir 0 ou 1 enfant.*
- double \_ (tresdip)  
*Coefficient devant le fait d'être très diplômé par rapport à sa génération et son sexe.*
- double \_ (peudip)  
*Coefficient devant le fait d'être peu diplômé par rapport à sa génération et son sexe.*
- double \_ (agebis)  
*Coefficient devant l'âge moins 50. Note : dans les simulations, un âge encore modifié peut être employé pour tenir compte des scénarios d'évolution des incapacités, voir [\[evrsij\]](#) .*
- double \_ (agebis2)  
*Coefficient devant l'âge moins 50, au carré.*
- double \_ (agebis3)  
*Coefficient devant l'âge moins 50, au cube.*
- double \_ (sante)  
*Coefficient devant l'état de santé, quand pertinent.*

## 5.10.1 Description détaillée

Il s'agit des paramètres nécessaires à l'estimation des états de santé (à partir de 50 ans) décrits par **[evrsi]** : les restrictions dans les activités de la vie quotidienne (adl), les restrictions dans les activités instrumentales de la vie quotidienne (iadl), et l'indicateur du gali.

Il y a trois types d'équations (dépendant du sexe) : des équations de prévalence (pour attribuer un état de santé à la date initiale ou à 50 ans), des équations d'incidence (pour faire varier cet état de santé, dépendant de l'état de santé deux ans avant), et des équations de mortalité (dépendant de l'état de santé deux ans avant).

Source : **[evrsi]** sur la partie française de l'enquête SHARE.

La documentation de cette structure a été générée à partir du fichier suivant :

— src/[Simulation.h](#)

## 5.11 Référence de la structure EqTrans

[EqTrans](#) contient les équations de transitions sur le marché du travail.

```
#include <Transitions.h>
```

## Fonctions membres publiques

- double `_ (a55)`  
Coefficient devant l'indicatrice d'âge égale à 55 ans  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (a56)`  
Coefficient devant l'indicatrice d'âge égale à 56 ans  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (a57)`  
Coefficient devant l'indicatrice d'âge égale à 57 ans  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (a58)`  
Coefficient devant l'indicatrice d'âge égale à 58 ans  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age)`  
Coefficient devant l'âge  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age_a)`  
Coefficient devant l'âge l'année suivant la fin des études  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age_b)`  
Coefficient devant l'âge, de 1 à 3 ans après la fin des études  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age2)`  
Coefficient devant l'âge au carré  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age2_a)`  
Coefficient devant l'âge l'année suivant la fin des études, au carré  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age2_b)`  
Coefficient devant l'âge, de 1 à 3 ans après la fin des études, au carré  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age3)`  
Coefficient devant l'âge au cube  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age3_a)`  
Coefficient devant l'âge l'année suivant la fin des études, au cube  
Source : estimé à l'occasion de **[bachelet2014biographies]**.
- double `_ (age3_b)`  
Coefficient devant l'âge, de 1 à 3 ans après la fin des études, au cube  
Source : estimé à l'occasion de **[bachelet2014biographies]**.

- double \_ (dur\_emp)  
Coefficient devant le nombre d'années en emploi depuis la fin de la formation initiale  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (dur\_statut)  
Coefficient devant le nombre d'années consécutives dans le dernier statut  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (durchom)  
Coefficient devant la durée au chômage dans les 5 dernières années  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (durlNA)  
Coefficient devant la durée en inactivité dans les 5 dernières années  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (durina)  
Coefficient devant le nombre d'années en inactivité (hors scolarité) dans les 5 dernières années  
Source : estimé à l'occasion de **[batchelet2014biographies]** Note : les descriptions de durlNA et de durina sont peut-être interverties.
- double \_ (exp1a3)  
Coefficient devant une variable caractéristique d'avoir de 1 à 3 années d'expériences après les études  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (findet\_rel)  
Coefficient devant l'âge de fin d'études relatif  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (findet\_rel\_a)  
Coefficient devant l'âge de fin d'études relatif, l'année suivant la fin de la scolarité  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (findet\_rel\_b)  
Coefficient devant l'âge de fin d'études relatif, de 1 à 3 ans d'expérience suivant la fin de la scolarité  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (Intercept)  
Coefficient devant la constante  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (nbenf)  
Coefficient devant le nombre d'enfants  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (NbEnf1)  
Coefficient devant le nombre d'enfants de moins de 1 an  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (NbEnf3)  
Coefficient devant le nombre d'enfants de moins de 3 ans  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (nbenf98)  
Coefficient devant le nombre d'enfants (pour les équations sur le début de la vie professionnelle)  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (nbenf98\_a)  
Coefficient devant le nombre d'enfants la première année après la fin des études  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (nbenf98\_b)  
Coefficient devant le nombre d'enfants entre 1 et 3 ans après la fin des études  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (PreRetr\_ARPE)  
Coefficient, non utilisé dans le modèle, devant le fait d'être en pré-retraite.  
Note : Ce coefficient était nécessaire lors de la phase d'estimation, pour ne pas biaiser les projections. D'après **[batchelet2014biographies]** : L'estimation des équations de transition a été effectuée à une période durant laquelle les actifs pouvaient partir en préretraite ; le volume d'inactifs y était donc particulièrement élevé. Afin de neutraliser cet effet des préretraites et donc de ne pas surestimer en projection les volumes d'inactifs, deux variables ont été créées sur l'échantillon d'estimation : une variable d'éligibilité à la préretraite pour l'individu considéré, à partir de l'âge, l'année et la durée cotisée ; une variable relative au volume de préretraites l'année d'observation. Le produit de ces deux variables permet de créer une nouvelle variable utilisée comme variable explicative des équations de transition pour tenir compte des préretraites et notamment mieux modéliser l'effet de l'extinction de certains dispositifs. Les préretraites considérées sont les préretraites publiques : ARPE, PRP, CFA, CAATA (amiante), ASFNE et CATS.  
Source : estimé à l'occasion de **[batchelet2014biographies]**.
- double \_ (PreRetrCAATA)  
voir PreRetr\_ARPE
- double \_ (PreRetrhorsCAATA)  
voir PreRetr\_ARPE
- double \_ (probpact)  
Coefficient visant à prendre en compte l'évolution de la participation des femmes au marché du travail, avec une

- fonction croissante et coudée dans le temps, en log. Les raisons expliquant sa modalité d'implémentation actuelle ne sont plus connues.*
- double `_` (sortant)  
*Coefficient devant une indicatrice pour la première transition à la sortie des études initiales.*  
*Source : estimé à l'occasion de [bachelet2014biographies].*
  - int `_` (destination)  
*Le statut de destination de la transition modélisée (voir `_origine`).*
  - int `_` (ordre)  
*ordre de la transition car les estimations de transition sont des logit emboîtés. De 0 à 6.*
  - int `_` (type\_trans)  
*Le type de transition modélisée. Par sexe, le début de carrière (à partir l'enquête Génération 1998), le milieu de carrière ou la fin de carrière (avec l'échantillon inter-régimes de cotisants (EIC)). Les modalités prises sont donc : DEBUTH, DEBUTF, MILIEUH, MILIEUF, FINH, FINF.*
  - int `_` (origine)  
*Le statut de départ de la transition modélisée. Il y a 7 statuts envisagés : PRI (salarié du secteur privé), INA (inactif), CHO (chômage), MAL (malade ou invalide), IND (être indépendant), TIT (titulaire de la Fonction Publique), CON (les contractuels de la Fonction Publique).*  
*Note : Dans l'estimation, une personne est considérée malade ou invalide une année si le nombre de trimestres cotisés est nul, si sa rémunération est nulle et si elle a validé des trimestres au titre de la maladie (et plus de trimestres maladie que de trimestres au titre du chômage).*
  - int `_` (indic)  
*indic==2 correspond à la dernière transition (état complémentaire des états précédemment examinés).*

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Transitions.h](#)

## 5.12 Référence de la structure Fam

[Fam](#) contient la chronique des événements familiaux depuis 2009.

```
#include <Simulation.h>
```

### Fonctions membres publiques

- IntegerVector `_` (Id)  
*Identifiant de l'individu.*
- IntegerVector `_` (annee)  
*Année.*
- IntegerVector `_` (pere)  
*Identifiant du père.*
- IntegerVector `_` (mere)  
*Identifiant du mère.*
- IntegerVector `_` (matri)  
*Statut matrimonial.*
- IntegerVector `_` (conjoint)  
*Identifiant du conjoint.*
- IntegerVector `_` (enf1)  
*Identifiant de l'enfant n°1.*
- IntegerVector `_` (enf2)  
*Identifiant de l'enfant n°2.*
- IntegerVector `_` (enf3)  
*Identifiant de l'enfant n°3.*
- IntegerVector `_` (enf4)  
*Identifiant de l'enfant n°4.*
- IntegerVector `_` (enf5)  
*Identifiant de l'enfant n°5.*
- IntegerVector `_` (enf6)  
*Identifiant de l'enfant n°6.*
- IntegerVector `_` (anaiss\_enf1, 0)  
*Année de naissance de l'enfant n°1.*
- IntegerVector `_` (anaiss\_enf2, 0)  
*Année de naissance de l'enfant n°2.*
- IntegerVector `_` (anaiss\_enf3, 0)



- *Année de naissance de l'enfant n°3.*  
IntegerVector \_ (anaiss\_enf4, 0)
- *Année de naissance de l'enfant n°4.*  
IntegerVector \_ (anaiss\_enf5, 0)
- *Année de naissance de l'enfant n°5.*  
IntegerVector \_ (anaiss\_enf6, 0)
- *Année de naissance de l'enfant n°6.*

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Simulation.h](#)

### 5.13 Référence de la structure FinEtudeMoy

[FinEtudeMoy](#) désigne l'âge moyen de fin d'études par sexe et par génération.

```
#include <Simulation.h>
```

#### Fonctions membres publiques

- NumericVector \_ (homme)  
*Série des âges de fin d'études moyens des hommes.*
- NumericVector \_ (femme)  
*Série des âges de fin d'études moyens des femmes.*

#### 5.13.1 Description détaillée

Source : enquête Patrimoine 2003 puis hypothèse de stabilité pour les générations récentes

Dernière mise à jour avec :

Feuille : PARAM\_etude / TABLEFINDET0

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Simulation.h](#)

### 5.14 Référence de la structure Indic\_annee

La structure [Indic\\_annee](#) contient des agrégats sur un régime de retraite au cours d'une année, tels que les effectifs de cotisants, de retraités, de liquidants, les montants de cotisations, de dépenses, la pension moyenne, l'âge moyen de liquidation...

```
#include <Indicateurs_annee_COR.h>
```

## Attributs publics

- [Somme Eff\\_Cotisants](#)  
*Effectif d'affiliés au 31/12.*
- [Somme Sal\\_Plaf](#)  
*Montant total des salaires plafonnés (au PSS) et proratisés par la durée en emploi.*
- [Somme Sal\\_Tot](#)  
*Montant total des salaires proratisés par la durée en emploi.*
- [Somme M\\_Cotisations](#)  
*Montant total des cotisations proratisées par la durée en emploi.*
- [Somme Eff\\_DD](#)  
*Effectif de retraités de droits directs au 31/12.*
- [Somme M\\_DD](#)  
*Somme des pensions annualisées de droit direct.*
- [Somme Eff\\_Flux](#)  
*Nombre de nouveaux retraités de droit direct.*
- [Somme Eff\\_Flux\\_Prime](#)  
*Nombre de nouveaux primo-liquidants.*
- [Somme NB\\_vfu](#)  
*Nombre de bénéficiaires d'un versement forfaitaire unique.*
- [Somme M\\_vfu](#)  
*Masse des versements forfaitaires uniques versés.*
- [Somme Eff\\_Min\\_Flux](#)  
*Flux d'individus bénéficiant d'un minimum de pension.*
- [Somme M\\_Min\\_Flux](#)  
*Masse des pensions annualisées de droit direct des nouveaux bénéficiaires d'un minimum de pension.*
- [Somme Eff\\_Min](#)  
*Nombre de bénéficiaires d'un minimum de pension.*
- [Somme M\\_Min](#)  
*Masse des pensions annualisées de droit direct des bénéficiaires d'un minimum de pension.*
- [Somme Eff\\_Derive](#)  
*Nombre de réversataires.*
- [Somme M\\_Derive](#)  
*Montant total des droits dérivés.*
- [Somme Eff\\_Derive\\_seul](#)  
*Nombre d'individus bénéficiant uniquement d'une pension de réversion.*
- [Somme M\\_Derive\\_seul](#)  
*Masse des pensions de réversion des individus ne bénéficiant pas de droit direct.*
- [Somme Eff\\_Flux\\_Derive](#)  
*Flux de réversataires.*
- [Somme M\\_Pensions](#)  
*Montant total des pensions annualisées.*
- [Somme Eff\\_DD\\_Flux\\_coeffSoli](#)  
*Flux de liquidants soumis à un coefficient de solidarité (cf. accord Arrco-Agirc d'octobre 2015)*
- [Somme M\\_DD\\_Flux\\_coeffSoli](#)  
*Masse des pensions soumises à un coefficient de solidarité (cf. accord Arrco-Agirc d'octobre 2015)*
- [Somme M\\_Flux](#)  
*Montant total des pensions des nouveaux retraités.*
- [Somme Eff\\_MV](#)  
*Nombre de bénéficiaires du minimum vieillesse.*
- [Somme Eff\\_Flux\\_MV](#)  
*Nombre de nouveaux bénéficiaires du minimum vieillesse.*
- [Somme M\\_MV](#)  
*Masse des pensions annualisées des bénéficiaires du minimum vieillesse.*
- [Somme M\\_Flux\\_MV](#)  
*Masse des pensions annualisées des nouveaux bénéficiaires du minimum vieillesse.*
- [Somme Eff\\_DD\\_Deces](#)  
*Nombre de décès parmi les retraités de droit direct.*
- [Somme Eff\\_Derive\\_Deces](#)  
*Nombre de décès parmi les réversataires.*
- [Somme M\\_DD\\_Flux](#)  
*Montant total des nouveaux droits directs.*
- [Somme M\\_Flux\\_Derive](#)  
*Montant des flux de droits dérivés.*
- [Somme M\\_DD\\_Deces](#)  
*Montant des droits directs (au 31/12) des décédés.*
- [Somme M\\_Derive\\_Deces](#)  
*Montant des droits dérivés des décédés.*
- [Somme Eff\\_Cotisants\\_ma](#)  
*Effectif d'affiliés au prorata de la durée passée en emploi.*

- [Somme Eff\\_DD\\_ma](#)  
*Effectif de retraités de droits directs au prorata de la durée passée en retraite.*
- [Somme Eff\\_Min\\_ma](#)  
*Nombre de bénéficiaires d'un minimum de pension, au prorata de la durée passée en retraite.*
- [Somme Eff\\_DD\\_Flux\\_coeffSoli\\_ma](#)  
*Flux de liquidants soumis à un coefficient de solidarité (cf. accord Arrco-Agirc d'octobre 2015), au prorata de la durée passée en emploi.*
- [Somme Eff\\_Min\\_Flux\\_ma](#)  
*Flux d'individus bénéficiant d'un minimum de pension, au prorata de la durée passée en retraite.*
- [Somme M\\_DD\\_Flux\\_ma](#)  
*Flux des pensions proratisées de droit direct.*
- [Somme M\\_DD\\_ma](#)  
*Somme des pensions proratisées de droit direct.*
- [Somme M\\_DD\\_Flux\\_coeffSoli\\_ma](#)  
*Masse des pensions proratisées soumises à un coefficient de solidarité (cf. accord Arrco-Agirc d'octobre 2015)*
- [Somme M\\_Min\\_Flux\\_ma](#)  
*Masse des pensions proratisées de droit direct des nouveaux bénéficiaires d'un minimum de pension.*
- [Somme M\\_Min\\_ma](#)  
*Masse des pensions proratisées de droit direct des bénéficiaires d'un minimum de pension.*
- [Somme Eff\\_Flux\\_ma](#)  
*Nombre de nouveaux retraités de droit direct, au prorata de la durée passée en retraite.*
- [Somme M\\_Pensions\\_ma](#)  
*Montant total des pensions proratisées.*
- [Somme M\\_MICO\\_flux\\_ma](#)  
*Montant des nouvelles dépenses, proratisées, effectuées au titre des minima de pensions.*
- [Somme M\\_MICO\\_stock\\_ma](#)  
*Montant des dépenses, proratisées, effectuées au titre des minima de pensions.*
- [Somme Sal\\_Plaf\\_31dec](#)  
*Montant total des salaires plafonnés (au PSS), au 31 décembre.*
- [Somme Sal\\_Tot\\_31dec](#)  
*< Montant total des salaires, au 31 décembre*
- [Moyenne Age\\_Ret\\_Flux\\_Primo](#)  
*Âge moyen de liquidation des primo-liquidants.*
- [Moyenne Coeff\\_Prorat\\_Flux](#)  
*Coefficient de proratisation moyen des liquidants.*
- [Moyenne Sal\\_fin\\_carr](#)  
*Salaire moyen de fin de carrière.*
- [Moyenne AgeMoyRetr\\_finanee](#)  
*Âge moyen (au 31/12) des retraités de droit direct.*
- [Moyenne PMoy\\_Flux](#)  
*Pension moyenne de droit direct des liquidants.*
- [Moyenne Age\\_Ret\\_Flux](#)  
*Âge moyen de liquidation totale.*
- [Moyenne Dur\\_Ass\\_Flux](#)  
*Durée moyenne d'assurance, y compris majorations, des liquidants.*
- [Moyenne PMoy\\_Flux\\_Derive](#)  
*Montant moyen du droit de réversion des nouveaux réversataires.*
- [Moyenne PMoy\\_DD](#)  
*< Pension moyenne de droit direct, en euros 2012*
- [Ratio TauxRemplace](#)  
*< Taux de remplacement de la somme des salaires du régime par la somme des pensions tous régimes*

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/Indicateurs\\_annee\\_COR.h](#)

## 5.15 Référence de la structure Indic\_gen

contient des indicateurs par génération : les différents motifs de liquidation, les moyennes d'âge à la liquidation, de durée validée, taux de remplacement...

```
#include <Indicateurs_gen.h>
```

## Attributs publics

- Moyenne AGEPRIMOLIQ  
*Âge moyen à la primo-liquidation sur la génération.*
- Moyenne AGETOTLIQ  
*Âge moyen à la liquidation sur la génération.*
- Moyenne DUREEVAL  
*Durée validée moyenne.*
- Taux DEPARTAVANT  
*(à vérifier) obsolète ?*
- Taux DEPARTNORM  
*(à vérifier) obsolète ?*
- Taux DEPARTENTRE  
*(à vérifier) obsolète ?*
- Taux DEPARTANNDDEC  
*(à vérifier) obsolète ?*
- Taux DEPARTAPRES  
*(à vérifier) obsolète ?*
- Taux DEPARTNODECFP  
*(à vérifier) obsolète ?*
- Taux DEPARTTPDUR  
*(à vérifier) obsolète ?*
- Taux DEPARTTPAGE  
*(à vérifier) obsolète ?*
- Taux DEPARTTPINAPT  
*(à vérifier) obsolète ?*
- Taux DEPARTDECOTE  
*(à vérifier) obsolète ?*
- Taux DEPARTSURCOTE  
*(à vérifier) obsolète ?*
- Moyenne PMOYLIQ  
*Pension moyenne à la liquidation.*
- Moyenne COEFFPRORAT  
*Coefficient de proratisation moyen.*
- Somme NBAFF  
*(à vérifier) obsolète ?*
- Somme NBRETTOT  
*(à vérifier) Nombre de personnes qui seront partis en retraite*
- Somme NBRETPRIMO  
*(à vérifier) Nombre de personnes qui auront eu une primo-liquidation*
- Somme NBVFU  
*(à vérifier) Nombre de personnes qui auront bénéficié d'un versement forfaitaire unique*
- Taux PARTAFF  
*(à vérifier) obsolète ?*
- Taux PARTRETTOT  
*(à vérifier) obsolète ?*
- Taux PARTRETPRIMO  
*(à vérifier) obsolète ?*
- Taux PARTVFU  
*(à vérifier) obsolète ?*
- Moyenne TAUXREPLBRUT  
*Taux de remplacement brut.*
- Moyenne SALFINCARR  
*Salaire moyen de fin de carrière.*
- Moyenne AGEFINEMPL  
*Âge moyen de fin d'emploi.*
- Moyenne PENSION68  
*Pension moyenne à 68 ans.*

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Indicateurs\\_gen.h](#)

## 5.16 Référence de la structure Indic\_gen\_age

Cette structure contient les principaux indicateurs calculés par génération et par âge.

```
#include <Indicateurs_gen_age.h>
```

## Attributs publics

- [Somme PopTOT](#)  
*Nombre de personnes.*
- [Somme PopEmploi](#)  
*Nombre de personnes en emploi.*
- [Somme PopActive](#)  
*Nombre de personnes actives.*
- [Somme PopChomage](#)  
*Nombre de personnes au chômage.*
- [Somme PopRetraitee](#)  
*Nombre de retraités.*
- [Somme SalTot](#)  
*Montant total des salaires.*
- [Somme immigrTot](#)
- [Moyenne SMPTEch](#)  
*< Salaire moyen par tête dans l'échantillon*
- [Taux TxChomage](#)  
*Taux de chômage.*
- [Taux TxActif](#)  
*Taux d'activité*
- [Taux TxCadre](#)  
*< Pourcentage de cadres parmi les personnes en emploi*

La documentation de cette structure a été générée à partir du fichier suivant :

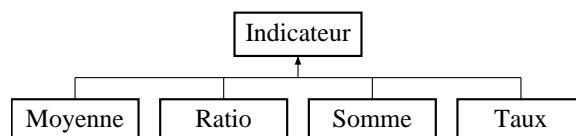
- [src/Indicateurs\\_gen\\_age.h](#)

## 5.17 Référence de la structure Indicateur

Structure à l'origine des structures [Moyenne](#), [Ratio](#), [Somme](#) et [Taux](#).

```
#include <Statistiques.h>
```

Graphe d'héritage de Indicateur :



## Fonctions membres publiques

- virtual double **resultat** ()=0
- virtual **operator double** ()

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/Statistiques.h](#)

## 5.18 Référence de la classe Indiv

Classe contenant l'ensemble des informations sur un individu.

```
#include <Indiv.h>
```

## Fonctions membres publiques

- bool `est_present` (const int t) const  
*Indique si la personne est vivante à la date t et est présente en France.*
- bool `decede` (int t)  
*Indique si la personne décède l'année t.*
- bool `ACHarge` (int t)  
*Indique si la personne est à la charge d'un de ses parents.*
- `Indiv & PersRef` (int t)  
*Renvoie la personne de référence du ménage de l'individu.*
- int `NbEnfC` (int t, int ageEnfMax=`AGE_MAX`)  
*Renvoie le nombre d'enfants à charge une année donnée.*
- int `TailMen` (int t)  
*Renvoie le nombre d'individus du ménage.*
- double `SalNet` (int t)  
*Salaire net de l'année t.*
- double `RevNet` (int t)  
*Revenu net de l'année t.*
- double `SalMoyBrut` (double agefin, int nb) const  
*Salaire moyen brut.*
- int `age` (int t) const  
*Renvoie l'âge de l'individu à la date t.*
- double `datefm` (double `age`, double mois)  
*Renvoie la date précise à un âge fin.*
- int `date` (int `age`) const  
*Renvoie la date (modulo 1900) pour un âge donné*
- bool `est_persRef` (int t)  
*Indique si la personne est personne de référence de son ménage l'année t.*
- bool `est_decede` (int t)  
*Indique si la personne est décédée l'année t.*
- double `survie` (int age1, int age0)  
*Probabilité de survie à l'âge age1 sachant la survie à l'âge age0.*
- double `esp_vie` (int age0)  
*Espérance de vie à l'âge age0.*
- int `nb_enf` (int `age`) const  
*Nombre d'enfants à l'âge age.*
- int `nb_enf` () const  
*Nombre d'enfants sur l'ensemble de la vie.*
- int `nb_enf` (int `age`, int ageEnfMax) const  
*Nombre d'enfants à un âge donné, avec un âge maximal pour les enfants.*
- `Indiv` ()  
*Constructeur.*
- `Indiv` (`Ech` &ech, `Emp` &emp, `Fam` &fam, int &i, int &i\_e, int &i\_f)  
*Constructeur à partir des tables Ech, Fam et Emp.*
- `Indiv` (int `ld`, int `sexe`, int `anaiss`, int `findet`, int `age`, int `pere`, int `mere`)  
*Constructeur en précisant certains paramètres.*
- `Indiv` (const `Indiv` &indivACopier)  
*Constructeur d'un individu en copiant un individu.*
- `~Indiv` ()

## Attributs publics

- int `ld` = 0  
*Identifiant de l'individu.*
- int `sexe` = 0  
*Sexe.*
- int `anaiss` = 0  
*Année de naissance.*
- int `findet` = 0  
*Âge de fin d'étude.*
- int `typeFP` = `NONFP`  
*Type de fonction publique (fonction publique d'Etat ou fonction publique territoriale et hospitalière)*
- int `neFrance` = 0  
*Indicatrice né en france.*
- int `emigrant` = 0  
*Indicatrice emigrant.*
- int `ageMax` = 0

- *Âge maximal de statut renseigné (min âge décès, âge en fin de projection)*
- int `moisnaiss` = 0
- *Mois de naissance (0=janvier, 11=décembre)*
- int `dateMinFlux` = 999999
- *Date d'entrée dans le minimum vieillesse.*
- int `peudip` = 0
- *Indicatrice d'être peu diplômé par rapport à sa génération et son sexe.*
- int `tresdip` = 0
- *Indicatrice d'être très diplômé par rapport à sa génération et son sexe.*
- int `dipl` = 0
- *Niveau de diplôme de l'individu en classes.*
- double `taux_prim` = 0
- *Taux de prime FP.*
- double `k` = 0
- *Préférence pour le loisir.*
- double `age_exo` = 0
- *Âge de liquidation exogène.*
- vector< double > `salaires` = vector<double>(AGE\_MAX, 0.0)
- *tableau des salaires indicés par l'âge*
- vector< int > `statuts` = vector<int>(AGE\_MAX, 0)
- *tableau des statuts indicés par l'âge*
- vector< int > `sante` = vector<int>(AGE\_MAX, 0)
- *tableau des états de santé indicés par l'âge*
- vector< int > `enfprestafam` = vector<int>(AGE\_MAX, 0)
- *tableau indiquant si l'enfant a donné droit à des prestas fam pour un de ses parents (évite les doubles comptes des enfants)*
- vector< int > `prestafam` = vector<int>(AGE\_MAX, 0)
- *tableau des types de prestation familiale indexé par l'âge*
- vector< double > `cotis_rg` = vector<double>(AGE\_MAX, 0.0)
- *tableau des cotisations retraite au régime général par l'âge (part patronale et salariale)*
- vector< double > `cotis_fp` = vector<double>(AGE\_MAX, 0.0)
- *tableau des cotisations retraite au régime de la fonction publique par l'âge*
- vector< double > `cotis_in` = vector<double>(AGE\_MAX, 0.0)
- *tableau des cotisations retraite au régime des indépendants par l'âge*
- vector< int > `matri` = vector<int>(AGE\_MAX, 0)
- *tableau des situation matrimoniales par âge*
- vector< int > `conjoint` = vector<int>(AGE\_MAX, 0)
- *tableau des identifiants de conjoint par âge*
- int `pere` = 0
- *identifiant du pere*
- int `mere` = 0
- *identifiant de la mère*
- int `pseudo_conjoint` = 0
- *identifiant du pseudo-conjoint*
- vector< int > `enf` = vector<int>(6, 0)
- *tableau des identifiants des enfants*
- vector< int > `anaissEnf` = vector<int>(6, 0)
- *tableau des années de naissances des enfants*
- ptr< `Retraite` > `retr` = nullptr
- *Pointeur vers un objet de la classe `Retraite`.*

### 5.18.1 Description détaillée

Cette classe contient l'ensemble des informations sur l'individu lues dans les tables ech, emp, et fam. Elle contient de plus un pointeur sur un objet `Retraite` relatif à sa situation courante.

### 5.18.2 Documentation des constructeurs et destructeur

#### 5.18.2.1 ~Indiv()

```
~Indiv ( ) [inline]
```

Destructeur

### 5.18.3 Documentation des fonctions membres

#### 5.18.3.1 ACharge()

```
bool ACharge (
    int t )
```

##### Renvoie

true si la personne est à charge false sinon

#### 5.18.3.2 decede()

```
bool decede (
    int t )
```

##### Renvoie

true si année du décès, false sinon

#### 5.18.3.3 esp\_vie()

```
double esp_vie (
    int age0 )
```

Espérance de vie à l'âge age0

#### 5.18.3.4 est\_present()

```
bool est_present (
    const int t ) const [inline]
```

##### Renvoie

Renvoie vrai si l'individu est présent à une date t et faux si l'individu n'est pas vivant ou pas encore présent sur le territoire.

#### 5.18.3.5 PersRef()

```
Indiv & PersRef (
    int t )
```

##### Renvoie

Identifiant de la personne de référence du ménage

#### 5.18.3.6 SalMoyBrut()

```
double SalMoyBrut (
    double agefin,
    int nb ) const
```

##### Renvoie

Renvoie le salaire moyen de l'individu pour un âge donné et un nombre de salaires donné



### 5.18.3.7 SalNet()

```
double SalNet (
    int t )
```

Renvoie le salaire net d'un individu

### 5.18.3.8 survie()

```
double survie (
    int age1,
    int age0 )
```

Probabilité de survie à l'âge age1 sachant la survie à l'âge age0

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/Indiv.h
- src/Indiv.cpp

## 5.19 Référence de la classe Leg

Contient les valeurs des paramètres législatifs pour un individu, un âge et une date de législation donnés.

```
#include <Legislation.h>
```

### Fonctions membres publiques

- [Leg](#) (const [Indiv](#) &X, int age, int [an\\_leg](#))  
*Crée un objet législation pour un individu, un âge et une date de législation donnés.*

### Attributs publics

- double [DureeMinFPA](#)  
*Durée minimale de carrière pour bénéficier d'une pension FP en catégorie active (ajout 23/09/2011)*
- int [an\\_leg](#)  
*Année de législation.*
- int [an\\_leg\\_DRA](#)  
*Année de la législation appliquée en ce qui concerne les départs en retraite anticipé*
- int [an\\_leg\\_SAM](#)  
*Année de législation appliquée en ce qui concerne le calcul du SAM.*
- double [DureeCibRG](#)  
*Durée cible au régime général.*
- double [DureeProratRG](#)  
*Durée de proratisation du régime général.*
- double [DureeCalcSAM](#)  
*Nombre d'années pour calcul du SAM dans le RG.*
- double [TauxPleinRG](#)  
*Taux plein du régime général.*
- double [DecoteRG](#)  
*Coefficient de décote du régime général (par année d'écart au taux plein)*
- double [SurcoteRG1](#)  
*Coefficient de surcote du régime général, 1ere année.*
- double [SurcoteRG2](#)  
*Coefficient de surcote du régime général, 2eme année.*
- double [SurcoteRG3](#)  
*Coefficient de surcote du régime général, 3eme année.*
- double [TauxRGMax](#)  
*Taux maximal de la pension RG.*

- double [MajTauxRGMax](#)  
*Majoration du taux maximal de pension pris en compte dans le plafonnement.*
- double [DureeCibFP](#)  
*Durée cible à la fonction publique.*
- double [DureeProratFP](#)  
*Durée de référence pour une carrière complète à la FP (au dénominateur du coefficient de proratisation)*
- double [DecoteFP](#)  
*Decote fonction publique.*
- double [SurcoteFP](#)  
*Surcote fonction publique.*
- double [AgeAnnDecFP](#)  
*Âge d'annulation de la décote fonction publique.*
- double [AgeMinMG](#)  
*Âge minimal de jouissance du minimum garanti (ajout 27/10/2010)*
- double [AgeMaxFP](#)  
*Âge de liquidation maximal FP.*
- int [LegMin](#)  
*Millésime de la législation appliquée pour le calcul des minima de pensions.*
- int [LegDRA](#)  
*Millésime de la législation appliquée pour les départs anticipés.*
- int [LegSAM](#)  
*Millésime de la législation appliquée pour le calcul du SAM.*
- double [AgeMinRG](#)  
*Âge minimal d'ouverture des droits, RG.*
- double [AgeMaxRG](#)  
*Âge maximal de liquidation RG.*
- double [AgeAnnDecRG](#)  
*Âge d'annulation de la décote au RG.*
- double [DureeMinFP](#)  
*Durée minimale de carrière pour bénéficier d'une pension à la fonction publique (ajout 25/10/2010)*
- double [AnOuvDroitFP](#)  
*Année d'ouverture des droits dans la fonction publique (peut être différente de l'année des 60 ans pour les catégories actives et les mères de 3 enfants) (ajout 09/12/2011)*
- double [AgeSurcote](#)  
*Âge minimal pour le calcul de la surcote (60 ans avant la réforme de 2010, 62 ans progressivement après)*
- double [DeplafProrat](#)  
*Durée au-delà de la durée de référence prise en compte dans le calcul du coefficient de proratisation (dans tous les régimes) (dans le cadre d'un déplafonnement progressif de ce coefficient)*
- double [DureeValCibDRA](#) [5]  
*Durée validée requise pour un départ anticipé pour carrière longue (plusieurs "jeux" de condition de départ sont possibles, il s'agit donc d'un tableau)*
- double [DureeCotCibDRA](#) [5]  
*Durée cotisée requise pour un départ anticipé pour carrière longue (plusieurs "jeux" de condition de départ sont possibles, il s'agit donc d'un tableau)*
- double [DebActCibDRA](#) [5]  
*Âge de début d'activité requis pour un départ anticipé pour carrière longue (plusieurs "jeux" de condition de départ sont possibles, il s'agit donc d'un tableau)*
- double [AgeDRA](#) [5]  
*Âge d'ouverture des droits dans le cadre d'un départ anticipé pour carrière longue (sous conditions de durée et d'âge définies par l'indice)*
- double [AgeMinFP](#)  
*Âge minimal d'ouverture des droits, FP (cet âge est défini par individu, après analyse de la catégorie pertinente)*
- double [ageouvdroitfp](#)  
*Âge d'ouverture des droits pour les fonctionnaires sédentaires calculé en fonction de la génération.*
- int [durtpa](#)  
*Durée passée en catégorie active.*

#### 5.19.1 Description détaillée

[Leg](#) contient les valeurs des paramètres législatifs pour un individu X et une année donnés. Ces paramètres sont ceux qui ne sont pas directement importés depuis le fichier Excel de paramètres sociaux. Ces paramètres législatifs dépendent des caractéristiques individuelles, telles que la génération, le sexe, l'année, et qui sont utiles au calcul des droits à la liquidation avec [DroitsRetr](#).

## 5.19.2 Documentation des constructeurs et destructeur

### 5.19.2.1 Leg()

```
Leg (
    const Indiv & X,
    int age,
    int an_leg )
```

Crée un objet législation pour un individu particulier, à un age donné, et une législation donnée.

Appelle les sous-méthodes suivantes :

- leg\_priv (Législation régimes RG, RSI, Arrco, Agirc)
- leg\_pub (Législation régime FP)
- leg\_spe (Régimes spéciaux)
- leg\_dra (Législation sur les départ en retraite anticipée pour carrière longue)
- an\_ouv\_droit\_fp (Calcul de l'année d'ouverture des droits à la FP)

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/Legislation.h
- src/Legislation.cpp

## 5.20 Référence de la structure Macro

**Macro** contient l'ensemble des paramètres macroéconomiques et des paramètres spécifiques aux retraites.

```
#include <Simulation.h>
```

### Fonctions membres publiques

- NumericVector \_ (Prix)
  - Indice des Prix à la Consommation (IPC) France entière, y compris tabac (évolution annuelle appliquée à la série initiale à partir de 2006)*
  - Source : Insee*
  - Dernière mise à jour avec :*
  - Feuille : ParamSociauxXXXX / ParamGene.*
- NumericVector \_ (PlafondSS)
  - Plafond de la Sécurité Sociale, en euros courants*
  - Source : CNAV*
  - Dernière mise à jour avec :*
  - Feuille : ParamSociauxXXXX / ParamGene.*
- NumericVector \_ (SMIC)
  - Montant annuel du SMIC, tel qu'utilisé pour déterminer les cotisations AVPF (2028 heures par an (169\*12) - 39 heures par semaine), moyenne annuelle*
  - Source : Insee*
  - Dernière mise à jour avec :*
  - Feuille : ParamSociauxXXXX / ParamGene.*
- NumericVector \_ (PointFP)
  - Valeur du point d'indice de la Fonction Publique, en euros courants*
  - Source :*
  - Dernière mise à jour avec :*
  - Feuille : ParamSociauxXXXX / ParamGene.*
- NumericVector \_ (SMPT)
  - [Salaire](#) Moyen Par Tête (SMPT) dans l'ensemble de l'économie*
  - Source : Insee*
  - Dernière mise à jour avec :*
  - Feuille : ParamSociauxXXXX / ParamGene.*
- NumericVector \_ (PIB)

- Produit Intérieur Brut (PIB), en euros courants*  
*Source :*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamGene.*
- NumericVector \_ (SMPTp)  
*Évolution en projection du SMPT réel (ensemble de l'économie)*  
*Source : COR, scénario défini en entrée du modèle*  
*Feuille : ParamSociauxXXXX / choX\_pibY%.*
  - NumericVector \_ (PIBp)  
*Évolution en projection du PIB en volume*  
*Source : COR, scénario défini en entrée du modèle*  
*Feuille : ParamSociauxXXXX / choX\_pibY%.*
  - NumericVector \_ (PlafondSSp)  
*Évolution en projection du Plafond de la Sécurité Sociale*  
*Source : COR, scénario défini en entrée du modèle*  
*Feuille : ParamSociauxXXXX / choX\_pibY%.*
  - NumericVector \_ (SMICp)  
*Évolution en projection du SMIC*  
*Source : COR, scénario défini en entrée du modèle*  
*Feuille : ParamSociauxXXXX / choX\_pibY%.*
  - NumericVector \_ (Prixp)  
*Évolution en projection de l'indice des prix*  
*Source : COR, scénario défini en entrée du modèle*  
*Feuille : ParamSociauxXXXX / choX\_pibY%.*
  - NumericVector \_ (RevaloSPC)  
*Pour l'année N, ratio (coefficient de revalorisation du Salaire Porté au Compte (SPC) de l'année N / coefficient de revalorisation du SPC de l'année N+1)*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (RevaloRG)  
*Revalorisation des pensions liquidées au Régime Général Si plusieurs valeurs au cours d'une année N, on chaîne les coefficient en les multipliant*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (RevaloFP)  
*Revalorisation des pensions liquidées pour la Fonction Publique*  
*Source initiale : Drees, Taux de revalorisation des pensions dans la Fonction Publique utilisé par la DREES dans son modèle CALIPER*  
*Dernière mise à jour avec : ServicePublic.fr*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (TauxSalRGSP)  
*Taux de cotisation salarié, sous le plafond, au Régime Général*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (TauxEmpRGSP)  
*Taux de cotisation employeur, sous le plafond, au Régime Général*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (TauxSalRGSalTot)  
*Taux de cotisation salarié portant spécifiquement sur le salaire total, au Régime Général (=assurance veuvage avant 2004)*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (TauxEmpRGSalTot)  
*Taux de cotisation employeur portant spécifiquement sur le salaire total, au Régime Général*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRetrBase.*
  - NumericVector \_ (TauxFP)  
*Taux de cotisation Employé dans la Fonction Publique.*  
*Note : D'ici à 2020 la loi de 2010 prévoit l'augmentation du taux de cotisation à 10,55%.*

- Source : IPP  
Dernière mise à jour avec : IPP  
Feuille : ParamSociauxXXXX / ParamRetrBase.
- NumericVector \_ (TauxRSIsP)  
Taux de cotisation (sous le plafond de la Sécurité Sociale) des travailleurs indépendants (ex-RSI)  
Source :  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (TauxRSIsurP)  
Taux de cotisation (au dessus du plafond de la Sécurité Sociale) des travailleurs indépendants (ex-RSI)  
Source :  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (TauxEmplFPE)  
Taux de cotisation Employeur au sein de la Fonction Publique de l'Etat Taux qui équilibre le solde élargi en 2011 figé militaires+civils. Pour le passé, en réropolation et on reprend avant 2006 chiffres du Jaune Budgétaire.  
Sources : COR et Jaune Budgétaire.  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (TauxEmplFPTH)  
Taux de cotisation Employeur à la CNRACL (pour la fonction publique Territoriale et Hospitalière)  
Source : COR  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (TauxEmplFPMoy)  
Pondération selon parts masses salariales FPE et FPT/H.  
Source : Calculs à partir données DGAFP.  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (MinVieil1)  
Minimum Vieillesse pour une personne seule  
Source : CNAV  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (MinVieil2)  
Minimum Vieillesse pour une personne en couple  
Source : CNAV  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (Mincont1)  
Minimum contributif  
Source : CNAV  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (Mincont2)  
Minimum contributif majoré (pour les personnes totalisant un nombre suffisant de trimestres effectivement cotisés)  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (CoefMdaRG)  
Majoration de durée d'assurance pour âge, au Régime général  
Pour une personne ayant eu le taux plein par condition sur l'âge, permet de surcoter voir par exemple : [http://www.legislation.cnav.fr/Pages/expose.aspx?Nom=mda\\_mda\\_age\\_mda\\_age\\_ex](http://www.legislation.cnav.fr/Pages/expose.aspx?Nom=mda_mda_age_mda_age_ex)  
Source : CNAV  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (CoefMdaFP)  
Majoration de durée d'assurance pour âge, dans la Fonction Publique Pour une personne ayant eu le taux plein par condition sur l'âge, permet de surcoter voir par exemple : [http://www.legislation.cnav.fr/Pages/expose.aspx?Nom=mda\\_mda\\_age\\_mda\\_age\\_ex](http://www.legislation.cnav.fr/Pages/expose.aspx?Nom=mda_mda_age_mda_age_ex)  
Source : CNAV  
Dernière mise à jour avec :  
Feuille : ParamSociauxXXXX / ParamRetrBase.
  - NumericVector \_ (SeuilVFURG)  
Seuil du Versement Forfaitaire Unique (VFU) au régime général  
Source : CNAV  
Dernière mise à jour avec :

- Feuille : ParamSociauxXXXX / ParamRetrBase.*

— NumericVector \_ (SalValid)

*Salaire validant un trimestre au régime général*

*Source : CNAV*

*Dernière mise à jour avec :*
- Feuille : ParamSociauxXXXX / ParamRetrBase.*

— NumericVector \_ (TauxARRCO\_1)

*Taux de cotisation moyen à l'ARRCO sur la tranche 1*

*Source : COR*

*Dernière mise à jour avec :*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxARRCO\_2)

*Taux de cotisation moyen à l'ARRCO sur la tranche 2*

*Source : COR*

*Dernière mise à jour avec :*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxARRCO\_S1)

*Taux de cotisation employé moyen à l'ARRCO sur la tranche 1*

*Source : Calcul : 40% de \_ (TauxARRCO\_1)*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxARRCO\_S2)

*Taux de cotisation employé moyen à l'ARRCO sur la tranche 2*

*Source : Calcul : 40% de \_ (TauxARRCO\_2)*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAppARRCO)

*Taux d'appel à l'ARRCO*

*Source : COR*

*Dernière mise à jour avec :*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (SalRefARRCO)

*Salaire de référence ARRCO (i.e. valeur d'achat du point UNIRS/ARRCO)*

*Source : AGIRC/ARRCO*

*Dernière mise à jour avec : IPP*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (ValPtARRCO)

*Valeur de service du point UNIRS/ARRCO*

*Source : AGIRC/ARRCO*

*Dernière mise à jour avec : IPP*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxMalComp)

*Taux cotisation maladie sur les retraites complémentaires*

*Source :*

*Dernière mise à jour avec :*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_B)

*Taux de cotisation total à l'AGIRC sur la tranche B*

*Source : COR*

*Dernière mise à jour avec :*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_C)

*Taux de cotisation total à l'AGIRC sur la tranche C*

*Source : COR*

*Dernière mise à jour avec :*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_SB)

*Taux de cotisation salarié à l'AGIRC sur la tranche B*

*Source : Calcul : 40% de \_ (TauxAGIRC\_B)*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_SC)

*Taux de cotisation salarié à l'AGIRC sur la tranche C*

*Source : Calcul : 40% de \_ (TauxAGIRC\_C)*

*Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAppAGIRC)

*Taux d'appel à l'AGIRC*

*Source : COR*

*Dernière mise à jour avec :*

- Feuille : ParamSociauxXXXX / ParamRetrComp.*
 — NumericVector \_ (SalRefAGIRC)
  - Salaire* de référence AGIRC (i.e. valeur d'achat du point AGIRC)
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- Feuille : ParamSociauxXXXX / ParamRetrComp.*
 — NumericVector \_ (ValPtAGIRC)
  - Valeur de service du point AGIRC
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (GMP)
  - Garantie Minimale de Point (GMP) à l'AGIRC
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_ARRCO\_1)
  - Taux* de cotisation total au régime unifié AGIRC/ARRCO, sur la tranche 1
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_ARRCO\_2)
  - Taux* de cotisation total au régime unifié AGIRC/ARRCO, sur la tranche 2
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAppAGIRC\_ARRCO)
  - Taux* d'appel au régime unifié AGIRC/ARRCO
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_ARRCO\_S1)
  - Taux* de cotisation salarié au régime unifié AGIRC/ARRCO, sur la tranche 1
  - Source : Calcul : 40% de \_ (TauxAGIRC\_ARRCO\_1)
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxAGIRC\_ARRCO\_S2)
  - Taux* de cotisation salarié au régime unifié AGIRC/ARRCO, sur la tranche 1
  - Source : Calcul : 40% de \_ (TauxAGIRC\_ARRCO\_2)
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (SalRefAGIRC\_ARRCO)
  - Salaire* de référence AGIRC/ARRCO (i.e. valeur d'achat du point AGIRC/ARRCO)
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (ValPtAGIRC\_ARRCO)
  - Valeur de service du point AGIRC/ARRCO
  - Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRetrComp.*
- NumericVector \_ (TauxRevAGIRC\_ARRCO)
  - Taux* de réversion dans le régime unifié AGIRC/ARRCO Source : AGIRC/ARRCO
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (TauxRevRG)
  - Taux* de réversion au Régime général
  - Note : Laissé à 54% (depuis 2010 majoration de 11,1% pour les faibles pensions)
  - Source :
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (MinRevRG)
  - Minimum de pension de réversion au régime général, en euros courants
  - Source : CNAV
  - Dernière mise à jour avec :
  - Feuille : ParamSociauxXXXX / ParamRev.*



- NumericVector \_ (MaxRevRG)  
*Maximum de pension de réversion au régime général, en euros courants*  
*Source : Calcul : 0,27 fois le plafond de la sécurité sociale*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (PlafRevRG)  
*Plafond de ressources pour toucher la réversion, au régime général*  
*Source : CNAV*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (TauxRevARRCO)  
*Taux de réversion à l'ARRCO*  
*Source :*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (TauxRevAGIRC)  
*Taux de réversion à l'AGIRC*  
*Source :*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (TauxRevFP)  
*Taux de réversion dans la fonction publique*  
*Source :*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (TauxRevInd)  
*Taux de réversion chez les indépendants*  
*Source :*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamRev.*
- NumericVector \_ (TauxPR1, 0)  
*Plafond correspondant à l'allocation spéciale du fond national pour l'emploi (ASFNE)*  
*Note : N'est plus utilisée dans le modèle, voir AgeEligPR*  
*Source : Dares*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamPreRet.*
- NumericVector \_ (TauxPR2, 0)  
*Plafond correspondant à l'allocation spéciale du fond national pour l'emploi (ASFNE)*  
*Note : N'est plus utilisée dans le modèle, voir AgeEligPR*  
*Source : Dares*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamPreRet.*
- NumericVector \_ (TauxPR3, 0)  
*Plafond correspondant à l'allocation spéciale du fond national pour l'emploi (ASFNE)*  
*Note : N'est plus utilisée dans le modèle, voir AgeEligPR*  
*Source : Dares*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamPreRet.*
- NumericVector \_ (AgeEligPR, 0)  
*Âge correspondant à l'allocation spéciale du fond national pour l'emploi (ASFNE)*  
*N'est plus utilisé dans le modèle.*  
*Note (site Pole-Emploi) : "L'allocation de préretraite licenciement (ASFNE) est financée par l'Etat avec la participation de l'employeur, du salarié et de l'assurance chômage. Elle est versée aux personnes de 57 ans et plus dont l'entreprise a conclu avec l'Etat une convention d'allocations spéciales du Fonds National de l'Emploi. L'article numéro 152 de la loi de finances numéro 2011-1977 du 28 décembre 2011 procède à la suppression définitive du dispositif mais les conventions conclues antérieurement au 1er janvier 2012 continuent à produire leurs effets jusqu'à leur terme."*  
*Source : Dares*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamPreRet.*
- NumericVector \_ (BMAF)  
*Base mensuelle de calcul des allocations familiales (BMAF)*  
*Source : CNAF*  
*Dernière mise à jour avec :*  
*Feuille : ParamSociauxXXXX / ParamFam.*
- NumericVector \_ (PlafCF1)



*Plafond de ressources du complément familial (1 enfant)*

Source :

Dernière mise à jour avec :

Note 1 : une condition pour valider des périodes en AVPF est d'avoir perçu des allocations tel le complément familial, (plus parfois une seconde condition de ressource propre à l'AVPF). Les PlafCF correspondent aux plafonds pour le complément familial, indépendamment de l'AVPF.

Note 2 : En 2018, le complément familial est versé, sous certaines conditions, aux personnes ayant au moins 3 enfants à charge. Historiquement, il a pu être versé dès 1 enfant, d'où probablement la définition de plafonds dès 1 enfant. Il se peut aussi qu'il corresponde dans l'historique aussi à d'autres allocations.

Note 3 : (présente dans la documentation historique de Destinie) L'allocation pour jeune enfant (APJE) est intégrée à la PAJE (prestation d'accueil pour jeune enfant) depuis 2004. La valeur de 2007 est obtenue par moyenne des deux valeurs Cnaf (avant/après juillet 2007), puis par rétropolation de la série par application de l'évolution Cnaf.

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafCF2)

*Plafond de ressources du complément familial (2 enfants)*

Source :

Dernière mise à jour avec :

Notes : voir PlafCF1

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafCF3)

*Plafond de ressources du complément familial (3 enfants)*

Source :

Dernière mise à jour avec :

Notes : voir PlafCF1

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafCF4)

*Plafond de ressources du complément familial (4 enfants)*

Source :

Dernière mise à jour avec :

Notes : voir PlafCF1

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafCF5)

*Plafond de ressources du complément familial (enfants au delà de 4 enfants)*

Source :

Dernière mise à jour avec :

Notes : voir PlafCF1

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (MajoPlafCF)

*Majoration du plafond ressources du complément familial*

Source :

Dernière mise à jour avec :

Notes : voir PlafCF1

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafARS1, 0)

*Plafond de ressources pour recevoir l'Allocation spéciale de rentrée, pour 1 enfant*

Note 1 : Ces plafonds PlafARS sont utilisés car ce sont les conditions de ressources spécifiques pour être affilié à l'AVPF.

Note 2 : La valeur de 2007 est la valeur Cnaf. Les précédentes sont obtenues par rétropolation de la série par application de l'évolution Cnaf

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafARS2, 0)

*Plafond de ressources pour recevoir l'Allocation spéciale de rentrée, pour 2 enfants* Note : La valeur de 2007 est la valeur Cnaf. Les précédentes sont obtenues par rétropolation de la série par application de l'évolution Cnaf

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafARS3, 0)

*Plafond de ressources pour recevoir l'Allocation spéciale de rentrée, pour 3 enfants* Note : La valeur de 2007 est la valeur Cnaf. Les précédentes sont obtenues par rétropolation de la série par application de l'évolution Cnaf

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafARS4, 0)

*Plafond de ressources pour recevoir l'Allocation spéciale de rentrée, pour 4 enfants* Note : La valeur de 2007 est la valeur Cnaf. Les précédentes sont obtenues par rétropolation de la série par application de l'évolution Cnaf

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafARS5, 0)

*Plafond de ressources pour recevoir l'Allocation spéciale de rentrée, quand plus de 4 enfants* Note : La valeur de 2007 est la valeur Cnaf. Les précédentes sont obtenues par rétropolation de la série par application de l'évolution Cnaf

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafAB1, 0)

*Plafond de ressources pour l'allocation de Base (AB) de la PAJE (prestation d'accueil pour jeune enfant), 1 enfant* Note 1 : une condition pour valider des périodes en AVPF est d'avoir perçu des allocations dont la PAJE-AB

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafAB2, 0)

*Plafond de ressources pour l'allocation de Base (AB) de la PAJE (prestation d'accueil pour jeune enfant), 2 enfants* Note 1 : une condition pour valider des périodes en AVPF est d'avoir perçu des allocations dont la PAJE-AB

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafAB3, 0)

*Plafond de ressources pour l'allocation de Base (AB) de la PAJE (prestation d'accueil pour jeune enfant), 3 enfants* Note 1 : une condition pour valider des périodes en AVPF est d'avoir perçu des allocations dont la PAJE-AB

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafAB4, 0)

*Plafond de ressources pour l'allocation de Base (AB) de la PAJE (prestation d'accueil pour jeune enfant), 4 enfants* Note 1 : une condition pour valider des périodes en AVPF est d'avoir perçu des allocations dont la PAJE-AB

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (PlafAB5, 0)

*Plafond de ressources pour l'allocation de Base (AB) de la PAJE (prestation d'accueil pour jeune enfant), plus de 4 enfants*

Note 1 : une condition pour valider des périodes en AVPF est d'avoir perçu des allocations dont la PAJE-AB

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamFam.

— NumericVector \_ (TauxMalSP, 0)

*Taux de cotisation maladie salarié dans le secteur privé, sous le plafond de la sécurité sociale*

Source : CNAV

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxMalRSI, 0)

*Taux de cotisation maladie des indépendants*

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxMalTot)

*Taux de cotisation maladie salarié dans le secteur privé,*

Source : CNAV

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxMalPubTot)

*Taux salarié pour fonctionnaire, sur l'ensemble du salaire (traitement et toutes les primes)*

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxMalPubTr)

*Taux de cotisation maladie salarié des fonctionnaires, seulement sur le traitement et la nouvelle bonification indiciaire*

- (NBI)  
 Source :  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
- NumericVector \_ (Taux\_FDS)  
 Taux Fond de Solidarité (FP)  
 Source :  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxAGFF\_1)  
 Taux de cotisation pour Association pour le financement du fonds de financement de l'Agirc et de l'Arrco (AGFF) tranche 1  
 Source : IPP  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxAGFF\_2)  
 Taux de cotisation pour Association pour le financement du fonds de financement de l'Agirc et de l'Arrco (AGFF) tranche 2  
 Source : IPP  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxAGFF\_S1)  
 Taux de cotisation pour Association pour le financement du fonds de financement de l'Agirc et de l'Arrco (AGFF) tranche 1, Part salarié  
 Source : IPP  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxAGFF\_S2)  
 Taux de cotisation pour Association pour le financement du fonds de financement de l'Agirc et de l'Arrco (AGFF) tranche 2, Part salarié  
 Source : IPP  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxAssedic)  
 Taux de cotisation Assedic (chômage)  
 Source : IPP  
 Dernière mise à jour avec : IPP  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (Taux\_APEC)  
 Taux de cotisation de l'Association pour l'emploi des cadres (APEC)  
 Source :  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxCSGSal)  
 Taux Contribution Sociale Généralisée (CSG), part salarié, et taux de la contribution à la réduction de la dette sociale (CRDS)  
 Source : Calcul : (taux csg + taux crds) \* (1 - abatement csg sous 4 PSS)  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxMalRet)  
 Taux de cotisation maladie des retraités (existait jusqu'en 1997 incluse)  
 Note : Avant 1998, il y avait une cotisation d'assurance maladie, instituée à compter du 01/07/1980. Elle était prélevée sur la retraite des personnes soumises à l'impôt sur le revenu.  
 Source : [http://www.legislation.cnaf.fr/Pages/expose.aspx?Nom=prelevement\\_retraite\\_ancienne\\_disposition\\_ex](http://www.legislation.cnaf.fr/Pages/expose.aspx?Nom=prelevement_retraite_ancienne_disposition_ex)  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxMalRetFra)  
 Taux de cotisation maladie prélevé sur les retraites complémentaires.  
 Source : <https://www.agirc-arrco.fr/particuliers/vivre-retraite/pension-retraite-versement/#c>  
 Dernière mise à jour avec :  
 Feuille : ParamSociauxXXXX / ParamAutres.
  - NumericVector \_ (TauxMalRetEtr)  
 Taux de cotisation maladie des retraités domiciliés hors de France ou de certains retraités étrangers vivant en France.  
 Note 1 : Le retraité est soumis au prélèvement d'une cotisation d'assurance maladie s'il est domicilié fiscalement

hors de France et relève à titre obligatoire d'un régime français d'assurance maladie. La cotisation est également prélevée sur la retraite des personnes de nationalité étrangère qui ne relèvent pas à titre obligatoire d'un régime français d'assurance maladie, mais réunissent au moins 15 ans d'assurance en France. La cotisation prélevée à tort à l'assuré qui ne réunit pas 15 ans d'assurance est remboursée sans application de la prescription.

Note 2 : actuellement pas utilisé dans le modèle Destinie 2.

Source : [http://www.legislation.cnav.fr/Pages/expose.aspx?Nom=prelevement\\_retraite\\_cotisation\\_assurance\\_maladie\\_ex](http://www.legislation.cnav.fr/Pages/expose.aspx?Nom=prelevement_retraite_cotisation_assurance_maladie_ex)

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxCSGRetFort)

Taux fort de la Contribution Sociale Généralisée (CSG) + Contribution pour le remboursement de la dette sociale (CRDS) + Contribution additionnelle de solidarité pour l'autonomie (Casa)

Note 1 : La contribution sociale généralisée (CSG) est prélevée sur le montant brut des avantages de vieillesse sauf la majoration pour tierce personne et le versement exceptionnel payé en 2015. Le prélèvement est effectué par l'organisme qui paie ces avantages. La CSG est prélevée même si la retraite est payée en un versement forfaitaire unique. Il existe 2 taux différents de CSG (un taux fort et un taux minoré). Le taux de la CSG prélevée dépend du revenu fiscal de référence compte tenu du nombre de parts fiscales et du lieu de résidence. Avant 2015, le taux de la CSG dépendait de la cotisation d'impôt. La Contribution additionnelle de solidarité pour l'autonomie (Casa) ne s'applique qu'aux retraités soumis au taux fort de la CSG.

Note 2 : Ajout de la contribution de 0,3% à partir de 2013 (pour financer la dépendance)

Source : CNAV

Dernière mise à jour avec : [http://www.legislation.cnav.fr/Pages/bareme.aspx?Nom=prelevement\\_retraite\\_taux\\_prelevement\\_retraite\\_bar](http://www.legislation.cnav.fr/Pages/bareme.aspx?Nom=prelevement_retraite_taux_prelevement_retraite_bar)

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxCSGRetMin)

Taux minoré de la Contribution Sociale Généralisée (CSG) + CRDS

Note : voir TauxCSGRetFort

Source : CNAV

Dernière mise à jour avec : [http://www.legislation.cnav.fr/Pages/bareme.aspx?Nom=prelevement\\_retraite\\_taux\\_prelevement\\_retraite\\_bar](http://www.legislation.cnav.fr/Pages/bareme.aspx?Nom=prelevement_retraite_taux_prelevement_retraite_bar)

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (SeuilExoCSG)

Seuil d'exonération à la Contribution Sociale Généralisée (CSG) pour la retraite, quand 1 part fiscale

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (SeuilExoCSG2)

Seuil d'exonération à la Contribution Sociale Généralisée (CSG) pour la retraite, quand 2 parts fiscales

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (SeuilTxReduitCSG)

Seuil d'application du taux réduit de la Contribution Sociale Généralisée (CSG) quand 1 part fiscale

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (SeuilTxReduitCSG2)

Seuil d'application du taux réduit de la Contribution Sociale Généralisée (CSG) quand 2 parts fiscales

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (SeuilPauvrete)

Série des seuils de pauvreté, à 60% du niveau de vie médian

Source : INSEE/COR

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxCET\_S)

Contribution exceptionnelle temporaire (CET), part salariés

Source :

Dernière mise à jour avec :

Feuille : ParamSociauxXXXX / ParamAutres.

— NumericVector \_ (TauxCET)

Contribution exceptionnelle temporaire (CET), totale

Source :

Dernière mise à jour avec :

- *Feuille : ParamSociauxXXXX / ParamAutres.*
- `NumericVector _` (`correct_demo`, 0)  
*Correcteur démographique*  
*Source : Calculé dans la fonction interne de Destinie.cpp : destinieSimOptim. N'est utilisé que dans le cas où l'option options->coeff\_demo est mise en oeuvre. Voir [dubois\_koubi\_2017].*

### Attributs publics

- `NumericVector` [daterevalobase](#)  
*Mois de revalorisation des régimes de base.*
- `NumericVector` [daterevalocomp](#)  
*Mois de revalorisation des régimes complémentaires.*
- `NumericVector` [RevaloCumFP](#)  
*Revalorisations cumulées des pensions liquidées au FP.*
- `NumericVector` [RevaloCumRG](#)  
*Revalorisations cumulées des pensions liquidées au RG.*
- `vector< NumericMatrix >` [q\\_mort](#)  
*Tables de survie.*
- `vector< NumericMatrix >` [survie](#)  
*Tables des taux de survie.*
- `vector< NumericMatrix >` [espvie](#)  
*Tables des espérance de vie.*
- `double` [poids](#)  
*Pondération de la population.*

### 5.20.1 Description détaillée

[Macro](#) contient l'ensemble des paramètres macroéconomiques (PIB, inflation, ...) mais aussi des paramètres plus spécifiques aux retraites (taux de cotisation, valeur des points dans les complémentaires, ou encore montant du minimum vieillesse).

### 5.20.2 Documentation des données membres

#### 5.20.2.1 `q_mort`

```
vector<NumericMatrix> q_mort
```

Tables des quotients de mortalité

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/Simulation.h](#)

### 5.21 Référence de la structure `Mortadiff_dip_F`

Quotients de mortalité par âge et niveau de diplôme des femmes observés sur la période 2009-2013.

```
#include <Simulation.h>
```

## Fonctions membres publiques

- NumericVector \_ (sansdip)  
*quotients de mortalité par âge des femmes sans diplôme*
- NumericVector \_ (Brevet)  
*quotients de mortalité par âge des femmes de niveau Brevet-CEP*
- NumericVector \_ (CAPBEP)  
*quotients de mortalité par âge des femmes de niveau CAP-BEP*
- NumericVector \_ (BAC)  
*quotients de mortalité par âge des femmes de niveau baccalauréat*
- NumericVector \_ (sup)  
*quotients de mortalité par âge des femmes diplômées de l'enseignement supérieur*

## 5.21.1 Description détaillée

Le calcul de ces quotients de mortalité a été réalisée par **[blanpain2016esperance]** . Pour les années 2010 à 2012 incluses, elles ont été fournies aux fins de ces exercices de modélisation par la division Enquêtes et études démographiques de l'Insee, avec l'avertissement que certaines peuvent être fragiles pour certains âges fins. Pour cette raison, elles ne sont pas publiées. Ces valeurs sont utilisées uniquement pour compléter des séries, et ne doivent l'être qu'à ce titre.

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Simulation.h](#)

## 5.22 Référence de la structure Mortadiff\_dip\_H

Quotients de mortalité par âge et niveau de diplôme des hommes observés sur la période 2009-2013.

```
#include <Simulation.h>
```

## Fonctions membres publiques

- NumericVector \_ (sansdip)  
*quotients de mortalité par âge des hommes sans diplôme*
- NumericVector \_ (Brevet)  
*quotients de mortalité par âge des hommes de niveau Brevet-CEP*
- NumericVector \_ (CAPBEP)  
*quotients de mortalité par âge des hommes de niveau CAP-BEP*
- NumericVector \_ (BAC)  
*quotients de mortalité par âge des hommes de niveau baccalauréat*
- NumericVector \_ (sup)  
*quotients de mortalité par âge des hommes diplômés de l'enseignement supérieur*

## 5.22.1 Description détaillée

Le calcul de ces quotients de mortalité a été réalisée par **[blanpain2016esperance]** . Pour les années 2010 à 2012 incluses, elles ont été fournies aux fins de ces exercices de modélisation par la division Enquêtes et études démographiques de l'Insee, avec l'avertissement que certaines peuvent être fragiles pour certains âges fins. Pour cette raison, elles ne sont pas publiées. Ces valeurs sont utilisées uniquement pour compléter des séries, et ne doivent l'être qu'à ce titre.

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Simulation.h](#)

### 5.23 Référence de la structure Mortalite\_diff

[Mortalite\\_diff](#) permet de réaliser l'ancienne mortalité différentielle à la [\[dubois2015taux\]](#) .

```
#include <Simulation.h>
```

#### Fonctions membres publiques

- NumericVector \_ (findetH1)  
*quotient de mortalité des hommes avec un âge de fin d'études relatif inférieur à 2 ans à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetH2)  
*quotient de mortalité des hommes avec un âge de fin d'études relatif inférieur de 2 ans à 1 an à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetH3)  
*quotient de mortalité des hommes avec un âge de fin d'études relatif inférieur de au plus 1 an et supérieur d'au plus 1 an à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetH4)  
*quotient de mortalité des hommes avec un âge de fin d'études relatif supérieur d'1 an à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetF1)  
*quotient de mortalité des femmes avec un âge de fin d'études relatif inférieur à 2 ans à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetF2)  
*quotient de mortalité des femmes avec un âge de fin d'études relatif inférieur de 2 ans à 1 an à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetF3)  
*quotient de mortalité des femmes avec un âge de fin d'études relatif inférieur de au plus 1 an et supérieur d'au plus 1 an à l'âge de fin d'étude moyen, par âge au delà de 30 ans*
- NumericVector \_ (findetF4)  
*quotient de mortalité des femmes avec un âge de fin d'études relatif supérieur d'1 an à l'âge de fin d'étude moyen, par âge au delà de 30 ans*

#### 5.23.1 Description détaillée

Il s'agit des quotients de mortalité différentiel par sexe, par âge (age=a) et par âge de fin d'études relatif : findet1='<-2', findet2=['-2,-1[, findet3=['-1,1[, findet4=['1,+).

Note : Ces quotients sont appliqués au Solde Migratoire (sexe=2,age=a atteint dans l'année) et à la population totale 1er janvier (age=a-1 au 1er janvier)

Source : Des travaux de Davezies et Le Minez à partir des projections démographiques Insee 2006 et de tables CS\*findet relatif issues de Patrimoine 2003.

Dernière mise à jour avec :

Feuille : PARAM\_mortalite\_diff / mortalite\_diff

La documentation de cette structure a été générée à partir du fichier suivant :

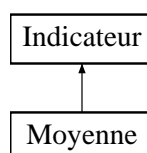
— [src/Simulation.h](#)

### 5.24 Référence de la structure Moyenne

Structure permettant de manipuler des moyennes.

```
#include <Statistiques.h>
```

Graphe d'héritage de Moyenne :



## Fonctions membres publiques

- void **push** (bool cond, double val, double poids)
- void **push** (double val)
- double **resultat** ()

## Attributs publics

- double **somme** = 0
- double **compteur** = 0

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Statistiques.h](#)

## 5.25 Référence de la structure Opt

Structure utilisée pour coder destinieFinCar.

## Fonctions membres publiques

- bool \_ (redress\_avpf)

La documentation de cette structure a été générée à partir du fichier suivant :

- src/Destinie.cpp

## 5.26 Référence de la structure Option\_optim

Structure utilisée pour coder destinieSimOptim.

## Fonctions membres publiques

- bool \_ (min\_prix)
- bool \_ (min\_smpt)
- bool \_ (prix)

La documentation de cette structure a été générée à partir du fichier suivant :

- src/Destinie.cpp

## 5.27 Référence de la structure Options

[Options](#) contient l'ensemble des options de simulation.

```
#include <Simulation.h>
```



## Fonctions membres publiques

- bool \_ (platecretMinContSMPT)
- bool \_ (coeff\_demo)
- bool \_ (ecrit\_dr\_test)
- bool \_ (cotisations)
- bool \_ (redresseSal)
- bool \_ (tirage\_simple\_naiss)
- bool \_ (FM)
- bool \_ (sante)
- bool \_ (mort\_diff\_dip)
- bool \_ (mort\_diff)
- int \_ (comp, [comp\\_tp](#))
- int \_ (anLeg, 2013)
- int \_ (AN\_MAX)
- double \_ (pas1, 3/12.0)
- double \_ (pas2, 1.0)
- double \_ (k\_median\_ho, 1.6)
- double \_ (k\_median\_fe, 1.6)
- double \_ (k\_def\_q\_ho, 0.01)
- double \_ (k\_def\_q\_fe, 0.01)
- double \_ (k\_val\_q\_ho, 1.0)
- double \_ (k\_val\_q\_fe, 1.0)
- double \_ (delta\_k, 0.01)
- double \_ (seedAlea, 12345689.0)
- double \_ (m\_incapacite, 0.0)
- bool \_ (SalNoEffetLinGen60)
- bool \_ (redresseSalNoFem)
- bool \_ (redresseCar)
- bool \_ (redresseFindet)
- bool \_ (prolongeDuree2013)
- bool \_ (effet\_hrzn)
- bool \_ (super\_effet\_hrzn)
- bool \_ (effet\_hrzn\_indiv)
- bool \_ (SalNoEffetGen)
- bool \_ (transNonCalees)
- bool \_ (Deplnact)
- bool \_ (SecondLiq)

## Options réglementaires

- bool \_ (NoRetrAntFPA)  
*Supprime le dispositif de retraites anticipées pour les fonctionnaires de catégorie active.*
- bool \_ (NoRetrAntFP3Enf)  
*Supprime le dispositif de retraites anticipées pour les fonctionnaires parents de 3 enfants. Ce dispositif a par ailleurs été supprimé par la réforme de 2010.*
- bool \_ (BlockDuree40)  
*Ne tient pas compte des réformes des retraites qui ont porté au-delà de 40 ans la durée d'assurance nécessaire à l'atteinte du taux plein.*
- bool \_ (NoAssimil)  
*Neutralise la prise en compte des périodes assimilées dans le calcul de la durée validée.*
- bool \_ (NoAVPF)  
*Neutralise les périodes d'AVPF dans le calcul de la pension au régime général (ni prise en compte de la durée validée, ni des salaires portés au compte).*
- bool \_ (NoMDA)  
*Neutralise la prise en compte des trimestres de majorations de durée d'assurance (MDA) dans le calcul de la durée d'assurance.*
- bool \_ (NoRetrAntCarrLongues)  
*Neutralise le dispositif de départ anticipé pour carrières longues.*
- bool \_ (NoInaptLiqTauxPlein)  
*Supprime le dispositif de départ au taux plein pour les inaptés.*
- bool \_ (NoBorneCoeffProrat)  
*Supprime la borne 1 pour le coefficient de proratisation.*
- bool \_ (NoPtsGratuits)  
*Neutralise le calcul et la prise en compte des points gratuits dans les régimes complémentaires du privé (chômage, préretraite et GMP pour l'AGIRC)*
- bool \_ (SAMSepare)  
*Force le calcul séparé des SAM au RG et au RSI.*
- bool \_ (NoVFU)  
*Supprime les versements forfaitaires uniques.*
- bool \_ (NoSpcAVPF)

- Neutralise la prise en compte du SMIC comme salaire porté aux comptes pour les périodes d'avpf (ces périodes restent comptabilisées dans la durée validée, comme des périodes assimilées). Les personnes qui n'ont validé que des périodes d'AVPF voient leur pension au régime général portée au niveau du minimum contributif.*
- bool \_ (NoMG)  
*Neutralise l'application du minimum garanti dans la fonction publique.*
- bool \_ (NoMC)  
*Neutralise l'application du minimum contributif au régime général et dans le régime des indépendants.*
- bool \_ (NoBonif)  
*Neutralise l'application de la bonification de pension pour trois enfants et plus (dans tous les régimes)*
- bool \_ (FPCarTot)  
*Applique à la FP un calcul du SAM identique au RG (salaires des 25 meilleurs années)*
- bool \_ (SpctoutesAnnees)  
*Calcul du SAM sur l'intégralité de la carrière.*
- bool \_ (SAMRglnUnique)  
*Calcul du SAM à partir d'un SAM unique pour le RSI et le RG.*
- bool \_ (SAMUnique)  
*Calcul du SAM à partir d'un SAM unique tous régimes.*
- bool \_ (ProlongDuree)  
*Prolonge l'augmentation des durées d'assurance pour pouvoir liquider au taux plein.*
- bool \_ (age64\_67ans)  
*Porte l'âge d'ouverture des droits à 64 ans.*
- bool \_ (age64\_69ans)  
*Porte l'âge d'ouverture des droits et l'âge d'annulation de la décôte progressivement à 69 ans.*
- bool \_ (age60\_65ans)  
*Maintient les deux bornes d'âge à 60 et 65 ans.*
- bool \_ (age60\_67ans)  
*Maintient l'âge d'ouverture des droits à 60 ans mais l'âge d'annulation de la décôte est porté à 67 ans.*
- bool \_ (BlockDureegen1960)  
*Bloque la durée d'assurance nécessaire à l'obtention du taux plein au niveau de celle appliquée à la génération 1960.*
- bool \_ (NoAccordAgircArrco)  
*Neutralise l'application de l'accord AGIRC-ARRCO de 2015 (hormis les valeurs des revalorisations de la valeur de service introduite dans le fichier de simulation)*
- bool \_ (NoRegUniqAgircArrco)  
*Ne crée pas le régime unique des complémentaires du privé à partir de 2019.*
- bool \_ (NoCoeffTemp)  
*N'applique pas les coefficients temporaires des régimes complémentaires du privé décidés dans l'accord de 2015.*
- bool \_ (NoMDAAge)  
*Ne tient pas compte de la majoration de la durée d'assurance pour âge (en y ajoutant l'option NoMDA)*

#### Options sur le mode de liquidation des assurés

- \_ (umixt)  
*Comportement de départ en fonction d'une utilité mixte.*
- \_ (uinst)  
*Comportement de départ avec utilité instantannée.*
- \_ (tp)  
*Liquidation des assurés au taux plein.*
- \_ (inapte\_exo)  
*Option utilisée dans la méthode DecoteSurcote() afin d'imputer la liquidation pour inaptitude, auquel cas celle-ci se fait sans décote. Les liquidations pour inaptitude concernent les individus dont le taux de préférence (X.k) pour l'inactivité est supérieur à un certain seuil. Lorsque inapte\_exo=TRUE, X.k=1,6.*

#### 5.27.1 Description détaillée

Les options regroupent à la fois les options réglementaires sur le calcul des droits retraites, les hypothèses de comportement de liquidations des individus ou encore le pas de simulation.

La documentation de cette structure a été générée à partir du fichier suivant :

— [src/Simulation.h](#)

## 5.28 Référence de la structure Options\_salaires

La structure [Options\\_salaires](#) contient un attribut booléen, valant 1 pour une simulation des salaires sans aléa, et 0 sinon.

```
#include <Salaires.h>
```

#### Fonctions membres publiques

- `bool _ (noAlea)`

La documentation de cette structure a été générée à partir du fichier suivant :

- `src/Salaires.h`

### 5.29 Référence de la structure OptionsTRI

La structure `OptionsTRI` contient les 3 options pour le calcul du taux de rendement interne.

```
#include <CotisationsTRI.h>
```

#### Fonctions membres publiques

- `bool _ (NoCotisationsIndirectes)`
- `bool _ (NoAllegementsCot)`
- `NumericVector _ (Calage_cotisationDirectes, 0)`

La documentation de cette structure a été générée à partir du fichier suivant :

- `src/CotisationsTRI.h`

### 5.30 Référence de la structure ordre\_findet

Compare les individus en fonction de leur âge de fin d'étude.

#### Fonctions membres publiques

- `bool operator() (const int c1, const int c2) const`

La documentation de cette structure a été générée à partir du fichier suivant :

- `src/Naissance.cpp`

### 5.31 Référence de la structure Paire

`Paire` contient les identifiants et leurs probabilités ainsi que la structure d'ordre associée.

```
#include <OutilsBase.h>
```

#### Attributs publics

- `int id`
- `double proba`

La documentation de cette structure a été générée à partir du fichier suivant :

- `src/OutilsBase.h`

### 5.32 Référence de la structure quantile

Structure permettant de manipuler des quantiles.

```
#include <Statistiques.h>
```

#### Fonctions membres publiques

- void **push** (double val)
- double **resultat** (double q)

#### Attributs publics

- deque< double > **liste**
- bool **sorted** = false

La documentation de cette structure a été générée à partir du fichier suivant :

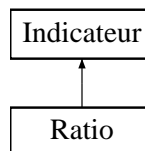
- src/[Statistiques.h](#)

### 5.33 Référence de la structure Ratio

Structure permettant de manipuler des ratios.

```
#include <Statistiques.h>
```

Graphe d'héritage de Ratio :



#### Fonctions membres publiques

- void **push** (bool cond, double val\_num, double val\_den)
- double **resultat** ()

#### Attributs publics

- double **somme\_den** = 0
- double **somme\_num** = 0

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Statistiques.h](#)

### 5.34 Référence de la classe Retraite

Pensions de droits directs et dérivés ainsi que le minimum vieillesse à l'âge courant.

```
#include <Retraite.h>
```

## Fonctions membres publiques

- **Retraite** (Indiv & X, int t)
- void **SimDir** (int age)  
La fonction **SimDir** teste la liquidation (et la primo-liquidation le cas échéant) pour un individu X pour une année donnée, ainsi que les droits directs.
- void **SimDer** (Indiv & Y, int t)  
La fonction **SimDer** initialise ou projette des droits dérivés.
- void **minvieil** (int t)  
La fonction **minvieil** calcule le montant du minimum vieillesse auquel a droit le cas échéant le ménage.
- void **tx\_cotis\_sur\_les\_retraites** (int t)  
Calcule le taux de CSG/Crds à appliquer à la pension.
- void **retraite\_nette** (int t)  
Calcul du montant de la retraite nette.
- void **revaloDir** (int t)  
Revalorisation des pensions de droits directs.
- void **revaloDer** (int t)  
Revalorisation des pensions de droits dérivés.
- double **partavtlq** (int t)  
Renvoie la part de l'année avant la première liquidation.
- double **partavttotliq** (int t)  
Renvoie la part de l'année avant la liquidation totale.
- double **reversion** (int regime)  
Renvoie la valeur annuelle au 31/12 des pensions de réversion touchées.
- double **pension** (int regime)  
Renvoie la valeur annuelle au 31/12 des pensions de droits directs touchées.
- double **pension\_prorat** (int regime)  
Renvoie la pension proratisée.
- double **revenu** ()  
Renvoie la valeur annualisée au 31/12 des revenus touchés par un individu.
- double **revenuCoupleBrut** ()  
Renvoie la valeur annualisée au 31/12 des revenus touchés par un ménage.
- double **revenuReversion** ()  
Calcule une approximation des revenus pris en compte dans la condition de revenus pour la réversion des régimes alignés.
- double **revenuReversionCouple** ()  
Calcule une approximation des revenus pris en compte dans la condition de revenus pour la réversion des régimes alignés au niveau du ménage.
- double **nbUCCouple** ()  
Calcule le nombre d'unités de consommation (UC) du ménage (sans tenir compte d'enfants à charge)
- double **pension\_age** (int age, int regime)  
Renvoie la pension perçue à un âge donné, à un régime donné (tout régime par défaut)
- double **pension\_prorat** (double pension, double daterevalo, double RevaloReg, double partavtlq=0)  
Renvoie la pension perçue au cours de l'année en tenant compte de la proratisation et de la revalorisation.
- double **pension\_prorat** (int age, int regime)  
Renvoie la pension perçue au cours de l'année en tenant compte de la proratisation et de la revalorisation.

## Attributs publics

- **Indiv & X**  
Individu X.
- int **age**  
Âge de l'individu.
- double **rev\_tot** = 0  
Réversion totale tous régimes.
- double **rev\_rg** = 0  
Réversion au Régime Général.
- double **rev\_fp** = 0  
Réversion à la FP.
- double **rev\_in** = 0  
Réversion au RSI.
- double **rev\_ar** = 0  
Réversion à l'ARRCO.
- double **rev\_ag** = 0  
Réversion à l'AGIRC.
- double **rev\_ag\_ar** = 0  
Réversion au régime unifié des complémentaires du privé

- double `min_vieil` = 0  
*montant perçu pour l'ASPA (anciennement : minimum vieillesse). N'est renseigné que pour la personne de référence du ménage*
- double `txCSG` = 0  
*taux de CSG à appliquer aux retraites de base et complémentaires*
- double `txMal` = 0  
*taux de maladie à appliquer aux retraites complémentaires*
- double `retraite_tot` = 0  
*Total des pensions de droits directs et dérivés.*
- double `retraite_net` = 0  
*Total des pensions de droits directs et dérivés nets de la CSG de la CRDS et des cotisations maladies pour les complémentaires.*
- ptr< `DroitsRetr` > `primoliq` = nullptr  
*Pointeur vers la première liquidation.*
- ptr< `DroitsRetr` > `totliq` = nullptr  
*Pointeur vers la liquidation totale.*
- ptr< `DroitsRetr` > `liq` = nullptr  
*Pointeur vers la dernière liquidation.*
- ptr< `Reversion` > `liqrev` = nullptr  
*Pointeur vers la dernière liquidation de droits dérivé*

### Pensions de droits directs

- double `pension_tot` = 0  
*Pension de droit direct total.*
- double `pension_rg` = 0  
*Pension au Régime Général.*
- double `pension_fp` = 0  
*Pension à la FP.*
- double `pension_in` = 0  
*Pension au RSI.*
- double `pension_ar` = 0  
*Pension à l'ARRCO.*
- double `pension_ag` = 0  
*Pension à l'AGIRC.*
- double `pension_ag_ar` = 0  
*Pension au regime complementaire unifié*

#### 5.34.1 Description détaillée

Pensions de droits directs et dérivés ainsi que le minimum vieillesse à l'âge courant. La classe `Retraite` est le coeur de la simulation.

#### Exemple d'utilisation :

```
Indiv& X = pop[12];

// crée un objet retraite
Retraite retraite(X, t);
// calcule les droits directs
retraite.SimDir(65);
if(retraite.primoliq && !retraite.totliq) {
    Rcout << "L'individu a effectué un primo-liquidation" << endl;
    Rcout << "Sa pension est : " << retraite.pension_tot << endl;
} else if (retraite.totliq) {
    Rcout << "L'individu a liquidé tous ses droits directs" << endl;
    Rcout << "Sa pension est : " << retraite.pension_tot << endl; *
} else {
    Rcout << "L'individu n'a pas liquidé" << endl;
}
retraite.SimDer(Y, t) // Calcul des droits dérivés
retraite.revalo(t+1) // calcul des pensions en t+1
retraite.minvieil(t+1) //calcul ASPA
```

### 5.34.2 Documentation des fonctions membres

#### 5.34.2.1 minvieil()

```
minvieil (
    int t )
```

##### Paramètres

<i>t</i>	année de projection
----------	---------------------

#### 5.34.2.2 partavtliq()

```
partavtliq (
    int t ) [inline]
```

partavtliq renvoie la part de l'année avant la première liquidation et permet le calcul de la proratisation des pensions.

##### Paramètres

<i>t</i>	année de la simulation
----------	------------------------

##### Renvoie

1 si pas encore liquidé, 0 si en retraite toute l'année

#### 5.34.2.3 partavttotliq()

```
partavttotliq (
    int t ) [inline]
```

La fonction renvoie la part de l'année avant la liquidation totale et permet le calcul de la proratisation des pensions.

##### Paramètres

<i>t</i>	année de la simulation
----------	------------------------

##### Renvoie

1 si pas encore liquidé, 0 si en retraite toute l'année

#### 5.34.2.4 pension()

```
pension (
    int regime ) [inline]
```

##### Paramètres

<i>regime</i>	régime d'affiliation
---------------	----------------------

5.34.2.5 `pension_age()`

```
pension_age (
    int age,
    int regime )
```

## Paramètres

<i>age</i>	âge
<i>regime</i>	régime servant la pension

5.34.2.6 `pension_prorat()` [1/3]

```
pension_prorat (
    int regime ) [inline]
```

## Paramètres

<i>regime</i>	régime d'affiliation
---------------	----------------------

5.34.2.7 `pension_prorat()` [2/3]

```
pension_prorat (
    double pension,
    double daterevalo,
    double RevaloReg,
    double partavtliq = 0 )
```

## Paramètres

<i>pension</i>	montant de la pension au 31/12
<i>daterevalo</i>	date de la relavoration des pensions
<i>RevaloReg</i>	valeur de la revalorisation
<i>partavtliq</i>	période de l'année avant la liquidation des droits retraites

5.34.2.8 `pension_prorat()` [3/3]

```
pension_prorat (
    int age,
    int regime )
```

## Paramètres

<i>age</i>	Âge de l'individu
<i>regime</i>	régime servant la pension

5.34.2.9 `retraite_nette()`

```
retraite_nette (
```



```
int t )
```

**Paramètres**

<i>t</i>	année de projection
----------	---------------------

**5.34.2.10 revaloDer()**

```
revaloDer (  
    int t )
```

**Paramètres**

<i>t</i>	année de projection
----------	---------------------

**5.34.2.11 revaloDir()**

```
revaloDir (  
    int t )
```

**Paramètres**

<i>t</i>	année de projection
----------	---------------------

**5.34.2.12 reversion()**

```
reversion (  
    int regime ) [inline]
```

**Paramètres**

<i>regime</i>	régime d'affiliation
---------------	----------------------

**5.34.2.13 SimDer()**

```
SimDer (  
    Indiv & Y,  
    int t )
```

**Paramètres**

<i>Y</i>	individu
<i>t</i>	année de projection

**5.34.2.14 SimDir()**

```
SimDir (  
    int age )
```

Teste la liquidation (et de primo-liquidation le cas échéant) pour un individu X pour une année donnée, ainsi que les droits directs.

Cette fonction est une alternative à `SimDirIndiv` qui teste et calcule les droits pour toute la durée de vie de l'individu. Les arguments sont :

- `X` : L'individu
- `age` : Âge de l'individu.

La fonction stocke les variables de liquidation dans le tableau `deque_liq` pour chaque liquidation.

Les pas de test de la liquidation des droits peuvent prendre les valeurs (1= annuel, 0.25 = trimestriel, 1/12=mensuel). Deux pas sont spécifiés : le premier est utilisé entre l'âge d'ouverture des droits et l'âge d'annulation de la décote au RG, le second est utilisé en dehors de cette fenêtre d'âge. L'idée est qu'on veut souvent tester la liquidation avec un pas temporel plus fin dans le premier cas que dans le second (par exemple, `pas1 = 0.25` et `pas2 = 1`).

#### Paramètres

<code>age</code>	âge en fin d'année de l'individu
------------------	----------------------------------

#### 5.34.2.15 `tx_cotis_sur_les_retraites()`

```
tx_cotis_sur_les_retraites (
    int t )
```

#### Paramètres

<code>t</code>	année de projection
----------------	---------------------

La documentation de cette classe a été générée à partir des fichiers suivants :

- `src/Retraite.h`
- `src/Retraite.cpp`

## 5.35 Référence de la structure `Retraite_comp`

Contient les éléments permettant de calculer les retraites complémentaires.

#### Attributs publics

- double `points_arrco` =0  
*Points au régime Arrco.*
- double `points_agirc` =0  
*Points au régime Agirc.*
- double `maj_points_arrco` =0  
*Nombre de points Arrco acquis pour majoration pour enfants.*
- double `maj_points_agirc` =0  
*Nombre de points Agirc acquis pour majoration pour enfants.*
- double `points_FP` =0  
*Points Arrco "fictifs" calculés si le fonctionnaire n'atteint pas la durée minimale pour toucher une pension FP.*
- double `assietteCad` =0  
*Assiette de cotisations pour l'Agirc.*
- double `assietteNC` =0  
*Assiette de cotisations pour l'Arcco.*
- double `assietteFP` =0  
*Assiette de cotisations pour la Fonction publique.*

- int `durpergratuit` =0  
*Compteur de durée de périodes "gratuites" avec ouverture de droits sans cotisations.*
- double `points_agirc_arrco` =0  
*Points au régime unifié complémentaire du privé*
- double `maj_points_agirc_arrco` =0  
*Nombre de points au régime unifié complémentaire du privé acquis pour majoration pour enfants.*

La documentation de cette structure a été générée à partir du fichier suivant :

- `src/DroitsRetr.cpp`

## 5.36 Référence de la classe Reversion

Calcul de la liquidation d'une reversion.

```
#include <Reversion.h>
```

### Fonctions membres publiques

- `Reversion` (`Indiv` &X, `Indiv` &Y, int t, int legRetroMax)

### Attributs publics

- double `rev`  
*Montant totale de la reversion à la liquidation.*
- double `rev_rg`  
*Montant de la pension de reversion au RG à la liquidation.*
- double `rev_fp`  
*Montant de la pension de reversion au FP à la liquidation.*
- double `rev_in`  
*Montant de la pension de reversion au RSI à la liquidation.*
- double `rev_ag`  
*Montant de la pension de reversion à l'AGIRC à la liquidation.*
- double `rev_ar`  
*Montant de la pension de reversion à l'ARRCO à la liquidation.*
- double `rev_ag_ar`  
*Montant de la pension de reversion dans les complémentaires unifiées à la liquidation.*
- int `ageliq_rev`  
*Âge à la liquidation.*
- int `nbEnfCharge`  
*Nombre d'enfants à charge lors du décès du conjoint.*
- int `idConj`  
*Identifiant du défunt.*

### 5.36.1 Documentation des constructeurs et destructeur

#### 5.36.1.1 Reversion()

```
Reversion (
    Indiv & X,
    Indiv & Y,
    int t,
    int legRetroMax )
```

Calcul le montant de reversion pour un individu

Calcul de la liquidation d'une reversion pour le reversataire X, au moment du décès de son conjoint Y à la date t.

La documentation de cette classe a été générée à partir des fichiers suivants :

- `src/Reversion.h`
- `src/Reversion.cpp`

## 5.37 Référence de la structure Salaire

La structure [Salaire](#) contient les équations et les fonctions permettant l'imputation des revenus d'activité.

```
#include <Salaires.h>
```

### Fonctions membres publiques

- **Salaire** (Environment &env)
- void [imput\\_statut\\_cadre](#) (int t)
 

*Impute le statut cadre à une date t. Après la phase de simulation des transitions, tous les salariés du privé ont par défaut le statut non-cadre. L'imputation du statut de cadre dépend uniquement du niveau de revenu, en respectant une cible p. Les cadres correspondent alors aux salariés du privé faisant partie des p% plus hauts salaires. Cela revient à calculer le quantile d'ordre 1-p de la distribution des salaires dans le privé et à sélectionner les salaires supérieurs à ce quantile. p est une fonction affine par morceaux de t, déterminée par les valeurs suivantes : p(1930)=5%, p(2005)=20%, p(2015)=23%, p(2025)=24% et p(2050)=25%. Le statut de cadre étant supposé relativement absorbant, les salaires des individus cadres en t-1 sont multipliés par 2, de sorte à accroître leur rang dans l'échelle des salaires et à augmenter leur chance de rester cadres.*
- double [imput\\_sal](#) (lndiv &X, int age, double ei1, double ei2, double e1, double e2)
 

*Renvoie le revenu d'activité de l'individu X à l'âge age, en fonction du sexe et du statut.*
- void [imput\\_salaires](#) ()
- void [imput\\_avpf](#) (int t)
 

*Impute le statut d'AVPF en l'année t.*
- void [input\\_taux\\_prim](#) ()
 

*Impute le taux de primes pour les titulaires de la fonction publique.*

### Attributs publics

- Environment & **env**
- vector< [EqSalaires](#) > **eqs** = Rdin<[EqSalaires](#)>("EqSalaires","NomVar",levels(FPE\_F,FPE\_H,FPHT\_F,FHT\_PHT\_H,IND\_F,IND\_H,PRI\_F\_deb,PRI\_F\_fin,PRI\_H\_deb,PRI\_H\_fin))
- CStructSexeAge **cs** = Rdin<CStructSexeAge>("CStructSexeAge")
- [Options\\_salaires](#) **options\_sal** = Rdin<[Options\\_salaires](#)>("options\_salaires")

#### 5.37.1 Description détaillée

La structure [Salaire](#) contient :

- en attributs :
  - les coefficients des variables explicatives de l'équation de salaire, et ce, pour chaque sexe et chaque secteur
  - la chronique annuelle du coefficient correcteur utilisé lors de la simulation des salaires, pour les femmes et les hommes
  - l'option sur la simulation des salaires (présence d'aléa ou non)
- en méthodes :
  - une fonction imputant le statut de cadre au cours d'une année
  - une fonction imputant le statut d'AVPF au cours d'une année
  - une fonction imputant le taux de primes dans la fonction publique en fin de carrière
  - une fonction qui recode le statut, uniquement dans le but de faciliter la simulation
  - une fonction qui calcule la durée passée dans le statut courant, sans interruption
  - une fonction qui calcule la durée passée en emploi

### 5.37.2 Documentation des fonctions membres

#### 5.37.2.1 `imput_avpf()`

```
imput_avpf (
    int t )
```

L'affiliation à l'AVPF étant soumise à la perception de certaines prestations et à une condition de ressources, cette fonction est appelée après `imput_sal`. Dans Destinie, seuls les inactifs ayant des enfants à charge sont susceptibles d'être affiliés à l'AVPF. Cette affiliation permet d'acquérir des droits à la retraite dans le régime général sans versement de cotisations : les droits sont calculés sur la base d'une rémunération fictive égale au SMIC. Le bénéfice de l'AVPF est testé sur chaque individu. La méthode `prestafam` détermine le type de prestation familiale perçue (allocation de base, complément familial, complément de libre choix d'activité) en `t`, ce qui permet à la méthode `test_avpf` de tester l'éligibilité à l'AVPF en fonction des ressources et de la prestation perçue. L'imputation de l'AVPF devient alors effective en modifiant le statut des inactifs concernés : celui-ci passe de `S_INA` à `S_AVPF`.

#### 5.37.2.2 `imput_sal()`

```
imput_sal (
    Indiv & X,
    int age,
    double ei1,
    double ei2,
    double e1,
    double e2 )
```

Les paramètres `ei1` et `ei2` (resp. `e1` et `e2`) sont les aléas utilisés pour la simulation des effets permanents (resp. du résidu annuel).

#### 5.37.2.3 `input_taux_prim()`

```
input_taux_prim ( )
```

Le taux de prime des fonctionnaires est simulé selon une distribution discrète dépendant du niveau de salaire et du sexe. Le salaire pris en compte est celui de la dernière année en qualité de fonctionnaire (hors invalidité), et les distributions du taux de prime sont déclinées suivant 9 niveaux de salaire définis par pas de `SMPT/4`. Ce taux ne dépend pas du nombre d'années dans la fonction publique, ni du type de fonction publique (FPE ou FPTH). Une fois imputé, il reste constant tout au long de la carrière de l'individu.

La documentation de cette structure a été générée à partir des fichiers suivants :

- `src/Salaires.h`
- `src/Salaires.cpp`

### 5.38 Référence de la structure Simulation

La structure `Simulation` contient l'ensemble des paramètres, options et séries macroéconomique ou de paramètres législatifs de la simulation, ainsi que le vecteur d'individus `pop` qui contient l'ensemble des individus.

```
#include <Simulation.h>
```

#### Fonctions membres publiques

- `Simulation` (Environment env)  
Constructeur d'un objet `Simulation` à partir d'un environnement `R`.
- `~Simulation` ()  
Destructeur.

## Attributs publics

- double **seed**  
*Graine pour le générateur de nombres aléatoires.*
- Environnement **envSim**  
*Environnement R contenant les paramètres de la simulation.*
- Context **context**

## Attributs publics statiques

- static double **heure\_debut** = 0  
*Heure de début de la simulation.*

La documentation de cette structure a été générée à partir des fichiers suivants :

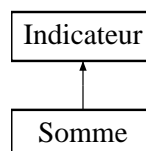
- src/[Simulation.h](#)
- src/[Simulation.cpp](#)

## 5.39 Référence de la structure Somme

Structure permettant de manipuler des sommes.

```
#include <Statistiques.h>
```

Graphe d'héritage de Somme :



## Fonctions membres publiques

- void **push** (bool cond, double val, double poids)
- void **push** (double val)
- double **resultat** ()

## Attributs publics

- double **somme** = 0

La documentation de cette structure a été générée à partir du fichier suivant :

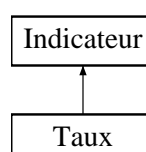
- src/[Statistiques.h](#)

## 5.40 Référence de la structure Taux

Structure permettant de manipuler des taux.

```
#include <Statistiques.h>
```

Graphe d'héritage de Taux :



## Fonctions membres publiques

- void **push** (bool cond, bool val, double poids)
- double **resultat** ()

## Attributs publics

- double **somme** = 0
- double **compteur** = 0

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Statistiques.h](#)

## 5.41 Référence de la structure Transitions

La structure [Transitions](#) contient tous les éléments utilisés dans la fonction destinieTransMdt.

```
#include <Transitions.h>
```

## Fonctions membres publiques

- [Transitions](#) (Environment &env)  
*Constructeur de la structure [Transitions](#).*
- void [make\\_transitions\\_noncalees](#) (int t, int age\_min, int age\_max)  
*Simule, sans calage, les transitions de l'année t par logits emboîtés.*
- void [make\\_transitions](#) (int t, int age\_min, int age\_max)  
*Simule, avec calage, les transitions de l'année t par logits emboîtés. Afin de mettre en oeuvre le calage, les trois premiers noeuds de l'arbre sont identiques quel que soit le statut en t-1 : transition vers l'inactivité, vers le chômage puis vers la FP.*
- void [applique\\_trans](#) (int id, int t, int position\_arbre)  
*Applique effectivement la transition en remplissant le statut de l'individu id en t. Cette fonction affine le statut de transition, jusqu'alors rangé dans un groupe.*
- void [transMigr](#) ([Indiv](#) &X, int age)  
*Simule la transition pour le migrant X à l'âge age. Pour les migrants, les transitions respectent, en coupe, des distributions de statut par sexe et par groupes d'âge (trois en tout).*
- void [ProbTrans](#) ([Indiv](#) &X, int t)  
*Calcule l'arbre de décision associé à X à la date t ainsi que les probas de transition à chaque noeud de l'arbre. L'arbre de décision dépend du statut de X en t-1.*

## Attributs publics

- Environment & **env**
- vector3< [EqTrans](#) > [eqtd](#)  
*eqtd contient les coefficients des variables de l'équation de transition, et ce, pour une période, un sexe, un statut initial et un noeud donnés.*
- [CiblesTrans](#) **cibles** = [Rdin](#)<[CiblesTrans](#)>("CiblesTrans")  
*cibles contient les cibles de taux de chômage et d'inactivité*
- int **cpt** = 0
- vector2< double > [probas](#)  
*probas est une matrice contenant les probabilités de transition de chaque individu à chaque noeud de l'arbre de décision*
- vector< double > [probas\\_fpe](#)  
*probas\_fpe contient, pour chaque individu, la probabilité de transiter vers la FPE conditionnellement au fait de transiter vers la FP*
- vector2< int > [destination](#)  
*destination contient, pour chaque individu et chaque noeud de l'arbre de décision, le statut de sortie éventuelle de l'arbre. Ce statut dépend de la période de transition, du sexe, du statut antérieur et de la profondeur de l'arbre*
- vector< int > [champ](#)  
*champ contient, pour chaque individu, une indicatrice de sortie de l'arbre de décision. Cette indicatrice vaut 1 tant que l'individu n'a pas transité, et 0 sinon.*

## 5.41.1 Description détaillée

La structure [Transitions](#) contient :

- en attributs :
  - les coefficients des variables explicatives des équations de transition
  - les cibles annuels de taux de chômage et d'inactivité, utilisés en cas de transitions calées
  - les probabilités de transition de chaque individu à chaque noeud de l'arbre de décision
  - la probabilité, pour chaque individu, de transiter vers la FPE conditionnellement au fait de transiter vers la FP
  - pour chaque individu, une indicatrice de sortie de l'arbre de décision.
  - pour chaque individu et chaque noeud de l'arbre de décision, le statut de sortie éventuelle de l'arbre.
- en méthodes :
  - une fonction simulant, sans calage, les transitions d'une année
  - une fonction simulant, avec calage, les transitions d'une année
  - une fonction qui applique effectivement la transition à un individu, en remplissant le nouveau statut
  - une fonction qui simule la transition pour un migrant
  - une fonction calculant, pour un individu et une date donnés, l'arbre de décision ainsi que les probas de transition à chaque noeud de l'arbre.

## 5.41.2 Documentation des fonctions membres

## 5.41.2.1 ProbTrans()

```
ProbTrans (
    Indiv & X,
    int t )
```

Modifie champ et probas

## 5.41.3 Documentation des données membres

## 5.41.3.1 eqtd

```
vector3<EqTrans> eqtd
```

**Valeur initiale :**

```
= Rdin<EqTrans>("EqTrans",
    "type_trans", levels(DEBUT, DEBUTH, FINF, FINH, MILIEUF, MILIEUH),
    "origine", levels(CHO, CON, INA, IND, MAL, PRI, TIT),
    "ordre", levels(TRANS1, TRANS2, TRANS3, TRANS4, TRANS5, TRANS6, TRANS7))
```

Source : estimés à l'occasion de [\[bachelet2014biographies\]](#)

La documentation de cette structure a été générée à partir des fichiers suivants :

- [src/Transitions.h](#)
- [src/Transitions.cpp](#)



## 6 Documentation des fichiers

### 6.1 Référence du fichier src/Constantes.h

Fichier contenant les constantes générales.

```
#include "OutilsBase.h"
#include "Simulation.h"
```

#### Énumérations

- enum `code_sexe` { `HOMME` =1, `FEMME` =2 }
- Codes sexe :*
- enum `code_matri` {  
`CELIB` =1, `MARIE` =2, `VEUF` =3, `SEPAR` =4,  
`CONCUBINAGE` =5 }
- Énumération des statuts matrimoniaux :*
- enum `code_dipl` {  
`SSDIPL` =1, `BREVET` =2, `CAP` =3, `BAC` =4,  
`UNIV` =5 }
- Énumération des codes diplôme :*
- enum `code_typeFP` { `FPE`, `FPTH`, `NONFP` }
- Codes de la variable individuelle typeFP.*
- enum `comportement` {  
`comp_tp`, `comp_uinst`, `comp_umixt`, `comp_exo`,  
`comp_ageMax`, `comp_uinst_old` }
- Codes de comportement de liquidation des droits directs.*
- enum `type_liq` {  
`liq_non` = 0, `liq_tp` = 1, `liq_invald` = 2, `liq_inapte` = 3,  
`liq_decote` = 4, `liq_surcote` = 5, `liq_agemax` = 6, `liq_cible` = 7,  
`liq_tp_aod` = 8, `liq_tp_duree` = 9, `liq_tp_aad` = 10, `liq_inact` = 11 }
- Type de liquidation (obsolète ?)*
- enum `regime_indicateurs` {  
`REG_TOT`, `REG_RG`, `REG_IN`, `REG_FP`,  
`REG_AR`, `REG_AG`, `REG_AGIRC_ARRCO`, `REG_FPE`,  
`REG_FPT`, `REG_AVPF` }
- Énumération des codes de régime de retraite.*
- enum `Statuts` {  
`S_NC` = 1, `S_NONTIT` = 11, `S_SCOEMP` = 12, `S_SNAT` = 13,  
`S_CAD` = 2, `S_FPAE` = 311, `S_FPSE` = 321, `S_FPAE` = 331,  
`S_FPATH` = 312, `S_FPSTH` = 322, `S_FPAATH` = 332, `S_IND` = 4,  
`S_CHO` = 5, `S_CHOBIT` = 51, `S_CHONONBIT` = 52, `S_CHOEMPL` = 53,  
`S_INA` = 6, `S_INVAL` = 62, `S_INVALRG` = 621, `S_INVALFP` = 623,  
`S_INVALIND` = 624, `S_SCO` = 63, `S_PR` = 7, `S_RET` = 8,  
`S_AVPF` = 9 }
- Énumération des modalités de statut.*
- enum `code_prestafam` { `AB` = 1, `CF3enfs` = 2, `CLCA` = 3, `AB_CLCA` = 4 }
- Énumération des prestations familiales.*
- enum `statutPrive` { `NC`, `CAD` }
- Statuts des salariés du secteur privé*
- enum {  
`FPE_F`, `FPE_H`, `FPHT_F`, `FPHT_H`,  
`IND_F`, `IND_H`, `PRI_F_deb`, `PRI_F_fin`,  
`PRI_H_deb`, `PRI_H_fin` }
- Définition des classes relatives aux équations de salaire.*
- enum `nom_eqsante` {  
`INCID_0_F`, `INCID_0_H`, `INCID_1_F`, `INCID_1_H`,  
`MORT_F`, `MORT_H`, `PREVAL_F`, `PREVAL_H` }

*Types d'équation pour la simulation d'un état de santé*

```
— enum classeagesante {
    C50_54 = 0, C55_59 = 1, C60_64 = 2, C65_69 = 3,
    C70_74 = 4, C75_79 = 5, C80_84 = 6, C85_89 = 7,
    C90 = 8 }
Définition des classes relatives à l'imputation d'un état de santé
```

## Variables

```
— const int AN_BASE = 109
    Année base de l'échantillon.
— const int AGE_MAX = 128
    Âge en vie maximal.
— const int NB_ENF_MAX = 6
    Nombre maximal d'enfants.
— const vector< string > type_liq_labels = {"Non", "taux plein", "invalid", "inapte", "decote", "surcote", "agemax", "cible
umixt", "AOD", "Duree", "AAD"}
    étiquettes associées à type_liq.
— const vector< int > Statuts_FPA = { S_FPAE, S_FPAE, S_FPATH, S_FPAATH }
— const vector< int > Statuts_FPS = { S_FPSE, S_FPSTH }
— const vector< int > Statuts_cho = { S_CHOBIT, S_CHONONBIT }
— const vector< int > Statuts_occ
— const vector< int > Statuts_act
— const vector< int > Statuts_FP = { S_FPAE, S_FPSE, S_FPAE, S_FPATH, S_FPSTH, S_FPAATH, S_INV ←
ALFP }
— const vector< int > Statuts_RG = { S_NC, S_NONTIT, S_SCOEMP, S_SNAT, S_CAD, S_CHO, S_CHOB ←
IT, S_CHONONBIT, S_CHOEMPL, S_INVAL, S_INVALRG, S_PR, S_AVPF }
— const vector< int > Statuts_IN = { S_IND, S_INVALID }
— const vector< int > Statuts_SAL = { S_NC, S_CAD, S_NONTIT }
— const vector< int > Statuts_FPAS = { S_FPAE, S_FPSE, S_FPATH, S_FPSTH }
— const vector< int > Statuts_RGIN = { S_NC, S_CAD, S_NONTIT, S_IND }
— const vector< int > Statuts_PRPRRG = { S_PR }
— const vector< int > Statuts_FPE = { S_FPAE, S_FPSE, S_FPAE }
— const vector< int > Statuts_FPT = { S_FPATH, S_FPSTH, S_FPAATH }
— const vector< int > Statuts_Priv = { S_NC, S_NONTIT, S_SCOEMP, S_SNAT, S_CAD, S_IND, S_CHO, S_C ←
HOBIT, S_CHONONBIT, S_CHOEMPL, S_INVAL, S_INVALRG, S_INVALID, S_PR, S_AVPF }
— const vector< string > STATUTS_EQS = { "FPE_F", "FPE_H", "FPHT_F", "FPHT_H", "IND_F", "IND_H", "PRI" ←
" F", "PRI_H" }
— const vector< string > EQS_SANTE = { "INCID_0_F", "INCID_0_H", "INCID_1_F", "INCID_1_H", "MORT" ←
" F", "MORT_H", "PREVAL_F", "PREVAL_H" }
```

## 6.1.1 Documentation du type de l'énumération

### 6.1.1.1 `code_dipl`

```
enum code_dipl
```

#### Valeurs énumérées

SSDIPL	sans diplôme
BREVET	brevet, CEP
CAP	CAP, BEP.
BAC	baccalauréat
UNIV	diplômé du supérieur

### 6.1.1.2 `code_matri`

```
enum code_matri
```

## Valeurs énumérées

CELIB	Statut matrimonial : modalité célibataire.
MARIE	Statut matrimonial : modalité marié(e)
VEUF	Statut matrimonial : modalité veuf/veuve.
SEPAR	Statut matrimonial : modalité séparé(e)
CONCUBINAGE	Statut matrimonial : modalité en concubinage ou pacsé(e)

## 6.1.1.3 code\_prestafam

enum `code_prestafam`

## Valeurs énumérées

AB	allocation de base pour enfant de moins de 3 ans avec condition de ressources
CF3enfs	complément familial pour 3 enfants dont le plus jeune a plus de trois ans
CLCA	prestation pour enfant de moins de trois ans sans condition ressource
AB_CLCA	cumul de l'AB et du CLCA

## 6.1.1.4 code\_sexe

enum `code_sexe`

## Valeurs énumérées

HOMME	Sexe : modalité HOMME.
FEMME	Sexe : modalité FEMME.

## 6.1.1.5 code\_typeFP

enum `code_typeFP`

## Valeurs énumérées

FPE	Type de fonction publique : modalité fonction publique d'Etat.
FPTH	Type de fonction publique : modalité fonction publique territoriale ou hospitalière.
NONFP	Type de fonction publique : modalité n'appartient pas à la fonction publique.

## 6.1.1.6 comportement

enum `comportement`

## Valeurs énumérées

comp_tp	Départ au taux plein.
comp_uinst	Départ avec Utilité instantanée cible.
comp_umixt	Départ comportement UMIXT.

## Valeurs énumérées

comp_exo	Départ à un âge exogène.
comp_ageMax	Départ à l'âge de la mise à la retraite d'office.
comp_uinst_old	Obsolète.

## 6.1.1.7 nom\_eqsante

enum `nom_eqsante`

## Valeurs énumérées

INCID_0_F	incidence chez les femmes sans incapacité
INCID_0_H	incidence chez les hommes sans incapacité
INCID_1_F	incidence chez les femmes avec incapacité
INCID_1_H	incidence chez les hommes avec incapacité
MORT_F	mortalité chez les femmes, tenant compte de l'état de santé
MORT_H	mortalité chez les hommes, tenant compte de l'état de santé
PREVAL_F	prévalence chez les femmes
PREVAL_H	prévalence chez les hommes

## 6.1.1.8 regime\_indicateurs

enum `regime_indicateurs`

## Valeurs énumérées

REG_TOT	ensemble des régimes
REG_RG	régime général
REG_IN	régime des indépendants
REG_FP	régime de la fonction publique
REG_AR	régime Arrco
REG_AG	régime Agirc
REG_AGIRC_ARRCO	régime unifié Agirc-Arrco
REG_FPE	régime de la fonction publique d'état
REG_FPT	régime des fonctions publiques territoriale et hospitalière
REG_AVPF	affiliés au titre de l'AVPF

## 6.1.1.9 statutPrive

enum `statutPrive`

## Valeurs énumérées

NC	Non Cadre.
CAD	Cadre.

## 6.1.1.10 Statuts

enum `Statuts`

## Valeurs énumérées

S_NC	Non cadre du privé
S_NONTIT	Contractuel.
S_SCOEMP	Scolaire en emploi.
S_SNAT	Service militaire.
S_CAD	Cadre du privé
S_FPAE	Titulaire de la FP d'Etat catégorie active.
S_FPSE	Titulaire de la FP d'Etat emploi sédentaire.
S_FPAE	Régimes spéciaux : FP d'état.
S_FPATH	Titulaire de la FP hospitalière ou territoriale catégorie active.
S_FPSTH	Titulaire de la FP hospitalière ou territoriale emploi sédentaire.
S_FPAATH	Régimes spéciaux : FP territoriale.
S_IND	Indépendant.
S_CHO	Chômeur.
S_CHOBIT	Chômeur.
S_CHONONBIT	Dispensé de recherche d'emploi.
S_CHOEMPL	Obsolète : cumul emploi-chômage.
S_INA	Inactif pur.
S_INVAL	Invalide.
S_INVALRG	Invalide/ Malade du secteur privé
S_INVALFP	Invalide/ Malade du secteur public.
S_INVALIND	Invalide/ Malade indépendant.
S_SCO	En scolarité
S_PR	En préretraite.
S_RET	Retraité
S_AVPF	AVPF.

## 6.1.1.11 type\_liq

enum `type_liq`

## Valeurs énumérées

liq_non	Type de liquidation : modalité Non.
liq_tp	Type de liquidation : modalité taux plein.
liq_invald	Type de liquidation : modalité invalide.
liq_inapte	Type de liquidation : modalité inapte.
liq_decote	Type de liquidation : modalité decote.

## Valeurs énumérées

liq_surcote	Type de liquidation : modalité surcote.
liq_agemax	Type de liquidation : modalité agemax.
liq_cible	Type de liquidation : modalité cible umixt.
liq_tp_aod	Type de liquidation : modalité AOD.
liq_tp_duree	Type de liquidation : modalité Durée.
liq_tp_aad	Type de liquidation : modalité AAD.
liq_inact	Type de liquidation : modalité inactif.

## 6.1.2 Documentation des variables

## 6.1.2.1 EQS\_SANTE

```
const vector<string> EQS_SANTE {"INCID_0_F", "INCID_0_H", "INCID_1_F", "INCID_1_H", "MORT_F", "MORT_H", "PREVAL_F", "PREVAL_H"}
```

Types d'équation pour la simulation d'un état de santé

## 6.1.2.2 Statuts\_act

```
const vector<int> Statuts_act
```

Valeur initiale :

```
= {S_NC, S_CAD, S_NONTIT, S_SCOEMP, S_FPAE, S_FPSE, S_FPAAE, S_FPATH, S_FPSTH, S_FPAATH, S_IND, S_CHOEMPL, S_CHOBIT, S_CHONONBIT}
```

Statuts d'activité

## 6.1.2.3 Statuts\_cho

```
const vector<int> Statuts_cho = {S_CHOBIT, S_CHONONBIT}
```

Statuts de chômage

## 6.1.2.4 STATUTS\_EQS

```
const vector<string> STATUTS_EQS {"FPE_F", "FPE_H", "FPHT_F", "FPHT_H", "IND_F", "IND_H", "PRI_F", "PRI_H"}
```

Définition des classes relatives aux équations de salaire

## 6.1.2.5 Statuts\_FP

```
const vector<int> Statuts_FP = {S_FPAE, S_FPSE, S_FPAAE, S_FPATH, S_FPSTH, S_FPAATH, S_INVALIDFP}
```

Statuts FP

### 6.1.2.6 Statuts\_FPA

```
const vector<int> Statuts_FPA = { S_FPAE, S_FPAAE, S_FPATH, S_FPAATH}
```

Statuts de la fonction publique, catégorie active

### 6.1.2.7 Statuts\_FPAS

```
const vector<int> Statuts_FPAS = {S_FPAE, S_FPSE, S_FPATH, S_FPSTH}
```

Statuts fonctionnaire en catégorie active ou sédentaire

### 6.1.2.8 Statuts\_FPE

```
const vector<int> Statuts_FPE = {S_FPAE, S_FPSE, S_FPAAE}
```

Statuts en emploi, fonction publique d'Etat

### 6.1.2.9 Statuts\_FPS

```
const vector<int> Statuts_FPS = {S_FPSE, S_FPSTH}
```

Statuts de la fonction publique, catégorie sédentaire

### 6.1.2.10 Statuts\_FPT

```
const vector<int> Statuts_FPT = {S_FPATH, S_FPSTH, S_FPAATH}
```

Statuts en emploi, fonction publique territoriale ou hospitalière

### 6.1.2.11 Statuts\_IN

```
const vector<int> Statuts_IN = {S_IND, S_INVALID}
```

Statuts d'affiliation à un régime des indépendants

### 6.1.2.12 Statuts\_occ

```
const vector<int> Statuts_occ
```

**Valeur initiale :**

```
= {S_NC, S_CAD, S_NONTIT, S_SCOEMP, S_FPAE,
    S_FPSE, S_FPAAE, S_FPATH,
    S_FPSTH,
    S_FPAATH, S_IND, S_CHOEMPL}
```

Statuts d'emploi

### 6.1.2.13 Statuts\_Priv

```
const vector<int> Statuts_Priv = {S_NC, S_NONTIT, S_SCOEMP, S_SNAT, S_CAD, S_IND, S_CHO, S_CHOBIT, S_↔
    _CHONONBIT, S_CHOEMPL, S_INVALID, S_INVALIDRG, S_INVALIDIND, S_PR, S_AVPF}
```

Statuts privé

## 6.1.2.14 Statuts\_PRPRRG

```
const vector<int> Statuts_PRPRRG = {S_PR}
```

Statuts de préretraité

## 6.1.2.15 Statuts\_RG

```
const vector<int> Statuts_RG = {S_NC, S_NONTIT, S_SCOEMP, S_SNAT, S_CAD, S_CHO, S_CHOBIT, S_CHONONB←
IT, S_CHOEMPL, S_INVAL, S_INVALRG, S_PR, S_AVPF}
```

Statuts d'affiliation au régime général

## 6.1.2.16 Statuts\_RGIN

```
const vector<int> Statuts_RGIN = {S_NC, S_CAD, S_NONTIT, S_IND}
```

Statuts d'emploi non fonctionnaire

## 6.1.2.17 Statuts\_SAL

```
const vector<int> Statuts_SAL = {S_NC, S_CAD, S_NONTIT}
```

Statuts d'emploi salarié au régime général

## 6.2 Référence du fichier src/Cotisations.h

Fichier contenant les outils permettant le calcul des cotisations prélevées sur les revenus d'activité et la pension de retraite.

## Classes

- struct [Cotisations](#)  
*La classe [Cotisations](#) contient, entre autres choses, le salaire et les cotisations retraites à une date donnée.*

## Fonctions

- double [CSGRet](#) ([Indiv](#) &X, int age)  
*[CSGRet](#) renvoie le montant de CSG prélevé sur la pension totale (droit direct et réversion) perçue par X à l'âge age.*
- double [CSGSal](#) ([Indiv](#) &X, int age)  
*[CSGSal](#) renvoie le montant de CSG prélevé sur les revenus d'activité perçus par X à l'âge age.*
- double [CotAut](#) ([Indiv](#) &X, int age)  
*Renvoie le montant des cotisations salariales, autres que la CSG et les cotisations retraite, prélevées sur les revenus d'activité perçus par X à l'âge age, à savoir :*
- double [CotRet](#) ([Indiv](#) &X, int age)  
*Renvoie le montant des cotisations retraite (part salariale) prélevées sur les revenus d'activité perçus par X à l'âge age.*
- double [CotRetrPatr](#) ([Indiv](#) &X, int age)  
*Renvoie le montant des cotisations retraite (part patronale) prélevées sur les revenus d'activité perçus par X à l'âge age.*
- double [SalMoy](#) ([Indiv](#) &X, int age, int nb\_an)  
*Renvoie le salaire moyen net entre l'âge age-nb\_an+1 et l'âge age. Les salaires pris en compte sont revalorisés sur le SMPT.*
- double [PNet](#) ([Indiv](#) &X, int age)  
*Renvoie la pension totale (droit direct et réversion) perçue par X à l'âge age, nette de la CSG et des cotisations maladies dans les complémentaires du privé*
- double [cotAGFFTot](#) ([Indiv](#) &X, int age)  
*Renvoie le montant de la cotisation AGFF (parts employeur et salariale) prélevée sur les revenus d'activité perçus par X à l'âge age.*



## 6.2.1 Documentation des fonctions

### 6.2.1.1 cotAGFFTot()

```
double cotAGFFTot (
    Indiv & X,
    int age )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	âge de l'individu

### 6.2.1.2 CotAut()

```
double CotAut (
    Indiv & X,
    int age )
```

- dans le public : la cotisation maladie, la cotisation au Fonds de solidarité
- dans le privé (non-cadre) : les cotisations au titre du chômage et de la maladie
- dans le privé (cadre) : les cotisations au titre du chômage, de la maladie, et APEC
- pour les non-salariés : la cotisation au titre de la maladie

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu

### 6.2.1.3 CotRet()

```
double CotRet (
    Indiv & X,
    int age )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu

### 6.2.1.4 CotRetrPatr()

```
double CotRetrPatr (
    Indiv & X,
    int age )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu

### 6.2.1.5 CSGRet()

```
double CSGRet (
    Indiv & X,
    int age )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu

### 6.2.1.6 CSGSal()

```
double CSGSal (
    Indiv & X,
    int age )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu

### 6.2.1.7 PNet()

```
double PNet (
    Indiv & X,
    int age )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu

### 6.2.1.8 SalMoy()

```
double SalMoy (
    Indiv & X,
    int age,
    int nb_an )
```

#### Paramètres

<i>X</i>	individu
<i>age</i>	Âge de l'individu
<i>nb_an</i>	nombre d'années prises en compte dans le calcul du salaire moyen

### 6.3 Référence du fichier src/CotisationsTRI.h

Code pour le calcul du taux de rendement interne.

```
#include "Simulation.h"
#include "DroitsRetr.h"
#include "OutilsComp.h"
#include "Reversion.h"
#include "Retraite.h"
#include "OutilsRcpp.h"
#include "range.h"
```

#### Classes

- struct [Cotisations](#)  
*La classe [Cotisations](#) contient, entre autres choses, le salaire et les cotisations retraites à une date donnée.*
- struct [OptionsTRI](#)  
*La structure [OptionsTRI](#) contient les 3 options pour le calcul du taux de rendement interne.*
- struct [CotisationsTRI](#)  
*La structure [CotisationsTRI](#) contient la fonction [CotisationsTRI](#) qui calcule l'intégralité des cotisations d'un individu.*

#### 6.3.1 Description détaillée

Ce module sur le taux de rendement interne a été développé à l'occasion du document de travail **[du-bois2015indicateurs]**.

### 6.4 Référence du fichier src/Destinie.h

contient les fonctions [calage\\_salaires](#), [destinieSim](#) et [destinieSimIndiv](#) pouvant être appelées depuis R.

#### Macros

- #define **UNDEF** 9999

#### Fonctions

- void [calage\\_salaires](#) (Environment envSim)
- void [destinieSim](#) (Environment envSim)
- void [destinieSimIndiv](#) (Environment envSim)

#### 6.4.1 Description détaillée

Ces fonctions prennent en paramètre un environnement de simulation. Celui-ci doit contenir 3 tables de données individuelles (généralement, données générées par le genebios) :

- `ech` : contient une ligne par individu
- `emp` : contient une ligne par individu \* âge
- `fam` : contient une ligne par individu \* année à partir de 2009

Il contient de plus une table `macro` contenant l'ensemble des séries macroéconomiques et des paramètres législatifs.

Enfin il contient une liste de paramètres et d'options.

## 6.4.2 Documentation des fonctions

### 6.4.2.1 calage\_salaires()

```
void calage_salaires (
    Environment envSim )
```

Fonction calage\_salaires :

Cale les salaires sur le SMPT cible

### 6.4.2.2 destinieSim()

```
void destinieSim (
    Environment envSim )
```

Fonction destinieSim : Lance une simulation en utilisant la fonction SimDir. La fonction boucle sur chaque année de simulation. Pour chaque année elle boucle sur chaque individu pour le calcul des droits directs, puis boucle à nouveau sur chaque individu pour le calcul des droits indirects et enfin boucle sur chaque individu pour le calcul du minimum vieillesse.

L'avantage de la fonction destinieSim sur la fonction destinieSimIndiv est qu'elle calcule les droits indirects et le minimum vieillesse.

### 6.4.2.3 destinieSimIndiv()

```
void destinieSimIndiv (
    Environment envSim )
```

Fonction destinieSimIndiv : Lance une simulation en utilisant la fonction SimDir. Elle calcule les droits retraites de droit direct (pas le minimum vieillesse ou les réversions). Elle boucle sur chaque individu, puis sur chaque âge.

## 6.5 Référence du fichier src/DroitsRetr.h

Calcul des droits à la retraite (cf. [DroitsRetr](#))

```
#include "Simulation.h"
#include "Indiv.h"
#include "Legislation.h"
```

### Classes

— class [DroitsRetr](#)  
*Droits directs par régime.*

## 6.6 Référence du fichier src/Indicateurs\_annee\_COR.h

Ce fichier permet de calculer des agrégats annuels par régime et par sexe, ainsi que la chronique annuelle des cotisations et du salaire net par individus. Les résultats, stockés dans les tables Indicateurs\_an, cotisations et salairenet, sont exportés dans R.

```
#include "Simulation.h"
#include "DroitsRetr.h"
#include "Reversion.h"
#include "Statistiques.h"
#include "Retraite.h"
#include "Cotisations.h"
```

### Classes

- struct `Indic_annee`  
*La structure `Indic_annee` contient des agrégats sur un régime de retraite au cours d'une année, tels que les effectifs de cotisants, de retraités, de liquidants, les montants de cotisations, de dépenses, la pension moyenne, l'âge moyen de liquidation...*

### Fonctions

- double `cotisation_retraite` (`Indiv &X`, int age, int regime)
- deque< int > `regimes_cot` (`Indiv &X`, int age)
- void `ecriture_indicateurs_an` (int t)
- void `ecriture_cotisations` ()  
*La fonction `ecriture_cotisations` exporte les cotisations prélevées (part patronale et salariale) sur chaque individu à chaque âge, dans le public, le privé et les non-salariés.*
- void `ecriture_salairenet` ()  
*La fonction `ecriture_salairenet` exporte le salaire net perçu par chaque individu à chaque âge.*

## 6.7 Référence du fichier src/Indicateurs\_gen.h

Permet d'exporter la structure éponyme avec notamment les motifs de liquidation de la pension.

```
#include "Simulation.h"
#include "Retraite.h"
#include "DroitsRetr.h"
#include "Reversion.h"
#include "Statistiques.h"
```

### Classes

- struct `Indic_gen`  
*contient des indicateurs par génération : les différents motifs de liquidation, les moyennes d'âge à la liquidation, de durée validée, taux de remplacement...*

### Fonctions

- void `ecriture_indicateurs_gen` ()  
*Calcule et exporte dans R les éléments de `Indic_gen`.*

## 6.8 Référence du fichier src/Indicateurs\_gen\_age.h

Contient la structure éponyme qui contient des indicateurs agrégés par génération et par âge.

```
#include <Rcpp.h>
#include "Simulation.h"
#include "DroitsRetr.h"
#include "Reversion.h"
#include "Statistiques.h"
```

### Classes

- struct [Indic\\_gen\\_age](#)  
*Cette structure contient les principaux indicateurs calculés par génération et par âge.*

### Fonctions

- void [ecriture\\_indicateurs\\_gen\\_age](#) (int t)  
*Cette fonction calcule les indicateurs et exporte la table R Indicateurs\_gen\_age.*

### 6.8.1 Documentation des fonctions

#### 6.8.1.1 [ecriture\\_indicateurs\\_gen\\_age\(\)](#)

```
ecriture_indicateurs_gen_age (
    int t )
```

#### Paramètres

<i>t</i>	année
----------	-------

## 6.9 Référence du fichier src/Indiv.h

Contient la définition de la classe [Indiv](#).

```
#include "Simulation.h"
```

### Classes

- class [Indiv](#)  
*Classe contenant l'ensemble des informations sur un individu.*

### Fonctions

- bool [present](#) (int id, int t)

### 6.9.1 Documentation des fonctions

#### 6.9.1.1 present()

```
bool present (
    int id,
    int t )
```

indique si une personne est présente à la date t à partir de son identifiant

## 6.10 Référence du fichier src/Legislation.h

Contient la classe [Leg](#) qui définit l'ensemble des paramètres législatifs pour un individu à un âge et une date de législation donnés.

```
#include "Simulation.h"
#include "Indiv.h"
```

### Classes

- class [Leg](#)  
*Contient les valeurs des paramètres législatifs pour un individu, un âge et une date de législation donnés.*

### Macros

- #define **UNDEF** 9999

## 6.11 Référence du fichier src/Migration.h

Migration détermine les individus qui émigrent et crée de nouveaux individus immigrants en respectant les cibles de solde migratoire par tranche d'âge.

### Fonctions

- double [multinomiale](#) (double alea, const vector< double > &vect)  
*Renvoie la réalisation d'une multinomiale définie par sa fonction de répartition.*
- int [age\\_arrive](#) (int sexe, bool estEnfant, int t)  
*Renvoie l'âge d'arrivée d'un migrant en fonction de son sexe et du fait qu'il soit enfant ou non à la date t donnée.*
- int **findet\_migrant** ([Indiv](#) &X)
- void [migrant](#) (int t)  
*Simule les migrations (entrées et sorties) à la date t.*

### 6.11.1 Documentation des fonctions

#### 6.11.1.1 age\_arrive()

```
int age_arrive (
    int sexe,
    bool estEnfant,
    int t )
```

L'âge d'arrivée des migrants est simulé suivant la distribution de l'âge d'arrivée des migrants. Cette distribution donne, pour les femmes (enfants ou adultes) et les hommes (enfants ou adultes), la répartition de l'âge par tranches quinquennales. L'âge est ensuite réparti uniformément au sein de chaque tranche.

## Paramètres

<i>sexe</i>	du migrant
<i>estEnfant</i>	'vrai' s'il s'agit d'un enfant
<i>t</i>	année de la simulation

## 6.11.1.2 migrant()

```
void migrant (
    int t )
```

L'émigration concerne la tranche des 1-60 ans, et le nombre d'émigrants est calé par tranches quinquennales, soit 10 tranches entre 1 et 60 ans. Pour chaque sexe, des enfants et des adultes sont ajoutés à l'échantillon, conformément aux cibles. Une fois qu'ils sont ajoutés, leur âge est simulé via un appel à la fonction [age\\_arrive](#). S'agissant des liens conjugaux, les migrants sont tous célibataires : aucune vie de couple ne leur est imputée. Le seul lien familial simulé concerne le lien mère/enfant. Une migrante a au plus trois enfants selon la procédure suivante. Chaque enfant se voit attribuer comme mère une femme ayant terminé ses études au plus tard un an après sa naissance, et minimisant l'écart entre son année de naissance et l'année de fin d'études. Elle devient sa mère si elle a moins de 45 ans et a terminé ses études au plus tôt dix ans avant sa naissance. A l'issue de cette boucle, les migrantes ayant au moins un enfant sont ainsi sélectionnées. Ensuite, à chaque enfant n'ayant pas de mère est assigné la première migrante ayant eu un enfant à l'étape précédente. Si cette femme a moins de 50 ans et que son premier enfant est plus jeune, elle devient la mère de l'enfant. De cette manière, les migrantes avec au moins deux enfants sont choisies. Enfin, pour chaque enfant n'ayant toujours pas de mère, on choisit une migrante avec deux enfants de l'étape précédente ayant le plus petit âge de fin d'études. Elle devient la mère si la différence d'âge est d'au plus 50 ans et si ses deux premiers enfants sont plus jeunes. Après la constitution des liens mère/enfant, l'âge de fin d'études de chaque enfant migrant est simulé via un appel à `findet_migrant`. Si un enfant n'a pas encore atteint son âge de fin d'études en *t*, il est réputé être en scolarité, comme pour les non-migrants. Dans Destinie, l'émigration concerne deux catégories : les nouveaux-nés et les 18-26 ans. Parmi les nouveaux-nés, les sorties sont simulées dans la fonction [naissance](#). Pour les 18-26 ans, chaque individu se voit attribuer une probabilité d'émigrer, fonction de l'âge (entre 18 et 25 ans pour les femmes, 18 et 26 ans pour les hommes) et du sexe. Les émigrants sont alors sélectionnés par sexe (et non par âge fin) suivant l'algorithme d'alignement par tri avec pour cible le nombre de jeunes émigrants fourni par les hypothèses démographiques.

## 6.11.1.3 multinomiale()

```
multinomiale (
    double alea,
    const vector< double > & vect )
```

## Paramètres

<i>alea</i>	à choisir entre 0 et 1
<i>vect</i>	fonction de répartition de la loi multinomiale

## 6.12 Référence du fichier src/Mortalite.h

Ce fichier contient la fonction [mortalite](#) qui simule les décès en projection.

## Fonctions

- void [mortalite](#) (int t, bool option\_mort\_diff, bool option\_mort\_tirageCale, bool option\_sante, bool option\_mort\_diff\_dip)  
*Simule la mortalité pour l'année t en fonction des options choisies.*



### 6.12.1 Documentation des fonctions

#### 6.12.1.1 mortalite()

```
void mortalite (
    int t,
    bool option_mort_diff,
    bool option_mort_tirageCale,
    bool option_sante = false,
    bool option_mort_diff_dip = false )
```

La simulation des décès consiste, pour chaque âge et chaque sexe, à attribuer une probabilité de décès aux individus présents en  $t - 1$  et à effectuer, lorsque `option_mort_tirageCale==TRUE`, un tirage d'alignement par tri en prenant pour cible  $K = (\text{probabilité de décès}) \times (\text{nombre de présents en } t - 1)$ . Une fois la liste des morts constituée, leurs éventuels conjoints deviennent veufs et pourront se remettre en couple en  $t$ . Quant aux survivants, leurs statuts professionnels et matrimoniaux en  $t - 1$  sont prolongés - ils seront éventuellement mis à jour ultérieurement via les mises en couples, séparations et transitions sur le marché du travail.

Lorsque `option_mort_diff==FALSE`, `option_mort_diff_dip==FALSE` et `option_mort_tirageCale==TRUE`, les probabilités de décès correspondent aux quotients de mortalité des projections démographiques, déclinés par âge, par année et par sexe (voir **[blanpainprojections]**).

Lorsque `option_mort_diff_dip==TRUE`, les quotients de mortalité dépendent, à partir de 31 ans, du sexe et du niveau de diplôme. Le niveau de diplôme dépend de la variable `findet`, et il est partitionné en cinq classes : sans diplôme, brevet, CAP-BEP, baccalauréat, diplôme du supérieur. En 2010, les quotients de mortalité par diplôme correspondent à ceux observés sur la période 2009-2013 (voir **[blanpain2016esperance]**). En projection, ils évoluent comme les quotients de mortalité de la population générale.

Lorsque `option_mort_diff==TRUE`, les quotients de mortalité dépendent, à partir de 31 ans, du sexe et de la tranche de `findet` (cette dernière dépendant de la génération). Cette option reproduit la mortalité différentielle par CS via une table de passage entre CS et `findet`, obtenue à partir de l'enquête Patrimoine (voir **[dubois2015taux]**).

Lorsque `option_sante==TRUE`, les quotients de mortalité dépendent d'un état de santé, défini à partir de 50 ans.

#### Paramètres

<i>t</i>	année
<i>option_mort_diff</i>	
<i>option_mort_tirageCale</i>	
<i>option_sante</i>	
<i>option_mort_diff_dip</i>	

### 6.13 Référence du fichier `src/Naissance.h`

Contient la fonction `naissance` qui simule les naissances.

#### Fonctions

- void `naissance` (int t, bool option\_mort\_tirageCale)  
*La fonction naissance comprend deux grandes parties : le choix des mères et l'initialisation des enfants.*

### 6.13.1 Documentation des fonctions

#### 6.13.1.1 naissance()

```
void naissance (
    int t,
    bool option_tirageNonCale )
```

La première partie sert à calculer les probabilités des femmes en couples d'avoir un enfant l'année t considérée selon :

- le nombre d'enfants qu'ont déjà chacun des parents
- la durée des études de la mère (soit courte soit longue)
- que l'enfant précédent de la mère a ou non le même père.

Les probabilités ont été estimées par **[duee2005modelisation]** .

Dans la deuxième partie, on sélectionne les mères potentielles avec la fonction tirage soit calé soit non-calé. Et puis, on impute les âges de fin d'étude des nouveaux bébés et complète toutes les informations nécessaires pour ajouter ces nouveaux individus dans la liste de la population. Finalement sont simulés les décès des enfants avant l'âge d'un an. Ceci n'est pas fait dans la fonction [mortalite](#) puisque cette fonction est appelée avant la fonction naissance dans la boucle temporelle. De même, le solde migratoire des enfants de moins d'un an étant négatif, on simule l'émigration des enfants de moins d'un an.

#### Paramètres

<i>t</i>	année
<i>option_mort_tirageCale</i>	choix de caler ou non le nombre de décès d'enfants de moins d'un an

10- Redressements divers :

## 6.14 Référence du fichier src/OutilsBase.h

Contient une bibliothèque de fonctions utilitaires.

```
#include <cmath>
#include <vector>
#include <string>
#include <array>
#include <deque>
#include <Rcpp.h>
#include <ctime>
#include <algorithm>
#include <memory>
#include <utility>
#include "range.h"
#include <sstream>
```

#### Classes

- struct [Paire](#)  
*Paire* contient les identifiants et leurs probabilités ainsi que la structure d'ordre associée.

## Macros

— `#define r_assert(e) (((e) ? (void)0 : Rf_error("assert false %s ligne %d : %s \n", __FILE__, __LINE__, #e));)`

## Définitions de type

— `template<typename T >`  
`using ptr = std::shared_ptr< T >`

## Énumérations

— `enum { TIRAGE_SIMPLE, TIRAGE_ESPERANCE, TIRAGE_TAUX, TIRAGE_NB }`  
`enum { SANS_REMISE, AVEC_REMISE }`

## Fonctions

— `double affn (double v, const vector< double > &u)`  
*Calcule la valeur d'une fonction affine par morceaux.*  
 — `double esc (double v, const vector< double > &u)`  
*Retourne la valeur d'une fonction constante par morceaux.*  
 — `int arr (double x)`  
*Arrondit un nombre réel à un nombre entier.*  
 — `double arr_mois (double annee)`  
*Arrondit une date au mois près stockée sous forme d'un double.*  
 — `double arr_mois (double annee, double mois)`  
*Arrondit une date au mois près stockée sous forme d'un double.*  
 — `int int_mois (double annee)`  
*Renvoie un nombre d'année sous forme d'entier.*  
 — `int int_mois (double annee, double mois)`  
*Renvoie un nombre d'année sous forme d'entier.*  
 — `bool in (int x, const vector< int > &ens)`  
*Test d'appartenance à un ensemble de taille quelconque.*  
 — `double min_max (double x, double y, double z)`  
*Borne majeure et mineure un réel.*  
 — `int trimInf (double an, double mois)`  
*Détermine le nombre de trimestres validés l'année de agetest.*  
 — `double ArrTrimSup (double x)`  
*Arrondi une valeur positive au trimestre supérieur.*  
 — `double ArrTrimInf (double x)`  
*Arrondi une valeur positive au trimestre inférieur.*  
 — `double LogLogist (double q1, double v1, double q2, double v2, double alea)`  
*fonction LogLogist*  
 — `double part (double base, double seuil1, double seuil2)`  
*Calcule la part d'un nombre comprise entre deux seuils.*  
 — `double partTx (double base, double seuil1, double seuil2)`  
*Calcule la part d'un nombre comprise entre deux seuils.*  
 — `double Logist (double mu, double se, double x)`  
*Fonction réciproque de la fonction de répartition logistisque.*  
 — `double Pareto (double seuil, double param, double x)`  
*Fonction réciproque de la fonction de répartition de Pareto.*  
 — `double Multinom (double x, const vector< double > &v)`  
*Fonction réciproque de la fonction de répartition d'une loi multinomiale.*  
 — `vector< double > lisseAlea (vector< int > &strate1, vector< int > &strate2, double alea)`  
*Renvoie un tableau d'aléas lissés.*  
 — `template<typename T >`  
`T : :value_type sum (const T &vect)`  
 — `bool greatP (Paire &x, Paire &y)`  
*Indique si la première **Paire** a une probabilité supérieure à la seconde **Paire**.*  
 — `template<typename T1 , typename T2 >`  
`void order (T1 &val, T2 &val2, bool reverse=false)`  
 — `deque< int > tirage (vector< int > champ, vector< double > &probas, int type_cible, double cible, string label="")`

- deque< int > **tirage** (vector< double > &probas, int type\_cible, double cible, string label="")
- vector< bool > **select** (vector< int > champ, vector< double > &probas, int type\_cible, double cible, string label="")
- vector< bool > **select** (vector< double > &probas, int type\_cible, double cible, string label="")
- template<int N>  
vector< array< int, 2 > > **appariement** (const deque< array< double, N > > &candidats1, const deque< array< double, N > > &candidats2, vector< double > poids, int option)
- vector< vector< double > > **cholesky** (vector< vector< double > > A)
- vector< double > **vecteur\_gaussien** (vector< double > m, vector< vector< double > > L)

#### 6.14.1 Description détaillée

Bibliothèque de fonctions utilitaires, en particulier pour la gestion des dates au mois près.

Les dates et les âges dans Destinie sont exprimés pour la liquidation des droits directs au mois près, et sous format de nombres à virgules flottantes (doubles) avec pour unité le nombre d'années. Pour gérer les problèmes d'arrondis, et effectuer des tests d'égalité on utilise en particulier les fonctions [arr\\_mois](#) et [int\\_mois](#).

#### 6.14.2 Documentation des macros

##### 6.14.2.1 r\_assert

```
#define r_assert(
    e ) ((e) ? (void)0 : Rf_error("assert false %s ligne %d : %s \n", __FILE__, __←
LINE__, #e));
```

[Macro](#) testant une assertion et arrêtant le programme si l'assertion est invalide

#### 6.14.3 Documentation des fonctions

##### 6.14.3.1 affn()

```
double affn (
    double v,
    const vector< double > & u ) [inline]
```

Calcule la valeur en un point donné d'une fonction affine par morceaux. On donne le point où on souhaite l'évaluer et un nombre arbitraire de paires xi,yi (avec x croissant) correspondant aux points anguleux. La fonction est supposée constante avant et après le dernier point fournis

Exemple : `double y = affn(x, {2010,0,2020,10});` y vaut 0 pour  $x \leq 2010$ , 3 pour  $x = 2013$ , 10 pour  $x \geq 2020$

##### 6.14.3.2 arr\_mois() [1/2]

```
double arr_mois (
    double annee ) [inline]
```

Arrondit une date précise passée sous forme d'un double de façon à pouvoir effectuer des comparaisons de dates au mois près. En effet la manière dont sont calculés les nombres à virgule flottante peuvent entrainer des phénomènes d'arrondis de calcul tels que par exemple  $20 + 1/12 \neq 10 + 121/12$ . `arr_mois` assure que `arr_mois(20 + 1/12) == arr_mois(20 + 121/12)` renverra le bon résultat.

**6.14.3.3 arr\_mois()** [2/2]

```
double arr_mois (
    double annee,
    double mois ) [inline]
```

Idem première fonction arr\_mois, avec comme paramètre supplémentaire le nombre de mois sous forme d'entier.

```
arr_mois(60+4/12.0) == arr_mois(60,4) == arr_mois(50+124/12.0)
```

**6.14.3.4 esc()**

```
double esc (
    double v,
    const vector< double > & u ) [inline]
```

Calcule la valeur en un point donné d'une fonction constante par morceaux. On donne le point où on souhaite l'évaluer et un nombre arbitraire de paires  $x_i, y_i$  (avec  $x$  croissant) correspondant aux points de discontinuités. La fonction est supposée constante avant et après le dernier point fournis

Exemple : `double y = esc(x, {0, 2010, 5, 2020, 10});` y vaut 0 pour  $x < 2010$ , 5 pour  $2010 \leq x < 2020$ , 10 pour  $x \geq 2020$

**6.14.3.5 in()**

```
bool in (
    int x,
    const vector< int > & ens ) [inline]
```

Test d'appartenance à un ensemble de taille quelconque

**Renvoie**

true si  $x$  est présent dans le vecteur ens, false sinon

**Paramètres**

<i>x</i>	entier dont l'appartenance est testé
<i>ens</i>	vecteur de l'ensemble des valeurs auquel on teste l'appartenance

**6.14.3.6 int\_mois()** [1/2]

```
int int_mois (
    double annee ) [inline]
```

Renvoie un nombre d'année sous forme d'entier : `int_mois(60+10/12.0) == 60`

**6.14.3.7 int\_mois()** [2/2]

```
int int_mois (
    double annee,
    double mois ) [inline]
```

Renvoie un nombre d'année sous forme d'entier : `int_mois(60+10/12.0) == 60`

Remarque : Un individu atteindra l'âge de 60 ans et 6 mois au cours de l'année : `int_mois(X.anaiss+60, 6 + X.moisnaiss)`

## 6.14.3.8 Logist()

```
double Logist (
    double mu,
    double se,
    double x ) [inline]
```

Fonction réciproque de la fonction de répartition logistique.

Exemple : tirage d'une v.a.  $y$  suivant une Pareto : `double y = Logist(1, 1/3.0, S.alea())`

## 6.14.3.9 LogLogist()

```
double LogLogist (
    double q1,
    double v1,
    double q2,
    double v2,
    double alea ) [inline]
```

Tirage d'une VA selon une loi LogLogistique de fonction de répartition  $z^{**a}/(b+z^{**a})$  paramétrée par deux quelconques de ses quantiles. Par exemple `x=LogLogist(.25, 2, .5, 3, S.alea())` tirera une VA à valeurs positives dont le premier quartile est à 2 et la médiane est à 3.

Si l'on ajoute un argument supplémentaire, cet argument est pris comme valeur de l'aléa par défaut. Sinon, un aléa est tiré selon une loi uniforme.

## 6.14.3.10 min\_max()

```
double min_max (
    double x,
    double y,
    double z ) [inline]
```

## Renvoie

$x$  si  $y \leq x \leq z$ ,  $y$  si  $x < y$  et  $z$  si  $x > z$

## Paramètres

$x$	valeur initiale
$y$	borne inférieure
$z$	borne supérieure

## 6.14.3.11 Multinom()

```
double Multinom (
    double x,
    const vector< double > & v ) [inline]
```

Fonction réciproque de la fonction de répartition d'une loi multinomiale.

Exemple : tirage d'une v.a.  $y$  suivant une loi multinomiale

`double y = Multinom(S.alea(), {1/3.0, 1/4.0})` renvoie 1 avec une proba 1/3. renvoie 2 avec une proba 1/4. renvoie 3 avec une proba 5/12.

#### 6.14.3.12 Pareto()

```
double Pareto (
    double seuil,
    double param,
    double x ) [inline]
```

Fonction réciproque de la fonction de répartition de Pareto.

Exemple : tirage d'une v.a.  $y$  suivant une Pareto :

```
double y = Pareto(1, 1/3.0, S.alea())
```

#### 6.14.3.13 part()

```
double part (
    double base,
    double seuil1,
    double seuil2 ) [inline]
```

Calcule la part d'un nombre  $x$  comprise entre deux seuils  $s1$  et  $s2$ , c'est-à-dire 0 si  $x < s1$ ,  $x-s1$  si  $x$  entre  $s1$  et  $s2$  et  $s2-s1$  si  $x > s2$ .

#### 6.14.3.14 partTx()

```
double partTx (
    double base,
    double seuil1,
    double seuil2 ) [inline]
```

Calcule la part d'un nombre  $x$  comprise entre deux seuils  $s1$  et  $s2$ , c'est-à-dire 0 si  $x < s1$ ,  $(x-s1)/(s2-s1)$  si  $x$  entre  $s1$  et  $s2$  et 1 si  $x > s2$ .

### 6.15 Référence du fichier src/OutilsComp.h

```
#include "Retraite.h"
```

#### 6.15.1 Description détaillée

Cette bibliothèque propose des outils de simulation des comportements de départ en retraite. Elle utilise l'information d'un objet de [DroitsRetr](#) pour déterminer, selon le comportement choisi, si l'individu choisit de liquider ses droits directs à l'âge testé ou de reporter son départ. Du fait de la possibilité de liquider en deux fois (liquidation d'abord à la FP puis ensuite hors FP), cette bibliothèque contient deux fonctions de test de liquidation.

La première, `TestLiq`, teste la première liquidation. Si les droits ne sont pas intégralement liquidés à la première liquidation, la seconde fonction, `TestSecondeLiq`, teste la seconde liquidation.

Contrairement à [DroitsRetr](#), les fonctions de cette bibliothèque sont des outils généraux qui pourraient être utilisés pour simuler des comportements dans des régimes entièrement différents.

## 6.16 Référence du fichier src/Retraite.h

Contient la définition de la classe [Retraite](#) et l'implémentation de quelques fonctions du module retraite.

```
#include "Simulation.h"
#include "DroitsRetr.h"
#include "Reversion.h"
#include "Indiv.h"
```

### Classes

- class [Retraite](#)  
*Pensions de droits directs et dérivés ainsi que le minimum vieillesse à l'âge courant.*

## 6.17 Référence du fichier src/Reversion.h

Simule la liquidation de la pension de réversion.

```
#include "Simulation.h"
#include "DroitsRetr.h"
```

### Classes

- class [Reversion](#)  
*Calcul de la liquidation d'une reversion.*

## 6.18 Référence du fichier src/Salaires.h

Fichier contenant les outils permettant de simuler les salaires, à la fois en rétrospectif et en projection.

```
#include "Simulation.h"
#include "Indiv.h"
```

### Classes

- struct [EqSalaires](#)  
*La structure [EqSalaires](#) contient les coefficients des variables explicatives des équations de salaires, pour un sexe et un secteur donnés.*
- struct [Options\\_salaires](#)  
*La structure [Options\\_salaires](#) contient un attribut booléen, valant 1 pour une simulation des salaires sans aléa, et 0 sinon.*
- struct [CStructSexeAge](#)  
*La structure [CStructSexeAge](#) contient la série des coefficients correcteurs utilisés lors de la simulation des salaires, pour les femmes et les hommes.*
- struct [Salaire](#)  
*La structure [Salaire](#) contient les équations et les fonctions permettant l'imputation des revenus d'activité.*



## 6.19 Référence du fichier src/Sante.h

Imputation d'un état de sante pour les plus de 50 ans (adl, iadl, gali).

### Fonctions

- void **evolsante** (int t)  
*Opère les transitions pour les personnes ayant déjà un état de santé.*
- void **imputsante** (int t)
- void **premiere\_sante** (int t, vector< int > &champ, int sexe)  
*Impute le premier état de santé pour le vecteur des identifiants passés dans champ.*

### 6.19.1 Description détaillée

Ce module a été implémenté à l'occasion de l'étude sur l'espérance de vie sans incapacité sévère réalisée par [evrsi] .

### 6.19.2 Documentation des fonctions

#### 6.19.2.1 evolsante()

```
void evolsante (
    int t )
```

#### Paramètres

<i>t</i>	année
----------	-------

#### 6.19.2.2 premiere\_sante()

```
void premiere_sante (
    int t,
    vector< int > & champ,
    int sexe )
```

#### Paramètres

<i>t</i>	année
<i>champ</i>	vecteur des identifiants
<i>sexe</i>	précise le sexe des individus auxquels on attribue un état de santé.

## 6.20 Référence du fichier src/Separations.h

Contient les fonctions simulant les mises en couple et les séparations.

## Fonctions

- int `duree_etat` (`Indiv &X`, int `age`)  
*Renvoie la durée passée dans le statut matrimonial actuel.*
- int `NEnfunion` (`Indiv &X`, int `age`)
- void `mise_en_couple` (int `t`)  
*La fonction mise\_en\_couple choisit les personnes seules qui vont se mettent en couple et les appariant.*
- void `separation` (int `t`)  
*La fonction separation sert à déterminer les couples qui se séparent.*

## 6.20.1 Documentation des fonctions

6.20.1.1 `duree_etat()`

```
int duree_etat (
    Indiv & X,
    int age )
```

## Paramètres

<i>X</i>	l'individu
<i>age</i>	son âge

6.20.1.2 `mise_en_couple()`

```
void mise_en_couple (
    int t )
```

Il y a deux grandes parties dans la fonction `mise_en_couple`. D'abord le calcul de la probabilité de mettre en couple ou remettre en couple pour chaque hommes et chaque femme de l'échantillon. Puis l'appariement entre femmes et hommes. Pour réaliser l'appariement, on calcule un score basé sur la différence d'âge et la différence d'âge de fin d'étude pour respecter l'homogamie des couples. Puis successivement à chaque femme ayant été choisie comme se mettant en couple, on attribue le partenaire le plus proche au sens du score précédemment calculé. Une fois le couple formé, les caractéristiques des deux membres du couple sont mises à jour.

Les estimations des paramètres de cette fonction ont été faites par **[duee2005modelisation]**

## Paramètres

<i>t</i>	année
----------	-------

6.20.1.3 `separation()`

```
void separation (
    int t )
```

On calcule la probabilité de se séparer pour chaque femme en fonction de la durée du couple et du nombre d'enfants. En suite, on réalise un tirage simple des femmes qui vont se séparer. Puis on met à jour ses informations et celles de son ex-conjoint.

Les estimations des paramètres de cette fonction ont été faites par **[duee2005modelisation]**

## Paramètres

<i>t</i>	année
----------	-------

## 6.21 Référence du fichier src/Simulation.h

Fichier contenant la structure [Simulation](#), qui importe les paramètres et l'échantillon depuis les objets en R et les rend accessibles en C avec des objets globaux (accessibles dans tout le programme).

```
#include "OutilsBase.h"
#include "range.h"
#include "Constantes.h"
#include "OutilsRcpp.h"
```

## Classes

- struct [CiblesDemo](#)  
*CiblesDemo* contient les cibles démographiques de naissances et de solde migratoire.
- struct [FinEtudeMoy](#)  
*FinEtudeMoy* désigne l'âge moyen de fin d'études par sexe et par génération.
- struct [Mortalite\\_diff](#)  
*Mortalite\_diff* permet de réaliser l'ancienne mortalité différentielle à la [\[dubois2015taux\]](#).
- struct [Mortadiff\\_dip\\_F](#)  
Quotients de mortalité par âge et niveau de diplôme des femmes observés sur la période 2009-2013.
- struct [Mortadiff\\_dip\\_H](#)  
Quotients de mortalité par âge et niveau de diplôme des hommes observés sur la période 2009-2013.
- struct [Ech](#)  
*Ech* contient les caractéristiques invariantes au cours du temps des individus.
- struct [Emp](#)  
*Emp* contient les statuts sur le marché du travail et les revenus d'activité
- struct [Fam](#)  
*Fam* contient la chronique des événements familiaux depuis 2009.
- struct [Options](#)  
*Options* contient l'ensemble des options de simulation.
- struct [Macro](#)  
*Macro* contient l'ensemble des paramètres macroéconomiques et des paramètres spécifiques aux retraites.
- struct [EqSante](#)  
*EqSante* contient les coefficients pour les équations de santé (adl, iadl, gali).
- struct [Simulation](#)  
La structure [Simulation](#) contient l'ensemble des paramètres, options et séries macroéconomique ou de paramètres législatifs de la simulation, ainsi que le vecteur d'individus *pop* qui contient l'ensemble des individus.

## Fonctions

- void [ecriture\\_liquidations](#) ()  
Crée une table R liquidations.
- void [ecriture\\_retraites](#) (int t)  
Exporte dans la table R retraite l'ensemble des pensions versées dans l'année.
- void [ecriture\\_droitsRetr](#) ([Indiv](#) &X, [DroitsRetr](#) &r)
- void [pointage](#) (string str)

## Variables

- ptr< [Macro](#) > [M](#)  
*pointeur qui contient les paramètres macroéconomiques*
- ptr< [Options](#) > [options](#)  
*pointeur qui contient les options de législation*
- ptr< [Mortalite\\_diff](#) > [mortalite\\_diff](#)
- ptr< [Mortadiff\\_dip\\_F](#) > [mortadiff\\_dip\\_F](#)
- ptr< [Mortadiff\\_dip\\_H](#) > [mortadiff\\_dip\\_H](#)
- ptr< [FinEtudeMoy](#) > [finEtudeMoy](#)
- ptr< [CiblesDemo](#) > [ciblesDemo](#)
- vector< [Indiv](#) > [pop](#)  
*tableau de l'ensemble des individus (cf. classe [Indiv](#))*
- int [AN\\_FIN](#)  
*Année de fin de la simulation.*
- int [AN\\_NB](#)  
*Nombre d'années (AN\_FIN+1)*
- vector< [EqSante](#) > [eq\\_sante](#)  
*Equations régissant les transitions entre états de santé dans l'option santé*

## 6.21.1 Description détaillée

Les objets globaux correspondant sont :

- [pop](#) : vector d'objets individus contenant l'information pour chaque individu (info issue des tables ech, emp et fam)
- [M](#) : pour les séries macroéconomiques et les séries de paramètres législatifs
- [options](#) : [Options](#) spécifiques à une simulation
- [mortalite\\_diff](#) : table de la mortalité différentielle
- [finEtudeMoy](#) : table des âges de fin d'étude moyens par génération
- [ciblesdemo](#) : table des cibles démographiques
- [AN\\_FIN](#) : dernière année de simulation

Lorsque l'objet [Simulation](#) est détruit les tables R créées avec les objets Rdout sont exportées en R. Usage :

```
// [[Rcpp::export]]
void destinie(Environnement envSim) {
  // double smpt = M->SMPT[109]; << donnerait une erreur
  auto S = Simulation(envSim);
  double smpt = M->SMPT[109]; // Ok
  // A la fin de la fonction les tables R sont exportées et tous les objects C sont détruits
}
```

## 6.21.2 Documentation des fonctions

## 6.21.2.1 pointage()

```
void pointage (
    string str ) [inline]
```

Affiche la durée écoulée depuis le début de la simulation

## 6.21.3 Documentation des variables

## 6.21.3.1 eq\_sante

```
vector<EqSante> eq_sante
```

Equations régissant les transitions entre états de santé dans l'option santé

### 6.21.3.2 pop

```
vector<Indiv> pop
```

tableau de l'ensemble des individus (cf. classe [Indiv](#))

## 6.22 Référence du fichier src/Statistiques.h

Statistiques contient des structures qui facilitent la manipulation des indicateurs suivants : moyenne, somme, taux, ratio, quantile. Ces structures sont notamment utilisées dans [Indicateurs\\_annee\\_COR.h](#).

```
#include <vector>
#include <algorithm>
```

### Classes

- struct [Indicateur](#)  
*Structure à l'origine des structures [Moyenne](#), [Ratio](#), [Somme](#) et [Taux](#).*
- struct [Moyenne](#)  
*Structure permettant de manipuler des moyennes.*
- struct [Somme](#)  
*Structure permettant de manipuler des sommes.*
- struct [Taux](#)  
*Structure permettant de manipuler des taux.*
- struct [Ratio](#)  
*Structure permettant de manipuler des ratios.*
- struct [quantile](#)  
*Structure permettant de manipuler des quantiles.*

## 6.23 Référence du fichier src/Transitions.h

Fichier contenant la structure [Transitions](#).

```
#include "Simulation.h"
```

### Classes

- struct [EqTrans](#)  
*[EqTrans](#) contient les équations de transitions sur le marché du travail.*
- struct [CiblesTrans](#)  
*[CiblesTrans](#) contient les cibles de calage sur le marché du travail.*
- struct [Transitions](#)  
*La structure [Transitions](#) contient tous les éléments utilisés dans la fonction [destinieTransMdt](#).*

## Énumérations

- enum {  
**CHO, CON, INA, IND,**  
**MAL, PRI, TIT, XXX,**  
**RET, PRE }**  
*énumération des groupes dans lesquels sont rangés les statuts. Les équations de transitions gouvernent les transitions entre ces groupes. Les statuts finaux sont affinés par la fonction applique\_trans.*
- enum {  
**TRANS1, TRANS2, TRANS3, TRANS4,**  
**TRANS5, TRANS6, TRANS7, CHO\_H1519,**  
**CHO\_H2024, CHO\_H2529, CHO\_H3034, CHO\_H3539,**  
**CHO\_H4044, CHO\_H4549, CHO\_H5054, CHO\_F1519,**  
**CHO\_F2024, CHO\_F2529, CHO\_F3034, CHO\_F3539,**  
**CHO\_F4044, CHO\_F4549, CHO\_F5054, INA\_H1519,**  
**INA\_H2024, INA\_H2529, INA\_H3034, INA\_H3539,**  
**INA\_H4044, INA\_H4549, INA\_H5054, INA\_F1519,**  
**INA\_F2024, INA\_F2529, INA\_F3034, INA\_F3539,**  
**INA\_F4044, INA\_F4549, INA\_F5054, FP\_M60,**  
**FPE\_M60, NBTRANS }**  
*énumération contenant les labels :*
- enum {  
**DEBUTF, DEBUTH, FINF, FINH,**  
**MILIEUF, MILIEUH, NBTYPE\_TRANS }**  
*énumération contenant les labels des périodes de transition (début, milieu et fin de carrière) pour les deux sexes.*

## 6.23.1 Description détaillée

[Transitions](#) contient à la fois les imports des équations de transition sur le marché du travail et des données de calage. Elle contient aussi les fonctions réalisant les transitions ainsi définies. Usage :

```
// [[Rcpp::export]]
void destinieTransMdt(Environnement envSim) {
  // A la fin de la fonction les tables R sont exportées et tous les objects C sont détruits
}
```

## 6.23.2 Documentation du type de l'énumération

## 6.23.2.1 anonymous enum

anonymous enum

- du noeud de l'arbre de décision
- des cibles de taux de chômage
- des cibles de taux d'inactivité

## 7 Guide de prise en main du modèle Destinie

## 7.1 Lancer une projection avec le modèle Destinie

Ce guide vise à faciliter la première prise en main du modèle, tant pour des utilisateurs voulant utiliser le modèle dans sa forme actuelle (voir parties [7.1.1](#) et [7.1.2](#)), que pour des utilisateurs plus experts voulant accéder au code source.

### 7.1.1 Installation minimale

#### 1. Installations préalables :

- Installer R (<https://cran.r-project.org/>)
- Installer Rtools (<https://cran.r-project.org/bin/windows/Rtools/>), si votre système d'exploitation est Windows pour pouvoir installer le package destinie à partir des fichiers source. Choisir la version de Rtools compatible avec la version de R choisie précédemment.

#### 2. Installations recommandées :

- Installer Rstudio (<https://www.rstudio.com/products/rstudio/download/>), environnement de travail R auquel il est parfois fait référence dans la documentation.
- Installer git (<https://git-scm.com/>) de façon à pouvoir suivre les futures mises à jour du modèle.
- Installer par exemple TortoiseGit (<https://tortoisegit.org/>) pour pouvoir utiliser Git en "clic-bouton".
- Disposer d'un tableur (tel Microsoft Excel ou LibreOffice Calc).

#### 3. Installer les packages nécessaires à Destinie :

---

```
install.packages(c("devtools","pkgbuild"))
```

---

#### 4. Installation du package Destinie :

L'installation se faisant à partir des fichiers sources, il est nécessaire pour les utilisateurs de Windows de lancer au préalable :

---

```
devtools::find_rtools()  
# ou pkgbuild::find_rtools()
```

---

pour obtenir un "TRUE".  
Puis

---

```
devtools::install_github("InseeFr/Destinie-2",ref="release")
```

---

(par défaut les packages dont dépend destinie seront installés)

### 7.1.2 Réaliser une première simulation

Dans le logiciel R,

1. On charge le package : `library(destinie)`
2. On lance une simulation exemple : `demo(simulation,package="destinie",encoding="utf8")` Le fichier source `simulation.R` du répertoire `demo` permet de lancer une simulation à partir d'un échantillon test non représentatif<sup>2</sup> en choisissant des hypothèses classiques.  
Le déroulé du programme est le suivant :

---

```
#####  
#echantillon de depart  
#####  
library (destinie)  
data("test")  
simul=test  
rm(test)  
#####  
#remplacer ici test par l'échantillon disponible sur Quetelet
```

---

2. Attention : Le fichier `test` fourni pour prendre en main le logiciel n'est pas représentatif de la population au 1er janvier 2010. Le fichier représentatif à utiliser est mis à disposition sur la plateforme Quetelet (<http://quetelet.progedo.fr/>).

```
#####

#####
# Champ de la simulation
#####
champ<-"FE" # ou FM
fin_simul<-2070 #2110 au maximum ou 2070 plus classiquement

#####
#choix d'options
#####
simul$options_salaires <- list()

simul$options <- list("tp", anLeg=2016, pas1=3/12, pas2=1,
                     AN_MAX=as.integer(fin_simul), champ,
                     NoAccordAgircArrco=F, NoRegUniqAgircArrco=T,
                     SecondLiq=F, mort_diff_dip=T, effet_hrzn=T)
                     # pas1=1/12 (pas mensuel pour proj COR)

#####
#choix du scenario demographique
#####
# ici la fecondite, l'esperance de vie et le solde migratoire suivent le scenario central des
# projections de l'Insee
# pour la France entiere (attention a la coherence avec le champ precedemment choisi)
# deux autres scenarios sont deja crees le premier ou tous les scenarios sont a bas et ce qui aboutit
# a une population agee
# le second tous les scenarios sont a haut et ce qui aboutit a une population jeune
# les autres scenarios s'obtiennent en utilisant le programme \data_raw\obtention_hypdemo.R
data("fec_Cent_vie_Cent_mig_Cent")
demo=fec_Cent_vie_Cent_mig_Cent
rm(fec_Cent_vie_Cent_mig_Cent)

#####
#chargement des equations regissant le marche du travail, de sante
#####

data("eq_struct")

#####
#chargement des parametres economiques puis projection des parametres dans le futur
#####

data("eco_cho_7_prod13")
eco=eco_cho_7_prod13
rm(eco_cho_7_prod13)
eco$macro <-eco$macro%>%
mutate(
  SMPTp = ifelse(is.na(SMPTp),0,SMPTp),
  SMICp = ifelse(is.na(SMICp),0,SMICp),
  PIBp = ifelse (is.na(PIBp),0,PIBp),
  PlafondSSp = ifelse(is.na(PlafondSSp),0,PlafondSSp),
  Prixp = ifelse (is.na(Prixp),0,Prixp),
  MinPRp = 1.02,
  RevaloRG = ifelse(is.na(RevaloRG),1+Prixp,RevaloRG),
```



```

RevaloFP = ifelse(is.na(RevaloFP),1+Prixp,RevaloFP),
RevaloSPC = ifelse(is.na(RevaloSPC),1+Prixp,RevaloSPC)
) %>%
projection (
  SMPT ~ cumprod((1+SMPTp)*(1+Prixp)),
  PIB ~ cumprod((1+PIBp)*(1+Prixp)),
  PlafondSS ~ cumprod((1+PlafondSSp)*(1+Prixp)),
  SMIC ~ cumprod((1+SMICp)*(1+Prixp)),
  Prix ~ cumprod(1+Prixp),
  PointFP|PlafRevRG ~ SMPT,
  SalValid ~ SMIC,
  PlafARS1|PlafARS2|PlafARS3|PlafARS4|PlafARS5|PlafCF3|PlafCF4|PlafCF5|
  MajoPlafCF|sGMP|BMAF|SeuilPauvrete ~ SMPT,
  MaxRevRG ~ PlafondSS,
  MinPR ~ cumprod(MinPRp*(1+Prixp)),
  MinVieil1|MinVieil2|Mincont1|Mincont2 ~ lag(cumprod(1+Prixp)), # indexation standard. En
  evolution, indexation sur l' inflation de t-1.
  SalRefAGIRC_ARRCO|SalRefARRCO|SalRefAGIRC ~
  cumprod(ifelse(annee%in%c(2016,2017,2018),(1+SMPTp+0.02)*(1+Prixp),(1+SMPTp)*(1+Prixp))),
  ValPtAGIRC|ValPtARRCO|ValPtAGIRC_ARRCO ~
  cumprod(1+ifelse(annee%in%c(2016,2017,2018),pmax(Prixp-0.01,0),Prixp)),
  MinRevRG|SeuilExoCSG|SeuilExoCSG2|SeuilTxReduitCSG|SeuilTxReduitCSG2
  ~cumprod(1+Prixp),
  .~1
)

eco$macro=eco$macro%>%filter(annee<=fin_simul)
eco$CiblesTrans <- left_join(eco$macro %>% select(annee), eco$CiblesTrans)
#####
# rassemblement dans un unique environnement
#####
eco=as.list(eco)
demo=as.list(demo)
simul=as.list(simul)
eq_struct=as.list(eq_struct)
simulation=as.environment(c(demo,eco,simul,eq_struct))
rm(demo,eco,simul,eq_struct)
#####
#simulation
#####
destinieDemographie(simulation)
destinieTransMdt(simulation)
destinieImputSal(simulation)
destinieCalageSalaires(simulation)
destinieSim(simulation)

#####
# Resultats -----
#age moyen de liquidation pour tous et par sexes
simulation$Indicateurs_an %>%
  filter (regime=="tot" & annee > 2000) %>%
  ggplot(aes(x=annee,y=Age_Ret_Flux,color=sexe)) + geom_line()
#masse des pensions sur le Pib
simulation$Indicateurs_an %>%
  filter (regime=="tot"& sexe=="ens" & annee > 2010& annee<=2070)%>%
  left_join(simulation$macro)%>%

```

---

```
ggplot(aes(x=annee,y=M_Pensions_ma/10/PIB,color=sexe)) + geom_line()+
theme_bw()
```

---

## 7.2 Pour obtenir/utiliser les sources

1. Récupérer le dossier contenant le modèle : Créer un dossier Destinie ; puis y cloner le dépôt en le nommant destinie.<sup>3</sup>  
Par exemple, pour les utilisateurs de TortoiseGit :
  - Faire un clic droit dans l'explorateur
  - Cliquer sur "Git cloner"
  - dans URL, inscrire : <https://github.com/InseeFr/Destinie-2>
  - dans Répertoire inscrire l'emplacement où vous souhaitez installer Destinie.
  - Récupérer le dossier contenant le modèle : Créer un dossier Destinie ; puis y cloner le dépôt en le nommant destinie.
2. Installer le package devtools, et dans Rstudio utiliser les commandes de Build pour charger destinie ou compiler le package source .tar.gz ou le windows binaires.

### En cas de difficultés :

Si vous avez des difficultés liées au modèle que vous n'arrivez pas à résoudre, sur le dépôt public (actuellement sur <https://github.com/InseeFr/Destinie-2>), vous trouverez un onglet : "Issues".

Vous pouvez rechercher si d'autres personnes ont déjà rencontré ce problème, et si la communauté des utilisateurs y a répondu, notamment en filtrant sur les étiquettes (*label*) "*Good for newcomers*".

Si le problème persiste, vous pouvez y rédiger un message. Déclinez votre nom, puis présentez du mieux possible un exemple minimal, complet et vérifiable (voir quelques conseils sur ce sujet ici : <https://stackoverflow.com/help/mcve>). Enfin, étiquetez le nouveau sujet à l'aide par exemple d'un ou de plusieurs des codes suivants :

- "*Good for newcomers*" (Bien pour les débutants)
- "*bug*"
- "*Something isn't working*" (Quelque chose ne marche pas)
- "*duplicate*" (répétition)
- "*enhancement*" (amélioration)
- "*New feature or request*" (Nouvelle fonctionnalité)
- "*This doesn't seem right*" (Cela ne semble pas correct)
- "*question*" (Question)
- "*Further information is requested*" (De plus amples informations sont demandées)

### A noter :

Au sein du projet lui-même, quelques conseils pratiques très courants sont également intégrés au sein d'un fichier "[conseils\\_pratiques.txt](#)".

---

3. Le fait que le dossier soit dans [/Destinie/destinie](#) permet aux liens dans les fichiers de paramètres de rester corrects.

## Références

- AUBERT, P. et BACHELET, M. (2012). *Disparités de montant de pension et redistribution dans le système de retraite français*. INSEE, série des documents de travail de la Direction des Études et Synthèses Économiques, G1206.
- BACHELET, M., LEDUC, A. et MARINO, A. (2014). *Les biographies du modèle Destinie II: rebasage et projection*. Document de travail INSEE/DESE, g201401.
- BLANCHET, D. et CHANUT, J.-M. (1998). *Les retraites individuelles à long terme: une projection par microsimulation*. *Economie et statistique*, 315(1):95–106.
- BLANPAIN, N. (2016). *L'espérance de vie par catégorie sociale et par diplôme - Méthode et principaux résultats*. Document de travail INSEE/DSDS, F1602.
- BLANPAIN, N. et BUISSON, G. (2016). *Projections de population 2013-2070 pour la France*. Insee Résultats, 187.
- BUISSON, G. et LAPINTE, A. (2013). *Le couple dans tous ses états*. Insee première, 1435.
- CAZENAVE-LACROUTZ, A. et GODET, F. (2017). *L'espérance de vie en retraite sans incapacité sévère des générations nées entre 1960 et 1990 : une projection à partir du modèle Destinie*. Document de travail INSEE/DESE, G1703.
- DUBOIS, Y. et KOUBI, M. (2017). *Règles d'indexation des pensions et sensibilité des dépenses de retraites à la croissance économique et aux chocs démographiques*. Document de travail INSEE/DESE, G1702.
- DUBOIS, Y. et MARINO, A. (2015a). *Indicateurs de rendement du système de retraite français*. Document de travail INSEE/DESE, G1502.
- DUBOIS, Y. et MARINO, A. (2015b). *Le taux de rendement interne du système de retraite français: quelle redistribution au sein d'une génération et quelle évolution entre générations?* *Economie et statistique*, 481(1):77–95.
- DUÉE, M. (2005). *La modélisation des comportements démographiques dans le modèle de microsimulation Destinie*. Document de travail INSEE/DESE, G0504.
- MARBOT, C. et ROY, D. (2015). *Projections du coût de l'APA et des caractéristiques de ses bénéficiaires à l'horizon 2040 à l'aide du modèle Destinie*. *Economie et statistique*, 481(1):185–209.