Computational Finance
FIN-472
Homework 6 - Solutions

October 27, 2017

Exercise 1: Suppose that $S = \exp(X)$ is the price of a financial asset and that spot interest rate is equal to r. We define, for T given, the share measure $\mathbb{P}^S$ by

$$\mathbb{P}^S(A) = \frac{\mathbb{E}[S_T \mathbf{1}_A]}{\mathbb{E}[S_T]},$$

where $\mathbf{1}_A(\omega) = 1$ if $\omega \in A$ and $\mathbf{1}_A(\omega) = 0$ otherwise. For any bounded random variable X we have that

$$\mathbb{E}^{\mathbb{P}^S}[X] = \frac{\mathbb{E}[S_T X]}{\mathbb{E}[S_T]}.$$

a) Deduce the following expression for the price of a put option $P(k)$ and a call option $C(k)$ with strike $K = \mathrm{e}^k$ and expiration date T:

$$\begin{aligned} P(k) &= \mathrm{e}^{k-rT}\mathbb{P}(X_T < k) - \mathrm{e}^{-rT}\mathbb{E}[S_T]\mathbb{P}^S(X_T < k), \\ C(k) &= \mathrm{e}^{-rT}\mathbb{E}[S_T]\mathbb{P}^S(X_T > k) - \mathrm{e}^{k-rT}\mathbb{P}(X_T > k). \end{aligned} \tag{1}$$

Solution: We have

$$\begin{aligned} P(k) &= \mathrm{e}^{-rT}\mathbb{E}[(\mathrm{e}^k - \mathrm{e}^{X_T})_+] \\ &= \mathrm{e}^{-rT}\mathbb{E}[(\mathrm{e}^k - \mathrm{e}^{X_T})\mathbf{1}_{\{X_T<k\}}] \\ &= \mathrm{e}^{k-rT}\mathbb{E}[\mathbf{1}_{\{X_T<k\}}] - \mathrm{e}^{-rT}\mathbb{E}[\mathrm{e}^{X_T}\mathbf{1}_{\{X_T<k\}}] \\ &= \mathrm{e}^{k-rT}\mathbb{P}(X_T < k) - \mathrm{e}^{-rT}\mathbb{E}[S_T]\frac{\mathbb{E}[S_T\mathbf{1}_{\{X_T<k\}}]}{\mathbb{E}[S_T]} \\ &= \mathrm{e}^{k-rT}\mathbb{P}(X_T < k) - \mathrm{e}^{-rT}\mathbb{E}[S_T]\mathbb{P}^S(X_T < k). \end{aligned}$$

A similar computation proves the expression for call options. Alternatively, we have by put-call parity (see Exercise 5(c) of Homework 1) that

$$\begin{aligned} C(k) &= P(k) + \mathrm{e}^{-rT}\mathbb{E}[S_T] - \mathrm{e}^{k-rT} \\ &= \mathrm{e}^{k-rT}(\mathbb{P}(X_T < k) - 1) - \mathrm{e}^{-rT}\mathbb{E}[S_T](\mathbb{P}^S(X_T < k) - 1) \\ &= \mathrm{e}^{-rT}\mathbb{E}[S_T]\mathbb{P}^S(X_T > k) - \mathrm{e}^{k-rT}\mathbb{P}(X_T > k). \end{aligned}$$

b) Let $\phi(\nu) = \mathbb{E}[\exp(i\nu X_T)]$ and $\phi^S(\nu) = \mathbb{E}^{\mathbb{P}^S}[\exp(i\nu X_T)]$, be the characteristic functions of X_T with respect to $\mathbb{P}$ and $\mathbb{P}^S$, respectively. Show that $\phi(\nu - i)$ is well-defined and

$$\phi^S(\nu) = \frac{\phi(\nu - i)}{\mathbb{E}[S_T]}.$$

Solution: Since $\mathbb{E}[S_T] = \mathrm{e}^{rT} S_0 < \infty$ we have that $\phi(\nu - i)$ is well-defined and

$$\begin{aligned}
\phi^S(\nu) &= \mathbb{E}^{\mathbb{P}^S}[\exp(i\nu X_T)] \\
&= \frac{\mathbb{E}^{\mathbb{P}}[S_T \exp(i\nu X_T)]}{\mathbb{E}[S_T]} \\
&= \frac{\mathbb{E}^{\mathbb{P}}[\exp(X_T)\exp(i\nu X_T)]}{\mathbb{E}[S_T]} \\
&= \frac{\mathbb{E}^{\mathbb{P}}[\exp(i(\nu - i)X_T)]}{\mathbb{E}[S_T]} \\
&= \frac{\phi(\nu - i)}{\mathbb{E}[S_T]}.
\end{aligned}$$

Exercise 2: Consider a Variance Gamma model with the following parameters

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

a) Make an appropriate choice of the discretization parameters η (for the numerical integration) and λ (for the log strikes) to 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 FrFFT.

Solution: With $\alpha = -2$ as damping factor, $\eta = 0.25$, $N = 2^{12}$, $\lambda = \frac{2}{N}$ we have the results reported in Table 1. Alternatively, put prices could be calculated from call prices by put-call-parity. A possible Matlab implementation can be found hereunder.

```
function [P tmp beta] = Call_FrFFT_VG(K, T, r, theta, gamma, sigma, ...
    S,alpha)
% Computes the value of call option option using FrFT technique in ...
    the 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 produces by algorithm

%%%%%%%%%%%%%%%%%%%%
% FrFT setup
%%%%%%%%%%%%%%%%%%%%%
eta = 0.25;
N = 2^12;
lambda = 2/N;
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);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FrFFT algo 1-- see equation (55) of lecture notes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alpha_C = eta * lambda / (2*pi);

tmp = exp(1i * alpha_C * pi * ((0:N).^2));
a_C = tmp(1:end-1);
a_bar_C = tmp(end:- 1:2);

y2_C = [a_C , a_bar_C];
y2_tild_C = fft(y2_C);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 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 characteristed function
                              % is 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);

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

```
%FrFFT algo 2-- see equation (56) of lecture notes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
func = exp(-1i*beta(1)*nu) .* cgf2...
        ./ ((alpha + 1i*nu).*(1+alpha+1i*nu))...
        .* eta .* trapezoid_weights;

y1_C = [func ./ a_C , zeros(1,N)];
y1_tild_C = fft(y1_C);
y3_C = y1_tild_C .* y2_tild_C;
y3_tild_C = ifft(y3_C);
tmp = exp(-alpha.*beta-r*T) ./ pi .* real(y3_tild_C(1:N) ./ a_C);

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

b) Compare your results with the ones obtained using the FFT in Exercise 3a) of Homework 3. What is the advantage of the FrFFT method?

Solution: With $T = 5$, the FrFFT produces prices shown in Figure 1. The FFT produces the prices shown in Figure 2. The relevant prices in this graph lie somewhere in the middle of the graph and due to the scale they are not even recognizable.

Exercise 3: The saddle point method to price derivatives addresses the following points (circle all the valid statements):

a) It is faster than the FFT and FrFFT algorithms to price derivatives

b) It uses a damping factor such that the integrand in the pricing formula has rapid descent

c) It is a better method than FFT to price out-of-the-money options

Solution (b) and (c) are true.

Strikes	Method	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
	FrFFT	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
	FrFFT	0.0065	0.0154	0.0257	0.0594	0.1154	0.1603	0.1549	0.1068
90	FFT	0.0822	0.1606	0.2307	0.3879	0.5350	0.5489	0.4592	0.2741
	FrFFT	0.0820	0.1604	0.2304	0.3875	0.5347	0.5487	0.4590	0.2740
95	FFT	0.2795	0.4771	0.6222	0.8777	1.0303	0.9296	0.7336	0.4135
	FrFFT	0.2792	0.4767	0.6218	0.8773	1.0300	0.9295	0.7335	0.4134
100	FFT	0.9891	1.3637	1.5730	1.8373	1.8547	1.4963	1.1235	0.6030
	FrFFT	0.9851	1.3609	1.5707	1.8357	1.8538	1.4958	1.1232	0.6028
105	FFT	4.2747	3.8373	3.6972	3.5418	3.1290	2.2980	1.6556	0.8530
	FrFFT	4.2728	3.8325	3.6927	3.5395	3.1277	2.2974	1.6552	0.8528
110	FFT	9.1067	8.2753	7.5576	6.2445	4.9619	3.3818	2.3564	1.1742
	FrFFT	9.1066	8.2751	7.5573	6.2442	4.9617	3.3817	2.3563	1.1742
120	FFT	19.0051	18.0204	17.0505	14.3313	10.5028	6.5449	4.3611	2.0733
	FrFFT	19.0047	18.0200	17.0501	14.3305	10.5016	6.5441	4.3605	2.0731
150	FFT	48.7559	47.5214	46.2972	42.6852	35.7458	23.9279	15.9556	7.4236
	FrFFT	48.7552	47.5207	46.2965	42.6846	35.7451	23.9267	15.9544	7.4229

Table 1: European put option prices in VG model with FFT and FrFFT.

Exercise 4: The *Cox-Ingersoll-Ross (CIR) process* is defined as a solution of the stochastic differential equation

$$dX_t = \kappa(\theta - X_t)dt + \sigma\sqrt{X_t}dW_t. \tag{2}$$

Let $\mathrm{Pol}_n(\mathbb{R})$ be the space of univariate polynomials on $\mathbb{R}$ of maximal degree $n \in \mathbb{N}$, and let $\mathcal{H}_n = \{h_0, \dots, h_n\}$ be a basis for $\mathrm{Pol}_n(\mathbb{R})$. Write

$$H_n(x) = (h_0(x), \dots, h_n(x))$$

and denote by G_n the matrix representation with respect to $\mathcal{H}$ of the generator $\mathcal{G}$ associated to X_t and restricted to $\mathrm{Pol}_n(\mathbb{R})$, i.e.

$$\mathcal{G}p(x) = H_n(x)G_n\vec{p}, \tag{3}$$

where p is a polynomial in $\mathrm{Pol}_n(\mathbb{R})$ and has coordinate vector $\vec{p} \in \mathbb{R}^{n+1}$ with respect to $\mathcal{H}_n$. Then, the moments of X_T can be computed via the formula

$$\mathbb{E}[p(X_T)] = H_n(X_0)e^{G_n T}\vec{p}. \tag{4}$$

For this exercise we consider the monomial basis for $\mathrm{Pol}_n(\mathbb{R})$, i.e.

$$H_n(x) = (1, x, \cdots, x^n)^T.$$

a) Derive the generator $\mathcal{G}$ associated to the CIR process X_t, solution of (2). Write a Matlab code `GenCir.m` with input parameters $\kappa, \theta, \sigma, n$, that constructs the matrix G_n defined as in (3).

Solution: The generator of X is defined as

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

so that the action of $\mathcal{G}$ on a basis polynomial of the form x^p gives

$$\mathcal{G}x^p = \kappa(\theta - x)px^{p-1} + \frac{\sigma^2}{2} xp(p-1)x^{p-2} = -\kappa p x^p + (\kappa\theta p + p(p-1)\frac{\sigma^2}{2})x^{p-1}.$$

Consequently, the matrix G_n is of the form

$$G_n = \begin{pmatrix} 0 & \kappa\theta & 0 & 0 & \dots & 0 \\ 0 & -\kappa & 2\kappa\theta + 2\frac{\sigma^2}{2} & 0 & \dots & 0 \\ 0 & 0 & -2\kappa & 3\kappa\theta + 6\frac{\sigma^2}{2} & \ddots & 0 \\ 0 & 0 & 0 & -3\kappa & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & n\kappa\theta + n(n-1)\frac{\sigma^2}{2} \\ 0 & 0 & 0 & 0 & \dots & -n\kappa \end{pmatrix} \in \mathbb{R}^{n+1 \times n+1}.$$

Following Matlab codes provides a possible implementation for G_n.

```
function [ G_n ] = GenCir(kappa, theta, sigma, n)
% The function constructs the matrix G_n as in Exercise 4a, HW6.
% Input:  model parameters kappa, theta, sigma
%         maximal polynomial degree n
% Output: Matrix G_n
%
% Specify the two non-zero diagonals
d1 = (0:n)* (-kappa);
d2 = (1:n)* kappa * theta + 0.5 * sigma^2 * (1:n).* (0:n-1);
% Construct the matrix G_n
G_n = diag(d1, 0) + diag(d2, 1);
end
```

b) Write a Matlab code `MomCIR.m` which computes the moments of X_T as in formula (4). Input parameters: $G_n, X_0, T, \vec{p}$.

Solution: A possible implementation is shown hereunder.

```
function [ Mom ] = MomCIR( G_n, X_0, T, p )
% The function applies the moment formula (4) in Exercise 4, HW 6
% Input:  Matrix representation G_n of the generator G
%         Initial asset price X_0
%         Maturity time T
%         Coordinate vector p
% Output: Corresponding moment
```

```
% Evaluate basis vector in X_0
n = size(G_n, 1);
H_n = X_0 .^ (0:n-1);
Mom = H_n * expm(G_n * T) * p;
end
```

c) Compute the first three moments of X_T (using the previously implemented functions) for the following choice of model parameters:

$$\kappa = 2, \quad \theta = 0.4, \quad \sigma = 0.5, \quad X_0 = 0.4, \quad T = 1.$$

Check that your moments have been correctly computed, knowing that (see slides 9-10 of Lecture 5)

$$\frac{X_t}{\frac{\sigma^2}{4\kappa}(1 - e^{-\kappa t})}$$

has a non-central χ^2 distribution with parameters

$$k = \frac{4\kappa\theta}{\sigma^2}, \quad \alpha = \frac{e^{-\kappa t} X_0}{\frac{\sigma^2}{4\kappa}(1 - e^{-\kappa t})}.$$

Solution: We just insert the given parameters and we get

$$\mathbb{E}[X_T] = 0.4, \quad \mathbb{E}[X_T^2] = 0.1845, \quad \mathbb{E}[X_T^3] = 0.0964.$$

The first three moments of a non-central χ^2 distributed random variable Y with parameters k and α are given by

$$\begin{aligned}
\mathbb{E}[Y] &= k + \alpha \\
\mathbb{E}[Y^2] &= (k + \alpha)^2 + 2(k + 2\alpha) \\
\mathbb{E}[Y^3] &= (k + \alpha)^3 + 6(k + \alpha)(k + 2\alpha) + 8(k + 3\alpha).
\end{aligned}$$

Following codes shows that the moments of X_T are consistent with the moments of Y (after correct scaling).

```
% Exercise 4c, HW 6
kappa = 2;
theta = 0.4;
sigma = 0.5;
n = 3;
X_0 = 0.4;
T = 1;
% Construct matrix G_n
G_n = GenCir(kappa, theta, sigma, n);
% Compute moments using moment formula
M1 = MomCIR( G_n, X_0, T, [0 ; 1; 0; 0] );
M2 = MomCIR( G_n, X_0, T, [0 ; 0; 1; 0] );
```

```
M3 = MomCIR( G_n, X_0, T, [0 ; 0; 0; 1] );
%
%
% Compute moments of non-central chi-squared distribution
k = 4 * kappa * theta / sigma^2;
alpha = exp(-kappa * T) * X_0 / ((sigma^2 / (4*kappa)) * (1 - exp( ...
    -kappa * T)));

M1chi = k + alpha;
M2chi = (k + alpha)^2 + 2 * (k + 2*alpha);
M3chi = (k + alpha)^3 + 6 * (k + alpha) * (k + 2*alpha) + 8 * (k + ...
    3*alpha);

% We compare the moments after scaling M1, M2, M3
scaling = (sigma^2 / (4*kappa)) * (1 - exp( -kappa * T));
abs(M1 - M1chi * scaling)
abs(M2 - M2chi * scaling^2)
abs(M3 - M3chi * scaling^3)
% We observe that they coincide (up to numerical precision)
```

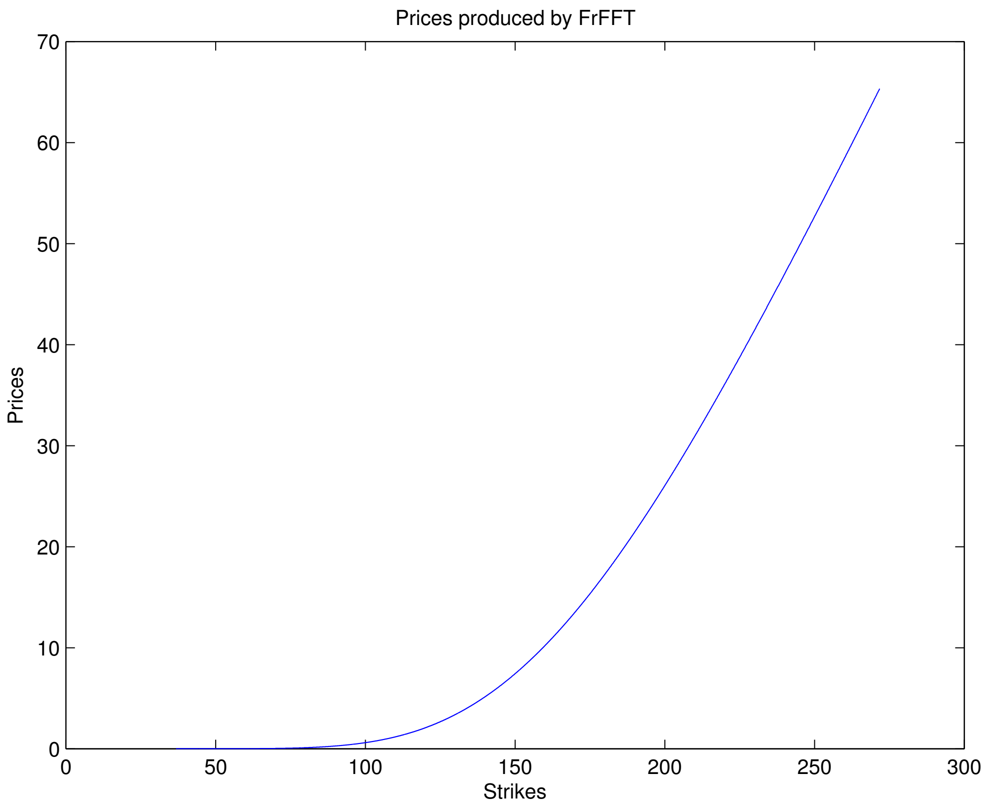

Figure 1: FrFFt prices

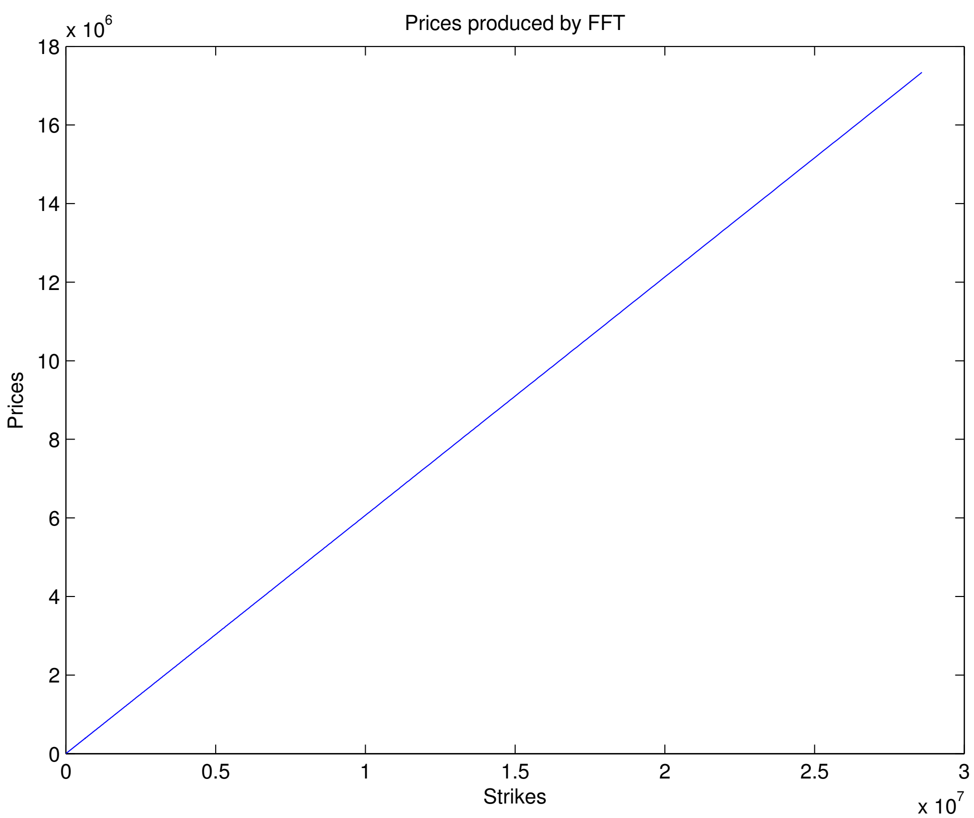

Figure 2: FFt prices