Worst of Call Option - Three Ways 07-30-16

N. T. Gladd

Initialization: Be sure the files *NTGStylesheet2.nb* and *NTGUtilityFunctions.m* is are in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

In[37]:=
```
SetDirectory[NotebookDirectory[]];
(* set directory where source files are located *)
SetOptions[EvaluationNotebook[], (* load the StyleSheet *)
 StyleDefinitions → Get["NTGStylesheet2.nb"]];
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Background

Original notebook *Worst of Call Option - Three Ways 11-25-12*

I consider a European style call option on the worst performing of two stocks, with payoff

$$V_T = \max(\min(S_1(T),\ S_2(T)) - K,\ 0) \qquad (1)$$

Under geometric Brownian motion price dynamics, a plain vanilla call option can be expressed in terms of the cumulative normal distribution function (classic Black Scholes formula). In the case of the “worst of” call option, a closed form expression can be obtained in terms of the cumulative binormal distribution function — but the calculation is tedious. The fair value of a worst of option and a sketch of the derivation is given in *Options on the Minimum or the Maximum of Two Risky Assets*, Rene M. Stulz, *Journal of Financial Economics* 10, (1982) pp 161-185. However, I cannot find further details of this derivation in subsequent literature. There do exist similar results for min and max options valued using numeraire methods.

The derivation of a closed form expression for the worst of option is mainly of technical interest. Almost any real world complication would break the assumption of pure GBM dynamics and Monte Carlo simulation would be the method of choice for valuation under more realistic dynamics.

However, valuing this option provides an opportunity to practice some *Mathematica* techniques that are useful in other contexts. I develop and implement three valuation methods for this option —

1) Straightforward numerical approach using double quadrature —*WorstOfCallDQ*

2) Numerical approach using single quadrature that requires the derivation of the distribution of min(Log(S_1(T)), Log(S_2(T))) —*WorstOfCallSQ*

3) Derivation of the closed form analytic expression using symbolic manipulation techniques —*WorstOfCallAnalytic*

In the beginning, I summarize results using typeset equations. The Mathematica derivations from which some of these equations are generated are presented below.

Under arbitrage free pricing theory, the fair value of a call option on the minimum of two assets is

$$V = \mathbb{E}_Q\left[e^{-r\,T}\max[\min(S_1(T),\ S_2(T)) - K,\ 0]\right]$$

$$= e^{-r\,T}\int_0^\infty \mathrm{dM}\, f(M)\max[M - K,\ 0] \tag{2}$$

$$= e^{-r\,T}\int_K^\infty \mathrm{dM}\, f(M)\, M - e^{-r\,T} K \int_K^\infty \mathrm{dM}\, f(M)$$

where f(M) is the risk-neutral distribution for

$$M = M(T) = \min(S_1(T),\ S_2(T)) \tag{3}$$

1) Valuation by double quadrature

The GBM dynamics are

$$\begin{aligned}\frac{\mathrm{dS_1}}{S_1} &= (r - q_{\mathrm{S1}})\,\mathrm{dt} + \sigma_{\mathrm{S1}}\,\mathrm{dz}_{1\,t}\\ \frac{\mathrm{dS_2}}{S_2} &= (r - q_{\mathrm{S2}})\,\mathrm{dt} + \sigma_{\mathrm{S2}}\,\mathrm{dz}_{2,t}\end{aligned} \tag{4}$$

where the dz are correlated Weiner processes. For these dynamics

$$\begin{aligned}S_1(T) &= S_{10}\exp\left(\left(r - q_{\mathrm{S1}} - \frac{\sigma_{\mathrm{S1}}^2}{2}\right)T + \sigma_{\mathrm{S1}}\sqrt{T}\ \epsilon_1\right)\\ S_2(T) &= S_{20}\exp\left(\left(r - q_{\mathrm{S2}} - \frac{\sigma_{\mathrm{S2}}^2}{2}\right)T + \sigma_{\mathrm{S2}}\sqrt{T}\ \epsilon_2\right)\end{aligned} \tag{5}$$

where ϵ_1, ϵ_2 are random numbers drawn from the cumulative standard bivariate normal distribution $\mathcal{N}_2$({0,0}, {1,1}, ρ). Using *Mathematica* this distribution is

In[39]:=
```
PDF[BinormalDistribution[{0, 0}, {1, 1}, ρ], {ϵ1, ϵ2}]
```

Out[39]=
$$\frac{e^{-\frac{\epsilon_1^2-2\rho\,\epsilon_1\,\epsilon_2+\epsilon_2^2}{2\,(1-\rho^2)}}}{2\,\pi\,\sqrt{1-\rho^2}}$$

It is often more convenient to use a Cholesky decomposition to write $\epsilon_1 = \eta_1$, $\epsilon_2 = \rho\,\eta_1 + \sqrt{1-\rho^2}\,\eta_2$ with the η being iid draws from $\mathcal{N}(0,1)$.

In[40]:=
```
{ϵ1, ϵ2} == CholeskyDecomposition[{{1, ρ}, {ρ, 1}}].{η1, η2} //
 Simplify[#, {ρ ∈ Reals}] &
```

Out[40]=
$$\{\epsilon_1, \epsilon_2\} == \left\{\eta_1 + \rho\,\eta_2,\ \sqrt{1-\rho^2}\,\eta_2\right\}$$

It is straightforward to use these defining formulas to implement a double quadrature method.

Coded form for double quadrature method *WorstOfCallDQ*

In[41]:=
```
Clear[WorstOfCallDQ];
WorstOfCallDQ[K_, T_, S10_, q1_, σ1_, S20_, q2_, σ2_, ρ_, r_] :=
 Module[{μ1 = r - q1, μ2 = r - q2, ρb = Sqrt[1 - ρ^2], S1T, S2T},
  S1T[η1_] := S10 Exp[(μ1 - σ1^2/2) T + σ1 Sqrt[T] η1];
  S2T[η1_, η2_] := S20 Exp[(μ2 - σ2^2/2) T + σ2 Sqrt[T] (η1 ρ + η2 ρb)];
  Exp[-r T] NIntegrate[
    n[η1] n[η2] Max[Min[S1T[η1], S2T[η1, η2]] - K, 0], {η1, -∞, ∞}, {η2, -∞, ∞}]]
```

2) Single quadrature

For the single quadrature method, I follow Stulz and derive a formula for

$$m = m(T) = \min(\log(S_1(T),\ \log(S_2(T)) = \min(s_1,\ s_2) \tag{6}$$

The dynamics for s can be obtained from the logarithm of (5)

$$s_1 - s_{10} = \left(r - q_{S1} - \frac{\sigma_{S1}^2}{2}\right) T + \sigma_{S1}\sqrt{T}\,\epsilon_1$$

$$s_2 - s_{20} = \left(r - q_{S2} - \frac{\sigma_{S2}^2}{2}\right) T + \sigma_{S2}\sqrt{T}\,\epsilon_2 \tag{7}$$

For example, solving for ϵ_1 I can write

$$\epsilon_1(s_1) = \frac{s_1 - s_{10} - \left(r - q_{S1} - \frac{\sigma_{S1}^2}{2}\right) T}{\sigma_{S1}\sqrt{T}} \equiv \frac{s_1 - \mu_1}{\sigma_1}$$

The bivariate distribution for the log prices

In[43]:=
```
PDF[BinormalDistribution[{μ1, μ2}, {σ1, σ2}, ρ], {s1, s2}]
```

Out[43]=
$$e^{-\frac{\frac{(s_1-\mu_1)^2}{\sigma_1^2}+\frac{(s_2-\mu_2)^2}{\sigma_2^2}-\frac{2\rho\,(s_1-\mu_1)\,(s_2-\mu_2)}{\sigma_1\,\sigma_2}}{2\,(1-\rho^2)}} \Big/ \left(2\,\pi\,\sqrt{1-\rho^2}\;\sigma_1\,\sigma_2\right)$$

$$f_{\text{biv}}(s_1, s_2) = \frac{e^{-\frac{\frac{(s_1-\mu_1)^2}{2\,\sigma_1^2}+\frac{(s_2-\mu_2)^2}{2\,\sigma_2^2}-\frac{\rho(s_1-\mu_1)(s_2-\mu_2)}{\sigma_1\,\sigma_2}}{1-\rho^2}}}{2\,\pi\sqrt{1-\rho^2}\;\sigma_1\,\sigma_2} \tag{8}$$

I require f(m) = $f(\min(s_1, s_2))$. The probability that m < k is

$$\mathbb{P}(m < \alpha) = \int ds_1 \int_{\mathcal{A}(m)} ds_2\, f_{\text{biv}}(s_1, s_2) \tag{9}$$

where $\mathcal{A}$ represents the area for which min[s_1, s_2] is true. Following Stulz, I can write this

$$\mathbb{P}(m < \alpha) = 1 - \int_\alpha^\infty \int_\alpha^\infty f(s_1, s_2)\, ds_2\, ds_1 \tag{10}$$

and differentiate to obtain the probability density

$$f(m) = \int_m^\infty f(\alpha, y)\, dy + \int_m^\infty f(x, \alpha)\, dx \tag{11}$$

In Section 1, I calculate

$$f(m) = \frac{n\left(\frac{m-\mu_2}{\sigma_2}\right) \mathcal{N}\left(\frac{\rho\,\sigma_1\,(m-\mu_2)+\sigma_2\,(\mu_1-m)}{\sigma_1\,\sigma_2\,\bar{\rho}}\right)}{\sigma_2} + \frac{n\left(\frac{m-\mu_1}{\sigma_1}\right) \mathcal{N}\left(\frac{\rho\,\sigma_2\,(\mu_1-m)+\sigma_1\,(m-\mu_2)}{\sigma_1\,\sigma_2\,\bar{\rho}}\right)}{\sigma_1}$$

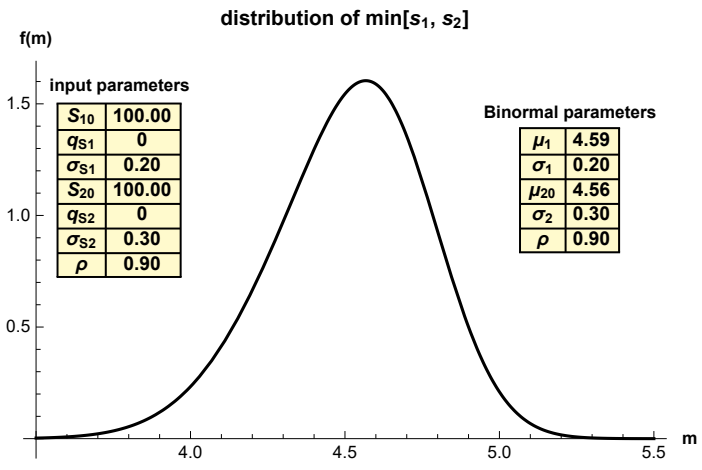
For a nominal set of parameters, f(m) takes the form

In[44]:=

```
Module[{S10 = 100, S20 = 100, qS1 = 0, qS2 = 0, σS1 = 0.2, σS2 = 0.3,
  ρ = 0.9, T = 1.0, r = 0, K = 90, nSamples = 10000, μ1, μ2, σ1, σ2, g},
 μ1 = Log[S10] + (r - qS1 - σS1^2/2) T;
 μ2 = Log[S20] + (r - qS2 - σS2^2/2) T;
 σ1 = σS1 √T;
 σ2 = σS2 √T;

 Module[{info, insert},
  info[1] = {{"S10", "qS1", "σS1", "S20", "qS2", "σS2", "ρ"},
    NF2 /@ {S10, qS1, σS1, S20, qS2, σS2, ρ}};
  insert[1] = LGrid[Transpose@info[1], "input parameters"];
  info[2] = {{"μ1", "σ1", "μ20", "σ2", "ρ"}, NF2 /@ {μ1, σ1, μ2, σ2, ρ}};
  insert[2] = LGrid[Transpose@info[2], "Binormal parameters"];
  Plot[fMin[m, μ1, μ2, σ1, σ2, ρ, √(1 - ρ^2)], {m, 3.5, 5.5},
   AxesLabel → {Stl["m"], Stl["f(m)"]},
   PlotLabel → Stl["distribution of min[s1, s2]"],
   Epilog → {Inset[insert[1], Scaled[{0.15, 0.7}]],
     Inset[insert[2], Scaled[{0.85, 0.7}]]}, PlotStyle → Black]]]
```

Out[44]=



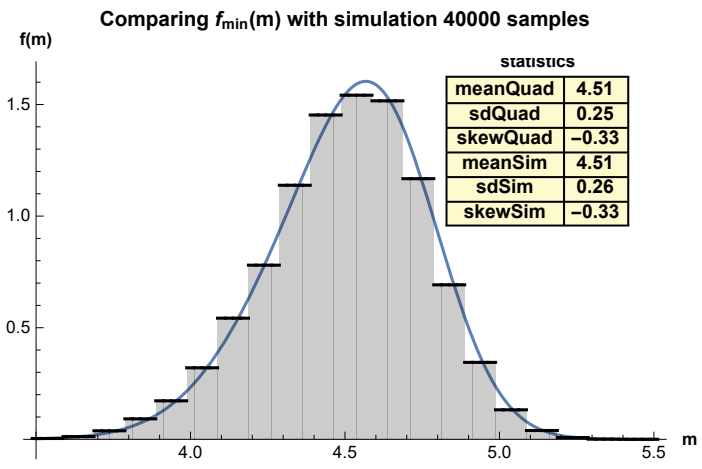
Although there little need in this case, it can be useful to check the derived distribution against simulation. An instance of the simulation consists of generating m = {s_1, s_2}

In[45]:=

```
Module[{S10 = 100, S20 = 100, qS1 = 0, qS2 = 0, σS1 = 0.2,
  σS2 = 0.3, ρ = 0.9, T = 1.0, r = 0, K = 90, nSamples = 40000,
  μ1, μ2, σ1, σ2, f, dataSim, mean, var, sd, skew, ℋ, g},
 μ1 = Log[S10] + (r - qS1 - σS1^2/2) T;
 μ2 = Log[S20] + (r - qS2 - σS2^2/2) T;
 σ1 = σS1 √T;
 σ2 = σS2 √T;
 mean = NIntegrate[m fMin[m, μ1, μ2, σ1, σ2, ρ, √(1 - ρ^2)], {m, 0, ∞}];
 var = NIntegrate[(m - mean)^2 fMin[m, μ1, μ2, σ1, σ2, ρ, √(1 - ρ^2)], {m, 0, ∞}];
 skew =
  NIntegrate[(m - mean)^3 fMin[m, μ1, μ2, σ1, σ2, ρ, √(1 - ρ^2)], {m, 0, ∞}] / var^(3/2);

 dataSim =
  Min /@ RandomVariate[BinormalDistribution[{μ1, μ2}, {σ1, σ2}, ρ], nSamples];
 ℋ = HistogramDistribution[dataSim];
 Module[{info, insert, lab},
  info = {{"meanQuad", "sdQuad", "skewQuad", "meanSim", "sdSim", "skewSim"},
    NF2 /@ {mean, √var, skew, Mean[ℋ], StandardDeviation[ℋ], Skewness[ℋ]}};
  insert = LGrid[Transpose@info, "statistics"];
  lab = Stl@StringForm["Comparing f_min(m) with simulation `` samples", nSamples];
  g[1] = Plot[fMin[m, μ1, μ2, σ1, σ2, ρ, √(1 - ρ^2)],
    {m, 3.5, 5.5}, AxesLabel → {Stl["m"], Stl["f(m)"]},
    PlotLabel → lab,
    Epilog → Inset[insert, Scaled[{0.8, 0.8}]]];
  g[2] = DiscretePlot[PDF[ℋ, m],
    {m, 3.5, 5.5, 0.025}, ExtentSize → Full, PlotStyle → Black];
  Show[{g[1], g[2]}]]]
```

Out[45]=



Given f(m), the worst of option is determined by

$$V = e^{-r\,T} \int_k^\infty \mathrm{dm}\, f(m)\,(e^m - k) \tag{12}$$

where k = Log[K].

3) Analytic method

The integral in equation (12) can be evaluated in terms of known functions. Write

$$V = e^{-r\,T}\,(\mathcal{J}(1) - K\,\mathcal{J}(0)) \tag{13}$$

where

$$\mathcal{J}(\alpha) = \int_k^\infty \mathrm{dm}\, e^{\alpha\, m}\, f(m) \tag{14}$$

In Section 2 I calculate

$$\mathcal{J}(\alpha) = e^{\alpha\,\mu_2}\,\mathcal{I}\left[\frac{\rho\,\sigma_1 - \sigma_2}{\overline{\rho}\,\sigma_1},\ \frac{\mu_1 - \mu_2}{\overline{\rho}\,\sigma_1},\ \frac{k - \mu_2}{\sigma_2},\ \alpha\,\sigma_2\right] + e^{\alpha\,\mu_1}\,\mathcal{I}\left[\frac{\sigma_1 - \rho\,\sigma_2}{\overline{\rho}\,\sigma_2},\ \frac{\mu_1 - \mu_2}{\overline{\rho}\,\sigma_2},\ \frac{k - \mu_1}{\sigma_1},\ \alpha\,\sigma_1\right] \tag{15}$$

where

$$\mathcal{I}(a,\ b,\ c,\ d) = \int_c^\infty \mathrm{dx}\, n(x)\,\mathcal{N}(a\,x + b)\,e^{d\,x}$$

$$= e^{\frac{d^2}{2}}\left(\mathcal{N}\left(\frac{a\,d + b}{\sqrt{a^2 + 1}}\right) - \mathcal{N}_2\left(c - d,\ \frac{a\,d + b}{\sqrt{a^2 + 1}},\ -\frac{a}{\sqrt{a^2 + 1}}\right)\right) \tag{16}$$

with

$$n(x) = \frac{1}{\sqrt{2\,\pi}}\,e^{-\frac{x^2}{2}}$$

$$\mathcal{N}(A) = \int_{-\infty}^A \mathrm{dx}\ n(x) \tag{17}$$

$$\mathcal{N}_2(A,\ B,\ \rho) = \int_{-\infty}^A \mathrm{dx} \int_{-\infty}^B \mathrm{dy}\ \frac{1}{\sqrt{2\,\pi\,(1 - \rho^2)}}\,e^{-\frac{x^2 - 2\,x\,y\,\rho + y^2}{2\,(1 - \rho^2)}}$$

The formula (16) is derived in *Section 3*.

Obtaining this analytical result is a somewhat dubious accomplishment. Unless one has confidence in an approximate form for $\mathcal{N}_2$, the double quadrature in (16) must still be performed. If that calculation is indeed required it is easier to use the method implemented in *WorstOfCallDQ*.

Nonetheless, it is instructive to utilize the symbolic manipulation capabilities of *Mathematica* to perform the derivation of the closed form since the techniques are useful in other contexts. I present the calculations in Section 2.

Results and discussion

I first check that the three different approaches give the same answer.

In[46]:=
```
Module[{S10 = 100, S20 = 100, qS1 = 0, qS2 = 0,
  σS1 = 0.2, σS2 = 0.3, ρ = 0.0, T = 1.0, r = 0, K = 90},
 {WorstOfCallDQ[K, T, S10, qS1, σS1, S20, qS2, σS2, ρ, r],
  WorstOfCallSQ[K, T, S10, qS1, σS1, S20, qS2, σS2, ρ, r],
  WorstOfCallAnalytic[K, T, S10, qS1, σS1, S20, qS2, σS2, ρ, r]}]
```

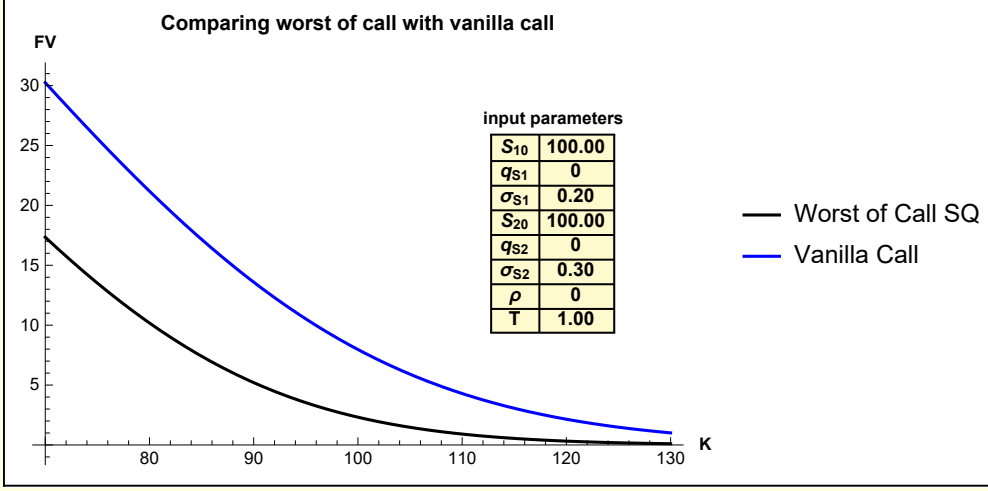
Out[46]=
```
{5.20237, 5.20237, 5.20353}
```

It is instructive to compare the worst-of call with a plain vanilla call. The S_1 parameters are used to evaluate the vanilla call. The parameters for S_2 are the same except that $\sigma_2 = 1.5\ \sigma_1$.

As intuition would suggest, a call option on the minimum of S_1 and S_2 is less valuable than a call option on just S_1.

In[47]:=
```
Module[{S10 = 100, S20 = 100, qS1 = 0, qS2 = 0, σS1 = 0.2, σS2 = 0.3, ρ = 0.0,
  T = 1.0, r = 0, Kmin = 70, Kmax = 130, info, insert, results, lab, FVWOC, FVEC},
 info = {{"S10", "qS1", "σS1", "S20", "qS2", "σS2", "ρ", "T"},
   NF2 /@ {S10, qS1, σS1, S20, qS2, σS2, ρ, T}};
 insert = LGrid[Transpose@info, "input parameters"];
 results = Table[{K, WorstOfCallSQ[K, T, S10, qS1, σS1, S20, qS2, σS2, ρ, r],
    EuroCall[S10, K, r, qS1, σS1, T]}, {K, Kmin, Kmax, 5}];
 FVWOC = Interpolation@results[[All, {1, 2}]];
 FVEC = Interpolation@results[[All, {1, 3}]];
 lab = Stl["Comparing worst of call with vanilla call"];
 Plot[{FVWOC[K], FVEC[K]}, {K, 70, 130}, PlotStyle → {Black, Blue},
  PlotLegends → {"Worst of Call SQ", "Vanilla Call"},
  PlotLabel → lab, AxesLabel → {Stl["K"], Stl["FV"]},
  Epilog → Inset[insert, Scaled[{0.8, 0.6}]]]]
```

Out[47]=



Variants of this “Worst-of” option and discussion of their financial use may be found by Googling on “Worst of,” “ Best of,” or “Rainbow” option.

Derivations

Section 0 *Mathematica* preliminaries

In the following calculations, various integrals will result in conditional expressions that have to be manually simplified by imposing constraints on parameters. Labor can be reduced if parametric assumptions are specified in the global variable $Assumptions.

In[48]:=
```
$Assumptions =
  {σ1 ∈ Reals, σ1 > 0, σ2 ∈ Reals, σ2 > 0, ρ ∈ Reals, ρ^2 < 1, ρ̄ ∈ Reals, ρ̄ > 0};
```

Mathematica expresses various integrals of interest in terms of the error function. I introduce some rules for transforming error functions into forms involving the cumulative standard normal distribution, which is preferred in the financial world.

In[49]:=
```
ErfRules = {Erfc[x_] → 1 - Erf[x], Erf[x_] → 2 N[√2 x] - 1};
```

Section 1 Derivation of probability density f(m) in eqn (14)

The bivariate normal distribution will be used

In[50]:=
```
fbiv = PDF[BinormalDistribution[{μ1, μ2}, {σ1, σ2}, ρ], {s1, s2}]
```

Out[50]=
$$e^{-\frac{\frac{(s_1-\mu_1)^2}{\sigma_1^2}+\frac{(s_2-\mu_2)^2}{\sigma_2^2}-\frac{2\rho\,(s_1-\mu_1)\,(s_2-\mu_2)}{\sigma_1\,\sigma_2}}{2\,(1-\rho^2)}} \Big/ \left(2\,\pi\,\sqrt{1-\rho^2}\;\sigma_1\,\sigma_2\right)$$

Construct the probability of equation (9)

In[51]:=
```
w1[1] = 1 - Integrate[f[s1, s2], {s1, m, ∞}, {s2, m, ∞}]
```

Out[51]=
$$1-\int_m^\infty\int_m^\infty f[s_1, s_2]\,ds_2\,ds_1$$

Differentiate to obtain the probability density

In[52]:=
```
w1[2] = D[w1[1], m] // Simplify
```

Out[52]=
$$\int_m^\infty f[m, s_2]\,ds_2-\int_m^\infty -f[s_1, m]\,ds_1$$

It is operationally convenient to evaluate these two integrals separately

In[53]:=
```
w11[1] = fbiv /. s₁ → m;
w11[2] = Integrate[w11[1], {s₂, m, ∞}] /. (-1 + ρ²) → - ρb² //. ErfRules ;
w11[2] = w11[2] // PowerExpand;
w11[3] = w11[2] /. a_. Exp[b_] → a √(2 π) n[√(-2 b)] // PowerExpand // Simplify;
w11[3] = w11[3] /. N[x_] → 1 - N[-x] /. 1 - ρ² → ρb² // PowerExpand
```

Out[57]=
$$\frac{1}{\sigma_1} n\left[\frac{m-\mu_1}{\sigma_1}\right] N\left[-\frac{(m-\mu_2)\ \sigma_1+\rho\ (-m+\mu_1)\ \sigma_2}{\rho_b\ \sigma_1\ \sigma_2}\right]$$

Note on Mathematica practice: The choice of operations and the sequence involved in obtaining the previous result are not intuitive. In practice, one performs such operations step by step and makes modifications to achieve a desirable form for the expression. Only in retrospect can the operations be combined into a compact sequence, or encapsulated in a function.

In[58]:=
```
w12[1] = fbiv /. s₂ → m;
w12[2] =
  Integrate[w12[1], {s₁, m, ∞}] /. (-1 + ρ²) → -ρb²  //. ErfRules  // Simplify ;
w12[2] = w12[2] /. 1 / √(1 - ρ²) → 1 / ρb // PowerExpand // Simplify;
w12[3] = w12[2] /. a_. Exp[b_] → a √(2 π) n[√(-2 b)] // PowerExpand // Simplify
```

Out[61]=
$$\frac{n\left[\frac{m-\mu_2}{\sigma_2}\right] N\left[\frac{\rho\ (m-\mu_2)\ \sigma_1+(-m+\mu_1)\ \sigma_2}{\rho_b\ \sigma_1\ \sigma_2}\right]}{\sigma_2}$$

In[62]:=
```
w1["final"] = w11[3] + w12[3]
```

Out[62]=
$$\frac{n\left[\frac{m-\mu_2}{\sigma_2}\right] N\left[\frac{\rho\ (m-\mu_2)\ \sigma_1+(-m+\mu_1)\ \sigma_2}{\rho_b\ \sigma_1\ \sigma_2}\right]}{\sigma_2}+\frac{1}{\sigma_1} n\left[\frac{m-\mu_1}{\sigma_1}\right] N\left[-\frac{(m-\mu_2)\ \sigma_1+\rho\ (-m+\mu_1)\ \sigma_2}{\rho_b\ \sigma_1\ \sigma_2}\right]$$

In[63]:=
```
f[m] == w1["final"] // TraditionalForm
```

Out[63]//TraditionalForm=
$$f(m)=\frac{n\left(\frac{m-\mu_2}{\sigma_2}\right) N\left(\frac{\rho\,\sigma_1\,(m-\mu_2)+\sigma_2\,(\mu_1-m)}{\sigma_1\,\sigma_2\,\rho_b}\right)}{\sigma_2}+\frac{n\left(\frac{m-\mu_1}{\sigma_1}\right) N\left(-\frac{\rho\,\sigma_2\,(\mu_1-m)+\sigma_1\,(m-\mu_2)}{\sigma_1\,\sigma_2\,\rho_b}\right)}{\sigma_1}$$

Coded form for the single quadrature method *WorstOfCallSQ*

Coded form - f(m) distribution

It is really really good practice to code an expression by copying and pasting an analytically derived expression into a function expression, as opposed to retyping it. In this regard *Mathematica* has a problem with respect to subscripted variables. They can be easily used in analytic calculations but, because they are not formal “Symbols” they cannot be used as arguments in a function call (not easily, that is).

So, the practice here will be to transform a derived Mathematica expression (replete with subscripts and

other decorated notation) into a form involving symbols by means of rewrite rules.

In[64]:=
```
w1["final"] /. {μ1 → μ1, μ2 → μ2, σ1 → σ1, σ2 → σ2, ρb → ρb}
```

Out[64]=
$$\frac{n\left[\frac{m-\mu 2}{\sigma 2}\right]\mathcal{N}\left[\frac{(m-\mu 2)\,\rho\,\sigma 1+(-m+\mu 1)\,\sigma 2}{\rho b\,\sigma 1\,\sigma 2}\right]}{\sigma 2}+\frac{1}{\sigma 1}n\left[\frac{m-\mu 1}{\sigma 1}\right]\mathcal{N}\left[-\frac{(m-\mu 2)\,\sigma 1+(-m+\mu 1)\,\rho\,\sigma 2}{\rho b\,\sigma 1\,\sigma 2}\right]$$

The resulting expression can then copied and pasted into a function definition.

In[65]:=
```
Clear[fMin];
fMin[m_, μ1_, μ2_, σ1_, σ2_, ρ_, ρb_] :=

 1/σ2 n[(m - μ2)/σ2] 𝒩[((m - μ2) ρ σ1 + (-m + μ1) σ2)/(ρb σ1 σ2)] +
  1/σ1 n[(m - μ1)/σ1] 𝒩[-(((m - μ2) σ1 + (-m + μ1) ρ σ2)/(ρb σ1 σ2))]
```

The option is valued by

In[67]:=
```
Clear[WorstOfCallSQ];
WorstOfCallSQ[K_, T_, S10_, qS1_, σS1_, S20_, qS2_, σS2_, ρ_, r_] :=
 Module[{V, μ1, σ1, μ2, σ2},
  μ1 = Log[S10] + (r - qS1 - σS1^2/2) T;
  μ2 = Log[S20] + (r - qS2 - σS2^2/2) T;
  σ1 = σS1 √T;
  σ2 = σS2 √T;
  V = Exp[-r T]
    NIntegrate[(Exp[m] - K) fMin[m, μ1, μ2, σ1, σ2, ρ, √(1 - ρ^2)], {m, Log[K], ∞}]]
```

Section 2a Transformation of $\mathcal{J}(\alpha)$ into a form involving integrals of normal distributions

I transform the integral appearing in equation (14) into a form involving known functions

$$\mathcal{J}(\alpha)=\int_k^\infty \mathrm{dm}\, e^{\alpha m} f(m)$$

To facilitate manipulations of the unevaluated integral, I introduce a special data structure *Int* for representing an integral

$$\int_{-\infty}^{a} \mathbf{dx}\, \mathbf{f[x]} = \mathbf{Int[d[x]\, f[x],\ \{x,\ -\infty,\ a\}]}$$

I can invoke *Mathematica*'s usual integration capabilities by applying some rules to this structure. For example

In[69]:=
```
Int[d[x] Exp[-x^2/2], {x, -∞, a}] /. d[x] → 1 /. Int → Integrate
```

Out[69]=
$$\sqrt{\frac{\pi}{2}}\left(1+\text{Erf}\left[\frac{a}{\sqrt{2}}\right]\right)$$

or

In[70]:=
```
Int[d[x] Exp[-x^2/2], {x, -∞, a}] /. d[x] → 1 /. Int → Integrate /. ErfRules
```

Out[70]=
$$\sqrt{2\pi}\,\mathcal{N}[a]$$

I write a function that simplifies expressions involving Int

In[71]:=
```
Clear[IntSimplification];
IntSimplification[f_, x_] :=
 Module[{g = ExpandAll[f]},
  (* distribute Int across sums *)
  g = g //. Int[a_ + b_, lim_] → Int[a, lim] + Int[b, lim];
  (* remove a constant term from Int *)
  g = g //. Int[a_. b_ /; FreeQ[a, x], lim_] :→ a Int[b, lim];
  (* remove a constant term occurring in an exponential function from Int *)
  g = g //. Int[a_. Exp[b_. + c_ x], lim_] :→ Exp[b] Int[a Exp[c x], lim]]
```

Preliminaries complete, the calculation of the probability starts with

In[73]:=
```
w2[1] = Int[d[m] Exp[α m] f[m], {m, k, ∞}]
```

Out[73]=
$$\text{Int}\left[e^{m\alpha}\,d[m]\,f[m],\{m,k,\infty\}\right]$$

Introduce the explicit expression for f(m) derived in the previous section (w11["final"])

In[74]:=
```
w2[2] = w2[1] /. f[m] → w1["final"]
```

Out[74]=
$$\text{Int}\left[e^{m\alpha}\,d[m]\left(\frac{1}{\sigma_2}n\left[\frac{m-\mu_2}{\sigma_2}\right]\mathcal{N}\left[(\rho\,(m-\mu_2)\,\sigma_1+(-m+\mu_1)\,\sigma_2)\,/\,(\rho_b\,\sigma_1\,\sigma_2)\right]+\right.\right.$$
$$\left.\left.\frac{1}{\sigma_1}n\left[\frac{m-\mu_1}{\sigma_1}\right]\mathcal{N}\left[-\left(\left((m-\mu_2)\,\sigma_1+\rho\,(-m+\mu_1)\,\sigma_2\right)\,/\,(\rho_b\,\sigma_1\,\sigma_2)\right)\right]\right),\{m,k,\infty\}\right]$$

In[75]:=
```
w2[3] = IntSimplification[w2[2], m] // Simplify
```

Out[75]=
$$\frac{1}{\sigma_1}\text{Int}\left[e^{m\alpha}\,d[m]\,n\left[\frac{m-\mu_1}{\sigma_1}\right]\mathcal{N}\left[\frac{(-m+\mu_2)\,\sigma_1+\rho\,(m-\mu_1)\,\sigma_2}{\rho_b\,\sigma_1\,\sigma_2}\right],\{m,k,\infty\}\right]+$$
$$\frac{1}{\sigma_2}\text{Int}\left[e^{m\alpha}\,d[m]\,n\left[\frac{m-\mu_2}{\sigma_2}\right]\mathcal{N}\left[\frac{\rho\,(m-\mu_2)\,\sigma_1+(-m+\mu_1)\,\sigma_2}{\rho_b\,\sigma_1\,\sigma_2}\right],\{m,k,\infty\}\right]$$

The next step is to standardize these integrals by a change of variable that results in the explicit appearance of the normal distribution *n*[...].

The following function effects this change of variable.

In[76]:=
```
Clear[IntChangeVariables];
IntChangeVariables[a_. Int[arg_, {m_, ll_, ul_}], eqn_] :=
 Module[{rule, drule, narg, nll, nul, nlim},
  rule = Solve[eqn, m][[1, 1]];
  drule = d[m] → D[rule[[2]], M] d[M];
  narg = arg /. drule /. rule // Simplify;
  nlim = {M, eqn[[2]] /. m → ll, eqn[[2]] /. m → ul};
  a Int[narg, nlim] // Simplify]
```

In[78]:=
```
w2[4] = IntChangeVariables[#, M == (# /. a_. Int[b_. n[c_], lim_] → c)] & /@ w2[3]
```

Out[78]=
$$\frac{1}{\sigma_1}\text{Int}\left[e^{\alpha\,(\mu_1+M\,\sigma_1)}\,d[M]\;\sigma_1\,n[M]\;N\left[\frac{-\mu_1+\mu_2-M\,\sigma_1+M\,\rho\,\sigma_2}{\rho_b\,\sigma_2}\right],\left\{M,\frac{k-\mu_1}{\sigma_1},\infty\right\}\right]+$$
$$\frac{1}{\sigma_2}\text{Int}\left[e^{\alpha\,(\mu_2+M\,\sigma_2)}\,d[M]\;\sigma_2\,n[M]\;N\left[\frac{\mu_1-\mu_2+M\,\rho\,\sigma_1-M\,\sigma_2}{\rho_b\,\sigma_1}\right],\left\{M,\frac{k-\mu_2}{\sigma_2},\infty\right\}\right]$$

Simplify again

In[79]:=
```
w2[5] = IntSimplification[w2[4], M]
```

Out[79]=
$$e^{\alpha\,\mu_1}\,\text{Int}\left[e^{M\,\alpha\,\sigma_1}\,d[M]\;n[M]\;N\left[\frac{M\,\rho}{\rho_b}-\frac{\mu_1}{\rho_b\,\sigma_2}+\frac{\mu_2}{\rho_b\,\sigma_2}-\frac{M\,\sigma_1}{\rho_b\,\sigma_2}\right],\left\{M,\frac{k}{\sigma_1}-\frac{\mu_1}{\sigma_1},\infty\right\}\right]+$$
$$e^{\alpha\,\mu_2}\,\text{Int}\left[e^{M\,\alpha\,\sigma_2}\,d[M]\;n[M]\;N\left[\frac{M\,\rho}{\rho_b}+\frac{\mu_1}{\rho_b\,\sigma_1}-\frac{\mu_2}{\rho_b\,\sigma_1}-\frac{M\,\sigma_2}{\rho_b\,\sigma_1}\right],\left\{M,\frac{k}{\sigma_2}-\frac{\mu_2}{\sigma_2},\infty\right\}\right]$$

I make another simplification to make things easier for the pattern recognition operations to come

In[80]:=
```
w2[6] = w2[5] /.
  a_. Int[b_. N[c_], lim_] :> a Int[b N[Coefficient[c, M] M + (c /. M → 0)], lim]
```

Out[80]=
$$e^{\alpha\,\mu_1}\,\text{Int}\left[e^{M\,\alpha\,\sigma_1}\,d[M]\;n[M]\;N\left[M\left(\frac{\rho}{\rho_b}-\frac{\sigma_1}{\rho_b\,\sigma_2}\right)-\frac{\mu_1}{\rho_b\,\sigma_2}+\frac{\mu_2}{\rho_b\,\sigma_2}\right],\left\{M,\frac{k}{\sigma_1}-\frac{\mu_1}{\sigma_1},\infty\right\}\right]+$$
$$e^{\alpha\,\mu_2}\,\text{Int}\left[e^{M\,\alpha\,\sigma_2}\,d[M]\;n[M]\;N\left[\frac{\mu_1}{\rho_b\,\sigma_1}-\frac{\mu_2}{\rho_b\,\sigma_1}+M\left(\frac{\rho}{\rho_b}-\frac{\sigma_2}{\rho_b\,\sigma_1}\right)\right],\left\{M,\frac{k}{\sigma_2}-\frac{\mu_2}{\sigma_2},\infty\right\}\right]$$

Apply the pattern matching rule

In[81]:=
```
w2[7] = w2[6] /. coeff_. Int[ Exp[dd_. M] d[M] n[M] N[aa_ M + bb_], {M, cc_, ∞}] :>
    coeff I[aa, bb, cc, dd]
```

Out[81]=
$$e^{\alpha\,\mu_1}\,\mathcal{I}\left[\frac{\rho}{\rho_b}-\frac{\sigma_1}{\rho_b\,\sigma_2},-\frac{\mu_1}{\rho_b\,\sigma_2}+\frac{\mu_2}{\rho_b\,\sigma_2},\frac{k}{\sigma_1}-\frac{\mu_1}{\sigma_1},\alpha\,\sigma_1\right]+$$
$$e^{\alpha\,\mu_2}\,\mathcal{I}\left[\frac{\rho}{\rho_b}-\frac{\sigma_2}{\rho_b\,\sigma_1},\frac{\mu_1}{\rho_b\,\sigma_1}-\frac{\mu_2}{\rho_b\,\sigma_1},\frac{k}{\sigma_2}-\frac{\mu_2}{\sigma_2},\alpha\,\sigma_2\right]$$

In[82]:=
```
w2["final"] = w2[7] // Simplify
```

Out[82]=
$$e^{\alpha \mu_2}\, \mathcal{I}\left[\frac{\rho\,\sigma_1-\sigma_2}{\rho_b\,\sigma_1}, \frac{\mu_1-\mu_2}{\rho_b\,\sigma_1}, \frac{k-\mu_2}{\sigma_2}, \alpha\,\sigma_2\right] + e^{\alpha \mu_1}\, \mathcal{I}\left[\frac{-\sigma_1+\rho\,\sigma_2}{\rho_b\,\sigma_2}, \frac{-\mu_1+\mu_2}{\rho_b\,\sigma_2}, \frac{k-\mu_1}{\sigma_1}, \alpha\,\sigma_1\right]$$

In[83]:=
```
𝒥[α] == w2["final"] // TraditionalForm
```

Out[83]//TraditionalForm=
$$\mathcal{J}(\alpha) = e^{\alpha \mu_2}\, \mathcal{I}\left(\frac{\rho\,\sigma_1-\sigma_2}{\sigma_1\,\rho_b}, \frac{\mu_1-\mu_2}{\sigma_1\,\rho_b}, \frac{k-\mu_2}{\sigma_2}, \alpha\,\sigma_2\right) + e^{\alpha \mu_1}\, \mathcal{I}\left(\frac{\rho\,\sigma_2-\sigma_1}{\sigma_2\,\rho_b}, \frac{\mu_2-\mu_1}{\sigma_2\,\rho_b}, \frac{k-\mu_1}{\sigma_1}, \alpha\,\sigma_1\right)$$

where $\mathcal{I}$ is defined in equation (16).

$$\mathcal{I}(a,\ b,\ c,\ d) = \int_c^\infty dx\, n(x)\, \mathcal{N}(a\,x+b)\, e^{d\,x} = e^{\frac{d^2}{2}}\left(\mathcal{N}\left(\frac{a\,d+b}{\sqrt{a^2+1}}\right) - \mathcal{N}_2\left(c-d, \frac{a\,d+b}{\sqrt{a^2+1}}, -\frac{a}{\sqrt{a^2+1}}\right)\right)$$

Section 2b Derivation of explicit formula for $\mathcal{I}(a,\ b,\ c,\ d) = \int_c^\infty dx\, n(x)\, \mathcal{N}(a\,x+b)\, e^{d\,x}$

I start with the definition

In[84]:=
```
w2b[1] = Int[d[x] Exp[d x] n[x] 𝒩[a x + b], {x, c, ∞}]
```

Out[84]=
```
Int[ⅇ^(d x) d[x] n[x] 𝒩[b + a x], {x, c, ∞}]
```

Change limits of integration

In[85]:=
```
w2b[2] = w2b[1] /. Int[a_, {x, c, ∞}] → Int[a, {x, -∞, ∞}] - Int[a, {x, -∞, c}]
```

Out[85]=
```
-Int[ⅇ^(d x) d[x] n[x] 𝒩[b + a x], {x, -∞, c}] + Int[ⅇ^(d x) d[x] n[x] 𝒩[b + a x], {x, -∞, ∞}]
```

Introduce integral form for $\mathcal{N}$

In[86]:=
```
w2b[3] = w2b[2] /. 𝒩[b_. + a_. x] → Int[n[a x + y], {y, -∞, b}]
```

Out[86]=
```
-Int[ⅇ^(d x) d[x] Int[n[a x + y], {y, -∞, b}] n[x], {x, -∞, c}] +
 Int[ⅇ^(d x) d[x] Int[n[a x + y], {y, -∞, b}] n[x], {x, -∞, ∞}]
```

Rearrange for convenience.

In[87]:=
```
w2b[4] = w2b[3] /. co_. Int[a_. Int[b_, limy_], limx_] → co Int[a b, limx, limy]
```

Out[87]=
```
-Int[ⅇ^(d x) d[x] n[x] n[a x + y], {x, -∞, c}, {y, -∞, b}] +
 Int[ⅇ^(d x) d[x] n[x] n[a x + y], {x, -∞, ∞}, {y, -∞, b}]
```

Make the integrand explicit

In[88]:=
```
w2b[5] = w2b[4] /. n[a_] → 1/Sqrt[2 π] Exp[-a^2/2]
```

Out[88]=
$$-\text{Int}\left[\frac{e^{d\,x-\frac{x^2}{2}-\frac{1}{2}(a\,x+y)^2}\,d[x]}{2\,\pi},\ \{x,\ -\infty,\ c\},\ \{y,\ -\infty,\ b\}\right]+\text{Int}\left[\frac{e^{d\,x-\frac{x^2}{2}-\frac{1}{2}(a\,x+y)^2}\,d[x]}{2\,\pi},\ \{x,\ -\infty,\ \infty\},\ \{y,\ -\infty,\ b\}\right]$$

Note that these integrands have the form of the binormal distribution. I perform a side calculation to identify the parameters of that distribution

In[89]:=
```
aside[1] = fbiv /. {s1 → x, s2 → y}
```

Out[89]=
$$\frac{e^{-\frac{\frac{(x-\mu_1)^2}{\sigma_1^2}+\frac{(y-\mu_2)^2}{\sigma_2^2}-\frac{2\rho\,(x-\mu_1)\,(y-\mu_2)}{\sigma_1\,\sigma_2}}{2\,(1-\rho^2)}}}{2\,\pi\,\sqrt{1-\rho^2}\ \sigma_1\,\sigma_2}$$

I determine the coefficients that match the general form to the specific expression

In[90]:=
```
aside[2] = (w2b[5][[1]] /. a_. Int[b_. Exp[c_], limy_, limx_] → c) -
  (aside[1] /. a_. Exp[b_] → b)
```

Out[90]=
$$d\,x-\frac{x^2}{2}-\frac{1}{2}(a\,x+y)^2+\left(\frac{(x-\mu_1)^2}{\sigma_1^2}+\frac{(y-\mu_2)^2}{\sigma_2^2}-\frac{2\rho\,(x-\mu_1)\,(y-\mu_2)}{\sigma_1\,\sigma_2}\right)\Big/\left(2\left(1-\rho^2\right)\right)$$

Set up equations for the parameters

In[91]:=
```
aside[3] = (# == 0) & /@
  {Coefficient[aside[2], x^2], Coefficient[aside[2], y^2], Coefficient[aside[2], x y],
   (Coefficient[aside[2], x] /. y → 0), (Coefficient[aside[2], y] /. x → 0)}
```

Out[91]=
$$\left\{-\frac{1}{2}-\frac{a^2}{2}+\frac{1}{2\left(1-\rho^2\right)\sigma_1^2}==0,\ -\frac{1}{2}+\frac{1}{2\left(1-\rho^2\right)\sigma_2^2}==0,\ -a-\frac{\rho}{\left(1-\rho^2\right)\sigma_1\,\sigma_2}==0,\right.$$
$$\left.d-\frac{\mu_1}{\left(1-\rho^2\right)\sigma_1^2}+\frac{\rho\,\mu_2}{\left(1-\rho^2\right)\sigma_1\,\sigma_2}==0,\ -\frac{\mu_2}{\left(1-\rho^2\right)\sigma_2^2}+\frac{\rho\,\mu_1}{\left(1-\rho^2\right)\sigma_1\,\sigma_2}==0\right\}$$

In[92]:=
```
aside[4] = Solve[aside[3], {μ1, μ2, σ1, σ2, ρ}] // Simplify
```

Out[92]=
$$\left\{\left\{\mu_1 \to d,\ \mu_2 \to -a\,d,\ \sigma_1 \to -1,\ \sigma_2 \to -\sqrt{1+a^2},\ \rho \to -\frac{a}{\sqrt{1+a^2}}\right\},\right.$$
$$\left\{\mu_1 \to d,\ \mu_2 \to -a\,d,\ \sigma_1 \to -1,\ \sigma_2 \to \sqrt{1+a^2},\ \rho \to \frac{a}{\sqrt{1+a^2}}\right\},$$
$$\left\{\mu_1 \to d,\ \mu_2 \to -a\,d,\ \sigma_1 \to 1,\ \sigma_2 \to \sqrt{1+a^2},\ \rho \to -\frac{a}{\sqrt{1+a^2}}\right\},$$
$$\left.\left\{\mu_1 \to d,\ \mu_2 \to -a\,d,\ \sigma_1 \to 1,\ \sigma_2 \to -\sqrt{1+a^2},\ \rho \to \frac{a}{\sqrt{1+a^2}}\right\}\right\}$$

The financially appropriate branch is the one with both σ_1 and σ_2 positive

In[93]:=
```
aside[5] = aside[4][[3]]
```

Out[93]=
$$\left\{\mu_1 \to d,\ \mu_2 \to -a\,d,\ \sigma_1 \to 1,\ \sigma_2 \to \sqrt{1+a^2},\ \rho \to -\frac{a}{\sqrt{1+a^2}}\right\}$$

The parameters have been matched but I also have to equate the integrands.

In[94]:=
```
aside[6] =
 ((w2b[5][[1]] /. a_ Int[b_, limx_, limy_] → b) /. d[x] → 1) == coeff aside[1]
```

Out[94]=
$$\frac{e^{d\,x-\frac{x^2}{2}-\frac{1}{2}(a\,x+y)^2}}{2\,\pi} == \left(\text{coeff}\ e^{-\frac{\frac{(x-\mu_1)^2}{\sigma_1^2}+\frac{(y-\mu_2)^2}{\sigma_2^2}-\frac{2\rho\,(x-\mu_1)\,(y-\mu_2)}{\sigma_1\,\sigma_2}}{2\,(1-\rho^2)}}\right) \Big/ \left(2\,\pi\,\sqrt{1-\rho^2}\ \sigma_1\,\sigma_2\right)$$

In[95]:=
```
aside[7] = aside[6] /. x → 0 /. y → 0 /. aside[5] // Simplify // PowerExpand
```

Out[95]=
$$e^{\frac{d^2}{2}} == \text{coeff}$$

In[96]:=
```
aside[8] = Solve[aside[7], coeff][[1, 1]]
```

Out[96]=
$$\text{coeff} \to e^{\frac{d^2}{2}}$$

The combined rules that equate the integrands in w2b to the standard bivariate form are

In[97]:=
```
aside["final"] = Join[aside[5], {aside[8]}]
```

Out[97]=
$$\left\{\mu_1 \to d,\ \mu_2 \to -a\,d,\ \sigma_1 \to 1,\ \sigma_2 \to \sqrt{1+a^2},\ \rho \to -\frac{a}{\sqrt{1+a^2}},\ \text{coeff} \to e^{\frac{d^2}{2}}\right\}$$

I showed the steps of this calculation in detail but they could be encapsulated within a callable function is this type of transformation was to be performed repetitively

With the integrands in standard form, I apply a rule for $\mathcal{N}_2$ as defined in equation (17)

In[98]:=
```
w2b[6] = w2b[5] /.
    co_. Int[arg_, {x, -∞, A_}, {y, -∞, B_}] → co coeff 𝒩₂[(A - μ₁)/σ₁, (B - μ₂)/σ₂, ρ] /.
  ρ̄ → √(1 - ρ²) // Simplify[#, {σ₁ > 0, σ₂ > 0}] &
```

Out[98]=
$$\text{coeff}\left(\mathcal{N}_2\left[\infty, \frac{b-\mu_2}{\sigma_2}, \rho\right] - \mathcal{N}_2\left[\frac{c-\mu_1}{\sigma_1}, \frac{b-\mu_2}{\sigma_2}, \rho\right]\right)$$

In[99]:=
```
w2b[7] = w2b[6] /. aside["final"] // Simplify
```

Out[99]=
$$e^{\frac{d^2}{2}}\left(-\mathcal{N}_2\left[c-d, \frac{b+a\,d}{\sqrt{1+a^2}}, -\frac{a}{\sqrt{1+a^2}}\right] + \mathcal{N}_2\left[\infty, \frac{b+a\,d}{\sqrt{1+a^2}}, -\frac{a}{\sqrt{1+a^2}}\right]\right)$$

I perform a second side calculation to find a simpler expression for $\mathcal{N}_2(\infty, A, B)$

In[100]:=
```
aside2[1] =
 Int[PDF[BinormalDistribution[{0, 0}, {1, 1}, ρ], {X, Y}], {X, -∞, A}, {Y, -∞, B}]
```

Out[100]=
$$\text{Int}\left[\frac{e^{-\frac{X^2+Y^2-2XY\rho}{2(1-\rho^2)}}}{2\pi\sqrt{1-\rho^2}}, \{X, -\infty, A\}, \{Y, -\infty, B\}\right]$$

In[101]:=
```
aside2[2] = aside2[1] /. a_. Int[arg_, limx_, limy_] → a Inty[Intx[arg, limx], limy]
```

Out[101]=
$$\text{Inty}\left[\text{Intx}\left[\frac{e^{-\frac{X^2+Y^2-2XY\rho}{2(1-\rho^2)}}}{2\pi\sqrt{1-\rho^2}}, \{X, -\infty, A\}\right], \{Y, -\infty, B\}\right]$$

In[102]:=
```
aside2[3] = Simplify[aside2[2] /. A → ∞ /. Intx → Integrate] // PowerExpand
```

Out[102]=
$$\text{Inty}\left[\frac{e^{-\frac{Y^2}{2}}}{\sqrt{2\pi}}, \{Y, -\infty, B\}\right]$$

In[103]:=
```
aside2[4] = Refine[aside2[3] /. Inty → Integrate] /. ErfRules
```

Out[103]=
$$\mathcal{N}[B]$$

Returning to the main line calculation

In[104]:=
```
w2b[8] = w2b[7] /. 𝒩₂[∞, b_, c_] → 𝒩[b]
```

Out[104]=
$$e^{\frac{d^2}{2}}\left(\mathcal{N}\left[\frac{b+a\,d}{\sqrt{1+a^2}}\right] - \mathcal{N}_2\left[c-d, \frac{b+a\,d}{\sqrt{1+a^2}}, -\frac{a}{\sqrt{1+a^2}}\right]\right)$$

In[105]:=
```
w2b["final"] = w2b[8]
```

Out[105]=
$$e^{\frac{d^2}{2}}\left(\mathcal{N}\left[\frac{b+a\,d}{\sqrt{1+a^2}}\right]-\mathcal{N}_2\left[c-d,\ \frac{b+a\,d}{\sqrt{1+a^2}},\ -\frac{a}{\sqrt{1+a^2}}\right]\right)$$

which is the result in equation (16).

In[106]:=
```
ℐ[a, b, c, d] == w2b["final"] // TraditionalForm
```

Out[106]//TraditionalForm=
$$\mathcal{I}(a,\,b,\,c,\,d)=e^{\frac{d^2}{2}}\left(\mathcal{N}\left(\frac{a\,d+b}{\sqrt{a^2+1}}\right)-\mathcal{N}_2\left(c-d,\ \frac{a\,d+b}{\sqrt{a^2+1}},\ -\frac{a}{\sqrt{a^2+1}}\right)\right)$$

Coded form for the analytic method *WorstOfCallAnalytic*

In[107]:=
```
Clear[WorstOfCallAnalytic];
WorstOfCallAnalytic[K_, T_, S10_, qS1_, σS1_, S20_, qS2_, σS2_, ρ_, r_] :=
 Module[{V, k, μ1, σ1, μ2, σ2},
  μ1 = Log[S10] + (r - qS1 - σS1^2/2) T;
  μ2 = Log[S20] + (r - qS2 - σS2^2/2) T;
  σ1 = σS1 √T;
  σ2 = σS2 √T;
  k = Log[K];
  V = Exp[-r T]
    (ℐNumeric[1, k, μ1, μ2, σ1, σ2, ρ] - K ℐNumeric[0, k, μ1, μ2, σ1, σ2, ρ]) ]
```

Coded form - $\mathcal{J}(\alpha)$

$$\mathcal{J}(\alpha)=e^{\alpha\,\mu_2}\,\mathcal{I}\left(\frac{\rho\,\sigma_1-\sigma_2}{\sigma_1\,\rho_b},\ \frac{\mu_1-\mu_2}{\sigma_1\,\rho_b},\ \frac{k-\mu_2}{\sigma_2},\ \alpha\,\sigma_2\right)+e^{\alpha\,\mu_1}\,\mathcal{I}\left(\frac{\rho\,\sigma_2-\sigma_1}{\sigma_2\,\rho_b},\ \frac{\mu_2-\mu_1}{\sigma_2\,\rho_b},\ \frac{k-\mu_1}{\sigma_1},\ \alpha\,\sigma_1\right)$$

In[109]:=
```
w2["final"] /. {μ1 → μ1, μ2 → μ2, σ1 → σ1, σ2 → σ2, ρb → ρb}
```

Out[109]=
$$e^{\alpha\,\mu2}\,\mathcal{I}\left[\frac{\rho\,\sigma1-\sigma2}{\rho b\,\sigma1},\ \frac{\mu1-\mu2}{\rho b\,\sigma1},\ \frac{k-\mu2}{\sigma2},\ \alpha\,\sigma2\right]+e^{\alpha\,\mu1}\,\mathcal{I}\left[\frac{-\sigma1+\rho\,\sigma2}{\rho b\,\sigma2},\ \frac{-\mu1+\mu2}{\rho b\,\sigma2},\ \frac{k-\mu1}{\sigma1},\ \alpha\,\sigma1\right]$$

In[110]:=
```
Clear[ℐ];
ℐ[α_, k_, μ1_, μ2_, σ1_, σ2_, ρ_] :=
 Module[{ρb = √(1 - ρ^2)},
  e^(α μ2) ℐ[(ρ σ1 - σ2)/(ρb σ1), (μ1 - μ2)/(ρb σ1), (k - μ2)/σ2, α σ2] + e^(α μ1) ℐ[(-σ1 + ρ σ2)/(ρb σ2), (-μ1 + μ2)/(ρb σ2), (k - μ1)/σ1, α σ1]]
```

In[112]:=
```
Clear[ℐNumeric];
ℐNumeric[α_, k_, μ1_, μ2_, σ1_, σ2_, ρ_] :=
 Module[{ρb = √(1 - ρ^2)},
  e^(α μ2) ℐNumeric[(ρ σ1 - σ2)/(ρb σ1), (μ1 - μ2)/(ρb σ1), (k - μ2)/σ2, α σ2] +
   e^(α μ1) ℐNumeric[(-σ1 + ρ σ2)/(ρb σ2), (-μ1 + μ2)/(ρb σ2), (k - μ1)/σ1, α σ1]]
```

Coded form - $\mathcal{I}$(a, b, c, d)

In[114]:=
```
w2b["final"] /. 𝒩₂ → 𝒩2
```

Out[114]=
$$e^{\frac{d^2}{2}}\left(\mathcal{N}\left[\frac{b+a\,d}{\sqrt{1+a^2}}\right]-\mathcal{N}2\left[c-d,\frac{b+a\,d}{\sqrt{1+a^2}},-\frac{a}{\sqrt{1+a^2}}\right]\right)$$

In[115]:=
```
Clear[ℐAnalytic];
ℐAnalytic[a_, b_, c_, d_] := e^(d^2/2) (𝒩[(b + a d)/√(1 + a^2)] - 𝒩2[c - d, (b + a d)/√(1 + a^2), -a/√(1 + a^2)])
```

I can either numerically implement $\mathcal{I}$ directly in terms of its integral definition (a 1-D integral)

In[117]:=
```
Clear[ℐNumericDirect];
ℐNumericDirect[a_, b_, c_, d_] :=
 NIntegrate[n[x] 𝒩[a x + b] Exp[d x], {x, c, ∞}]
```

or in terms of the closed form normal distribution resulting from its analytic valuation.

In[119]:=
```
Clear[ℐNumeric];
ℐNumeric[a_, b_, c_, d_] :=
  e^(d^2/2) (𝒩[(b + a d)/√(1 + a^2)] - 𝒩2DoubleQuadrature[c - d, (b + a d)/√(1 + a^2), -a/√(1 + a^2)])
```

but here I am evaluating $\mathcal{N}_2$ using double quadrature. Below, I also implement the Drenzer Wesolosky approximate form for $\mathcal{N}_2$.

Tests would be required to decide which of these alternate numerical forms is the most computationally efficient in different situations. I worked as a model risk quant, focusing on model soundness, accuracy and whether the model was appropriate for its intended use. Issues of computational efficiency were seldom of concern to me. However, efficiency was a serious issue for those quants tasked with implementing models into production level trading and risk management systems where they might be called thousands of times a day.

Utility functions

In[121]:=
```
Clear[n];
n[ϵ_?NumericQ] := 1/Sqrt[2 π] Exp[-ϵ^2/2]
```

In[27]:=
```
(* Cumulative Standard Normal Distribution *)
Clear[N];
N[z_?NumericQ] := N[1/2 + 1/2 Erf[z/Sqrt[2]]];
```

Note on Mathematica practice: The use of _?NumericQ to constrain the function argument is a useful trick. If the argument is , the pattern will match and the function will evaluate to a numerical value, e.g., $n[0.5] = 0.3520653267642995$. However if the argument is a symbol or otherwise not “Numeric”, the function will not evaluate and simply return the calling expression.

In[29]:=
```
Clear[n2];
n2[ϵ1_, ϵ2_, ρ_] := E^((ϵ1^2+ϵ2^2-2 ϵ1 ϵ2 ρ)/(2 (-1+ρ^2)))/(2 π Sqrt[1-ρ^2])
```

In[31]:=
```
Clear[N2DoubleQuadrature];
N2DoubleQuadrature[a_, b_, ρ_] :=
 NIntegrate[n2[ϵ1, ϵ2, ρ], {ϵ1, -4, a}, {ϵ2, -4, b}]
```

In[33]:=
```
Clear[N2DrenzerWesolosky];
N2DrenzerWesolosky[a_, b_, ρ_] :=
 Module[{x, y},
  x1 = 0.018854042; y1 = 0.04691008;
  x2 = 0.038088059; y2 = 0.23076534;
  x3 = 0.045270739; y3 = 0.5;
  x4 = x2;          y4 = 0.76923466;
  x5 = x1;          y5 = 0.95308992;
  N[a] N[b] + ρ Sum[xj/Sqrt[1 - yj^2 ρ^2] Exp[(2 a b yj - a^2 - b^2)/(2 (1 - yj^2 ρ^2))], {j, 5}]]
```

Implementation of a plain vanilla European call (Black-Scholes)

In[35]:=
```
Clear[EuroCall];
EuroCall[S_, K_, r_, q_, σ_, τ_] :=
Module[{d1},
 If[τ <= 0.00001, Return[Max[S - K, 0.0]]];
 d1 = (Log[S/K] + (r - q + σ^2/2) τ)/(σ Sqrt[τ]);
 S Exp[-q τ] N[d1] - K Exp[-r τ] N[d1 - σ Sqrt[τ]]];
```