

## Pricing using the probability density

AVt, Dec 2004

```
> restart: with(avtbslib): Digits:=14: #interface(prompt=" " ):
```

### Pricing with the pdf

Pricing of a call may done through the payoff  $\max(0, X - K)$  and the pdf at expiry by

```
> call:= exp(-r*t) * int( max( xi-K,0)*StockPricePdf(xi),xi=0..infinity);
#convert(call,piecewise,K) assuming (0<K), (0<t), (0<r);
```

$$\text{call} := e^{(-r t)} \int_0^{\infty} \max(0, \xi - K) \text{StockPricePdf}(\xi) d\xi$$

or likewise as  $e^{(-r t)} \int_K^{\infty} \text{StockPricePdf}(\xi) (\xi - K) d\xi$ .

Here `StockPricePdf` is a function of the stock itself and not of the returns.

One can rediscover the pdf by differentiating the call two times by its strike up to discounting (which is called the formula of Breeden-Litzenberger):

```
> 'diff(call,K$2)': '%' = simplify(%) assuming(0<=K);
```

$$\frac{\partial^2}{\partial K^2} \left( e^{(-r t)} \int_0^{\infty} \max(0, \xi - K) \text{StockPricePdf}(\xi) d\xi \right) = e^{(-r t)} \text{StockPricePdf}(K)$$

In the Black-Scholes setting this reads as

```
> 'diff(exp(r*t)*BSCall(S,K,t,r,v),K$2)'='diffN(dTwo(S,K,t,r,v))/(K*v*sqrt(t));
is(simplify(expand(%))) assuming 0<K ;
```

$$\frac{\partial^2}{\partial K^2} (e^{(r t)} \text{BSCall}(S, K, t, r, v)) = \frac{\text{diffN}(dTwo(S, K, t, r, v))}{K v \sqrt{t}}$$

true

so the (undiscounted) pdf has the explicit formula

```
> diffN(dTwo(S,K,T,r,v))/K/v/sqrt(t): p:= unapply( %, S,K,T,r,v):
'p(S,K,t,r,v)': '%'=%;
```

$$p(S, K, t, r, v) = \frac{1}{2} \frac{e^{\left( -1/2 \left( \frac{\ln\left(\frac{S}{K}\right) + r t}{v \sqrt{t}} - \frac{v \sqrt{t}}{2} \right)^2 \right)}}{\sqrt{\pi} K v \sqrt{t}}$$

Try some test data (normally avoid the special case spot = strike for testing) and compare the integral with the usual BS formula:

```
> tstData:= [S=100, K=90, t=1.0, r=0.05, v=0.25];
tstData1:= [S=100, K=60, t=5/365.25, r=0.05, v=0.35]; # 5 days
tstData2:= [S=100, K=140, t=0.25, r=0.05, v=0.25]; # 3 month
tstData3:= [S=100, K=90, t=5.0, r=0.05, v=0.20]; # 5 years

tstData := [S = 100, K = 90, t = 1.0, r = 0.05, v = 0.25]
tstData1 := [S = 100, K = 60, t = 0.013689253935661, r = 0.05, v = 0.35]
```

```

tstData2 := [S = 100, K = 140, t = 0.25, r = 0.05, v = 0.25]
tstData3 := [S = 100, K = 90, t = 5.0, r = 0.05, v = 0.20]
> 'exp(-r*t)*Int(max(S1-K,0) * p(S,S1,t,r,v),S1=0..infinity)':
'% ' = evalf(eval(%,tstData));
`;
'BSCall(S,K,t,r,v)': '% ' = evalf(eval(%,tstData));

```

$$e^{(-rt)} \int_0^{\infty} \max(S1 - K, 0) p(S, S1, t, r, v) dS1 = 18.140762950606$$

$$\text{BSCall}(S, K, t, r, v) = 18.140762950606$$

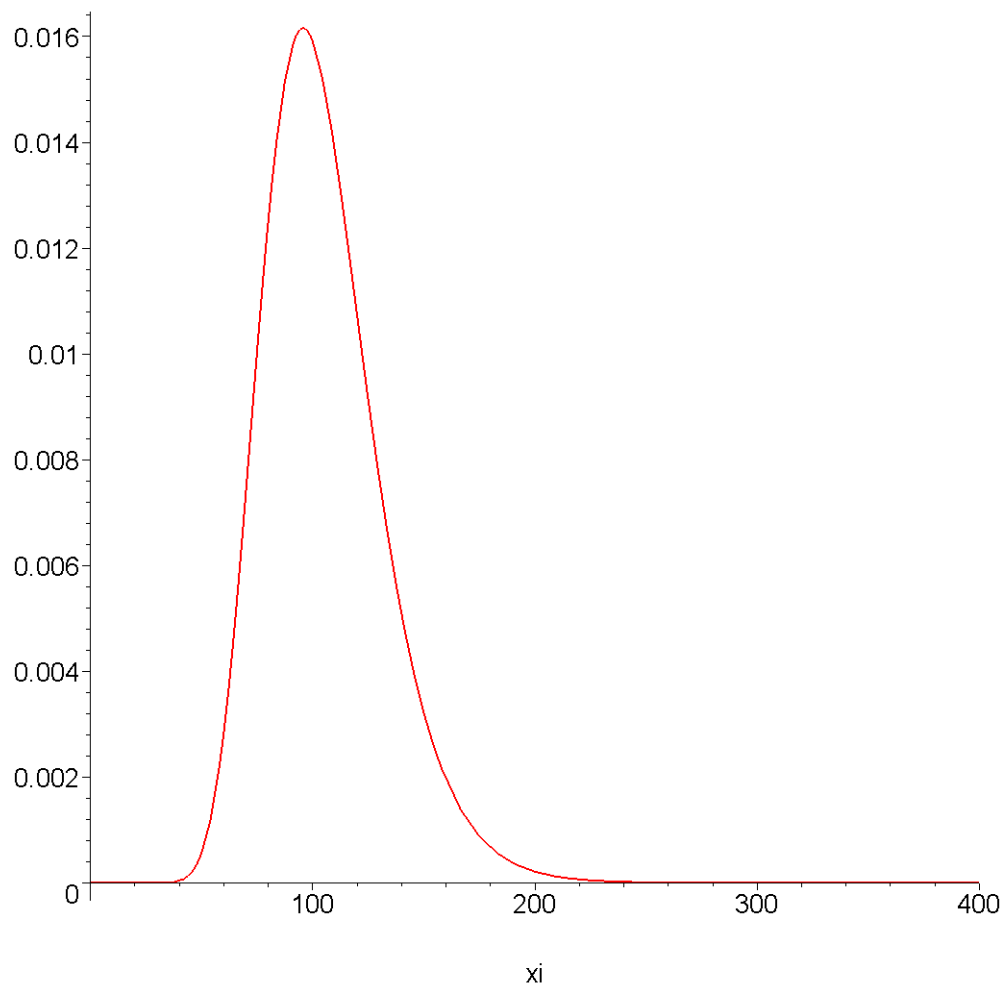
Additionally let us plot the (lognormal) pdf (and note which parameter becomes a variable):

```

> StockPrice_pdf:= eval(p(S,xi,t,r,v),tstData);
plot(%,xi=0..400, thickness=2);

```

$$\text{StockPrice\_pdf} := \frac{2.000000000000000 e^{\left( -1/2 \left( 4.000000000000000 \ln\left( \frac{100}{\xi} \right) + 0.075000000000000 \right)^2 \right)} }{\sqrt{\pi} \xi}$$



and note which error is made if integration is cut off too early (restricting to a range between 0.1 and 400) ...

```

> 'exp(-r*t)*Int(max(S1-K,0) * p(S,S1,t,r,v),S1=0..0.1) +
exp(-r*t)*Int(max(S1-K,0) * p(S,S1,t,r,v),S1=400..infinity)':
'% ' = evalf(eval(%,tstData));

```

$$e^{(-r t)} \int_0^{0.1} \max(S1 - K, 0) p(S, S1, t, r, v) dS1 + e^{(-r t)} \int_{400}^{\infty} \max(S1 - K, 0) p(S, S1, t, r, v) dS1 = 0.70131248476699 \cdot 10^{-5}$$

variant: log moneyness  $\mu = \ln\left(\frac{\text{strike}}{\text{forward}}\right)$

This means to change the variables as  $\text{strike} = e^{\mu} \text{fwd}$  with  $\text{fwd} = e^{(r t)} \text{spot}$  :

```
> P:='P':
'exp(-r*t)*Int((S1-K) * P(S,S1,t,r,v),S1=K..infinity)';
subs(S=exp(-r*t)*fwd,%);
changevar(S1= exp(mu)*fwd,%,mu);
```

$$e^{(-r t)} \int_K^{\infty} (S1 - K) P(S, S1, t, r, v) dS1$$

$$e^{(-r t)} \int_K^{\infty} (S1 - K) P(e^{(-r t)} \text{fwd}, S1, t, r, v) dS1$$

$$e^{(-r t)} \int_{\ln\left(\frac{K}{\text{fwd}}\right)}^{\infty} (e^{\mu} \text{fwd} - K) P(e^{(-r t)} \text{fwd}, e^{\mu} \text{fwd}, t, r, v) e^{\mu} \text{fwd} d\mu$$

for the BS density p that is

```
> 'exp(-r*t)*Int((S1-K) * p(S,S1,t,r,v),S1=K..infinity)';
subs(S=exp(-r*t)*fwd,%):
changevar(S1= exp(mu)*fwd,%,mu):
CancelInverses(%);
```

$$e^{(-r t)} \int_K^{\infty} (S1 - K) p(S, S1, t, r, v) dS1$$

$$e^{(-r t)} \int_{\ln\left(\frac{K}{\text{fwd}}\right)}^{\infty} \frac{1}{2} \frac{(-e^{\mu} \text{fwd} + K) e^{\left(-\frac{(2\mu + v^2 t)^2}{8 v^2 t}\right)}}{\sqrt{\pi} v \sqrt{t}} \sqrt{2} d\mu$$

and after inserting known data again one actually gets the same price:

```
> 'exp(-r*t)*Int((S1-K) * p(S,S1,t,r,v),S1=K..infinity)';
subs(S=exp(-r*t)*fwd,%):
changevar(S1= exp(mu)*fwd,%,mu):
CancelInverses(%);

subs(fwd=exp(r*t)*S,%);
eval(%,tstData):
'price'=evalf(%);
'BSCall(S,K,t,r,v)': '%'= evalf(eval(%,tstData));
```

$$e^{(-r t)} \int_K^\infty (S1 - K) p(S, S1, t, r, v) dS1$$

$$e^{(-r t)} \int_{\ln\left(\frac{K}{fwd}\right)}^\infty \frac{1}{2} \frac{(-e^\mu fwd + K) e^{\left(-\frac{(2\mu + v^2 t)^2}{8 v^2 t}\right)}}{\sqrt{\pi} v \sqrt{t}} \sqrt{2} d\mu$$

$$e^{(-r t)} \int_{\ln\left(\frac{K}{e^{(r t)} S}\right)}^\infty \frac{1}{2} \frac{(-e^\mu e^{(r t)} S + K) e^{\left(-\frac{(2\mu + v^2 t)^2}{8 v^2 t}\right)}}{\sqrt{\pi} v \sqrt{t}} \sqrt{2} d\mu$$

price = 18.140762950605

BSCall(S, K, t, r, v) = 18.140762950606

Within that the pdf (over the spot) transforms as

```
> p(S, K, t, r, v) ;
subs(S=exp(-r*t)*fwd,%):
subs(K= exp(mu)*fwd,%): # <-- change of variables
CancelInverses(%);
```

$$\frac{1}{2} \frac{e^{\left(-1/2 \left(\frac{\ln\left(\frac{S}{K}\right) + r t}{v \sqrt{t}} - \frac{v \sqrt{t}}{2}\right)^2\right)}}{\sqrt{\pi} K v \sqrt{t}} \sqrt{2}$$

$$\frac{1}{2} \frac{e^{\left(-\frac{\left(\frac{\mu}{v \sqrt{t}} - \frac{v \sqrt{t}}{2}\right)^2}{2}\right)}}{\sqrt{\pi} e^\mu fwd v \sqrt{t}} \sqrt{2}$$

which is the same "exp" as above since

```
> 1/2*exp(-1/8*(2*mu+v^2*t)^2/v^2/t)*2^(1/2)/Pi^(1/2)/exp(mu)/fwd/v/t^(1/2) =
1/2*exp(-1/2*(-mu/v/t^(1/2)-1/2*v*t^(1/2))^2)*2^(1/2)/Pi^(1/2)/exp(mu)/fwd/v/t^(1/2);
simplify(% ,symbolic): is(%);
```

$$\frac{1}{2} \frac{e^{\left(-\frac{(2\mu + v^2 t)^2}{8 v^2 t}\right)}}{\sqrt{\pi} e^\mu fwd v \sqrt{t}} \sqrt{2} = \frac{1}{2} \frac{e^{\left(-\frac{\left(\frac{\mu}{v \sqrt{t}} - \frac{v \sqrt{t}}{2}\right)^2}{2}\right)}}{\sqrt{\pi} e^\mu fwd v \sqrt{t}} \sqrt{2}$$

```

true
> q := unapply(
  1/2*exp(-1/2*(-mu/V/T^(1/2)-1/2*V*T^(1/2))^2)*2^(1/2)/Pi^(1/2)/exp(mu)/fwd/V/T^(1/2),
  fwd, mu, T, r, V);

```

$$q := (fwd, \mu, T, r, V) \rightarrow \frac{1}{2} \frac{e^{\left(-1/2 \left(-\frac{\mu}{V\sqrt{T}} - 1/2 V\sqrt{T}\right)^2\right)}}{\sqrt{\pi} e^{\mu} fwd V \sqrt{T}} \sqrt{2}$$

If we name the last one by q the pricing formula reads as

```

> 'exp(-r*t)*int((exp(mu)-exp(mu_K))*exp(mu)*fwd^2*q(fwd, mu, t, r, v),
mu=mu_K..infinity)';
subs(mu_K = ln(K/fwd),%):
subs(fwd=exp(r*t)*S,%):
eval(%,tstData): 'price'=evalf(%);
'BSCall(S,K,t,r,v)': '%'= evalf(eval(%,tstData));

```

$$e^{(-r t)} \int_{\mu_K}^{\infty} (e^{\mu} - e^{\mu_K}) e^{\mu} fwd^2 q(fwd, \mu, t, r, v) d\mu$$

price = 18.140762950605

BSCall(S, K, t, r, v) = 18.140762950606

where  $\mu_K = \ln\left(\frac{K}{fwd}\right)$  (Maple does not like indexed variables that much ...).

## Simplified variant

```

> Q:= (fwd, mu, T, r, V) ->
fwd*exp(-1/2*(mu/V/T^(1/2)+1/2*V*T^(1/2))^2)/V/T^(1/2)/sqrt(2*Pi);

```

$$Q := (fwd, \mu, T, r, V) \rightarrow \frac{fwd e^{\left(-1/2 \left(\frac{\mu}{V\sqrt{T}} + 1/2 V\sqrt{T}\right)^2\right)}}{V \sqrt{T} \sqrt{2 \pi}}$$

Defining Q the above pricing formula now reads as

```

> 'exp(-r*t)*Int((exp(mu)-exp(mu_K))*Q(fwd, mu, t, r, v),mu=mu_K..infinity)';

```

$$e^{(-r t)} \int_{\mu_K}^{\infty} (e^{\mu} - e^{\mu_K}) Q(fwd, \mu, t, r, v) d\mu$$

and indeed gives the correct value:

```

> 'exp(-r*t)*Int((exp(mu)-exp(mu_K))*Q(fwd, mu, t, r, v),mu=mu_K..infinity)';
subs(mu_K = ln(K/fwd),%):
subs(fwd=exp(r*t)*S,%):
eval(%,tstData): evalf(%);
'BSCall(S,K,t,r,v)': '%'= evalf(eval(%,tstData));

```

$$e^{(-r t)} \int_{\mu_K}^{\infty} (e^{\mu} - e^{\mu_K}) Q(fwd, \mu, t, r, v) d\mu$$

18.140762950605

BSCall(S, K, t, r, v) = 18.140762950606

Since Q does not depend on r and the fwd may be seen as a 'constant' for integration a somewhat nicer form is:

```

> q:= (mu, T, V) -> exp(-1/2*(mu/V/T^(1/2)+1/2*V*T^(1/2))^2)/V/T^(1/2)/sqrt(2*Pi);

```

$$q := (\mu, T, V) \rightarrow \frac{e^{\left(-1/2 \left(\frac{\mu}{V\sqrt{T}} + 1/2 V\sqrt{T}\right)^2\right)}}{V\sqrt{T}\sqrt{2\pi}}$$

so pricing reads as  $e^{(-rt)} \text{fwd} \int_{\mu_K}^{\infty} \left(e^{\mu} - e^{\mu_K}\right) q(\mu, t, v) d\mu$  which is check against the test data:

```
> 'exp(-r*t)*fwd*Int((exp(mu)-exp(mu[K]))*q(mu, t, v),mu=mu[K]..infinity)';
subs(mu[K] = ln(K/fwd),%):
subs(fwd=exp(r*t)*S,%):
eval(%,tstData): 'price'=evalf(%);
'BSCall(S,K,t,r,v)': '%'= evalf(eval(%,tstData));
```

$$e^{(-rt)} \text{fwd} \int_{\mu_K}^{\infty} \left(e^{\mu} - e^{\mu_K}\right) q(\mu, t, v) d\mu$$

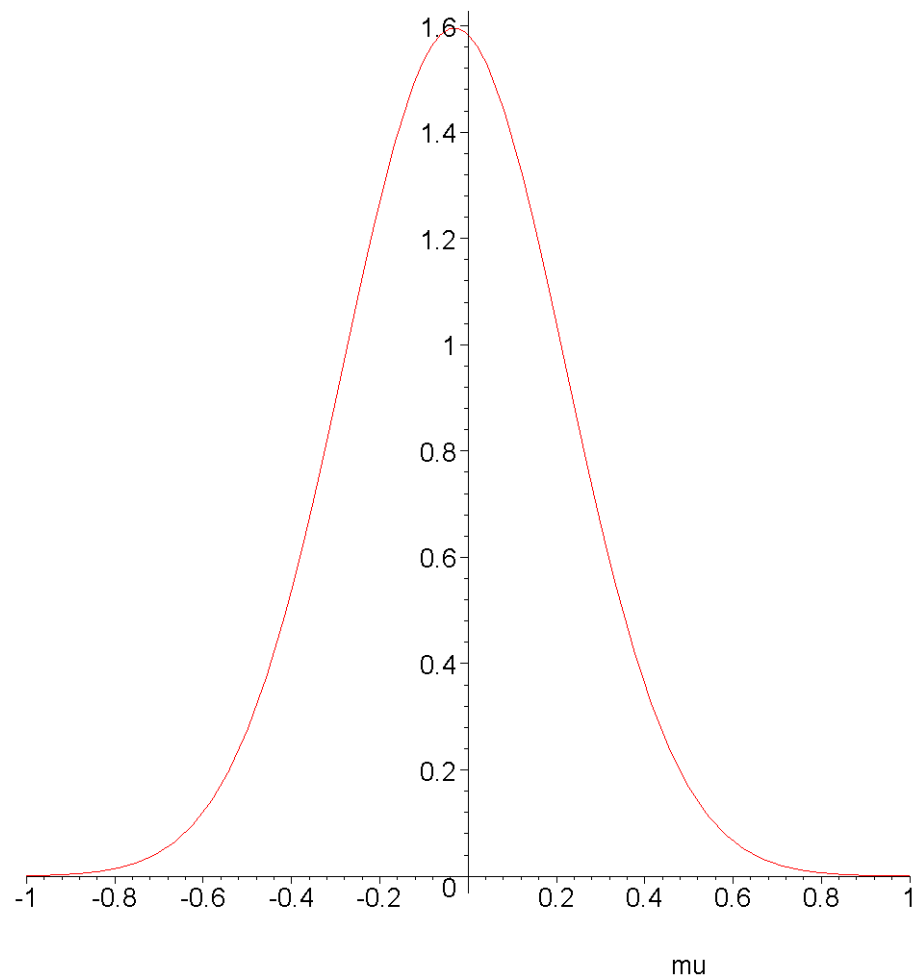
price = 18.140762950603

BSCall(S, K, t, r, v) = 18.140762950606

The pdf (over log moneyness) now looks like that:

```
> 'q(mu,t,v)': '%'= eval(%,tstData); rhs(%):
plot(%,mu=-1..1);
```

$$q(\mu, t, v) = \frac{2.00000000000000 e^{\left(-\frac{(4.00000000000000 \mu + 0.12500000000000)^2}{2}\right)}}{\sqrt{\pi}} \sqrt{2}$$



```
> #'q(mu,t,0.25)'; plots[animate]( plot, [%mu=-0.5..0.5], t=2..0.1 );
```

### Compact version

Using  $\sigma = v\sqrt{t}$ ,  $\mu_K = \ln\left(\frac{K}{\text{fwd}}\right)$  and the following version Q of the pdf

```
> Q:= (mu, sigma) -> exp(-1/2*(mu/sigma+sigma/2)^2)/sigma/sqrt(2*Pi);
```

$$Q := (\mu, \sigma) \rightarrow \frac{e^{\left(-1/2\left(\frac{\mu}{\sigma} + 1/2\sigma\right)^2\right)}}{\sigma\sqrt{2\pi}}$$

the pricing simply reads as  $\text{call} = e^{(-rt)} \text{fwd} \int_{\mu_K}^{\infty} \left(e^{\mu} - e^{\mu_K}\right) Q(\mu, \sigma) d\mu = S \int_{\ln\left(\frac{K}{\text{fwd}}\right)}^{\infty} \left(e^{\mu} - \frac{K}{\text{fwd}}\right) Q(\mu, \sigma) d\mu :$

```
> 'exp(-r*t)*fwd*Int((exp(mu)-exp(mu[K]))*Q(mu, sigma),mu=mu[K]..infinity)';
subs(sigma = v*sqrt(t),%):
subs(mu[K] = ln(K/fwd),%):
subs(fwd=exp(r*t)*S,%):
eval(%,tstData):
`price`=evalf(%);
'BSCall(S,K,t,r,v)': '%'= evalf(eval(%,tstData));
```

$$e^{(-r t)} \text{fwd} \int_{\mu_K}^{\infty} \left( e^{\mu} - e^{\mu_K} \right) Q(\mu, \sigma) d\mu$$

price = 18.140762950603

BSCall(S, K, t, r, v) = 18.140762950606

which works for the other test data as well

```
> 'exp(-r*t)*fwd*Int((exp(mu)-exp(mu[K]))*Q(mu, sigma),mu=mu[K]..infinity)';
``;
for tmpData in [tstData1,tstData2,tstData3] do
  'exp(-r*t)*fwd*Int((exp(mu)-exp(mu[K]))*Q(mu, sigma),mu=mu[K]..infinity)';
  subs(sigma = v*sqrt(t),%):
  subs(mu[K] = ln(K/fwd),%):
  subs(fwd=exp(r*t)*S,%):
  eval(%,tmpData):
  print('price'=evalf(%));
  'BSCall(S,K,t,r,v)':
  print('%'= evalf(eval(%,tmpData)));
  print(``);
end do:
tmpData:='tmpData':
```

$$e^{(-r t)} \text{fwd} \int_{\mu_K}^{\infty} \left( e^{\mu} - e^{\mu_K} \right) Q(\mu, \sigma) d\mu$$

price = 40.041053710335

BSCall(S, K, t, r, v) = 40.041053710338

price = 0.022050120704628

BSCall(S, K, t, r, v) = 0.022050120705

price = 34.385242031228

BSCall(S, K, t, r, v) = 34.385242031229

Just a remark: indeed Q is a density since

```
> 'int(Q(mu,sigma),mu=-infinity..infinity)': '%'= % assuming 0<sigma;
```

$$\int_{-\infty}^{\infty} Q(\mu, \sigma) d\mu = 1$$

## Cutting the upper integration bound

That compact version also admits to find out where to cut off the upper integration bound (to restrict to finite numerical integration).

For  $c = e^{\mu_K}$  we have

```
> 'int((exp(mu)-c)*Q(mu,sigma),mu = cutoff .. infinity)';
% assuming 0<sigma:
collect(%,c):
'limit(%,cutoff=infinity)': '%'= % assuming 0<sigma;
```



$$\int_{\text{cutoff}}^{\infty} (e^{\mu} - c) Q(\mu, \sigma) d\mu$$

$$\lim_{\text{cutoff} \rightarrow \infty} \left( \frac{1}{2} \operatorname{erf} \left( \frac{\sqrt{2} (2 \text{cutoff} + \sigma^2)}{4 \sigma} \right) - \frac{1}{2} \right) c - \frac{1}{2} \operatorname{erf} \left( \frac{\sqrt{2} (2 \text{cutoff} - \sigma^2)}{4 \sigma} \right) + \frac{1}{2} = 0$$

Now take  $\text{cutoff} = \frac{n \sigma^2}{2}$  and look at that as

```
> '2*int((exp(mu)-c)*Q(mu,sigma),mu = n*sigma^2/2 .. infinity)' < 2*epsilon;
% assuming 0<sigma:
collect(% ,c);
#'limit(% ,n=infinity)': '%' = % assuming 0<sigma;
```

$$2 \int_{\frac{n \sigma^2}{2}}^{\infty} (e^{\mu} - c) Q(\mu, \sigma) d\mu < 2 \epsilon$$

$$\left( \operatorname{erf} \left( \frac{1}{4} n \sigma \sqrt{2} + \frac{1}{4} \sigma \sqrt{2} \right) - 1 \right) c - \operatorname{erf} \left( \frac{1}{4} n \sigma \sqrt{2} - \frac{1}{4} \sigma \sqrt{2} \right) + 1 < 2 \epsilon$$

As  $c = \frac{K}{\text{fwd}}$  may range between 1/4 and 4 this is to find bounds for the arguments in the error functions to give values close to 1 and we can take the 'smaller' one.

With  $\Sigma = \frac{\sigma \sqrt{2}}{4}$  this means to find n such that  $|-1 + \operatorname{erf}((n-1) \Sigma)| < \epsilon$  and for  $\epsilon = 1\text{E-}16$  (covering a large c as well) this gives (with roundings):

```
> cutoff:='cutoff':
remDigits:=Digits:
Digits:=20:
'eval((n-1)*Sigma,Sigma= 1/4*sigma*2^(1/2))' = 'fsolve( (1-erf(x))=1e-16,x)';
lhs(%)=evalf(rhs(%));
solvefor[n](%): expand(%): evalf(%); #evalf[2](%);

cutoff=n*sigma^2/2;
'eval(% ,%)';
expand(%): evalf(%); evalf[3](%);
Digits:=remDigits:
```

$$(n-1) \Sigma \left| \Sigma = \frac{\sigma \sqrt{2}}{4} \right| = \text{fsolve}(1 - \operatorname{erf}(x) = 0.1 \cdot 10^{-15}, x)$$

$$\frac{(n-1) \sigma \sqrt{2}}{4} = 5.8723700904539631451$$

$$n = 1.00000000000000000000 + \frac{16.609570850388227244}{\sigma}$$

$$\text{cutoff} = \frac{n \sigma^2}{2}$$

$$\left( \text{cutoff} = \frac{n \sigma^2}{2} \right) \left| n = 1.00000000000000000000 + \frac{16.609570850388227244}{\sigma} \right|$$

$$\text{cutoff} = 0.50000000000000000000 \sigma^2 + 8.3047854251941136220 \sigma$$

$$\text{cutoff} = 0.500 \sigma^2 + 8.30 \sigma$$

so  $\text{cutoff} = \frac{\text{vola}^2 \text{time}}{2} + 8 \text{ vola} \sqrt{\text{time}}$  should give a good exactness for the 'compact version'.

For the test data this results in about 15 decimal points of exactness:

```
> cutoff := v^2*t/2+8*v*t^(1/2);
'cutoff': '%' = eval(%,tstData);
'Int((exp(mu)-exp(mu[K]))*Q(mu, sigma),mu=cutoff..infinity)';
subs(sigma = v*sqrt(t),%):
subs(mu[K] = ln(K/fwd),%):
subs(fwd=exp(r*t)*S,%):
eval(%,tstData):
`cutoff error`=evalf(%);
cutoff:='cutoff':
```

$$\text{cutoff} := \frac{v^2 t}{2} + 8 v \sqrt{t}$$

$$\text{cutoff} = 2.0312500000000$$

$$\int_{\text{cutoff}}^{\infty} \left( e^{\mu} - e^{\mu_K} \right) Q(\mu, \sigma) d\mu$$

$$\text{cutoff error} = 0.55429476536153 \cdot 10^{-15}$$

and for the other test data it is similar:

```
> cutoff := v^2/2*t+8*v*t^(1/2);
``;
for tmpData in [tstData1,tstData2,tstData3] do
  print('cutoff'= eval(cutoff,tmpData));
  'Int((exp(mu)-exp(mu[K]))*Q(mu, sigma),mu=cutoff..infinity)';
  subs(sigma = v*sqrt(t),%):
  subs(mu[K] = ln(K/fwd),%):
  subs(fwd=exp(r*t)*S,%):
  eval(%,tmpData):
  `cutoff error`=evalf(%);
  print(%);
  print('');
end do:
tmpData:='tmpData':
cutoff:='cutoff':
```

$$\text{cutoff} := \frac{v^2 t}{2} + 8 v \sqrt{t}$$

$$\text{cutoff} = 0.32844150533584$$

$$\text{cutoff error} = 0.35525665578161 \cdot 10^{-15}$$

$$\text{cutoff} = 1.0078125000000$$

$$\text{cutoff error} = 0.31283242604693 \cdot 10^{-15}$$

$$\text{cutoff} = 3.6777087639997$$

$$\text{cutoff error} = 0.61164050793550 \cdot 10^{-15}$$

A different way would be:

for large arguments numerical one would pass to asymptotics of erf or cdfNormal.

## Rediscovering the BS formula

Maple can solve the compact version by symbolic integration:

```
> 'S*int((exp(mu)-K/fwd)*Q(mu,sigma),mu = ln(K/fwd) .. infinity)';
% assuming 0<sigma;
```

$$S \int_{\ln\left(\frac{K}{\text{fwd}}\right)}^{\infty} \left(e^{\mu} - \frac{K}{\text{fwd}}\right) Q(\mu, \sigma) d\mu$$

$$\frac{1}{2} \frac{S \left( -\text{erf}\left(\frac{1}{4} \frac{\sqrt{2} \left( 2 \ln\left(\frac{K}{\text{fwd}}\right) - \sigma^2 \right)}{\sigma}\right) \text{fwd} + K \text{erf}\left(\frac{1}{4} \frac{\sqrt{2} \left( 2 \ln\left(\frac{K}{\text{fwd}}\right) + \sigma^2 \right)}{\sigma}\right) \text{fwd} + \text{fwd} - K \right)}{\text{fwd}}$$

After substituting fwd and sigma that of course is seen to be the BS formula

```
> 'BSCall(S,K,t,r,v)='S*int((exp(mu)-K/fwd)*Q(mu,sigma),mu = ln(K/fwd) ..
infinity)';
% assuming 0<sigma:
subs(fwd=exp(r*t)*S,%): subs(sigma=v*sqrt(t),%):
expand(%): simplify(%symbolic):
is(%);
```

$$\text{BSCall}(S, K, t, r, v) = S \int_{\ln\left(\frac{K}{\text{fwd}}\right)}^{\infty} \left(e^{\mu} - \frac{K}{\text{fwd}}\right) Q(\mu, \sigma) d\mu$$

true

The 'reason' is the following formula for an undiscounted call (kicking out rates and time with 'normalized' pricing in terms of log moneyness  $m = \ln\left(\frac{K}{\text{fwd}}\right)$  and std deviation  $\sigma = v\sqrt{t}$ ):

```
> 'fwd*BSCall(1,exp(m),1,0,sigma)='exp(r*t)*BSCall(S,K,t,r,v)';

subs(m=ln(K/fwd),%):
subs(fwd=exp(r*t)*S,%):
subs(sigma=v*sqrt(t),%):

combine(%ln):
expand(%):
simplify(%symbolic):
normal(%):
is(%);
```

$$\text{fwd BSCall}(1, e^m, 1, 0, \sigma) = e^{(r t)} \text{BSCall}(S, K, t, r, v)$$

true

## Rediscovering the pdf over log moneyness

Similar to Breeden-Litzenberger one can get the pdf over moneyness by differentiation, but with respect to logarithmic moneyness. For that take any P and go:

```
> Integral:=Int((exp(mu)-exp(m))*P(mu, sigma),mu=m..infinity);
Diff(Integral,m): % = value(%);
Diff(Integral,m$2): % = value(%);
```

$$\text{Integral} := \int_m^{\infty} (e^{\mu} - e^m) P(\mu, \sigma) d\mu$$

$$\frac{\partial}{\partial m} \left( \int_m^{\infty} (e^{\mu} - e^m) P(\mu, \sigma) d\mu \right) = \int_m^{\infty} -e^m P(\mu, \sigma) d\mu$$

$$\frac{\partial^2}{\partial m^2} \left( \int_m^{\infty} (e^{\mu} - e^m) P(\mu, \sigma) d\mu \right) = \int_m^{\infty} -e^m P(\mu, \sigma) d\mu + e^m P(m, \sigma)$$

Hence  $P(\mu, \sigma)$  is given through combining the first and second derivative:

```
> Call:=exp(-r*t)*fwd*Int((exp(mu)-exp(m))*P(mu, sigma),mu=m..infinity);
``;
exp(-r*t)*fwd*P(m,sigma)= 'exp(-m)*(diff(Call,m$2)-diff(Call,m$1))';
expand(%): simplify(%):
is(%);
```

$$\text{Call} := e^{(-r t)} \text{fwd} \int_m^{\infty} (e^{\mu} - e^m) P(\mu, \sigma) d\mu$$

$$e^{(-r t)} \text{fwd} P(m, \sigma) = e^{(-m)} \left( \left( \frac{\partial^2}{\partial m^2} \text{Call} \right) - \left( \frac{\partial}{\partial m^1} \text{Call} \right) \right)$$

true

which for the undiscounted call can be read as

```
> Call:='fwd*Int((exp(mu)-exp(m))*Q(mu,sigma),mu = m .. infinity)';
``;
'Q(m,sigma)=' '1/exp(m)/fwd*(diff(Call,m$2)-diff(Call,m$1))';
expand(%): simplify(%):
is(%);
```

$$\text{Call} := \text{fwd} \int_m^{\infty} (e^{\mu} - e^m) Q(\mu, \sigma) d\mu$$

$$Q(m, \sigma) = \frac{\left( \frac{\partial^2}{\partial m^2} \text{Call} \right) - \left( \frac{\partial}{\partial m^1} \text{Call} \right)}{e^m \text{fwd}}$$

true

note that the denominator is just the strike:

```
> K = 'eval(exp(m)*fwd,m=ln(K/fwd))';
simplify(%): is(%);
```

$$K = e^m \text{fwd} \Big|_{m = \ln\left(\frac{K}{\text{fwd}}\right)}$$

true

By homogeneity one can restrict to  $\text{fwd} = 1$  and then the RND over log space is

```
> 'logRND(m)=' 'exp(-m)*(diff(Call(m),m$2)-diff(Call(m),m$1))';
'Call(m)=' 'Int((exp(mu)-exp(m))*logRND(mu),mu = m .. infinity)';
```

$$\log \text{RND}(m) = e^{(-m)} \left( \left( \frac{d^2}{dm^2} \text{Call}(m) \right) - \left( \frac{d}{dm^1} \text{Call}(m) \right) \right)$$

$$\text{Call}(m) = \int_m^\infty (e^\mu - e^m) \log \text{RND}(\mu) d\mu$$

Let us rediscover that identity again: the undiscouted call in terms of log moneyness is given by fwd BSCall(1, e<sup>m</sup>, 1, 0, σ), so the log RND is

$$e^{(-m)} \left( \left( \frac{\partial^2}{\partial m^2} \text{BSCall}(1, e^m, 1, 0, \sigma) \right) - \left( \frac{\partial}{\partial m^1} \text{BSCall}(1, e^m, 1, 0, \sigma) \right) \right).$$

Now let Maple evaluate the integral (with positive σ)

```
> 'fwd*Int((exp(mu)-exp(m))*eval(exp(-m)*
(diff(BSCall(1,exp(m),1,0,sigma),`$`(m,2))-diff(BSCall(1,exp(m),1,0,sigma),`$`(m,
1))),m = mu),
mu = m .. infinity)';
```

```
value(%): CancelInverses(%): % assuming 0<sigma;
```

$$\text{fwd} \int_m^\infty (e^\mu - e^m) e^{(-m)} \left( \left( \frac{\partial^2}{\partial m^2} \text{BSCall}(1, e^m, 1, 0, \sigma) \right) - \left( \frac{\partial}{\partial m^1} \text{BSCall}(1, e^m, 1, 0, \sigma) \right) \right) d\mu$$

$$\text{fwd} \left( \frac{1}{2} + \frac{1}{2} e^m \operatorname{erf} \left( \frac{(2m + \sigma^2) \sqrt{2}}{4\sigma} \right) - \frac{1}{2} e^m - \frac{1}{2} \operatorname{erf} \left( \frac{\sqrt{2}(2m - \sigma^2)}{4\sigma} \right) \right)$$

and after substituting fwd = e<sup>(rt)</sup> S, m = ln(K/fwd) and σ = v√t we get

this as the undiscounted BS call price:

```
> 'exp(r*t)*BSCall(S,K,t,r,v)' = 'fwd*Int((exp(mu)-exp(m))*eval(exp(-m)*
(diff(BSCall(1,exp(m),1,0,sigma),`$`(m,2))-diff(BSCall(1,exp(m),1,0,sigma),`$`(m,
1))),m = mu),
mu = m .. infinity)';
```

```
value(%): CancelInverses(%): % assuming 0<sigma:
```

```
# now proceed further with substituting
```

```
subs(m=ln(K/fwd),%): subs(fwd=exp(r*t)*S,%): subs(sigma=v*sqrt(t),%):
```

```
expand(%): simplify(%symbolic):
```

```
is(%);
```

$$e^{(rt)} \text{BSCall}(S, K, t, r, v) =$$

$$\text{fwd} \int_m^\infty (e^\mu - e^m) e^{(-m)} \left( \left( \frac{\partial^2}{\partial m^2} \text{BSCall}(1, e^m, 1, 0, \sigma) \right) - \left( \frac{\partial}{\partial m^1} \text{BSCall}(1, e^m, 1, 0, \sigma) \right) \right) d\mu$$

true