# Project-2: Auto-Callable Barrier Notes with Contingent Coupons

S&P 500:
https://www.sec.gov/Archives/edgar/data/927971/000121465922011753/b930221424b2.htm

---

Group:

*Venu Madhav Menday (vmenday2)*
*Vineeth Kotala (vkotala2)*

## Contents

# BMO  $438,000 Autocallable Barrier Notes with Contingent Coupons due November 03, 2023 Linked to the S&P 500® Index

## Introduction:

### Value:

The estimated value of a note is **$975.94**

### Data:

Most of the data is picked from the term sheet while data on interest rates, underlying volatility and dividend yield are taken from the Bloomberg portal.

| | | Time (Yrs) | Days |
|---|---|---|---|
| *Pricing Date* | 9/28/2022 | 0.0000 | 0 |
| *Settlement Date* | 10/03/2022 | 0.0137 | 5 |
| *Valuation Date* | 10/31/2023 | 1.0904 | 398 |
| *Maturity Date* | 11/03/2023 | 1.0986 | 401 |

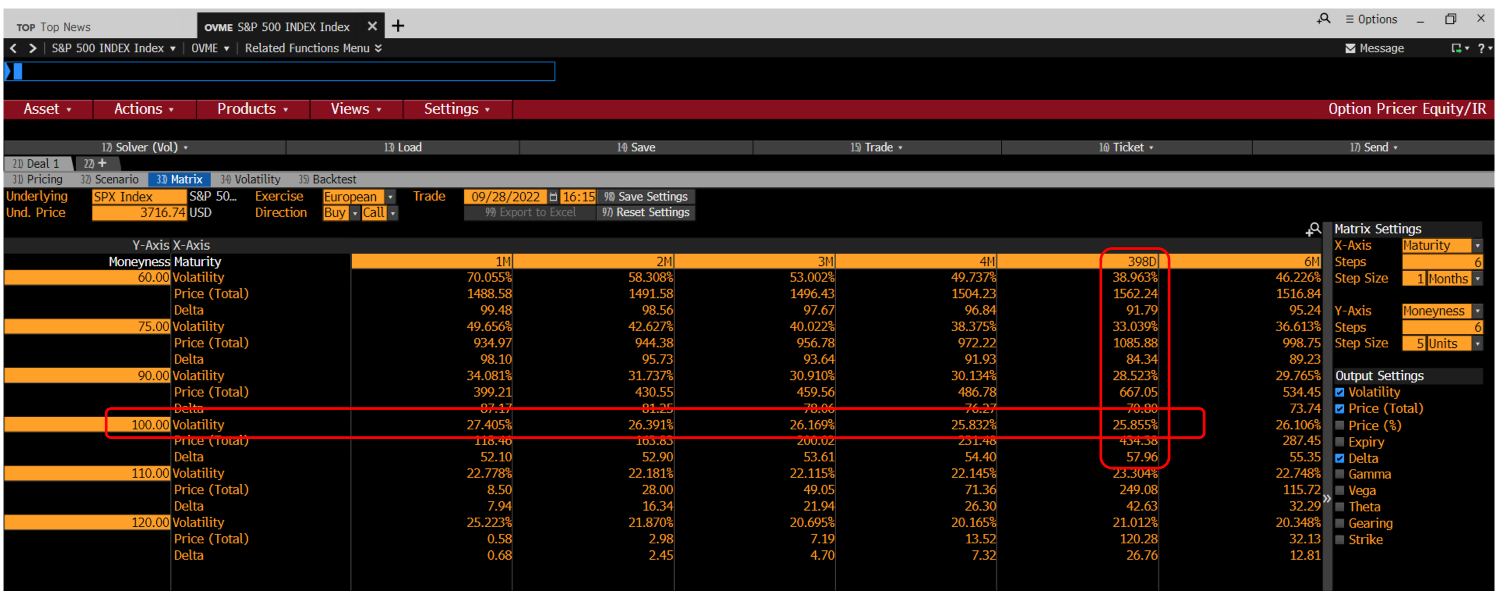
| ***Underlying asset*** | **S&P 500** |
|---|---|
| *Principal Amount / Face Value* | $1,000.00 |
| *Underlying Initial value* | $3,719.04 |
| *Coupon Barrier 50% of the initial value* | $2,789.28 |
| *Downside threshold 50% of the initial value* | $2,789.28 |
| *Annual Contingent Coupon rate* | 11.60% |
| *Monthly coupon rate* | 0.9667% |
| *Coupon dollar amount* | $9.67 |
| **Data collected from Bloomberg (*refer to screenshots)** | |
| *Volatility*<br>*(Implied volatility of European call ATM at maturity)* | 0.25855 |
| *Risk neutral rate*<br>*-log(Df)/T where Df discount factor from OIS* | 0.04212 |
| *Risk-free rate* | 0.04212 |
| *Dividend yield*<br>*(Annual, assumed to be continuous)* | 0.01814 |

**Assumption on observation and coupon payment dates:**

Dates marked in red are those where automatic redemption is applicable.

| Observation Dates | Days | Coupon Payment Dates | Days |
|---|---|---|---|
| 3-Nov-22 | 36 | 31-Oct-22 | 33 |
| 3-Dec-22 | 66 | 30-Nov-22 | 63 |
| 3-Jan-23 | 97 | 31-Dec-22 | 94 |
| 3-Feb-23 | 128 | 31-Jan-23 | 125 |
| 3-Mar-23 | 156 | 28-Feb-23 | 153 |
| 3-Apr-23 | 187 | 29-Mar-23 | 182 |
| 3-May-23 | 217 | 30-Apr-23 | 214 |
| 3-Jun-23 | 248 | 31-May-23 | 245 |
| 3-Jul-23 | 278 | 30-Jun-23 | 275 |
| 3-Aug-23 | 309 | 31-Jul-23 | 306 |
| 3-Sep-23 | 340 | 31-Aug-23 | 337 |
| 3-Oct-23 | 370 | 30-Sep-23 | 367 |
| 3-Nov-23 | 401 | 31-Oct-23 | 398 |

Bloomberg data:




| Y-Axis Moneyness | X-Axis Maturity | 1M | 2M | 3M | 4M | 398D | 6M |
|---|---|---|---|---|---|---|---|
| 60.00 | Volatility | 70.055% | 58.308% | 53.002% | 49.737% | 38.963% | 46.226% |
| | Price (Total) | 1488.58 | 1491.58 | 1496.43 | 1504.23 | 1562.24 | 1516.84 |
| | Delta | 99.48 | 98.56 | 97.67 | 96.84 | 91.79 | 95.24 |
| 75.00 | Volatility | 49.656% | 42.627% | 40.022% | 38.375% | 33.039% | 36.613% |
| | Price (Total) | 934.97 | 944.38 | 956.78 | 972.22 | 1085.88 | 998.75 |
| | Delta | 98.10 | 95.73 | 93.64 | 91.93 | 84.34 | 89.23 |
| 90.00 | Volatility | 34.081% | 31.737% | 30.910% | 30.134% | 28.523% | 29.765% |
| | Price (Total) | 399.21 | 430.55 | 459.56 | 486.78 | 667.05 | 534.45 |
| | Delta | 87.17 | 81.25 | 78.06 | 76.27 | 70.80 | 73.74 |
| 100.00 | Volatility | 27.405% | 26.391% | 26.169% | 25.832% | 25.855% | 26.106% |
| | Price (Total) | 118.46 | 163.83 | 200.02 | 231.48 | 434.38 | 287.45 |
| | Delta | 52.10 | 52.90 | 53.61 | 54.40 | 57.96 | 55.35 |
| 110.00 | Volatility | 22.778% | 22.181% | 22.115% | 22.145% | 23.304% | 22.748% |
| | Price (Total) | 8.50 | 28.00 | 49.05 | 71.36 | 249.08 | 115.72 |
| | Delta | 7.94 | 16.34 | 21.94 | 26.30 | 42.63 | 32.29 |
| 120.00 | Volatility | 25.223% | 21.870% | 20.695% | 20.165% | 21.012% | 20.348% |
| | Price (Total) | 0.58 | 2.98 | 7.19 | 13.52 | 120.28 | 32.13 |
| | Delta | 0.68 | 2.45 | 4.70 | 7.32 | 26.76 | 12.81 |

*Figure 1Volatility*

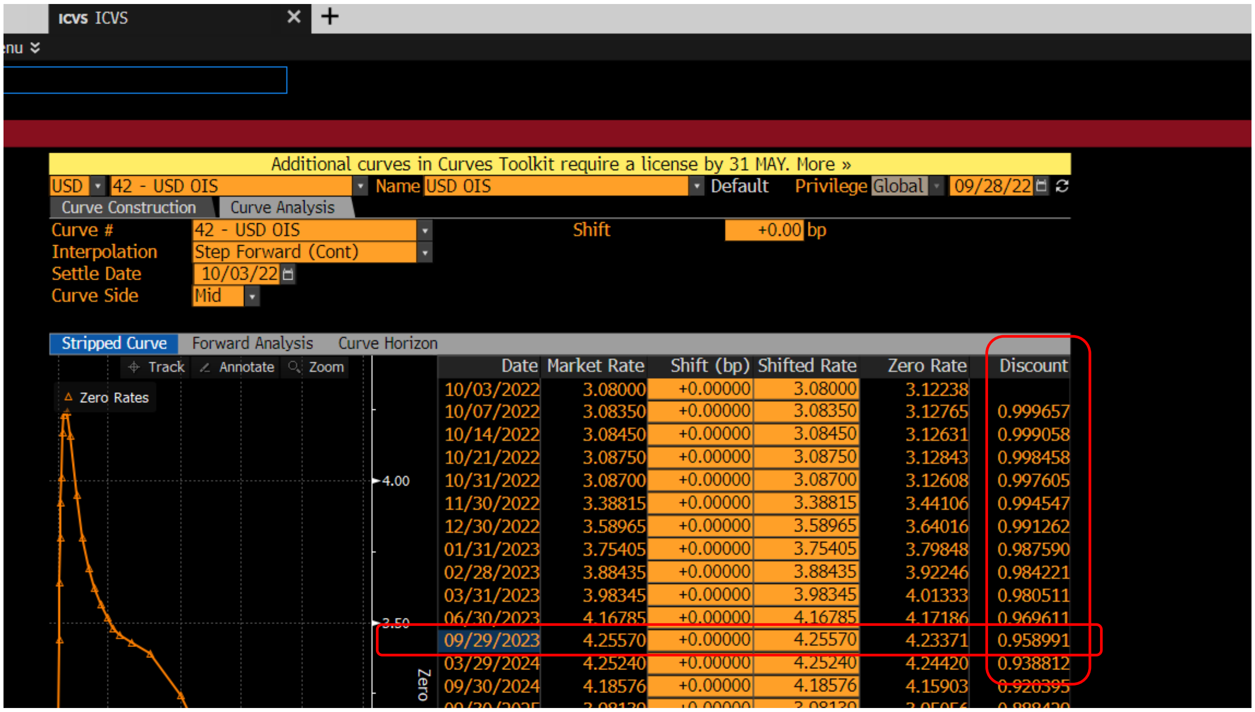



*Figure 2Discount factors for Interest rates*


*Figure 1Dividend Yield*

# Crank Nicolson Model:

## Assumptions:

1. We estimated the value of BMO notes with S&P500 as our underlying asset for which we assumed the implied volatilities of European options from Bloomberg as the volatility for our pricing model.
2. Implied volatility ATM at maturity is considered assuming a higher probability of $S_T >= S_0$.

3. USD OIS rates are taken as Interest rates. Both risk neutral and risk-free rates are assumed to be same for the sake of simplicity.
    a. Risk-neutral rate (r1), used for estimating the risk-neutral probabilities, is calculated between the Trade date and the Valuation date, by linear interpolation of discount factors available for dates in Bloomberg.
    b. Similarly risk-free rate (r2), used for discounting the value of options between steps of the grid, is calculated between the Trade date and Maturity date
4. Dividends are assumed to be continuous since the underlying asset is an index.

| | **S.No** | **Dates** | **Discount factor** |
|---|---|---|---|
| Linear Interpolation of discount factors for dates 1 & 2 gives discount factor for date 3 | 1 | 9/29/2023 | 0.958991 |
| | 2 | 3/29/2024 | 0.938812 |
| | 3 | 11/3/2023 | 0.9551104 |

## Model:

**Grid Structure:** for final converged estimate

| | |
|---|---|
| **jmax** *(number of delta S movements)* | 1000 |
| **imax** *(number of time steps)* | 398 X 4 |
| **SU** *(maximum value of S in grid)* | 2.5 X $S_0$ |

**Defining Function:**

We have defined our primary function "CNFD_1" with 17 input parameters given below, and further two more functions are defined for analysis of results and sensitivity study.

**CNFD_1(Face, S0, K, Barr, Tv, Tm, tcp, tob, tauto, cpn, r1, r2, q, sigma, SU, imax, jmax)**

| *Definition of input parameters of function CNFD_1( )* | |
|---|---|
| **Face** | *Principle value* |
| **S0** | *Underlying initial value* |
| **K** | *Autocall barrier* |
| **Barr** | *Coupon barrier* |
| **Tv** | *Valuation Date* |
| **Tm** | *Maturity Date* |
| **tcp** | *Coupon payment dates* |

| | |
|---|---|
| **tob** | *Observation dates* |
| **tauto** | *Automatic redemption beginning date* |
| **cpn** | *Annual Contingent Coupon rate* |
| **r1** | *Risk neutral rate* |
| **r2** | *Risk free rate* |
| **q** | *Dividend yield (annual continuous)* |
| **sigma** | *Volatility* |
| **SU** | *Maximum price of underlying for the grid* |
| **imax** | *maximum time steps* |
| **jmax** | *maximum delta S steps* |

| ***Definition of variables inside the function*** | |
|---|---|
| V1 | *Option value matrix when trigger event is observed* |
| **V2** | *Option value matrix when trigger event has not occurred* |
| **A, B, C, D, D1, D2, D3 alpha, CN_S** | *Parameters from PDE to be solved to get Option values* |
| **dt** | *delta t* |
| **dS** | *delta S* |
| **jb** | *barrier position in stock steps* |
| **ji** | *Stock Initial level* |
| **icp1** | *Coupon payment dates* |
| **icp** | *nearest time point before coupon payment dates* |
| **inc** | *Time points close to coupon payments before Automatic redemption begins* |
| **iob1** | *Observation dates* |
| **iob / ibarrier** | *nearest time point before the observation date* |
| **iauto** | *nearest time points before autocallable observation dates* |
| **TNCi** | *Next coupon payment date* |
| **cpk** | *Array of accrued coupons for calculating option value on observation dates before automatic redemption* |
| **jcritreal** | *Stock steps of initial level* |
| **jcrit** | *Nearest dS step before S0* |
| **jcritB** | *Nearest dS step after barrier* |
| **Vcrit** | *Adjusted option value at S0 and time zero* |
| **Blambda** | *lambda* |
| **cnfddb_result** | *final result* |
| **fun1()** | *function based on primary function CNFD_1 to analyze option values with change in volatilities with one input sigma while others remaining constant* |
| **CNFD_discbarrier()** | *function based on primary function CNFD_1 to analyze change in option values wrt to change in jmax* |

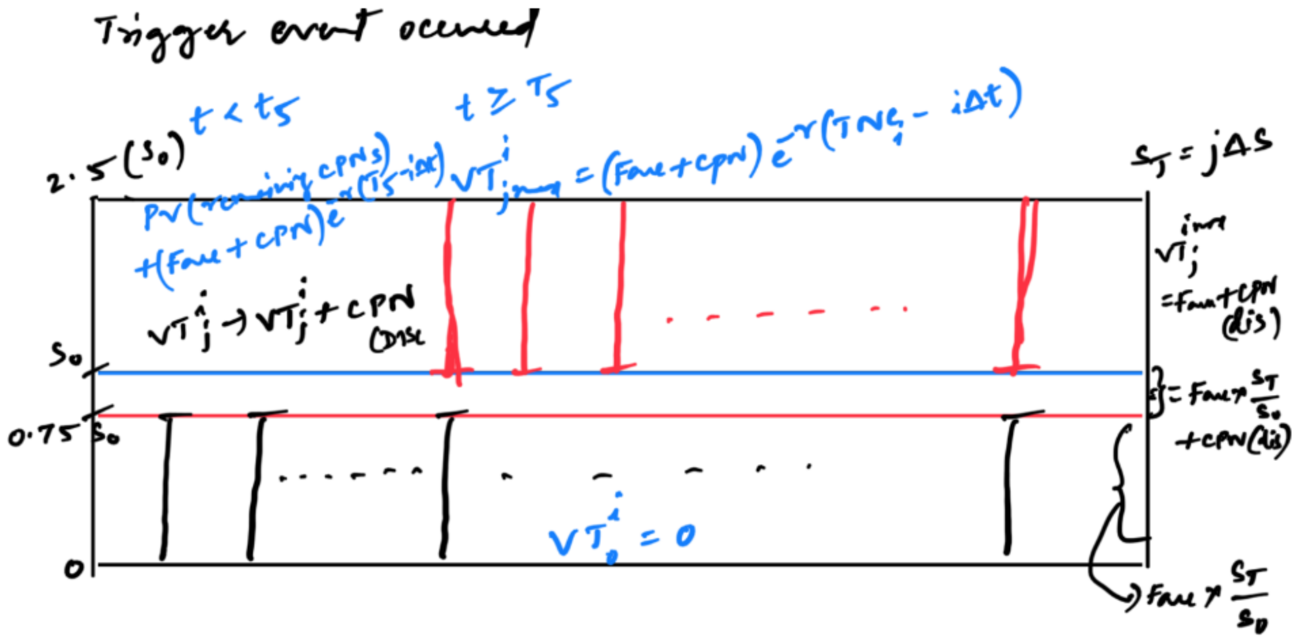
**Methodology:**

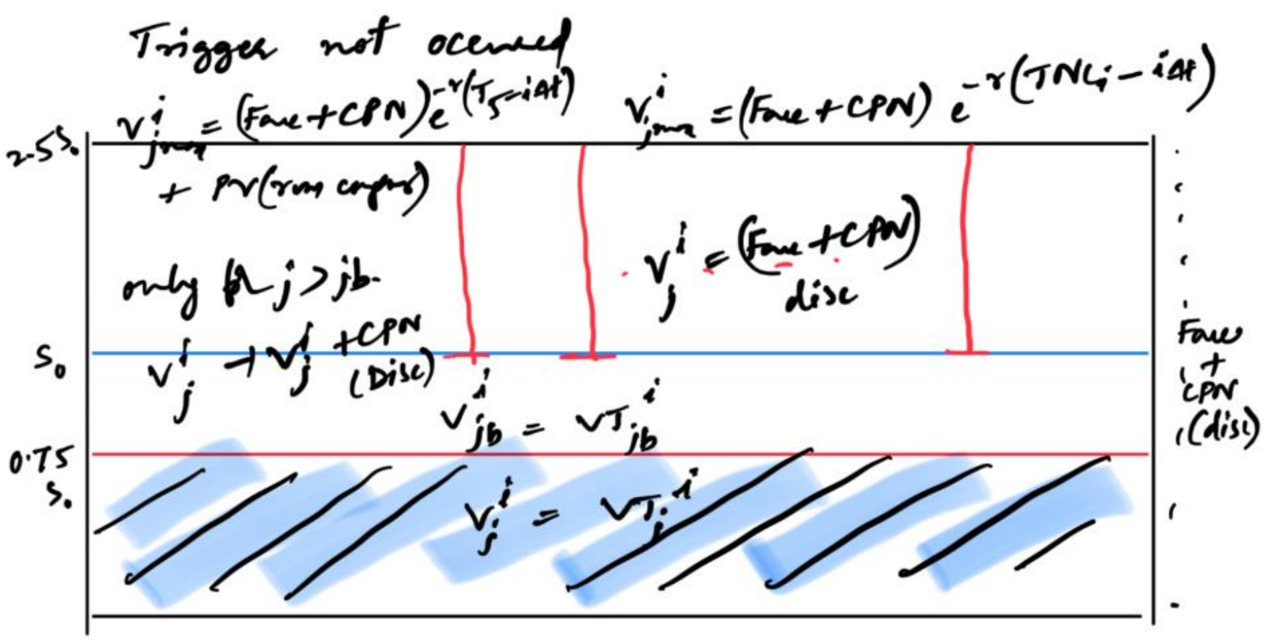
The model is constructed in two stages,

1. Firstly construction of grid considering Trigger event occurred
2. And secondly using the former to construct a new grid assuming trigger event has not occurred

The primary difference between two grids, as it can be seen the picture given below,

- Payment at maturity
- Lower boundary condition, and hence we had to split the for loop for calculating PDE parameters A, B, C, D as given in the snippet of code.



**Function Overview:**

| S.No | Steps |
|---|---|
| 1 | Inputs from term sheet and bloomberg are defined |
| 2 | Function CNFD_1 |
| 2.1 | Empty list is created to save results |
| 2.2 | Creating of required arrays with dummy values:<br>Two matrices with dummy values (zeros) for saving option prices from grid with trigger event and another grid with no trigger event are created<br>Similarly arrays for saving probabilities and parameters (alpha and Z (CN_S) are also created |
| 2.3 | Then we set up structure for grid by using t_Steps and S_steps, and marking barrier points, coupon payment points |
| | **Calculating Option values and probabilities** |
| 2.4 | Calculate option values at maturity |
| 2.5 | Iterated calculating option and probability values for time steps backwards conditional on barrier positions |
| 2.6 | Defined lower boundary condition; *used values from grid with trigger for creating lower boundary condition for grid with no trigger* |
| 2.7 | Iterated for ABCD values using option values at maturity |
| 2.8 | Defined upper boundary condition |
| 2.9 | Dmax |
| 2.10 | Computed intermediate values, alpha and CN_S |
| 2.11 | Then finally calculated option values starting from upper boundary condition (mechanism shown in the pictures in later section) |
| 2.12 | Finally computed option value at S_0 from grid with no trigger adjusting if required |

**Formulae for computing parameters and stepwise flow:**

| | | |
|---|---|---|
| Lower boundary condition | A | 0 |
| | B | 1 |
| | C | 0 |
| | D | 0 |
| Upper boundary condition | A | 0 |
| | B | 1 |
| | C | 0 |
| | D | Formula in grid |

$$A_j V_{j-1}^i + B_j V_j^i + C_j V_{j+1}^i = D_j^i$$

where,

$$A_j = \frac{1}{4}\left(\sigma^2 j^2 - (r-q)j\right)$$

$$B_j = \left(-\frac{\sigma^2 j^2}{2} - \frac{r}{2} - \frac{1}{\Delta t}\right)$$

$$C_j = \frac{1}{4}\left(\sigma^2 j^2 + (r-q)j\right)$$

$$D_j = D_1 + D_2 + D_3$$

where,

$$D_1 = -V_{j-1}^{i+1}\left(\sigma^2 j^2 - (r-q)j\right)\frac{1}{4}$$

- $$D_2 = -V_j^{i+1}\left(\frac{1}{\Delta t} - \frac{\sigma^2 j^2}{2} - \frac{r}{2}\right)$$

$$D_3 = -V_{j+1}^{i+1}\left(\sigma^2 j^2 + (r-q)j\right)\frac{1}{4}$$

By solving the above equations starting at $j=0$ which is lower boundary condition, we get respective alphas & 'z's;

Now we solve for $V_j^i$ starting from jmax using upper boundary condition, & solve further down till $j=0$

-

$$\left\{\begin{array}{l}\alpha_0 = B_0 \\ z_0 = D_0\end{array}\right\} \rightarrow \left\{\begin{array}{l}\alpha_j = B_j - \dfrac{(A_j c_{j-1})}{\alpha_{j-1}} \\ z_j = D_j - \dfrac{(A_j z_{j-1})}{\alpha_{j-1}}\end{array}\right\} \rightarrow \left\{ V^i_{j_{max}} = \dfrac{z_{j_{max}}}{\alpha_{j_{max}}} \right\}$$

① ② ③

$$\downarrow$$

$$\left\{ V^i_j = \dfrac{z_j - c_j V^i_{j+1}}{\alpha_j} \right\}$$

④

*Figure illustrating step wise solving of equations for calculating option values:*

Each section is further modeled using two sub steps each,

1. Option values at barrier positions where present values of coupons are added as described in the grid picture before automatic redemption and the upper boundary value is discounted from next autocall coupon date to the current time.
2. Options values at rest of the grid

## Results:

We have noticed that after reaching to 1000 S_steps the convergence rate is very low, <0.015 dollars/100 steps, therefore the value is computed with 1000 S_steps and 398*4 t_steps

```
[{'S_steps': 1000,
  't_steps': 1592,
  'CN': 975.9356496051284,
  'Barrier Lambda': 0.9999999999999865}]
```

## Sensitivity analysis:

We have evaluated the value of options using a range of

- S_steps (100 – 1000) and
- t_steps 398 X (1 to 10)  = (1194 to 3980).
- We have also tried to analyze the change in value with respect to volatility while others kept constant.

**Volatility:-**

| Volatility | Option value |
|---|---|
| 0.33039 | 930.398115 |
| 0.25855 | 975.935650 |

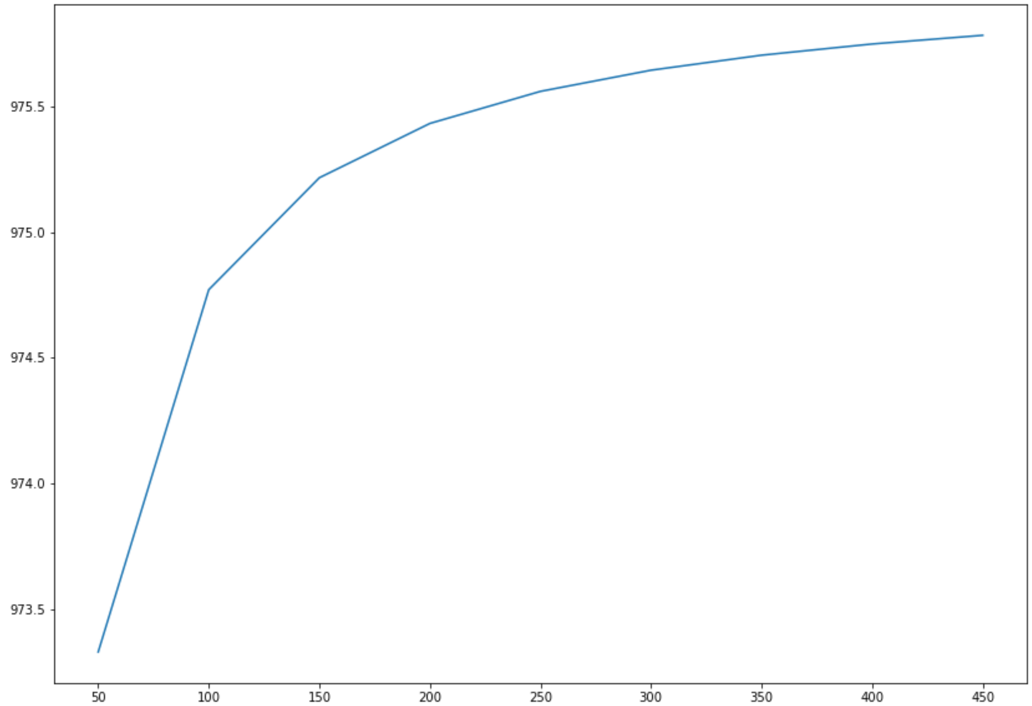
The implied volatility at 0.75 $S_0$ is 33.04% while at initial level i.e $S_0$ is 25.86% which resulted in a significant variation in the option value, therefore the range of option price could be between (930.398 to 975.936)

**Non- linearity & Convergence:-**

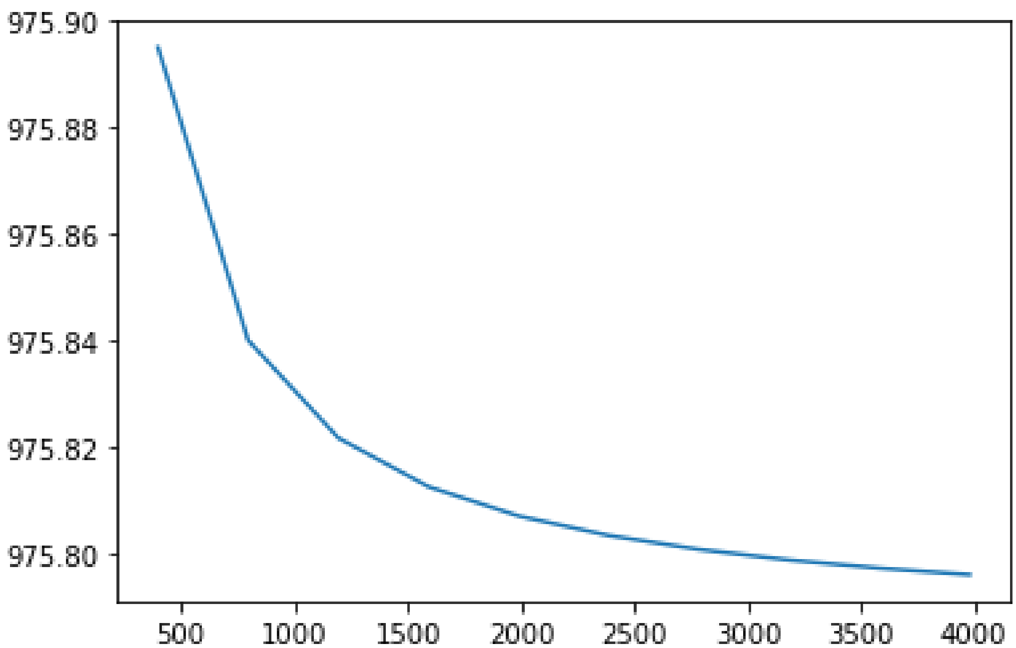
Since the model error is a function of square of delta S and delta t, therefore the convergence is quick.

We tried to fit the barrier on the nodes of the grid which reduced our non-linearity errors which is evident from the smooth convergence given below.

| | S_steps | t_steps | CN | Barrier Lambda |
|---|---|---|---|---|
| **0** | 50 | 1592 | 973.328215 | 1.0 |
| **1** | 100 | 1592 | 974.771551 | 1.0 |
| **2** | 150 | 1592 | 975.217500 | 1.0 |
| **3** | 200 | 1592 | 975.433954 | 1.0 |
| **4** | 250 | 1592 | 975.561737 | 1.0 |
| **5** | 300 | 1592 | 975.646054 | 1.0 |
| **6** | 350 | 1592 | 975.705853 | 1.0 |
| **7** | 400 | 1592 | 975.750467 | 1.0 |
| **8** | 450 | 1592 | 975.785029 | 1.0 |




***Given below is the smooth convergence observed with respect to t_steps:***



But to understand the effect of non-linearity issue we have selected grid which positions barrier off the grid nodes, thus resulting in the following nonlinear convergence.

| | S_steps | t_steps | CN | Barrier Lambda |
|---|---|---|---|---|
| **45** | 190 | 1592 | 975.400066 | 1.0 |
| **46** | 192 | 1592 | 978.544752 | 0.4 |
| **47** | 194 | 1592 | 976.400729 | 0.8 |
| **48** | 196 | 1592 | 979.528507 | 0.2 |
| **49** | 198 | 1592 | 977.437236 | 0.6 |