1. Introduction

This document outlines the design and implementation of a pricing system for two financial instruments: **ValueNotes** and **DeliveryContracts**, as described in the provided problem statement.

2. System Architecture Overview

The system is organized into two core classes:

- **ValueNote**: Represents individual tradable fixed-income instruments.
- **DeliveryContract**: Represents derivative contracts for a basket of ValueNotes.

Supporting structures such as `BasketElement` and utility methods ensure clean abstraction and modularity.

3. Design Choices & Assumptions

- **Object-Oriented Approach**: Classes were used to encapsulate behavior and data of financial instruments.
- **Exception Handling**: Errors relating to invalid inputs have been handled using assert.
- **Binary Search**: Used for rate-to-price inversions where analytical inverses aren't easily derivable.
- **Probability calculation:** Since binomial distributions are difficult to evaluate exactly over closed integrals, estimates similar to midpoint reimann sum have been used. Taking into consideration its slowness, integrateBinomialFast has been implemented to estimate this integral faster.

4. ValueNote Class

- **Class Parameters:**
    - Notational
    - Maturity
    - Value rate
    - Payment Frequency
- **Methods**:

- **For calculating the price:**
  - **getPriceLinear**
  - **getPriceCumulative**
  - **getPriceRecursive**
- **For calculating the Rate:**
  - **getRateLinear**
  - **getRateCumulative**
  - **getRateRecursive**
- **For calculating** $VP'(ER_0)$**:**
  - **getSinglePriceSensitivityLinear**(input is optional)
  - **getSinglePriceSensitivityCumulative**
  - **getSinglePriceSensitivityRecursive**
- **For calculating** $ER'(VP_0)$**:**
  - **getSingleRateSensitivityLinear**(input is optional)
  - **getSingleRateSensitivityCumulative**
  - **getSingleRateSensitivityRecursive**
- **For calculating** $VP''(ER_0)$**:**
  - **getDoublePriceSensitivityLinear**(input is optional)
  - **getDoublePriceSensitivityCumulative**
  - **getDoublePriceSensitivityRecursive**
- **For calculating** $ER''(VP_0)$**:**
  - **getDoubleRateSensitivityLinear**(input is optional)
  - **getDoubleRateSensitivityCumulative**
  - **getDoubleRateSensitivityRecursive**

All first and second derivatives were analytically derived as required and implemented accordingly.

5. DeliveryContract Class

- **Class Parameters:**
  - **Basket**: Stores all the added valueNotes in an array in the structure of Bas.
  - **Standard Value Rate(SVR)**: Used for finding Relative Factor.
  - **Expiration Time**
  - **Risk Free Interest Rate**

- **Methods:**
  - **getRelativeFactor**
  - **addValueNote**
  - **getEffectiveRate**: returns $ER_{T_{ED}}$
  - **For estimating quadratic coefficients:**

        - **weightedQuadraticFit**
        - **computeQuadraticCoefficients**
    - **getContractPrice**
    - **getDeliveryProbabilities**: returns a vector of doubles representing the delivery probability of each ValueNote.
    - **getPriceSensitivity**

6. Challenges & Validation

- **Challenges:**
    - Recursive pricing derivatives were non-trivial; handled via carefully built recurrence relations.
    - Binary search for rate ensures precision of upto .0001% and can be changed in the macros statements.

- **Validation**:

    - Consistency checks for price-to-rate and rate-to-price.
    - Reasonable delivery probabilities (summing close to 1).
    - Manual comparison of outputs with theoretical expectations.
    - Validated the Delivery Contract Price using function getContractPriceAccurate which finds the best economic option at all points instead of only at intersections to get the same answer. This function is also scalable as long as strength of basket is of the order 10^4(after which its execution time exceeds a few seconds). The same is true for getPriceSensitivityAccurate and getDeliveryProbabilitiesAccurate.

7. Extensibility & Maintenance

- Clearly defined class interfaces make it easy to extend:
    - Add new rate conventions.
    - Support stochastic interest rate models.
- Separation of logic and data enables integration with external data feeds or UI.

8. Analytical Calculations:

- **For question 1:**

Linear :

$$VP_0 = N\left(1 - ER_0 \times \frac{M}{100}\right)$$

$$\frac{\partial VP_0}{\partial ER_0} = -\frac{NM}{100} \qquad \frac{\partial ER_0}{\partial VP_0} = \frac{100}{-NM}$$

$$\frac{\partial^2 VP_0}{\partial^2 ER_0} = 0 \qquad \frac{\partial^2 ER}{\partial^2 VP} = 0$$

Cumulative :

$$VP_0 = \frac{VR \times N}{100\,PF} \sum_{i=1}^{n} \left(1+\frac{ER_0}{100PF}\right)^{-PF \times t_i} + \frac{N}{\left(1+\frac{ER_0}{100PF}\right)^{PF \times t_n}}$$

Here $n = \frac{M}{\cancel{100}} \times PF$

$$t_i = \frac{i}{PF}$$

$$\therefore VP_0 = \frac{VR \times N}{100PF} \times \frac{1-\left(1+\frac{ER_0}{100PF}\right)^{-\left(M+\frac{1}{PF}\right)PF}}{\left(1-\left(1+\frac{ER_0}{100PF}\right)^{-1}\right)\left(1+\frac{ER_0}{100PF}\right)} + \frac{N}{\left(1+\frac{ER_0}{100PF}\right)^{M.PF}}$$

$$= \frac{VR \times N}{100PF} \frac{\left(1-\left(1+\frac{ER_0}{100PF}\right)^{-(M.PF+1)}\right)}{1+\frac{ER_0}{100PF}-1} + \frac{N}{\left(1+\frac{ER}{100PF}\right)^{M.PF}}$$

$$= \frac{VR \times N}{ER_0}\left(1-\frac{1}{\left(1+\frac{ER_0}{100PF}\right)^{MPF+1}}\right) + \frac{N}{\left(1+\frac{ER}{100PF}\right)^{M.PF}}$$

$$VP'(ER_0) = \frac{-VR \times N}{(ER_0)^2}\left(1-\frac{1}{\left(1+\frac{ER_0}{100PF}\right)^{MPF+1}}\right) + \frac{VR \times N}{ER_0} \times \frac{(MPF+1)}{100PF\left(1+\frac{ER_0}{100PF}\right)^{MPF+2}} - \frac{N.\cancel{M.PF}}{100\cancel{PF}\left(1+\frac{ER}{100PF}\right)^{M.PF+1}}$$

$$= -\frac{VR \times N}{(ER_0)^2} + \frac{VR \times N}{ER_0\left(1+\frac{ER_0}{100PF}\right)^{MPF+1}}\left(\frac{1}{ER_0}+\frac{MPF+1}{100PF+ER_0}\right) - \frac{N.M}{100\left(1+\frac{ER_0}{100PF}\right)^{M.PF+1}}$$

$$ER'(VP_0) = \frac{1}{VP'(ER_0)}$$

$$VP''(ER_0) = \frac{2VR \times N}{(ER_0)^3} - \frac{VR \times N}{ER_0^2\left(1+\frac{ER_0}{100PF}\right)^{2(MPF+1)}}\left(\left(1+\frac{ER_0}{100PF}\right)^{MPF+1} + \frac{(MPF+1)}{100PF}ER_0\left(1+\frac{ER_0}{100PF}\right)^{MPF}\right)\left(\frac{1}{ER_0}+\frac{MPF+1}{100PF+ER_0}\right)$$

$$- \frac{VR \times N}{ER_0\left(1+\frac{ER_0}{100PF}\right)^{MPF+1}}\left(\frac{1}{ER_0^2}+\frac{MPF+1}{(100PF+ER_0)^2}\right) + \frac{NM(MPF+1)}{10^4 PF\left(1+\frac{ER_0}{100PF}\right)^{MPF+2}}$$

$$ER''(VP_0) = -\frac{VP''(ER_0)}{\left(VP'(ER_0)\right)^3}$$

Recursive: $FV_1 = VF_1\left(1 + \frac{ER_0 m_1}{100}\right)$

$$FV_2 = \left(VF_1\left(1 + \frac{ER_0 m_1}{100}\right) + VF_2\right)\left(1 + \frac{ER_0 m_2}{100}\right)$$

$$= VF_1 + VF_2 + \frac{ER_0 m_1}{100} + \frac{ER_0 m_2}{100}(VF_1 + VF_2) + \frac{ER_0^2 m_1 m_2}{100^2} VF_1 \ldots$$

$$FV_1'(ER_0) = \frac{VF_1 m_1}{100}$$

$$m_i = \frac{1}{pF}$$

$$FV_i'(ER_0) = \left(\left(VF_{i-1} + FV_{i-1}\right)\left(1 + \frac{ER_0 m_i}{100}\right)\right)'$$

$$= \frac{m_i}{100}\frac{FV_i}{\left(1 + \frac{ER_0 m_i}{100}\right)} + FV_{i-1}'\left(1 + \frac{ER_0 m_i}{100}\right) = \frac{m_i}{100}(VF_{i-1} + FV_{i-1}) + FV_{i-1}'\left(1 + \frac{ER_0 m_i}{100}\right)$$

($\because VF_i$ is const)

$\therefore FV_n'(ER_0)$ can be found by recursion

$$VP_0'(ER_0) = \frac{FV_n'\left(1 + \frac{ER_0 m}{100}\right) + \frac{m}{100} FV_n}{\left(1 + \frac{ER_0 m}{100}\right)^2}$$

$$ER_0'(VP_0) = \frac{1}{VP'(ER_0)}$$

~~$VP''(ER_0) =$~~

$$FV_1''(ER_0) = 0$$

$$FV_2''(ER_0) = \frac{2 m_1 m_2 \cdot VF_1}{10^4}$$

$$FV_i''(ER_0) = \frac{m_i FV_{i-1}'}{100} + FV_{i-1}''\left(1 + \frac{ER_0 m_i}{100}\right) + FV_{i-1}' \cdot \frac{m_i}{100}$$

$$= \frac{2 m_i}{100} FV_{i-1}' + FV_{i-1}''\left(1 + \frac{ER_0 m_i}{100}\right)$$

$FV_n''(ER_0)$ can be found by recursion.

$$\therefore VP''(ER_0) = \left(\frac{FV_n'}{1+\frac{ER_0 m}{100}} + \frac{m}{100} \frac{FV_n}{\left(1+\frac{ER_0 m}{100}\right)^2} \right)'$$

$$= \frac{FV_n''\left(1+\frac{ER_0 m}{100}\right) - \frac{m}{100} FV_n'}{\left(1+\frac{ER_0 m}{100}\right)^2} + \frac{m}{100} \frac{FV_n'\left(1+\frac{ER_0 m}{100}\right)^2 - FV_n \frac{2m}{100} \cancel{\left(1+\frac{ER_0 m}{100}\right)}}{\left(1+\frac{ER_0 m}{100}\right)^{\cancel{4}\,3}}$$

$$= \frac{FV_n''}{1+\frac{ER_0 m}{100}} - \frac{FV_n \, 2m^2}{10^4\left(1+\frac{ER_0 m}{100}\right)^3}$$

$$ER''(VP_0) = \partial \cancel{\;} \frac{\partial ER}{\partial VP} \times \frac{\partial\left(\frac{1}{VP'(ER)}\right)}{\partial ER}$$

$$= -\frac{VP''(ER_0)}{\left(VP'(ER_0)\right)^3}$$

For Question 2:

Question 2

2.9a) $\text{Price} = \cancel{R(T)} \sum_{i=1}^{n} P(\text{value note } i \text{ is most economical}) \times \left(\frac{VP_i(ER^i_{TED})}{RF_i}\right)$

$$\Rightarrow \frac{d\,\text{Price}}{d\sigma_i} = P(\text{Value of Note } i \text{ is most economical}) \times \frac{d}{d\sigma_i}\left(\frac{VP_i(ER^i_{TED})}{RF_i}\right)$$

$$\frac{d}{d\sigma_i}\left(\frac{VP_i(ER^i_{TED})}{RF_i}\right) = \frac{1}{RF_i} \times \frac{d\,VP(ER^i_{TED})}{d(ER^i_{TED})} \times \frac{d(ER^i_{TED})}{d\sigma_i}$$

$$= \frac{VP'(ER_{TED})}{RF_i} \times ER^i_{TED}\left(W_T - \sigma_i T_{ED}\right)$$

b) $$\frac{d\,\text{Price}}{d\sigma_i} = \int \frac{VP'(ER_{TED})}{RF_i}\left(W_T - \sigma_i T_{ED}\right) \phi(W_T)$$

over all z where z_i th value not is most economical

$N(0, \sqrt{T_{ED}})$

most economical

b) $\frac{\partial \, Price}{\partial VP_{0i}} = \frac{\partial}{\partial VP_{0i}} \left(\frac{VP_i(ER^i_{TED})}{\partial VP_{0i}} \right) \times$ ~~Prob that~~ (1th value note is most economical)

$$\frac{\partial}{\partial VP_{0j}} \left(\frac{VP_i(ER_{TED})}{RF_i} \right) = \frac{1}{RF_i} \times \frac{\partial}{\partial VP_0} (VP_i(ER_{TED}))$$

$$\frac{\partial \widetilde{VP}_{TED}}{\partial VP_0} = (1 + \tilde{r} t_{TED})$$

$$\therefore \frac{d\widetilde{ER}^i}{dVP_0} = \frac{\partial \widetilde{VP}_{TED}}{dVP_0} \times \frac{d\widetilde{ER}^i_*}{dVP_{TED}}$$

$$= (1+\tilde{r}t)\, \widetilde{ER}'^i (VP_{TED}) = \Delta \widetilde{ER} \quad \text{(say)}$$

$$\overline{ER}^{i\,2}_T \left(\frac{1}{2} VP^{i\,\prime\prime}_T (\widetilde{ER}^i_T) e^{\sigma_i^2 T} \right) + \overline{ER}^i_T \left(VP^{i\,\prime}_T (\widetilde{ER}^i_T) - VP^{i\,\prime\prime}_T (\widetilde{ER}^i_T) \widetilde{ER}^i_T \right)$$

$$+ \frac{1}{2} VP^{i\,\prime\prime}_T (\widetilde{ER}^i_T) - VP^{i\,\prime}_T (\widetilde{ER}^i_T) \widetilde{ER}^i_T$$

$$\overline{ER}^i_T \frac{d\overline{ER}^i_T}{d\widetilde{ER}^T} \left\{ x \, VP^{i\,\prime\prime}_T (\widetilde{ER}^i_T) e^{\sigma_i^2 T} \right\} + \frac{\overline{ER}^{i\,2}_T}{2} e^{\sigma_i^2 T} VP^{\prime\prime\prime}_T (\widetilde{ER}^i_T) + \overline{ER}^i_T \left(VP^{i\,\prime\prime}_T (\widetilde{ER}^i_T) - VP^{i\,\prime\prime\prime}_T (\widetilde{ER}^i_T) \widetilde{ER}^i_T \right.$$

$$-VP_T$$

(unfinished)

9. Conclusion

This system meets the outlined requirements, demonstrates analytical rigor, and is built with clean, modular C++ practices that can scale to production use.