The COS Method:

Using Markov's Inequality to Define a Truncation Range

Master of Quantitative Finance

Tom Haynes (14392144)

March 27, 2023

$$\int_{\mathbb{R}} v(x) f(x)\, dx$$

Table of Contents

The COS Method
1 Introduction

- The COS method is a Fourier option pricing method.
- It is one of the fastest known methods for pricing options quickly and accurately.
  - It also allows us to compute the greeks at no additional computational cost.
- Calculating options quickly, often for a number of strikes at once, is a very important problem in computational finance

> *"...stock price models are typically calibrated to gives prices of liquid call and put options by minimizing the mean-square-error between model prices and given market prices. During the optimization routine, model prices of call and put options need to be evaluated very often for different model parameters."[1]*

The Basic Idea

The essence of the problem the COS method solves is fairly straightforward given the following conditions:

Conditions

- **The stochastic component (underling price) is modelled by random variable X where in which the probability density function f is unknown.**
- **We do have a characteristic function of f explicitly, denoted φ.**
- **We have an option payoff $v(X)$. Then the integral we are trying to compute is the following as quickly as possible:**

$$\int_{\mathbb{R}} v(x) f(x) dx \tag{1}$$

Table of Contents

The COS Method and Density Recovery:

- *The idea is that we replace the density by the Fourier Cosine Series Expansion - where the series coefficients A_k are directly related to φ*

The density and characteristic function form a Fourier pair, that is:

$$\varphi_X(u) = \int_{\mathbb{R}} e^{(iyu)} f_X(y)dy \xrightarrow{\mathscr{F}} f_x(y) = \frac{1}{2\pi}\int_{\mathbb{R}} e^{-iuy}\varphi_x(u)du \tag{2}$$

Now, moving forward with the Fourier Cosine Expansion on $[-1, 1]$, giving us:

$$g(\theta) = {\sum_{k=0}^{\infty}}' A_k \cos(k\pi\theta) + \sum_{k=1}^{\infty} B_k \sin(k\pi\theta) \tag{3}$$

With A_k and B_k defined as follows:

$$A_k = \int_{-1}^{1} g(\theta)\cos(k\pi\theta)d\theta, \quad B_k = \int_{-1}^{1} g(\theta)\sin(k\pi\theta)d\theta \tag{4}$$

Setting $B_k = 0$ we obtain an even function:

$$\bar{g}(\theta) = \begin{cases} g(\theta), & \theta \geq 0 \\ g(-\theta), & \theta < 0 \end{cases} \tag{5}$$

From here we do quite a complicated derivation - omitted due to the technical level. What is important is that we use a change of variables to find the Fourier Cosine expansion defined on a different interval: $[a, b] \in \mathbb{R}$.

Assuming this a and b are chosen such that the truncated integral derived from this bound does approximate its infinite counterpart well enough, we arrive essentially arrive at the following:

$$\varphi_X(u) = \int_{\mathbb{R}} e^{iuy} dy \approx \int_a^b e^{iuy} dy \tag{6}$$

Taking the real part of Euler's formula: $\Re\{e^{iu}\} = \cos(u)$

Density Recovery - Efficiency

Where this model really excels is in the efficiency of the density recovery - below, using the cumulants method (choosing $n_c = 4$), we can reach the PDF of the normal, with an error in the area of machine precision, with 64 expansion terms:

Problems - Truncation Ranges

2 Fang and Osterlee:

In this program, we are trying to rectify 3 main issues, those which arise from Fang and Osterlee's cumulants based definition of a truncation range.

1. Ambiguity surrounding user choice of cumulants.
2. Models where in which the COS method diverges (MJD).
3. Options with extreme moneyness.

Table of Contents

3 Solutions

Solving These Issues

There are two main issues I am trying to get at here - those described by Junike and Pankrashkin [], and those I have previously described. Discussion around the truncation range is not new, there are also moments where this method diverges - as in Junike and Pankraskin's demonstration with an MJD model

The idea is to incorporate Markov's inequality into the derivation of the COS method, allowing us to select the truncation range based on the

Problems - Inefficiency

Table of Contents

4 Implementation

Extension - Automatic Truncation of Moment Calculation