POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Group 5 Solvency II Project

Insurance Project

Author:
Callini Matteo
Carzaniga Carlotta
Gaspari Cecilia
Torba Matteo

Student ID:
10721592
10739623
10707989
10742071

Academic Year: 2023-24

Contents

Original Text

Consider a simplified insurance company whose assets and liabilities sides are characterized as follows:

ASSETS

- there is a unique fund made of equity (80%) and property (20%), $F_t = EQ_t + PR_t$
- at the beginning (t=0) the value of the fund is equal to the invested premium $F_0 = C_0 = 100,000$
- equity features
  - listed in the regulated markets in the EEA
  - no dividend yields
  - to be simulated with a Risk Neutral GBM ($sigma = 20\%$) and a time varying instantaneous rate r
- property features
  - listed in the regulated markets in the EEA
  - no dividend yield
  - to be simulated with a Risk Neutral GBM ($sigma = 25\%$) and a time varying instantaneous rate r

LIABILITIES

- contract terms
  - whole Life policy
  - benefits
    * in case of lapse, the beneficiary gets the value of the fund at the time of lapse, with 20 euros of penalties applied

        * in case of death, the beneficiary gets the maximum between the invested premium and the value of the fund

    - others

        * Regular Deduction, RD of 2.20%

        * Commissions to the distribution channels, COMM (or trailing) of 1.40%

- model points

    - just 1 model point

    - male with insured aged x=60 at the beginning of the contract

- operating assumptions

    - mortality: rates derived from the life table SI2022 [1]

    - lapse: flat annual rates $l_t = 15\%$

    - expenses: constant unitary (i.e. per policy) cost of 50 euros per year, that grows following the inflation pattern

- economic assumption

    - risk free: rate r derived from the yield curve (EIOPA IT without VA 31.03.24),

    - inflation: flat annual rate of 2%

Other specifications:

- time horizon for the projection: 50 years.
  In case of outstanding portfolio in T=50, let all the people leave the contract with a massive surrender

- the interest rates dynamic is deterministic, while the equity and property ones are stochastic.

[1] `https://demo.istat.it/index_e.php`

Questions:

1. code a Matlab/Python script to compute the Basic Solvency Capital Requirement via Standard Formula and provide comments on the results obtained. The risks to be considered are:
    - Market Interest
    - Market equity
    - Market property
    - Life mortality
    - Life lapse
    - Life cat
    - Expense
2. Split the BEL value into its main PV components: premiums (=0), death benefits, lapse benefits, expenses, and commissions
3. Replicate the same calculations in an Excel spread sheet using a deterministic projection.
    - Do the results differ from 1? If so, what is the reason behind?
    - For the base case only :
        - calculate the Macaulay duration of the liabilities;
        - calculate the sources of profit for the insurance company, deriving its PVFP
        - check the magnitude of leakage by verifying the equation MVA = BEL + PVFP (i.e. MVA=BEL+PVFP+LEAK)
        - sense check the PVFP using a proxy calculation, based on the annual profit and the duration of the contract
4. Open questions:
    - what happens to the asset and liabilities when the risk-free rate increases/decreases with a parallel shift of, say, 100bps? Describe the effects for all the BEL components;
    - what happens to the liabilities if the insured age increases? What if there were two model points, one male and one female?

1 | Summary

	MVA	BEL	BoF	d_BoF	dur_L
BASE	100000	94493.6159	5506.3842	0.0000	5.6131
EQ STRESS 1	64600	65083.8822	3716.1177	1790.2664	5.6133
PROP STRESS	95000	89780.5175	5219.4825	286.9017	5.6131
IR UP	100000	94477.9304	5522.0696	-15.6855	5.6126
IR DOWN	100000	94853.0729	5146.9271	359.4570	5.6400
EXPENSE STRESS	100000	94531.7223	5468.2777	38.1064	5.6139
MORTALITY	100000	94498.6077	5501.3923	4.9919	5.6059
LIFE CAT	100000	94498.6077	5501.3923	4.9919	5.6059
LAPSE UP	100000	95624.0886	4375.9114	1130.4727	4.0158
LAPSE DOWN	100000	92119.8092	7880.1908	-2373.8067	9.0321
LAPSE MASS	100000	96040.5279	3959.4721	1546.9120	3.4029
BSCR	3022.2438				

Table 1.1: Results in the deterministic projection case.

Results	MVA	BEL	BoF	*d_BoF*	dur_L
BASE	100000	95550.3476	4449.6524	0.0000	5.6793
EQ STRESS 1	68800	61340.2631	2859.7369	1589.9155	5.6710
PROPERTY STRESS	95000	90835.4339	4164.5661	285.0863	5.6827
IR UP	100000	95212.3433	4787.6567	-338.0042	5.6524
IR DOWN	100000	95876.8149	4123.1851	326.4673	5.7048
EXPENSE STRESS	100000	95588.4540	4411.5460	38.1064	5.6800
MORTALITY STRESS	100000	95698.0467	4301.9533	147.6991	5.6429
LIFE CAT	100000	95562.2265	4437.7735	11.8789	5.6716
LAPSE UP	100000	96113.3845	3886.6155	563.0369	4.0351
LAPSE DOWN	100000	95355.1657	4644.8343	-195.1819	9.3182
LAPSE MASS	100000	96584.6637	3415.3363	1034.3161	3.4507
BSCR	2494.0727				

Table 1.2: Results in the stochastic projection case.

2 | Formulas Adopted

2.1. Martingale Test

The purpose of this section was to find the optimal number of time steps and the optimal number of simulations for the assets generation.
In order to compute the test, we considered the SDE related to the Geometric Brownian Motion, which is the stochastic process describing the Stock path.

$$\begin{cases} dS(t) = r(t)S(t)dt + \sigma S(t)dWt & \text{(2.1a)} \\ S(t_0) = S_0 & \text{(2.1b)} \end{cases}$$

In order to find the best one, we compared the value of the asset, for every time step t, with the expected value of the process (we considered the sample mean): in this way, for every number of simulations we computed the L^2 norm and we chose the combination which minimized it.
We considered different numbers of possible simulations, from 10000 to 100000 with step 10000, and time steps from 50 to 200 with step 50.
Since we were interested in the assets for each year, we also computed a Martingale test considering the same set of simulations, but with a yearly time steps until T = 50.

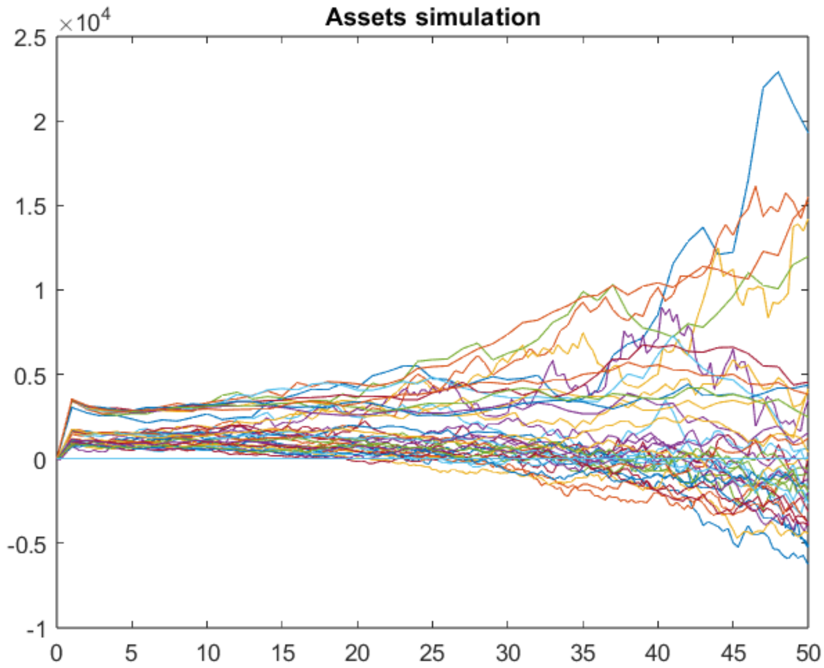

Figure 2.1: Martingale Test

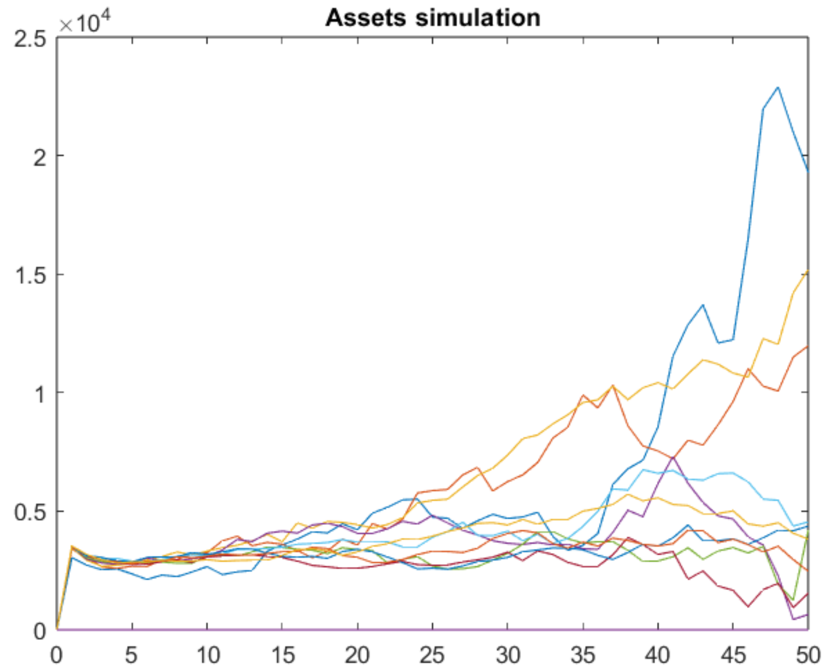

Figure 2.2: Martingale Test with 1year time steps

So, in the first case we obtained that 200 steps and 50000 simulations is the best combination: we expected this result for the time steps since it is more precise to discretize the

timeline in more parts.
In the second case we obtained that 70000 is the best number of simulations and we decided to use it for the following part of the project.

2.2. Assets

We considered a simplified model in which there is a single fund made of 80% of equity and 20% of property, the initial value of the fund is equal to the invested premium. In order to simulate the behaviour of both property and equity we used a Risk Neutral Geometric Brownian Motion, as follows:

$$\begin{cases} dEq(t) = r(t)Eq(t)dt + \sigma_E Eq(t)dWt & \text{(2.2a)} \\ Eq(t_0) = 0.80 \cdot F_0 & \text{(2.2b)} \end{cases}$$

$$\begin{cases} dPr(t) = r(t)Pr(t)dt + \sigma_P Pr(t)dWt & \text{(2.3a)} \\ Pr(t_0) = 0.20 \cdot F_0 & \text{(2.3b)} \end{cases}$$

Notice that Property and Equity have two different volatilities.
In order to simulate the assets we exploited the properties of the Brownian Motion, in particular the fact that the increment $W(t_i) - W(t_{i-1})$ is distributed like a Gaussian random variable, therefore the behaviour of the equity is:

$$Eq(t_{i+1}) = Eq(t_i)exp[(r(t_i) - \tfrac{1}{2}\sigma_E^2)dt_i + \sigma_E\sqrt{dt_i}g(t_i)](1 - RD)$$
$$\text{where } g \sim N(0,1)$$

In this case we also took into account the Regular deduction term, applied at each yearly step of the assets' simulations. Proceeding in this way, and using the optimal number of iterations we had previously found in the Martingale Test, we derived the value of Equity and Property at every year. The total value of the fund was then given by the sum of the two.

To reduce the variance of the simulated assets, we implemented the simulation via the antithetic variables technique as well, sampling half of the number of the gaussian random variables and using the following expressions:

$$Eq(t_{i+1}) = Eq(t_i)exp[(r(t_i) - \tfrac{1}{2}\sigma_E^2)dt_i + \sigma_E\sqrt{dt_i}g(t_i)](1 - RD) \text{ for } t = 0, ..., M/2 - 1$$

$$Eq(t_{i+1}) = Eq(t_i)exp[(r(t_i) - \tfrac{1}{2}\sigma_E^2)dt_i - \sigma_E\sqrt{dt_i}g(t_i)](1 - RD) \text{ for } t = M/2, ..., M - 1$$

where $g \sim N(0,1)$ and M is the number of simulations obtained from the Martingale

Test. The same technique was applied to simulate the property and the sum of Equity and Property yielded the value of the fund.

In all this computations we used the forward rates that we derived from the EIOPA IT. We were given the spot rates, from them we derived the discount factors as $B(t_0, t_i) = (1+r(t_i))^{t_i}$, we evaluated the forward discounts, and lastly we obtained the forward rates.

2.3. Liabilities

The liabilities, in case of an insurance company, are the amount of money the company should reserve in order to fulfill possible future obligations. Since we were working on a simplified insurance contract we had the liabilities connected to the cases of Death, Lapse, Survival and Commissions.
We define the following notations to make the chapter easier to read:

- *C_ lapse(t)* represents the amount to return to the insured at time t in case of lapse at year t; it was computed as:

$$C\_lapse(t) = (S(t) - Pen)$$

  where Pen is the penalty in case of lapse, equal to 20 and S is the simulated value of the fund at time t.

- *C_ death(t)* represents the amount to return at time t in case of death at year t; it was computed as the maximum between the value of the fund at the time of death t and the invested premium C_0:

$$C\_death(t) = max(S(t), C_0)$$

- *q(x+t)* represents the death probability (taken form ISTAT 2022 tables) of a x+t-1 years old man to die within the next year (these probabilities start from x+t-1 = 0).
- *lapse(t)* represents the probability that a x+t-1 years old man lapse in the year t.
- *discount(t)* represents the discount factor from t=0 to t, obtained from the EIOPA spot rates as previously described.
- *prob_ not_ lapse(t)* represents the probability that the 60 years old man considered

does not lapse for t-1 years (prob_not_lapse(1) = 1):

$$prob\_not\_lapse(t+1) = \prod_{i=1}^{t} prob\_not\_lapse(i) \cdot (1 - lapse(i))$$

for $t = 1, ..., T$.

- *surv_prob(t)* represents the probability that the 60 years old man considered does not die for t-1 years (surv_prob(1) = 1):

$$surv\_prob(t+1) = \prod_{i=1}^{t} surv\_prob(i) \cdot (1 - q(x+i))$$

for $t = 1, ..., T$.

Now we describe each liability.

- **Liabilities in case of Lapse**: in case of lapse the beneficiary of the contract receives $C\_lapse(t)$, the value of the fund at the time of lapse minus a penalty, of 20 euros. We expected that a penalty of this magnitude, compared to the value of the fund, would not have a really great impact.
The amount received should then be multiplied by the probability of not dying up until that year (which is in position $t + 1$ in the vector $surv\_prob$), by the probability to not lapse up to the previous year (which is in position t in the vector $prob\_not\_lapse$) and by the probability of lapsing in the year considered. After discounting for these probabilities, we computed the mean of the simulations of the expected liabilities, fixing the time t.
So we obtained the mean of the following expected value at time t

$$Liab\_lapse(t) = mean(C_n\_lapse(t) \cdot lapse(t) \cdot prob\_not\_lapse(t) \cdot surv\_prob(t+1))$$

where t = 1,...,T and the mean the is done with respect to n which represents the simulation.
In this way we obtain a T-dimensional vector which contains the expected liability for each year.
Then we obtain the discounted expected liability in the following way:

$$Liab\_lapse = \sum_{i=1}^{t} Liab\_lapse(t) \cdot discount(t)$$

We remark that we also took into account the case in which the insured person withdraws from the contract between the 49th and the 50th year: the constraint that the contract necessarily ends at T=50 is addressed in the case "Liabilities in case of Survival".

- **Liabilities in case of Death**: in case of death of the insured, the beneficiary receives the maximum amount between the value of the fund, at the time of death, and the invested premium. Similarly to the previous case, this amount needed to take in consideration the probability of not lapsing up until that year, of dying within the year and to survive up to the previous year.
  Again, we computed the mean of the simulations of the expected liabilities, fixing the time t. So we obtained the mean of the following expected value at time t:

  $$Liab\_death(t) = mean(C_n\_death(t) \cdot q(x+t) \cdot prob\_not\_lapse(t) \cdot surv\_prob(t))$$

  where t = 1,...,T and the mean was done with respect to n which represents the simulation's number.
  In this way we obtained a T-dimensional vector which contained the expected lapse liability for each year.
  Then we discounted the expected liability in the following way:

  $$Liab\_death = \sum_{i=1}^{t} Liab\_death(t) \cdot discount(t)$$

- **Liabilities in case of Survival**: if the insured does not lapse and is still alive by the end of the contract (50 years from today), he receives the value of the fund at that time. We computed the liability in this case as follows:

  $$\text{Liab\_survive} = \left(\text{mean}\left(\text{surv\_prob(end)} \cdot \text{fund}_n\text{(end)} \cdot \text{prob\_not\_lapse(end)}\right)\right) \cdot \text{discount(end)}$$

- **Liabilities from Commissions**: for each year that the contract is still present the insurance company has to pay commissions both on Equity and Property. When simulating the behaviour of both of them we evaluated this component by multiplying by the Commission coefficient of 1.40%. Therefore, the liabilities connected to commissions were computed as the sum of the commissions, for every year, coming from Equity and Property, taking in account the probabilities of not lapsing and surviving up until that year.

As before, we took the mean of the simulations.

$$Commissions(t) = mean(Commissions_n(t) \cdot prob\_not\_lapse(t+1) \cdot surv\_prob(t+1))$$

where $t = 1, ..., T$. And then we discounted them:

$$Commissions = \sum_{i=1}^{t} Commissions(t) \cdot discount(t)$$

- **Liabilities from Expenses**: at the end of every year of the contract the company has to pay some fixed expenses. The amount is 50 euros (per policy), which follows the trend of inflation: we made the assumption that the inflation rate has an impact only starting from the second year, since at time $t = 0$, when the contract is signed, the expenses for the first year should be already known. As for all the above liabilities, to compute this amount we needed to take into consideration the probability of surviving and of not lapsing until the year in which the expenses are paid.

$$Expense(t) = 50 \cdot (1 + i)^{t-1} \cdot prob\_not\_lapse(t + 1) \cdot surv\_prob(t + 1)$$

where $i = 0.02$ is the inflation rate. Then we obtained the Expense value in zero by summing up the discounted value just computed:

$$Expense = \sum_{i=1}^{t} Expense(t) \cdot discount(t)$$

We then computed the BoF as the difference between the fund value in zero and the sum of the discounted liabilities:

$$BoF = F_0 - \sum Liabilities$$

Since our aim was to evaluate the Basic Solvency Capital Requirement, we analysed various cases in which either a Market component or a Life one were stressed. Regarding the Market we dealt with a stress of the Equity, the Property and of the interest rates, whereas for the life we considered a stress on mortality, probability of lapsing, Catastrophic events and a stress in the expenses.
For each one of this stress scenarios we evaluated the Solvency Capital Requirement as:

$$SCR = max(BoF_{base} - BoF_{stress}, 0)$$

We then computed the SCR both for Life and Market as follows:

$$SCR_{Life} = \sqrt{\sum_{i,j=1}^{n} Corr_{i,j} \cdot SCR_i \cdot SCR_j}$$
$$SCR_{Market} = \sqrt{\sum_{i,j=1}^{n} Corr_{i,j} \cdot SCR_i \cdot SCR_j}$$

The correlation matrices for Market and Life are the following:

	IR	EQ	PR
IR	1	A	A
EQ	A	1	0.75
PR	A	0.75	1

	MOR	LAPSE	EXP	CAT
MOR	1	0	0.25	0.25
LAPSE	0	1	0.5	0.25
EXP	0.25	0.5	1	0.25
CAT	0.25	0.25	0.25	1

Where A is equal to 0 in case of IR stress Up, or 0.5 in case of IR stress Down.
Lastly we computed the BSCR as:

$$BSCR = \sqrt{\sum_{i,j=1}^{n} Corr_{i,j} \cdot SCR_i \cdot SCR_j}$$

In this case the correlation matrix is the one between Life and Market:

	Market	Life
Market	1	0.25
Life	0.25	1

2.3.1. Base case

The first scenario we began to analyse was the base scenario. In this case we did not apply any stress and we evaluated the liabilities in the base case. The procedure we followed to do the computations is exactly the one we presented in the previous section.

LiabDeath	*LiabLapse*	*LiabSurv*	*Expenses*	*Commi*	*TotLiab*	*BoF*	*dBoF*
7453.5278	81256.5420	6.5804e-06	247.6198	6592.6578	95550.3475	4449.6524	0

Table 2.1: Liabilities, BoF and Δ_{BoF} in the base case via stochastic assets' simulation

These values of the Liabilities were the one used as a comparison for the stressed results. We can observe that the main component in the total liabilities is by far the value of the liabilities in case of lapse: this could be related to the fact that the penalty in case of lapse is not very significant with respect to the fund's value over time, and the lapse probability is quite high (15% annually). Other numerically relevant liabilities are the ones in case of death and the commissions; the expenses and in particular the survival term do not have a significant impact on the liabilities in the base case: for the *LiabSurv* this result was widely expected, since it is very unlikely that the insured person will still be alive when he will be 110 years old. The numerical difference between the different liabilities values can be graphically appreciated in the pie chart in Figure 2.3 (liabilities in case of survival were omitted since they would have not been visible).

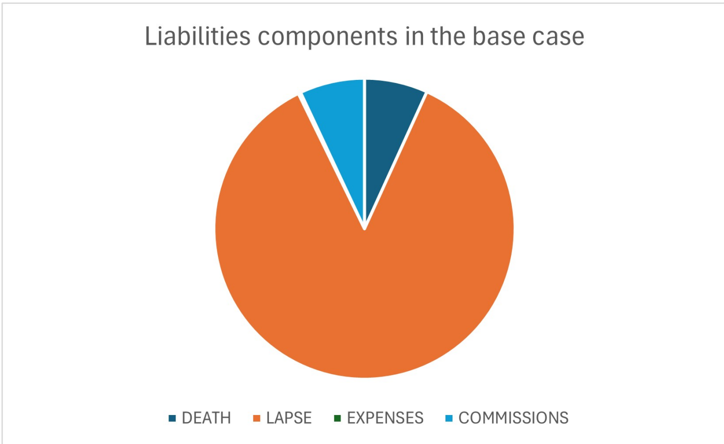

Figure 2.3: Percentage of the liabilities components over the total liabilities.

2.3.2. Equity stress type 1

Equity risk comes from the level of volatility of market prices for equities. Exposure to equity risk includes all assets and liabilities whose value is affected by changes in equity prices.

The equity risk sub-module is divided into two parts: one for type 1 equities and another for type 2 equities:

- Type 1 equities comprise those listed on regulated markets in countries that are members of the European Economic Area (EEA) or the Organisation for Economic Co-operation and Development (OECD).
- Type 2 equities include those listed on stock exchanges in countries that are not members of the EEA or OECD, unlisted equities, commodities, and other alternative investments.

We assume that we are considering only European or American equities, se we consider only the Type 1 case.

The equity type 1 shock refers to a decrease in equity of 39% plus 5.25% given by the symmetric adjustment (obtained from EIOPA), so the total decreas is 44.25%; the results obtained in this stressed case are reported in Table 2.2:

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
4699.1476	52161.0730	4.2280e-06	247.6199	4232.4883	61340.2631	2859.7369	1589.9155

Table 2.2: Liabilities, BoF and Δ_{BoF} in the Equity stress of type 1 case via stochastic assets' simulation.

We observe that all the liabilities decrease with respect to the base case, except for the Expenses, which do not depend on the value of the Equity. In the case of death or lapse this decrease in the liabilities is due to the fact that the benefit received by the insured will be lower. However, since also the fund's value decreases, the BoF is lower too, and the Δ_{BoF} is relevant.
Since we consider only Type 1 case, the SCR_{equity} is equal to **1589.9155**.

2.3.3. Property stress

Property risk arises due to the sensitivity of assets, liabilities, and financial investments to changes in property market prices, whether in terms of level or volatility.

The following types of investments should be treated as property, and their associated risks should be assessed accordingly within the property risk sub-module:

- Land, buildings, and rights to immovable property
- Property investments used directly by the insurance company

$$SCR_{property} = max(\Delta BOF|Prop\,shock; 0)$$

The SCR is computed on an immediate 25% decrease in the market value of assets.

The impact of this scenario should also be assessed under conditions where the value of future discretionary benefits can fluctuate, and the insurance company has the ability to adjust its assumptions regarding future bonus rates in response to the shock being tested.

The obtained results are reported in Table 2.3.

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
7131.8876	77192.9025	6.2495e-06	247.6198	6263.0237	90835.4338	4164.5661	285.0683

Table 2.3: Liabilities, BoF and Δ_{BoF} in the Property stress of case via stochastic assets' simulation.

We obtain a $SCR_{property}$ value of **285.0862**, significantly lower than in the Equity stress case: this result is not surprising since only the 20% of the portfolio's of assets of the insurance company is made of Property, while the remaining 80% is made of Equity.

Property shock reflects the vulnerability of assets, liabilities, and financial investments to fluctuations in property market conditions. It underscores the risk of a decline bigger than expected in the value of real estate holdings within the insurance company's portfolio. When the insurance company's PCR is positive, a drop in property market prices becomes a concern. This is because significant decreases in property values can reduce the company's assets and affect its financial stability.

2.3.4. Interest Rates Stress Up

Interest rate risk comes from all assets and liabilities whose value is affected by changes in the term structure of interest rates or interest rate volatility. Assets with sensitivity to interest rates are fixed-income investments, financing instruments, policy loans, interest rate derivatives; but the discounted liabilities values decrease since the discounts decrease: so, the final result we expect to have is a negative BoF and a null SCR, which would imply that the company does not have to address a capital for this particular risk.

$$SCR_{IR_{up}} = max(\Delta BOF|Up; 0)$$

In order to compute the stress we used the shocked up rates took from EIOPA rates' tables. From these we retrieved the discounts, the forward discounts and the forward rates, and then we computed the Liabilities and the BoF as previously done; the obtained results are displayed in the following Table.

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
7131.2090	81257.5659	6.5805e-06	230.9104	6592.6578	95212.3433	4787.6566	-338.0042

Table 2.4: Liabilities, BoF and Δ_{BoF} in the IR up stress case via stochastic assets' simulation.

The interest rates up shock represents the risk associated with unexpected increases in interest rates. It emphasizes the potential impact of rising interest rates on the financial assets and liabilities of an insurance company.
When interest rates rise, it becomes a concern for the insurance company. This is because higher interest rates can reduce the market value of the company's bonds and other fixed-income investments, which can affect its financial health.
However, as expected, the Δ_{BoF} obtained in this case is negative, and this risk does not have to be addressed by the company.

2.3.5. Interest Rates Stress Down

In order to calculate all the interested quantities for Interest Rates Shock Down we took the shocked down EIOPA rates' table and we proceeded as above. The results are reported in Table 2.5.

$$SCR_{IR_{down}} = max(\Delta BOF|Down; 0)$$

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
7766.2018	81255.6341	6.5804e-06	262.3211	6592.6578	95876.8149	4123.1851	326.4673

Table 2.5: Liabilities, BoF and Δ_{BoF} in the IR down stress case via stochastic assets' simulation.

The interest rates down shock represents the risk associated with unexpected decreases in interest rates. It emphasizes the potential impact of falling interest rates on the financial assets and liabilities of an insurance company.

When interest rates fall, it becomes a concern for the insurance company. This is because lower interest rates can decrease the income generated from the company's bonds and other fixed-income investments, which can affect its financial health.
Indeed, in our case this shock decreases the simulated value of the fund and of the benefits, while it increases the value of the discounts. We obtained a SCR equal to **326.4673**; we could expect a positive value because with increasing rates we need a bigger coverage.

2.3.6. Expense stress

The expense risk arises from a change in expenses due to servicing insurance and reinsurance contracts. This risk can be computed by increasing future expenses by 10% and the expense inflation rate by 1% per annum.

$$SCR_{Expenseshock} = max(\Delta BOF|Expenseshock; 0)$$

The main assumptions for the expense risk sub-module include:

- Companies face the risk of changing expenses primarily due to: employee salaries, commissions paid to sales intermediaries (based on contractual agreements), IT infrastructure costs, and costs related to land and buildings they occupy.
- The company functions within a macroeconomic context where inflation is generally kept in check, despite occasional fluctuations, thanks to inflation targeting policies.

We report the results of this stressed case in Table 2.6: it can be immediately noticed that the only liability which is affected by this shock are the Expenses, and the increase is not extremely relevant.

Liab Death	*Liab Lapse*	*Liab Surv*	*Expenses*	*Commi*	*Tot Liab*	*BoF*	*dBoF*
7453.5278	81256.5420	6.58045e-06	285.7262	6592.6578	95588.4539	4411.5460	38.1064

Table 2.6: Liabilities, BoF and Δ_{BoF} in the expense stress case via stochastic assets' simulation.

The expense stress reflects the risk associated with unexpected increases in operational costs. It highlights the potential impact of rising expenses on the financial stability of an insurance company.

When operational costs increase, it could become a problem for the insurance company. This is because higher expenses can reduce the company's profitability and potentially strain its financial resources, affecting its financial health.

In our case, we have a low SCR equal to **38.10640524**: it is reasonably small because expenses represent a minimum percentage of the whole amount.

2.3.7. Mortality stress

The mortality risk refers to the potential loss or negative impact on the value of insurance liabilities due to fluctuations in mortality rates.
This risk arises when changes in mortality rates, whether in their level, trend, or volatility, result in an increase in the value of insurance liabilities.

$$SCR_{mortality} = max(\Delta BOF|Mortshock; 0)$$

The mortality shock refers to a sudden and lasting surge of 15% in mortality rates.

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
8390.2561	80529.0482	1.5478e-07	245.10872	6533.6335	95698.0466	4301.9533	147.6991

Table 2.7: Liabilities, BoF and Δ_{BoF} in the mortality stress case via stochastic assets' simulation.

The stress factor for mortality risk captures the uncertainty in mortality parameters due to misestimation and/or shifts in the level, trend, and volatility of mortality rates. It signifies the risk of a higher-than-anticipated number of policyholders passing away during the policy term.

In our context, we can observe in Table 2.7 that the Liabilities related to death increase, while the ones related to Lapse and Survival decrease, as we could have expected.
Since SCR is positive and equal to **147.6991**, an increase in mortality rates necessitates consideration from the insurance company. This outcome was expected because elevated mortality probabilities significantly raise death liabilities: in particular, death liabilities are way bigger than the standard ones.

2.3.8. Life CAT

The CAT risk stems from extreme or irregular events whose effects are not sufficiently captured in the other life underwriting risk sub-modules. Examples could be a pandemic event or a nuclear explosion.

$$SCR_{CAT} = max(\Delta BOF|CATshock; 0)$$

The revision shock is an absolute 1.5 per mille increase in the rate of policyholders dying over the following year. Also assess whether scenarios in health module apply and, if so, apply these scenarios in addition to the 1.5 per mille.

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
7598.41224	81133.8637	6.5705e-06	247.2460	6582.70445	95562.2264	4437.7735	11.8789

Table 2.8: Liabilities, BoF and Δ_{BoF} in the CAT stress case via stochastic assets' simulation.

We achieve a positive SCR, equal to **11.8789**, aligning with expectations given its catastrophic nature. Consequently, the insurance company must address this risk.

The sole concern lies in its magnitude, which is notably small. One possible explanation is that a higher mortality rate leads to a lower survival probability, thus reducing lapse liabilities: this is due to the fact that the increase in death liabilities is almost compensated by the decrease in the lapse liabilities, which are the main component in the liabilities, especially during the first years.

2.3.9. Lapse stress up

Lapse risk encompasses the potential for loss or adverse effects on liabilities arising from changes in the anticipated rates at which policyholders exercise their options. These options include contractual rights allowing policyholders to fully or partially terminate, surrender, reduce, limit, or suspend insurance coverage, or permit the insurance policy to lapse.

$$SCR_{lapseup} = max(\Delta BOF | lapse\ up\ shock;\ 0)$$

$$lapse_{up} = min(1.5\ lapse, 1)$$

There is an immediate and enduring surge of 50% in the anticipated exercise rates for the relevant options in all future years, capped at 100%.

$LiabDeath$	$LiabLapse$	$LiabSurv$	$Expenses$	$Commi$	$TotLiab$	BoF	$dBoF$
4134.8959	87505.3942	6.4922e-08	157.6183	4315.4759	96113.3844	3886.6155	563.264

Table 2.9: Liabilities, BoF and Δ_{BoF} in the Lapse stress up case via stochastic assets' simulation.

By observing these results, we can notice that the liabilities in case of death and the commissions decrease, while the liabilities related to lapse become higher. The positive SCR, equal to **563.264** underscores the necessity for the insurance company to address this risk. This outcome could have been anticipated, as a rising probability of lapses implies increased liabilities from lapses. Moreover, we anticipate a significant impact given that the escalating lapse rate affects the entire time horizon, with a notably high increase in probability: as a consequence, the lapse liabilities grow a lot.

2.3.10. Lapse stress down

$$SCR_{lapsedown} = max(\Delta BOF | lapse\ down\ shock;\ 0)$$

$$lapse_{down} = max(0.5\ lapse, lapse\ -\ 0.2)$$

There is an immediate and enduring reduction of 50% in the anticipated option exercise rates for the relevant options in all future years, with the absolute variation not exceeding 20%.

LiabDeath	*LiabLapse*	*LiabSurv*	*Expenses*	*Commi*	*TotLiab*	*BoF*	*dBoF*
18955.6106	64551.1881	0.000451	449.5713	11398.7951	95355.1656	4644.8343	-195.1819

Table 2.10: Liabilities, BoF and Δ_{BoF} in the Lapse stress down case via stochastic assets' simulation.

From Table 2.10, we observe that the liabilities in case of death increase (as a consequence of the fact that the probability of not having a lapse increases), and the commissions increase as well, while the lapse probabilities drop. In this scenario, the Standard Capital Requirement is zero, since dBoF is negative, indicating that the insurance company does not need to address this risk. We expected this result since this situation contrasts with the lapse-up case: a decrease in the lapse rate signifies reduced lapse liabilities to maintain.

2.3.11. Lapse MASS

This combination comprises immediate alterations where 'discontinuance' refers to surrender, lapse without value, conversion to a paid-up policy, activation of automatic nonforfeiture provisions, or exercising other discontinuity options, or not exercising continuity options.

$$SCR_{lapsemass} = max(\Delta BOF | lapse\ mass;\ 0)$$

Specifically, it involves discontinuing 40% of insurance policies, excluding group pensions, for which discontinuance would raise technical provisions without the risk margin: this means that the lapse rate for the first year is $lapse_{mass} = 0.4 + lapse$, while the other lapse rates are the same.

Liab Death	Liab Lapse	Liab Surv	Expenses	Commi	Tot Liab	BoF	dBoF
4260.7602	88702.5799	3.4837e-06	131.0929	3490.2306	96584.6636	3415.3363	1034.3161

Table 2.11: Liabilities, BoF and Δ_{BoF} in the Lapse MASS stress case via stochastic assets' simulation.

The positive SCR implies that the insurance company needs to address this risk. As with the lapse up scenario, this outcome was anticipated. Furthermore, the magnitude of this result exceeds that of the lapse-up case: the initial lapse increase of 40% is substantial and immediate, leading to a higher discounted value due to its proximity to the settlement date. In contrast, the lapse-up scenario considers the entire time horizon and entails a smaller increase.

2.3.12. Lapse

The total Lapse SCR is given by the following formula:

$$max(SCR_{Lapseup}, SCR_{Lapsedown}, SCR_{Lapsemass})$$

So we obtain that it is equal to **1034.3161**: we expected this result since the MASS stress was supposed to be the highest as we said in the previous subsection.

2.4. Basic Solvency Capital Requirement

The Basic Solvency Capital Requirement is the monetary amount the companies are mandated to hold by regulators to ensure they have enough funds to cover potential losses. In order to derive its value you need to analyse all the Liabilities and the Assets, as we did in the last sections.

2.4.1. Market risk

Market risk refers to the potential for fluctuations in the value of a company's investments due to changes in factors like property and stock prices, interest rates, and foreign exchange rates. Understanding this risk involves analyzing how these factors impact our investments.

$$SCR_{mkt} = \sqrt{\sum_{i,j=1}^{n} Corr_{i,j} \cdot SCR_i \cdot SCR_j}$$

The correlation matrix is the following one

	Interest	Equity	Property
Interest	100%	A	A
Equity	A	100%	75%
Property	A	75%	100%

Where A = 0, if it is exposed to the Interest Up stress, or A = 0.5, if it is exposed to Interest Down stress. In our case we consider A = 0.5 since $SCR_{Interest\ up} = 0$.

The SCR_{mkt} equal to **2001.9261** indicates that the insurance company is moderately exposed to the risks related to the Equity and Property shocks and to the interest rates downward shift.

2.4.2. Life underwriting risk

The Life Underwriting Risk module is used to determine the capital needed for managing risks associated with life insurance and reinsurance obligations, excluding those pertaining to health insurance: in our case we computed it from the results about the mortality stress, the lapse stress, the expenses stress and the catastrophic scenario.

$$SCR_{life} = \sqrt{\sum_{i,j=1}^{n} Corr_{i,j} \cdot SCR_i \cdot SCR_j}$$

The correlation matrix is the following one:

	Mortality	Lapse	Expense	CAT
Mortality	100%	0	25%	25%
Lapse	0	100%	50%	25%
Expense	25%	50%	100%	25%
CAT	25%	25%	25%	100%

The SCR_{life} is equal to **1068.9682**: from this result we can conclude that the insurance company is more exposed on the market risks than on the life underwriting risk.

2.4.3. BSCR

$$BSCR = \sqrt{\sum_{i,j=1}^{n} Corr_{i,j} \cdot SCR_i \cdot SCR_j}$$

The correlation matrix is the following one:

	Market	Life
Market	100%	25%
Life	25%	100%

We obtain BSCR equal to **2494.0727**.
The BSCR reflects the level of risk exposure the funds. In this case, after having evaluated the potential risks, we have determined the amount of capital necessary to absorb losses under adverse circumstances.
Overall, we require a SCR of 2494.0727 with respect to 100 000 of funds, thus the SCR is roughly the 2.5% of the funds in t=0.

3 | Deterministic Case

Up until now we have made the assumption that the Equity and Property follow the behaviour of a Geometric Brownian Motion. From now on we will assume that they have a deterministic behaviour instead. We have implemented all the computations we are going to describe both in Matlab and in Excel, in order to be able to double check the results.
The main difference with respect to the previous computations stands in the way in which we simulate the assets.

$$Eq(t_i) = Eq(t_{i-1})e^{r(i)} \cdot (1 - RD)$$
$$Pr(t_i) = Pr(t_{i-1})e^{r(i)} \cdot (1 - RD)$$

We are using a deterministic simulation, this result is obtained starting from the initial GBM and considering $\sigma = 0$, in this way we eliminate the stochastic component. The rates are the forward rates and we are, as before, considering the Regular Deduction.
In the same way we simulate the commissions connected to Equity and Property:

$$Eq_{Comm}(t_i) = Eq_{Comm}(t_{i-1})e^{r(i)} \cdot COMM$$
$$Pr_{Comm}(t_i) = Pr_{Comm}(t_{i-1})e^{r(i)} \cdot COMM$$

Apart from these, the computations that follow in order to evaluate the BSCR are the same.
We obtained the following results regarding the different liabilities connected to eache stress.

Results	Lapse	Death	Survive	Expenses	Commissions	SCR
BASE	81227.10	6428.625	6.608e-06	247.6199	6590.26957	-
EQ STRESS 1	55879.2622	4422.8946	4.5463e-06	247.6199	4534.1054	1790.2663
PROPERTY STRESS	77164.94689	6107.19465	6.2776e-06	247.6199	6260.75609	286.9017
IR UP	81228.1243	6428.6259	6.60802e-06	230.91047	6590.26958	0
IR DOWN	81226.1926	6774.2896	6.60802e-06	262.32107	6590.2695	359.4570
EXPENSE STRESS	81227.1004	6428.6260	6.60802e-06	285.7263	6590.2696	38.1064
MORTALITY STRESS	80499.8702	7246.6768	1.5544e-07	245.1087	6531.2666	29.3067
LIFE CAT	81104.4666	6566.5752	6.5980e-06	247.2460	6580.3198	4.9919
LASPE UP	87486.7067	3665.2089	6.5194e-08	157.6184	4314.5545	1130.4727
LAPSE DOWN	64525.8032	15750.1208	0.0004531	449.5712	11394.3134	0
LAPSE MASS	88719.4043	3701.0644	3.49836e-06	131.0928	3488.9662	1546.9120
SCR_{MKT}	2221.1874					
SCR_{Life}	1568.0585					
BSCR	3022.2438					

Table 3.1: Liabilities, SCR values, BSCR components and BSCR for different stress scenarios in the deterministic projection case.

By comparing the results of Table 1.2 and Table 3.1 it can be appreciated that the deterministic and the stochastic projections give coherent results. However, we can immediately notice that the liabilities results in the stochastic projection case are higher than in the deterministic case. This difference is noticeable in the liabilities components related to the lapse and the death, while expenses do not depend on the chosen simulation approach. The divergence of these results causes an overestimation of the BSCR whenever the deterministic projection of the assets is chosen.

In the death case we can suggest that this discrepancy is due to the fact that in the deterministic case, when the value of the death benefits C_death is computed, it is always taken the value of the fund at time t: indeed, for how the deterministic projection is made, the value of the funds always increases over time and it is always greater than the invested premium.

In order to give an interpretation to this result it is important to keep in mind that the deterministic projection ignores the uncertainty of the investments of the funds, and it may not capture some extreme events which increase the liabilities.

3.1. Macaulay Duration

We evaluated the Macaulay Duration of the contract only for the base case, as requested. The formula to compute it is given by the sum of the discounted liabilities of each year multiplied by the time, divided by the sum of all discounted liabilities:

$$Duration = \frac{\sum_{i=1}^{50} Liab_i \cdot discount(i) \cdot t_i}{TotLiab}$$

The Duration can be seen as the average time in which you expect to have an alliance between the value of the fund and the amount of money the company is expected to pay. The result we obtain in the deterministic base case is:

$$Duration = \mathbf{5.6131}$$

This numerical result seems to suggest that a contract with the same characteristics of the contract given in this case study lasts, on average, less than six years: this could be due to the fact that the male's population mortality rate rapidly increases after the age of 65; moreover, the lapse rate is quite relevant, therefore the probability of exiting from the contract in a short period of time is significant.

3.2. PVFP

Then, we proceeded to evaluate the sources of profit for the insurance company. Our end goal was to compute the the PVFP; to do so we started by analysing all the profit the company has, which are not connected to the liabilities. This kind of profits are the one connected to regular deductions, whereas there is also a component that the company has to pay which is the commissions portion. We supposed that since both of this terms depend on Equity and Property, they also have a deterministic behaviour. Therefore the profit behaves as follow:

$$Eq_{Comm}(t_i) = Eq_{Comm}(t_{i-1})e^{r(i)} \cdot (RD - COMM)$$
$$Pr_{Comm}(t_i) = Pr_{Comm}(t_{i-1})e^{r(i)} \cdot (RD - COMM)$$

The entire profit is given by the the sum of this two components which has also to be discounted, using the discount factors we had previously computed.
After all of this computations we obtained the following value for the PVFP:

$$PVFP = \mathbf{3\,765.8683}$$

We observe that the order of magnitude of this result is comparable with the order of magnitude of the BoF in the base case.

3.3. Leakage

The leakage is the monetary amount that can be deduced using the following relation:

LEAK= MVA - BEL - PVFP

where

- MVA: Market Value Adjustement
- BEL: Best Estimate Liabilities
- PVFP: Present Value of Future Profits

In practice, the leakage captures any mismatch between the market value of the assets held by the insurance company (MAV) and the actuarial valuation of future profits (PVFP). In our case, we obtained a leak equal to:

$$Estimated\ LEAK = \mathbf{1\,740.5158}$$

In our case the obtained leakage term is non-negligible with respect to the BoF and the Liabilities magnitude, even though it is less than 2% of the initial fund value. This discrepancy between the MVA and the sum of BEL and PVFP can arise because of different factors: it can depend on different market expectations regarding interest rates, mortality rates or inflation or it can be arise from market volatility of the economic conditions.

3.4. PVFP Proxy calculation

In order to have a rule of thumb to compute the PVFP, we retrieved a proxy value by using the value of the duration and the initial value of the fund as follows:

$$PVFP_{proxy} = (RD - COMM) \cdot F_0 \cdot Duration$$

In this formula the calculations are simplified by assuming that the profits are received for an amount of years equal to the duration and they are not discounted. The obtained results and the difference from the exact value are reported in Table 3.2.

$PVFP$	$PVFP_{proxy}$	Δ_{proxy}
3 765.8683	4 490.4587	+19%

Table 3.2: Proxy calculation of the PVFP against the exact calculation and percentage difference of the two PVFP results.

Even if the variation of the proxy value of the PVFP from the exact value is relevant (19%), the two values have the same order of magnitude, thus this proxy calculation can be an effective and easy to implement rule of thumb to compute the PVFP.

4 | Open Questions

4.1. Parallel shift Rates curve

In this section we repeated the same calculation as the first and the second point considering a parallel shift, both positive and negative, of 100bps of the interest rates curve: this shift would have had a huge impact on the discount factors and on the forward rates used to simulate the assets' values (obviously the assets' values would have increased in case of a positive shift, while it would have decreased with respect to the "standard" case, when a negative shift was done); the changement in the simulation of the assets would not have only an impact on the funds value, but also on the benefits in cases of death and lapse.
Overall, from the trade off among these different responses to the rates' shift, we expected that the value of BOF for the base case and all the stressed cases would have been higher for the positive shift, similarly to what happened when the interest rates were stressed up in section 2.3.4. The opposite should have happened for the negative shift.

4.1.1. Positive shift

We created a new Excel file from the one we had for the EIOPA rates and we added 1% to each rate.
We obtained the following values for the base case.

Liab Death	Liab Lapse	Liab Surv	Expenses	Commi	Tot Liab	BoF	dBoF
6428.6259	81227.1087	6.6080e-06	247.4803	6590.2695	94493.4846	5506.5153	-1056.8629

Table 4.1: Liabilities, BoF and Δ_{BoF} in the case of positive shift of the rates by 1bps, via deterministic assets' simulation.

We can clearly see that all the liabilities are smaller than the standard situation, as we expected: shifting up the rates curve we obtain that the dominant effect is the decrease of the discount factors; as a consequence all the liabilities decrease, especially the death

ones.
The same holds for all the stresses, as it can be seen in the Matlab code.

4.1.2. Negative shift

We created a new Excel file from the one we had for the EIOPA rates and we subtracted 1% to each rate.
We obtained the following values for the base case.

Liab Death	Liab Lapse	Liab Surv	Expenses	Commi	Tot Liab	BoF	dBoF
7860.4672	81248.4710	6.6207e-06	247.7596	6592.0038	95548.7016	4051.2992	398.4232

Table 4.2: Liabilities, BoF and Δ_{BoF} in the case of negative shift of the rates by 1bps, via deterministic assets' simulation.

We can observe that all the liabilities are bigger than in the standard situation, as we expected: shifting down the rates curve we obtain that the main result is the discount factors increase; as a consequence all the liabilities increase, especially the death ones.
The same holds for all the other stresses.

4.2. Increasing Insured age

In the assignment, we were asked to find the BSCR in the case of a 60 years old male.
In this section we had to verify what happens if the insured age increases. We considered 1 model point, a male, for every age from 61 to 65.
We computed the BoF for this interval of ages and we plotted the results in Figure 4.1.

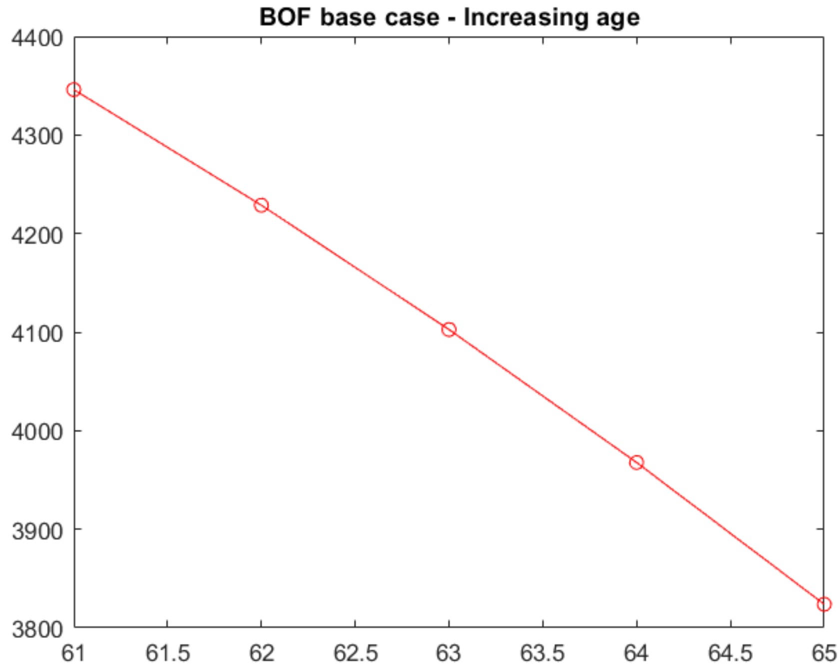

Figure 4.1: BOF base for ages between 61 and 65

From the graph we can immediately see that the BOF decreases if the insured age increases.
For example, we obtained the following values for the liabilities in the 65 year old man base case:

Liab Death	Liab Lapse	Liab Surv	Expenses	Commi	Tot Liab	BoF	dBoF
11458.6369	78144.1005	1.3230e-09	236.8974	6340.1338	96179.7687	3820.2312	629.4212

Table 4.3: Liabilities, BoF and Δ_{BoF} in the case of increasing the insured age, via deterministic assets' simulation.

These values are coherent because if the insured age increases, the value of the liabilities, especially the death liabilities, increase since the probability to die is higher.
Moreover, all the other liabilities are smaller the Lapse liabilities are smaller since the survival probabilities decrease: in particular, the Lapse liabilities decrease because they strongly depend on the survival probabilities.

4.3. Two model points

In this case we had to comment the difference between a one model point with a 60 years old male and a two model point with a 60 years old man and a 60 years old woman.
Here we took the death probability considering both male and female and we computed only the base case.

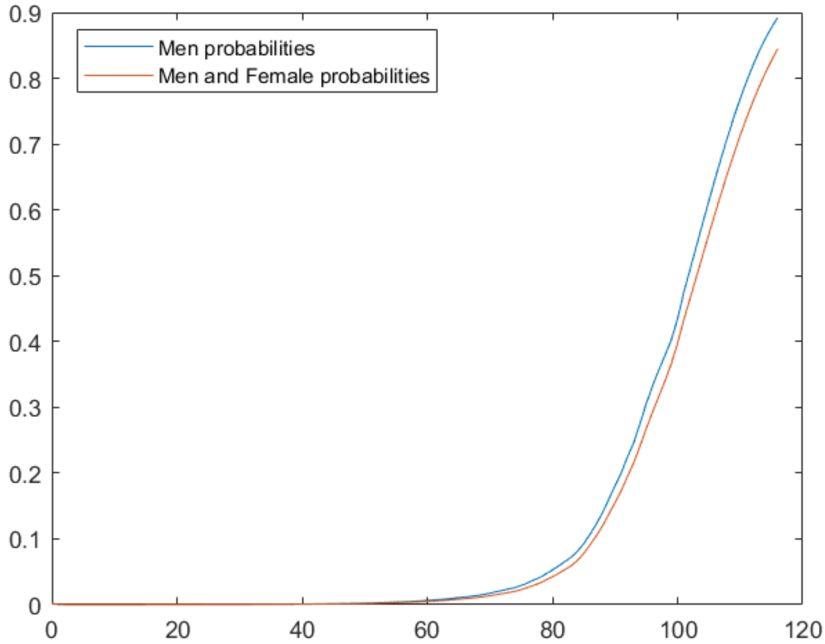

Figure 4.2: Graph representing the Only Males death probabilities (blue line) and the Males and Females death probabilities (orange line).

We can see that the death probabilities are smaller in the second case: women probabilities to die are smaller than men's ones. We can also notice that both the probabilities have the same behaviour and rapidly increase as the age increases over 60 years.
From this graph we would expect that since the death probabilities of males and females combined are lower than in the only males' case, the BoF will be higher when two model points, one male and one female are taken into account. We also computed the liabilities, the BoF and the Δ_{BoF} in this case.

Liab Death	Liab Lapse	Liab Surv	Expenses	Commi	Tot Liab	BoF	dBoF
6034.3632	82357.7861	4.6559e-05	251.4259	6682.0058	95325.5812	4674.4188	-224.7664

Table 4.4: Liabilities, BoF and Δ_{BoF} in the case of two model points, one male and one female, via deterministic assets' simulation.

We can see that the BOF is higher: we expected this result because we said that the death probabilities are smaller, so the total liabilities decrease. In particular, the death liabilities are way smaller, while the lapse ones are quite bigger.

5 | Matlab Code

5.1. Main

```
%% Insurance project AY2023-2024
% Callini Matteo, Carzaniga Carlotta, Gaspari Cecilia, Torba Matteo
clear, clc, close all;
format long
seed = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% QUESTION 1 AND 2: Compute the Basic Solvency Capital Requirement
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Read data and compute the discounts
% Save life tables' data in a struct:
% MaleData = readLifeTables('Tavole.xlsx',1); % Males life tables
% AllData = readLifeTables('Tavole.xlsx',2);  % Males and females life tables
load("MaleData.mat")
load("AllData.mat")

% Save rates' data:
% rates = readRatesData('EIOPA_RFR_20240331_Term_Structures.xlsx');
load("rates.mat")

% Compute the discounts, the forward discounts and the forward rates:
[discounts, fwd_discounts, fwd_rates] = Compute_Df_Fwddf_Fwdrates(rates.spot);

%% Data
% Assets
F0 = 1e+5;     % fund value
```

```
C0 = F0;        % invested premium value
Eq0 = 0.8*F0;   % Equity in t0
Pr0 = 0.2*F0;   % Property in t0
sigmaEq = 0.2; % Equity's volatility in GBM dynamics
sigmaPr = 0.1; % Property's volatility in GBM dynamics

x = 60; % Age of the insured
T = 50; % Years

% Liabilities
Pen = 20;          % Penalty in case of lapes
RD = 0.022;        % Regular deduction
COMM = 0.014;      % Commisions percentage
lapse = 0.15*ones(1,T); % Lapse annual probability vector
expenses = 50;     % Yearly expenses
inflation = 0.02; % Yearly inflation rate

%% Probabilities
q = MaleData.qx/1000; % death probability
p = 1-q;              % survival probability

%% Martingale Test
tic
[opt_t,opt_sim] = Martingale_Test(rates.spot,F0,sigmaEq,sigmaPr,T,seed);
figure
[opt_t_1,opt_sim_1] = Martingale_Test_1(rates.spot,F0,sigmaEq,sigmaPr,T,seed);
% Set the number of simulations equal to the otpimal value found in the
% Martingale test:
M = opt_sim_1;
fprintf("Martingale test computation time: ")
toc

%% Base scenario
% Base is a struct which contains all the liabilities and the BOF
rng(seed)
% Simulate assets, compute liabilities and BOF:
base = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
```

```
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,discounts);
% Display the results:
fprintf('\n --- Point a and Point b ---\n\n')
disp('Base scenario data:')
fprintf('- Liabilities: %.8f\n',base.liab)
fprintf('- Duration: %.8f\n',base.Duration)
fprintf('- BOF: %.8f\n\n',base.BOF)

%% Equity stress - Type 1
% Decrese in equity type 1
s1 = 0.3950 + 0.0525; % We consider the symmetric adjustment term (5.25%)
rng(seed)
% Simulate assets, compute liabilities and BOF in the equity stress type 1:
eq1 = BOF(Eq0*(1-s1),Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,Eq0*(1-s1)+Pr0,inflation,expenses,...
                  Eq0*(1-s1)+Pr0,discounts);
% Compute the SCR in the equity stress type 1:
SCR_s1 = max(base.BOF-eq1.BOF,0);

% Display the results:
disp('Equity stress type 1 data:')
fprintf('- Liabilities: %.8f\n',eq1.liab)
fprintf('- Duration: %.8f\n',eq1.Duration)
fprintf('- BOF: %.8f\n',eq1.BOF)
fprintf('- SCR: %.8f\n\n',SCR_s1)

%% Property stress
% Property shock:
P_sp = 0.25;
rng(seed)
% Simulate assets, compute liabilities and BOF in the property shock case:
pr = BOF(Eq0,Pr0*(1-P_sp),M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,Eq0+Pr0*(1-P_sp),inflation,expenses,...
                  Eq0+Pr0*(1-P_sp),discounts);
% Compute the SCR:
SCR_sp = max(base.BOF-pr.BOF,0);
```

```
% Display the results:
disp('Property stress data:')
fprintf('- Liabilities: %.8f\n',pr.liab)
fprintf('- Duration: %.8f\n',pr.Duration)
fprintf('- BOF: %.8f\n',pr.BOF)
fprintf('- SCR: %.8f\n\n',SCR_sp)

%% Interest rates UP
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates from
% the shifted rates:
[discounts_irup, fwd_discounts_irup, fwd_rates_irup] = Compute_Df_Fwddf_Fwdrates(...
                                                              rates.shockup);
rng(seed)
% Simulate assets, compute liabilities and BOF in the IR shock up case:
I_sup = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_irup,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,discounts_irup);
% Compute the SCR:
SCR_irup = max(base.BOF-I_sup.BOF,0);

% Display the results:
disp('Interest rates up stress data:')
fprintf('- Liabilities: %.8f\n',I_sup.liab)
fprintf('- Duration: %.8f\n',I_sup.Duration)
fprintf('- BOF: %.8f\n',I_sup.BOF)
fprintf('- SCR: %.8f\n\n',SCR_irup)

%% Interest rates DOWN
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates from
% the shifted rates:
[discounts_irdown, fwd_discounts_down, fwd_rates_down] = Compute_Df_Fwddf_Fwdrates(...
                                                              rates.shockdown);
rng(seed)
% Simulate assets, compute liabilities and BOF in the IR shock down case:
I_down = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_down,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,discounts_irdown);
```

```
% Compute the SCR:
SCR_irdown = max(base.BOF-I_down.BOF,0);

% Display the results:
disp('Interest rates down stress data:')
fprintf('- Liabilities: %.8f\n',I_down.liab)
fprintf('- Duration: %.8f\n',I_down.Duration)
fprintf('- BOF: %.8f\n',I_down.BOF)
fprintf('- SCR: %.8f\n\n',SCR_irdown)

%% EXPENSE stress
rng(seed) % set the seed
% Simulate assets, compute liabilities and BOF after shifting inflation and
% expenses:
exp_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation+0.01,expenses*1.1,F0,...
                  discounts);
% Compute the SCR:
SCR_exp = max(base.BOF-exp_up.BOF,0);

% Display the results:
disp('Expense stress data:')
fprintf('- Liabilities: %.8f\n',exp_up.liab)
fprintf('- Duration: %.8f\n',exp_up.Duration)
fprintf('- BOF: %.8f\n',exp_up.BOF)
fprintf('- SCR: %.8f\n\n',SCR_exp)

%% Mortality stress
rng(seed) % set the seed
% Simulate assets, compute liabilities and BOF after shifting the
% mortality:
mort = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q*1.15,x,Pen,C0,inflation,expenses,F0,...
                  discounts);
% Compute the SCR:
SCR_sm = max(base.BOF-mort.BOF,0);
```

```
% Display the results:
disp('Mortality stress data:')
fprintf('- Liabilities: %.8f\n',mort.liab)
fprintf('- Duration: %.8f\n',mort.Duration)
fprintf('- BOF: %.8f\n',mort.BOF)
fprintf('- SCR: %.8f\n\n',SCR_sm)

%% Lapse stress UP
rng(seed) % set the seed
% Lapse stressed up:
lapse_lu = min(lapse*1.5,1);
% Simulate assets, compute liabilities and BOF with the stressed Lapse:
lapse_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse_lu,q,x,Pen,C0,inflation,expenses,F0,discounts);
% Compute the SCR:
SCR_lu = max(base.BOF-lapse_up.BOF,0);

% Display the results:
disp('Lapse stress up data:')
fprintf('- Liabilities: %.8f\n',lapse_up.liab)
fprintf('- Duration: %.8f\n',lapse_up.Duration)
fprintf('- BOF: %.8f\n',lapse_up.BOF)
fprintf('- SCR: %.8f\n\n',SCR_lu)

%% Lapse stress DOWN
rng(seed) % set the seed
% Lapse stressed down:
lapse_ld = max(lapse*0.5,lapse-0.2);
% Simulate assets, compute liabilities and BOF with the stressed Lapse:
lapse_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse_ld,q,x,Pen,C0,inflation,expenses,F0,discounts);
% Compute the SCR:
SCR_ld = max(base.BOF-lapse_dw.BOF,0);

% Display the results:
disp('Lapse stress down data:')
fprintf('- Liabilities: %.8f\n',lapse_dw.liab)
```

```
fprintf('- Duration: %.8f\n',lapse_dw.Duration)
fprintf('- BOF: %.8f\n',lapse_dw.BOF)
fprintf('- SCR: %.8f\n\n',SCR_ld)

%% Lapse stress MASS
rng(seed)   % set the seed
% Stress Lapse MASS:
lapse_mass = zeros(size(lapse));
lapse_mass(1) = lapse(1)+0.4;
lapse_mass(2:end) = lapse(2:end);
% Simulate assets, compute liabilities and BOF with the stressed Lapse:
lapse_ms = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse_mass,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts);
% Compute the SCR of the Lapse MASS shock:
SCR_mass = max(base.BOF-lapse_ms.BOF,0);

% Compute the SCR for the lapse shocks:
SCR_lapse = max(SCR_lu,max(SCR_mass,SCR_ld));

% Display the results:
disp('Lapse mass stress data:')
fprintf('- Liabilities: %.8f\n',lapse_ms.liab)
fprintf('- Duration: %.8f\n',lapse_ms.Duration)
fprintf('- BOF: %.8f\n',lapse_ms.BOF)
fprintf('- SCR: %.8f\n\n',SCR_mass)

fprintf('- SCR lapse: %.8f\n\n',SCR_lapse)

%% CAT scenario
rng(seed)
q1 = q;
q1(x+1) = q(x+1) + 0.0015;
% Simulate assets, compute liabilities and BOF:
CAT = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q1,x,Pen,C0,inflation,expenses,F0,discounts);
% Compute the SCR:
```

```
SCR_CAT = max(base.BOF-CAT.BOF,0);

% Display the results:
disp('CAT stress data:')
fprintf('- Liabilities: %.8f\n',CAT.liab)
fprintf('- Duration: %.8f\n',CAT.Duration)
fprintf('- BOF: %.8f\n',CAT.BOF)
fprintf('- SCR: %.8f\n\n',SCR_CAT)

%% RM MKT
A = 0.5; % 0 UP, 0.5 DOWN
CorrM = [1, A, A; % IR
        A, 1, 0.75; % Eq
        A, 0.75, 1]; % Pr

SCR_mkt = sqrt([SCR_irdown, SCR_s1, SCR_sp]*CorrM*[SCR_irdown, SCR_s1, SCR_sp]');

% A = 0; % 0 UP, 0.5 DOWN
% CorrM = [1, A, A; % IR
%         A, 1, 0.75; % Eq
%         A, 0.75, 1]; % Pr
%
% SCR_mkt = sqrt([SCR_irup, SCR_eq, SCR_sp]*CorrM*[SCR_irup, SCR_eq, SCR_sp]');

%% RM LIFE
CorrL = [1, 0, 0.25, 0.25; % Mort
         0, 1, 0.5, 0.25; % Lap
         0.25, 0.5, 1, 0.25; % Exp
         0.25, 0.25, 0.25, 1]; % CAT

SCR_life = sqrt([SCR_sm, SCR_lapse, SCR_exp, SCR_CAT]*CorrL*[SCR_sm, ...
                SCR_lapse, SCR_exp, SCR_CAT]');

%% Matrix Mkt - Life
CorrML = [1, 0.25; % Mkt
          0.25, 1]; % Life
```

```
BSCR = sqrt([SCR_mkt, SCR_life]*CorrML*[SCR_mkt, SCR_life]');

fprintf('- SCR life: %.8f\n\n',SCR_life)
fprintf('- SCR market: %.8f\n\n',SCR_mkt)
fprintf('- BSCR: %.8f\n\n',BSCR)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Point d

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Open questions
fprintf('\n --- Point d ---\n\n')
% Save rates' data:
rates_SHIFTED = readRatesData('1bpsSHIFT_UP_EIOPA_RFR_20240331.xlsx');

% Positive shift
[discounts_up, fwd_discounts_up, fwd_rates_up] = Compute_Df_Fwddf_Fwdrates(...
                                                 rates_SHIFTED.spot);

%% Base scenario
% Set the seed:
rng(seed)
base_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,discounts_up,3);

% Display the results:
disp('Base scenario data - Shift +1 bps rates curve')
fprintf('- Liabilities: %.8f\n',base_up.liab)
fprintf('- Duration: %.8f\n',base_up.Duration)
fprintf('- BOF: %.8f\n\n',base_up.BOF)

%% Equity stress - Type 1
rng(seed)
eq1_up = BOF(Eq0*(1-s1),Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,Eq0*(1-s1)+Pr0,...
                  discounts_up);
```

```
%% Property stress
rng(seed)
pr_up = BOF(Eq0,Pr0*(1-P_sp),M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,Eq0+Pr0*(1-P_sp),...
                  discounts_up);

%% Interest rates UP
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates:
[discounts_irup_su, ~, fwd_rates_irup_su] = Compute_Df_Fwddf_Fwdrates(...
    rates_SHIFTED.shockup);
rng(seed)
I_sup_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_irup_su,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_irup_su);

%% Interest rates DOWN
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates:
[discounts_irdown_sd, ~, fwd_rates_down_sd] = Compute_Df_Fwddf_Fwdrates(...
    rates_SHIFTED.shockdown);
rng(seed)
I_down_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_down_sd,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_irdown_sd);

%% EXPENSE stress
rng(seed) % set the seed
exp_up_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation+0.01,expenses*1.1,...
                  F0,discounts_up);

%% Mortality stress
rng(seed) % set the seed
mort_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse,q*1.15,x,Pen,C0,inflation,expenses,F0,...
```

```
                  discounts_up);

%% Lapse stress UP
rng(seed) % set the seed
lapse_up_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse_lu,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_up);

%% Lapse stress DOWN
rng(seed) % set the seed
lapse_dw_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse_ld,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_up);

%% Lapse stress MASS
rng(seed)   % set the seed
lapse_ms_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse_mass,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_up);

%% CAT scenario
rng(seed)
CAT_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,...
                  RD,COMM,lapse,q1,x,Pen,C0,inflation,expenses,F0,discounts_up);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Negative shift
% Save rates' data:
rates_SHIFTED_DOWN = readRatesData('1bpsSHIFT_DOWN_EIOPA_RFR_20240331.xlsx');
[discounts_dw, fwd_discounts_dw, fwd_rates_dw] = Compute_Df_Fwddf_Fwdrates(...
                                                 rates_SHIFTED_DOWN.spot);

%% Base scenario
% Set the seed:
rng(seed)
base_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,discounts_dw);
```

```
% Display the results:
disp('Base scenario data - Shift -1 bps rates curve')
fprintf('- Liabilities: %.8f\n',base_dw.liab)
fprintf('- Duration: %.8f\n',base_dw.Duration)
fprintf('- BOF: %.8f\n\n',base_dw.BOF)

%% Equity stress - Type 1
rng(seed)
eq1_dw = BOF(Eq0*(1-s1),Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,Eq0*(1-s1)+Pr0,...
                  discounts_dw);

%% Property stress
rng(seed)
pr_dw = BOF(Eq0,Pr0*(1-P_sp),M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,Eq0+Pr0*(1-P_sp),...
                  discounts_dw);

%% Interest rates UP
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates:
[discounts_irup_su, fwd_discounts_irup_su, fwd_rates_irup_su] = ...
                      Compute_Df_Fwddf_Fwdrates(rates_SHIFTED_DOWN.shockup);
rng(seed)
I_sup_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_irup_su,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_irup_su);

%% Interest rates DOWN
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates:
[discounts_irdown_sd, fwd_discounts_down_sd, fwd_rates_down_sd] = ...
                    Compute_Df_Fwddf_Fwdrates(rates_SHIFTED_DOWN.shockdown);
rng(seed)
I_down_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_down_sd,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,...
```

```
                  discounts_irdown_sd);

%% EXPENSE stress
rng(seed) % set the seed
exp_up_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation+0.01,expenses*1.1,F0,...
                  discounts_dw);

%% Mortality stress
rng(seed) % set the seed
mort_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse,q*1.15,x,Pen,C0,inflation,expenses,F0,...
                  discounts_dw);

%% Lapse stress UP
rng(seed) % set the seed
lapse_up_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse_lu,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_dw);

%% Lapse stress DOWN
rng(seed) % set the seed
lapse_dw_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse_ld,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_dw);

%% Lapse stress MASS
rng(seed)   % set the seed
lapse_ms_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse_mass,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts_dw);

%% CAT scenario
rng(seed)
CAT_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,...
                  RD,COMM,lapse,q1,x,Pen,C0,inflation,expenses,F0,...
                  discounts_dw);
```

```
%% Plots
% Assets computation
[Eq,Pr,~] = assets(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,RD,COMM);
% Sum the simulated assets:
S = Eq + Pr;

% Assets computation
[Eq_up,Pr_up,~] = assets(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_up,RD,COMM);
% Sum the simulated assets:
S_up = Eq_up + Pr_up;

% Assets computation
[Eq_dw,Pr_dw,~] = assets(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_dw,RD,COMM);
% Sum the simulated assets:
S_dw = Eq_dw + Pr_dw;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Increasing age
% Base scenario

% Set the seed:
BOF_age = zeros(1,5);
for i=1:5
    rng(seed)
    base_1 = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                      RD,COMM,lapse,q,x+i,Pen,C0,inflation,expenses,...
                      F0,discounts);
    % Display the results:
    fprintf('Base scenario data - Insured age = %d:\n',x+i)
    fprintf('- Liabilities: %.8f\n',base_1.liab)
    fprintf('- Duration: %.8f\n',base_1.Duration)
    fprintf('- BOF: %.8f\n\n',base_1.BOF)
    BOF_age(i) = base_1.BOF;
end

figure
plot(x+1:x+5,BOF_age,'-ro')
```

```
title('BOF base case - Increasing age')

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Males and females
% 1 man - 1 female
% Probabilities
q_all = AllData.qx/1000; % death probability
p_all = 1-q_all; % survival probability

figure
plot(0:length(q)-1,q,0:length(q_all)-1,q_all)
legend('Men probabilities','Men and Female probabilities')

%% Base scenario
% Set the seed:
rng(seed)
base_all = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q_all,x,Pen,C0,inflation,expenses,F0,...
                  discounts);

% Display the results:
disp('Base scenario data - 1 male and 1 female')
fprintf('- Liabilities: %.8f\n',base_all.liab)
fprintf('- Duration: %.8f\n',base_all.Duration)
fprintf('- BOF: %.8f\n\n',base_all.BOF)
```

5.2. Functions

5.2.1. assets

```
    function [Eq,Pr,S_comm,S_profit] = assets(Eq0,Pr0,M,T,sigmaEq,sigmaPr,...
                                        fwd,RD,COMM)
% Function to simulate the stochastic dynamics of the equity and of the
% property via a GBM.
%
% INPUTS:
```

```
% Eq0:       Equity's value in t0
% Pr0:       Property's value in t0
% M:         Number of simulations
% T:         Number of years
% sigmaEq:   Equity's GBM volatility
% sigmaPr:   Property's GBM volatility
% fwd:       Forward rates curve
% RD:        Regular deduction from the fund value
% COMM:      Commisions to be computed on the fund value
%
% OUTPUTS:
% Eq:        Matrix of simulated equity values
% Pr:        Matrix of simulated property values
% S_comm:    Commissions matrix: element in position i,j represents the
%            commission in the i-th simulation and in the j-th year
% S_profit: Profits matrix: element in position i,j represents the
%            profit in the i-th simulation and in the j-th year

% Initialize the output variables:
Eq = zeros(M,T+1);
Pr = zeros(M,T+1);
Eq_comm = zeros(M,T+1);
Pr_comm = zeros(M,T+1);
Eq_profit = zeros(M,T+1);
Pr_profit = zeros(M,T+1);
Eq(:,1) = Eq0;
Pr(:,1) = Pr0;
% Vector of times:
dt = ones(1,T);
% Normal random variables:
g = randn(M,T);
g1 = randn(M,T);

for i=1:T
    % Equity and property values at the i-th year for every simulation:
    Eq(:,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)+sigmaEq*sqrt(dt(i))...
            *g(:,i))*(1-RD);
```

```
    Pr(:,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)+sigmaPr*sqrt(dt(i))*...
                g1(:,i))*(1-RD);
    % Commissions on equity and property values at the i-th year for every
    % simulation:
    Eq_comm(:,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)+sigmaEq*...
                     sqrt(dt(i))*...
                     g(:,i))*COMM;
    Pr_comm(:,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)+sigmaPr*...
                     sqrt(dt(i))*...
                     g1(:,i))*COMM;
    % Compute the profit at the i-th year for every simulation:
    Eq_profit(:,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)+sigmaEq*...
                       sqrt(dt(i))*...
                       g(:,i))*(RD-COMM);
    Pr_profit(:,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)+sigmaPr*...
                       sqrt(dt(i))*...
                       g1(:,i))*(RD-COMM);

end

% Matrix of total commissions computed on the total fund value:
S_comm = Eq_comm(:,2:end) + Pr_comm(:,2:end);

% Matrix of profits:
S_profit = Eq_profit(:,2:end) + Pr_profit(:,2:end);

end
```

5.2.2. assets_antithetic_variables

```
    function [Eq,Pr,S_comm,S_profit] = assets_antithetic_variables(Eq0,Pr0,M,...
                                    T,sigmaEq,sigmaPr,fwd,RD,COMM)
% Function to simulate the stochastic dynamics of the equity and of the
% property via a GBM by using the antithetic variables technique.
%
% INPUTS:
% Eq0:      Equity's value in t0
```

```
% Pr0:       Property's value in t0
% M:         Number of simulations
% T:         Number of years
% sigmaEq:   Equity's GBM volatility
% sigmaPr:   Property's GBM volatility
% fwd:       Forward rates curve
% RD:        Regular deduction from the fund value
% COMM:      Commisions to be computed on the fund value
%
% OUTPUTS:
% Eq:        Matrix of simulated equity values
% Pr:        Matrix of simulated property values
% S_comm:    Commissions matrix: element in position i,j represents the
%            commission in the i-th simulation and in the j-th year
% S_profit: Profits matrix: element in position i,j represents the
%            profit in the i-th simulation and in the j-th year

% Initialize the output variables:
Eq = zeros(M,T+1);
Pr = zeros(M,T+1);
Eq_comm = zeros(M,T+1);
Pr_comm = zeros(M,T+1);
Eq(:,1) = Eq0;
Pr(:,1) = Pr0;
% Vector of times:
dt = ones(1,T);
% Normal random variables:
g = randn(M/2,T);
g1 = randn(M/2,T);

for i=1:T
    % Equity and property values at the i-th year for every simulation:
    Eq(1:M/2,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)+sigmaEq*...
                    sqrt(dt(i))*g(:,i))*(1-RD);
    Pr(1:M/2,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)+sigmaPr*...
                    sqrt(dt(i))*g1(:,i))*(1-RD);
    Eq(M/2+1:M,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)-sigmaEq*...
```

```
                      sqrt(dt(i))*g(:,i))*(1-RD);
    Pr(M/2+1:M,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)-sigmaPr*...
                      sqrt(dt(i))*g1(:,i))*(1-RD);
    % Commissions on equity and property values at the i-th year for
    % every simulation:
    Eq_comm(1:M/2,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)+...
                         sigmaEq*sqrt(dt(i))*g(:,i))*COMM;
    Pr_comm(1:M/2,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)+...
                         sigmaPr*sqrt(dt(i))*g1(:,i))*COMM;
    Eq_comm(M/2+1:M,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)-...
                           sigmaEq*sqrt(dt(i))*g(:,i))*COMM;
    Pr_comm(M/2+1:M,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)-...
                           sigmaPr*sqrt(dt(i))*g1(:,i))*COMM;
    % Compute the profit at the i-th year for every simulation:
    Eq_profit(1:M/2,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)+...
                           sigmaEq*sqrt(dt(i))*g(:,i))*(RD-COMM);
    Pr_profit(1:M/2,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)+...
                           sigmaPr*sqrt(dt(i))*g1(:,i))*(RD-COMM);
    Eq_profit(M/2+1:M,i+1) = Eq(:,i).*exp((fwd(i)-sigmaEq^2/2)*dt(i)-...
                             sigmaEq*sqrt(dt(i))*g(:,i))*(RD-COMM);
    Pr_profit(M/2+1:M,i+1) = Pr(:,i).*exp((fwd(i)-sigmaPr^2/2)*dt(i)-...
                             sigmaPr*sqrt(dt(i))*g1(:,i))*(RD-COMM);
end

% Matrix of total commissions computed on the total fund value:
S_comm = Eq_comm(:,2:end) + Pr_comm(:,2:end);

% Matrix of profits:
S_profit = Eq_profit(:,2:end) + Pr_profit(:,2:end);

end
```

5.2.3. assets_det

```
    function [Eq,Pr,S_comm,S_profit] = assets_det(Eq0,Pr0,T,rates,RD,COMM)
% Function to simulate the stochastic dynamics of the equity and of the
% property via a GBM.
```

```
%
% INPUTS:
% Eq0:      Equity's value in t0
% Pr0:      Property's value in t0
% T:        Number of years
% rates:    Zero rates curve
% RD:       Regular deduction from the fund value
% COMM:     Commisions to be computed on the fund value
%
% OUTPUTS:
% Eq:       Matrix of simulated equity values
% Pr:       Matrix of simulated property values
% S_comm:   Commissions matrix: element in position i,j represents the
%           commission in the i-th simulation and in the j-th year
% S_profit: Profits matrix: element in position i,j represents the
%           profit in the i-th simulation and in the j-th year

% Initialize the output variables:
Eq = zeros(1,T+1);
Pr = zeros(1,T+1);
Eq_comm = zeros(1,T+1);
Pr_comm = zeros(1,T+1);
Eq_profit = zeros(1,T+1);
Pr_profit = zeros(1,T+1);
Eq(:,1) = Eq0;
Pr(:,1) = Pr0;

for i=1:T
    % Equity and property values at the i-th year for every simulation:
    Eq(:,i+1) = Eq(:,i)*exp(rates(i))*(1-RD);
    Pr(:,i+1) = Pr(:,i)*exp(rates(i))*(1-RD);
    % Commissions on equity and property values at the i-th year for every simulation:
    Eq_comm(:,i+1) = Eq(:,i)*exp(rates(i))*COMM;
    Pr_comm(:,i+1) = Pr(:,i)*exp(rates(i))*COMM;
    % Compute the profit at the i-th year for every simulation:
    Eq_profit(:,i+1) = Eq(:,i)*exp(rates(i))*(RD-COMM);
    Pr_profit(:,i+1) = Pr(:,i)*exp(rates(i))*(RD-COMM);
```

```
end

% Matrix of total commissions computed on the total fund value:
S_comm = Eq_comm(:,2:end) + Pr_comm(:,2:end);

% Matrix of profits:
S_profit = Eq_profit(:,2:end) + Pr_profit(:,2:end);

end
```

5.2.4. BOF

```
    function base = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,RD,COMM,lapse,q,...
                    x,Pen,C0,inflation,expenses,F0,discounts,flag)

% Function to simulate the assets, compue the liabilities and the BOF given
% the input parameters.
%
% INPUTS:
% Eq0:          Equity's value in t0
% Pr0:          Property's value in t0
% M:            Number of simulations
% T:            Number of years
% sigmaEq:      Equity's GBM volatility
% sigmaPr:      Property's GBM volatility
% fwd_rates:    Forward rates curve
% RD:           Regular deduction from the fund value
% COMM:         Commisions to be computed on the fund value
% lapse:        Annual lapse probability vector
% q:            Death probability
% x:            Age of the insured person
% Pen:          Penalties applied in case of lapse
% C0:           Invested premium value
% inflation:    Inflation annually rate
% expenses:     Yearly expenses
% F0:           Fund initial value
% discounts:    Vector of discount factors
```

```
% flag:          Flag variable to choose the technique to do the simulation:
%                -if flag==1, do the 'classic' simulation
%                -if flag==2, do the simulation via antithetic variables
%                             technique
%                -if flag==3 do the deterministic projection
%
% OUTPUT:
% base:          Struct of the output with the following fields:
%                -base.Liab_death -> vector of yearly liabilities in case of
%                                    death (not discounted)
%                -base.Liab_lapse -> vector of yearly liabilities in case of
%                                    lapse (not discounted)
%                -base.Liab_survive -> vector of yearly liabilities in case of
%                                      no death nor lapse (not discounted)
%                -base.Expense -> vector of yearly expenses (not discounted)
%                -base.Commissions -> vector of yearly commissions
%                                     (not discounted)

% Set the flag variable such that the 'classic' simulation is done if it is
% not specified differently:
if nargin<19
    flag = 1;
end

% Simulate equity, property and commisions matrix:
if flag == 2 % simulation via antithetic variables technique
    [Eq,Pr,Comm_mat,Profit_mat] = assets_antithetic_variables(Eq0,Pr0,M,T,...
                                    sigmaEq,sigmaPr,fwd_rates,RD,COMM);
elseif flag==3 % deterministic projection
    [Eq,Pr,Comm_mat,Profit_mat] = assets_det(Eq0,Pr0,T,fwd_rates,RD,COMM);
else    % 'classic' simulation
    [Eq,Pr,Comm_mat,Profit_mat] = assets(Eq0,Pr0,M,T,sigmaEq,sigmaPr,...
                                    fwd_rates,RD,COMM);
end
% Sum the simulated assets:
S = Eq + Pr;
```

```
% Compute the liabilities:
[base.Liab_death, base.Liab_lapse, base.Liab_survive, base.Expense,...
        base.Commissions,base.Profit] = liabilities(S,T,lapse,q,x,...
                             Pen,C0,inflation,expenses,Comm_mat,Profit_mat);
% Compute the duration:
liab_duration = (((base.Liab_death + base.Liab_lapse + base.Expense +...
                 base.Commissions) + [zeros(1,T-1),base.Liab_survive]).*...
                 (1:T))*discounts(1:T);
disc_liab = ((base.Liab_death + base.Liab_lapse + ...
                base.Expense + base.Commissions) +...
                [zeros(1,T-1),base.Liab_survive])*discounts(1:T);
base.Duration = liab_duration/disc_liab;

% Discount and sum up all the liabilities:
base.Liab_death = base.Liab_death*discounts(1:T);
base.Liab_lapse = base.Liab_lapse*discounts(1:T);
base.Expense = base.Expense*discounts(1:T);
base.Commissions = base.Commissions*discounts(1:T);
base.Liab_survive = base.Liab_survive*discounts(T);
base.liab = (base.Liab_death + base.Liab_lapse + base.Expense + ...
             base.Commissions) + base.Liab_survive;

% Compute the BOF:
base.BOF = F0-base.liab;

% Discount the profits:
base.Profit = base.Profit*discounts(1:T);

end
```

5.2.5. Compute_Df_Fwddf_Fwdrates

```
    function [discounts, fwd_discounts, fwd_rates] = ...
                                        Compute_Df_Fwddf_Fwdrates(rates)
% Function to compute the discounts, the forward discounts and the forward
% rates from the spot rates.
%
```

```
% INPUT:
% rates:            Vector of discrete annual compuended rates
%
% OUTPUTS:
% discounts:        Vector of annual discounts
% fwd_discounts:    Vector of annual forward discounts
% fwd_rates:        Vector of forward rates computed in t0

% Compute the discounts:
% discounts = exp(-rates.*(1:length(rates))');
discounts = (1+rates).^(-(1:length(rates))');
% Compute the forward discounts:
fwd_discounts = discounts./[1;discounts(1:end-1)];
% Compute the forward rates:
fwd_rates = -log(fwd_discounts);

end
```

5.2.6. funds

```
    function [C_lapse, C_death] = funds(S,Pen,C0)
% Function to compute the lapse and death benefits of the contract.
%
% INPUTS:
% S:          Assets value
% Pen:        Penalty in case of lapse
% C0:         Initial capital
%
% OUTPUTS:
% C_lapse:  Lapse benefits
% C_death:  Death benefits

% Compute the benefits:
C_lapse = (S(:,2:end)-Pen);
C_death = max(S(:,2:end),C0);

end
```

5.2.7. liabilities

```
function [Liab_death, Liab_lapse, Liab_survive, Expense, Commissions, ...
            Profits] = liabilities(S,T,lapse,q,x,Pen,C0,infl,expen,...
            Comm_mat,Profit_mat)

% Function to compute the liabilities and the profits.
%
% INPUTS:
% S:              Assets value
% T:              Time (in years) to the end of the contract
% lapse:          Lapse annual probability vector
% q:              Death probability
% x:              Age of the insured person
% Pen:            Penalties to pay in case of lapse of the insured
% C0:             Invested premium value
% infl:           Yearly inflation rate
% expen:          Yearly expenses
% Comm_mat:       Commissions matrix: element in position i,j represents the
%                 commission in the i-th simulation and in the j-th year
% Profit_mat:     Profits matrix: element in position i,j represents the
%                 profit in the i-th simulation and in the j-th year
%
% OUTPUTS:
% Liab_death:     Vector of yearly liabilities caused by death
% Liab_lapse:     Vector of yearly liabilities caused by lapse
% Liab_survive:   Vector of yearly liabilities caused by survival and not lapse
%                 of the insured until the end of the contract
% Expense:        Vector of yearly liabilities caused by expenses
% Commissions:    Vector of yearly liabilities caused by commissions
% Profits:        Vector of profits

% Initialization of the survival and the "not lapse" probabilities:
surv_prob = zeros(T+1,1); % probability to survive from x to x+t for each t
surv_prob(1) = 1;
prob_not_lapse = zeros(T+1,1); % probability to survive from x to x+t for
                               % each t
```

```
prob_not_lapse(1) = 1;

% Compute the survival probabilities and the "not lapse" probabilities:
for i=1:T
    surv_prob(i+1) = surv_prob(i)*(1-q(x+i));
    prob_not_lapse(i+1) = prob_not_lapse(i)*(1-lapse(i)); % Consider the
                                                           % cumulative product
                                                           % of the non-lapse
                                                           % probability
end

% Compute lapse and death benefits
[C_lapse,C_death] = funds(S,Pen,C0);

% Liabilities in case of death:
Liab_death = mean(C_death.*(q((x+1):(x+T)).*(prob_not_lapse(1:end-1).*...
                surv_prob(1:end-1)))',1);
% Liabilities in case of lapse:
Liab_lapse = mean(C_lapse.*(lapse'.*(prob_not_lapse(1:end-1).*surv_prob...
                  (2:end)))',1);

% Liabilities in case of survival:
Liab_survive = mean(surv_prob(end)*S(:,end)*prob_not_lapse(end),1);
% Liabilities due to expenses:
Expense = (expen*(1+infl).^(0:T-1)'.*prob_not_lapse(2:end).*surv_prob(2:end))';
% Liabilities due to commissions:
Commissions = mean(Comm_mat.*(prob_not_lapse(2:end).*surv_prob(2:end))',1);

% Profits:
Profits = mean(Profit_mat.*(prob_not_lapse(2:end).*surv_prob(2:end))',1);

end
```

5.2.8. Martingale_Test

```
function [n_dt,n_simulations]=Martingale_Test(rates,F0,sigmaEq,sigmaPr,T,...
                          seed)
```

```
% Function to compute the Martingale Test
%
% INPUTS
% rates :            Vector of discrete annual compuended rates
% F0 :               Initial Fund value
% sigmeEq :          Equity's volatility
% sigmaPr :          Property's volatility
% T :                Time horizon
% seed :             Seed for random generation
%
% OUTPUTS
% n_dt :             Best number of time steps
% n_simulations :    Best number of simulations

rates=[0;rates];
rates = rates(1:T+1);
% setting a time of simulations and a number of steps
N_sim=1e4:1e4:1e5;
N_timesteps=50:50:200;
err=zeros(length(N_sim),length(N_timesteps));
for i=1:length(N_sim)
    Number_Simulations=N_sim(i);
    for j=1:length(N_timesteps)
        Number_TimeSteps=N_timesteps(j);
        % scenarios generation
        rng(seed)
        g=randn(Number_Simulations,Number_TimeSteps);
        time_step=linspace(0,T,Number_TimeSteps+1)';
        length_dt= T/Number_TimeSteps;
        % interpolating the rates on the dates we just found
        rates_interp=interp1((0:T)',rates,time_step);
        discounts = (1+rates_interp).^(-(1:length(rates_interp))');
        % Discounts
        fwd_discounts=(discounts(2:end)./discounts(1:end-1))';
        fwd_rates=-log(fwd_discounts);
        S_eq=zeros(Number_Simulations,Number_TimeSteps+1);
        S_pr=zeros(Number_Simulations,Number_TimeSteps+1);
```

```
        S_eq(:,1)=F0; % Equity
        S_pr(:,1)=F0; % Property

        for k=2:Number_TimeSteps+1
             S_eq(:,k)=S_eq(:,k-1).*exp((fwd_rates(k-1)-sigmaEq^2/2)*...
                              length_dt+sigmaEq*sqrt(length_dt).*g(:,k-1));
             S_pr(:,k)=S_pr(:,k-1).*exp((fwd_rates(k-1)-sigmaPr^2/2)*...
                              length_dt+sigmaPr*sqrt(length_dt).*g(:,k-1));
        end
        S=0.8*S_eq+0.2*S_pr;
        err(i,j) = norm(-F0*exp(rates_interp.*time_step)+ mean(S)');
        plot(0:length_dt:T,-F0*exp(rates_interp.*time_step)+mean(S)')
        hold on
    end
end

plot(0:T,zeros(T+1,1))
title("Assets simulation")
% hold on
% legend('n\_simulation = 1e4, n\_time\_step = 1e2',...
%      'n\_simulation = 1e4, n\_time\_step = 1e3',...
%      'n\_simulation = 1e5, n\_time\_step = 1e2',...
%      'n\_simulation = 1e5, n\_time\_step = 1e3','error=0');
% hold off
[minimumValue, linearIndex] = min(err(:));
[rowIndex, colIndex] = ind2sub(size(err), linearIndex);
n_dt=N_timesteps(colIndex);
n_simulations=N_sim(rowIndex);
fprintf(['The minimum error is %d, the best number of time steps is %d and ' ...
    '                of simulations %d.\n'], minimumValue, n_dt, n_simulations);
end
```

5.2.9. Martingale_Test_1

```
    function [n_dt,n_simulations]=Martingale_Test_1(rates,F0,sigmaEq,sigmaPr,T,...
                                seed)
% Function to compute the Martingale Test in the case of a single number of
```

```
% possible time steps
%
% INPUTS
% rates :           Vector of discrete annual compuended rates
% F0 :              Initial Fund value
% sigmeEq :         Equity's volatility
% sigmaPr :         Property's volatility
% T :               Time horizon
% seed :            Seed for random generation
%
% OUTPUTS
% n_dt :            Best number of time steps
% n_simulations :   Best number of simulations

rates=[0;rates];
rates = rates(1:T+1);
% setting a time of simulations and a number of steps
N_sim=1e4:1e4:1e5;
N_timesteps=T;
err=zeros(length(N_sim),length(N_timesteps));
for i=1:length(N_sim)
    Number_Simulations=N_sim(i);
    for j=1:length(N_timesteps)
        Number_TimeSteps=N_timesteps(j);
        % scenarios generation
        rng(seed)
        g=randn(Number_Simulations,Number_TimeSteps);
        time_step=linspace(0,T,Number_TimeSteps+1)';
        length_dt= T/Number_TimeSteps;
        % interpolating the rates on the dates we just found
        rates_interp=interp1((0:T)',rates,time_step);
        discounts = (1+rates_interp).^(-(1:length(rates_interp))');
        % Discounts
        fwd_discounts=(discounts(2:end)./discounts(1:end-1))';
        fwd_rates=-log(fwd_discounts);
        S_eq=zeros(Number_Simulations,Number_TimeSteps+1);
        S_pr=zeros(Number_Simulations,Number_TimeSteps+1);
```

```
        S_eq(:,1)=F0; % Equity
        S_pr(:,1)=F0; % Property

        for k=2:Number_TimeSteps+1
             S_eq(:,k)=S_eq(:,k-1).*exp((fwd_rates(k-1)-sigmaEq^2/2)*...
                       length_dt+sigmaEq*sqrt(length_dt).*g(:,k-1));
             S_pr(:,k)=S_pr(:,k-1).*exp((fwd_rates(k-1)-sigmaPr^2/2)*...
                       length_dt+sigmaPr*sqrt(length_dt).*g(:,k-1));
        end
        S=0.8*S_eq+0.2*S_pr;
        err(i,j) = norm(-F0*exp(rates_interp.*time_step)+ mean(S)');
        plot(0:length_dt:T,-F0*exp(rates_interp.*time_step)+mean(S)')
        hold on
    end
end

plot(0:T,zeros(T+1,1))
title("Assets simulation")
% hold on
% legend('n\_simulation = 1e4, n\_time\_step = 1e2',...
%      'n\_simulation = 1e4, n\_time\_step = 1e3',...
%      'n\_simulation = 1e5, n\_time\_step = 1e2',...
%      'n\_simulation = 1e5, n\_time\_step = 1e3','error=0');
% hold off
[minimumValue, linearIndex] = min(err(:));
[rowIndex, colIndex] = ind2sub(size(err), linearIndex);
n_dt=N_timesteps(colIndex);
n_simulations=N_sim(rowIndex);
fprintf(['The minimum error is %d, the best number of time steps is  %d and ' ...
    'of simulations %d.\n'], minimumValue, n_dt, n_simulations);
end
```

5.2.10. readLifeTables

```
    function Data = readLifeTables(filename, flag)
% Function to extract data from the life tables saved in a .xls file
%
```

```
% INPUTS:
% filename: Name of the file
% flag:     Flag variable:
%           -if flag==1, consider males only
%           -if flag==2, consider both males and females
%
% OUTPUT:
% Data:     Struct with extracted data

% Extract data on the age:
Data.x = xlsread(filename,flag,'A3:A119');
% Extract data on survivors:
Data.lx = xlsread(filename,flag,'B3:B119');
% Extract data on deaths:
Data.dx = xlsread(filename,flag,'C3:C119');
% Extract data on death probability (per one thousand):
Data.qx = xlsread(filename,flag,'D3:D119');
% Extract data on years lived:
Data.Lx = xlsread(filename,flag,'E3:E119');
% Extract data on probabilities on prospective of survival:
Data.Px = xlsread(filename,flag,'F3:F119');
% Extract data on life expectancy:
Data.ex = xlsread(filename,flag,'G3:G119');

end
```

5.2.11. readRatesData

```
    function rates = readRatesData(filename)
% Function to extract data about the rates from EIOPA saved in a .xls file
%
% INPUTS:
% filename: Name of the file
%
% OUTPUT:
% rates:    Struct with extracted data
```

```
% Extract European spot rates (no VA):
rates.spot = xlsread(filename,3,'C11:C160');
% Extract European spot rates (no VA) shocked up:
rates.shockup = xlsread(filename,5,'C11:C160');
% Extract European spot rates (no VA) shocked down:
rates.shockdown = xlsread(filename,6,'C11:C160');

end
```

5.3. Deterministic Case

```
    %% POINT c: Deterministic case
% The function BOF is called everytime with the flag variable of the asset
% simlation equal to 3 in order to simulate the assets in a deterministic
% way.
clc, clear, close all;
format long
seed = 10;

%% Read data and compute the discounts
% Save life tables' data in a struct:
% MaleData = readLifeTables('Tavole.xlsx',1); % Males life tables
% AllData = readLifeTables('Tavole.xlsx',2);  % Males and females life tables
load("MaleData.mat")
load("AllData.mat")

% Save rates' data:
% rates = readRatesData('EIOPA_RFR_20240331_Term_Structures.xlsx');
load("rates.mat")

%%
% Compute the discounts, the forward discounts and the forward rates:
[discounts, fwd_discounts, fwd_rates] = Compute_Df_Fwddf_Fwdrates...
                                        (rates.spot);


%% Data
```

```
% Assets
F0 = 1e+5;     % fund value
C0 = F0;       % invested premium value
Eq0 = 0.8*F0;  % Equity in t0
Pr0 = 0.2*F0;  % Property in t0
sigmaEq = 0.2; % Equity's volatility in GBM dynamics
sigmaPr = 0.1; % Property's volatility in GBM dynamics

x = 60; % Age of the insured
T = 50; % Years

% Liabilities
Pen = 20;          % Penalty in case of lapes
RD = 0.022;        % Regular deduction
COMM = 0.014;      % Commisions percentage
lapse = 0.15*ones(1,T); % Lapse annual probability vector
expenses = 50;     % Yearly expenses
inflation = 0.02; % Yearly inflation

%% Probabilities
q = MaleData.qx/1000; % death probability
p = 1-q;              % survival probability

%% Martingale test:
[opt_t,opt_sim] = Martingale_Test(rates.spot,F0,sigmaEq,sigmaPr,T,seed);
figure
[opt_t_1,opt_sim_1] = Martingale_Test_1(rates.spot,F0,sigmaEq,sigmaPr,...
                        T,seed);
% Set the number of simulations equal to the otpimal value found in the
% Martingale test:
M = opt_sim_1;

%% Base scenario
% Base is a struct which contains all the liabilities and the BOF
rng(seed)
% Simulate assets, compute liabilities and BOF:
base = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
```

```
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,discounts,3);

% Esimation of the LEAK:
LEAK = F0-base.liab-base.Profit;
% Proxy of the profit:
rates_for_proxy = interp1(1:length(rates.spot),rates.spot,1:floor...
                    (base.Duration));
profit_proxy = sum((RD-COMM)*F0*(1+rates_for_proxy).^(-(1:floor(...
                    base.Duration))));

% Display the results:
disp('Base scenario data:')
fprintf('- Liabilities: %.8f\n',base.liab)
fprintf('- Duration: %.8f\n',base.Duration)
fprintf('- BOF: %.8f\n',base.BOF)
fprintf('- PVFP: %.8f\n',base.Profit)
fprintf('- PVFP proxy: %.8f\n',profit_proxy)
fprintf('- Estimated LEAK: %.8f\n\n',LEAK)

%% Equity stress - Type 1
% Decrese in equity type 1:
s1 = 0.39;
rng(seed)
% Simulate assets, compute liabilities and BOF in the equity stress type 1:
eq1 = BOF(Eq0*(1-s1),Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,Eq0*(1-s1)+Pr0,inflation,expenses,...
                  Eq0*(1-s1)+Pr0,discounts,3);
% Compute the SCR in the equity stress type 1:
SCR_s1 = max(base.BOF-eq1.BOF,0);

% Display the results:
disp('Equity stress type 1 data:')
fprintf('- Liabilities: %.8f\n',eq1.liab)
fprintf('- Duration: %.8f\n',eq1.Duration)
fprintf('- BOF: %.8f\n',eq1.BOF)
fprintf('- SCR: %.8f\n\n',SCR_s1)
```

```
%% Equity stress - Type 2
% Decrese in equity type 2:
s2 = 0.49;
rng(seed)
% Simulate assets, compute liabilities and BOF in the equity stress type 2:
eq2 = BOF(Eq0*(1-s2),Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,Eq0*(1-s2)+Pr0,inflation,expenses,...
                  Eq0*(1-s2)+Pr0,discounts,3)
% Compute the SCR in the equity stress type 2:
SCR_s2 = max(base.BOF-eq2.BOF,0);

Mat = [1, 0.75;
       0.75, 1];
% SCR equity:
SCR_eq = sqrt([SCR_s1, SCR_s2]*Mat*[SCR_s1, SCR_s2]');

%% Property stress
% Property shock:
P_sp = 0.25;
rng(seed)
% Simulate assets, compute liabilities and BOF in the property shock case:
pr = BOF(Eq0,Pr0*(1-P_sp),M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,Eq0+Pr0*(1-P_sp),inflation,...
                  expenses,Eq0+Pr0*(1-P_sp),discounts,3)
% Compute the SCR:
SCR_sp = max(base.BOF-pr.BOF,0);

%% Interest rates UP
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates from
% the shifted rates:
[discounts_irup, fwd_discounts_irup, fwd_rates_irup] = ...
                                    Compute_Df_Fwddf_Fwdrates(rates.shockup);
rng(seed)
% Simulate assets, compute liabilities and BOF in the IR shock up case:
I_sup = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_irup,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,F0,...
```

```
                  discounts_irup,3)
% Compute the SCR:
SCR_irup = max(base.BOF-I_sup.BOF,0);

%% Interest rates DOWN
% Shift of the interest rates taken from EIOPA rates' tables
% Compute the discounts, the forward discounts and the forward rates
% from the shifted rates:
[discounts_irdown, fwd_discounts_down, fwd_rates_down] = ...
                            Compute_Df_Fwddf_Fwdrates(rates.shockdown);
rng(seed)
% Simulate assets, compute liabilities and BOF in the IR shock down case:
I_down = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates_down,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation,expenses,...
                  F0,discounts_irdown,3)
% Compute the SCR:
SCR_irdown = max(base.BOF-I_down.BOF,0);

%% EXPENSE stress
rng(seed) % set the seed
% Simulate assets, compute liabilities and BOF after shifting inflation and
% expenses:
exp_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q,x,Pen,C0,inflation+0.01,expenses*1.1,F0,...
                  discounts,3)
% Compute the SCR:
SCR_exp = max(base.BOF-exp_up.BOF,0);

%% Mortality stress
rng(seed) % set the seed
% Simulate assets, compute liabilities and BOF after shifting the
% mortality:
mort = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q*1.15,x,Pen,C0,inflation,expenses,F0,...
                  discounts,3)
% Compute the SCR:
SCR_sm = max(base.BOF-mort.BOF,0);
```

```
%% Lapse stress UP
rng(seed) % set the seed
% Lapse stressed up:
lapse_lu = min(lapse*1.5,1);
% Simulate assets, compute liabilities and BOF with the stressed Lapse:
lapse_up = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse_lu,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts,3)
% Compute the SCR:
SCR_lu = max(base.BOF-lapse_up.BOF,0);

%% Lapse stress DOWN
rng(seed) % set the seed
% Lapse stressed down:
lapse_ld = max(lapse*0.5,lapse-0.2);
% Simulate assets, compute liabilities and BOF with the stressed Lapse:
lapse_dw = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse_ld,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts,3)
% Compute the SCR:
SCR_ld = max(base.BOF-lapse_dw.BOF,0);

%% Lapse stress MASS
rng(seed)   % set the seed
% Stress Lapse MASS:
lapse_mass = zeros(size(lapse));
lapse_mass(1) = lapse(1)+0.4;
lapse_mass(2:end) = lapse(2:end);
% Simulate assets, compute liabilities and BOF with the stressed Lapse:
lapse_ms = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse_mass,q,x,Pen,C0,inflation,expenses,F0,...
                  discounts,3)
% Compute the SCR of the Lapse MASS shock:
SCR_mass = max(base.BOF-lapse_ms.BOF,0);

% Compute the SCR for the lapse shocks:
```

```
SCR_lapse = max(SCR_lu,max(SCR_mass,SCR_ld));

%% CAT scenario
rng(seed)
q1 = q;
q1(x+1) = q(x+1) + 0.0015;
% Simulate assets, compute liabilities and BOF:
CAT = BOF(Eq0,Pr0,M,T,sigmaEq,sigmaPr,fwd_rates,...
                  RD,COMM,lapse,q1,x,Pen,C0,inflation,expenses,F0,...
                  discounts,3)
% Compute the SCR:
SCR_CAT = max(base.BOF-CAT.BOF,0);

%% RM MKT
A = 0.5; % 0 UP, 0.5 DOWN
CorrM = [1, A, A; % IR
        A, 1, 0.75; % Eq
        A, 0.75, 1]; % Pr

SCR_mkt = sqrt([SCR_irdown, SCR_eq, SCR_sp]*CorrM*[SCR_irdown, SCR_eq,...
                SCR_sp]');

% A = 0; % 0 UP, 0.5 DOWN
% CorrM = [1, A, A; % IR
%         A, 1, 0.75; % Eq
%         A, 0.75, 1]; % Pr
%
% SCR_mkt = sqrt([SCR_irup, SCR_eq, SCR_sp]*CorrM*[SCR_irup, SCR_eq, SCR_sp]');

%% RM LIFE
CorrL = [1, 0, 0.25, 0.25; % Mort
         0, 1, 0.5, 0.25; % Lap
         0.25, 0.5, 1, 0.25; % Exp
         0.25, 0.25, 0.25, 1]; % CAT

SCR_life = sqrt([SCR_sm, SCR_lapse, SCR_exp, SCR_CAT]*CorrL*[SCR_sm,...
                SCR_lapse, SCR_exp, SCR_CAT]');
```

```
%% Matrix Mkt - Life
CorrML = [1, 0.25; % Mkt
          0.25, 1]; % Life

BSCR = sqrt([SCR_mkt, SCR_life]*CorrML*[SCR_mkt, SCR_life]');
```