$S_t = S_0 \exp(rt + X_t)$

by assumption.

In order to compute the price of a call option

$$C(k) = e^{-rT} E[(e^{rT+X_T} - e^k)^+]$$

IDEA:

we would like to express its Fourier transform in strike in terms of the characteristic function $\Phi_T(v)$ of X_T and then find the prices for a range of strikes by Fourier inversion. However we cannot do this directly because $C(k)$ is not integrable (it tends to a positive constant as $k \to -\infty$). The key idea of the method is to instead compute the Fourier transform of the (modified) time value of the option, that is, the function

$$z_T(k) = e^{-rT} \underbrace{E[(e^{rT+X_T} - e^k)^+]}_{C(k)} - (1 - e^{k-rT})^+. \tag{11.17}$$

$$z_T(k) = C(k) - (1 - e^{k-rT})^+.$$

Recall : $z_T(k) = C(k) - (1-e^{k-rT})^+$.

Let $\zeta_T(v)$ denote the Fourier transform of the time value:

Probabilists' F.T

$$\zeta_T(v) = \mathbf{F} z_T(v) = \int_{-\infty}^{+\infty} e^{ivk} z_T(k) dk. \tag{11.18}$$

It can be expressed in terms of characteristic function of X_T in the following way. First, we note that since the discounted price process is a martingale, we can write

Density of X_T

$$z_T(k) = e^{-rT} \int_{-\infty}^{\infty} \rho_T(x) dx (e^{rT+x} - e^k)(1_{k \le x+rT} - 1_{k \le rT}).$$

See in $(e^{rT+X_T} - e^k)^+$:

$e^{rT+X_T} - e^k \geqslant 0 \iff rT + X_T \geqslant k$.

See in $(1-e^{k-rT})^+$:

$1 - e^{k-rT} \geqslant 0 \iff$

$k - rT \leqslant 0 \iff k \leqslant rT$.

$\longrightarrow$ Now we can compute $\zeta_T(v) = (\mathcal{F} z_T)(v)$:

$$\zeta_T(v) = e^{-rT}\int_{-\infty}^{\infty} dk \int_{-\infty}^{\infty} dx e^{ivk}\rho_T(x)(e^{rT+x} - e^k)(1_{k\le x+rT} - 1_{k\le rT})$$

$$= e^{-rT}\int_{-\infty}^{\infty}\rho_T(x)dx\int_{x+rT}^{rT} e^{ivk}(e^k - e^{rT+x})dk$$

update the bounds using the $\mathbb{1}$ fcts.

$$= \int_{-\infty}^{\infty}\rho_T(x)dx\left\{\frac{e^{ivrT}(1-e^x)}{iv+1} - \frac{e^{x+ivrT}}{iv(iv+1)} + \frac{e^{(iv+1)x+ivrT}}{iv(iv+1)}\right\}$$

The first term in braces disappears due to martingale condition and, after computing the other two, we conclude that (cf my notes above for details)

probabilists' F.T

$$(\mathcal{F} z_T)(v) = \boxed{\zeta_T(v) = e^{ivrT}\frac{\Phi_T(v-i)-1}{iv(1+iv)}} \qquad (11.19)$$

$z_T : k \mapsto C(k) - (1-e^{k-rT})^+$.

The martingale condition guarantees that the numerator is equal to zero for $v = 0$ and the fraction has a finite limit for $v \to 0$.

So the above formula is well defined, even for $v = 0$.

the fact that $\Phi_T(-i) = 1$ (proof @ page 9).

Option prices can now be found by inverting the Fourier transform:

probabilists' I.F.T

$$z_T(k) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-ivk}\zeta_T(v)dv \tag{11.20}$$

$$\mathcal{F}^{-1}(\underbrace{\mathcal{F} z_T}_{\zeta_T}) = z_T$$

probabilists' F.T

once $z_T(k)$ is obtained, one can get $C(k)$, call price.

$$C(k) = z_T(k) + (1 - e^{k-rT})^+.$$

Now the missing point is how to compute 11.20 in an accurate & fast way: FFT.

Using the FFT

fft Discrete Fourier transform. The MATLAB's F.T ($\neq$ from probabilists' F.T).

fft(X) is the discrete Fourier transform (DFT) of vector X.

For length N input vector x, the DFT is a length N vector X, with elements

$$X(n) = \sum_{k=0}^{N-1} x(k)\, e^{-2\pi i \frac{kn}{N}} \quad \text{with } 0 \leq n \leq N-1.$$

} a "-" in the argument: $\neq$ def vs Rama C.

↳ indeed, MATLAB's F.T (with a "−" in the exp).

The inverse DFT (computed by IFFT) is given by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{2i\pi \frac{kn}{N}}, \quad 0 \leq n \leq N-1.$$

MATLAB's I.F.T (with a "+" in the exp).

Using FFT, let's try to compute 11.20 :

We have $z_T(k) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-ivk} g_T(v)\,dv$. And we want to compute:

probabilists' IFT (but for MATLAB: fft).

DFT : $$F_n = \sum_{k=0}^{N-1} f_k\, e^{-2\pi i n \frac{k}{N}}$$ (cf formula @ previous page)

This is MATLAB's F.T (Analog to the proba's IFT above)

1st step : to truncate the integral:

$$z_T(k) \simeq \frac{1}{2\pi}\int_{-A/2}^{A/2} e^{-ivk} g_T(v)\,dv$$ } between $-\frac{A}{2}$ & $\frac{A}{2}$.

2nd step : quadrature :

it's a formula that we admit. { TRAPEZOÏDAL QUADRATURE

$$z_T(k) \simeq \frac{1}{2\pi}\frac{A}{N-1}\sum_{j=0}^{N-1} w_j\, e^{-iv_j k} g_T(v_j)$$

$\boxed{\begin{array}{l} w_0 = w_{N-1} = 0.5 \\ \forall j \notin \{0, N-1\}, w_j = 1 \end{array}}$

where : $v_j = -\frac{A}{2} + j \times \underbrace{\frac{A}{N-1}}_{=\Delta} = -\frac{A}{2} + j\Delta$, w_j : weights.

$$z_T(k) \simeq \frac{1}{2\pi}\frac{A}{N-1}\sum_{j=0}^{N-1} w_j\, g_j\, e^{-ik\left(-\frac{A}{2} + j\Delta\right)}$$

$(g_j = g_T(v_j))$

$$Z_T(k) \approx \frac{1}{2\pi} \Delta e^{-ik A/2} \sum_{j=0}^{N-1} w_j g_j e^{-ik_j \Delta}.$$

Let's define: $k = \frac{2\pi n}{N \frac{A}{N-1}}$ so that:

$$F_n = \frac{1}{2\pi} \Delta e^{i \frac{\pi n (N-1)}{N}} \sum_{j=0}^{N-1} w_j g_j e^{-i \frac{2\pi n}{N} j}.$$

So we find the same expression as DFT:

MATLAB's FT but probabilists' IFT.

$$F_n = \underbrace{\frac{1}{2\pi} \Delta e^{i \frac{\pi n (N-1)}{N}}}_{\text{Known quantity}} \cdot \mathrm{DFT}(w_j g_j)$$

That's why in the MATLAB code we use fft @ that step.

Probabilists' I.F.T of ζ_T.

We have to compute

$$z_T(k) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-ivk}\zeta_T(v)dv.$$

Since $z_T(k)$ must be real, then it holds

Trick of Functional Analysis.

$$z_T(k) = \frac{1}{\pi}\int_{0}^{+\infty} e^{-ivk}\zeta_T(v)dv$$

To compute this integral, we can use a quadrature formula, i.e.,

$$\begin{aligned} z_T(k) &\approx \frac{1}{\pi}\int_{0}^{A(N-1)/N} e^{-ivk}\zeta_T(v)dv \\ &\approx \frac{1}{\pi}\sum_{j=0}^{N-1} w_j \eta e^{-i\eta jk}\zeta_T(\eta j). \end{aligned}$$

If we set $w_0 = w_{N-1} = 0.5$, 1 otherwise, we are using a trapezoidal formula with nodes $j\eta$, with $\eta = A/N$.

If we now consider the following grid for the log-strike $k_l = -\lambda N/2 + \lambda l$, with $\lambda = 2\pi/(N\eta)$ and $l = 0, ..., N-1$, we obtain

$$
\begin{aligned}
z_T(k_l) &\approx \frac{1}{\pi}\sum_{j=0}^{N-1} w_j \eta e^{-i\eta j(-\lambda N/2+\lambda l)} \zeta_T(\eta j) \\
&= \frac{1}{\pi}\sum_{j=0}^{N-1} w_j \eta e^{i\eta j\lambda N/2} e^{-i\eta j\lambda l} \zeta_T(\eta j) \\
&= \frac{1}{\pi}\sum_{j=0}^{N-1} w_j \eta e^{ij\pi} e^{-ijl2\pi/N} \zeta_T(\eta j) \\
&= \frac{1}{\pi} FFT(\{w_j \eta e^{ij\pi} \zeta_T(\eta j)\}_{j=0}^{N-1})
\end{aligned}
$$

MATLAB's FFT
but probabilists'
IFT

Not exactly the same formula, but in the end it is the same result.

$$\overbrace{\zeta_T(v)}^{\text{article}} = \overbrace{g_T(v)}^{\text{notes}}$$

Matlab code

```
function fii = trasf_fourier_BS (r,sig,T,v)
fii = (exp(1i*r*v*T)).*…
    ((characteristic_func_BS(sig,T,v-1i)…
      -1)./(1i*v.*(1+1i*v))   );
```

charac. fct

$\phi_T(y)$

```
function f = characteristic_func_BS(sig,T,y)
f = exp(1i* (-sig^(2)/2*T).*y …
     -(T*sig.^(2)*y.^(2))/2) ;
```

```matlab
function [P_i] = FFT_BS(K_i)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% European Call - BS model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [P_i] = FFT_BS(K_i) - could be a vector
% INPUT: K_i = strikes - could be a vector
% OUTPUT: P_i = prices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%--- model parameters
S = 100; % spot price
T = 1; % maturity
r = 0.0367; % risk-free interest rate
sig = 0.17801; % volatility
Npow = 15;
N = 2^(Npow); % grid point
A = 600; % upper bound
eta = A/N;
lambda = 2*pi/(N*eta);
k = -lambda * N/2  + lambda *(0:N-1); % log-strike grid
K = S * exp(k); % strike
v = eta*(0:N-1);
v(1)=1e-22; %correction term: could not be equal to zero
            %   (otherwise NaN)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PRICING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
% Fourier transform of z_k
tr_fou = trasf_fourier_BS(r,sig,T,v);
% Trapezoidal rule
w = [0.5  ones(1,N-2)  0.5];
h = exp(1i*(0:N-1)*pi).*tr_fou.*w*eta;
P = S * real( fft(h)/pi + max(1-exp(k-r*T),0)); % prices
time=toc
```

} Here you just have to change the model: Kou, B&S...

⚠ MATLAB's FFT is implement with a "−" in the exp whereas probabilists use the IFT with "−".

```
% delete too small and too big strikes
index=find( (K>0.1*S & K<3*S) );
K=K(index); P=P(index);
```