ASSIGNMENT 3 MATH60609A

QUADRATURE & BLACK-SCHOLES

Shihao Tong[*]

CONTENTS

1 DIRECTIONAL DERIVATIVE AND LINE SEARCH FUNCTION

First recall the definition of directional derivative. Let $\mathbf{x} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^n$. Also assume a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$. Assume some sufficient (may not be necessarily, unchecked) conditions that function f and all components of its first derivative are continuous. Then the directional derivative is defined to be

$$f'(\mathbf{x};\mathbf{v}) \equiv \frac{d}{d\mathbf{x}} f(\mathbf{x};\mathbf{v}) = \lim_{\lambda \to 0^+} \frac{f(\mathbf{x}+\lambda\mathbf{v}) - f(\mathbf{x})}{\lambda} = \mathbf{v}^{\mathrm{T}} \nabla f(\mathbf{x})$$

where the second equality is guaranteed by the sufficient conditions. Then recall the target function for linear search

$$\Theta(\lambda;\mathbf{v},\mathbf{x}) \equiv f(\mathbf{x}+\lambda\mathbf{v})$$

which is a function of λ given a known point $\mathbf{x}$ and direction vector $\mathbf{v}$. To show the potential relationship assume the directional vector used in both directional derivative and linear search function are the same vector $\mathbf{v}$. Taking

[*]shihao.tong@hec.ca

derivative w.r.t to λ for Θ we get

$$\begin{aligned}\frac{d}{d\lambda}\Theta(\lambda;\mathbf{v},\mathbf{x}) &= f'(\mathbf{x}+\lambda\mathbf{v})\frac{d}{d\lambda}[\mathbf{x}+\lambda\mathbf{v}] \\ &= f'(\mathbf{x}+\lambda\mathbf{v})\mathbf{v}^{\mathrm{T}} \\ &= \mathbf{v}^{\mathrm{T}}\nabla f(\mathbf{x}+\lambda\mathbf{v})\end{aligned}$$

where the fist equality is given by chain rule and the last equality is due to inner product $\mathbf{x}^{\mathrm{T}}\mathbf{v} = \mathbf{v}^{\mathrm{T}}\mathbf{x}$. Then the relationship could be

$$\frac{d}{d\lambda}\Theta(\lambda;\mathbf{v},\mathbf{x})\bigg|_{\lambda=0} = f'(\mathbf{x};\mathbf{v})$$

or taking limit on the LHS as $\lambda \longrightarrow 0^+$ also works.

2 USING QUADRATURE TO REPLICATE THE BLACK SCHOLES MODEL

2.1 *Gaussian Hermite*

The target function to evaluate is

$$c(\mathrm{S},\mathrm{T},\mathrm{K}) = \mathrm{S}\mathrm{N}(d_1) - \mathrm{K}e^{-r\mathrm{T}}\mathrm{N}(d_2)$$

where S is the stock price at $t = 0$, T is the time at which we want to compute, K is the strike price and c is the price of European call option. The N is the CDF of a standard normal distribution which is

$$\mathrm{N}(d_1) = \Phi(d_1) = \mathrm{P}r(\mathrm{X} \le d_1)$$

where

$$\begin{aligned} d_1 &= \frac{\ln\frac{\mathrm{S}}{\mathrm{K}} + (r + \frac{\sigma^2}{2})\mathrm{T}}{\sigma\sqrt{\mathrm{T}}} \\ d_2 &= d_1 - \sigma\sqrt{\mathrm{T}}\end{aligned}$$

In order to compute $\mathrm{N}(d)$, we need Gaussian Hermite quadrature (GHQ). The target integral is

$$\begin{aligned}\mathrm{N}(d_1) = \Phi(d_1) &= \int_{-\infty}^{d_1} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x)^2}\,\mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d_1} e^{-(\frac{x}{\sqrt{2}})^2}\,\mathrm{d}x\end{aligned}$$

Recall the GHQ is defined on $(-\infty,\infty)$ however our CDF is an definite integral cut off at some fixed point d_1. In order to use GHQ there are two ways to deal with this issue. The first one is to compute

$$\frac{1}{2}\int_{\infty}^{\infty} f(x)\mathrm{dx} \pm \int_{d_1}^{0} f(x)\mathrm{dx}$$

(the ± sign depends on the sign of d which is possible to be negative) where the second integral can be approximated by any of the quadrature rules. The second one, also the one we pick is to customize the integral boundary to match $(-\infty, \infty)$. This approach is exactly an application of change of variable. The logic is as follows.

The aim of changing variable is to adjust the intergrand function so that the upper bond d_1 becomes ∞ which, in a more mathematical way, is to find a function $\phi(y)$ such that $\phi(d) = \infty$. This is not doable since in the mean while we want $\phi(y)$ to be differentiable since recall the form of change of variable

$$\int_{a}^{b} f(y)\mathrm{dy} = \int_{\phi(a)}^{\phi(b)} f(\phi^{-1}(x))\phi^{-1\prime}(x)\mathrm{dx}$$

Thus the only way this can work is to consider a limit behaviour of the function $\phi(y)$. Formally it is

$$\int_{-\infty}^{d_1} f(y)\mathrm{dy} = \lim_{y^+\to d_1, y^-\to -\infty} \int_{\phi(y^-)}^{\phi(y^+)} f(\phi^{-1}(x))\phi^{-1\prime}(x)\mathrm{dx} = \int_{-\infty}^{\infty} f(\phi^{-1}(x))\phi^{-1\prime}(x)\mathrm{dx}$$

So we need to find a function s.t

$$\lim_{y\to d_1} \phi(y) = \infty, \lim_{y\to -\infty} \phi(y) = -\infty$$

Then consider the natural log function which is not defined at $x = 0$ but has potentially a desired limit behaviour. Thus, let

$$x = \phi(y) = -\ln^{(d_1 - y)}$$

which then implies

$$y = \phi^{-1}(x) = d - e^{-x}, \;\; \mathrm{dy} = \phi^{-1\prime}(x)\mathrm{dx} = e^{x}\mathrm{dx}$$

Then in our case

$$\begin{aligned}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d_1} e^{-(\frac{x}{\sqrt{2}})^2}\,\mathrm{dx} &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \exp[-\frac{1}{2}(d-e^x)^2]*e^2\mathrm{dx}\\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \underbrace{\exp[-\frac{1}{2}(d-e^x)^2]*e^2*e^{x^2}}_{f(x)} * \underbrace{e^{-x^2}}_{w(x)}\,\mathrm{dx}\end{aligned}$$

where last line matches the form of integration that is desirable for quadrature which is

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)w(x)\mathrm{dx} \approx \frac{1}{\sqrt{2\pi}}\sum_{i=1}^{n} f(x_i)w_i$$

Code Comments Our code implement the summation above. Recall the nodes $\{x_i\}_{i=1}^n$ are obtained by solving the roots of the Hermite polynomial of power n. We find a function written by David Terr and Raytheon [1] which gives the coefficients of Hermite polynomial of order n. Then the function *root()* is used to find all the roots as nodes $\{x_i\}_{i=1}^n$. Then for the weights $\{w_i\}_{i=1}^n$, we used the expression from Wikipedia

$$w_i = \frac{2^{n-1}n!\sqrt{\pi}}{n^2\left[\mathrm{H}_{n-1}(x_i)\right]^2}$$

where H_{n-1} represents the Hermite polynomial of order $(n-1)$.

2.2 *Gaussian Laguerre*

For the question 2.2, the Gaussian Laguerre quadrature (GLQ) is used here. The logic is the same as GHQ while the weight function $w(x)$ now is $\exp[-x]$. The form of integrand is derived as follows

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d_1} \exp[-\frac{1}{2}y^2]\mathrm{dy} = 1 - \frac{1}{2\pi}\int_{d_1}^{\infty} \exp[-\frac{1}{2}y^2]\mathrm{dy}$$

Consider changing of variable to adjust the lower bound form d_1 to 0, let

$$x = y - d_1 \implies \mathrm{dx} = \mathrm{dy},\;\; y = x + d_1$$

then

$$= 1 - \frac{1}{\sqrt{2\pi}} \int_0^\infty \exp[-\frac{1}{2}(x+d_1)^2]\mathrm{dx}$$

$$= 1 - \frac{1}{\sqrt{2\pi}} \int_0^\infty \underbrace{\exp[-\frac{1}{2}(x+d_1)^2]e^x}_{f(x)} \underbrace{e^{-x}}_{w(x)} \mathrm{dx}$$

where the last line matches the form of (GLQ) which is

$$\int_0^\infty f(x)w(x)\mathrm{dx} \approx \sum_{i=1}^{n} f(x_i)w_i$$

Code Comments Different from Q2-1, we found a function **GaussLaguerre_2** written by Geert Van Damme [2] which take the order n as input and explicitly return the nodes $\{x_i\}_i^n$ with corresponding weights $\{w_i\}_{i=1}^n$. Then take the inner product of weights vector and $\{f(x_i)\}_{i=1}^n$ vector to obtain the summation.

2.3 *Numerical Simulation*

In this session, we conducted numerical simulations to compare the sensitivity to different variables of two quadrature in approximating the Black-Scholes call price.

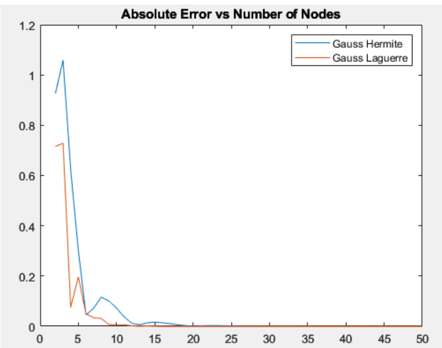

Figure 1: Price absolute error v.s. # nodes by two quadratures

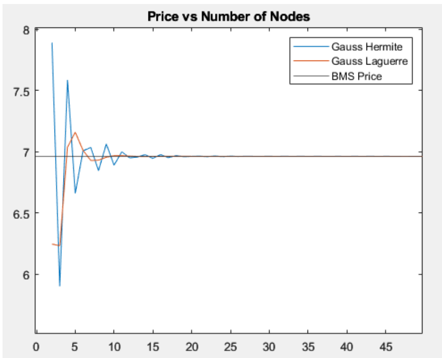

Figure 2: Price v.s. # nodes by two quadratures

Sensitivity to number of nodes We compare the converging rate of two different quadrature implemented in Q2-1 and Q2-2. Figure 2 shows the value of the call price obtained by approximation at various number of nodes comparing to the price offered by **blsprice()** in Matlab. Figure 1 shows the value of absolute error which is

$$e_i = \left|c_i - \widehat{c_i}\right|$$

at different number of nodes. It is clear that the GLQ converges to the true value faster than GHQ.

One other thing worth notice is that the GHQ has approximated values converging in a way that goes beyond and below the true value as number of nodes increase while Laguerre's approach is more consistent in a way that several values will be beyond or below the true value in a row. For Gaussian hermit, it overestimate the true price when number of nodes is even and underestimate when number of nodes is odd.

Sensitivity to interest rate Here we compare the performance of two quadrature at different interest rate r at the same number of nodes. Our trails here take the number of nodes up to 50 since the function (either from others [1] [2] or written by ourselves) will generate unreliable results like NaN or complex nodes when power (which equals the number of nodes) is big (N = 80 in our case). This can result from the technique utilized in roots searching process which is out of the scope of this course.

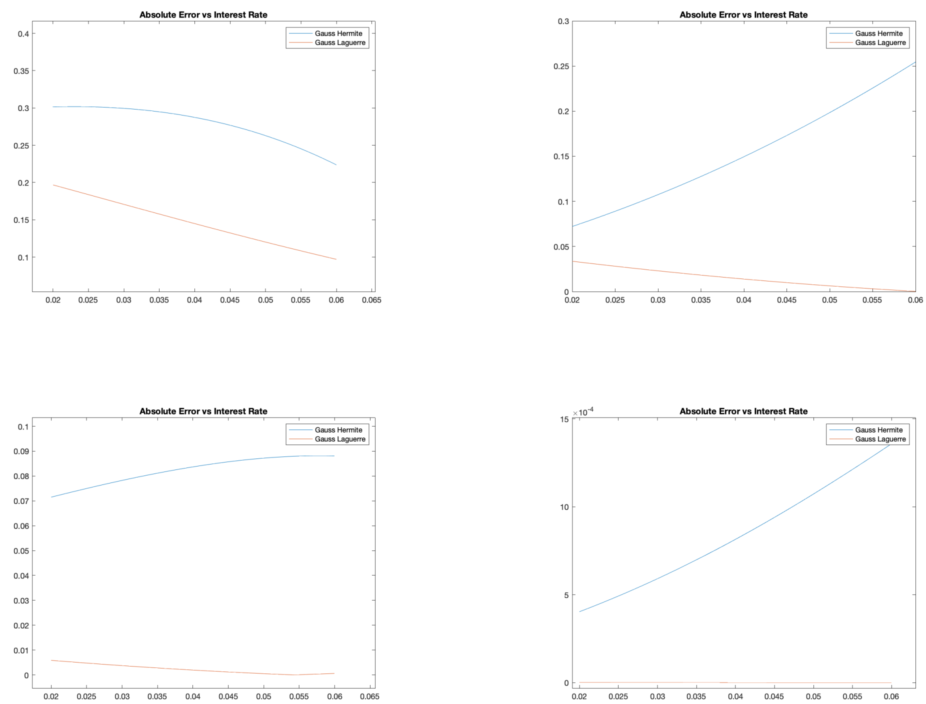

Figure 3: From left to right, top to bottom, absolute error v.s. interest rate by GHQ and GLQ at number of nodes N = 5, 7, 10 and 30. The scale of interest $r \in (0.02, 0.06)$ *incrementally at 0.01.*

Figure 3 shows that given the same number of nodes with $r \in (0.02, 0.06)$, the absolute error obtained by GHQ is always higher than GLQ which is as expected from our previous discussion. What is more interesting is that the approximation error increases as r increases using GHQ while it decreases using GLQ except for some small N like 5.

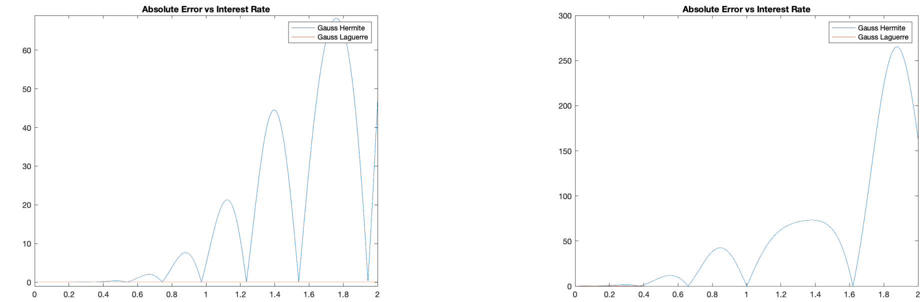

Figure 4: Left: absolute error v.s. interest rate by GHQ and GLQ at number of nodes N = 10; Right: absolute error v.s. interest rate by GHQ and GLQ at number of nodes N = 30. Both on $r \in (0, 2)$

Figure 4 shows that when r is getting bigger, especially beyond 0.4 (this threshold can be higher as N increases) the GHQ is no longer reliable. Its absolute error shows periodic behaviour. However GLQ is still perfectly approximate the true price no matter how big the interest rate is.

Sensitivity to K and S Our approach here is still fix number of nodes and change different combination of S and K. We first use the command **meshgrid()** to get different combination of (S, K) where S and K are two matrix. Then we evaluate the price at each (S_i, K_i) and plot it against the absolute error. The graphs are shown in figure 5.

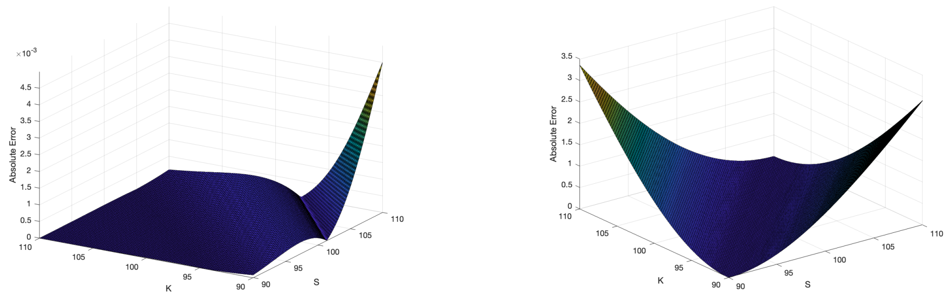

Figure 5: Left: absolute error v.s. price S and strike price K by GHQ at number of nodes N = 40; Right: absolute error v.s. price S and strike price K by GLQ at number of nodes N = 40.

The simulation shown above involves S and K of resolution of 0.2 both from 90 t0 110, so a 110×110 grid is made. Overall, comparing the two plot in figure 5, we found that at a fixed number of nodes, the significance level of absolute error generated by GHQ is much lower (e.g. 10^{-3}) than by GLQ (e.g. 10^{-1}) almost at all different combination of S and K. There performance of GHQ and GLQ can be very similar when K and S are close to each other however when they are even just 1$ apart, the error from GLQ increases dramatically.

Second, for GLW approach, the graph shows that the absolute error decreases when $K < S$ while it increases when $K > S$. Its absolute error also shows symmetry by the diagonal of S and k (e.g. the line where $S = K$). However for the GHQ approach, the error seems to get changed upward at the line $S = 100+K$. Intuitively both the phenomena should come from some multiplicative effect of S and K. Since consider the way that the call price is computed. Let's use * to represent "estimated by quadrature"

$$c^*(S, T, K) = SN^*(d_1) - Ke^{-rT}N^*(d_2)$$

At fixed number of nodes, no matter which type of quadrature, $N(d)$ does not change. Then when K or S changes, either one at a time or simultaneously, the error from $N(d) - N^*(d)$ will be inflated by the multiplier K and S.