

Open-Source Asset Liability

Open-Source Asset Liability (OSAL) is a free modelling solution that allows financial companies to simulate their financial performance in the future.

OSAL consists of different modules such as:

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

Note that in the prototype POC not all modules will be implemented. The exact scope of the prototype is still to be defined.

Modelling parameters

OSM 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

- $MD \dots$ modelling date

Economic environment setup

The economic environment in which the model will run is being simulated in the first 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. If the EIOPA curve is used (POC prototype only supports the EIOPA curves), then the Smith&Wilson calibration can be taken directly from their website.

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

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

Where:

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

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

...

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

Import selected risk free curve

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.

From the input files, the following information is retrieved:

- Vector of maturities for the EIOPA yield curve
- Vector of yields for the maturities selected
- Coupon frequency parameter
- Last liquid point

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

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:

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

With the forward rates calculated, each future yield curve can be calculated as:

$$y^i(t - i) = \prod_i^t (1 + fw(t))^{\frac{1}{t-i}}$$

Where i is the projection year starting at time MD_i .

Calibrate the parameter α

The OSAL uses a basic bisection algorithm to calibrate the *alpha* parameter needed to produce the yield curve function.

For each modelling date, the discrete term structure estimated in the previous step is used to calibrate the Smith&Wilson function assuming the same ultimate forward rate and other parameters as provided by EIOPA.

The results are saved in the object property:

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

Note that to use a consistent calibration throughout the OSAL 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.

Calculate the calibration vector b

Using the projected discrete term-structure, other EIOPA parameters and the calibrated α calculated in the previous step, the vector b is calculated for each projection step and saved in the object property:

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

Saving the projected curves

The advantage of this calibration is two-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, the list of maturities and a calibration vector b are needed.

Asset input and calibration

Government bond input and calibration

ToDo

Corporate bond input and calibration

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

- f ... frequency of dates
- M ... maturity date
- I ... issue date
- cy ... coupon yield
- N ... notional amount
- MV ... market value at modelling date
- BV ... book value at modelling date

1) Dates

The basic building block of OSAL 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.

We will denote these dates as:

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

2) Refactoring

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 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.

The calculation of the date fractions is as follows:

$$t_1^f = \frac{t_j - MD}{365.5}$$

$$t_2^f = \frac{t_{j+1} - MD}{365.5}$$

$$\dots$$

$$t_n^f = \frac{t_{j+k} - MD}{365.5}$$

Where:

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

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

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

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

$$t_{end}^f = \frac{t_{end} - MD}{365.5}$$

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

$$datesnotional = \{t_{end}^f\}$$

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.

Coupon cash flow

$$divcf = \{d_1, \dots, d_k\}$$

Where:

$$d_i = N * cy$$

Notional cash flow

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

$$notcf = \{N\}$$

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:

$$MV = \frac{d_1}{(1 + y_1 + s + z)^{t_1^f}} + \cdots + \frac{d_k}{(1 + y_k + s + z)^{t_k^f}} + \frac{N}{(1 + y_k + s + z)^{t_k^f}}$$

Where

- $MV \dots$
- $d_i \dots$ is the i-th coupon
- $N \dots$ is the notional amount
- $y_i \dots$ is the annual yield for a period between the modelling date and time t_i
- $s \dots$ is the sector specific spread

Equity input and calibration

OSAL 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:

- $f \dots$ frequency of dividend payments
- $END \dots$ end of the modelling period
- $I \dots$ issue date
- $dy \dots$ dividend yield
- $MV \dots$ market value at modelling date
- $BV \dots$ book value at modelling date

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.

Dividend cash flow dates

$$I \leq t_1 < \cdots < t_n \leq END$$

Terminal value date

$$t_{end} = END$$

2) Refactoring

Like refactoring for fixed income assets, this step consists of calculating the cash flow dates that 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.

The calculation of the date fraction is:

$$t_1^f = \frac{t_j - MD}{365.5}$$

$$t_2^f = \frac{t_{j+1} - MD}{365.5}$$

...

$$t_k^f = \frac{t_{j+k} - MD}{365.5}$$

Where:

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

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:

$$datescf = \{t_1^f, t_2^f, \dots, t_k^f\}$$

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

$$t_{end}^f = \frac{t_{end} - MD}{365.5}$$

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

$$datesterminal = \{t_{end}^f\}$$

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:

$$MV = \frac{MV * (1 + g)^{t_1^f} dy}{(1 + y_1)^{t_1^f}} + \frac{MV * (1 + g)^{t_2^f} dy}{(1 + y_2)^{t_2^f}} + \dots + \frac{MV * (1 + g)^{t_m^f} dy}{(1 + y_m)^{t_m^f}} + \frac{1}{(1 + y_m)^{t_m^f}} \frac{MV}{r}$$

$$1 = \frac{(1 + g)^{t_1^f} dy}{(1 + y_1)^{t_1^f}} + \frac{(1 + g)^{t_2^f} dy}{(1 + y_2)^{t_2^f}} + \dots + \frac{(1 + g)^{t_m^f} dy}{(1 + y_m)^{t_m^f}} + \frac{1}{(1 + y_m)^{t_m^f}} \frac{dy}{r - g}$$

$$\frac{1}{dy} = \frac{(1 + g)^{t_1^f}}{(1 + y_1)^{t_1^f}} + \frac{(1 + g)^{t_2^f}}{(1 + y_2)^{t_2^f}} + \dots + \frac{(1 + g)^{t_m^f}}{(1 + y_m)^{t_m^f}} + \frac{1}{(1 + y_m)^{t_m^f}} \frac{1}{r - g}$$

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.

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

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

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

$$d_i = MV_{t_i^f} * dy$$

The terminal value of the equity is calculated as

$$TV = \frac{MV}{r-g}$$

$$divcf = \{d_1, \dots, d_k\}$$

Property input and calibration

OSAL 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.

Property has the following fields:

- f ... frequency of rent payments
- END ... end of the modelling period
- I ... date of contract signing (used to calculate the rent payment dates)
- dy ... rent yield
- MV ... market value at modelling date
- BV ... book value at modelling date

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.

Dividend cash flow dates

$$I \leq t_1 < \dots < t_n \leq END$$

Cash input

Cash is an asset class that is supported by the OSAL 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.

$$\text{cash} = \text{CASH}$$

Where:

- CASH is the input cash value loaded from the input files

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 OSAL 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, OSAL 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

ToDo

Generation of future corporate bonds

Corporate bonds are generated at every period for a set of key maturities obtained from the input file XXXX. 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. 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:

$$I = MD_i$$

Coupon yield is set empty and will be calibrated in a further step to be priced at par.

Notional amount is set to 100.

Market value is set to 100.

Book value is set to 100.

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:

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

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.

$$t_1^f = \frac{t_1 - MD_i}{365.5}$$

$$t_2^f = \frac{t_2 - MD_i}{365.5}$$

...

$$t_n^f = \frac{t_{j+k} - MD_i}{365.5}$$

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

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

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

$$t_{end}^f = \frac{t_{end} - MD}{365.5}$$

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

$$datesnotional = \{t_{end}^f\}$$

Calibration of the yield

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

$$MV = \frac{N * cy}{(1 + y_1)^{t_1^f}} + \dots + \frac{N * cy}{(1 + y_k)^{t_k^f}} + \frac{N}{(1 + y_k)^{t_k^f}}$$

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

$$\begin{aligned} N &= \frac{N * cy}{(1 + y_1)^{t_1^f}} + \dots + \frac{N * cy}{(1 + y_k)^{t_k^f}} + \frac{N}{(1 + y_k)^{t_k^f}} \\ 1 &= \frac{cy}{(1 + y_1)^{cy}} + \dots + \frac{cy}{(1 + y_k)^{t_k^f}} + \frac{1}{(1 + y_k)^{t_k^f}} \end{aligned}$$

Calculation of cash flows

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

$$divcf = \{d_1, \dots, d_k\}$$

Where:

$$d_i = N * cy$$

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

$$notcf = \{N\}$$

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.

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 2. option 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.

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 and the growth/decline of the portfolio market value
- 2) Calculate the return of the portfolio by comparing the market value between the periods
- 3) Calculate the liability payment for the period between the modelling periods
- 4) Calculate net position
- 5) Sell to cover the balance or buy new instruments

0) Preparation for the movement into the next modelling period

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. The market value of the entire portfolio is calculated.

1) Calculate cash accrued between modelling periods and the asset growth rate

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.

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

2) Calculate the return of the portfolio by comparing the market value between the periods

Calculate the value of the portfolio at the end of the current modelling period.

3) Calculate the liability payment for the period between the modelling periods

Calculate the liability cash flows generated between the end of the previous modelling period to the end of the new modelling period.

4) Calculate net position

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.

5) Sell/buy to either 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

6) Close the period and save results

Prepare and save the final asset allocation

In []: