

# Open-Source Economic Model

Open-Source Economic Model (OSEM) is a modelling solution that allows financial companies to simulate their future financial performance. It is envisioned to be an alternative to classical internal models often found in insurance companies, pension funds, banks etc. The open design is especially suitable for medium size companies that do not have the resources for an end-to-end internal development team, but have the volume of assets that could benefit from tailored insights and scenario analysis made possible by such models.

OSEM takes into consideration the company's current assets and liabilities. It also allows the user to define any future investments and liability business. Through a simplified representation of the economy, it simulates decision-making such as trading and balance sheet rebalancing that a company would make as time progresses.

OSEM consists of different modules such as:

- Data preparation
- Investment universe definition
- Asset preparation and calibration
- Liability preparation and calibration
- Asset Liability preparation
- Asset Liability run
- Reporting

**Note that in the prototype stage not all modules are implemented. Reach out to one of the contributors to find out what parts can be used with a higher degree of confidence and which parts are still very much in development.**

## Modelling parameters

OSEM can be configured to the varied needs of the user. The user can specify the model granularity, length of the modelling (in years), the type of run, the type of curve used as the risk-free rate, the country, and the specific modelling date. Below we describe in more details each specific setting.

### Modelling granularity

The modelling granularity is yearly in the initial POC prototype. In the future, this will be extended to monthly, yearly or every 5 years.

### Length of the modelling horizon

This parameter sets the length of the modelling horizon. In the POC prototype, the length is in years. For example, a value of 20 means a modelling horizon of 20 years from the modelling date.

*END*

## Run type

This parameter differentiates different kinds of runs. The types of runs envisioned for OSL are:

- Risk neutral
- Market consistent
- Real world

The POC prototype supports only a risk neutral run, however further the roadmap envisions the development of other types of runs.

## Selection of the risk-free curve

This parameter determines the type of curve used as the risk-free rate. The POC prototype supports only the base curve and the base curve with the Volatility Adjustment published by EIOPA.

## Country

This parameter determines the home economy of the company. The POC prototype supports only one economy. In the future, multiple economies can be implemented.

*country*

## Modelling date

The modelling date is the start or time 0 of the model. The market values, curves and other parameters are assumed to be calculated as-at this timestamp. The modelling date is denoted as

*MD...*

## Data preparation

The POC prototype only supports the EIOPA risk free curve. The setting file of the run are the base country and the type of the EIOPA curve that is to be used.

The following files are imported

- Parameters.csv; Parameters related to the run
- EIOPA\_param\_file.csv; Assumed yield curve at modelling date and relevant maturities
- EIOPA\_curves\_file.csv; Parameters related to the EIOPA time 0 calibration

From the input files, the following information is retrieved:

- Vector of maturities for the EIOPA yield curve
- Vector of yields for the liquid EIOPA maturities
- Last liquid point

- Convergence parameter
- Ultimate forward rate
- Alpha parameter
- CRA parameter
- Tau parameter

## Economic environment setup

The economic environment in which the model will run is being simulated in the this step, the risk-free curve is imported and a Smith&Wilson algorithm is used to calculate a calibration vector. This vector provides a continuous term-structure function that is needed when dealing with real dates.

At a high level, the steps are the following:

- Import the complete yield curve at time 0 (At discrete points)
- Calculate 1-year forward rates
- Use forward rates to calculate projected yield curves (At discrete points)
- Use a bisection algorithm to calibrate the parameter  $\alpha$  at each projection step
- Save the calibration vector  $b$  for each projection step

A vector of modelling steps is calculated and saved for future use:

$$\text{modellingsteps} = \{MD, MD_1, \dots, MD_T\}$$

Where:

$$MD_1 = MD + 1\text{year}$$

$$MD_2 = MD + 2\text{years}$$

...

$$MD_N = MD + T\text{years}$$

## Calculation of the 1 year forward rate

As the modelling date moves forward in time, the yield curve must be recalculated at every step. To facilitate this, the 1 year forward curve is calculated.

The forward rates are calculated as:

$$fw(i) = \frac{(1 + y_i)^i}{(1 + y_{i-1})^{i-1}}$$

Where:

- $i$  is the year (Ex. 1,2,3, ...)
- $y_i$  is the yield for the year  $i$
- $fw_{EIOPA,i}$  is the 1-year forward rate between years  $i - 1$  and  $i$

## Forward yield curve

Each forward yield curve can be calculated by using the 1-year forward rates from the suitable moment onwards. The spot yield curve is calculated for each projection period. For the projection period  $j$ , the yield curve can be calculated as:

$$y_{i-j}^j = \prod_{k=j}^i (1 + fw_{EIOPA}(k))^{-(k-j)}$$

Where:

- $j$  is the modelling period at which the model is performing the calculations (The projection year starting at time  $MD_i$ )
- $i - j$  is the time period within the modelling period
- $y_k^j$  is the yield  $k$  years after the modelling period  $j$

### Calibrate Smith-Wilson on each yield curve

Each forward yield curve is used to calibrate the SW algorithm. This calibration can then be used to generate yield curves of arbitrary maturities.

Both at time 0 and in following projection years, the same calibration algorithm is performed. The algorithm looks to minimize the difference between the  $\omega$  and the ultimate forward rate as recommended by EIOPA:

$$T = \max(U + 40, 60)$$

$$g(\alpha) = |f(T) - \omega| = \frac{\alpha}{|1 - \kappa e^{\alpha T}|}$$

The calibration consists of using a bisection algorithm to minimize the difference between the ultimate forward rate  $\omega$ .

The calibration is:

$$\min \alpha$$

subject to constraints:

$$\alpha \geq \alpha_{min}$$

and

$$g(\alpha) \leq \tau$$

And  $\kappa$  is defined as:

$$\kappa = \frac{1 + \alpha u^T Q b}{\sinh(\alpha u^T) Q b}$$

$$Q = dIC$$

$$d = e^{-\omega u}$$

Where:

- $\alpha$  is the free parameter to be calibrated
- $\alpha_{min}$  is the lower bound of the allowable values
- $U$  is the last liquid point
- $g(\alpha)$  is the extrapolated
- $C$  is a matrix of cash flows

The output of this calibration is a vector of calibrated alphas

$$\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_N\}$$

and the calibration vectors

$$b = \{b_0, b_1, \dots, b_N\}$$

Note that to use a consistent calibration throughout the OSEM model, the parameter alpha is calibrated again using the bisection method even for the yield curve at "time 0" even though the already calibrated alpha is available in the EIOPA input files.

Note that other SW parameters such as the ultimate forward rate and precision are assumed to

### Saving the projected curves

The advantage of the calibration and projection using the Smith&Wilson algorithm is three-fold:

- 1) The S&W algorithm provides a continuous function therefore a yield for any date can be calculated
- 2) To initialize the continuous term-structure, only a handful of parameters is needed.
- 3) It is a well understood algorithm frequently used in practice.

Note that the parameters needed for obtaining a vector of yields for any maturity are only:

- The parameter  $\alpha$
- The vector  $b$
- List of maturities used to calibrate the vector  $b$  and  $\alpha$
- Other stationary SW parameters such as ufr,  $\tau$  etc.

## Asset input and calibration

### Government bond input and calibration

ToDo

## Corporate bond input and calibration

One of the types of assets that OSEM supports is corporate debt in the form of corporate bonds. Bonds, described in the input files, are calibrated to each specific run:

- $MD$  ... modelling date
- $I$  ... issue date
- $M$  ... maturity date
- $N$  ... notional amount
- $c$  ... coupon yield
- $MV$  ... market value at modelling date
- $BV$  ... book value at modelling date

### 1) Dates

The basic building block is a cash flow. The first step in the bond input process is to calculate dates at which the bond is expected to produce a cash flow.

The dates for all assets are separated into two sets: 1) Cash flow(s) when the principal amount is returned 2) Cash flows from coupon payments

We will denote the dates associated with the returned principal (Only one in the POC) from a hypothetical bond as:

$$t_M^d = M$$

The dates associated with coupon payments will be denoted as:

$$I < t_1^d < t_2^d < \dots < t_l^d \leq M$$

### 2) Date fractions with respect to the modelling date

The first step in refactoring for fixed income assets such as corporate bonds consist of calculating the cash flow dates that occur after the modelling date since the past cash flows are not relevant to the OSEM model. For each cash flow date, a time fraction is calculated with regards to the modelling date in years (A fraction of 1 equals exactly 1 year from the modelling date). This facilitates discounting in later stages.

The calculation of the date fractions is as follows.

First the final repayment of the notional amount is calculated. The date is refactored if the asset has not matured before the modelling date.

$$t_M = \frac{t_M - MD}{365.25}$$

The date fraction is saved and can be viewed in the following object property:

In the next step, the dates associated with the coupon payments are refactored:

$$\begin{aligned} t_1 &= \frac{t_j - MD}{365.25} \\ t_2 &= \frac{t_{j+1} - MD}{365.25} \\ &\dots \\ t_k &= \frac{t_{j+k} - MD}{365.25} \end{aligned}$$

Where:

$$j = \min_i(t_i^d \geq MD)$$

The date fractions are saved and can be viewed in the following object property:

$$datescf = \{t_1, t_2, \dots, t_{j+k}\}$$

### 3) Cashflow simulation

This section simulates the size of the cash flows in the local currency. For bonds there are two types of cash flows. The coupons and the notional payout.

Notional cash flow

The notional amount returned at the maturity of the bond is saved in the object property:

$$cf_M = N$$

And is saved as:

$$notcf = \{cf_M\}$$

Coupon cash flow

$$divcf = \{cf_1, \dots, cf_k\}$$

Where:

$$cf_i = N * c$$

### 4) Calibration

The calibration is the process of finding the correct discount rate for each bond. The correct discount rate at which the discounted price of the bond is equal to the market value of the bond observed at the modelling date.

Note that if the run is set to Risk Neutral, the sector specific spread is equal to 0 for all bonds.

The discount rate is calibrated using a constant factor called z-spread that increases-decreases the risk free discount rate such that:

$$MV = \frac{cf_1}{(1 + y(t_1) + s + z)^{t_1}} + \cdots + \frac{cf_k}{(1 + y(t_k) + s + z)^{t_k}} + \frac{cf_M}{(1 + y(t_k) + s + z)^{t_k}}$$

Where

- $N$  ... is the notional value of the bond
- $cf_i$  ... is the i-th coupon
- $cf_M$  ... is the final cash flow associated with the return of the principal
- $y(t_i)$  ... is the annual yield for a period between the modelling date and time  $t_i$
- $s$  ... is the sector specific spread
- $z$  ... is the z-Spread of the individual bond

## Equity input and calibration

OSEM supports the asset class of equities. Different kinds of shares in the company. The shares are modelled as instruments that offer a constant cash flow in the form of a dividend based on the market value at the time of payment and a constant growth as the company matures. At the end of the modelling period, a terminal value is calculated. The terminal value represents the discounted value of all future cash flows outside of the modelling period.

Each equity asset has the following fields:

- $MD$  ... is the modelling date
- $I$  ... issue date
- $END$  ... end of the modelling period
- $c$  ... dividend yield
- $MV$  ... market value at modelling date
- $BV$  ... book value at modelling date
- $r$  ... is the terminal annualized interest rate

### 1) Dates

The first step is to calculate the dates at which the equity asset generates a cash flow since its issue. This can be either a dividend or a terminal value sale.

Terminal value date

$$t_M = END$$

Dividend cash flow dates

$$I < t_1^d < \cdots < t_l^d \leq END$$

### 2) Date fractions with respect to the modelling date

Like refactoring for fixed income assets, this step consists of calculating the dates at which cash flows occur after the modelling date since the past cash flows are not considered in the model. For each cash flow date, a time fraction is calculated with regards to the modelling date in years. This facilitates discounting in later stages.

Terminal date is equal to the last day of the modelling window is calculated if the equity is issued after the modelling date but not after the last day of the modelling window.

$$t_M = \frac{t_M - MD}{365.25}$$

The date fraction is saved and can be viewed in the following object property:

$$\text{datesterminal} = \{t_M\}$$

The calculation of the date fraction for dividend payments is:

$$\begin{aligned} t_1 &= \frac{t_j^d - MD}{365.25} \\ t_2 &= \frac{t_{j+1}^d - MD}{365.25} \\ &\dots \\ t_k &= \frac{t_{j+k}^d - MD}{365.25} \end{aligned}$$

Where:

$$j = \min_i(t_i^d \geq MD)$$

Note that as OSLM starts supporting more daycount conventions, the date fraction calculation will be enhanced.

The date fractions are saved and can be viewed in the following object property:

$$\text{datescf} = \{t_1, t_2, \dots, t_k\}$$

### 3) Calibration

The calibration of an equity asset uses a Gordon growth model to calculate the implied future growth rate, which would imply the observed market value at the modelling date.

The calibration consists of finding a growth rate  $g$  such that:

$$\begin{aligned} MV &= \frac{MV * (1 + g)^{t_1} * c}{(1 + y(t_1))^{t_1}} + \frac{MV * (1 + g)^{t_2} * c}{(1 + y(t_2))^{t_2}} + \dots + \frac{MV * (1 + g)^{t_k} * c}{(1 + y(t_k))^{t_k}} + \frac{1}{(1 + y(t_M))^{t_M}} \\ 1 &= \frac{(1 + g)^{t_1} * c}{(1 + y(t_1))^{t_1}} + \frac{(1 + g)^{t_2} * c}{(1 + y(t_2))^{t_2}} + \dots + \frac{(1 + g)^{t_k} * c}{(1 + y(t_k))^{t_k}} + \frac{(1 + g)^{t_M}}{(1 + y(t_M))^{t_M}} \frac{c}{r - g} \end{aligned}$$

$$\frac{1}{c} = \frac{(1+g)^{t_1}}{(1+r-g)t_1} + \frac{(1+g)^{t_2}}{(1+r-g)t_2} + \dots + \frac{(1+g)^{t_k}}{(1+r-g)t_k} + \frac{1}{(1+r-g)t_M} \frac{(1+g)^{t_M}}{r-g}$$

## 4) Cashflows

After obtaining the growth rate, the growth of the value of the share and by consequence the size of each dividend can be projected (As well as the terminal value). The calculation of the market value is done using the date fractions calculated in step 1, the calculated yields for each period and the growth rate.

The terminal value of the equity is calculated as

$$cf_M = \frac{MV(1+g)^{t_M}}{r-g}$$

And saved as

$$notcf = \{cf_M\}$$

For each period where a dividend is supposed to be paid, the market value is calculated as:

$$MV_{t_i} = MV(1+g)^{t_i}$$

Based on the market value at that period, the cash flow related to the dividend is calculated as:

$$cf_i = MV_{t_i} * c$$

$$divcf = \{cf_1, \dots, cf_k\}$$

## Property input and calibration

OSEM supports the asset class of properties. They represent properties wholly or partly owned by the company. The property's value is assumed to grow with a fixed growth rate. The property also gives the owner a dividend in the form of rent payments. Property shares are modelled as instruments that offer a constant cash flow in the form of a dividend (rent income) based on the property value at the time of payment. The property also has intrinsic value that increases over time at a constant growth. At the end of the modelling period, a selling value is calculated. The selling value represents the hypothetical sale of the property at the end of the modelling period.

For the POC, property asset class is modelled according to the same methodology as Equity.

## Cash input

Cash is an asset class that is supported by the OSEM model. It also serves as a liquidity management field, serving as an intermediate between different trading steps in the projection module. Only a single entry for cash is kept within the model.

$$cash_{MD} = CASH$$

Where:

## Saving of asset positions

After the existing portfolio was processed, the intermediate results are saved in the Intermediate folder. This is done to allow the model to be more modular. The preparation steps consists of creating a master table with every date that contains a cashflow and the rows contain every single asset. Part of the table contains descriptive columns such as the asset type, market value, NACE code, ID etc.

## Generation of future investment assets

Companies are expected to generate excess cash from operation, capital return or investment in the future. That cash is normally invested into the investments available at that time. Most of the assets that will be in the investment universe at that time do not exist yet at the Modelling Date and are therefore not present in the portfolio.

To keep the model realistic, OSEM needs a way to create future assets.

This section describes the methodology to generate the following asset classes:

- Government bonds
- Corporate bonds
- Equity shares
- Property shares

### Generation of future government bonds

1) DATES 1.1) Terminal dates

$$t_M^d = MD + MAT$$

1.2) Coupon dates

$$MD < t_1^d < \dots < t_k^d \leq MD + MAT$$

2) DATEFRAC w.r.t modelling date

SAME AS NORMAL BOND

3) CALIBRATION

$\min c$  s.t.

$$1 = \frac{c}{(1 + y(t_1))^{t_1}} + \dots + \frac{c}{(1 + y(t_k))^{t_k}} + \frac{1}{(1 + y(t_M))^{t_M}}$$

4) CASHFLOWS

SAME AS NORMAL BOND

## Generation of future corporate bonds

Corporate bonds are generated at every period for a set of key maturities obtained from the input file Maturity\_Generated. To avoid the explosion of new assets, the model only generates new bonds from sectors obtained from the input file. The sector is defined by its NACE code from the file NACE\_Generated located in the Input folder. The POC prototype only supports a risk neutral run therefore every sector would have the same profile. Therefore, sector list is ignored. The input files also contain frequency  $f$ .

In each modelling period  $MD_i$ , the key maturity of  $j$  years and a sector  $k$ , the following bond is generated:

Frequency of coupon dates is

$$f$$

Maturity date is:

$$M = MD_i + j$$

Issue date is:

$$I = MD_i$$

1) DATES 1.1) Terminal dates

$$t_M^d = MD + MAT$$

1.2) Coupon dates

$$MD < t_1^d < \dots < t_k^d \leq MD + MAT$$

2) DATEFRAC w.r.t modelling date

SAME AS NORMAL BOND

3) CALIBRATION

$\min c$  s.t.

$$1 = \frac{c}{(1 + y(t_1))^{t_1}} + \dots + \frac{c}{(1 + y(t_k))^{t_k}} + \frac{1}{(1 + y(t_M))^{t_M}}$$

4) CASHFLOWS

SAME AS NORMAL BOND

Calibration of future corporate bonds

For simplicity, the dividend yield of newly created bonds is calibrated so that when priced using the risk free curve, the market value is equal to the notional amount. For a risk neutral run, each sector would have the same properties.

## 1) Dates

The first step in the new bond generation process is to calculate dates at which the bond is expected to produce a cash flow.

We will denote these dates as:

$$I < t_1^d < \dots < t_n^d \leq M$$

## 2) Refactoring

Refactoring for fixed income assets consists of calculating the cash flow dates with relation to the modelling date. For each cash flow date, a time fraction is calculated with regards to the modelling date in years.

$$\begin{aligned} t_1 &= \frac{t_1^d - MD_i}{365.25} \\ t_2 &= \frac{t_2^d - MD_i}{365.25} \\ &\dots \\ t_n &= \frac{t_{j+k}^d - MD_i}{365.25} \end{aligned}$$

Note that the bond was generated such that all coupon cash flows are after the modelling step.

The date fractions are saved and can be viewed in the following object property:

$$datescf = \{t_1, t_2, \dots, t_{j+k}\}$$

Terminal date is also refactored if the asset has not matured before the modelling date.

$$t_M = \frac{t_M^d - MD}{365.25}$$

The date fraction is saved and can be viewed in the following object property:

$$datesnotional = \{t_M\}$$

## Calibration of the yield

This step uses a bisection algorithm to calculate the dividend yield such that the bond prices at par.

$$MV = \frac{N * c}{(1 + y(t_1))^{t_1}} + \dots + \frac{N * c}{(1 + y(t_k))^{t_k}} + \frac{N}{(1 + y_k)^{t_k^f}}$$

The objective of this optimisation is to calculate a dividend yield such that the notional amount is equal to the market value therefore  $MV = N$

$$N = \frac{N * c}{(1 + y(t_1))^{t_1}} + \dots + \frac{N * c}{(1 + y(t_k))^{t_k}} + \frac{N}{(1 + y(t_k))^{t_k}}$$

$$1 = \frac{c}{(1 + u(t_1))^{t_1}} + \dots + \frac{c}{(1 + u(t_M))^{t_M}} + \frac{1}{(1 + u(t_M))^{t_M}}$$

### Calculation of cash flows

The notional amount returned at the maturity of the bond is saved in the object property:

$$cf_M = N$$

$$notcf = \{cf_M\}$$

Once the dividend yield  $c$  is calculated, the cash flows can be calculated.

$$divcf = \{cf_1, \dots, cf_k\}$$

Where:

$$cf_i = N * c$$

### Generation of future equity shares

The equity asset class does not mature like fixed income instruments. In the first modelling period, a single equity is generated for each sector and calibrated. The company then trades in and out of these equities as modelling time progresses.

Note that if needed, additional equity assets can be generated. For example, if different dividend frequencies are needed.

When generating a new set of equity shares, the following data is read from the input files:

- Modelling date MD
- List of NACE sectors which can be considered representative enough of the future investment universe
- Frequency of dividend payments for each instrument  $f$
- Length of the modelling period
- The dividend yield of each equity share

The equity share is initialized using the year of the modelling date MD and the day/month set such to align with the desired payout behaviour. The end of the modelling period is set by adding the modelling date MD the length of the modelling period.

1) In an analogy to the treatment of existing equity shares, the dividend cash-flows and the terminal value date are calculated. 2) All cashflow dates are refactored with comparison to the modelling date 3) The market value is set equal to 1. 4) The book value is set to be equal to 1. 5) In the POC prototype, the growth rate is calibrated in a way that makes the discounted cashflows

of its dividend payments and the terminal value equal to the market value of 1. 6) Using the calculated growth rate, the cash-flows are simulated using the same formulas as the existing equity shares

1) DATES 1.1) Terminal date

$$t_M^d = END$$

1.2) Dividend dates

$$MD < t_1^d < \dots < t_k^d \leq END$$

2) DATEFRAC w.r.t modelling date

SAME AS NORMAL EQUITY

3) CALIBRATION

$\min g$  s.t.

$$\frac{1}{c} = \frac{(1+g)^{t_1}}{(1+y(t_1))^{t_1}} + \dots + \frac{(1+g)^{t_k}}{(1+y(t_k))^{t_k}} + \frac{1}{(1+y(t_M))^{t_M}} \frac{(1+g)^{t_M}}{r-g}$$

4) CASHFLOWS

SAME AS NORMAL EQUITY

## Generation of future property bonds

Future property shares are generated, calibrated, and saved in the same way as equity shares.

ToDo

## Saving of future asset positions

The future assets are saved in a master matrix similar to existing assets. Fixed income assets have to be generated each turn so they are kept in a separate master matrix compared to the other asset classes.

## Liability preparation

The liability section simulates different products and commitments that a company might have in their portfolio presently and in the future. There are two settings to treat the passive side of the company's balance sheet.

- 1) The company can upload a cash flow profile that determines the projected outflows at each modelling period
- 2) The company can do a full simulation of different kinds of products

The POC prototype will only support the option 2 and a liability cash flow profile will be uploaded. The cash flow profile consists of a vector of expenses at the end of each modelling period. This negative cash position will be considered at every step of trading.

## Term life

ToDo

## Health

ToDo

## P&C

ToDo

## Saving of liability positions

A master matrix is created containing the outflows and descriptive fields for every type of liability with a similar structure to the assets.

## Upload of externally calculated liabilities

A simplified approach of liability modelling, supported by OSEM is to provide a vector of outflows for each modelling period. Each outflow is denominated in the modelling currency. In this mode, there is only one cash flow per a modelling period. This assumes that all outflows are settled in a single day right before one modelling period is about to end and a new one is about to start.

For example, if the previous modelling period is 31/12/2012 and the new one is at 31/12/2023, then the assumption is that all outflows have to be paid at 31/12/2022. The liability cash flows will be denoted as:

$$cf^{liab} = \{cf_1, cf_2, \dots, cf_N\}$$

## Generation of future liabilities

ToDo

## Projection

The projection module simulates the inner workings of a company at the end of a fiscal year given its mandate and financial conditions. The steps consist of:

- 1) Calculate the cash accumulated between the current modelling period and the next one
- 2) Refactor the cashflows and dates from the assets to start from the new modelling period
- 3) Price the assets to account for the new market conditions
- 4) Refactor the cashflows and dates from the liabilities to start from the new modelling period

- 5) Declare reserves in the new modelling period
- 6) Calculate the return of the portfolio between the two modelling periods
- 7) Calculate the net position of the company
- 8) Generate new assets
- 9) Sell to cover the balance or buy new instruments
- 10) Calculate the income statement, solvency ratio and other views of interest
- 11) Close the period and save results

## 0) Preparation for the movement into the next modelling period at time 0

After the preparation of the model, assets and liabilities generated and curves projected, the objects are read into memory for the first time. The market value of the entire portfolio is calculated.

### 1) Calculate the cash accumulated between the current modelling period and the next one

The asset and liability position at the end of the previous period is read into memory. The cash position at the end of the previous period is read into memory.

Calculate the cash flow generated between the end of the previous modelling period and the end of the new modelling period by summing the cash flow matrix that falls between the modelling dates.

$$\begin{aligned} CASH_{i+1}^0 = & CASH_{i+1}^X + \sum_j \sum_k \sum_t cf_{ijk}^A I_{t_{ijk} < (M_{i+1} - M_i)} \\ & - \sum_j \sum_k \sum_t cf_{ijk}^L I_{t_{ijk} < (M_{i+1} - M_i)} \end{aligned}$$

Calculate the increase in the value of the portfolio by applying the growth rates and the correct time fraction.

### 2) Refactor the cashflows and dates from the assets to start from the new modelling period

The date fraction is moved by the length of the step forward in the modelling period for both final cash flows and coupon/dividend cash flows. Assets that mature are removed from the list.

Final cash flow date fractions:

$$DF_{i+1}^{0,A,N} = DF_i^{1,A,N} - (M_{i+1} - M_i)$$

$$DF_{i+1}^{1,A,N} = DF_{i+1}^{0,A,N} I_{t_{ijk} \leq 0}$$

Where the indicator function removes the row of the expired asset in the matrix.

Coupon date fractions:

$$DF_{i+1}^{0,A,C} = DF_i^{1,A,C} - (M_{i+1} - M_i)$$

$$DF_{i+1}^{1,A,C} = DF_{i+1}^{0,A,C} I_{t_{ijk} \leq 0}$$

Where the indicator function removes the row of the expired asset in the matrix.

Cash flow matrix for both final and coupon cash flows:

$$CF_{i+1}^{A,N} = CF_i^{A,N} I_{t_{ijk} \leq 0}$$

$$CF_{i+1}^{A,C} = CF_{i+1}^{A,C} I_{t_{ijk} \leq 0}$$

Where the indicator function removes the row of the expired asset in the matrix.

Note that this step calculates the value and timing of the asset portfolio at the end of the current modelling period. This can be used to price every single asset in the new economic reality implied by the new modelling date.

Note that the matrices can change size as assets that have matured are removed from the list.

### 3) Price the assets to account for the new market conditions

Calculate the market value of each asset in the new modelling date. This is done using the pricing function of each asset type using the refactored cash flow and timing matrices as well as the new yield curve:

$$MV_{i+1,j,k} = PRICE(DF_{i+1}^{A,N,1}, DF_{i+1}^{A,C,1}, CF_{i+1}^{A,N}, CF_{i+1}^{A,C}, y)$$

### 4) Refactor the cashflows and dates from the liabilities to start from the new modelling period

Liability cash flows go through a similar process as the assets where the cash flows and timings are adjusted for the new modelling period

$$DF_{i+1}^{L,N,0} = DF_i^{L,N,1} - (M_{i+1} - M_i)$$

$$DF_{i+1}^{L,N,1} = DF_{i+1}^{L,N,0} I_{t_{ijk} \leq 0}$$

$$CF_{i+1}^{L,N} = CF_i^{L,N} I_{t_{ijk} \leq 0}$$

## 5) Declare reserves in the new modelling period

$$RV_{i+1,j,k} = RESERVE(DF_{i+1}^{L,N,1}, CF_{i+1}^{L,N}, y, *)$$

This is done using the reserving functions for each liability modelling point using the refactored cash flow and timing matrices, demographic date, the new yield curve etc.

also, the reserve requirement of the entire portfolio can be calculated as:

$$RV_{i+1} = \sum_{PTF} \omega_{i+1,j,k} RV_{i+1,j,k}$$

An alternative approach is to carry forward the reserve at time 0:

$$RV_{i+1,j,k} = RV_{i,j,k} * (1 + y(MD_{i+1} - MD_i))^{MD_{i+1} - MD_i} + cf_{ijk}^{IN} - cf_{ijk}^{OUT}$$

## 6) Calculate the return of the portfolio between the two modelling periods

$$R_{i+1,j,k} = \frac{MV_{i+1,j,k} - MV_{i,j,k}}{MV_{i,j,k}}$$

the return of the entire portfolio is:

$$R_{i+1} = \sum_{PTF} \omega_{i+1,j,k} R_{i+1,j,k}$$

also, save the market value of the entire portfolio, used to compare the asset balance sheet with the liability side:

$$MV_{i+1} = \sum_{PTF} \omega_{i+1,j,k} MV_{i+1,j,k}$$

## 7) Calculate the net position of the company

Calculate the net balance between cash returns from investments and the obligations from the passive part of the balance sheet. This will trigger either the selling of assets to cover this period's liquidity need or the investment of excessive cash flows. By selecting the option XXX, the algorithm can also keep a liquid position equal to a fixed percentage of market value. This parameter can be found in XXX.

## 8) Generate new assets

Generate new assets that were issued at the new time period.

## 9) Sell to cover the balance or buy new instruments

If selling needed:

- Apply the selling of assets. Using one of the strategies shown in XXX
- Prioritize selling of existing assets
- Check if the new asset allocation respects the target asset allocation
- Sell/buy again if needed.

If buying necessary:

- Check if can be kept in cash
- Apply buying in assets as per the strategies shown in XXX
- Check if asset allocation respected
- Sell/buy again

## 10) Calculate the income statement, solvency ratio and other views of interest

ToDo

## 11) Close the period and save results

Prepare and save the final asset allocation