JPMC Quant Challenge ’21
DERIVATIVES MODELLING

SOLUTION APPROACH

Rudransh Jaiswal
B.Tech. in Mechanical Engineering, M.Tech. (Dual Degree) in Data Science,
IIT-Madras

CONTENTS

1. No Arbitrage

❖ Method

- **Naive Implementation**: Check for all triangular arbitrage using graph traversal

Time Complexity : O(n³), corresponding to ${}^{n}C_3$

- **Better Implementation**

Only (n-1) equations, that is just one row is sufficient to derive no-arbitrage pricing for any exchange

Errors due to floating point operations and decimal rounding off must be handled

$$fx[i][j] \leftarrow \frac{1.0}{fx[j][i]} \;\forall i > j$$

- Taking the 1st currency as a reference, the relative prices must match for all the exchanges : $\dfrac{fx[i][j]}{fx[i][0]} = fx[0][j] \;\; \forall i, j$ in case of no arbitrage

- Formally a tolerance value of 0.01 was used to avoid floating point errors

  That is, there is an arbitrage if $\exists\ (i, j) : \left| \dfrac{fx[i][j]}{fx[i][0] \cdot fx[0][j]} - 1.0 \right| > 0.01$

Time Complexity : O(n²)

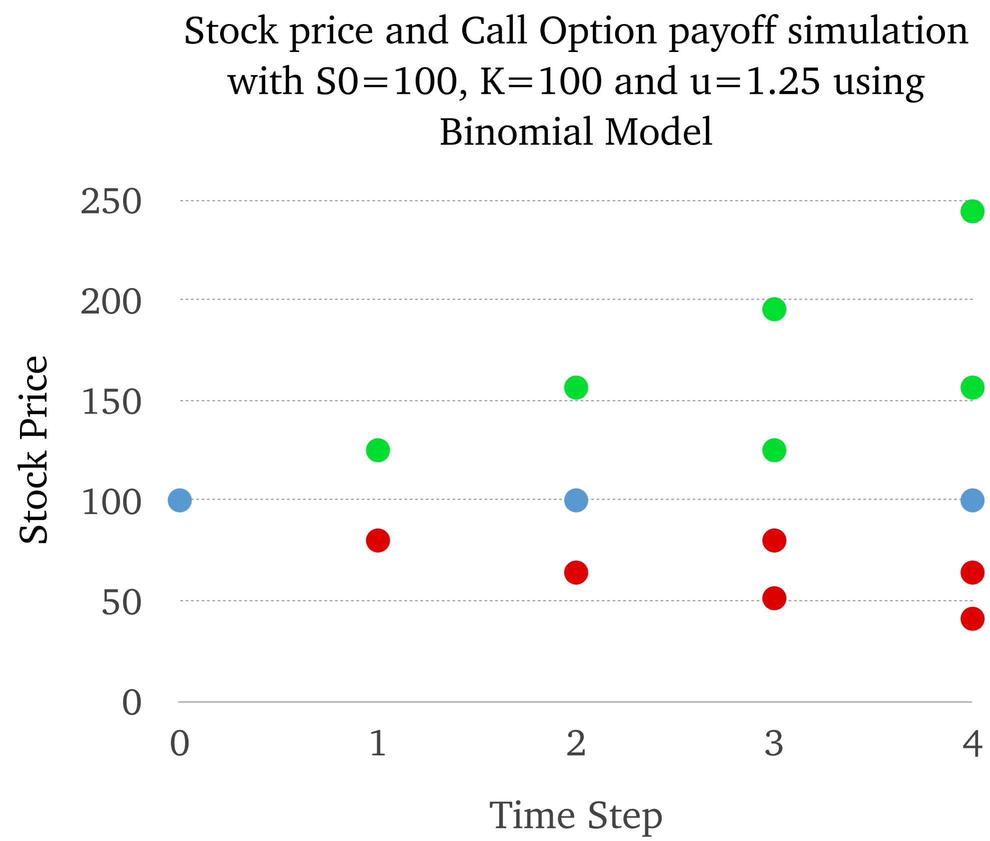
	USD	INR	EUR
USD	1.0	75	0.833
INR	0.013	1.0	0.02
EUR	1.2	50	1.0

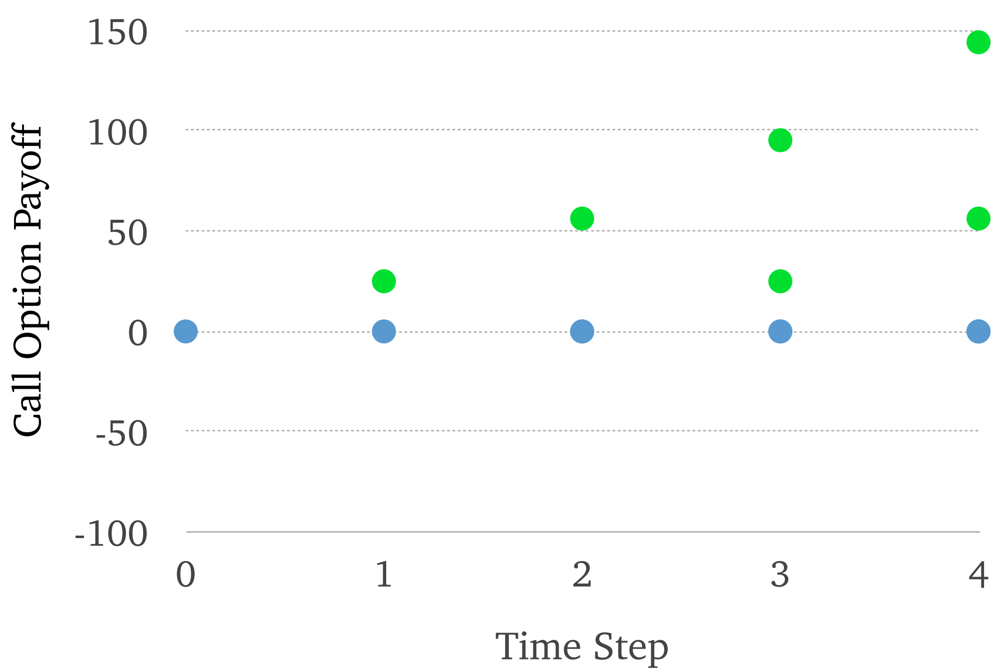
	USD	INR	EUR
USD	1.0	75	0.833
INR	1/75	1.0	0.02
EUR	1/0.833	50	1.0

	USD	INR	EUR
USD	1.0	75	0.833
INR x 75	1.0	75	1.5
EUR x 0.833	1.0	41.65	0.833

2. Binomial Model

❖ Method





- Equating initial stock price with discounted expectation of stock price after an unit time-step :

$$S_o R = p(uS_o) \; + \; (1-p)(dS_o)$$

$$p = \frac{R-d}{u-d}$$

- Price of Call Option is the discounted value of its expected payoff :

$$V = R^{-N}\, \mathbb{E}_{S_N}\left([S_N - K]^+\right)$$

$$\mathbb{P}\left(S_N = S_o\, u^i\, d^{N-i}\right) = \binom{N}{i}\, p^i\, \bar{p}^{N-i}$$

$$V = R^{-N} \sum_{i=0}^{N} \binom{N}{i}\, p^i\, \bar{p}^{N-i}\, \left([S_o\, u^i\, d^{N-i} - K]^+\right)$$

Time Complexity : O(n)

Space Complexity : O(n)

3. Bootstrapping

❖ Method

- Normally the interest rate can be determined using the following equation

$$r(T) = -\frac{1}{T}\log_e\left(\frac{P_b - \sum_{t=1}^{T-1} c \cdot \exp(-r(t)\cdot t)}{FV + c}\right)$$

- For the missing maturity time, the latest calculated values were used for extrapolation:

$$r(T) = 2r(T-1) - r(T-2)$$

- Naive Implementation:

Time Complexity : $O(n^2)$

- **Dynamic Programming approach:**

The computed value of $\sum_{t=1}^{T-1} \exp(-r(t) * t)$ can be stored in a single variable

Time Complexity : O(n)

Space Complexity : O(n)

4. Bootstrapping using Nelson Siegel (NS) model (1)

❖ Method

1. Determine all the payment times for each time of maturity, its corresponding interest rates and store in a hash-map
2. Build the discounted cash flow matrix using the hash-map
3. The bond price is simply the sum of the corresponding column

These bond prices can be compared with the market traded bond price to determine squared error

❖ Parameter Estimation

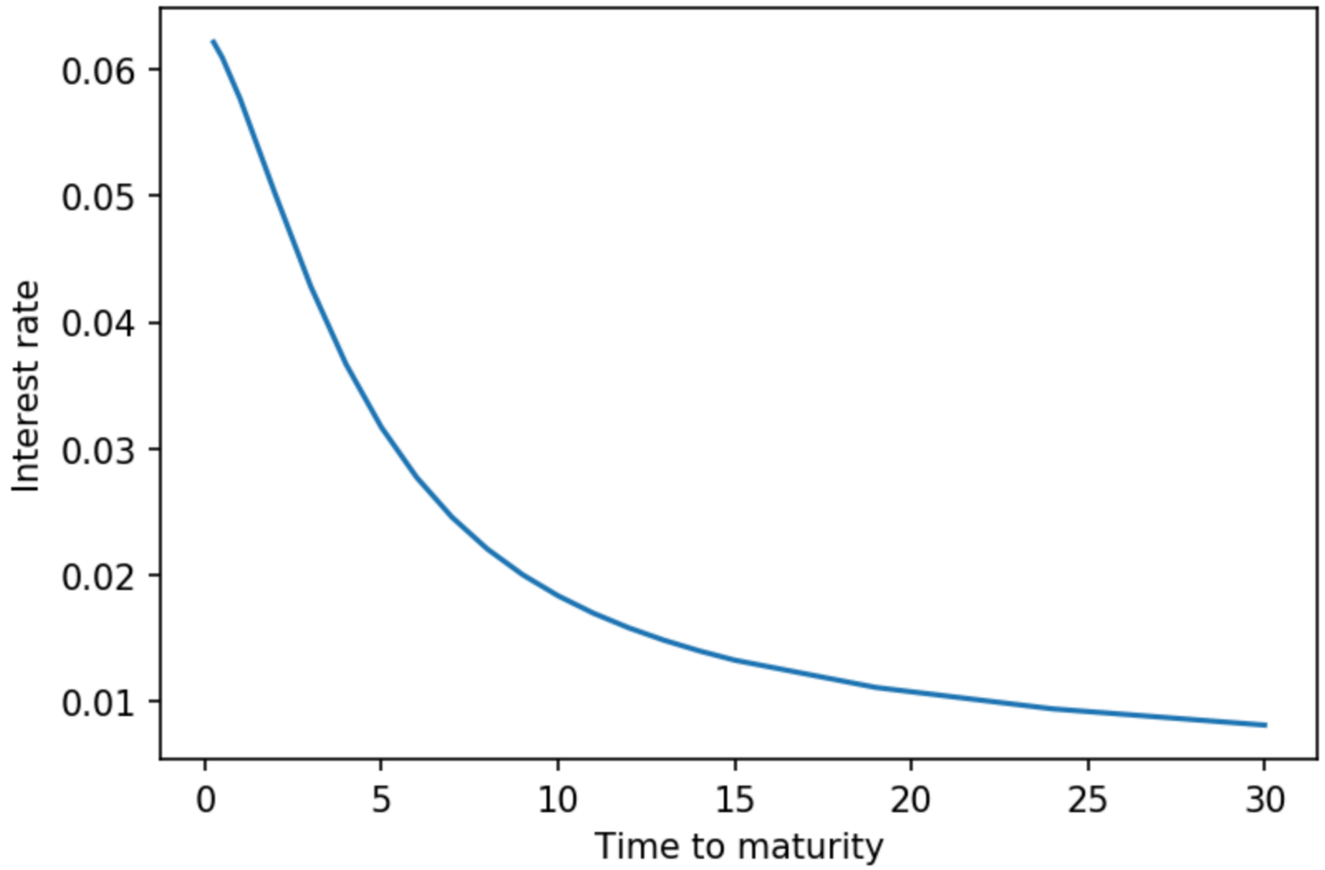
1. The NS model parameters can be estimated using non-linear regression techniques like **Levenberg Algorithm**:
   $f(m, A) = \hat{y}$; A : design vector
   $(Z^T Z + \lambda I)\Delta A = Z^T D$; Z : Jacobian matrix of the objective; D : $\hat{Y} - Y_{actual}$
2. If the parameter τ is fixed, simple least square regression or ridge regression can be used for estimation

❖ Analysis of NS model

1. $\lim_{m \to 0} y(m) = \beta_0 + \beta_1$: Signifies short term interest rates
2. $\lim_{m \to \infty} y(m) = \beta_0$: Signifies long run levels of interest rates
3. τ determines how quickly the the asymptotic value of $\beta_0 + \beta_1$ is achieved
4. Through simulations it was observed that β_2 contributes to the mid term interest rates, a high value of β_2 will lead to a hump
5. $y(m = \tau) = \beta_0 + 0.632\beta_1 + 0.264\beta_2$
   $y(m = 10\tau) = \beta_0 + 0.099\beta_1 + 0.099\beta_2 \approx \beta_0$; That is the curve almost saturates after $m = 10\tau$

4. Bootstrapping using Nelson Siegel (NS) model (2)

❖ **Inverted Yield Curve as an indicator for recession and prediction using NS model**



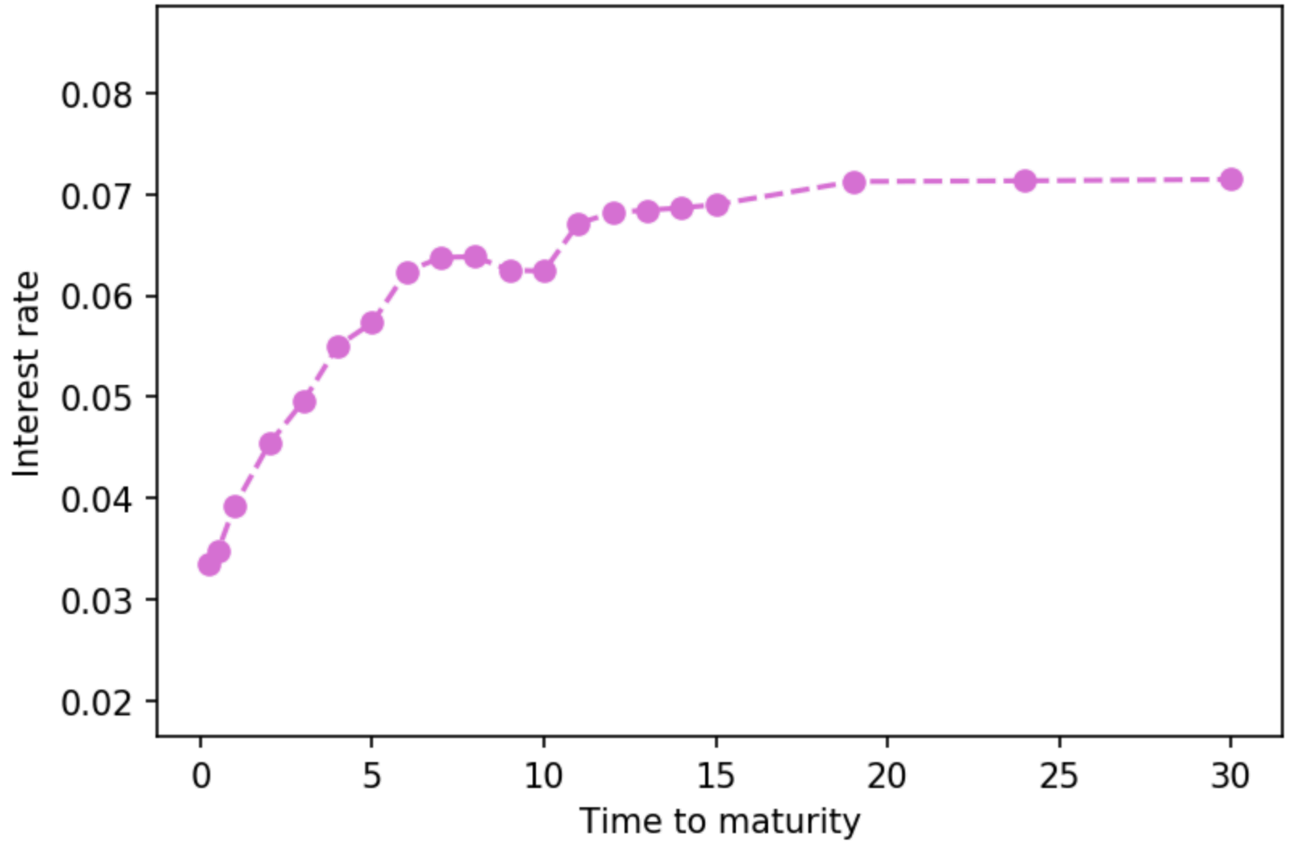
An inverted yield curve, $\beta_1 > 0$

- In the Inverted Yield Curve the short term interest rates are higher than the long term interest rates
  Similarly during recession, the short term interest rates exceed long term interest rate
- Therefore $\beta_1 > 0$ implies an inverted curve
  Using simulations it is also observed that high value of β_2 gives a partially inverted curve

4. Bootstrapping using Nelson Siegel (NS) model (3)

❖ Guess-Estimate of NS parameters in a developing market

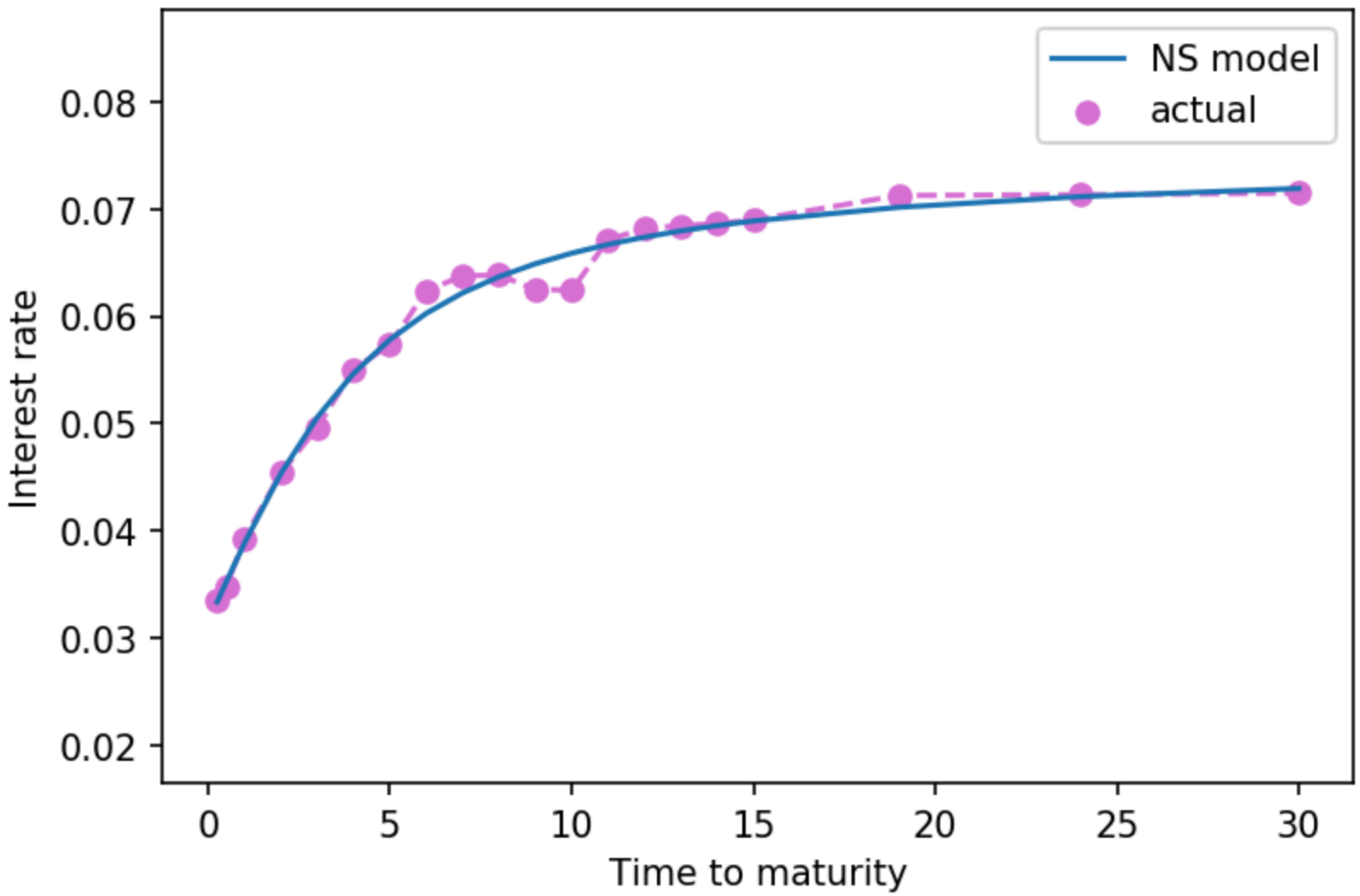
- In a developing country like India a normal yield curve is expected, $\beta_1 < 0$
  The short term interest rates are ~3.2% — 3.4%
  $\Rightarrow \beta_0 + \beta_1 \approx 0.033$
  Long term interest rates are ~6.8% — 7.5%
  $\beta_0 \approx 0.068 \;—\; 0.075$
  $\beta_1 \approx -0.043 \;—\; -0.034$
- Also we observe that the interest rates almost saturate after maturity time of 14 - 15 years:
  $10\tau \approx 15 \;\Rightarrow \tau \approx 1.4 \;—\; 1.5$
- $y(m = \tau) = \beta_0 + 0.632\beta_1 + 0.264\beta_2$
  $y(m = 1.5) \approx \beta_0 + \beta_1 \;\Rightarrow \beta_2 \approx 1.39\beta_1$
  $\beta_2 \approx -0.059 \;—\; -0.0473$



Actual yield curve for Indian bonds

4. Bootstrapping using Nelson Siegel (NS) model (4)

❖ **Estimates of NS parameters for Indian Bonds using Regression**



Estimates using OLS with grid search :

$\hat{\tau} = 1.36$

$\hat{\beta}_0 = 0.0749$

$\hat{\beta}_1 = -0.0433$

$\hat{\beta}_2 = -0.0233$

squared error $= 2.89 * 10^{-5}$

5. Delta Hedging using NS model

❖ Method

1. A hash-map was used to map payment times to the corresponding interest rate determined by NS model
2. The function written in the previous code was used to find bond prices given the interest rates
3. An iterative algorithm was run to calculate ΔB_{ij} by perturbing interest rates at the benchmark tenors
4. Since the matrix ΔB is invertible, weight vector can be determined using $W^T = \Delta V(\Delta B)^{-1}$

Time Complexity : $O(n^3)$ due to matrix multiplication and inversion

Space Complexity : $O(n^2)$

❖ Calculating Hedge weights in case of non-inverting matrix

m= number of bonds used for hedging, n= number of instruments to be hedged or the number of tenors

- The equation $W^T \Delta B = \Delta V$ is a set of 'n' linear equations with 'm' unknowns
  i. If m>n: Mathematically more than one solution exist
     Various combinations of weights (including zero weights) can be used to match the risk
  ii. If m<n: No solution exist

5. Delta Hedging using NS model

- We wish to find $W : (\Delta B)^T W = (\Delta V)^T$
  The Least Square estimate is given by : $\hat{W} = \text{argmin}_W || (\Delta B)^T W - (\Delta V)^T ||_2^2$
- Consider the case when m $\neq$ n:
  It is an ill-posed problem as there is no unique solution
  Tikhonov regularization can be used to address this

- For the ill posed problem the objective function can be regularized with weighted norm of design vector:
  $\mathscr{L} = || \Delta B^T W - \Delta V^T ||_2^2 + || \Gamma W ||_2^2$;
  Γ_{mxm} is a diagonal matrix which assigns relative importance to the weights (bonds)
  $\hat{W} = \text{argmin}_W \mathscr{L}$
  $\bar{\nabla}_w \mathscr{L} = \bar{0}$ gives $\hat{W} = (\Delta B \Delta B^T + \Gamma^T \Gamma)^{-1} \Delta B \Delta V^T$

- Regularization is used to reduce the variance of the model in general and get stable solution
  - Tikhonov regularization gives handle to assign relative importance to weights unlike Ridge Regression

References & Acknowledgement

1. Indian Bond Prices: http://www.worldgovernmentbonds.com/country/india/
2. Yield curve and effect on bond yield due to supply and demand: https://en.wikipedia.org/wiki/Yield_curve
3. Programming codes and Optimization techniques: Institute course-work

Thank You