Computational Finance
FIN-472
Homework 5 - Solutions

October 20, 2017

Exercise 1: The *Ornstein Uhlenbeck process* is defined as a solution of the Stochastic Differential Equation (SDE)

$$dX_t = \kappa(\theta - X_t)dt + \lambda dW_t. \tag{1}$$

If we let

$$v(t,x) := \mathbb{E}[\exp(i\nu X_T)|X_t = x]$$

then v satisfies the PDE

$$v_t + \mathcal{G}v = 0 \tag{2}$$

with terminal condition $v(T,x) = \mathrm{e}^{i\nu x}$, where

$$\mathcal{G}v = \kappa(\theta - x)v_x + \frac{\lambda^2}{2}v_{xx}.$$

Suppose that

$$v(t,x) = \exp(\varphi(T-t,\nu) + \psi(T-t,\nu)x). \tag{3}$$

a) Deduce a system of Ordinary Differential Equations (ODEs) for the functions φ and ψ.

Solution: The PDE (2) is solved by means of the ansatz (3). From the educated guess (3) we find that

$$\begin{aligned} v_t(t,x) &= v(t,x)\big(-\varphi_\tau(T-t,\nu) - x\psi_\tau(T-t,\nu)\big) \\ v_x(t,x) &= v(t,x)\,\psi(T-t,\nu) \quad \text{and} \\ v_{xx}(t,x) &= v(t,x)\,\psi(T-t,\nu)^2 \end{aligned}$$

where φ_τ and ψ_τ are the first-order derivatives of φ and ψ with respect to the time variable. Upon substituting these relations into the PDE (2) we obtain that

$$0 = v(t,x)\Big[\Big(-\varphi_\tau(T-t,\nu)+\kappa\theta\psi(T-t,\nu)+\frac{\lambda^2}{2}\psi(T-t,\nu)^2\Big)+x\Big(-\psi_\tau(T-t,\nu)-\kappa\psi(T-t,\nu)\Big)\Big]$$

for all $t \in [0,T]$ and all $x \in \mathbb{R}$. Therefore, as $v \neq 0$ on $[0,T] \times \mathbb{R}$ by ansatz (3), we find that the functions φ and ψ satisfy the following ODEs for $\tau := T - t \in [0,T]$

$$\begin{aligned} 0 &= -\varphi_\tau(\tau,\nu) + \kappa\theta\psi(\tau,\nu) + \frac{\lambda^2}{2}\psi(\tau,\nu)^2 \\ 0 &= -\psi_\tau(\tau,\nu) - \kappa\psi(\tau,\nu). \end{aligned} \tag{4}$$

The terminal condition $v(T,x) = \mathrm{e}^{i\nu x}$ yields initial conditions for φ and ψ:

$$\varphi(0,\nu) = 0 \quad \text{and} \quad \psi(0,\nu) = i\nu. \tag{5}$$

b) Solve this system and write explicitly the form of the characteristic function of X_t given $X_0 = x$.

Solution: The unique solution to the second ODE in (4) that additionally satisfies the initial condition (5) is given by

$$\psi(\tau,\nu) = i\nu \mathrm{e}^{-\kappa\tau}, \quad \tau \in [0,T].$$

Then the ODE for φ is solved by integration:

$$\varphi(\tau,\nu) = \int_0^\tau \left\{ \kappa\theta\psi(s,\nu) + \frac{\lambda^2}{2}\psi(s,\nu)^2 \right\} \mathrm{d}s = i\nu\theta(1-\mathrm{e}^{-\kappa\tau}) - \frac{\nu^2\lambda^2}{4\kappa}(1-\mathrm{e}^{-2\kappa\tau}).$$

This in particular yields that a solution to the PDE (2) that satisfies the required boundary condition is given by

$$v(t,x) = \exp\left\{ i\nu\Big(\theta\big(1-\mathrm{e}^{-\kappa(T-t)}\big) + x\mathrm{e}^{-\kappa(T-t)}\Big) - \frac{\nu^2}{2}\Big(\frac{\lambda^2}{2\kappa}\big(1-\mathrm{e}^{-2\kappa(T-t)}\big)\Big) \right\}. \tag{6}$$

Note that v given by (6) is a bounded function in $(t,x) \in [0,T] \times \mathbb{R}$ for every fixed ν. Therefore $\{\int_0^t v_x(s,X_s)\mathrm{d}W_s\}_{t\in[0,T]}$ is a martingale and, by applying the martingale property and the definition of $v(t,x)$, it follows that

$$v(0,x) = \mathbb{E}\big[v(T,X_T)\big|X_0 = x\big] = \mathbb{E}\big[\mathrm{e}^{i\nu X_T}\big|X_0 = x\big]$$

i.e. the solution (6) of the PDE (2) indeed coincides with the characteristic function of X_T.

c) Deduce that X_t is normally distributed. Write explicitly the mean and variance of X_t.

Solution: By b) we obtain that the characteristic function of X_t given $X_0 = x$ reads

$$\mathbb{E}\big[\mathrm{e}^{i\nu X_t}\big|X_0 = x\big] = \exp\left\{ i\nu\Big(\theta\big(1-\mathrm{e}^{-\kappa t}\big) + x\mathrm{e}^{-\kappa t}\Big) - \frac{\nu^2}{2}\Big(\frac{\lambda^2}{2\kappa}\big(1-\mathrm{e}^{-2\kappa t}\big)\Big) \right\}.$$

From this representation by the one-to-one correspondence between characteristic functions and distributions we immediately deduce that X_t is normally distributed with

$$\mathbb{E}\big[X_t\big|X_0 = x\big] = \theta\big(1-\mathrm{e}^{-\kappa t}\big) + x\mathrm{e}^{-\kappa t} \quad \text{and} \quad \mathrm{Var}(X_t|X_0 = x) = \frac{\lambda^2}{2\kappa}\big(1-\mathrm{e}^{-2\kappa t}\big).$$

d) Using Itô's formula solve (1) explicitly and explain why this is consistent with the results in the previous part.

Solution: Let $X_t = \theta + f(t,Y_t)$ where $f(t,x) = \mathrm{e}^{-\kappa t}x$ and $Y_t = X_0 - \theta + \int_0^t \lambda \mathrm{e}^{\kappa s}\, dW_s$. Then by Itô's formula

$$\begin{aligned} dX_t &= -\kappa \mathrm{e}^{-\kappa t} Y_t\, dt + \mathrm{e}^{-kt} dY_t \\ &= \kappa(\theta - X_t)\, dt + \lambda\, dW_t. \end{aligned}$$

From this computation we deduce that

$$X_t = \theta + \mathrm{e}^{-\kappa t}(X_0 - \theta) + \int_0^t \lambda \mathrm{e}^{\kappa(s-t)}\, dW_s$$

is a solution of the SDE and X_t is a Gaussian random variable with mean and variance as in the previous part. Recall, that if a function f is square-integrable on $[0,t]$, then $\int_0^t f(s)dW_s$ is normally distributed with mean 0 and variance $\int_0^t f^2(s)ds$.

Exercise 2: This exercise proposes a technique to price out-of-the-money options. Suppose that $r = 0$ and fix a maturity T. Without loss of generality assume that $S_0 = 1$. This can always be achieved by modeling prices with S/S_0 and scaling up by S_0 option prices. For a given log strike k denote by $c(k)$ and $p(k)$ the call and put prices, respectively. Define the function

$$z(k) = p(k)1_{k<0} + c(k)1_{k>0}. \tag{7}$$

a) Write a formula for the Fourier transform of $z(k)$ in terms of the characteristic function of $X_T = \log(S_T)$ with respect to the risk-neutral measure. Propose, through Fourier inversion, a pricing formula. *Hint:* $\mathbb{E}[S_T] = e^{rT}S_0 = 1$.

Solution: Let $\mu(dx)$ be the distribution of X_T and ϕ its characteristic function. We have

$$\begin{aligned}
\widehat{z}(\nu) &= \int_{\mathbb{R}} z(k)\mathrm{e}^{ik\nu}\, dk \\
&= \int_{-\infty}^{0}\int_{-\infty}^{k} (\mathrm{e}^k - \mathrm{e}^x)\mathrm{e}^{ik\nu}\mu(dx)\, dk + \int_0^\infty \int_k^\infty (\mathrm{e}^x - \mathrm{e}^k)\mathrm{e}^{ik\nu}\mu(dx)\, dk \\
&= -\int_{-\infty}^{0}\int_{0}^{x} (\mathrm{e}^k - \mathrm{e}^x)\mathrm{e}^{ik\nu}\, dk\mu(dx) + \int_0^\infty \int_0^x (\mathrm{e}^x - \mathrm{e}^k)\mathrm{e}^{ik\nu}\, dk\mu(dx) \\
&= \int_{-\infty}^{0} \left\{ \mathrm{e}^{ix(\nu-i)}\left(\frac{1}{i\nu} - \frac{1}{1+i\nu}\right) + \frac{1}{1+i\nu} - \frac{\mathrm{e}^x}{i\nu} \right\}\mu(dx) \\
&+ \int_{0}^{\infty} \left\{ \mathrm{e}^{ix(\nu-i)}\left(\frac{1}{i\nu} - \frac{1}{1+i\nu}\right) + \frac{1}{1+i\nu} - \frac{\mathrm{e}^x}{i\nu} \right\}\mu(dx) \\
&= \int_{\mathbb{R}} \left\{ \mathrm{e}^{ix(\nu-i)}\left(\frac{1}{i\nu} - \frac{1}{1+i\nu}\right) + \frac{1}{1+i\nu} - \frac{\mathrm{e}^x}{i\nu} \right\}\mu(dx) \\
&= \left\{ \phi(\nu - i)\left(\frac{1}{i\nu} - \frac{1}{1+i\nu}\right) + \frac{1}{1+i\nu} - \frac{1}{i\nu} \right\} \\
&= \frac{\phi(\nu-i) - 1}{i\nu - \nu^2}.
\end{aligned}$$

where we used that

$$1 = \mathbb{E}[S_T] = \int_{\mathbb{R}} e^x \mu(dx).$$

The proposed pricing formula (as long as the Fourier inversion formula holds) is

$$z(k) = \frac{1}{\pi}\int_0^\infty Re\left(\frac{\phi(\nu-i)-1}{i\nu-\nu^2}e^{-i\nu k}\right)\, d\nu.$$

b) Consider for $\alpha > 0$ the modified function

$$z_\alpha(k) = \sinh(\alpha k) z(k). \tag{8}$$

Write a formula for the Fourier transform of $z_\alpha(k)$ in terms of the characteristic function of $X_T = \log(S_T)$ with respect to the risk-neutral measure. Propose, through Fourier inversion, a pricing formula.

Solution: We have that

$$\begin{aligned}\widehat{z_\alpha}(\nu) &= \int_{\mathbb{R}} \frac{\mathrm{e}^{ik(\nu - i\alpha)} - \mathrm{e}^{ik(\nu + i\alpha)}}{2} z(k)\, dk \\ &= \frac{\widehat{z}(\nu - i\alpha) - \widehat{z}(\nu + i\alpha)}{2},\end{aligned}$$

with $\widehat{z}$ as in (a). The proposed pricing formula (as long as the Fourier inversion formula holds) is

$$z(k) = \frac{1}{\sinh \alpha k} \frac{1}{\pi} \int_0^\infty Re\left(\frac{\widehat{z}(\nu - i\alpha) - \widehat{z}(\nu + i\alpha)}{2} e^{-i\nu k} \right) d\nu.$$

c) Choose a particular affine stochastic model. Is there a difference in the behavior of the integrands appearing in the pricing formulas in a) and b)?

Solution: We take a VG model with $S = 1$, $r = 0$, $T = .0027$, $\alpha = 1.1$, $\sigma = .2$, $\nu = .12$, $\theta = .0027$ and consider the strike $K = 0.8$. The plots of the integrands are given in Figure 1. The blue line corresponds to the integrand in a). It is wide and oscillatory, in contrast to the red line, which corresponds to the integrand in b). For details see following Matlab file:

```
% Integrands in Hw 5 - Exercise 2c)
% Variance Gamma model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial price
S=1;
% Maturity
T=0.0027;
% Interest rate
r=0;

% VG parameters
gamma=0.12;
sigma=0.2;
theta=0.0027;

% Damping factor
alpha=1.1;

% Strike
K=0.8;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Characteristic function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b=@(u)(log(1-1i*theta*gamma.*u+sigma^2*0.5*gamma.*u.^2));
a=@(u)(exp(-(T/gamma)*b(u)));
phi=@(u)(exp(1i*u.*(r*T-log(a(-1i)))).*a(u));% see equations (41)--(44)
                                             % in lecture notes

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Auxiliary functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
phiaux=@(u)((phi(u-1i)./(1i*u-u.^2))+(1./(1+1i*u))-(exp(r*T)./(1i*u)));
%see Exercise 2a)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
phiaux2=@(u)(0.5*(phiaux(u-1i*alpha)-phiaux(u+1i*alpha)));
% see Exercise 2b)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Integrands
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

integrand=@(nu)(real((exp(-r*T)/pi)*phiaux(nu).*...
    exp(-1i*log(K)*nu)));

integrand2=@(nu)((1/sinh(alpha*log(K)))*real((exp(-r*T)/pi)...
    *phiaux2(nu).*exp(-1i*log(K)*nu)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fplot(integrand,[-150 150],'b');
hold on;
fplot(integrand2,[-150 150],'r');
hold off;
```

Exercise 3: Using the Variance Gamma model and the following parameters

$$S_0 = 100, \qquad \nu = 0.2, \qquad \theta = -0.14, \qquad r = 0.1, \qquad \sigma = 0.12$$

a) Compute the price of the European put options with maturities

$$T \in \{1/12, 1/6, 1/4, 1/2, 1, 2, 3, 5\}$$

and strikes

$$K \in \{50, 80, 90, 95, 100, 105, 110, 120, 150\},$$

using the FFT approach proposed by Carr and Madan (1999) with $e^{\alpha k}$ as damping factor. Here e^k denotes the strike of the option.

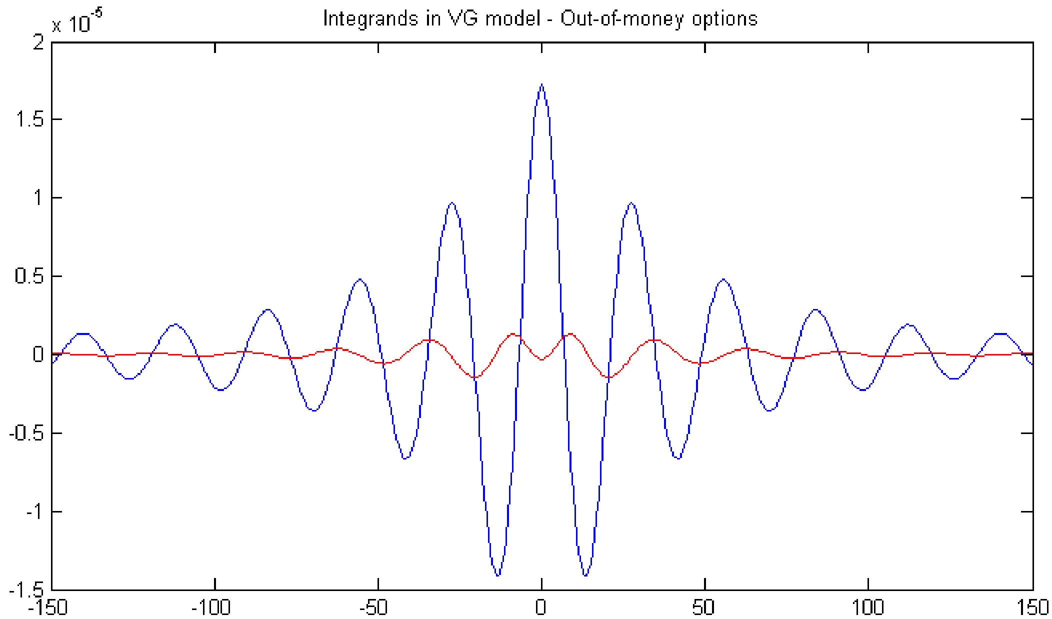

Figure 1: Plots of the integrands in Exercise 2 parts a) and b).

Solution: To compute put option prices one could use the put-call-parity formula (see Exercise 1(c) of Homework 1)

$$p(k) = c(k) - S_0 + K\mathrm{e}^{-rT}.$$

Alternatively, using damping factors $\mathrm{e}^{\alpha k}$ with $\alpha < -1$, one could use exactly the same formulas as the ones for call options (the integrand has the same form as for call options except that now $\alpha < -1$). The results by taking $\alpha = -2$, $\eta = 0.25$ and $N = 2^{12}$ in the same formula for call options are reported in Table 1 under FFT method.

b) Graph the implied volatility surface obtained in the previous part. This is the graph of the implied volatilities as a function of K and T. Recall that the implied volatility is the value $\sigma(K,T)$ such that

$$P(K,T) = P^{BS}(K,T;\sigma(K,T))$$

where $P(K,T)$ is the price of the put (in this case obtained by the FFT method) and $P^{BS}(K,T;\sigma^{BS})$ is the price of a put in the Black-Scholes model with parameter σ^{BS}. If you want, you can use the function `blsimpv` already implemented in Matlab.

Solution: See Figure 2.

c) Considering only the above options that are out-of-the-money, compute their price following the technique seen in Exercise 2b) with the sinh function as damping factor. Compare the results.

Solution: A simple verification shows that the formulas in Exercise 1 hold form arbitrary r and S_0 by taking

$$\widehat{z}(\nu) = \mathrm{e}^{-rT}\left(\frac{\phi(\nu - i)}{i\nu - \nu^2} + \frac{1}{1+i\nu} - \frac{\mathrm{e}^{rT}S_0}{i\nu}\right).$$

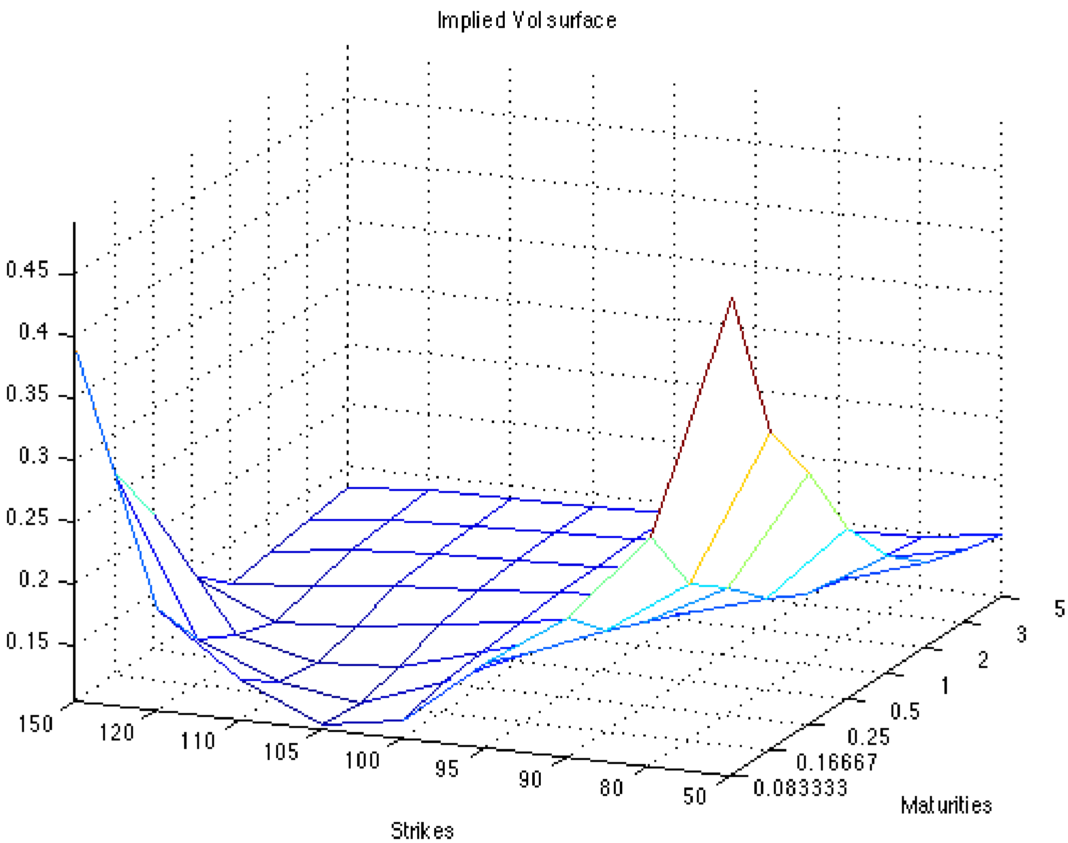

Figure 2: Implied volatility surface - Exercise 3 b.

The results are reported in Table 1 under FFT-out. Here $\alpha = 0.7$, $\nu = 0.25$ and $N = 2^{12}$.

d) Redo a) and b), but this time use the Simpson rule instead of the trapezoidal rule. Compare the results.

Solution: The results are reported in Table 2

Strikes	Methods	Maturities							
		1/12	1/6	1/4	1/2	1	2	3	5
50	FFT	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0010	0.0014
	FFT-Out	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0010	0.0014
80	FFT	0.0065	0.0154	0.0257	0.0595	0.1155	0.1603	0.1549	0.1069
	FFT-Out	0.0065	0.0154	0.0257	0.0595	0.1155	0.1603	0.1548	0.1069
90	FFT	0.0822	0.1606	0.2307	0.3879	0.5350	0.5489	0.4592	0.2741
	FFT-Out	0.0820	0.1606	0.2307	0.3879	0.5350	0.5486	0.4588	0.2741
95	FFT	0.2795	0.4771	0.6222	0.8777	1.0303	0.9296	0.7336	0.4135
	FFT-Out	0.2797	0.4774	0.6222	0.8778	1.0314	0.9285	0.7319	0.4136
100	FFT	0.9891	1.3637	1.5730	1.8373	1.8547	1.4963	1.1235	0.6030
105	FFT	4.2747	3.8373	3.6972	3.5418	3.1290	2.2980	1.6556	0.8530
110	FFT	9.1067	8.2753	7.5576	6.2445	4.9619	3.3818	2.3564	1.1742
120	FFT	19.0051	18.0204	17.0505	14.3313	10.5028	6.5449	4.3611	2.0733
150	FFT	48.7559	47.5214	46.2972	42.6852	35.7458	23.9279	15.9556	7.4236

Table 1: European put option prices in VG model. Trapezoid rule

Strikes	Methods	Maturities							
		1/12	1/6	1/4	1/2	1	2	3	5
50	FFT	-0.0001	-0.0001	-0.0001	-0.0000	0.0001	0.0005	0.0009	0.0014
	FFT-Out	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0010	0.0014
80	FFT	0.0064	0.0153	0.0256	0.0594	0.1154	0.1602	0.1548	0.1068
	FFT-Out	0.0065	0.0154	0.0257	0.0595	0.1155	0.1603	0.1548	0.1069
90	FFT	0.0821	0.1605	0.2306	0.3878	0.5349	0.5488	0.4591	0.27401
	FFT-Out	0.0820	0.1606	0.2307	0.3879	0.5350	0.5486	0.4587	0.2741
95	FFT	0.2794	0.4770	0.6221	0.8776	1.0302	0.9296	0.7336	0.4134
	FFT-Out	0.2797	0.4774	0.6222	0.8778	1.0314	0.9285	0.7319	0.4137
100	FFT	0.9890	1.3636	1.5729	1.8371	1.8546	1.4962	1.1234	0.6029
105	FFT	4.2746	3.8371	3.6971	3.5417	3.1289	2.2979	1.6555	0.8529
110	FFT	9.1066	8.2752	7.5574	6.2444	4.9618	3.3817	2.3563	1.1741
120	FFT	19.0050	18.0203	17.0504	14.3311	10.5027	6.5448	4.3610	2.0733
150	FFT	48.7557	47.5212	46.2970	42.6851	35.7456	23.9278	15.9555	7.4235

Table 2: European put option prices in VG model. Simpson's rule

Please find hereunder the needed Matlab files.

```
function [P beta tmp] = Call_FFT_VG(K, T, r, theta, gamma, sigma, S,alpha)
% Computes the value of call option option using Carr and Madan
% FFT technique for a VG model
%
% PARAMETERS:
%    Input:
%         K: strike price of the call option
%         T: maturity of the call option
%         r: risk free rate
%         theta, gamma, sigma: parameters of the VG model
%         S: initial stock price
%         alpha: damping factor
%    Output:
%        P: price of the call option with strike K and maturity T
%        beta: discretization log strikes
%        tmp: array of prices produced by algorithm

%%%%%%%%%%%%%%%%%%%%
% FFT setup
%%%%%%%%%%%%%%%%%%%%%
eta = 0.25;
N = 2^12;
lambda = 2*pi / (N * eta);
beta = -(N-1)*lambda/2+lambda.*(0:N-1); %discretization of log-strike
nu = eta .* (0:N-1); %discretization of integration variable

logStrikes = log(K/S);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Weights depending on the integration rule
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trapezoid_weights=ones(1,N);
trapezoid_weights(1)=0.5;
trapezoid_weights(N)=0.5;

Simpson_weights = repmat([2,4],1,N/2);
Simpson_weights(1) = 1;
Simpson_weights = Simpson_weights/3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Linear Interpolation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ind = min(max(1+floor( (logStrikes - beta(1)) / lambda ),1),N-1);
diff = logStrikes - beta(ind);


nu_tmp = (nu - 1i*(alpha+1));% Points where ch. function will be
                             % evaluated

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%characteristic function of VG model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b=@(u)(log(1-1i*theta*gamma.*u+sigma^2*0.5*gamma.*u.^2));
a=@(u)(exp(-(T/gamma)*b(u)));
phi=@(u)(exp(1i*u.*(r*T-log(a(-1i)))).*a(u));
cgf2 = phi(nu_tmp);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT algorithm; see equation (38) of lecture notes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
func = exp(-1i*beta(1)*nu) .* cgf2...
        ./ ((alpha + 1i*nu).*(1+alpha+1i*nu))...
        .* eta .* trapezoid_weights;

tmp = (exp(-alpha.*beta-r*T) ./ pi) .* real(fft(func));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Computation of prices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P = S*(tmp(ind) + diff .* (tmp(ind+1) - tmp(ind)) ./ lambda);
end
```

```
Strikes=[50;80;90;95;100;105;110;120;150]; % strikes
Mat=[1/12;1/6;1/4;1/2;1;2;3;5]; % maturities

[T,K] = meshgrid(Mat, Strikes);
nMat = length(Mat); nStrikes = length(Strikes);
d = length(K(:));
P=zeros(d,1); % matrix of prices using FFT scheme
IV=zeros(d,1); % matrix of implied volatilites

S=100;r=0.1; % initial price and int. rates
```

```
% parameters of the VG model
theta=-0.14;
nu=0.2;
sigma=0.12;


% damping factor alpha
alpha=-2;



for i=1:d

    P(i)=Call_FFT_VG(K(i), T(i), r, theta, nu, sigma, S,alpha);

    if P(i)<=0 % prices should be positive
     P(i)=eps;
    end

    if alpha>0 % call options
       IV(i)=blsimpv(S, K(i), r, T(i), P(i),[], 0, [], true);
    end

    if alpha<-1 % put options
       IV(i)=blsimpv(S, K(i), r, T(i), P(i),[], 0, [], false);
    end

end


IV = reshape(IV, nStrikes, nMat);
P = reshape(P, nStrikes, nMat);

mesh(IV);
axis tight;
set(gca,'XTickLabel',num2str(Mat));
set(gca,'YTickLabel',num2str(Strikes));
xlabel('Maturities');
ylabel('Strikes');
title('Implied Vol surface');
```

```
function [P tmp] = Call_FFT_VG_outmoney(K, T, r, theta, gamma, sigma, ...
    S,alpha)
% Computes the value of call option option using technique of Exercise 2b)
% of Homework 3 -- VG model
%
% PARAMETERS:
%    Input:
%          K: strike price of the call option
%          T: maturity of the call option
%          r: risk free rate
%          theta, gamma, sigma: parameters of the VG model
%          S: initial stock price
%          alpha: damping factor
```

```
%    Output:
%        P: price of the call option with strike K and maturity T
%        tmp: array of prices produced by algorithm

%%%%%%%%%%%%%%%%%%%%
% FFT setup
%%%%%%%%%%%%%%%%%%%%%
eta = 0.25;
N = 2^12;
lambda = 2*pi / (N * eta);
beta = -(N-1)*lambda/2+r*T+lambda.*(0:N-1); %discretization of log-strike
nu = eta .* (0:N-1); %discretization of integration variable

logStrikes = log(K/S);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Weights depending on the integration rule
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trapezoid_weights=ones(1,N);
trapezoid_weights(1)=0.5;
trapezoid_weights(N)=0.5;

Simpson_weights = repmat([2,4],1,N/2);
Simpson_weights(1) = 1;
Simpson_weights = Simpson_weights/3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Linear Interpolation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ind = min(max(1+floor( (logStrikes - beta(1)) / lambda ),1),N-1);
diff = logStrikes - beta(ind);

nu_tmp = (nu - 1i*(alpha));% points where function are evaluated
nu_tmp2= (nu + 1i*(alpha));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%characteristic function of VG model -- Integrand of Exercise 2b)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b=@(u)(log(1-1i*theta*gamma.*u+sigma^2*0.5*gamma.*u.^2));
a=@(u)(exp(-(T/gamma)*b(u)));
phi=@(u)(exp(1i*u.*(r*T-log(a(-1i)))).*a(u));
phiaux=@(u)((phi(u-1i)./(1i*u-u.^2))+(1./(1+1i*u))-(exp(r*T)./(1i*u)));
cgf2 = 0.5*(phiaux(nu_tmp)-phiaux(nu_tmp2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
func = exp(-1i*beta(1)*nu) .* cgf2...
        .* eta .* trapezoid_weights;

tmp = (exp(-r*T) ./ (pi*sinh(alpha*beta))) .* real(fft(func));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Computation of prices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
P = S*(tmp(ind) + diff .* (tmp(ind+1) - tmp(ind)) ./ lambda);
end
```

.