

# Implementation of the Sensitivities-Based Method (FRTB) with AAD: application to an equity portfolio

Michele Ercole

University of Verona

3 April 2025

# Table of Contents

1 Introduction

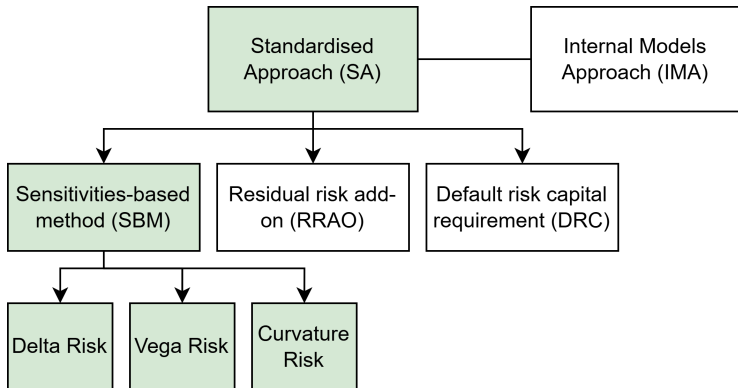
2 Sensitivities-Based Method Workflow

3 Conclusions

# Context and Objectives

- **Context:** within the sensitivities-based method, the FRTB regulation requires the use of finite differences ( $\rightarrow$  derivative approximation) to compute delta and vega.
- **Objective:** to build a Java project to replicate the sensitivities-based method by integrating Automatic Adjoint Differentiation ( $\rightarrow$  exact derivatives) for an equity portfolio.

# FRTB – Components



# Example Portfolio

## Composition:

- 2 stocks
- 2 European options
- 2 Bermudan options

Portfolio	DealNumber	AssetType	OptionStyle	Underlying	OptionType	Currency	Amount	Volatility	Strikes	UnderlyingPrice	Maturity	ExerciseDates	RiskFreeRate
EQ_PORT	1	Stock		WMT		USD	1			90			
EQ_PORT	2	Stock		JPM		USD	1			100			
EQ_PORT	3	Option	European	AAPL		1 USD	1	0.32	100	100	1.5		0.02
EQ_PORT	4	Option	European	AMZN		1 USD	1	0.25	110	90	2		0.02
EQ_PORT	5	Option	Bermudan	MSFT		1 USD	1	0.3	100;120	100	2	1.0;2.0	0.02
EQ_PORT	6	Option	Bermudan	NFLX		1 USD	1	0.32	100;90	100	2	1.5;2.0	0.02

Input table: excerpt from `sample_trades.csv`

The screenshot displays a UML class diagram in the IntelliJ IDEA IDE. The diagram illustrates the relationships between several classes in a financial modeling project:

- SensitivityAggregator**: Contains methods for aggregating sensitivities, such as `aggregateSensitivities`, `getSensitivity`, and `calculateSensitivity`. It has a dependency on `VolatilityInterpolator`.
- VolatilityInterpolator**: Contains methods for interpolating volatility, such as `interpolate` and `calculateVolatility`. It has a dependency on `CurvatureRiskCalculator`.
- CurvatureRiskCalculator**: Contains methods for calculating curvature risk, such as `calculateCurvatureRisk` and `getCurvatureRisk`. It has a dependency on `CsvParser`.
- CsvParser**: Contains methods for parsing CSV data, such as `parseCsv` and `parseCsvLines`. It has a dependency on `AACPricer`.
- AACPricer**: Contains methods for pricing AAC, such as `priceAAC` and `calculateAAC`. It has a dependency on `CurvatureRiskCalculator`.
- Main**: The entry point of the application, containing a `main` method.

The diagram uses standard UML notation: solid lines for associations, dashed lines for dependencies, and solid lines with open arrowheads for generalizations. The classes are represented by rectangles with a header section containing the class name and a body section containing the methods.

# Sensitivities-Based Method Workflow

- 1 Reading input files
  - a) Bucket assignment
  - b) Identification of risk factors
- 2 Calculation of sensitivities with respect to risk factors
  - a) Net Sensitivities
  - b) Weighted Sensitivities
- 3 Aggregation
  - a) Intra-bucket → medium, high, low correlation
  - b) Inter-bucket → medium, high, low correlation
- 4 Calculation of the final requirement

# 1. a) Bucket Assignment

**Input:** sample\_trades.csv and underlying\_bucket\_mapping.csv

**Classes:** CSVParser, Trade, Main

Each record in sample\_trades.csv is assigned a bucket based on the underlying asset. Example:

Trade: 4 → Underlying: AMZN → Bucket: 5

Underlying	Bucket	Capitalizzazione	Economia	Settore
AAPL	8	Grande	Economia avanzata	Tecnologia
AMZN	5	Grande	Economia avanzata	Beni di consumo e servizi
BABA	1	Grande	Economia emergente	Beni di consumo e servizi

Input table: excerpt from underlying\_bucket\_mapping.csv



# 1. b) Risk Factor Identification

**Input:** sample\_trades.csv

**Classes:** CSVParser, VolatilityInterpolator, Trade, Main

- **For delta and curvature** → Underlying spot price
- **For vega** → Implied volatility mapped to regulatory tenors (0.5Y, 1Y, 3Y, 5Y, 10Y)

DealNumber	Underlying	Bucket	RiskFactorDelta	RiskFactorVega
1	WMT	5	Spot-WMT	null
2	JPM	8	Spot-JPM	null
3	AAPL	8	Spot-AAPL	ImpliedVol-AAPL-1.0Y
4	AMZN	5	Spot-AMZN	ImpliedVol-AMZN-3.0Y
5	MSFT	8	Spot-MSFT	ImpliedVol-MSFT-3.0Y
6	NFLX	5	Spot-NFLX	ImpliedVol-NFLX-3.0Y

# Focus: Automatic Differentiation (AD)

Automatic differentiation  $\leftrightarrow$  chain rule:

$$\frac{\partial y}{\partial x_m} = \sum_l \frac{\partial y}{\partial x_l} \cdot \frac{\partial x_l}{\partial x_m},$$

## Forward Mode

$$\frac{\partial x_m}{\partial x_i} = \sum_{l=1}^{k(m)} \frac{\partial x_{i_l^{(m)}}}{\partial x_i} \cdot \frac{\partial f_m}{\partial x_{i_l^{(m)}}}(x_{i_1^{(m)}}, \dots, x_{i_{k(l)}^{(m)}})$$

Forward mode applies the chain rule starting from the inputs: for each node  $x_m$  (from  $n$  to  $N$ ) and for each independent variable  $x_i$ ,  $\frac{\partial x_m}{\partial x_i}$  is computed by propagating information forward along the computational graph.

## Backward Mode (AAD)

$$\frac{\partial y}{\partial x_m} = \sum_{l \in I} \frac{\partial y}{\partial x_l} \cdot \frac{\partial f_l}{\partial x_m}(x_{i_1^{(l)}}, \dots, x_{i_{k(l)}^{(l)}})$$

Backward mode applies the chain rule in reverse: for each node  $x_m$ , the derivatives of all dependent nodes are aggregated, propagating  $\frac{\partial y}{\partial x_m}$  backward. With the initial condition:  $\frac{\partial y}{\partial y} = 1$

## 2. Sensitivity Calculation with Respect to Risk Factors

**Classes:** AADPricer, CurvatureRiskCalculator, Main

### Delta

$$s_{i,k}^{FD} = \frac{V_i(1.01 \cdot EQ_k) - V_i(EQ_k)}{0.01}$$

↓

$$s_{i,k}^{AAD} = \frac{\partial V_i}{\partial EQ_k} \cdot EQ_k$$

### Vega

$$s_{i,k}^{FD} = \frac{V_i(1.01 \cdot \sigma_k, x, y) - V_i(\sigma_k, x, y)}{0.01}$$

↓

$$s_{i,k}^{AAD} = \frac{\partial V_i}{\partial \sigma_k} \cdot \sigma_k$$

### Curvature

$$CVR_k^{\pm} = - \sum_i \left\{ V_i \left( x_k^{RW^{Curvature\pm}} \right) - V_i(x_k) \mp RW_k^{Curvature} \cdot s_{ik} \right\}$$

## 2. Sensitivity Calculation Output

Below is an excerpt of the **output** related to the calculation:

```
=== SENSITIVITY CALCULATION =====
```

```
Processing trade: 4
```

```
Underlying AMZN | Bucket: 5 | AssetType: Option | OptionStyle: European
```

```
Value AAD: 7.301334 | Value FD: 7.301334 | Value Analytic: 7.261207
```

```
Delta AAD: 35.392896 | Delta FD: 35.857758 | Analytic Delta: 35.157002
```

```
Vega AAD: 12.281135 | Vega FD: 12.281502 | Analytic Vega: 12.214215
```

```
Time AAD: 594.264 ms | Time FD: 897.606 ms | Time Analytic: 0.000 ms
```

```
=== CURVATURE RISK CALCULATION =====
```

```
RiskFactor: Spot-AAPL |Bucket: 8 | CVR +: -8.885463 | CVR -: -14.019368
```

```
RiskFactor: Spot-NFLX |Bucket: 5 | CVR +: -2.724541 | CVR -: -4.186483
```

```
RiskFactor: Spot-MSFT |Bucket: 8 | CVR +: -9.869637 | CVR -: -15.638343
```

```
RiskFactor: Spot-AMZN |Bucket: 5 | CVR +: -4.085770 | CVR -: -4.227744
```

## 2. a) Net Sensitivities

**Classes:** Main, SensitivityAggregator

**For delta and vega:** Net Sensitivity is computed for each risk factor:

$$s_k = \sum_i s_{i,k}$$

=== NET SENSITIVITIES DELTA =====

```
RiskFactor: Spot-JPM | Bucket: 8 | Net Delta: 100.000000
RiskFactor: Spot-AAPL | Bucket: 8 | Net Delta: 60.985109
RiskFactor: Spot-NFLX | Bucket: 5 | Net Delta: 71.012533
RiskFactor: Spot-MSFT | Bucket: 8 | Net Delta: 59.138392
RiskFactor: Spot-AMZN | Bucket: 5 | Net Delta: 35.392896
RiskFactor: Spot-WMT | Bucket: 5 | Net Delta: 90.000000
```

=== NET SENSITIVITIES VEGA =====

```
RiskFactor: ImpliedVol-AAPL-1.0Y | Bucket: 8 | Net Vega: 15.226626
RiskFactor: ImpliedVol-MSFT-3.0Y | Bucket: 8 | Net Vega: 14.659538
RiskFactor: ImpliedVol-AMZN-3.0Y | Bucket: 5 | Net Vega: 12.281135
RiskFactor: ImpliedVol-NFLX-3.0Y | Bucket: 5 | Net Vega: 15.626644
```

## 2. b) Weighted Sensitivities

**Classes:** Main, SensitivityAggregator

**For delta and vega:** Weighted Sensitivity is computed for each risk factor, using regulation-defined weights:

$$WS_k = RW_k \cdot s_k$$

=== WEIGHTED SENSITIVITIES DELTA =====

RiskFactor: Spot-JPM	Bucket: 8	Weighted Delta: 50.000000
RiskFactor: Spot-AAPL	Bucket: 8	Weighted Delta: 30.492555
RiskFactor: Spot-NFLX	Bucket: 5	Weighted Delta: 21.303760
RiskFactor: Spot-MSFT	Bucket: 8	Weighted Delta: 29.569196
RiskFactor: Spot-AMZN	Bucket: 5	Weighted Delta: 10.617869
RiskFactor: Spot-WMT	Bucket: 5	Weighted Delta: 27.000000

=== WEIGHTED SENSITIVITIES VEGA =====

RiskFactor: ImpliedVol-AAPL-1.0Y	Bucket: 8	Weighted Vega: 11.843270
RiskFactor: ImpliedVol-MSFT-3.0Y	Bucket: 8	Weighted Vega: 11.402189
RiskFactor: ImpliedVol-AMZN-3.0Y	Bucket: 5	Weighted Vega: 9.552267
RiskFactor: ImpliedVol-NFLX-3.0Y	Bucket: 5	Weighted Vega: 12.154403

### 3. a) Intra-Bucket Aggregation

**Classes:** Main, SensitivityAggregator

- **For delta and vega** →

$$K_b = \sqrt{\max(0, \sum_k WS_k^2 + \sum_k \sum_{l \neq k} (\rho_{kl} WS_k WS_l))}$$

- **For curvature** →  $K_b = \max(K_b^+, K_b^-)$ , where:

$$K_b^\pm = \sqrt{\max\left(0, \sum_k \max(CVR_k^\pm, 0)^2 + \sum_k \sum_{l \neq k} \rho_{kl} CVR_k^\pm CVR_l^\pm \psi(CVR_k^\pm, CVR_l^\pm)\right)}$$

```

=== INTRA-BUCKET AGGREGATION =====
--- Delta -----
Bucket: 5 | Delta (M: 39.593061, H: 40.442746, L: 38.724736)
Bucket: 8 | Delta (M: 72.665777, H: 74.326043, L: 70.966680)
--- Vega -----
Bucket: 5 | Vega (M: 16.370731, H: 16.590878, L: 16.147584)
Bucket: 8 | Vega (M: 17.433313, H: 17.672925, L: 17.190362)
--- Curvature -----
Bucket: 5 | Curvature (M: 0.000000, H: 0.000000, L: 0.000000)
Bucket: 8 | Curvature (M: 0.000000, H: 0.000000, L: 0.000000)

```

### 3. b) Inter-Bucket Aggregation

**Classes:** Main, SensitivityAggregator

■ **For delta and vega**  $\rightarrow K_{\text{class}} = \sqrt{\sum_b K_b^2 + \sum_b \sum_{c \neq b} \gamma_{bc} S_b S_c}$

■ **For curvature**  $\rightarrow k_{\text{class}} = \sqrt{\max\left(0, \sum_b K_b^2 + \sum_b \sum_{c \neq b} \gamma_{bc} S_b S_c \psi(S_b, S_c)\right)}$

```
=== INTER-BUCKET AGGREGATION =====  
--- Delta -----  
(M: 90.181645, H: 91.520037, L: 88.823088)  
--- Vega -----  
(M: 25.754159, H: 26.118930, L: 25.384147)  
--- Curvature -----  
(M: 0.000000, H: 0.000000, L: 0.000000)
```



## 4. Final Requirement Calculation

**Classes:** Main, SensitivityAggregator

The final requirement is:

$$K_{final} = \max\{K_{medium}, K_{high}, K_{low}\}$$

=== FINAL RESULTS =====

Final Capital Requirement Delta: 91.520037

Final Capital Requirement Vega: 26.118930

Final Capital Requirement Curvature: 0.000000

# Conclusions

The implementation of the project described above led to the following conclusions:

- AAD computation times are comparable to FD (the efficiency of AAD becomes more evident as the number of inputs increases).
- AAD provides greater accuracy in derivative calculations (taking into account the error from Monte Carlo simulations).