# I. Objective

The objective of this project is to price a Vanilla European call option with volatility term varying using different share dynamics namely, Heston model and Constant Elasticity of Variance (CVS). CVS is further augmented to calculate Monte Carlo estimates for the price of Vanilla Call option price and its standard deviation.

# II. Introduction

Stochastic volatility models are one approach to resolve a shortcoming of the non-stochastic models such as Black–Scholes and Cox–Ross–Rubinstein model which assume that the underlying volatility is constant over the life of the derivative, and unaffected by the changes in the price level of the underlying security. However, these models cannot explain long-observed features of the implied volatility surface such as volatility smile and skew, which indicate that implied volatility does tend to vary with respect to strike price and expiry. By assuming that the volatility of the underlying price is a stochastic process rather than a constant, it becomes possible to model derivatives more accurately.
For a stochastic volatility model, the constant volatility is replaced with a function that models the variance as Brownian motion, and its form depends on the SV model under study.

<u>Heston model</u>: it is a commonly used SV model, in which the randomness of the variance process varies as the square root of variance. It assumes that the variance is a random process that exhibits:- a tendency to revert towards a long-term mean “omega” at a rate theta, - exhibits a volatility proportional to the square root of its level, - and whose source of randomness is correlated (with correlation rho ) with the randomness of the underlying's price processes.

<u>Constant elasticity of variance (CEV) model</u>: it describes the relationship between volatility and price, introducing stochastic volatility. Conceptually, in some markets volatility rises when prices rise (e.g. commodities), so gamma >1. In other markets, volatility tends to rise as prices fall, modelled with gamma <1. Some argue that because the CEV model does not incorporate its own stochastic process for volatility, it is not truly a stochastic volatility model. Instead, they call it a local volatility model.

Other SVM’s are SABR (Stochastic Alpha, Beta, Rho), GARCH (Generalized Autoregressive Conditional Heteroskedasticity) and 3/2 model.

Once a particular SV model is chosen, it must be calibrated against existing market data. One popular technique is to use maximum likelihood estimation (MLE). Once the *calibration* has been performed, it is standard practice to re-calibrate the model periodically. An alternative to calibration is *statistical estimation*, thereby accounting for parameter uncertainty. Many frequentist and Bayesian methods have been proposed and implemented, typically for a subset of the abovementioned models.

In the next section we are going to understand and implement dynamics and pricing of Vanilla European call option using:

- Modification of the model presented by Heston, as proposed by Albrecher et al (2007)[1].
- CVS; while the CVS model assumes that volatility is continuous function of time and share price, for simplicity for the purpose of this submission, volatility is assumed to be constant over each simulation period.
- Statistical estimation using Monte Carlo
- Black Scholes for analytical price calculation.

# III. Algorithm

Six primary factors influence options pricing: the underlying price history (not only terminal value), strike price, time until expiration, volatility, interest rates and dividends.

**TASK 1: Using a simple Fourier pricing technique (using $N$= 100 intervals and using an effective upper bound of integration of 30), price a vanilla call option assuming that the underlying share follows the Heston model dynamics. Use the parameter values from the previous section (submission 1), as well as the following parameter values:**

- **$v0$=0.06**
- **κ= 9**
- **θ= 0.06**
- **$\rho$ = −0.4**

**We will now simulate a share price path.**

Heston was able to show that the characteristic function of log ($S_T$), $s_T$, is given by:

$\phi_{sT} =$ exp $((\tau;u)+D(\tau;u)v_t+iulog(S_t))$,

Where

$C(\tau;u) = ri\tau u+\theta\kappa\ [\tau x - \frac{1}{a}\ (\frac{1-ge^{d\tau}}{1-g}\ )]$

$(\tau;u) = (\frac{1-e^{d\tau}}{1-ge^{d\tau}}\ )\ x_-$

And

$g = \frac{x_-}{x_+}$

$x_\pm = \frac{b \pm d}{2a}$

d = $\sqrt{b^2 - 4ac}$

c = $- \frac{u^2+ui}{2}$

$b = \kappa - \rho\sigma iu$

a = $\frac{\sigma^2}{2}$

Using Gil-Palaez, the price of a call is given by:

$$c = S_0 \mathbb{Q}^S[S_T > K] - e^{-rT} K \mathbb{Q}[S_T > K]$$

$$= S_0 \left( \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \frac{Im\left[e^{-it \ln K} \varphi_{M_2}(t)\right]}{t} dt \right)$$
$$-e^{-rT} K \left( \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \frac{Im\left[e^{-it \ln K} \varphi_{M_1}(t)\right]}{t} dt \right),$$

Where,

$\varphi_{M1}(t) = \phi_{sT}(t)$

$\varphi_{M2}(t) = \frac{\phi sT(u-i)}{\phi sT(-i)}$

The python code to implement the above functions are be as follows:

- Import the libraries to be used. Four important libraries are used in this project are as follows:
  - Numpy
  - Matplotlib
  - Scipy
  - Random
- Define and declare all the variables that will be used in all four tasks, these are information about share, option and approximation. Values of these variables are given in problem statement of submission 1 and 2.
- Set up the functions as defined by Heston for the characteristic function as adj_char.
- Create few variables to vectorize the code
- Calculate an estimate for the integrals
- Calculate the Fourier estimate of the Vanilla call price
- For comparison reason, call price is calculated under Black-Scholes assumptions (volatility equals sigma)
- To illustrate the power of the characteristic pricing method, plot the integrands for the two integrals.

**TASK 2: Assume that $(ti,ti+1)=\sigma(Sti)$^$\gamma-1$, where σ= 0.3 and $\gamma$ = 0.75 (gamma). Using the formula below, simulate paths for the underlying share using sample sizes of 1000, 2000, …, 50000. Do monthly simulations for a period of a year.**
**We can simulate the next step in a share price path using the following formula:**

$$S_{t_{i+1}} = S_{t_i} e^{\left(r - \frac{\sigma^2(t_i, t_{i+1})}{2}\right)(t_{i+1} - t_i) + \sigma(t_i, t_{i+1})\sqrt{t_{i+1} - t_i} Z},$$

**Where $S_{ti}$ is the share price at time $ti$,σ($ti$,$ti$+1) is the volatility for the period [$ti$,$ti$+1],$r$ is the risk-free interest rate, and $Z$~$N$(0,1).**
**Not that we are attempting to run simulations using the CEV model. However, while the CEV model assumes that volatility is a continuous function of time and share price, we are making a simplifying assumption that volatility is constant over each simulation period.**

- Simulate paths for the underlying share using sample sizes of 1000, 2000, ..., 50000", for a period of a year on monthly basis using the following formula:

  S(t_i+1) = S(t_i)*exp((r - sigma^2(t_i,t_i+1)/2)(t_i+1-t_i)+Z*sigma(t_i,t_i+1)*sqrt(t_i+1-t_i))

Where:

- S(t_i) is the share price at time t_i,

- sigma(t_i,t_i+1) is volatility for the period [t_i,t_i+1]

- r is the risk-free interest rate

- Z~N(0,1),

- Set up functions to generate:
  - Volatility- <u>localvols</u> (given by sigma*(S(t_i)^(gamma-1)) as defined by CEV model.
  - Monthly share price-<u>price_path</u> (given by S_prev*np.exp((R - localvols[i-1]**2/2)*dT + localvols[i-1]*np.sqrt(dT)*Z[i-1])
  - Discounted payoff/price of a vanilla European call option-call_payoffs (given by np.exp(-R*T)*np.maximum((S_term-K),0))

- Pre-allocating space for the array of local volatilities, share price path and option prices and standard deviation

- Calculate share price path and option payoff at terminal share price for each sample for i in range(1,51) looping over samples of different sizes - 1000 to 50000 and for j in range(i*1000) iterations generating monthly random values for stock over 1 year from a standard normal distribution.

**TASK 3: Augment your code in part 2 to calculate Monte Carlo estimates, as well as the standard deviations for these estimates, for the price of a vanilla call option (with the same strike term as in Submission 1).**

- Calculate price of the vanilla European call option Monte Carlo estimates and standard deviations of these estimates using [np.mean(option_payoffs[i-1]) for i in range(1,51)], [np.std(option_payoffs[i-1])/np.sqrt(i*1000) for the generated values in the above step.

**TASK 4. Plot the Monte Carlo estimates generated in part 3 with respect to sample size, as well as three standard deviation error bounds around these estimates.**

- Calculate the analytical call prices for comparison. However, the Black-Scholes (BS) closed-form solution for analytical option price assumes fixed volatility so the average local volatility from each sample of CEV is assumed as the fixed volatility when applying the BS solution.

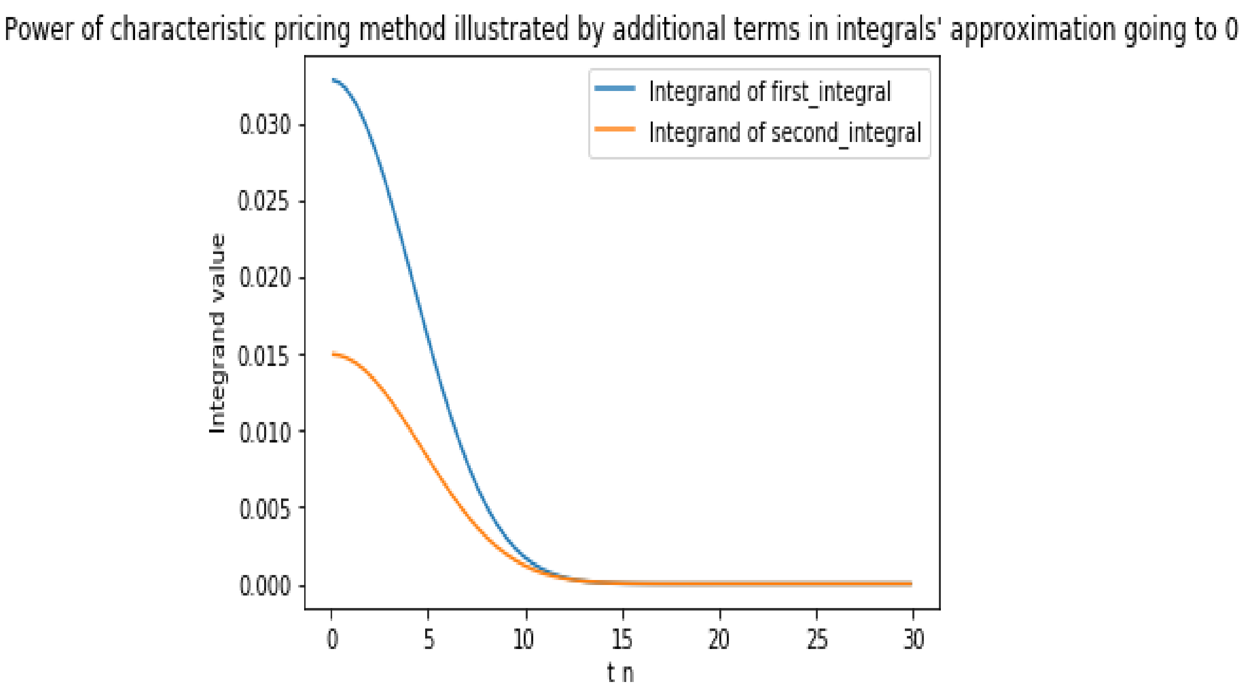
- Plot MC Estimates and three standard deviation error bounds around these estimates using Matplotlib.

# IV. Results

## Task 1

- The Fourier estimate of the Vanilla call price was calculated using modified the modified Heston model characteristic function; Fourier Call Price = 13.734895692109077
- For comparison reason, call price is calculated under Black-Scholes assumptions (volatility equals sigma); Analytical Call Price = 15.711312547892973
- The plot of the integrands for the two integrals is shown in figure 1 below to show the power of the characteristic pricing method in how quickly the additional terms in integral approximation go to 0.

Figure 1




## Task 2

Output of Task 2 depends on following functions:

- Function (CEV_local_volatility) for generating CEV Model according to dynamics given in problem.

- Function (share_path) for generating path of share prices using volatility according to given CEV model
- Function (call_payoff_dis) for calculating discounted payoff of a vanilla European call option.

All these functions help in creating discounted payoff matrix of call option (option_payoff) and further help in accomplishing Task 3 and 4.

## Task 3

Monte Carlo estimates of vanilla European call option and standard deviations of these estimates is done with help of option_payoff matrix created in Task 2. Results are stored in following two variables:

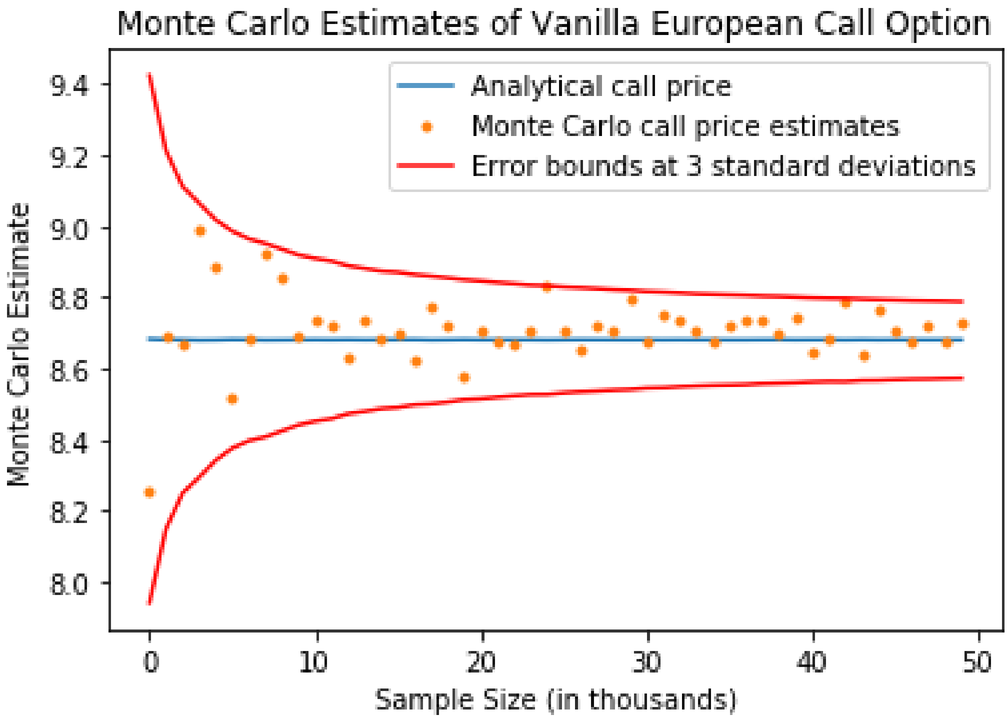
- option_price_estimates variable calculates the 50 mean values of option's payoff from 50 samples run in Task 2
- option_price_stdevs variable calculates the 50 standard deviation of option's payoff from 50 samples run in Task 2

These two variables will be used in Task 4 for plotting Monte Carlo estimates along with respective standard deviation.

## Task 4

- Monte Carlo estimates of Vanilla European call option is plotted below (in Figure 2) with increasing sample size – from 1000 to 50000. For comparison, the analytical price of the option is plotted as a horizontal line and error bounds (in red) at 3 standard deviations from the analytical value.

**Figure 1**




- The analytical price of the option is approximately 8.68 for average volatility of approximately 0.09, and the estimated price converges to approximately the same value.

## V. Bibliography


- Albrecher, H., Mayer, P., Schoutens, W. and Tistaert, J. (2007). “The Little Heston Trap”, Wilmott (1): 83–92.
- Heston, S. L. (1993). “A Closed-form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options”, The Review of Financial Studies 6(2): 327–343.
- Hsu, Y.-L., Lin, T. and Lee, C. (2008). “Constant Elasticity of Variance (cev) Option Pricing Model: Integration and Detailed Derivation”, Mathematics and Computers in Simulation 79(1): 60–71.
- https://en.wikipedia.org/wiki/Stochastic_volatility
- https://convention.vse.cz/media/Collection_2011_Volume_01.pdf#page=143/ Limitations of the Black-Scholes Model-Dean Teneng
- Carr, P. and Madan, D. (1999). “Option Valuation Using the Fast Fourier Transform”, Journal of Computational Finance 2(4): 61–73.
- Cooley, J.W. and Tukey, J.W. (1965). “An Algorithm for the Machine Calculation of Complex Fourier Series”, Mathematics of Computation 19(90): 297–301.