UNIVERSITY OF
COPENHAGEN

October 1, 2024

*FinKont2 - Continuous Time Finance 2*

# Option Pricing and Asset Allocation under Heston Model


***Student :***
Théophile Schmutz, s23359

***Teacher :***
Rolf Poulsen

# Contents

Chapter 1

# Characteristic functions in the Heston model

First, we derive a short reminder of the proof of the PDEs for the price $\mathcal{V}$ of a call option with strike $K$, maturity date $T$, on the underlying asset $S$. We are thus considering the $T$-claim,

$$f(S_T) = \max(S_T - K, 0). \tag{1.1}$$

**Heston model** We assume that the stock is ruled by the Heston model under real world and martingale measures. Thus, skipping the reasoning from real-world measures $\mathbb{P} = (\mathbb{P}_1, \mathbb{P}_2)$ to martingale measures $\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$, using the Cholesky decomposition and dsanov's Theorem,

$$dS_t = rS_t dt + \sqrt{v_t} S_t \left( \rho dW_t^{\mathbb{Q}_1} + \sqrt{1-\rho^2} dW_t^{\mathbb{Q}_2} \right), \tag{1.2}$$

$$dv_t = \tilde{\kappa} \left( \tilde{\theta} - v_t \right) dt + \sigma \sqrt{v_t} dW_t^{\mathbb{Q}_1}. \tag{1.3}$$

With $W^{\mathbb{Q}_1}$ and $W^{\mathbb{Q}_2}$ orthogonal Wiener processes under martingale measures.

Please note that this model is incomplete. Therefore there exists an infinity of martingale measures $\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$. These martingale measures are parameterised by a $\lambda$ which should be fixed according to real market price. Therefore, the parameters $\kappa$ and $\theta$ are defined by the real world model and the $\lambda$. We do not write this latter dependence for clarity, but we should keep in mind the existence of a such $\lambda$. Specifically, $\tilde{\kappa} = \kappa + \lambda$, $\tilde{\theta} = \kappa\theta/(\kappa + \lambda)$, so that the constants without tildes are the parameters under the real world measures. Thus, 1.3 can be rewritten as,

$$dv_t = (\kappa(\theta - v_t) - \lambda v_t) dt + \sigma \sqrt{v_t} dW_t^{\mathbb{Q}_1}. \tag{1.4}$$

**Price of the call** We have two ways of deriving the PDE to price $\mathcal{V}$ this option. Either we construct a self-financing portfolio that hedges the value of the call option $\mathcal{V}$, or, as we are about to do, we use the Feynman-Kac theorem.

Hence, let $W^{\mathbb{Q}} \overset{\text{def}}{=} (W^{\mathbb{Q}_1}, W^{\mathbb{Q}_2})^\top$, and the process $X \overset{\text{def}}{=} (S, v)^\top$ whose diffusion is thus written as, $X_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t^{\mathbb{Q}}$, with using 1.2 and 1.4,

$$\mu\left(t, (S,v)^\top\right) = \begin{pmatrix} rS \\ \kappa(\theta - v) - \lambda v \end{pmatrix}, \quad \sigma\left(t, (S,v)^\top\right) = \begin{pmatrix} \sqrt{v}S\rho & \sqrt{v}S\sqrt{1-\rho^2} \\ \sigma\sqrt{v} & 0 \end{pmatrix},$$

We recall that the $\lambda$ is the constant which parametrised the martingale measure $\mathbb{Q}$. Using the Feynman-Kac theorem (Proposition 5.6 in Björk 2019) for the payoff 1.1 $f(X_T)$ and the deterministic interest rate $r$,

$$\begin{aligned} 0 &= \mathcal{V}_t + \mu\left(t, S, v\right)^\top \cdot \nabla\mathcal{V} + \frac{1}{2}\operatorname{tr}\left[\sigma\left(t, S, v\right)^\top \cdot \mathbf{H}_{\mathcal{V}} \cdot \sigma\left(t, S, v\right)\right] - r\mathcal{V}, \\ f(S, v) &= \mathcal{V}(T, S, v), \quad \forall (S, v) \in ]0, +\infty[^2. \end{aligned}$$

With, $\mathcal{V}_t$ the partial time derivative, $\nabla\mathcal{V}$ the spatial gradient, $\mathbf{H}_{\mathcal{V}}$ Hessian matrix containing double derivatives w.r.t. spatial coordinates, and $\operatorname{tr}$ the trace. After calculation, using the Schwarz's theorem,

$$\frac{1}{2}\operatorname{tr}\left[\sigma\left(t, S, v\right)^\top \mathbf{H}_{\mathcal{V}} \sigma\left(t, S, v\right)\right] = \frac{1}{2}vS^2\mathcal{V}_{SS} + vs\sigma\rho\mathcal{V}_{vS} + \frac{1}{2}\sigma^2 v\mathcal{V}_{vv},$$

which leads us to,

$$\begin{aligned} -\mathcal{V}_t + r\mathcal{V} &= \frac{1}{2}vS^2\mathcal{V}_{SS} + v\rho\sigma S\mathcal{V}_{vS} + \frac{1}{2}v\sigma^2\mathcal{V}_{vv} + rS\mathcal{V}_S + \left(\kappa(\theta - v) - \lambda v\right)\mathcal{V}_v, \\ \mathcal{V}(T, S, v) &= f(S, v), \quad \forall (S, v) \in ]0, +\infty[^2. \end{aligned}$$

**Bijective change of variable** We make a bijective change of variable, $x \overset{\text{def}}{=} \ln(S)$ to get a PDE for $\mathcal{V}(t, x, v)$. As this bijection does not depend on time, $\mathcal{V}_t(t, S, v) = \mathcal{V}_t(t, x, v)$. In this respect, $\mathcal{V}_v(t, S, v) = \mathcal{V}_v(t, x, v)$, and then, $\mathcal{V}_{vv}(t, S, v) = \mathcal{V}_{vv}(t, x, v)$. Regarding the derivatives with respect to $S$,

$$\mathcal{V}_S(t, S, v) = \frac{\partial\mathcal{V}(t, x, v)}{\partial S} = \frac{\partial x}{\partial S}\mathcal{V}_x(t, x, v) = \frac{1}{S}\mathcal{V}_x(t, x, v).$$

In this respect,

$$\mathcal{V}_{SS}(t, S, v) = \frac{\partial\mathcal{V}_S(t, x, v)}{\partial S} = \frac{1}{S^2}\left(\mathcal{V}_{xx}(t, x, v) - \mathcal{V}_x(t, x, v)\right).$$

Moreover, using the Schwarz's theorem,

$$\mathcal{V}_{vS}(t, S, v) = \frac{\partial}{\partial v}\left(\mathcal{V}_S(t, S, v)\right) = \frac{1}{S}\mathcal{V}_{vx}(t, x, v) = \frac{1}{S}\mathcal{V}_{xv}(t, x, v).$$

In a nutshell,

$$\begin{aligned} -\mathcal{V}_t + r\mathcal{V} &= \frac{1}{2}v\mathcal{V}_{xx} - \frac{1}{2}v\mathcal{V}_x + v\rho\sigma\mathcal{V}_{xv} + \frac{1}{2}v\sigma^2\mathcal{V}_{vv} + r\mathcal{V}_x + \left(\kappa(\theta - v) - \lambda v\right)\mathcal{V}_v, \\ \mathcal{V}(T, x, v) &= f(e^x, v) = (e^x - K)^+, \quad \forall (x, v) \in \mathbb{R}\times]0, +\infty[. \end{aligned}$$

Which brings us to the set-up of the Hand-In. We use the following ansatz:

$$\mathcal{V}(t,x,v) = e^{x} Q_1(x,v,t) - K e^{-r(T-t)} Q_2(x,v,t), \tag{1.5}$$

after inserting this ansatz into the latter PDE with terminal condition, we get two PDEs for $Q_1$ and $Q_2$. We could not solve that PDEs in closed-form, but verified that they have a convenient probabilistic representation. In particular, they can be seen as, for $j \in \{1,2\}$,

$$Q_j\Big(x,v,t;\ln(K)\Big) = \tilde{\mathbb{P}}_j\Big(x(T) \geq \ln(K)\Big| x(t)=x, v(t)=v\Big), \tag{1.6}$$

where $\tilde{\mathbb{P}}_1$ and $\tilde{\mathbb{P}}_2$ are some "adjusted" probability measures. Using the Fourier Transform machinery, we can compute $Q_j(x,v,t;\ln(K))$ as,

$$Q_j\left(x,v,t;\ln(K)\right) = \frac{1}{2} + \frac{1}{\pi}\int_0^{+\infty} \mathfrak{Re}\left(\frac{e^{-iu\ln(K)}\Psi^j_{x(T)}\left(x,v,t;u\right)}{iu}\right) du, \tag{1.7}$$

where $\Psi^j_{x(T)}\left(x,v,t;u\right)$ is the Fourier transform of the probability measure $\mu_{x(T)}$ under $\mathbb{P}_j$ induced by $x(T)$ knowing $x(t)=x$ and $v(t)=v$,

$$\Psi^j_{x(T)} = \int_{-\infty}^{+\infty} e^{iuy}\mu_{x(T)}(dy) = \mathbb{E}^{\mathbb{P}_j}_{t,x,v}\left[e^{iux(T)}\right],$$

where the subscript in the expected value denotes knowing $x(t)=x, v(t)=v$. This characteristic functions solve the following PDEs, for $j \in \{1,2\}$

$$0 = \frac{\partial \Psi_j}{\partial t} + (r+u_j v)\frac{\partial \Psi_j}{\partial x} + \frac{1}{2}v\frac{\partial^2 \Psi_j}{\partial x^2} + \rho\sigma v\frac{\partial^2 \Psi_j}{\partial v \partial x} + \frac{1}{2}\sigma^2 v\frac{\partial^2 \Psi_j}{\partial v^2} + (a-b_j v)\frac{\partial \Psi_j}{\partial v}, \tag{1.8a}$$

$$e^{iux} = \Psi^j_{x(T)}(x,v,T;u), \quad \forall (x,v) \in \mathbb{R}\times\,]0,+\infty[\,. \tag{1.8b}$$

Where we have written $\Psi_j \overset{\text{def}}{=} \Psi^j_{x(T)}(x,v,t;u)$ for readability matter.

## 1.1 Changing variables and guessing the form of solution

Let the change of variable $\tau = T - t$. Hence,

$$\frac{\partial \Psi_j}{\partial t} = \frac{\partial \tau}{\partial t}\frac{\partial \Psi_j}{\partial \tau} = -\frac{\partial \Psi_j}{\partial \tau},$$

using the deterministic chain rule. As $\tau$ only depends on $t$, all the partial derivatives with respect to the other variables don't change. Moreover, seeing $t \mapsto \Psi^j_{x(T)}(x,v,t;u)$

as a function of $t$ or as a function of $\tau$ is equivalent as it is simply a translation in time. Therefore, from 1.8a and 1.8b,

$$\begin{aligned} 0 &= -\frac{\partial \Psi_j}{\partial \tau} + (r + u_j v)\frac{\partial \Psi_j}{\partial x} + \frac{1}{2}v\frac{\partial^2 \Psi_j}{\partial x^2} + \rho\sigma v\frac{\partial^2 \Psi_j}{\partial v \partial x} + \frac{1}{2}\sigma^2 v\frac{\partial^2 \Psi_j}{\partial v^2} \\ &\quad + (a - b_j v)\frac{\partial \Psi_j}{\partial v}, \qquad (1.1a) \\ e^{iux} &= \Psi^j_{x(T)}(x, v, 0; u), \quad \forall (x, v) \in \mathbb{R} \times ]0, +\infty[\,. \qquad (1.1b) \end{aligned}$$

where this time $\Psi_j \overset{\text{def}}{=} \Psi^j_{x(T)}(x, v, \tau; u)$.

Let assume that the solution of the latter equations 1.1a, 1.1b has the form,

$$\Psi^j_{x(T)}(x, v, \tau; u) = \exp\left\{C_j(\tau; u) + D_j(\tau; u)v + iux\right\}, \quad j \in \{1, 2\}.$$

Indeed,

$$\begin{aligned} \frac{\partial \Psi_j}{\partial x} &= iu\ \Psi_j, & \frac{\partial^2 \Psi_j}{\partial x^2} &= -u^2 \Psi_j, \\ \frac{\partial \Psi_j}{\partial v} &= D_j \Psi_j, & \frac{\partial^2 \Psi_j}{\partial v^2} &= D_j^2 \Psi_j, \\ \frac{\partial^2 \Psi_j}{\partial v \partial x} &= iu\ D_j \Psi_j, & \frac{\partial \Psi_j}{\partial \tau} &= \left(\frac{\partial C_j}{\partial \tau} + v\frac{\partial D_j}{\partial \tau}\right)\Psi_j, \end{aligned}$$

Hence, plugging these equations into 1.1a, 1.1b, we get,

$$\begin{aligned} 0 &= -\left(\frac{\partial C_j}{\partial \tau} + v\frac{\partial D_j}{\partial \tau}\right)\Psi_j + i\left(r + u_j v\right)u\Psi_j - \frac{1}{2}vu^2\Psi_j + i\rho\sigma u v D_j \Psi_j + \frac{1}{2}\sigma^2 v D_j^2 \Psi_j \\ &\quad (a - b_j v)D_j \Psi_j \\ e^{iux} &= \exp\{C_j(0; u) + D_j(0; u)v + iux\} \end{aligned}$$

Which can be re-written as,

$$\begin{aligned} 0 = \Psi_j &\left(-\frac{\partial C_j}{\partial \tau} + iru + aD_j\right) \\ &+ \Psi_j v\left(-\frac{\partial D_j}{\partial \tau} + iu_j u - \frac{1}{2}u^2 + i\rho\sigma u D_j + \frac{1}{2}\sigma^2 D_j^2 - b_j D_j\right) \end{aligned}$$

## 1.2 Transforming the PDE to a system of ODEs and solving it

### 1.2.1 Affine term structure

Assuming $\Psi_j$ is non null, we can rewrite the latter equation as,

$$0 = \left(-\frac{\partial C_j}{\partial \tau} + iru + aD_j\right) + v\left(-\frac{\partial D_j}{\partial \tau} + iu_j u - \frac{1}{2}u^2 + i\rho\sigma u D_j + \frac{1}{2}\sigma^2 D_j^2 - b_j D_j\right)$$

This equation holds for all $\tau$ and $v$, so let us consider it for a fixed choice of $\tau$. Since the equation holds for all values of $v$ the coefficient of $v$ must be equal to zero. Thus we have the equation,

$$-\frac{\partial D_j}{\partial \tau} + iu_j u - \frac{1}{2}u^2 + i\rho\sigma u D_j + \frac{1}{2}\sigma^2 D_j^2 - b_j D_j = 0. \tag{1.1}$$

Since the $v$-term in is zero we see that the other term must also vanish, giving us the equation,

$$-\frac{\partial C_j}{\partial \tau} + iru + aD_j = 0. \tag{1.2}$$

In particular, at $\tau = 0$ we then have to equations,

$$\frac{\partial C_j}{\partial \tau}(0;u) = iru + aD_j(0;u), \tag{1.3}$$

$$\frac{\partial D_j}{\partial \tau}(0;u) + (b_j - i\rho\sigma u)\, D_j(0;u) - \frac{1}{2}\sigma^2 D_j^2(0;u) = iu_j u - \frac{1}{2}u^2. \tag{1.4}$$

### 1.2.2 Linear ODE

We have written in that way so that we can see that 1.4 fully determine $D_j$ and 1.3 provide the link between $C_j$ and $D_j$. The equation 1.4 is not an ODE as there is a square term. However, we can assume,

$$D_j(\tau;u) = -\frac{2}{\sigma^2}\frac{\frac{\partial E_j(\tau;u)}{\partial \tau}}{E_j(\tau;u)}, \tag{1.5}$$

so that the equation in $D_j$ 1.1 becomes a linear ODE of the second order. First,

$$\frac{\partial D_j}{\partial \tau} = \frac{2}{\sigma^2}\frac{\frac{\partial^2 E_j}{\partial \tau^2} - \left(\frac{\partial E_j}{\partial \tau}\right)^2}{E_j^2}, \tag{1.6}$$

where we have omit the argument $(\tau;u)$ for readability. Then, rewriting the equation 1.4 accordingly,

$$\begin{aligned} 0 &= \frac{2}{\sigma^2}\frac{\frac{\partial^2 E_j}{\partial \tau^2} - \left(\frac{\partial E_j}{\partial \tau}\right)^2}{E_j^2} + \left(u_j u i - \frac{1}{2}u^2\right) - \frac{2\rho u}{\sigma} i\frac{\frac{\partial E_j}{\partial \tau}}{E_j} + \frac{1}{2}\sigma^2\left(\frac{2}{\sigma}\frac{\frac{\partial E_j}{\partial \tau}}{E_j}\right)^2 \\ &\quad + 2\frac{b_j}{\sigma^2}\frac{\frac{\partial E_j}{\partial \tau}}{E_j}, \\ &= \frac{2}{\sigma^2}\frac{\partial^2 E_j}{\partial \tau^2} + \left(u_j u i - \frac{1}{2}u^2\right)E_j + \left(-2i\frac{\rho u}{\sigma} + 2\frac{b_j}{\sigma^2}\right)\frac{\partial E_j}{\partial \tau}, \end{aligned}$$

Thus,

$$\boxed{\frac{\partial^2 E_j}{\partial \tau^2} - (\sigma\rho u i - b_j)\frac{\partial E_j}{\partial \tau} + \frac{\sigma^2}{2}\left(u_j u i - \frac{1}{2}u^2\right)E_j = 0.} \tag{1.7}$$

### 1.2.3 Solving this linear ODE

We have now a complex ODE which has the following characteristic equation in $r$,

$$r^2 - (\sigma\rho ui - b_j)\, r + \frac{\sigma^2}{2}\left(u_j ui - \frac{1}{2}u^2\right) = 0.$$

We can easily see that the two roots of the latter polynomial are,

$$x_{j,+} = \frac{\rho\sigma ui - b_j + d_j}{2}, \quad x_{j,-} = \frac{\rho\sigma ui - b_j - d_j}{2},$$

with

$$d_j = \sqrt{(\rho\sigma ui - b_j)^2 - \frac{\sigma^2}{2}(u_j ui - \frac{1}{2}u^2)}.$$

Then, a general solution of 1.7 is,

$$E_j(\tau; u) = A_j e^{\tau x_{j,+}} + B_j e^{\tau x_{j,-}},$$

with $A_j$ and $B_j$ complex constants. We have two unknown constant, so we need two equations. Hence, we use the limit conditions at $\tau = 0$,

$$E_j(0; u) = A_j + B_j, \quad \frac{E_j(0; u)}{\partial\tau} = x_{j,+} \cdot A_j + x_{j,-} \cdot B_j = 0.$$

We then have two equations for two unknown, we then have a Cramer system, providing us the existence and uniqueness of $A_j$ and $B_j$ ($2\pi$ modulo the argument). Denoting, $g_j = x_{j,-}/x_{j,+}$ and remarking $x_{j,+} - x_{j,-} = d_j$, we obtain:

$$A_j = g_j \frac{E_j(0; u) - \frac{1}{x_{j,-}}\frac{\partial E_j(0;u)}{\partial\tau}}{g_j - 1} = \frac{g_j E_j(0; u)}{g_j - 1},$$
$$B_j = E_j(0; u) - A_j = -\frac{E_j(0; u)}{g_j - 1}.$$

In a nutshell,

$$\boxed{A_j = \frac{g_j E_j(0; u)}{g_j - 1}, \quad B_j = -\frac{E_j(0; u)}{g_j - 1}.} \tag{1.8}$$

Plugging the expression of $A_j$ and $B_j$ into the linear ODE in $E_j$, we get,

$$\boxed{E_j(\tau; u) = \frac{E_j(0; u)}{g_j - 1}\left(g_j e^{\tau x_{j,+}} - e^{\tau x_{j,-}}\right)} \tag{1.9}$$

### 1.2.4 Moving back to our problem

We can now plug this into the equation linking $D_j$ and $E_j$ 1.6. First let's compute the derivative of $E_j$ with respect to $\tau$,

$$\begin{aligned}\frac{\partial E_j}{\partial \tau} &= \frac{E_j(0;u)}{g_j - 1}\left(g_j x_{j,+} e^{\tau x_{j,+}} - x_{j,-} e^{\tau x_{j,-}}\right) \\ &= \frac{E_j(0;u)}{g_j - 1} e^{\tau x_{j,-}} g_j x_{j,-}\left(e^{\tau d_j} - 1\right) \\ \frac{\partial E_j}{\partial \tau} &= E_j(0;u)\frac{g_j e^{\tau x_{j,-}}}{d_j}\left(e^{\tau d_j} - 1\right)\end{aligned}$$

Then, using 1.6, we get,

$$\begin{aligned}D_j &= -\frac{2}{\sigma^2}\frac{g_j}{g_j - 1} d_j \frac{e^{\tau d_j} - 1}{g_j e^{\tau d_j} - 1} \\ &= \frac{2}{\sigma^2}\frac{x_{j,-}}{x_{j,-} - x_{j,+}} d_j \frac{1 - e^{\tau d_j}}{1 - g_j e^{\tau d_j}} \\ &= -\frac{2}{\sigma^2} x_{j,-} \frac{1 - e^{\tau d_j}}{1 - g_j e^{\tau d_j}}\end{aligned}$$

In a nutshell,

$$\boxed{D_j(\tau;u) = \frac{d_j + b_j - \rho\sigma u i}{\sigma^2}\frac{1 - e^{\tau d_j}}{1 - g_j e^{\tau d_j}}} \tag{1.10}$$

We can now determine $C_j$, as 1.2 fully characterised by $C_j$ by $D_j$. Hence, integrating 1.2 with respect to $\tau$ between 0 and a fixed $\tau$,

$$\begin{aligned}C_j(\tau;u) &= rui\tau + a\int_0^\tau D_j(\tilde{\tau};u)d\tilde{\tau}, \\ &= rui\tau + a\frac{d_j + b_j - \rho\sigma u i}{\sigma^2}\int_0^\tau \frac{1 - e^{\tilde{\tau} d_j}}{1 - g_j e^{\tilde{\tau} d_j}} d\tilde{\tau},\end{aligned}$$

Which leads us to the closed form of $C_j$,

$$\boxed{C_j(\tau;u) = rui\tau + \frac{a}{\sigma^2}\left((d_j + b_j - \rho\sigma u i)\tau - 2\ln\left[\frac{1 - g_j e^{\tau d_j}}{1 - g_j}\right]\right)} \tag{1.11}$$

## 1.3 Numerical implementation of characteristic functions

### 1.3.1 Results

From the last equations, we can compute $\Psi_j$ using 1.11 and 1.10 and the absatz of $\Psi_j$.

We compute the characteristic function for $u \in [-20, 20]$ with the following parameters,

$$\begin{aligned} S_0 &= 100, & v_0 &= 0.06, & r &= 0.05, \\ \kappa &= 1, & \theta &= 0.06, & \sigma &= 0.3, \\ \rho &= -0.5, & \lambda &= 0.01, & \tau &= 1, \\ K &= 100, & n &= 100, & N &= 1000. \end{aligned}$$

Figure 1.1 and 1.2 gives an overlook of the computation.

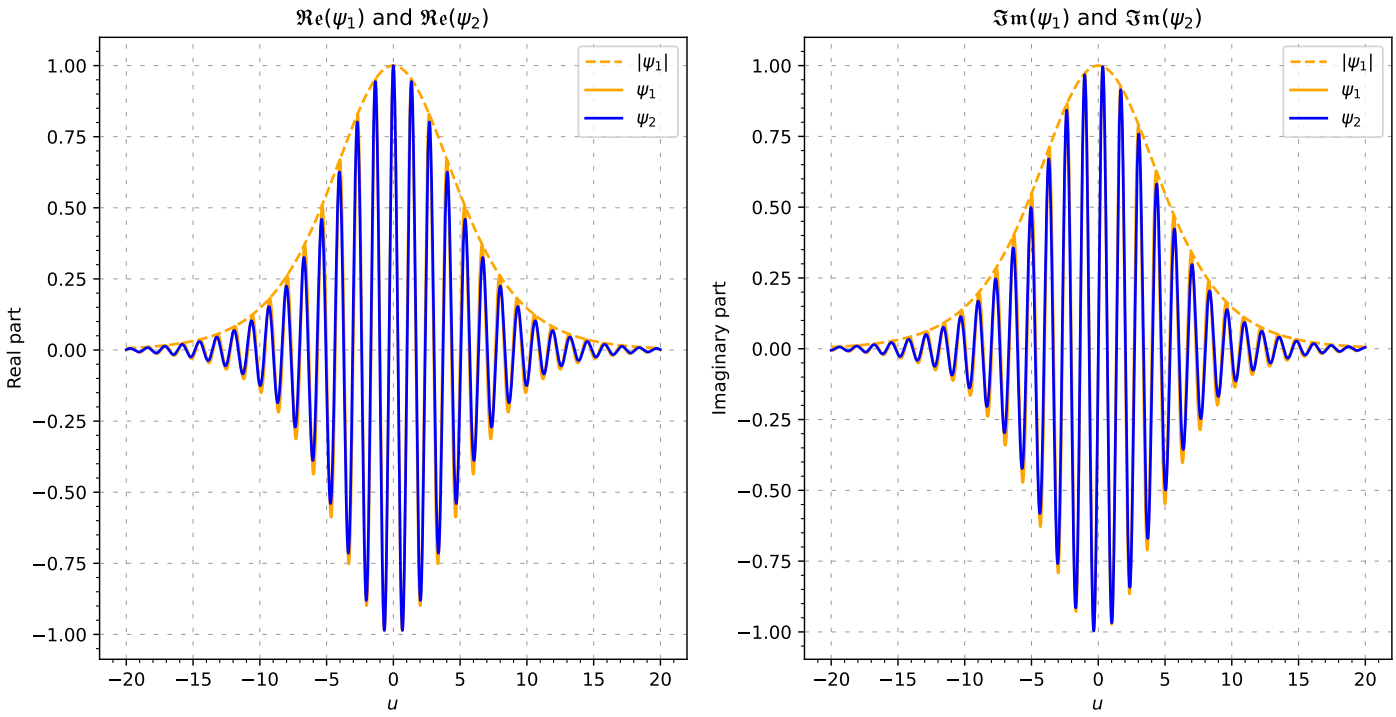


***Figure 1.1:*** *Real part (left hand side) and imaginary part (right hand side) of $\Psi_1$ and $\Psi_2$ for $\tau = 1$ and $u \in [-20, 20]$. The dash-line correspond to the absolute value of $\Psi_1$.*

From 1.1, it's apparent that the real part of $\Psi_1$ and $\Psi_2$ appears to be even, while the imaginary part seems to be odd. This observation aligns with the findings in 1.7, as the value of $Q_j$ at $(\ln(S_0), v_0, t; \ln(K))$ is determined in part by integrating an integrand that

directly depends on $\Psi_j$ over $\mathbb{R}_+$.

$$\frac{e^{-iu\ln(K)}\Psi^j_{x(T)}(x,v,t;u)}{iu} = -iu\frac{[cos(u\ln(K)) - isin(u\ln(K))]\left[\mathfrak{Re}\left(\Psi^j_{x(T)}\right) + i\mathfrak{Im}\left(\Psi^j_{x(T)}\right)\right]}{u^2}$$
$$= \frac{1}{u^2}[-iucos(u\ln(K)) - usin(u\ln(K))]\left[\mathfrak{Re}\left(\Psi^j_{x(T)}\right) + i\mathfrak{Im}\left(\Psi^j_{x(T)}\right)\right]$$

Thus, the integrand becomes,

$$\frac{1}{u^2}\left(-u \times sin(u\ln(K))\ \ \mathfrak{Re}\left(\Psi^j_{x(T)}\right) + cos(u\ln(K))\ \ u \times \mathfrak{Im}\left(\Psi^j_{x(T)}\right)\right).$$

The factor $1/u^2$ being even, it does not alter the parity of the remaining part of the function. As claimed earlier, $\mathfrak{Re}\Psi_j$ is even and $\mathfrak{Im}\Psi_j$ is odd. Additionally, the identity function is odd, cos is even, and sin is odd. Considering that the product of two even functions is even, the product of two odd functions is odd, and the product of an odd and an even function is odd, the integrand is therefore an even function. This implies that integrating it over $\mathbb{R}$ is equivalent to twice its integration over $\mathbb{R}_+$. This result is what we obtained.

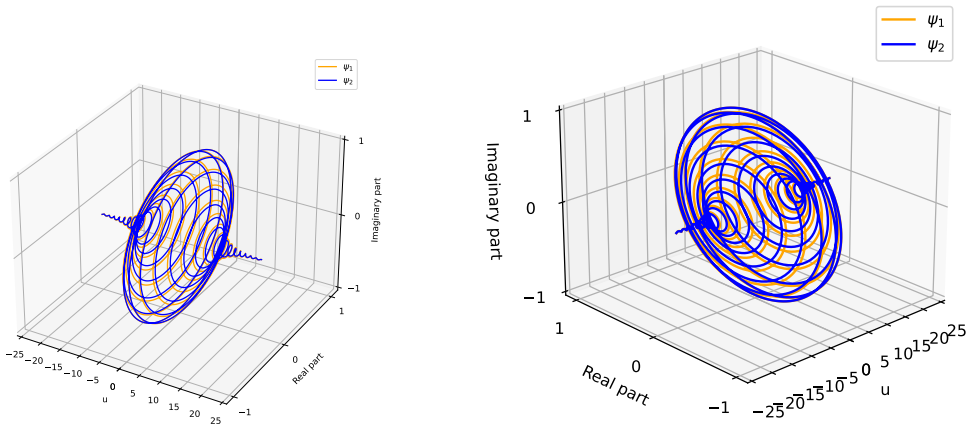


***Figure 1.2:*** *3D plot of $\Psi_1$ and $\Psi_2$. The $x$-axis represents $u \in [-20, 20]$, the $y$-axis represents the real part, and the $z$-axis represents the imaginary part.*

Knowing that that the integrand is an even function, we could have only plotted on $u \geq 0$,

### 1.3.2 Existence of the integral

Before using all the theoritical results to price an European Call under the Heston Model. We shall verify quickly that the integral in 1.7 exists. As the calculus may not be very pleasant to deal with, a computer verification shall be sufficient.

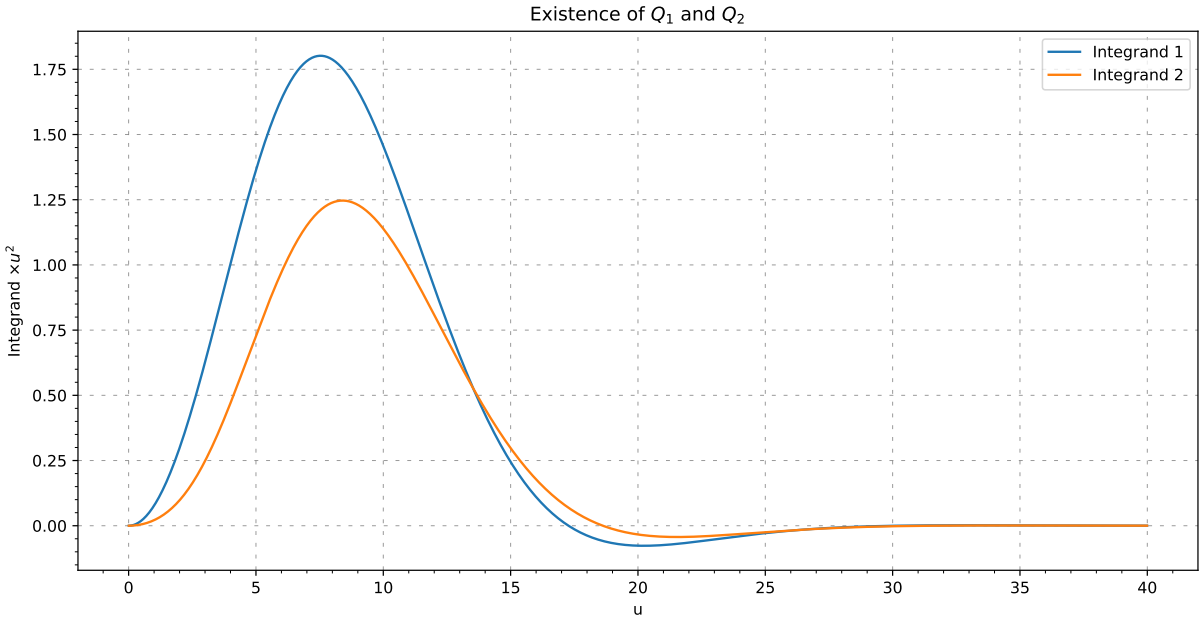


***Figure 1.3:*** *Plot of the function* $u \mapsto u^2 \times \mathfrak{Re}\left(e^{-iu\ln(K)}\Psi^j_{x(T)}/iu\right)$ *over* $[0, 40]$.

From the latter plot 1.3, we can see that,

$$\lim_{u\to+\infty} u^2 \times \mathfrak{Re}\left(\frac{e^{-iu\ln(K)}\Psi^j_{x(T)}}{iu}\right) = 0,$$

which implies,

$$\mathfrak{Re}\left(\frac{e^{-iu\ln(K)}\Psi^j_{x(T)}}{iu}\right) \underset{u\to+\infty}{=} o\left(\frac{1}{u^2}\right),$$

ensuring the existence of the integral.

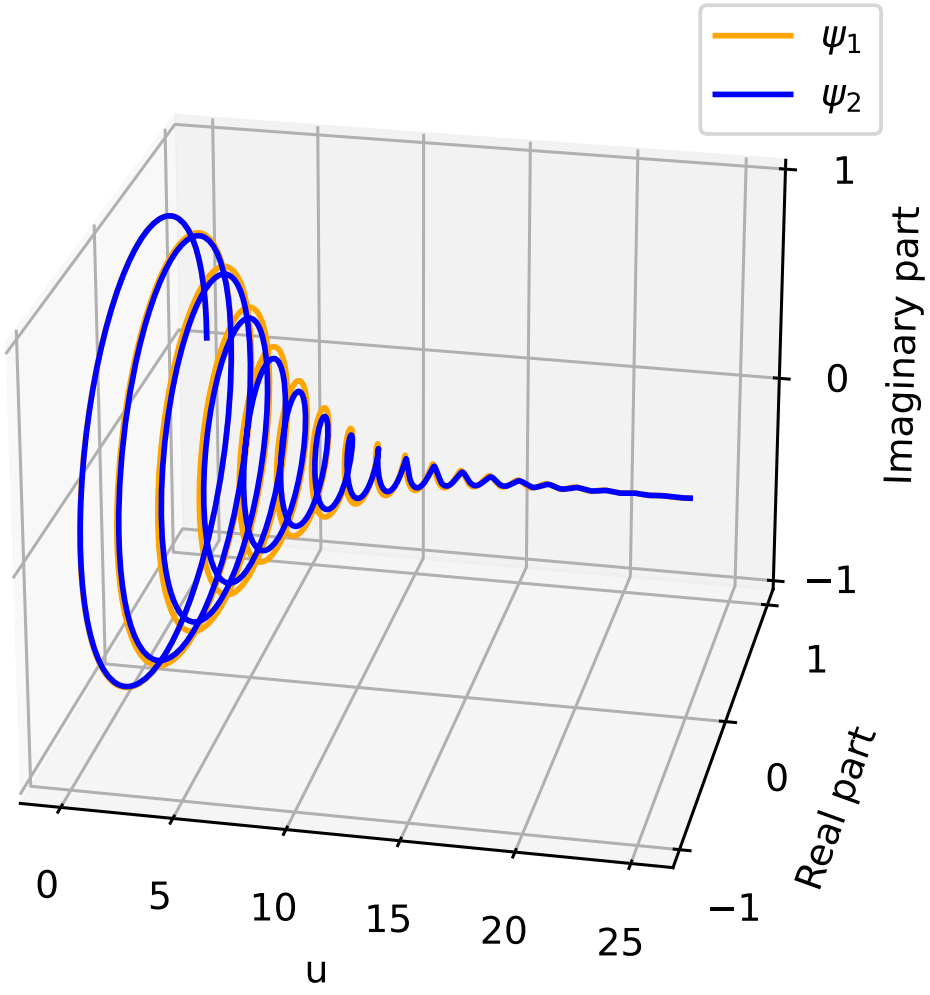


***Figure 1.4:*** *3D plot of $\Psi_1$ and $\Psi_2$. The $x$-axis represents $u \in [0, 25]$, the $y$-axis represents the real part, and the $z$-axis represents the imaginary part.*

Chapter 2

# Option pricing in the context of the Heston model

In this section, we shall compare different methods for pricing a call option on an underlying driven by the Heston model. In particular, we shall use a Monte Carlo method, original Heston formula using Fourier transforms, and the Carr-Madan formula.

Option pricing using Monte Carlo simulation means that we first simulate a large number $N > 0$ of generated diffusion paths following 1.2 and 1.4 over the time period $[0, T]$ with a time step $\Delta t = T/n$, where $n > 0$ is chosen, compute the option payoff for each path and, finally, obtain the option price by computing the average payoff of the option and discounting that quantity to the time when we price the option.

The generation of $\mathbf{S} = \{S_t : 0 \geq t \geq T\}$ and $\mathbf{v} = \{v_t : 0 \geq t \geq T\}$ trajectories for the Heston model may lead to negative value of $\mathbf{v}$ at some time points due to discretisation errors. When this occurs, we shall use a reflection scheme.

## 2.1 Euler–Maruyama and Milstein schemes

First of all, we shall quickly develop the equations enabling the paths sampling of $\{\mathbf{S}^{(\mathbf{j})}, \mathbf{v}^{(\mathbf{j})}\}_{j=1}^{N}$ over the time period $[0, T]$. We use Thygesen 2023, section 8.4 to get the following equations.

### 2.1.1 Euler-Maruyama

For each trajectory $j \in [1 : N]$ and discretisation time point $t \in \{\frac{kT}{n} : k \in [1 : n]\}$, $S_0^{(j)}$ and $v_0^{(j)}$ being $S_0$ and $v_0$, the Euler-Maruyama discretisation scheme is,

$$S_{t+\Delta t}^{(j)} = S_t^{(j)} + rS_t^{(j)}\Delta t + \sqrt{v_t^{(j)}} S_t^{(j)} \left(\rho N_{1,t}^{(j)} + \sqrt{1-\rho^2} N_{2,t}^{(j)}\right) \sqrt{\Delta_t} \tag{2.1a}$$

$$v_{t+\Delta t}^{(j)} = v_t^{(j)} + \left(\kappa(\theta - v_t^{(j)}) - \lambda v_t^{(j)}\right) \Delta t + \sigma \sqrt{v_t^{(j)}} N_{1,t}^{(j)} \sqrt{\Delta t} \tag{2.1b}$$

With $N_{1,t}^{(j)} \overset{i.i.d}{\sim} \mathcal{N}(0,1)$ and $N_{2,t}^{(j)} \overset{i.i.d}{\sim} \mathcal{N}(0,1)$.

To assess the error, we select an equation with an analytical solution, e.g., a geometric Brownian motion $\mathbf{X} = \{X_t : 0 \leq t \leq T\}$,

$$dX_t = f(t, X_t)dt + g(t, X_t)dW_t.$$

We first simulate a large number of realisations of the Brownian motion on a fine time grid. For each realisation, we compute the analytical solution, the Euler-Maruyama approximation, and the absolute error $\mathbb{E}|X_T - X_T^{(\Delta t)}|$.

$$\mathbb{E}|X_T - X_T^{(\Delta t)}| \sim \sqrt{\Delta t}.$$

We use the superscript $X_T^{(\Delta t)}$ to emphasise that the approximation of $X_t$ is based on the time step $\Delta t$. We fix the terminal time $T$ and measure the error $X_T - X_T^{(\Delta_T)}$, which is a random variable. From the latter equation, the error is $\mathcal{O}(\sqrt{\Delta t})$, indicating that the Euler-Maruyama scheme, in general, has a strong order of $0.5$. The error arises because $dX_t$ is approximated at the first order; the second-order term is,

$$\frac{1}{2}\frac{\partial g}{\partial x}(X_t, t)\, g(X_t, t)\left[(\Delta W)^2 - \Delta t\right], \quad \Delta W = W_{t+\Delta t} - W_t \sim \mathcal{N}(0, \Delta t).$$

### 2.1.2 Milstein

This analysis of the Euler-Maruyama scheme error suggests an immediate improvement. Since the dominant term of the local error is $g'g((\Delta W)^2 - \Delta t)/2$, we can simply include it in the discrete approximation. With this method, the error is of order $\mathcal{O}(\Delta t)$, so the Milstein scheme has a strong order of $1$.

This means that the error decreases linearly with $\Delta t$ in Milstein, whereas the error only decreases proportionally to $\sqrt{\Delta t}$ in Euler. Therefore, one needs a smaller time step with Euler-Maruyama (and so more iterations) to achieve a similar error as that obtained with Milstein. However, in the case of a Lamperti transformation, where $g = 1$ and thus $\partial g/\partial x = 0$, the error term vanishes and the Euler-Maruyama and Milstein methods are equivalent. In this regard, a priori, when the time step is very small (so that we have converged enough), similar paths should be obtained, as the Euler error term is directly proportional to $\Delta t$ (the $\Delta t$ is hidden in the variance of $\Delta W$).

For each trajectory $j \in [1 : N]$ and discretisation time point $t \in \{\frac{kT}{n} : k \in [1 : n]\}$, $S_0^{(j)}$ and $v_0^{(j)}$ being $S_0$ and $v_0$, the Milstein discretisation scheme is,

$$\begin{aligned} S_{t+\Delta t}^{(j)} &= S_t^{(j)} + rS_t^{(j)}\Delta t + \sqrt{v_t^{(j)}}S_t^{(j)}\left(\rho N_{1,t}^{(j)} + \sqrt{1-\rho^2}N_{2,t}^{(j)}\right)\sqrt{\Delta_t} \\ &\quad + \frac{1}{2}v_t^{(j)}S_t^{(j)}\left(\left(\rho N_{1,t}^{(j)} + \sqrt{1-\rho^2}N_{2,t}^{(j)}\right)^2 - 1\right)\Delta t \qquad (2.2a) \\ v_{t+\Delta t}^{(j)} &= v_t^{(j)} + \left(\kappa(\theta - v_t^{(j)}) - \lambda v_t^{(j)}\right)\Delta t + \sigma\sqrt{v_t^{(j)}}N_{1,t}^{(j)}\sqrt{\Delta t} \\ &\quad + \frac{\sigma^2}{4}\left(\left(N_{1,t}^{(j)}\right)^2 - 1\right)\Delta t. \qquad (2.2b) \end{aligned}$$

With $N_{1,t}^{(j)} \overset{i.i.d}{\sim} \mathcal{N}(0,1)$ and $N_{2,t}^{(j)} \overset{i.i.d}{\sim} \mathcal{N}(0,1)$.

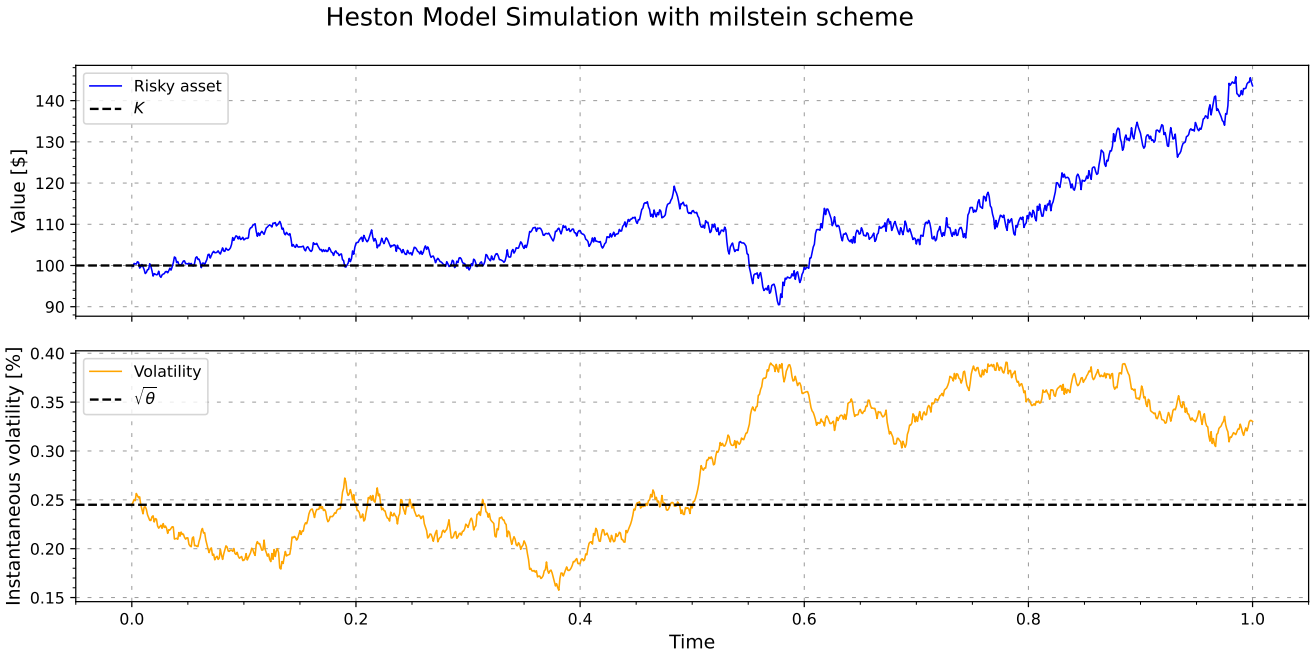


***Figure 2.1:*** *Path simulation of a Heston model with parameters given in Table 2.1 (with $n = 1000$) using the Milstein scheme.*

Given the large number of time points (i.e., the small time step), the path fig 2.1 has been sampled using the Milstein scheme, but the Euler scheme produces a very similar path. However, the difference between Euler and Milstein is barely discernible when plotting the trajectories.

## 2.2 Estimation by Monte Carlo

### 2.2.1 Principles

Let $\{S_T^{(j)}\}_{j=1}^N$ and $\{v_T^{(j)}\}_{j=1}^N$ the sampled risky asset and sampled variance at time $T$. The trajectories are mutually independent. These have been generated by an Euler–Maruyama or Milstein method. In this case, we estimate the arbitrage-free valuation at time 0 of the $T$-$K$-call option of payoff $f$, 1.1 by, $\mathcal{V} = e^{-rT}\mathbb{E}^{\mathbb{Q}}_{v_0,S_0}[f(S_T)]$ writing $\psi = e^{-rT}f$,

$$\hat{\mathcal{V}}_N = \frac{1}{N}\sum_{j=1}^{N} \psi\left(S_T^{(j)}\right),$$

so that, by the Central Limit Theorem,

$$\frac{\hat{\mathcal{V}}_N - \mathbb{E}^{\mathbb{Q}}_{v_0,S_0}\left[\psi(S_T)\right]}{\sqrt{\frac{\sigma^2}{N}}} \xrightarrow{w} \mathcal{N}(0,1) \tag{2.1}$$

With, the superscript $w$ meaning convergence in law (or weak convergence) and $\sigma$ the standard deviation of the option payoff. Nevertheless, this latter is an unknown quantity which can estimated by computing the following unbiased and strongly convergent estimator,

$$\hat{\sigma}_N = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}\left(\psi\left(S_T^{(j)}\right) - \hat{\mathcal{V}}_N\right)^2},$$

so that, we still have the convergence in 2.1 if we replace $\sigma$ by $\hat{\sigma}_N$.

Hence, from this two unbiased estimation of $\mathcal{V}$ and $\sigma$, we can derive a confidence interval on the estimator $\hat{\mathcal{V}}_N$. Indeed, weak convergence implies that, for all $\varepsilon > 0$,

$$\mathbb{P}\left(\sqrt{N}\frac{\hat{\mathcal{V}}_N - \mathbb{E}^{\mathbb{Q}}_{v_0,S_0}\left[\psi(S_T)\right]}{\hat{\sigma}_N} \in [-\varepsilon,\varepsilon]\right) \underset{N\to+\infty}{\longrightarrow} \mathbb{P}\left(|\eta| \leq \varepsilon\right),$$

where $\eta \sim \mathcal{N}(0,1)$, so that,

$$\left[\hat{\mathcal{V}}_N - \varepsilon\frac{\hat{\sigma}_N}{\sqrt{N}}; \hat{\mathcal{V}}_N + \varepsilon\frac{\hat{\sigma}_N}{\sqrt{N}}\right]$$

is a confidence interval for $\mathbb{E}^{\mathbb{Q}}_{v_0,S_0}\left[\psi(S_T)\right]$ of asymptotic level $\alpha$ if $\varepsilon$ is the quantile of order $1-\alpha/2$ of the standard Gaussian, i.e. $\mathbb{P}(|\eta| \leq \varepsilon) = 1-\alpha/2$, with still $\eta \sim \mathcal{N}(0,1)$. In the simulation, we take $\alpha = 0.05$, so that $\varepsilon \approx 1.96$.

### 2.2.2 Simulation results

We now apply the theory discussed above. For repeatability, we set the random seed, allowing us to compare the Milstein and Euler schemes as they will be sampled on the same path. We price a call option assuming the following values for the model parameters:

$$\begin{array}{lll} S_0 = 100, & v_0 = 0.06, & r = 0.05, \\ \kappa = 1, & \theta = 0.06, & \sigma = 0.3, \\ \rho = -0.5, & \lambda = 0.01, & T = 1, \\ K = 100, & n = 100, & N = 1000. \end{array}$$

Table 2.1 summarises the results obtained.

| Scheme | Price (in $) | Standard Deviation | Confidence Interval (in $) |
|---|---|---|---|
| Euler | 12.05 | 0.49 | $[12.01, 12.09]$ |
| Milstein | 12.02 | 0.47 | $[11.93, 12.03]$ |

***Table 2.1:*** *Results obtained from simulation with $n = 100$ time points and $N = 1000$ trajectories*

The two schemes yield similar results, with the Milstein scheme providing a slightly more precise estimate due to its smaller standard deviation. The confidence intervals highlight that our results are precise enough to be acceptable, as they provide narrow ranges within which the true option price is likely to lie with a high degree of confidence.

### 2.2.3 Sensitivities analysis

Intuitively, we have two parameters that we can adjust: the number of trajectories, $N$, and the number of time points, $n$.

Increasing the number of trajectories allows us to capture a larger sample of possible outcomes, it should provide a more accurate estimation of option prices. By comparing the results obtained with different values of $N$, we can assess how the precision of our estimates improves with larger sample sizes.

Furthermore, adjusting $n$, the number of time points, affects the granularity of our simulation. A higher value of $n$ results in more frequent observations along each trajectory, potentially leading to more accurate estimates of option prices.

**Sensitivity with the number of sampled paths $N$** First, we increase $N$ to $10^5$ and then to $10^6$, with a fixed $n = 100$, to observe the effect of the number of particles.

| Scheme | Price (in $) | Standard Deviation | Confidence Interval (in $) |
|---|---|---|---|
| Euler | 12.05 | 0.49 | $[12.01, 12.09]$ |
| Milstein | 12.02 | 0.47 | $[11.93, 12.03]$ |
| Euler | 11.89 | 0.16 | $[11.88, 11.89]$ |
| Milstein | 11.95 | 0.16 | $[11.95, 11.95]$ |
| Euler | 11.96 | 0.02 | $[11.96, 11.96]$ |
| Milstein | 11.96 | 0.02 | $[11.96, 11.96]$ |

***Table 2.2:*** *Results obtained from simulation with $n = 100$ time points, $N = 10^4$ (upper part of the table), $N = 10^5$ (middle part of the table) and $N = 10^6$ (lower part of the table)*

Analysing Table 2.2 compared to the initial case where $N = 10^3$ trajectories, the standard deviations are significantly smaller now, resulting in more precise confidence intervals. The standard deviations are now equal between the two schemes. It seems like

the number of trajectories compensates for the difference in the strong order between the two schemes. Further investigation is needed, but from this, to improve precision, it seems better to increase the number of trajectories rather than decrease the time step.

Moreover, the prices are noticeably different compared to the case with $N = 1000$ trajectories. We observe a significant change for the Euler scheme across all three changes in $N$, while there is only a minor difference between the cases of $N = 10^5$ and $N = 10^6$ for the Milstein scheme. This indicates that if we are "unlucky", our Monte Carlo estimator can provide a biased estimate even though theoretically, this is not expected. The estimation for Euler with $10^6$ (and $10^5$) trajectories is not even within the confidence interval when run for $10^3$ trajectories. We observe that this can be more easily addressed for the Milstein scheme than for the Euler scheme. Given that for $10^6$ trajectories, both schemes converge towards the same value, we can conclude that our two estimators are not biased a priori.

Now, let's conduct a fairly similar analysis regarding the number of time points $n$.

**Sensitivity with the number of time points $n$** First, we increase $n$ to $10^3$ and then to $10^4$, with a fixed $N = 1000$[1], to observe the effect of the number of paths.

| Scheme | Price (in $) | Standard Deviation | Confidence Interval (in $) |
|---|---|---|---|
| Euler | 12.05 | 0.49 | [12.01, 12.09] |
| Milstein | 12.02 | 0.47 | [11.97, 12.06] |
| Euler | 11.33 | 0.49 | [11.29, 11.38] |
| Milstein | 12.47 | 0.50 | [12.43, 12.52] |
| Euler | 12.31 | 0.52 | [12.27, 12.36] |
| Milstein | 12.00 | 0.50 | [11.95, 12.04] |

***Table 2.3:*** *Results obtained from simulation with $N = 1000$ time points, $n = 10^2$ (upper part of the table), $n = 10^3$ (middle part of the table) and $n = 10^5$ (lower part of the table)*

Which is consistent with our theoretical result as the estimator of the standard deviation of the payoff depends on the number of particles $N$ and not the number of time points $n$.

However, the prices obtained vary significantly depending on the number of time points. This echoes the issue raised previously—our estimators do not truly converge (furthermore, our two schemes do not converge towards the same values). In this respect, we run the simulation with $n = 10$ and get similar results as in the Table 2.3.

Ultimately, it is preferable to increase the number of trajectories rather than the number of time points. However, it is still necessary to maintain a sufficiently small time step to preserve some semblance of realism.

[1]The value of 1000 has been chosen so that we can observe improvements or losses.

**Limits of the simulation** Throughout all the previous simulations with different sets of time points and trajectories, the occurrences of negative and null variance processes have been counted. Surprisingly, both of these counts are zero. One might have expected to observe at least some negative variance in the Euler scheme due to error terms or when the number of time points is low, but this is not the case. It is possible that we have been fortunate with the random seed.

## 2.3 Estimation by Fourier Transform

Now that we've observed the Monte Carlo estimation, let's proceed to price a Call option under the Heston model using Fourier Transform.

### 2.3.1 Principles

In this method, we don't need to simulate paths. Instead, we only have to compute the quantity (1.5), which involves evaluating the two integrals for $j \in \{1, 2\}$ from (1.7), determining $Q_j(\ln(S_0), v_0, 0; \ln(K)))$, using the closed forms of $D_j(\tau; u)$ and $C_j(\tau; u)$, given by (1.10) and (1.11).

### 2.3.2 Simulation results

The following 2.4 summarises the results the output of the Call option pricing via Fourier Transform.

| Price (in \$) | Error (in \$) | Confidence Interval (in \$) |
|---|---|---|
| 11.94 | $1.8 \times 10^{-7}$ | $[11.94, 11.94]$ |

***Table 2.4:*** *Results obtained from Fourier Transform. Please note that we have rounded to the nearest hundredth. In reality, the price is* $11.935747$*, and the confidence interval is* $[11.939999, 11.940000]$*, which round to* $11.94$*.*

The error has been computed from the error in the computation of the integral in `Python`. We used the function `quad` from the `scipy.integrate` package, which also returns an integration error $\varepsilon_j^I$, where $j \in \{1, 2\}$. Then, we compute the pricing error, $\varepsilon$, as follows:

$$\varepsilon = S_0 \varepsilon_1^I + K e^{-r\tau} \varepsilon_2^I.$$

The reason we have an addition between the errors, whereas in the call valuation formula 1.5 it's a subtraction, is twofold: firstly, we want a positive error, and secondly, integration errors accumulate, making the result more uncertain. Intuitively, we can also say that we consider the "worst" case.

Comparing the pricing results obtained via Fourier Transform 2.4 and via Monte Carlo 2.2 with a large number of simulated paths, we can see that the two results con-

verge toward the same results. In the same respect the confidence interval are both very accurate.

From a computation time perspective, we can already note that the Fourier Transform method is not dependent on the number of particles, whereas the complexity of Monte Carlo estimations is in $\mathcal{O}(n \times N)$, where $n$ is the number of time points and $N$ is the number of simulations. Thus, we can already conjecture that the simulation time for Monte Carlo between $N = 10^3$ and $N = 10^6$ will have a factor of approximately $10^3$[2]. As the parameter $n$ does not significantly influence the accuracy of the Monte Carlo estimation, we fix it at $n = 100$, and compare the computation time of the Monte Carlo estimation in terms of the number of simulations $N$. The following table 2.5 summarises the computation time results.

| **Method** | **Parameters** | **Execution Time (s)** |
|---|---|---|
| Monte Carlo Euler | $n = 100$, $N = 10^3$ | 0.0199 |
| Monte Carlo Milstein | $n = 100$, $N = 10^3$ | 0.0318 |
| Monte Carlo Euler | $n = 100$, $N = 10^6$ | 26.273 |
| Monte Carlo Milstein | $n = 100$, $N = 10^6$ | 31.512 |
| Fourier Transform | - | 0.0586 |

***Table 2.5:*** *Summary of execution times for different pricing methods*

From 2.5, we indeed observe a multiplicative factor of $10^3$ between the Monte Carlo estimations with $N = 10^3$ and $N = 10^6$. Note that the Milstein scheme takes slightly longer than the Euler scheme, which is logical as more elementary operations are performed in the Milstein scheme than in the Euler scheme. The valuation method via Fourier Transform outperforms Monte Carlo estimations when the number of simulations is high. Since Monte Carlo methods are effective only when the number $N$ of simulations is high, we can conclude that the pricing method via Fourier Transform outperforms Monte Carlo valuation methods overall in term of accuracy and computation time.

## 2.4 Estimation by Carr-Madan

Now that we've observed the Monte Carlo estimation and the estimation via Fourier Transform, let's proceed to price a Call option via the Carr-Madan formula. This method aims at using the Fourier transform to value options when the characteristic function of the return is known analytically.

[2] As common mathematical operations are typically in $\mathcal{O}(\infty)$ in `Python`.

### 2.4.1 Principles

Let $f(x) = \max\{e^x - e^k, 0\}$ be the payoff of a vanilla call option with strike $K = e^k$ and maturity time $T$.

**The Generalised Fourier Transform** This function is not $\mathcal{L}^1$ and has not traditional Fourier Transform. It has however, a generalised Fourier Transform Singh, Gupta, and Joshi 2022. Normally, if $f : \mathbb{R} \mapsto \mathbb{R}$, then we define the Fourier Transform (in finance) to be,

$$\begin{array}{rccc} \hat{f}: & \mathbb{R} & \to & \mathbb{C} \\ & u & \mapsto & \int_{\mathbb{R}} e^{iux} f(x) dx \end{array}$$

For this integral to exist, $f$ needs to decay rapidly or be of compact support. The payoff function does not satisfy this.

The generalised Fourier Transform of $f$ is defined for a subset $\mathcal{S}_f \subset \mathbb{C}$,

$$\begin{array}{rccc} \hat{f}: & \mathcal{S}_f & \to & \mathbb{C} \\ & u & \mapsto & \int_{\mathbb{R}} e^{iux} f(x) dx \end{array}$$

As it turns out this $\mathcal{S}_f$ is a horizontal strip line in the complex plane. We can compute the transform for the payoff function as follows,

$$\begin{aligned} \hat{f}(u) &= \int_{-\infty}^{+\infty} e^{iux} \left(e^x - e^k\right)^+ dx \\ &= \int_{-\infty}^{k} \left(e^{x(iu+1)} - K e^{iux}\right)^+ dx \\ &= \left[\frac{e^{x(iu+1)}}{iu+1} - K\frac{e^{iux}}{iu}\right]_{x=k}^{x=+\infty} \\ &= -\frac{e^{ik(u-i)}}{u(u-i)}. \end{aligned}$$

This last step is only valid if the term indeed vanishes as $x \to +\infty$. This only happens if and only if $\mathfrak{Im}(u) > 1$. Thus, the strip line for the call option payoff is $\mathcal{S}_f = \{u \in \mathbb{C} : \mathfrak{Im}(u) > 1\}$.

Similarly, for a put option, we have $\mathcal{S}_f = \{u \in \mathbb{C} : \mathfrak{Im}(u) < 0\}$. These are the strips of integration which the payoff functions have valid Fourier transform. Note that both strips exclude the real line (i.e., there is no standard Fourier transform).

**The Fourier Transform of an Option Price** We denote the initial call value $\mathcal{V}_T(k)$ and the risk-neutral density by $x \mapsto \phi_T(x)$. We consider the modified call price $v_T(k)$ defined by,

$$v_T(k) \stackrel{\text{def}}{=} \exp(\alpha k)\mathcal{V}_T(k), \quad \alpha > 0.$$

For a range of positive values of $\alpha$, we expect that $v_T(k)$ is square-integrable in $k$ over the entire real line. Consider now the Fourier transform of $\hat{v}_T$ defined by,

$$\hat{v}_T(u) = \int_{-\infty}^{+\infty} e^{iux} v_T(k) dk.$$

We fist develop an analytical expression for $\hat{v}_T$ in terms of $\phi_T$ and then obtain call prices numerically using the inverse transform,

$$\mathcal{V}_T(k) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{+\infty} e^{-iuk} \hat{v}_T(u) du$$

$$\boxed{\mathcal{V}_T(k) = \frac{e^{-\alpha k}}{\pi} \int_{0}^{+\infty} e^{-iuk} \hat{v}_T(u) du} \tag{2.1}$$

The second equality holds because $\mathcal{V}_T(k)$ is real, which implies that the function $\hat{v}_T(u)$ is odd in its imaginary part and even in its real part.

The expression of $\hat{v}_T(u)$ is determined as follows, using that the initial call value is related to the risk-neutral density $\phi_T(x)$ by $\mathcal{V}_T(k) = \int_k^\infty e^{-rT}(e^x - e^k)\phi_T(x)$,

$$\begin{aligned}
\hat{v}_T(u) &= \int_{-\infty}^{+\infty} e^{iuk} v_T(k) dk \\
&= \int_{-\infty}^{+\infty} e^{iuk} \int_{k}^{+\infty} e^{\alpha k} e^{-rT} \left(e^x - e^k\right) \phi_T(x) dx dk \\
&= \int_{-\infty}^{+\infty} e^{-rT} \phi_T(x) \int_{-\infty}^{k} \left(e^{x+\alpha k} - e^{(1+\alpha)k}\right) e^{iuk} dk dx \\
&= \int_{-\infty}^{+\infty} e^{-rT} \phi_T(x) \left( \frac{e^{(\alpha+1+iu)x}}{\alpha + iu} - \frac{e^{(\alpha+1+iu)x}}{\alpha + 1 + iu} \right) dx
\end{aligned}$$

Which leads us to, denoting $\Psi_T$ the characteristic function of the risk neutral density,

$$\boxed{\hat{v}_T(u) = \frac{e^{-rT}}{\alpha^2 + \alpha - u^2 + iu(2\alpha + 1)} \Psi_T(u - i(\alpha + 1))} \tag{2.2}$$

The price of a vanilla call is determined by substituting 2.2 into 2.1 and computing the required integration.

**Choice of $\alpha$** Now we have to choose the damping coefficient $\alpha$[3]. According to Carr and Madan n.d. and OpenGamma Quantitative Research 2013 (Section 1.3), we choose $\alpha$ as,

$$\alpha = \frac{1}{2} \left( \frac{\kappa^2}{\sigma(2\kappa - \sigma)} + \frac{\kappa^2}{\sigma(-2\kappa - \sigma)} \right)$$

Even if OpenGamma Quantitative Research 2013 suggests a method for optimising the choice of $\alpha$, we stick ourselves with this naive choice. In the simulation set-up, $\alpha \approx 0.26$.

[3] I initially didn't notice that $\alpha$ was fixed in the question. I included this brief paragraph because I conducted some research, but the simulations were indeed conducted using the $\alpha$ provided in the assignment.

### 2.4.2 Simulation results

The following 2.6 summarises the results the output of the Call option pricing via Carr-Madan.

| Price (in $) | Error (in $) | Confidence Interval (in $) |
|---|---|---|
| 11.94 | $3.8 \times 10^{-13}$ | $[11.94, 11.94]$ |

***Table 2.6:*** *Results obtained from Fourier Transform. Please note that we have rounded to the nearest hundredth (except for the error term). The computation time has been* $0.0495s$

We obtain the same price as for the other methods. Similar to the method using pricing via Fourier Transform, the error here is given by

$$\varepsilon = \frac{e^{-\alpha \ln K}}{\pi} \varepsilon^I,$$

where $\varepsilon^I$ is the integration error provided by the `quad` function of `scipy.integrate`. Note that since here the price of the vanilla option is not obtained via addition, the error obtained by Carr-Madan is smaller than that obtained by Fourier Transform[4]. Also, it is worth noting that errors do not actually accumulate in the Fourier Transform method. Thus, as the source of error arises from numerical integration in both the Carr-Madan and Fourier Transform methods, the two methods are similar in terms of error.

Similarly, since both methods boil down to numerically calculating an integral, the computation times for both are similar.

## 2.5 Conclusion

Ultimately, pricing via Fourier Transform and the Carr-Madan formula yields similar results in terms of price, computation time, and error. This can be attributed to the fact that both methods essentially solve the same type of problem, which is numerically calculating an integral.

As for pricing via the Monte Carlo method, both the price and the accuracy of the price are highly dependent on the number of simulations performed. To achieve sufficient accuracy and to be reasonably certain that the result obtained is not biased by the simulations, a large number of trajectories need to be simulated. In this case, this method also converges to the same price as the other two methods. However, this leads to a higher computation time compared to the other two methods, all for a precision that is inferior to the other two.

In a nutshell, the estimation methods by Carr-Madan and Fourier Transform yield similar results and outperform Monte Carlo pricing in terms of precision and speed. However, Monte Carlo can still be useful for benchmarking and for its simplicity of implementation. It is worth noting that a multitude of Monte Carlo methods exist, some

[4] since we have considered the "worst-case scenario," we simply added the two errors.

aiming to reduce variance and increase speed, such as Importance Sampling, Antithetic Variates, Control Variates, or Stratified Sampling, to name a few.

Chapter 3

# Asset allocation under the Heston model

In this exercise, we shall answer the following question: how to optimally invest money in a market with one risk-free asset $S_0$ and one risky asset $S_1$ driven by the Heston model?

Consider an investor who has initial capital $x_0 > 0$ and wants to invest it two assets during the time period $[0, T]$. The portfolio is composed of two tradable assets:

- a risk-free asset $S_0$, e.g., a bank account with continuously compounding interest rate, which happens in reality on a daily or monthly basis.
- a risky asset $S_1$ , e.g., an exchange-traded fund (ETF) tracking S&P500 stock market index.

In contrast to the lecture notes, we will parameterise the Heston model in a more convenient way, in which we set $\mu = r + \bar{\lambda} v(t)$ with $\bar{\lambda}$ being interpreted as a premium for volatility risk:

$$dS_0(t) = rS_0(t)dt, \quad S_0(0) = 1, \tag{3.1}$$

$$dS_1(t) = \left(r + \bar{\lambda}\sqrt{v(t)}\right) S_1(t)dt + \sqrt{v(t)}S_1(t)dW_1^{\mathbb{P}}(t), \quad S_1(0) = s_1 > 0, \tag{3.2}$$

$$dv(t) = \kappa\left(\theta - v(t)\right)dt + \sigma\sqrt{v(t)}\left(\rho dW_1^{\mathbb{P}}(t) + \sqrt{1-\rho^2}dW_2^{\mathbb{P}}(t)\right), \quad v(0) > 0. \tag{3.3}$$

With $W_1^{\mathbb{P}}$ and $W_2^{\mathbb{P}}$ are independent Brownian motion. Only the assets $\mathbf{S_0} = \{S_0(t) : 0 \leq t \leq T\}$ and $\mathbf{S_1} = \{S_1(t) : 0 \leq t \leq T\}$ are traded.

Let denote the fraction of wealth invested the risky asset $\mathbf{S_1}$ at time $t \in [0, T]$ by $\pi = \{\pi(t) : 0 \leq t \leq T\}$. It is also called an investment strategy and is modelled by a continuous stochastic process that is integrable w.r.t.the return process $dS_1(t)/S_1(t)$.

We write $X^{x_0,\pi} = \{X^{x_0,\pi}(t) : 0 \leq t \leq T\}$ the portfolio value process under the real world measure $\mathbb{P}$, which evolves according to the following stochastic differential,

$$\begin{aligned} dX^{x_0,\pi}(t) &= (1-\pi(t))\, X^{x_0,\pi}(t)\frac{dS_0(t)}{S_0(t)} + \pi(t)X^{x_0,\pi}(t)\frac{dS_1(t)}{S_1(t)} \\ &= X^{x_0,\pi}(t)\left((r+\pi(t)\bar{\lambda}v(t))dt + \pi(t)\sqrt{v(t)}dW_1^{\mathbb{P}}(t)\right), \quad X^{x_0,\pi}(0) = x_0. \end{aligned} \tag{3.4}$$

Let us also denote the set of all admissible investment strategies for the initial capital $x_0$ and the initial variance $v_0$ by $\mathcal{A}(x_0, v_0)$.

We model the investor's preferences via a so-called power utility function

$$U(x) = \frac{x_p}{p},$$

where $p \in (-\infty, 0) \cup (0, 1)$ is the risk-aversion parameter. In the this respect, a rational investor solves the following asset-allocation problem by maximising his utility,

$$\max_{\pi \in \mathcal{A}(x_0,v_0)} \mathbb{E}\Big[U(X^{x_0,\pi}(T))\Big]. \tag{3.5}$$

Our main problem will be to determine an optimal strategy $\pi$ to satisfies the Portfolio Optimization Problem 3.5.

## 3.1 Derivation of the Hamilton-Jacobi-Bellman PDE

In this section, we use dynamic programming principle of stochastic control to heuristically derive a candidate optimal strategy $\pi^\star$ for the Portfolio Optimization Problem 3.5.

In our set up, the initial wealth $x_0$, the initial value $v_0$ of the variance, and the time $T$ till the end of the investment horizon are fixed. The key idea of the Bellman principle is that it is easier to solve a dynamic version of this problem, where all of these variables are allowed to vary. In this respect, let us consider the so-called value function $V(t, x, v)$, which describes the maximal expected utility of terminal wealth that can be attained by trading optimally on $[t, T]$, given that the starting wealth at $t \in [0, T]$ is $x$ and the respective value of the variance is $v$:

$$V(t, x, v) = \max_{\pi \in \mathcal{A}(t,x,v)} \mathbb{E}\Big[U(X^{\pi}(T))\Big].$$

As we attempt to solve a dynamic version of the initial problem 3.5, we need to somehow land on our feet by setting a final condition that allows us to revert to our initial problem,

$$V(T, x, v) = U(x).$$

Consider any constant investment strategy $\pi$ applied over a small time interval $[t, t+h]$, with $h \in [0, T-t]$. By applying Itō's formula to the transformed process $V(t, X^{\pi}(t), v(t))$,

we obtain:

$$\begin{aligned} dV(t, X^\pi, v) = &\Big[V_t + (V_x; V_v) \cdot \big(X^\pi(r + \pi\bar{\lambda}v); \kappa(\theta - v)\big)^\top\Big]dt \\ &+ \frac{1}{2}\Big[V_{xx}(X^\pi\pi)^2 v + V_{xv}\left(\sigma v X^\pi \pi + \sigma v X^\pi \pi\right) + V_{vv}\sigma^2 v\Big]dt \\ &+ V_x X^\pi \pi\sqrt{v} dW_1^{\mathbb{P}} + V_v \sigma\sqrt{v}\left(\rho dW_1^{\mathbb{P}} + \sqrt{1-\rho^2}dW_2^{\mathbb{P}}\right) \end{aligned}$$

Where the arguments of the functions are omitted for brevity, leading us to:

$$\begin{aligned} dV(t, X^\pi, v) = &\Big[V_t + V_x X^\pi(r + \pi\bar{\lambda}v) + V_v\kappa(\theta - v) + \frac{1}{2}V_{xx}(X^\pi\pi)^2 v + V_{xv}\sigma v X^\pi\pi + \frac{1}{2}V_{vv}\sigma^2 v\Big]dt \\ &+ \sqrt{v}\Big[V_x X^\pi\pi + \rho V_v\sigma\Big]dW_1^{\mathbb{P}} + \Big[V_v\sigma\sqrt{v}\sqrt{1-\rho^2}\Big]dW_2^{\mathbb{P}} \end{aligned}$$

Then, by integrating the latter equation between $t$ and $t + h$,

$$\begin{aligned} V(t+h, X^\pi(t+h), v(t+h)) = &V(t, X^\pi(t), v(t)) \\ &+ \int_t^{t+h} \Big[V_t + V_x X^\pi(s)(r + \pi\bar{\lambda}v(s)) \\ &\quad + V_v\kappa(\theta - v(s)) \\ &\quad + V_{xx}(X^\pi(s)\pi(s))^2 v(s) \\ &\quad + V_{xv}\sigma v(s) X^\pi(s)\pi(s) \\ &\quad + \frac{1}{2}V_{vv}\sigma^2 v(s)\Big]ds \\ &+ \int_t^{t+h} \sqrt{v(s)}\Big[V_x X^\pi(s)\pi(s) + \rho V_v\sigma\Big]dW_1^{\mathbb{P}}(s)ds \\ &+ \int_t^{t+h} \Big[V_v\sigma\sqrt{v(s)}\sqrt{1-\rho^2}\Big]dW_2^{\mathbb{P}}(s) \end{aligned}$$

Where we omit writing $(s, X^\pi(s), v(s))$ in front of the partial derivatives of $V$ for clarity. We can insert this into the following equation, which is assumed by the dynamic programming principle:

$$\begin{aligned} V(t, x, v) &\geq \mathbb{E}\Big[V(t+h, X^\pi(t+h), v(t+h))\Big| X^\pi(t) = x, v(t) = v\Big] \\ &\stackrel{\text{def}}{=} \mathbb{E}_{t,x,v}\Big[V(t+h, X^\pi(t+h), v(t+h))\Big]. \end{aligned} \tag{3.1}$$

Assuming that the stochastic integrals with respect to the Brownian motions $W_1^{\mathbb{P}}$ and $W_1^{\mathbb{P}}$ exist. Using the ugly equation obtained by applying the Ito Formula, the latter equation leads to, by utilizing the martingale property of the Wiener processes $W_1^{\mathbb{P}}$ and $W_2^{\mathbb{P}}$,

$$\begin{aligned} 0 \geq \mathbb{E}_{t,x,v}\Big[\int_t^{t+h} &\Big[V_t + V_x X^\pi(s)(r + \pi\bar{\lambda}v(s)) + V_v\kappa(\theta - v(s)) \\ &+ \frac{1}{2}V_{xx}(X^\pi(s)\pi(s))^2 v(s) + V_{xv}\sigma v(s) X^\pi(s)\pi(s) + \frac{1}{2}V_{vv}\sigma^2 v(s)\Big]ds\Big]. \end{aligned}$$

Now, divide by $h$ and let $h$ tend to zero, obtaining:

$$0 \geq \mathbb{E}_{t,x,v}\Big[V_t + V_x X^\pi(t)(r + \pi\bar{\lambda}v(t)) + V_v\kappa(\theta - v(t)) \\ + \frac{1}{2}V_{xx}(X^\pi(t)\pi(t))^2 v(t) + V_{xv}\sigma v(t)X^\pi(t)\pi(t) + \frac{1}{2}V_{vv}\sigma^2 v(t)\Big].$$

This yields:

$$\boxed{0 \geq V_t + V_x x(r + \pi\bar{\lambda}v) + V_v\kappa(\theta - v) + \frac{1}{2}V_{xx}(x\pi(t))^2 v + V_{xv}\sigma v x\pi(t) + \frac{1}{2}V_{vv}\sigma^2 v.} \tag{3.2}$$

Since the above inequality holds for any constant investment strategy $\pi$, it follows from the previous expression that:

$$0 \geq \sup_{\pi\in\mathcal{A}(t,x,v)}\left\{V_t + V_x x(r + \pi\bar{\lambda}v) + V_v\kappa(\theta - v) + \frac{1}{2}V_{xx}(x\pi(t))^2 v + V_{xv}\sigma v x\pi(t) + \frac{1}{2}V_{vv}\sigma^2 v\right\}.$$

Which is,

$$\boxed{0 \geq V_t + \frac{1}{2}V_{vv}\sigma^2 v + V_v\kappa(\theta - v) + \sup_{\pi\in\mathcal{A}(t,x,v)}\left\{\underbrace{V_x x(r + \pi(t)\bar{\lambda}v) + \frac{1}{2}V_{xx}(x\pi(t))^2 v + V_{xv}\sigma v x\pi(t)}_{\stackrel{\text{def}}{=} g(\pi)}\right\}.} \tag{3.3}$$

However, for the optimal strategy $\pi^\star$, we have equality in the dynamic programming principle inequality 3.1. Thus, we can repeat all previous calculations with equalities for the optimal strategy $\pi^\star$.

## 3.2 Solution to the Hamilton-Jacobi-Bellman PDE

We assume that $V_{xx} < 0$. For clarity, we rewrite $g$ as follows:

$$g(\pi) = \pi^2\frac{1}{2}x^2 v V_{xx} + \pi\left(x\sigma\rho v V_{xv} + x\bar{\lambda}v V_x\right) + xrV_x.$$

Then, we can clearly see that the function $g$ is a polynomial in $\pi$ with the highest degree term, $\frac{1}{2}x^2 v V_{xx} < 0$, being negative. Thus, given fixed admissible $(t, x, v)$, the maximum of $g$ is attained at the point $\pi^\star$ where its derivative vanishes, $g'(\pi^\star) = 0$,. This leads us to:

$$\pi^\star x V_{xx} + \sigma\rho V_{xv} + \bar{\lambda}V_x = 0,$$

$$\boxed{\pi^\star(t,x,v) = -\bar{\lambda}\frac{V_x(t,x,v)}{xV_{xx}(t,x,v)} - \sigma\rho\frac{V_{xv}(t,x,v)}{xV_{xx}(t,x,v)}.} \tag{3.1}$$

Inserting $\pi^\star$ into the equation 3.3, we obtain:

$$0 = v\frac{1}{2}\frac{\left(\bar{\lambda}V_x + \rho\sigma V_{xv}\right)^2}{V_{xx}} - v\frac{\left(\bar{\lambda}V_x + \rho\sigma V_{xv}\right)\left(\sigma\rho V_{xv} + \bar{\lambda}V_x\right)}{V_{xx}} + xrV_x + V_t + \frac{1}{2}\sigma^2 vV_{vv} + \kappa(\theta - v)V_v,$$

$$\boxed{0 = V_t + \kappa\theta V_v + xrV_x + v\left(\frac{1}{2}\sigma^2 V_{vv} - \kappa V_v - \frac{1}{2}\frac{\left(\bar{\lambda}V_x + \sigma\rho V_{xv}\right)^2}{V_{xx}}\right).} \tag{3.2}$$

To find a solution, we use the separation ansatz,

$$V(t,x,v) = \frac{x^p}{p}h(t,v), \quad h(T,v) = 1$$

Then,

$$\pi^\star(t,x,v) = -\bar{\lambda}\frac{x^{p-1}h(t,v)}{(p-1)x^{p-1}h(t,v)} - \sigma\rho\frac{x^{p-1}h_v(t,v)}{(p-1)x^{p-1}h(t,v)},$$

$$\boxed{\pi^\star(t,x,v) = \frac{\bar{\lambda}}{1-p} + \frac{\sigma\rho}{p-1}\frac{h_v(t,v)}{h(t,v)}} \tag{3.3}$$

In the same respect, we insert the separation ansatz into 3.2, and multiply by $(x^p/p)^{-1}$ both sides of the equality,

$$0 = h_t + \kappa\theta h_v + rph + v\left(\frac{1}{2}\sigma^2 h_{vv} - \kappa h_v - \frac{1}{2}p\frac{x^{2p-2}}{x^{2p-2}}\frac{\left(\bar{\lambda}h + \rho\sigma h_v\right)^2}{(p-1)h}\right)$$

$$\boxed{0 = h_t + \kappa\theta h_v + rph + v\left(\frac{1}{2}\sigma^2 h_{vv} - \kappa h_v + \frac{p}{2}\frac{\left(\bar{\lambda}h + \rho\sigma h_v\right)^2}{(1-p)h}\right)} \tag{3.4}$$

The structure of the latter equation implies that $h$ is exponentially affine,

$$h(t,v) = e^{a(T-t)+b(T-t)v}, \quad a(0) = b(0) = 0.$$

Inserting the exponentially affine structure of $h$ into 3.4, we obtain,

$$\begin{aligned} 0 = &-(a'(T-t) + b'(T-t)v)h + \kappa\theta b(T-t)h + prh \\ &+ v\left(\frac{1}{2}\sigma^2 b^2(T-t)h - \kappa b(T-t)h + \frac{p}{2}\frac{(\bar{\lambda}h)^2 + (\rho\sigma b(T-t)h)^2 + 2\bar{\lambda}h^2\sigma\rho b(T-t)}{(1-p)h}\right), \end{aligned}$$

Where we have omit the argument $(t, v)$ in $h$ and its partial derivatives. Denoting $\tau = T - t$, we re-arrange the latter equation,

$$\boxed{0 = -a'(\tau)h + \kappa\theta b'(\tau)h + prh + v\left(-b'(\tau)h + b^2(\tau)\left[\frac{1}{2}\sigma^2 h + \frac{p\sigma^2\rho^2 h}{2(1-p)}\right] + b(\tau)\left[-\kappa h + \frac{\bar{\lambda}h\sigma\rho p}{1-p}\right] + \frac{p\bar{\lambda}^2 h}{2(1-p)}\right.} \tag{3.5}$$

This equation holds for all $\tau$ and $v$, so let us consider it for a fixed choice of $\tau$. Since the equation holds for all values of $v$ the coefficient of $v$ must be equal to zero. Thus we have the equation,

$$0 = -b'(\tau)h + b^2(\tau)h\left[\frac{1}{2}\sigma^2 + \frac{p\sigma^2\rho^2}{2(1-p)}\right] + b(\tau)\left[-\kappa + \frac{\bar{\lambda}\sigma\rho p}{1-p}\right] + \frac{p\bar{\lambda}^2}{2(1-p)}$$

By isolating $b'$ and dividing by $h$, we obtain:

$$\boxed{b'(\tau) = b^2(\tau)\frac{1}{2}\left[\sigma^2 + \frac{p\sigma^2\rho^2}{1-p}\right] - b(\tau)\left[\kappa - \frac{\bar{\lambda}\sigma\rho p}{1-p}\right] + \frac{p\bar{\lambda}^2}{2(1-p)}} \tag{3.6}$$

Since the $v$-term in is zero we see that the other term must also vanish, giving us the equation,

$$\boxed{a'(\tau) = \kappa\theta b'(\tau) + pr} \tag{3.7}$$

Finally, we need to verify the assumption that $V_{xx}$ is negative for every triplet $(t, x, v)$. This is trivially verified by recalling what was mentioned in the introduction of this section, namely that the risk-aversion parameter $p \in (-\infty, 0) \cup (0, 1)$. Thus,

$$V_{xx}(t, x, v) = \underbrace{(p-2)}_{<0} x^{p-2}h(t, v)$$

and since $h$ is a real exponential, it is positive, and $x^{p-2}$ is also positive; otherwise, economically, it would be absurd. Thus, returning to the expression of $\pi^\star$ in Equation 3.3, since $h_v(t, v) = b(T-t)h(t, v)$, with $b$ given expression in the subject,

$$\boxed{\pi^\star(t) = \frac{\bar{\lambda}}{1-p} + \frac{\sigma\rho}{1-p}b(T-t).} \tag{3.8}$$

## 3.3 Numerical results

We have implemented four different strategies :

- So-called naive strategies defined by the percentage $\alpha\%$ of the portfolio placed into the bank account, $\pi_{50\%}$, $\pi_{70\%}$, $\pi_{100\%}$,

- The optimal strategy $\pi^\star$,

The following table summarises the results of the asset allocation strategies under a Heston model with 1000 simulations.

| Allocation strategy | Mean PnL (in \$) | Std PnL (in \$) | Mean-std ratio | $\bar{\Delta}$ |
|---|---|---|---|---|
| **Naive 50%** | 5.42 | 12.51 | 0.43 | 0.15 |
| **Naive 70%** | 5.31 | 7.54 | 0.70 | 0.06 |
| **Naive 100%** | 5.13 | 0.0 | | 0.0 |
| **Optimal** | 5.16 | 1.29 | 4.00 | 0.0 |

***Table 3.1:*** *Summaries of statistics regarding the different strategies. We have computed the mean and the standard deviation of the 1000 PnLs we have obtained. $\bar{\Delta}$ is computed as the mean of consecutive variations in the portfolio. The Mean-std ratio is the ration between the empirical mean and the empirical standard deviation.*

Firstly, from Table 3.1, it is clear that as the percentage allocated to the risk-free asset increases, both the mean PnL and the variance of the portfolio's returns decrease. This trend is particularly noticeable when comparing the naive strategies. For instance, the $\pi_{50\%}$ strategy, where 50% of the portfolio is invested in the risk-free asset, shows a mean PnL of 5.42 with a relatively high standard deviation of 12.51, resulting in a mean-std ratio of 0.43. This indicates that although the returns are higher on average, they come with significant volatility.

In contrast, the $\pi_{70\%}$ strategy, which allocates 70% to the risk-free asset, achieves a slightly lower mean PnL of 5.31, but with a substantially reduced standard deviation of 7.54. This reduction in volatility results in a better mean-std ratio of 0.70, suggesting a more favourable risk-return trade-off compared to the $\pi_{50\%}$ strategy.

The $\pi_{100\%}$ strategy, which places the entire portfolio in the risk-free asset, produces the lowest mean PnL at 5.13. However, as expected, this strategy has zero volatility, reflected by the standard deviation of 0.0, indicating no risk but also limited returns due to the absence of exposure to the risky asset. The lack of a mean-std ratio for this strategy is due to the absence of variability in the returns.

Finally, the optimal strategy $\pi^\star$ demonstrates a compelling balance between risk and return. While the mean PnL of 5.16 is only marginally higher than that of the $\pi_{100\%}$ strategy, the standard deviation is 1.29, significantly lower than the naive strategies. This results in a notably high mean-std ratio of 4.00, indicating that the optimal strategy not only reduces risk but also provides a relatively high return per unit of risk. This highlights the effectiveness of the optimal strategy in achieving an efficient portfolio allocation.

In summary, as shown in Table 3.1, the naive strategies reveal a trade-off between risk and return: increasing the allocation to the risk-free asset reduces volatility but also lowers the potential return. The optimal strategy, however, provides a superior risk-adjusted return, as indicated by its much higher mean-std ratio, demonstrating the

benefits of a more sophisticated asset allocation approach under the Heston model.

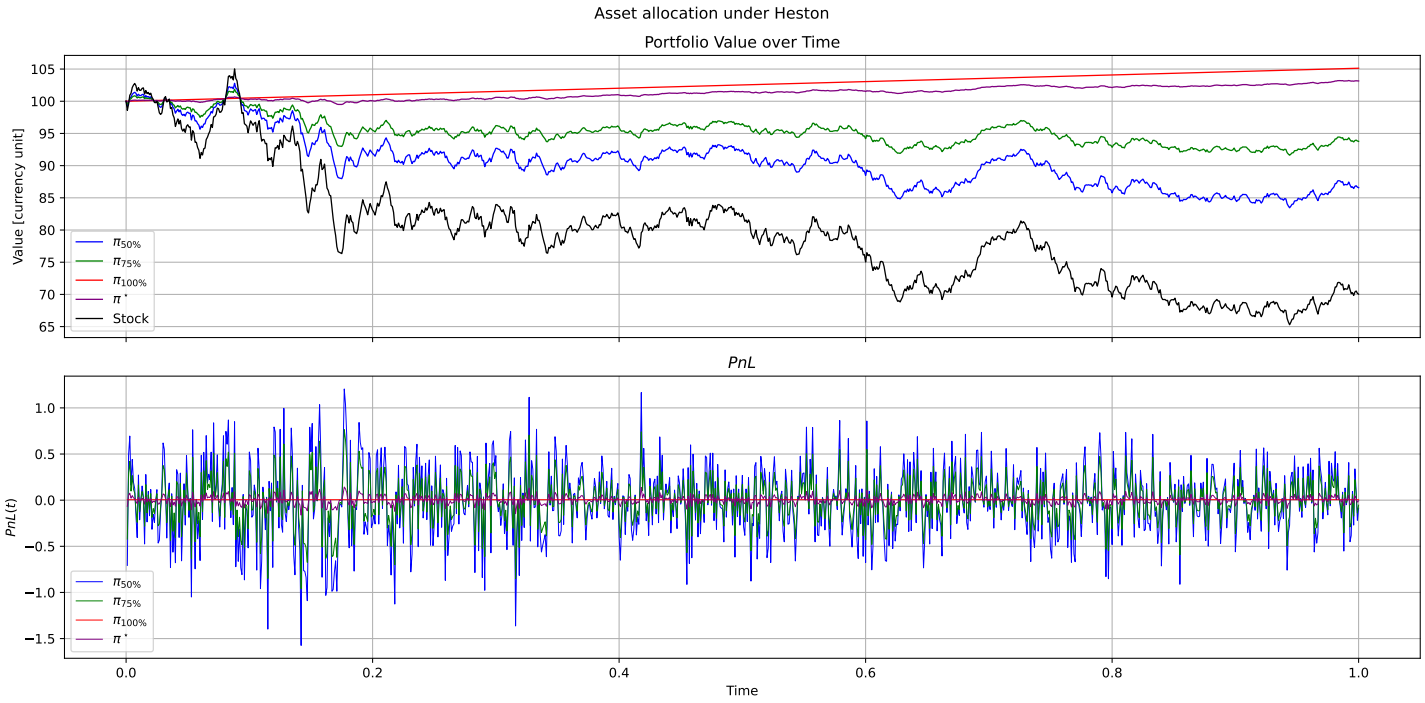


***Figure 3.1:*** *Example of the allocations on a simulated path*