

Pricing par EDP

dans le modèle de Black-Scholes

Y. Falaki — A. Feehan

Introduction

Qu'est ce qu'un produit dérivé ?

Un **produit dérivé** est un objet financier dont **la valeur dépend de sous-jacents**

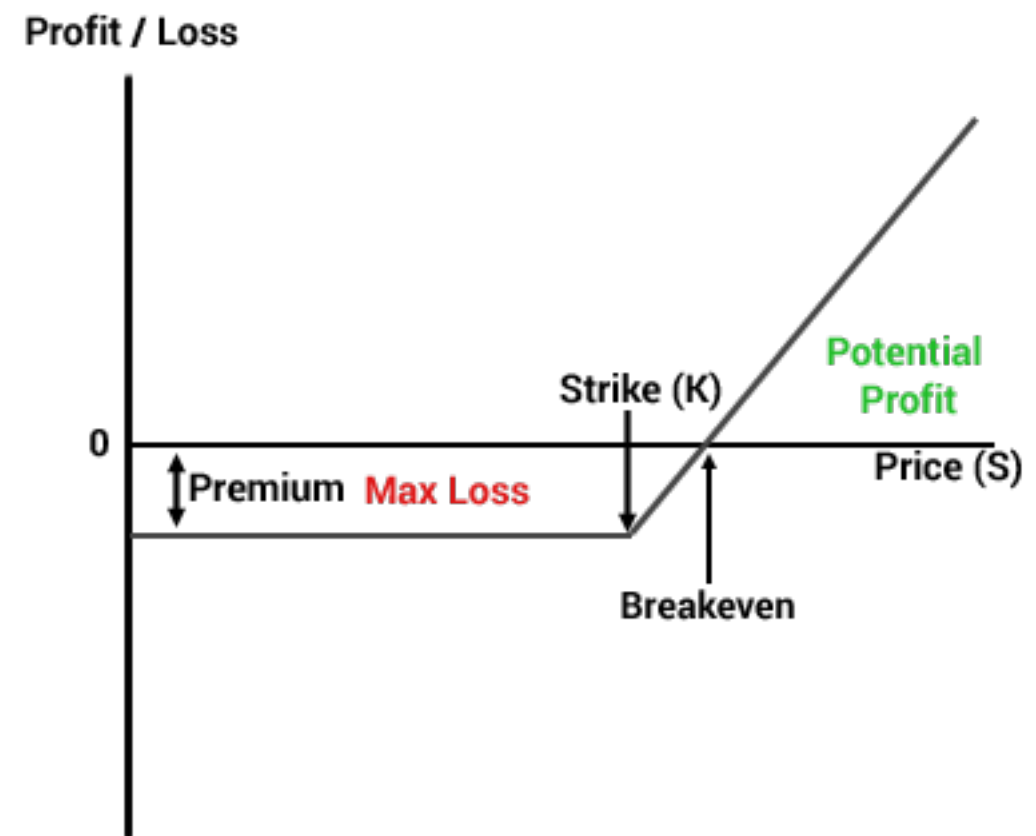


Introduction

Exemples de dérivés :

- **Call** : Droit d'**acheter**
- **Put** : Droit de **vendre**

Mais la **valeur** de ces dérivés dépend de **plusieurs paramètres** dont par exemple le risque lié au sous jacent.



Introduction



Pricer un dérivé dans le modèle de Black-Scholes en se basant sur les paramètres du modèle :

- la maturité T ;
- le taux d'intérêt sans risque r ;
- le risque σ (Calculé à partir des données historiques).

Objectifs :

1. **Valoriser** un portefeuille
2. Etablir une **stratégie** de trading

Que vaut un dérivé quelconque ?

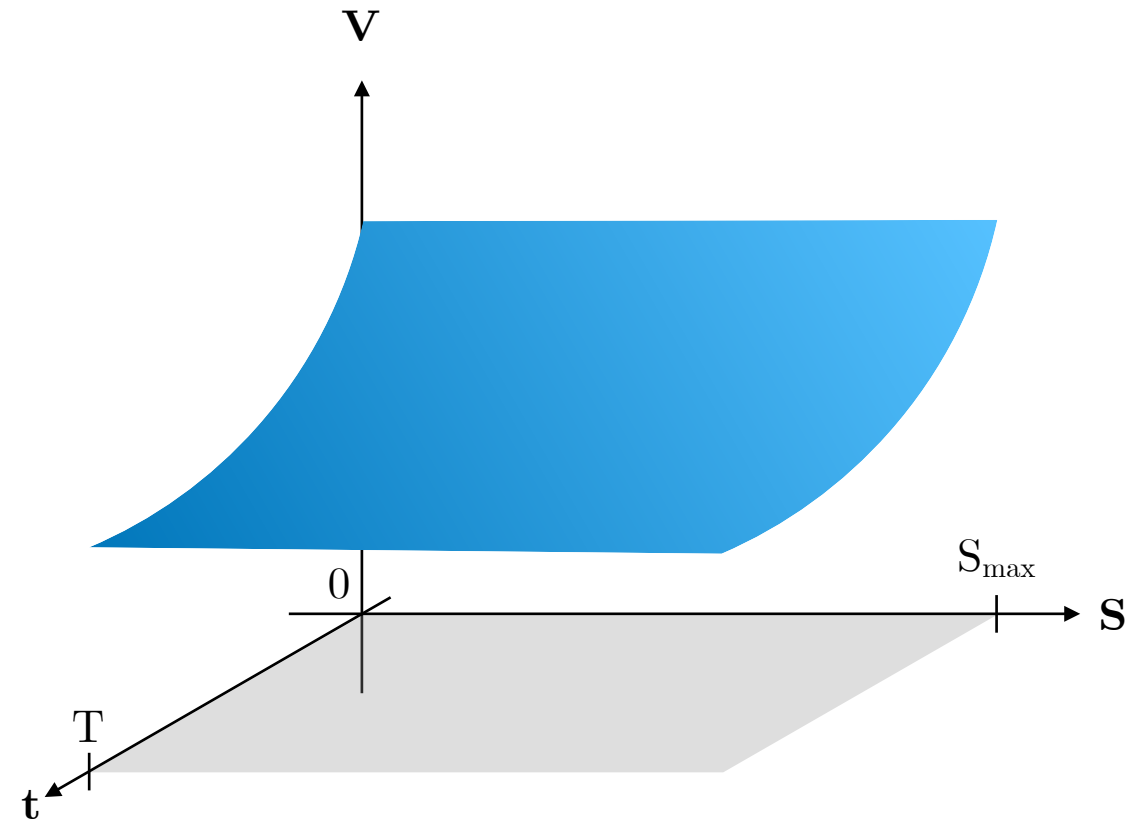
Sommaire

- I. **Modèle** de Black-Scholes
- II. **Implémentation** en C++
- III. **Bilan**, problèmes rencontrés

I. Le modèle de Black-Scholes

L'équation de B-S

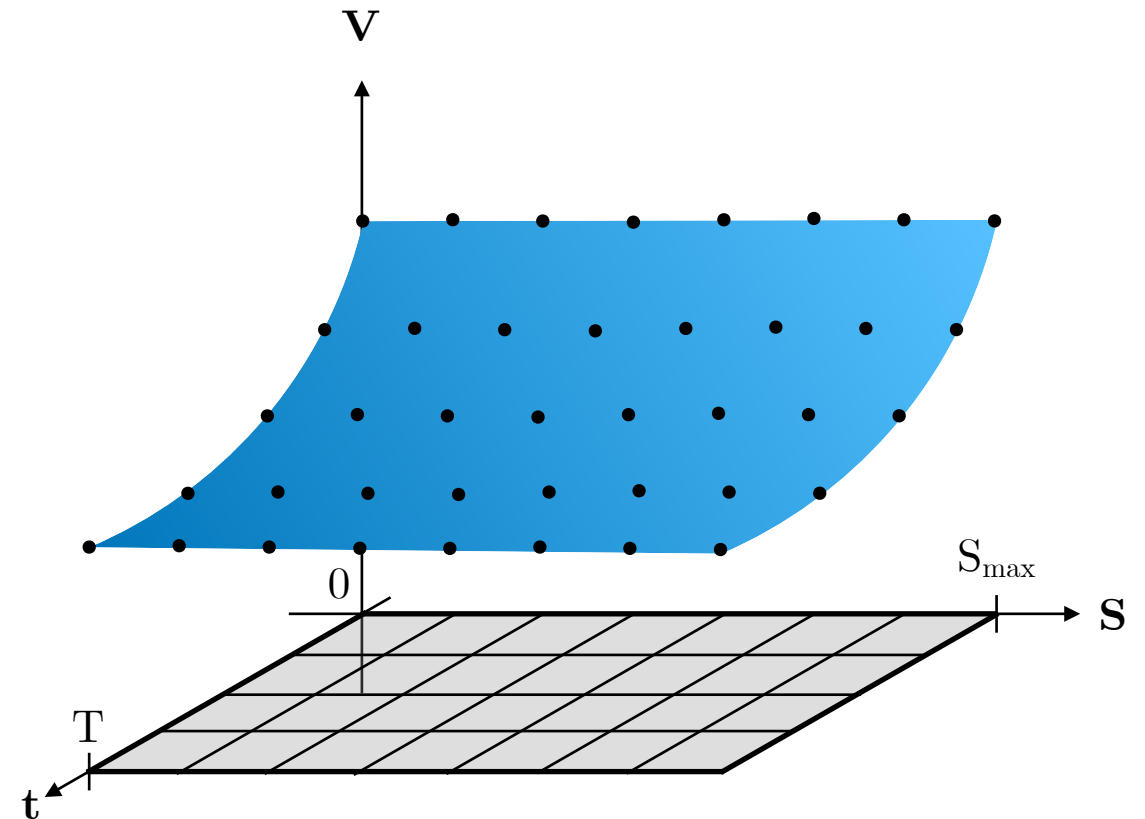
$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} = rV$$



I. Le modèle de Black-Scholes

L'équation de B-S

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} = rV$$

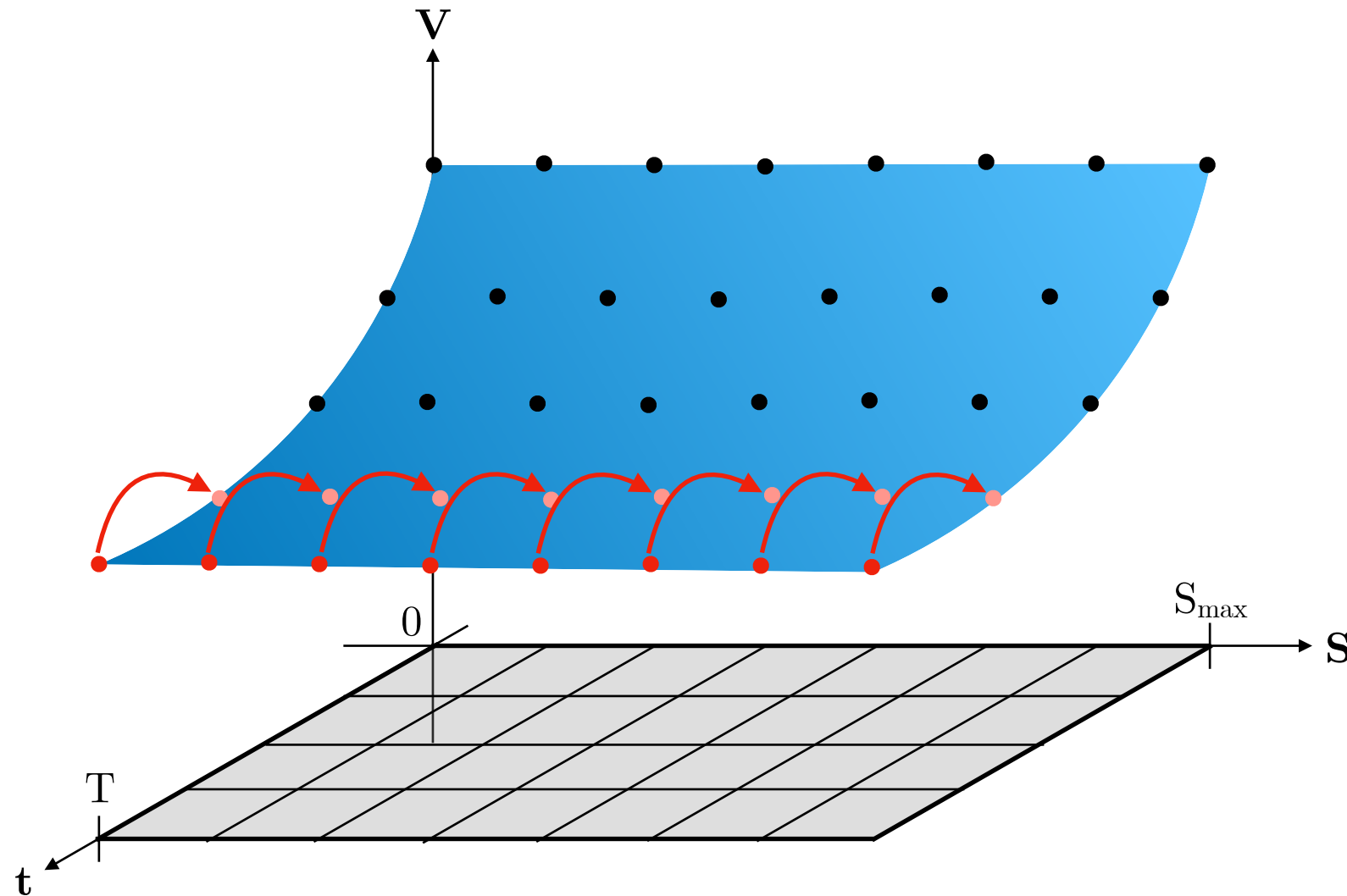


$$\frac{V_{n+1,m} - V_{n,m}}{\Delta t} + \frac{1}{2}\sigma^2 m^2 \Delta S^2 \frac{V_{n,m+1} - 2V_{n,m} + V_{n,m-1}}{\Delta S^2} + rm \Delta S \frac{V_{n,m+1} - V_{n,m-1}}{2\Delta S} = rV_{n,m}$$

Euler explicite

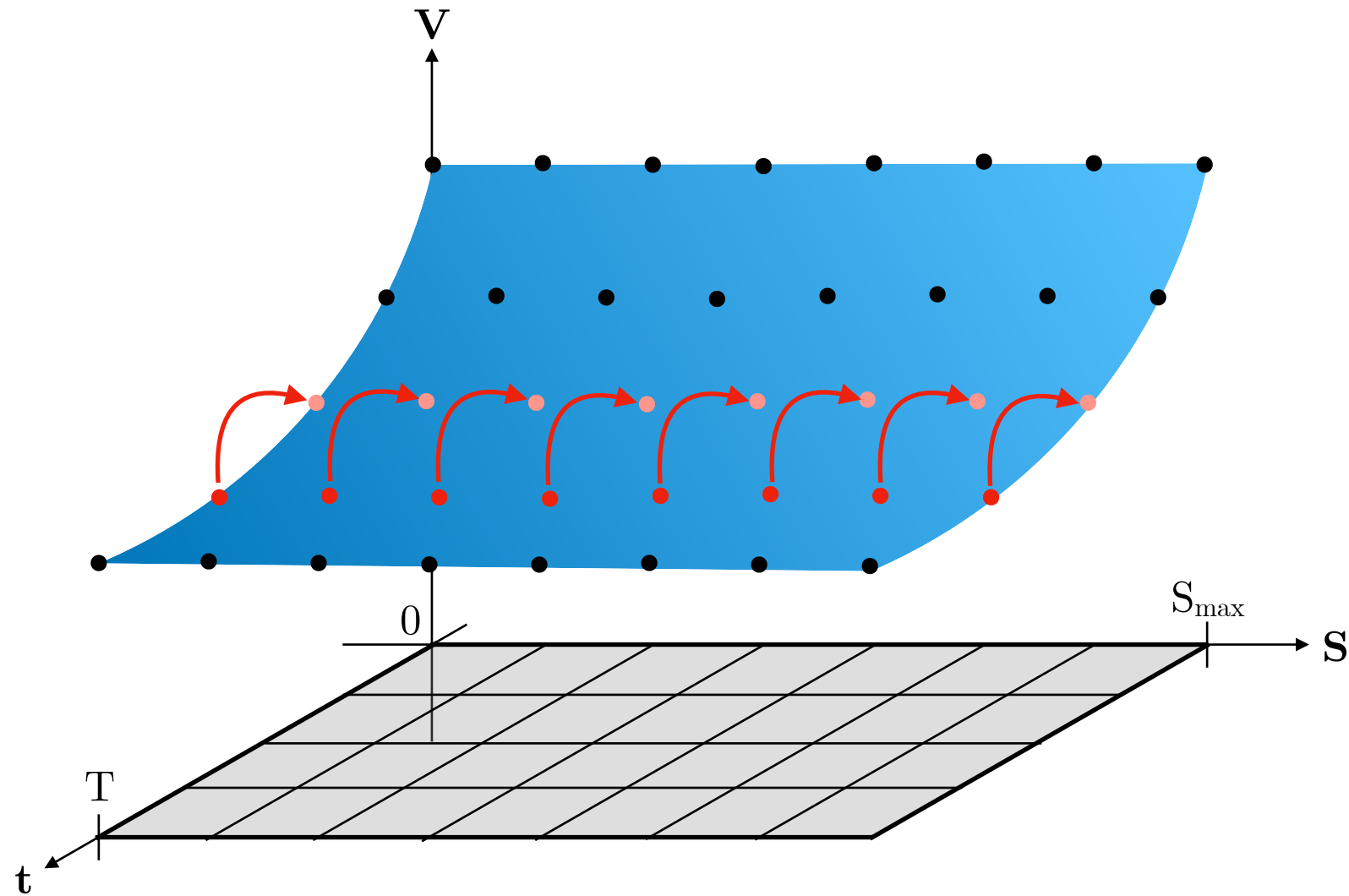
Crank-Nicolson

Différences finies



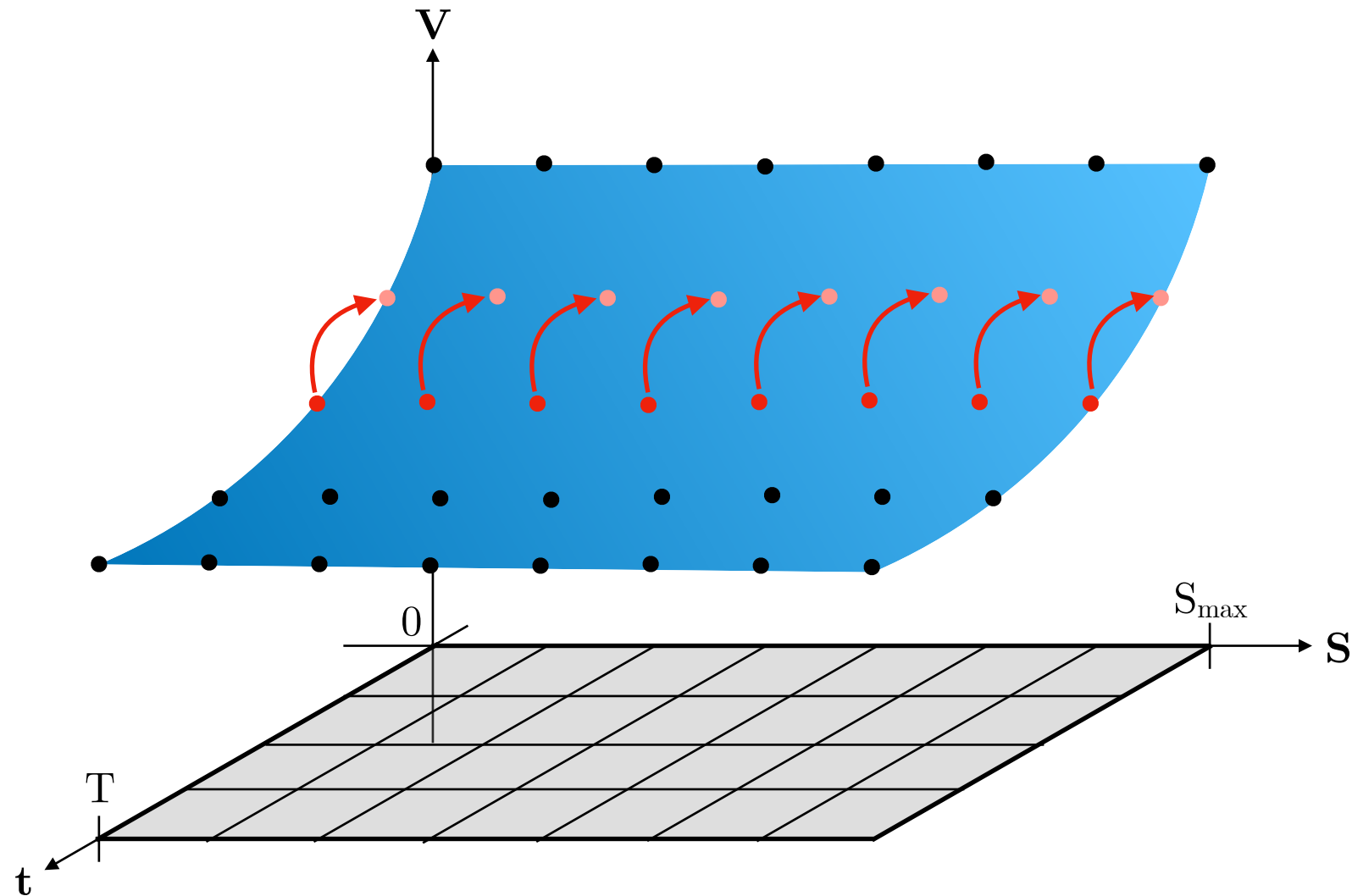
$$\forall n \in \{0, \dots, N\}, \quad \overrightarrow{V}_{n+1} = A \times \overrightarrow{V}_n$$

Différences finies



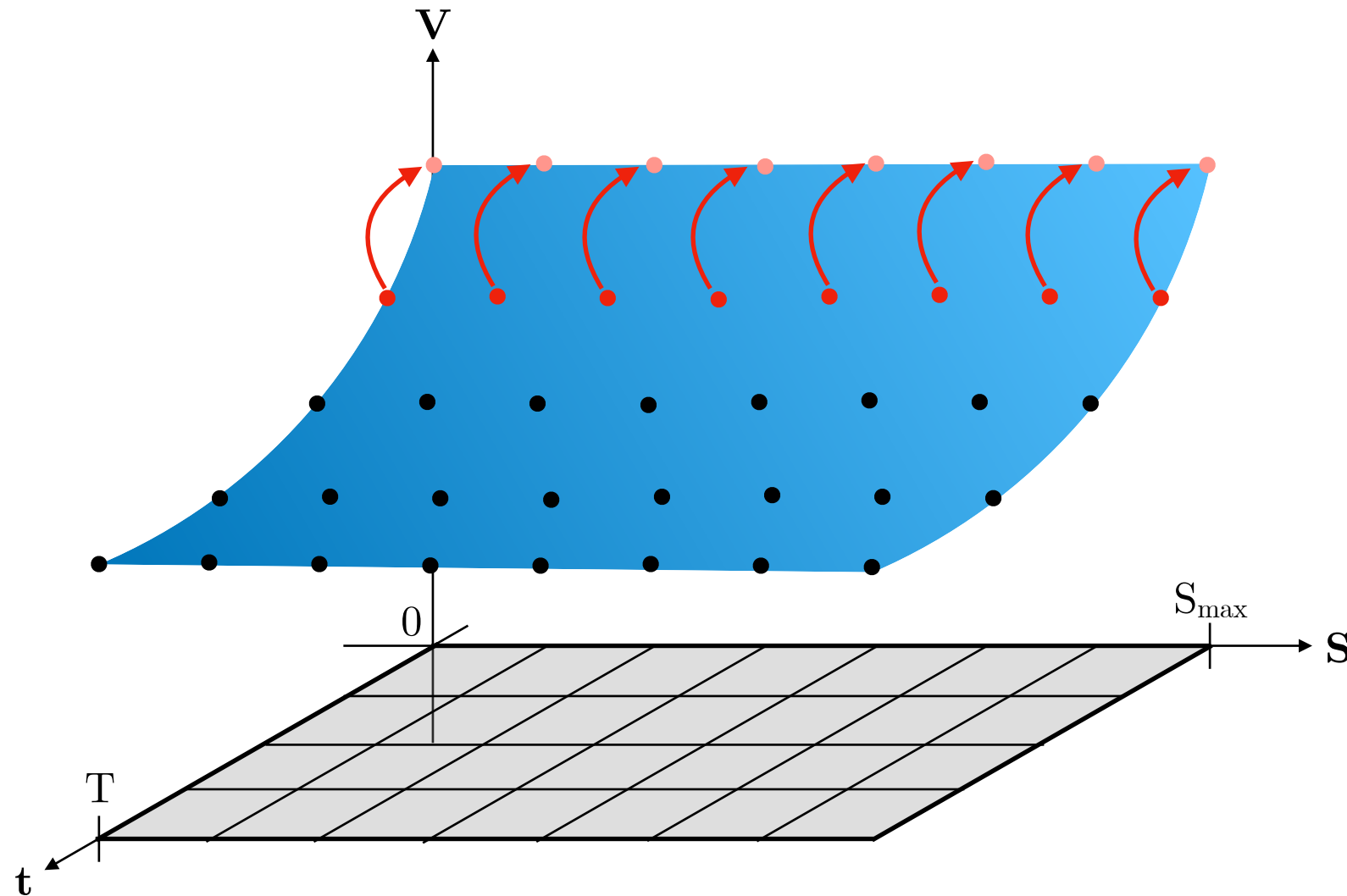
$$\forall n \in \{0, \dots, N\}, \quad \overrightarrow{V_{n+1}} = A \times \overrightarrow{V_n}$$

Différences finies



$$\forall n \in \{0, \dots, N\}, \quad \overrightarrow{V_{n+1}} = A \times \overrightarrow{V_n}$$

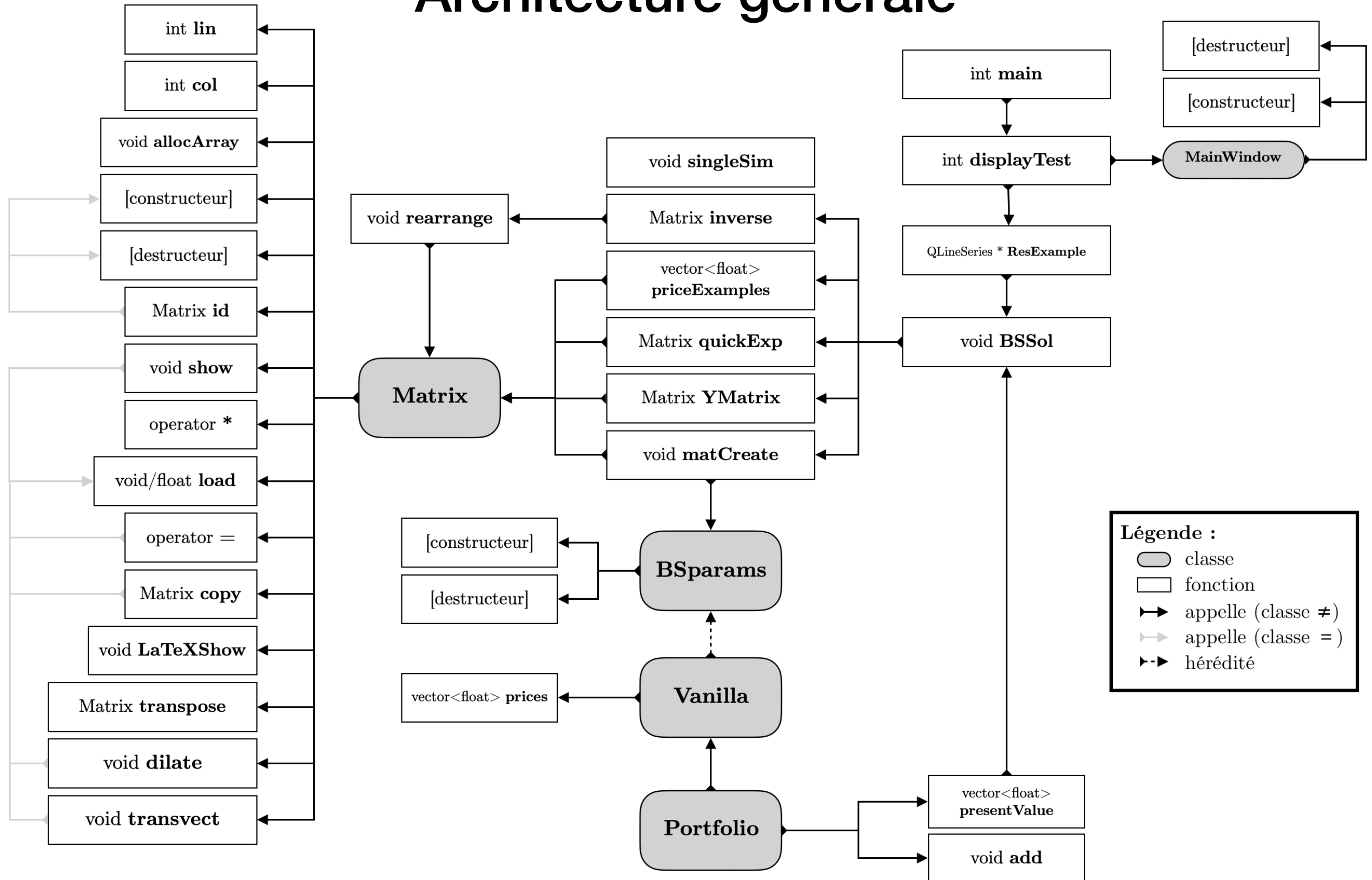
Différences finies



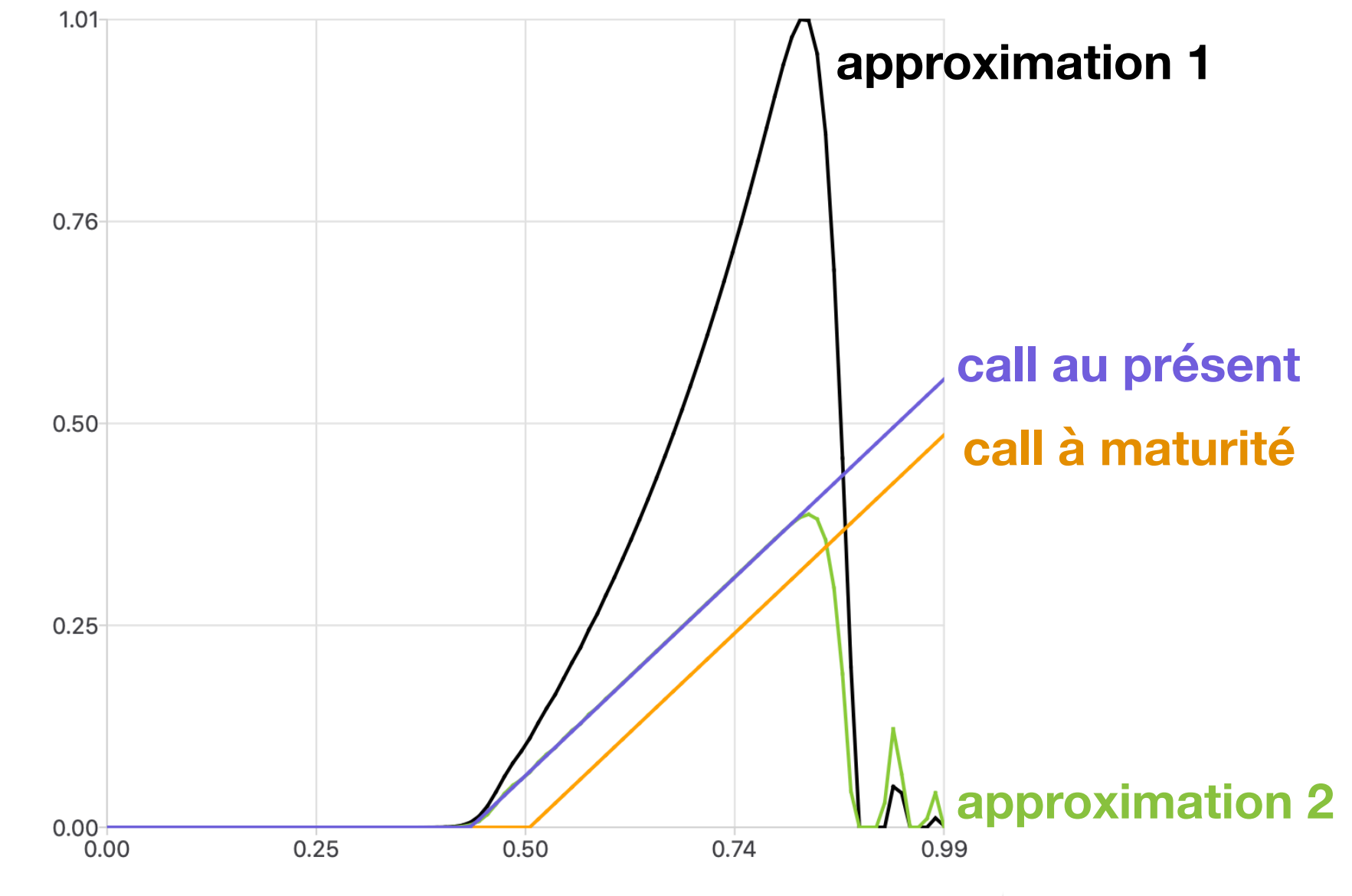
$$\forall n \in \{0, \dots, N\}, \quad \overrightarrow{V_{n+1}} = A \times \overrightarrow{V_n}$$

II. L'implémentation en C++

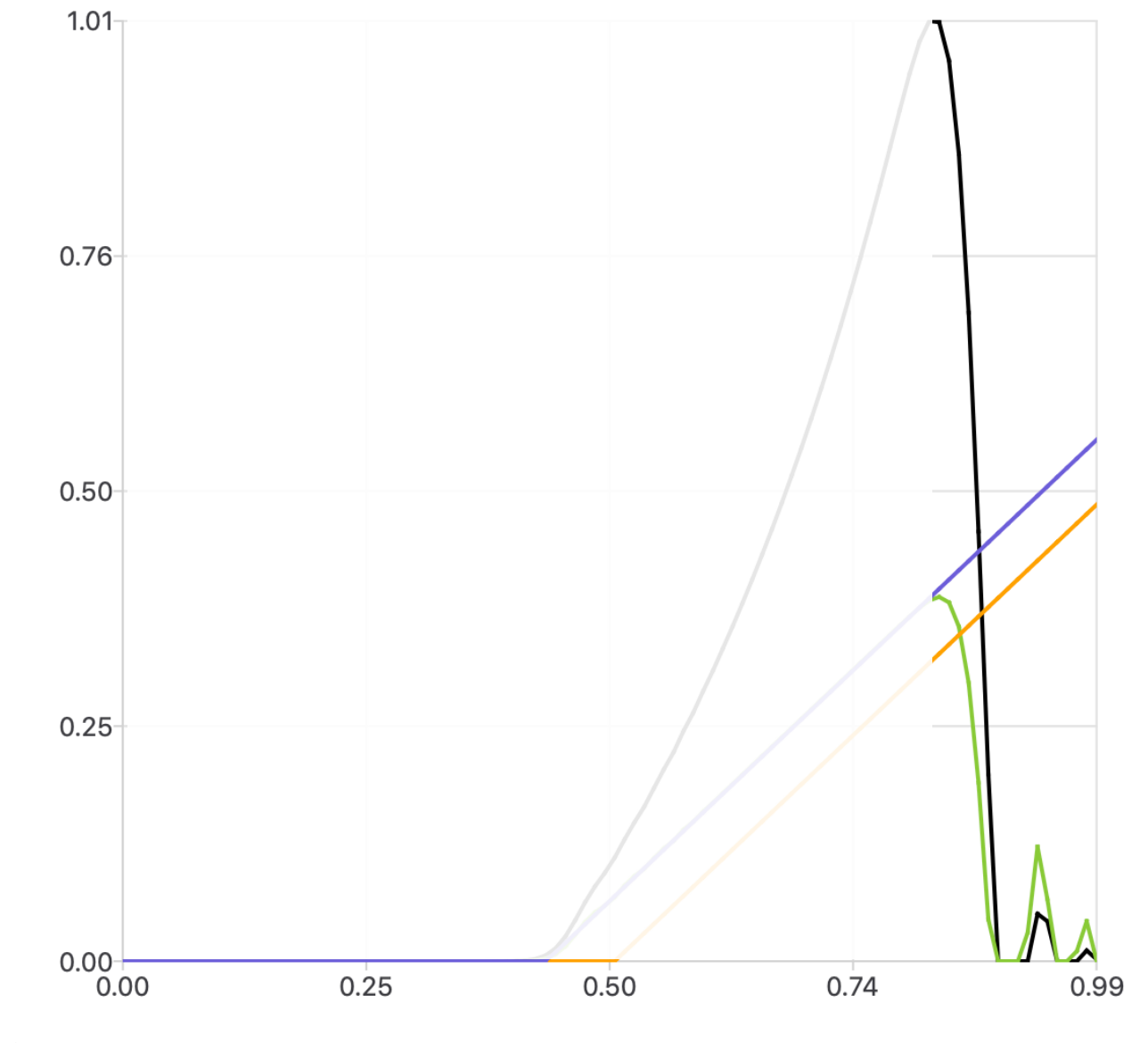
Architecture générale



Affichage avec Qt



Affichage avec Qt

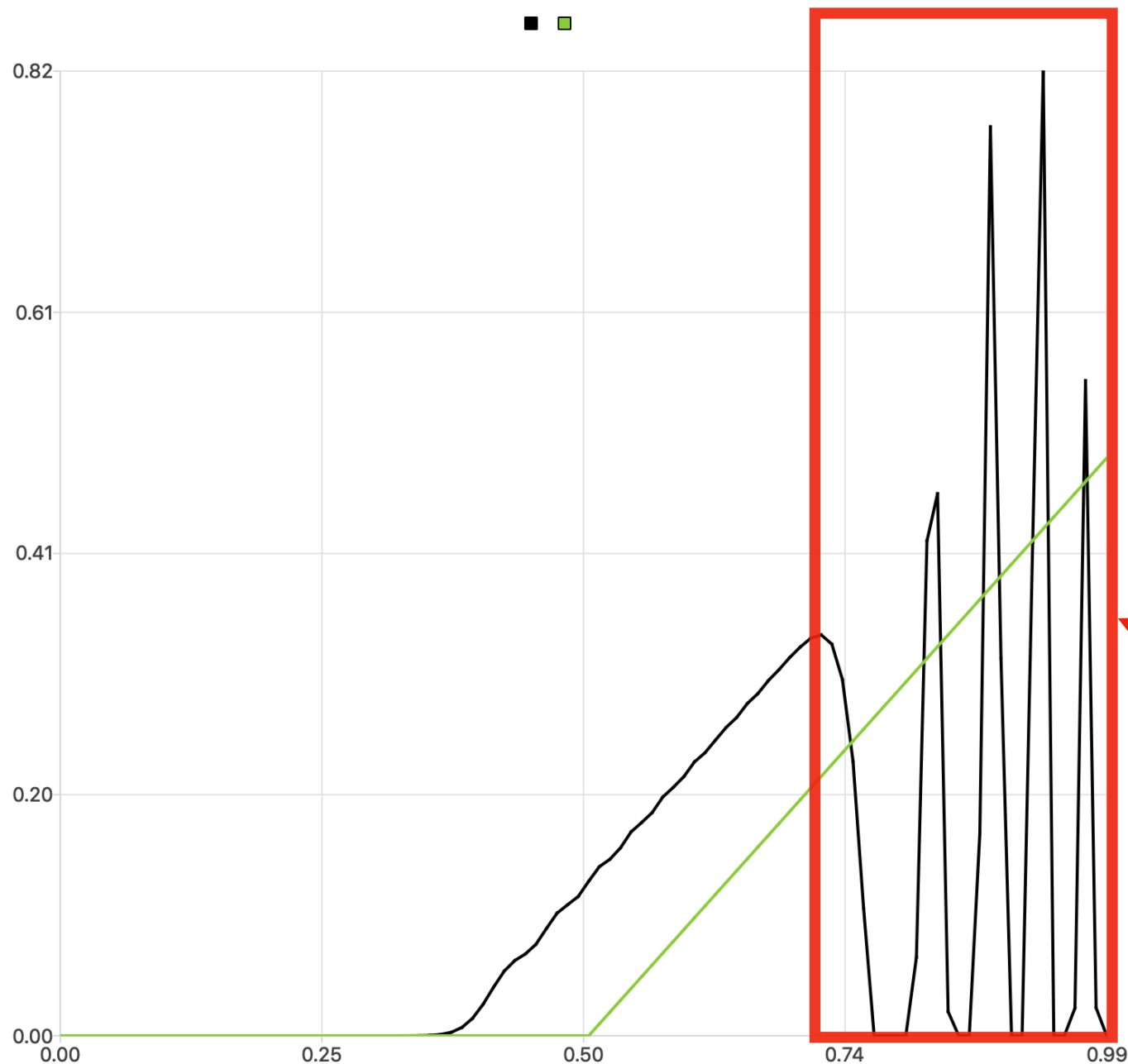


III. Bilan, problèmes rencontrés

Voies d'amélioration

1. Inversion matricielle plus stable

Value of derivative at two instants, as a function of underlying value



erreur dans
l'approximation
de l'inverse

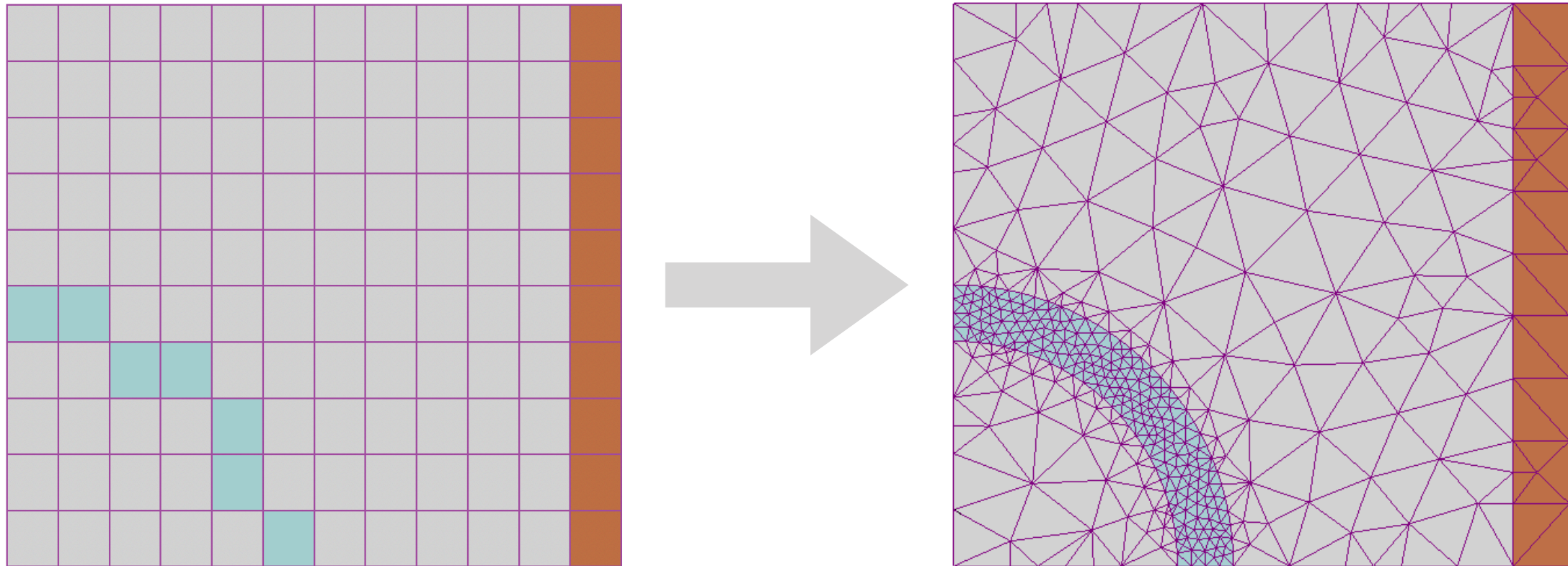
+

élévation à la
puissance N

=

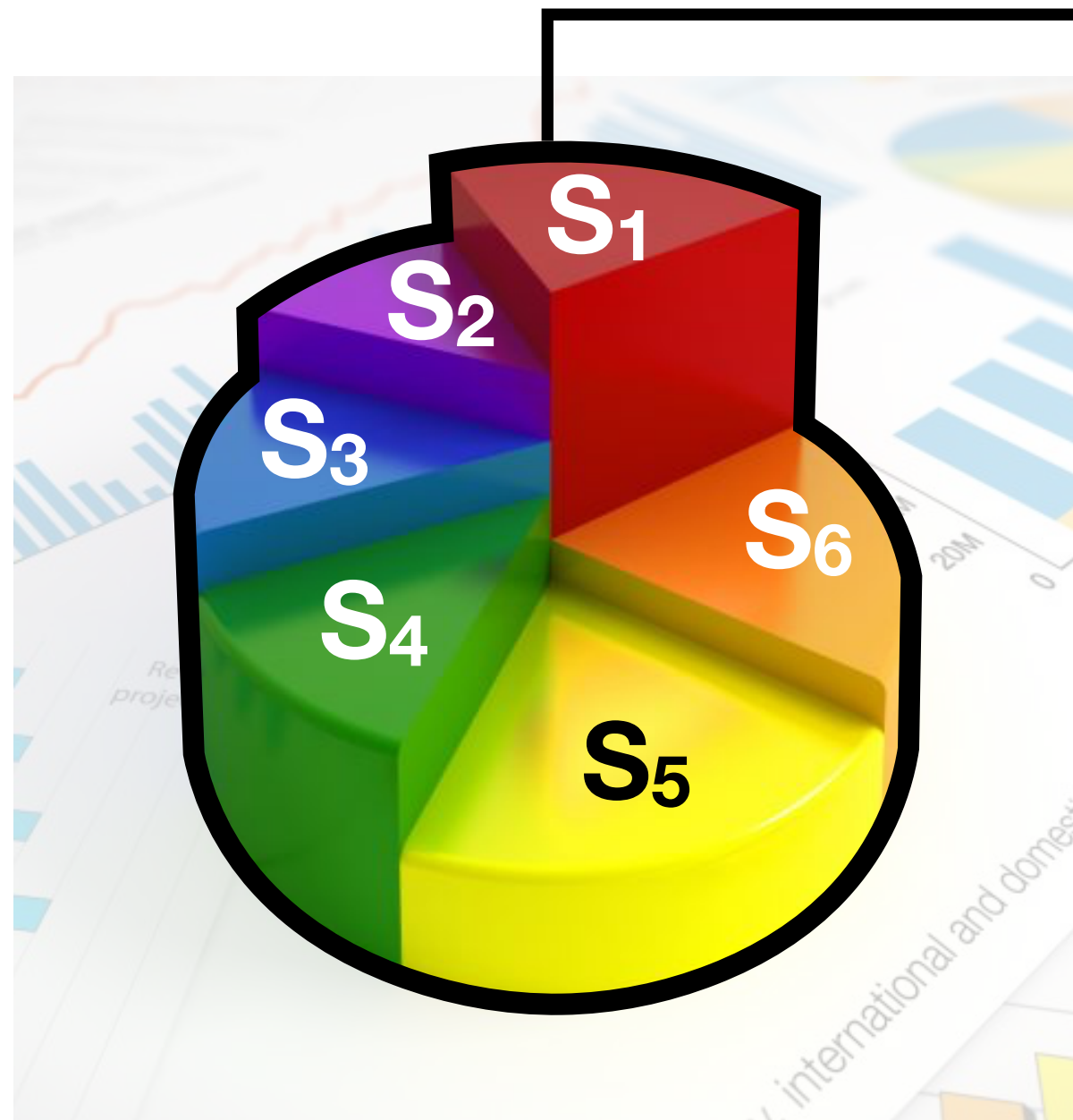
Voies d'amélioration

2. Maillage adapté



Voies d'amélioration

3. Plusieurs sous-jacents



Valeur ?

Dimension
élevée donc V
**plus difficile à
approximer**

Merci.