SimBEL: Calculate the best estimate in life insurance with Monte-Carlo techniques

Quentin Guibert

Univ Lyon, Université Claude Bernard Lyon 1, ISFA, Laboratoire SAF EA2429, F-69366, Lyon, France
Prim’Act, Paris, France
Email: quentin.guibert@primact.fr

R in Insurance 2017
8th June 2017, Paris, France

Joint work with G. de Kervénoaël and M. Tammar (Prim’Act)
Best estimate calculation

- Under Solvency II, liabilities in life insurance are valued based on a *market consistency* principle (Kemp, 2009; Vedani *et al*., 2017), taking into account:
  - financial options and guarantees,
  - future management actions, e.g. profit sharing rules,
  - the policyholder’ behavior,
  - both undertaking and financial risks.

- A stochastic Asset Liability Management (ALM) model based on Monte-Carlo balance sheet projection is generally implemented to compute the best estimate of liabilities (see art. 77 directive Solvency II)

\[
BE_t = \mathbb{E} \left[ \sum_{u > t} \delta_u CF_u \right]
\]

- \( \delta_u \), the stochastic deflator at time \( u \);
- \( CF_u \), the net payment cash-flows at time \( u \).
European Literature

- **Profit sharing** rules (see e.g. Grosen and Løchte Jørgensen, 2000; Bacinello, 2001; Ballotta *et al.*, 2006; Kling *et al.*, 2007).

- **ALM** (see e.g. Bauer *et al.*, 2006; Hainaut, 2009).

- **Policyholder’s behavior** (see e.g. Planchet and Théron, 2007; Milhaud *et al.*, 2011; Bauer *et al.*, 2006; Eling and Kochanski, 2013).

- The French valuation model is rarely described, but is quite complex as insurers have a higher leeway to distribute profit sharing (Borel-Mathurin *et al.*, 2015).
Aims

▶ In France, most of such valuation models are developed by commercial firms or directly by insurers. They are no available for students, researchers, ...

▶ No package to forecast assets and liabilities is available for insurance obligations.

▶ An ALM model requires algorithms to forecast both assets and liabilities at a very granular level under the local gaps. It is coupled with an Economic Scenarios Generator (ESG).

▶ With large asset and liability portfolios, the computation can be very time consuming.

▶ Our aims:
  ▶ Develop a flexible R-package to compute easily the best estimate of a life participating contract, especially a French euro-denominated contract.
  ▶ Usable for the Solvency Capital Requirement (SCR) computation.
  ▶ Flexible architecture allowing to project both data and assumptions as it is required for example for the Own Risk Solvency Assessment (ORSA) purpose.
The R-package SimBEL

- The package is implemented in an oriented object fashion in **S4**.
- An access to the last development version on GitHub is available on demand.
- The user guide and a large documentation (in French at the moment) is available.
- To install the package
  
  ```
  library(devtools)
  library(githubinstall)
  install_github("xxx", auth_token = "yyy")
  ```
- To load the package
  
  ```
  library(SimBEL)
  ```
Overview of the calculation process

The best estimate is calculated following this general process (Laurent et al., 2016)

1. **Generate risk neutral scenarios**
2. **Forecast balance sheet and P&L in local gap**
   - Compute assets and liabilities values and cash-flows before profit
   - Apply buy-and-sell strategies for assets
3. **Apply profit sharing rules**
   - Contractual and legal constraints
   - Take in account policyholders’ expectations and behaviors
   - Use profit sharing reserves and compensation between insured.
4. **Revaluate liabilities**
   - Instantenous or time-smoothing
   - Compute final balance sheet and P&L
Overview of the structure (only cash flows projection)
Main functionalities

1 Liability module
   ▶ Both saving and retirement products with participation can be modeled.
   ▶ The liability side is modeled using model points, which represent the technical characteristics of each contract.
   ▶ An user should provide additionally:
     ▶ Technical assumptions (tables with mortality rates and static lapse rates, parameters for dynamic lapses),
     ▶ expenses assumptions,
     ▶ the current value of other provisions.
Main functionalities

1 Liability module

▶ Both saving and retirement products with participation can be modeled.

▶ The liability side is modeled using model points, which represent the technical characteristics of each contract.

▶ An user should provide additionally:
  ▶ Technical assumptions (tables with mortality rates and static lapse rates, parameters for dynamic lapses),
  ▶ expenses assumptions,
  ▶ the current value of other provisions.
Main functionalities

1 Liability module

- Both saving and retirement products with participation can be modeled.
- The liability side is modeled using model points, which represent the technical characteristics of each contract.
- An user should provide additionally:
  - Technical assumptions (tables with mortality rates and static lapse rates, parameters for dynamic lapses),
  - expenses assumptions,
  - the current value of other provisions.
Main functionalities

Asset module

Four asset classes are modeled:
- Fixed bonds (Gouv. and Corporate),
- Equities,
- Properties,
- Cash account.

An user should provide:
- ESG tables for asset projection,
- a reference portfolio for future reinvestments,
- an investment strategy,
- the current value for asset provisions,
- fees on asset.
Main functionalities

2 Asset module

- Four asset classes are modeled:
  - Fixed bonds (Gouv. and Corporate),
  - Equities,
  - Properties,
  - Cash account.

- An user should provide:
  - ESG tables for asset projection,
  - a reference portfolio for future reinvestments,
  - an investment strategy,
  - the current value for asset provisions,
  - fees on asset.
Main functionalities

3 Balance sheet modules
   ▶ Asset and liability are linked together with an object called a Canton.
   ▶ When projecting a Canton, ALM and surplus appropriation scheme are applied.

4 Best estimate module
   ▶ Project a Canton for each simulation.
   ▶ Calculate the best estimate based on the initial situation of a canton.
Main functionalities

3 Balance sheet modules
   - Asset and liability are linked together with an object called a Canton.
   - When projecting a Canton, ALM and surplus appropriation scheme are applied.

4 Best estimate module
   - Project a Canton for each simulation.
   - Calculate the best estimate based on the initial situation of a canton.
### Loading data

- Lots of data and parameters are required.
- A module is designed to feed all these data from a repository contained csv files.
- Create a Canton for each shock defined in Solvency II standard formula.
Loading data

- To load the addresses for data repositories.

```r
root <- new("Initialisation", root_address = getwd())
root <- set_architecture(root)
```

- To load data related to initialize a Canton.

```r
init_SimBEL(racine)
```

- To create and save a Canton for each shocked situation as defined by the standard formula.

```r
init_scenario(racine)
```
Canton class

- A Canton stores the current picture of the balance sheet and all parameters to project it.

```r
canton@annee  # Number of projection years
canton@hyp_canton  # Some general assumptions
canton@mp_esg  # Extraction of an ESG table for the current projection year and simulation
```

- To call the Asset portfolio.

```r
canton@ptf_fin
```

- To call the Liability portfolio.

```r
canton@ptf_passif
```
Liability classes

- **Two classes**: euro-denominated French saving and pension guarantees.

- Each instantiation of these classes is a product with specific features in terms of expenses and contractual profit sharing rate.

- Liabilities are projected on an **annual** basis.

- Example for a saving product

```r
ee1 <- canton@ptf_passif@eei[[1]]  # The first saving product
class(ee1)  # "EpEuroInd"
```
**Liability classes**

- Model points are stored in a data.frame object.

```r
str(ee1@mp)
```

```r
'data.frame': 15 obs. of 29 variables:
# num_mp : int 1 2 3 ...
# ... 
# age : int 40 40 40 ...
# gen : int 1900 1900 1900 ...
# num_tab_mort : Factor w/ 1 level "TM2": 1 1 1 ...
# chgt_enc : num 0.007 0.007 0.007 ...
# ... 
# pm : int 900 11600 12000 ...
# nb_contr : int 1 1 1 ...
# anc : int 0 0 0 ...
# ... 
# tx_cible : Factor w/ 1 level "Meth1": 1 1 1 ...
# prime : int 0 0 0 ...
# tx_tech : num 0 0 0 ...
# ...
```
Liability cash flows

- Technical assumptions are attached to the liability portfolio.

\[
\text{canton@ptf\_passif@ht} \quad \# \quad \text{All technical assumption}
\]
\[
\# \quad \text{The mortality table 'TM1'}
\]
\[
canton@ptf\_passif@ht@tables\_mort[["TM1"]]
\]

- To compute premiums

\[
\text{prem} \leftarrow \text{calc\_primes}(ee1)
\]

- To compute lapse and mortality rates

\[
\text{rates} \leftarrow \text{calc\_proba\_flux}(ee1, \text{ptf\_passif@ht})
\]

- To compute minimal revalorisation rates and target rates

\[
\text{tx\_min} \leftarrow \text{calc\_tx\_min}(ee1)
\]
\[
\text{target} \leftarrow \text{calc\_tx\_cible}(ee1, \text{list}(\text{ptf\_passif@ht}, \text{list\_rd} = \text{list}(0.02, 0.01, 0.01, 0)))
\]
Liability cash flows

- To compute benefits

```r
ben <- calc_prest(ee1, rates, tx_min, an = 1,
                   method = "normal", tx_soc = 0.155)
```

- To compute mathematical reserves

```r
pm <- calc_pm(ee1, prem[["flux"]], ben[["flux"]],
               target, tx_min, an = 1, method = "normal",
               tx_soc = 0.155)
```

- To forecast liability portfolio over 1 year

```r
proj <- proj_annee_av_pb(an = 1, x = ptf_passif,
                          tx_soc = 0.155, coef_inf = 1,
                          list_rd = list(0.02,0.01,0.01,0))
```
Liability cash flows

- Cash-flows and mathematical reserves are aggregated by product

```r
# Outputs
# Cash-flows by product
proj[["flux_agg"]]
# Mathematical reserves and the number of contracts
proj[["stock_agg"]]
```

- These outputs can be used to build some checks.
Asset cash-flows

- Four asset classes: their dynamics are given by the ESG tables.
- To print asset allocation.

```r
print_alloc(canton@ptf_fin)
```

- For each class, the current "picture" of assets is stored in a data.frame.

```r
# Bond portfolio
canton@ptf_fin@ptf_oblig
```

- To calculate cash-flows and market values for bonds

```r
# Coupons and terminal cash-flows
calc_flux_annee(canton@ptf_fin@ptf_oblig)
# Market value
calc_vm_oblig(canton@ptf_fin@ptf_oblig, canton@mp_esg@yield_curve)
```
Canton forecasting

- A Canton can be very easily projected over 1 year:
  - Asset and liabilities cash-flows,
  - Apply ALM and profit sharing rules,
  - Compute P&L and other balance sheet items,
  - Set the value of an updated Canton.

```r
resultProjAn <- projAn(canton, nb_annee, pre_on = FALSE)
canton_updated <- resultProjAn[["canton"]]
# Extract cash flows by product
resultProjAn[["output_produit"]][["flux"]]
# Extract financial results
resultProjAn[["result_fin"]]
```
### Best estimate calculation

- A best estimate objet is defined with a Canton and ESG tables

```r
class(be)
[1] "Be"

# The initialized Canton
be@canton

# ESG tables
be@esg
```

- Compute Monte-Carlo simulations

```r
# To run simulation #10
run_be_simu(be, 10L, pre_on = F)

# To run all the simulations
be_results <- run_be(be, pre_on = F)

# Extract the amount of best estimate
be_results[["be"]][@tab_be]

# Extract the average cash-flows
be_results[["be"]][@tab_flux]
```
Performances

- Some functions are developed using the library `Rccp`.
- The function `run_be()` can be speed up and allows parallel computing with the package `doParallel`.
- Our performances are rather good with `R`!

**Table**: Performances with 1,000 simulations in minutes by using 1 core (Intel Core i7-5500U 2.40GHz) and 8 GB RAM.

<table>
<thead>
<tr>
<th>Number of model points for each Asset class</th>
<th>Number of Saving and Retirement model points (for each)</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
<td>13.2</td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td>14.0</td>
</tr>
<tr>
<td>100</td>
<td>10000</td>
<td>24.7</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>13.8</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>15.0</td>
</tr>
<tr>
<td>1000</td>
<td>10000</td>
<td>26.1</td>
</tr>
</tbody>
</table>
Perspectives

- Increase performances.
- Include UL products and contract an other specificities.
- Add new asset classes (e.g. floating rate bonds).
- Develop a toolkit with indicators for the results analysis.
- Take the inputs from the user with more security.
- Extend our group of developers.
Profit sharing algorithm

[Diagram]

- **Evaluer la PB contractuelle**
- **Reprendre PPB pour financer TMC**
- **Financement du taux cible par la PPB**
  - \( T_{xServ} = T_{xPBC} > T_{xCible} \)
  - **Reprendre PPB**
    - \( T_{xServ} = T_{xPBC} + \text{Reprise} > T_{xCible} \)
    - **Reprendre PVL action**
      - \( T_{xServ} = \text{new}_{-}T_{xPBC} + \text{Reprise} > T_{xCible} \)
      - **Reduire marge assureur**
  - **Calculer taux servi**
  - **Appliquer contrainte légale**
    - **Reducire marge assureur**
    - **Doter PPB**
      - \( T_{xServ} + \text{Reduction marge non dédiée} \)
References


