Spread Option with SABR and Copula

```
<< Graphics`Legend`
```

```
<< "C:\\Documents and Settings\\ocroissant\\My Documents\\NumericalIntegration.m"
```

```
<< "C:\\Documents and Settings\\ocroissant\\My
   Documents\\SpreadOptionLN3\\TriSpreadOption7.m"
```

$$\text{Nd}[x\_] := N\left[\frac{\text{Erf}\left[\frac{x}{\sqrt{2}}\right]+1}{2}, 20\right]$$

$$\text{NormalDis}[x\_] := \frac{\text{Erf}\left[\frac{x}{\sqrt{2}}\right]+1}{2}$$

```
NormDens[x_] := Exp[-x^2 / 2.] / Sqrt[2 Pi]
```

$$\text{BS}[f\_, k\_, t\_, v\_] := f\ \text{NormalDis}\left[\frac{\text{Log}\left[\frac{f}{k}\right]+\frac{v^2 t}{2}}{v\ \text{Sqrt}[t]}\right] - k\ \text{NormalDis}\left[\frac{\text{Log}\left[\frac{f}{k}\right]-\frac{v^2 t}{2}}{v\ \text{Sqrt}[t]}\right]$$

```
PhiVol[m_, u_] := Exp[m] Nd[m / u + u / 2] - Nd[m / u - u / 2]
```

```
ImpVolBS[F_, K_, t_, discount_, opt_] :=
 Module[{o = opt / (K discount), m = Log[F / K], ss},
  ss = FindRoot[PhiVol[m, u] == o, {u, 0.2, 0.0001, 15.},
    AccuracyGoal → 8, WorkingPrecision → 30, MaxIterations → 200];
  ss〚1, 2〛 / Sqrt[t]]
```

```
CallNewOption[f_, K_, T_, α_, β_, ρ_, ν_] := Re[
  Module[{z, x, b1, θ, ς, κ, result, σ, a, b},
```

$$z = \frac{f^{1-\beta} - K^{1-\beta}}{\alpha\,(1-\beta)};$$

$$x = \frac{\mathrm{Log}\left[\frac{-\rho+\nu\,z+\sqrt{1-2\,\nu\,\rho\,z+\nu^2\,z^2}}{1-\rho}\right]}{\nu};$$

$$a = \sqrt{-\frac{1}{8}\left(\alpha^2\,(-2+\beta)\,\beta\,K^{2\beta-2} + 6\,\alpha\,\beta\,K^{\beta-1}\,\nu\,\rho + \nu^2\left(2-3\,\rho^2\right)\right)};$$

$$\mathrm{result} = \mathrm{Max}[f-K, 0] + \frac{(f-K)}{2\sqrt{2\pi}}\;\frac{e^{\frac{1}{4}\,\alpha\,\beta\,K^{\beta-1}\,\nu\,\rho\,z^2+\mathrm{Log}\left[\frac{\alpha\,(f\,K)^{\beta/2}}{f-K}\,x\,\left(1-2\,\nu\,\rho\,z+\nu^2\,z^2\right)^{1/4}\right]}}{x}\left(\frac{1}{2\,a}\left(e^{-\sqrt{2}\,a\,\mathrm{Abs}[x]}\,\sqrt{\pi}\left(1-e^{2\sqrt{2}\,a\,\mathrm{Abs}[x]} + e^{2\sqrt{2}\,a\,\mathrm{Abs}[x]}\,\mathrm{Erf}\left[\frac{\mathrm{Abs}[x]}{\sqrt{2}\,\sqrt{T}}+a\,\sqrt{T}\right] - \mathrm{Erf}\left[\frac{\sqrt{2}\,\mathrm{Abs}[x]-2\,a\,T}{2\,\sqrt{T}}\right]\right)\right)\right)\Big]\Big];$$

```
PutNewOption[f_, K_, T_, α_, β_, ρ_, ν_] := K - f + CallNewOption[f, K, T, α, β, ρ, ν]
```

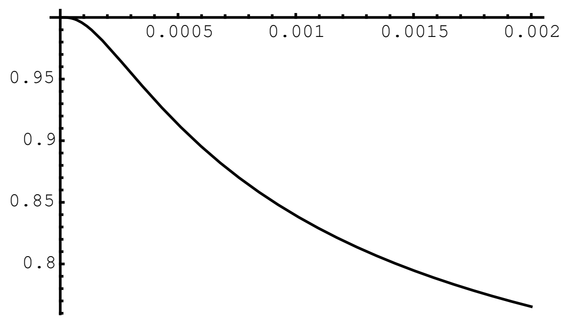
```
ImpVolSABR[f_, K_, T_, α_, β_, ρ_, ν_] :=
 ImpVolBS[f, K, T, 1.0, CallNewOption[f, K, T, α, β, ρ, ν]]
```

```
DigitalNewOption[f_, K_, T_, α_, β_, ρ_, ν_] :=
 - (CallNewOption[f, K + 0.00001, T, α, β, ρ, ν] -
     CallNewOption[f, K - 0.00001, T, α, β, ρ, ν]) / 0.00002
```

```
CallNewOption[0.05, 0.103, 2, 0.11, 0.7, -0.5, 0.2]
```

0.000071312

```
Plot[DigitalNewOption[0.05, K, 20, 0.11, 0.7, -0.5, 0.2],
 {K, 0.00001, 0.002}, PlotRange → All]
```



- Graphics -

```
CallNewOption[0.05, 0.050000001, 2, 0.11, 0.7, -0.5, 0.2]
```

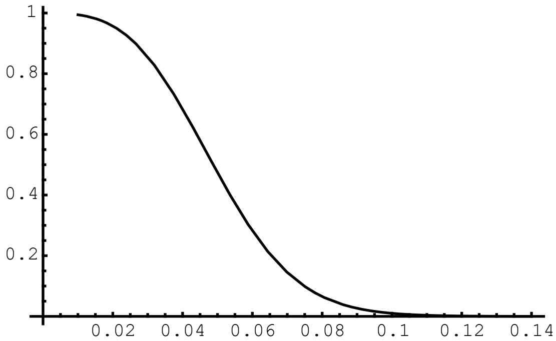
0.00754075

```
CallNewOption[0.05, 0.050000001, 2, 0.11, 0.7, -0.5, 0.2]
```

0.00754075

```
InverseSABRDist[f_, x_, tex_, alpha_, beta_, rho_, nu_] :=
 Module[{ss, K0}, ss = FindRoot[
    DigitalNewOption[f, K0, tex, alpha, beta, rho, nu] == x, {K0, f 0.0001, f 10.}];
  K0 /. ss]
```

```
Plot[N[DigitalNewOption[0.05, x, 2., 0.11, 0.7, -0.5, 0.2]], {x, 0.01, 0.14}]
```

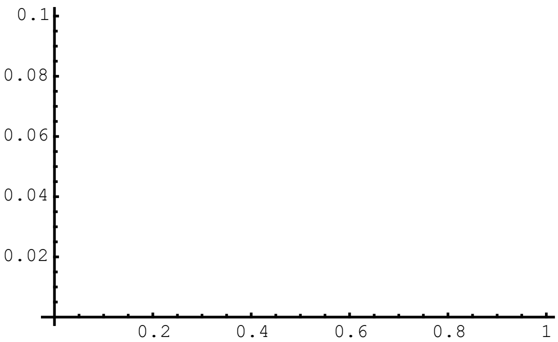


- Graphics -

```
InverseSABRDist[0.05, 0.1, 2., 0.11, 0.7, -0.5, 0.2]
```

0.0748457

```
ListPlot[Map[({#, InverseSABRDist[0.05, #, 2., 0.11, 0.7, -0.5, 0.2]}) &,
   Table[i / 100, {i, 1, 99}]]];
```




```
InverseSABRDist[0.05, 0.9999, 20., 0.11, 0.7, -0.5, 0.2]
```

FindRoot::cvmit : Failed to converge to the requested accuracy or precision within 100 iterations.
More…

23.6387

```
{N[InverseSABRDist[100, 0.99999999999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.9999999999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.999999999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.99999999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.9999999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.999999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.99999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.9999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.999, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.99, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.9, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.55, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.5, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.45, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.1, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.01, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.0001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.00001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.000001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.0000001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.00000001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.000000001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20],
 N[InverseSABRDist[100, 0.0000000001, 1, 15 / 100, 7 / 10, 5 / 10, 5 / 10], 20]}
```

```
{62.2291, 63.8316, 65.9727, 70.3943, 74.8658, 78.9712, 82.6947, 86.0751,
 89.1867, 92.1815, 95.5151, 99.0788, 99.505, 99.9462, 105.059, 112.485,
 121.075, 131.531, 144.49, 160.742, 181.365, 208.134, 238.966, 307.074}
```

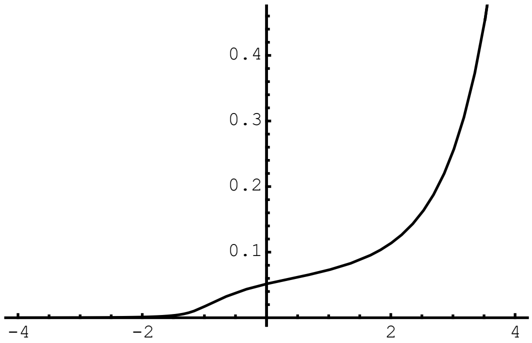
```
FromGaussTODistribution[f_, x_, tex_, alpha_, beta_, rho_, nu_] :=
 InverseSABRDist[f, Nd[-x], tex, alpha, beta, rho, nu]
```

```
FromGaussTODistribution[f_, x_, tex_, alpha_, beta_, rho_, nu_] :=
 Module[{y = Nd[-x]}, InverseSABRDist[f, y, tex, alpha, beta, rho, nu]]
```

```
FromGaussTODistribution[.0505, 4, 2, 0.15, 0.7, 0.5, -0.5]
```

```
0.346414
```

```
Plot[N[FromGaussTODistribution[0.05, x, 5, 0.101, 7 / 10, -5 / 10, 5 / 10], 20],
 {x, -4., +4.}]
```

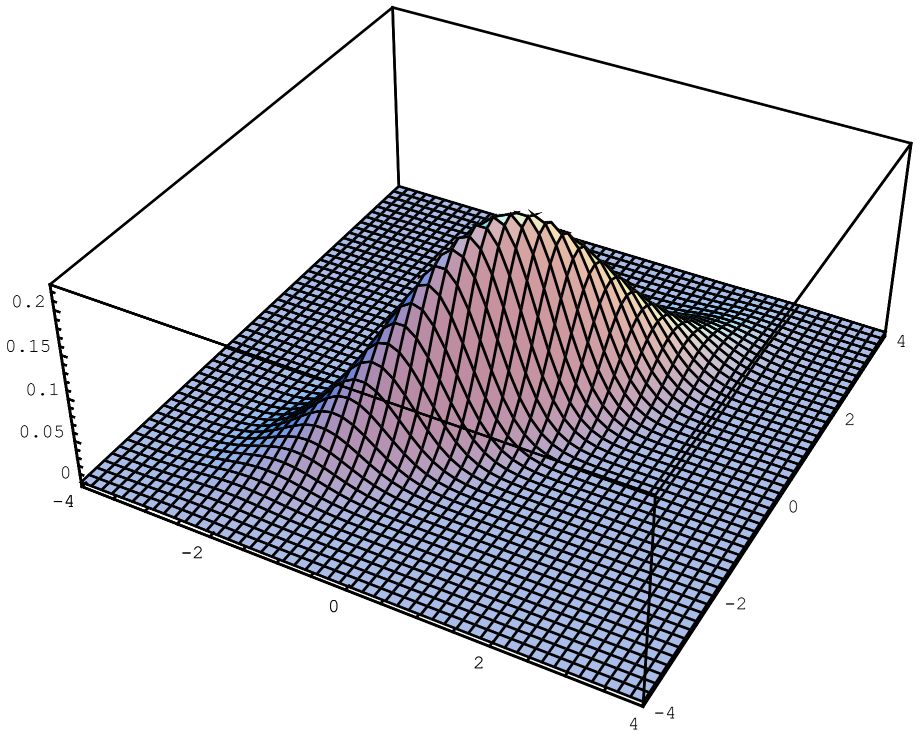


- Graphics -

```
(* Gaussian Copula with correlation = rho *)
```

```
Kernel[x_, y_, rho_] :=
 Exp[-x^2 / 2] Exp[- (y - x) ^2 / (2 × (1 - rho^2))] / (2 Pi Sqrt[1 - rho^2])
```

```
Plot3D[Kernel[x, y, 0.7], {x, -4, 4}, {y, -4, 4}, PlotRange → All, PlotPoints → 50]
```



- SurfaceGraphics -

NormalBivariateLimitInverse[a_, x_, f_, sigma_, t_, corr_] :=

$$\frac{\text{sigma}^2\, t - 2\, \text{corr0}\, \text{sigma}\, \sqrt{t}\, x + 2\, \text{Log}\left[\frac{a}{f}\right]}{2\, \sqrt{1 - \text{corr}^2}\, \text{sigma}\, \sqrt{t}}$$

```
SpreadOption[f1_, alpha1_, beta1_, rho1_, nu1_,
  f2_, alpha2_, beta2_, rho2_, nu2_, corr_, k_, tex_, n_] :=
 Module[{c = LegendreCoeffs[n], cy, i, j, ay, by, x, y, sum = 0, X, Y, prob},
  ay = -4;
  by = 4;
  coefs = BetweenCstCoeffs3X[c, ay, by];
  If[Abs[corr] ≤ 0.5,
   sum = CoeffBasedIntegrate[
      (Module[{X, Y, prob},
         X = FromGaussTODistribution[f1, #1, tex, alpha1, beta1, rho1, nu1];
         Y = FromGaussTODistribution[f2, #2, tex, alpha2, beta2, rho2, nu2];
         prob = Exp[-((#1)^2 + (#2)^2 - 2 corr #1 #2) / (2 × (1 - corr^2) tex)];
         Max[(X - Y - k), 0] prob]) &,
      coefs, coefs] / Sqrt[1 - corr^2],
   sum = CoeffBasedIntegrate[
     (Module[{X, Y, prob},
        X = FromGaussTODistribution[f1,
          (#1 - #2 Sqrt[1 - corr^2]) / corr, tex, alpha1, beta1, rho1, nu1];
        Y = FromGaussTODistribution[f2, #2, tex, alpha2, beta2, rho2, nu2];
        prob = Exp[-((#1)^2 + (#2)^2) / (2 tex)];
        Max[(X - Y - k), 0] prob]) &,
     coefs, coefs]
  ];
  sum / (2 Pi tex)]
```

```
General::spell1 : Possible spelling error: new symbol name "coefs" is similar to existing symbol
   "coeffs". More…
```

```
SpreadOption[func1_, func2_, corr_, k_, tex_, leglst_] :=
 Module[{cy, i, j, ay, by, x, y, sum = 0, X, Y, prob},
  ay = -4;
  by = 4;
  coefs = BetweenCstCoeffs3X[leglst, ay, by];
  sum = CoeffBasedIntegrate[
    (Module[{X, Y, prob},
       X = func1[#1 Sqrt[1 - corr^2] + #2 corr];
       Y = func2[#2];
       prob = Exp[-((#1)^2 + (#2)^2) / (2 tex)];
       Max[(X - Y - k), 0] prob]) &,
    coefs, coefs];
  sum / (2 Pi tex)]
```

```
SpreadOption[func1_, func2_, sig1_, sig2_, corr_, k_, tex_, leglst_] :=
 Module[{cy, i, j, ay, by, x, y, sum = 0, X, Y, prob},
  ay = -6;
  by = 6;
  coefs = BetweenCstCoeffs3X[leglst, ay, by];
  sum = CoeffBasedIntegrate[
    (Module[{X, Y, prob},
       X = FromGaussToDistributionInterpolator[
         func1, sig1, #1 Sqrt[1 - corr^2] + #2 corr];
       Y = FromGaussToDistributionInterpolator[func2, sig2, #2];
       prob = Exp[-((#1)^2 + (#2)^2) / (2 tex)];
       Max[(X - Y - k), 0] prob]) &,
    coefs, coefs];
  sum / (2 Pi tex )]
```

```
SpreadOptionExact[func1_, func2_, corr_, k_, tex_, xexten_] :=
 Module[{cy, i, j, ay, by, x, y, sum = 0, X, Y, prob},
  ay = -xexten;
  by = xexten;
  If[Abs[corr] ≤ 0.01,
   sum = NIntegrate[
      (Module[{X, Y, prob},
        X = func1[x];
        Y = func2[y];
        prob = Exp[-((x)^2 + (y)^2 - 2 corr x y) / (2 × (1 - corr^2) tex)];
        Max[(X - Y - k), 0] prob]),
      {x, ay, by}, {y, ay, by}] / Sqrt[1 - corr^2],
   sum = NIntegrate[
     (Module[{X, Y, prob},
       X = func1[x Sqrt[1 - corr^2] + y corr];
       Y = func2[y];
       prob = Exp[-((x)^2 + (y)^2) / (2 tex)];
       Max[(X - Y - k), 0] prob]),
     {x, ay, by}, {y, ay, by}]
  ];
  sum / (2 Pi tex )]
```

```
FromGaussToDistributionInterpolator[interpol_, sigma_, x_] :=
 Module[{x0 = interpol[[1, 1, 1]], x1 = interpol[[1, 1, 2]], y0, y1},
  If[x ≤ x0, interpol[x0] Exp[sigma (x - x0)],
   If[x ≥ x1, interpol[x1] Exp[sigma (x - x1)], interpol[x]]]]
```

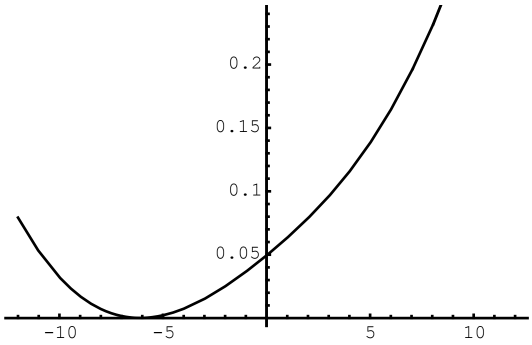
```
FromGaussToDistributionInterpolator[interpol_, sigma1_, sigma2_, x_] :=
 Module[{x0 = interpol[[1, 1, 1]], x1 = interpol[[1, 1, 2]], y0, y1},
  If[x ≤ x0, interpol[x0] Exp[sigma1 (x - x0)],
   If[x ≥ x1, interpol[x1] Exp[sigma2 (x - x1)], interpol[x]]]]
```

```
FromGaussToDistributionInterpolator2[interpol_, x_] :=
 Module[{x0 = interpol[[1, 1, 1]], x1 = interpol[[1, 1, 2]], y0, y1},
  deriv = (interpol[x0] - interpol[x1]) / (x0 - x1);
  If[x ≤ x0, interpol[x0] Exp[deriv / interpol[x0] (x - x0)],
   If[x ≥ x1, interpol[x1] Exp[deriv / interpol[x1] (x - x1)], interpol[x]]]]
```

```
FromGaussToDistributionInterpolator3[interpol_, x_] :=
 Module[{x0 = interpol[[1, 1, 1]], x1 = interpol[[1, 1, 2]], y0, y1},
  deriv0 = (interpol[x0] - interpol[x0 + (x1 - x0) / 10]) / ((x0 - x1) / 10);
  deriv1 = (interpol[x1] - interpol[x1 - (x1 - x0) / 10]) / ((x1 - x0) / 10);
  If[x ≤ x0, interpol[x0] Exp[deriv0 / interpol[x0] (x - x0)],
   If[x ≥ x1, interpol[x1] Exp[deriv1 / interpol[x1] (x - x1)], interpol[x]]]]
```

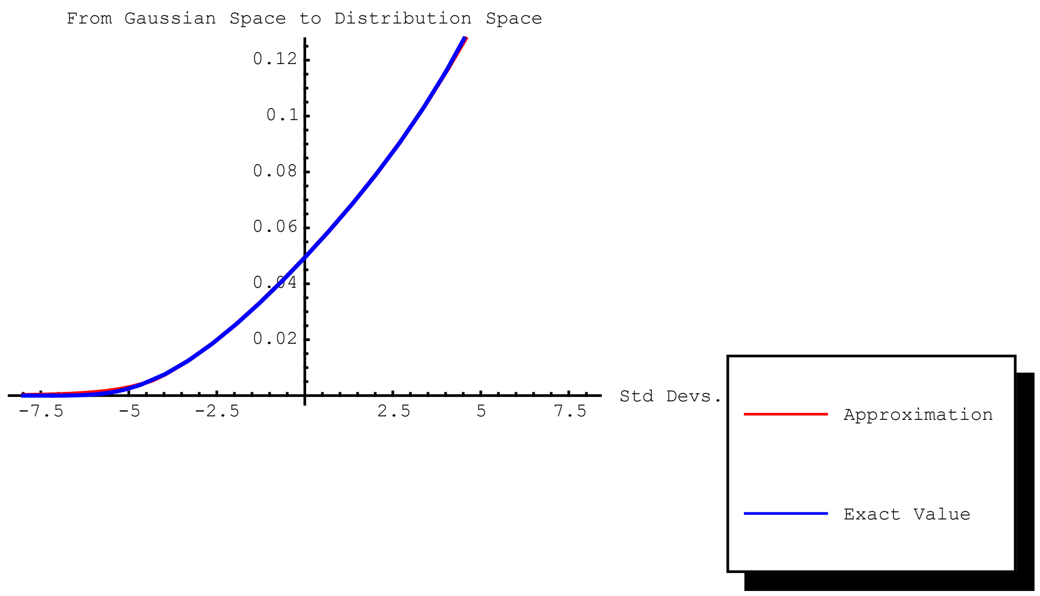
```
FromGaussToDistributionInterpolator4[interpol_, x_] :=
 Module[{x0 = interpol[[1, 1, 1]], x1 = interpol[[1, 1, 2]], y0, y1},
  y0 = interpol[x0] ; y1 = interpol[x1];
  derivFirst0 = (y0 - interpol[x0 + (x1 - x0) / 10]) / ((x0 - x1) / 10);
  derivSecond0 = (y0 + interpol[x0 + 2 (x1 - x0) / 10] - 2 interpol[x0 + (x1 - x0) / 10]) /
    ((x0 - x1) / 10)^2;
  derivFirst1 = (y1 - interpol[x1 - (x1 - x0) / 10]) / ((x1 - x0) / 10);
  derivSecond1 = (y1 + interpol[x1 - 2 (x1 - x0) / 10] - 2 interpol[x1 - (x1 - x0) / 10]) /
    ((x1 - x0) / 10)^2;
  If[x ≤ x0, y0 Exp[derivFirst0 / y0 (x - x0) +
      (derivSecond0 - derivFirst0^2 / y0) (x - x0)^2 / 2],
   If[x ≥ x1, y1 Exp[derivFirst1 / y1 (x - x1) +
       (derivSecond1 - derivFirst1^2 / y1) (x - x1)^2 / 2], interpol[x]]]]
```

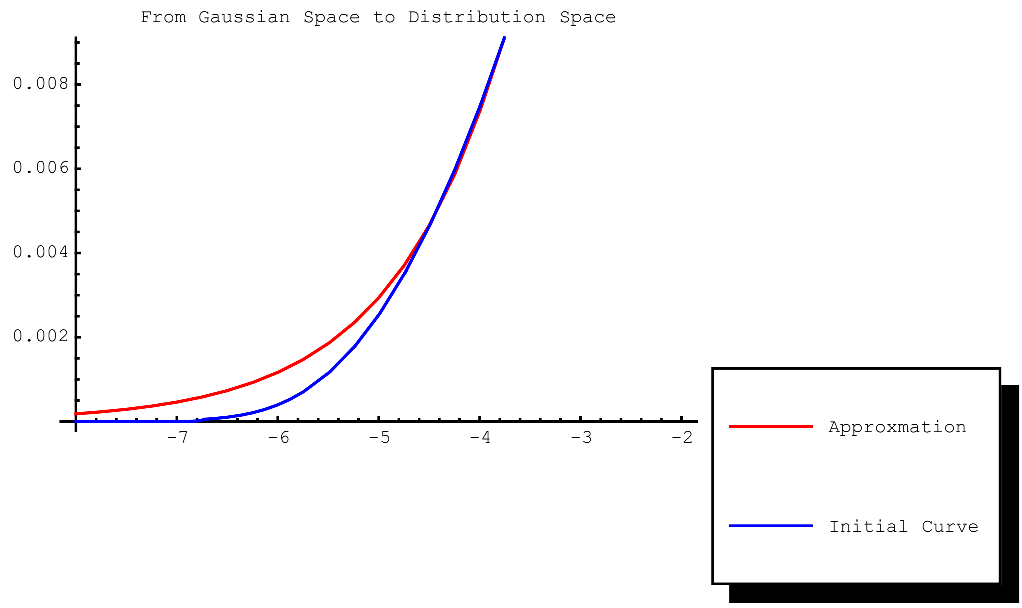
```
ff1 = Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7,
    rho1 = -0.5, nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
    nu2 = 0.18, corr = 0.4999, k = 0.01, tex = 1, n = 25, tab, xlen},
   xlen = 5;
   Func1 = Interpolation[
     Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
      Table[xlen i / (n + 1) - xlen / 2., {i, 1, n}]]]];
Plot[
 ff1[
  x],
 {x,
  -12,
  12}]
```




```
- Graphics -
```

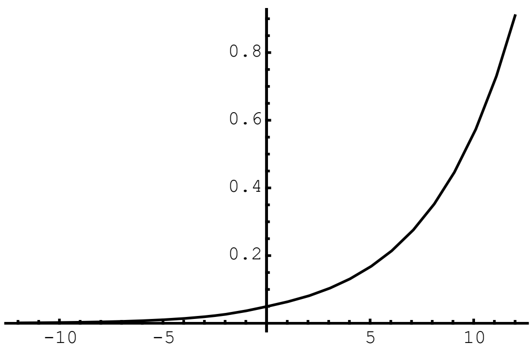
```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
  nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
  nu2 = 0.18, corr = 0.4999, k = 0.01, tex = 1, n, tab, xlen, Func},
 n = 50;
 xlen = 8;
 Func = Interpolation[
   Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
    Table[xlen i / (n + 1) - xlen / 2., {i, 1, n}]]];
 xlen2 = 14;
 TrueFunc = Interpolation[
   Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
    Table[xlen2 i / (n + 1) - xlen2 / 2., {i, 1, n}]]];
 Plot[{FromGaussToDistributionInterpolator4[Func, x],
   FromGaussToDistributionInterpolator3[TrueFunc, x]}, {x, -8, 8},
  PlotLabel → StringJoin["From Gaussian Space to Distribution Space"],
  PlotStyle → {{Thickness[0.007], RGBColor[1, 0, 0]},
    {Thickness[0.007], RGBColor[0, 0, 1]}, {Thickness[0.005], RGBColor[0, 1, 0]}},
  PlotLegend → {"Approximation", "Exact Value"}, LegendPosition → {1, -1},
  AxesLabel → {"Std Devs.", ""}];
 Plot[{FromGaussToDistributionInterpolator4[Func, x],
   FromGaussToDistributionInterpolator3[TrueFunc, x]}, {x, -8, -2},
  PlotLabel → StringJoin["From Gaussian Space to Distribution Space"],
  PlotStyle → {{Thickness[0.005], RGBColor[1, 0, 0]},
    {Thickness[0.005], RGBColor[0, 0, 1]}, {Thickness[0.005], RGBColor[0, 0, 1]}},
  PlotLegend → {"Approxmation", "Initial Curve"}, LegendPosition → {1, -1}];
]
```






```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
   nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
   nu2 = 0.18, corr = 0.4999, k = 0.01, tex = 1, n = 25, tab, xlen},
  xlen = 3;

  Func1 = Interpolation[
    Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
     Table[xlen i / (n + 1) - xlen / 2., {i, 1, n}]]];
  Print[Func1[Func1⟦1, 1, 1⟧], "", Func1[Func1⟦1, 1, 2⟧]];
  sigma1 = ImpVolSABR[f1, Func1[Func1⟦1, 1, 1⟧], tex, alpha1, beta1, rho1, nu1];
  sigma2 = ImpVolSABR[f1, Func1[Func1⟦1, 1, 2⟧], tex, alpha1, beta1, rho1, nu1];
  Plot[FromGaussToDistributionInterpolator[Func1, sigma1, sigma2, x], {x, -12, 12}]]
```

0.03194760.0691192



- Graphics -

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
   nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
   nu2 = 0.18, k = 0.01, tex = 1, nlegen, ninteg, Func1, Func2},
  nlegen = 35;
  ninteg = 35;
  xlen = 10;
  legenlst = LegendreCoeffs[nlegen];
  Func1 = Interpolation[
    Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
     Table[xlen ( i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  Func2 = Interpolation[Map[
     ({#, FromGaussTODistribution[f1, #, tex, alpha2, beta2, rho2, nu2]}) &,
     Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  {SpreadOption[Func1, Func2, 0.00, k, tex, legenlst] ,
   SpreadOption[Func1, Func2, 0.0100001, k, tex, legenlst] }]
```

```
{0.00363412, 0.00360145}
```

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
   nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, nu2 = 0.18,
   k = 0.0001, tex = 1, nlegen, ninteg, Func1, Func2, sig1, sig2},
  nlegen = 35;
  ninteg = 35;
  xlen = 3;
  sig1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
  sig2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
  legenlst = LegendreCoeffs[nlegen];
  Func1 = Interpolation[
    Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
     Table[xlen ( i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  Func2 = Interpolation[Map[
     ({#, FromGaussTODistribution[f1, #, tex, alpha2, beta2, rho2, nu2]}) &,
     Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  SpreadOption[Func1, Func2, sig1, sig2, 0.8, -0.02, tex, legenlst] ]
```

```
0.0200417
```

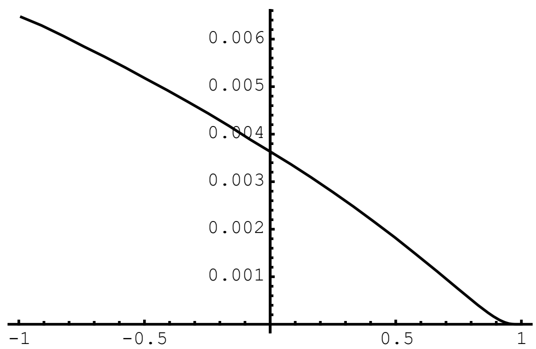
```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
   nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
   nu2 = 0.18, k, tex = 1, nlegen, ninteg, Func1, Func2, sig1, sig2},
  nlegen = 35;
  ninteg = 35;
  xlen = 5;
  sig1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
  sig2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
  legenlst = LegendreCoeffs[nlegen];
  Func1 = Interpolation[
    Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
     Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  Func2 = Interpolation[Map[
     ({#, FromGaussTODistribution[f1, #, tex, alpha2, beta2, rho2, nu2]}) &,
     Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  SpreadOption[Func1, Func2, sig1, sig2, 0.8, -0.03, tex, legenlst] ]
```

```
0.0300008
```

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
   nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
   nu2 = 0.18, corr = 0.4999, k = 0.01, tex = 1, ninteg, Func1, Func2},
  ninteg = 35;
  Func1 = Interpolation[
    Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
     Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  Func2 = Interpolation[Map[
     ({#, FromGaussTODistribution[f1, #, tex, alpha2, beta2, rho2, nu2]}) &,
     Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
  {SpreadOptionExact[Func1, Func2, 0.0099999, k, tex, 5] ,
   SpreadOptionExact[Func1, Func2, 0.0100001, k, tex, 5] }]
```

```
{0.00362078, 0.00362077}
```

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5,
  nu1 = 0.2, f2 = 0.05, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5,
  nu2 = 0.18, k = 0.01, tex = 1, nlegen, ninteg, Func1, Func2, xlen},
 nlegen = 35;
 ninteg = 35;
 xlen = 7;
 legenlst = LegendreCoeffs[nlegen];
 Func1 = Interpolation[
   Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
    Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 Func2 = Interpolation[Map[
    ({#, FromGaussTODistribution[f1, #, tex, alpha2, beta2, rho2, nu2]}) &,
    Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 Plot[SpreadOption[Func1, Func2, cc, k, tex, legenlst], {cc, -0.99, +0.99}] ]
```



- Graphics -

```
NormalOption[S_, k_, v_] := Module[{
   A = S - k,
   Σ = v },
  A NormalDis[A / Σ] + Σ Exp[-A^2 / (2 Σ^2)] / Sqrt[2 Pi]
 ]
```

```
NormalCallOption[f_, sigma_, k_, t_] := NormalOption[f, k, sigma Sqrt[t]]
```

```
NormalSpreadOption[S1_, S2_, k_, σ1_, σ2_, ρ_] :=
 NormalOption[S1 - S2, k, Sqrt[σ1^2 + σ2^2 - 2 ρ σ1 σ2]]
```

```
Off[InterpolatingFunction::"dmval", NIntegrate::"slwcon",
 NIntegrate::"ncvb", FindRoot::"precw", FindRoot::"nlnum"]
```

```
NormalCallOptionImplicitVol[f_, optionprice_, k_, t_] := Module[{ss, u},
  ss = FindRoot[NormalCallOption[f, u, k, t] == optionprice, {u, 0.2, 0.00001, 15.},
    AccuracyGoal → 8, WorkingPrecision → 30, MaxIterations → 200];
  ss[[1, 2]]]
```

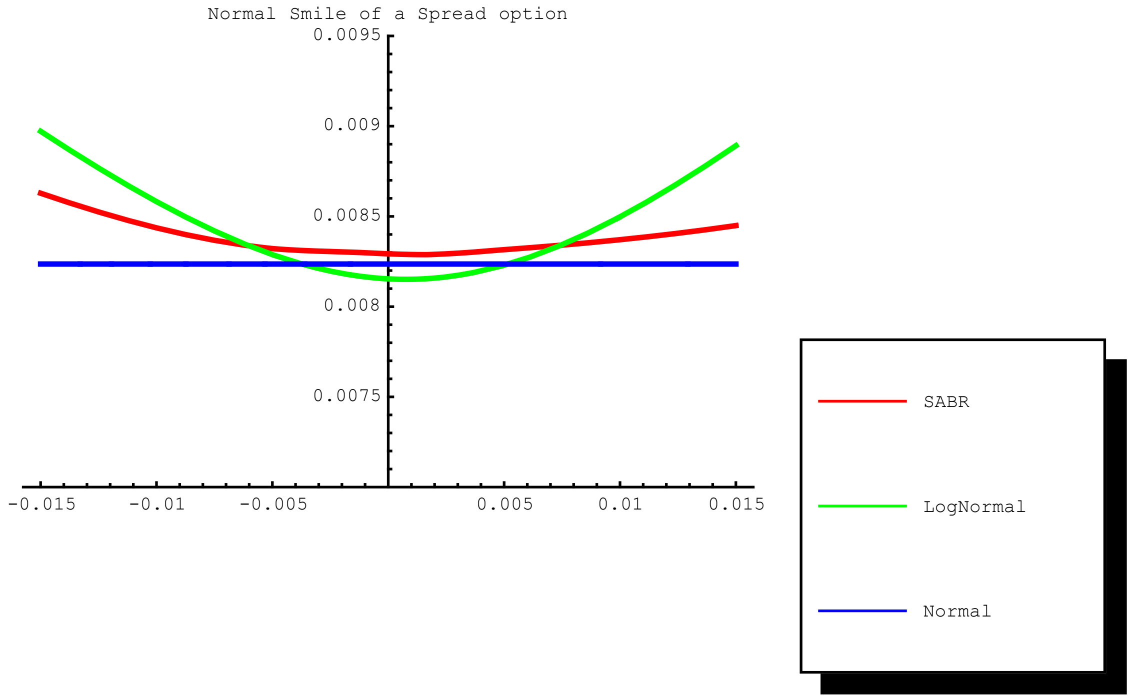
```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5, nu1 = 0.2, f2 = 0.045,
  alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, corr = 0.8, nu2 = 0.18, k = 0.01, tex = 1,
```

```
 nlegen, ninteg, nprix, prixlst0, prixlst, Func1, Func2, xlen, k0 = 0.01},
μ1 = 0;
μ2 = 0;
σ1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
σ2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
corr = 0.8;
α1 = 1;
α2 = -1;
nlegen = 40;
ninteg = 60;
nprix = 8;
xlen = 3;
legenlst = LegendreCoeffs[nlegen];
Func1 = Interpolation[
  Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
   Table[xlen ( i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
Func2 = Interpolation[Map[
   ({#, FromGaussTODistribution[f2, #, tex, alpha2, beta2, rho2, nu2]}) &,
   Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];

Print["essai=", {
  NormalCallOptionImplicitVol[f1 - f2,
   SpreadOption[Func1, Func2, σ1, σ2, corr, k0, tex, legenlst], k0, tex],
  NormalCallOptionImplicitVol[f1 - f2, Type1SpreadOption[f1, f2, μ1,
    μ2, σ1, σ2, corr, -k0, α1, α2, 0, 0, tex, legenlst], k0, tex],
  NormalCallOptionImplicitVol[f1 - f2, NormalSpreadOption[f1,
    f2, k0, σ1 f1 √tex, σ2 f2 √tex, corr], k0, tex]
 }
];
prixlst0 = Table[0.02 ( i / (nprix + 1) - 0.5), {i, 1, nprix}];
prixlst =
 Interpolation[Map[({#, NormalCallOptionImplicitVol[f1 - f2, SpreadOption[Func1,
       Func2, σ1, σ2, corr, #, tex, legenlst], #, tex]}) &,
   Table[0.03 ( i / (nprix + 1) - 0.5), {i, 1, nprix}]]];
Print["done"];
Plot[{
  prixlst[k1],
  NormalCallOptionImplicitVol[f1 - f2, Type1SpreadOption[f1,
    f2, μ1, μ2, σ1, σ2, corr, -k1, α1, α2, 0, 0, tex, legenlst], k1, tex],
  NormalCallOptionImplicitVol[f1 - f2, NormalSpreadOption[f1,
    f2, k1, σ1 f1 √tex, σ2 f2 √tex, corr], k1, tex]
 },
 {k1, -0.015, 0.015}, PlotRange → {0.007, 0.0095},
 PlotLabel → StringJoin[ "Normal Smile of a Spread option"],
 PlotStyle → {{Thickness[0.0075], RGBColor[1, 0, 0]},
   {Thickness[0.0075], RGBColor[0, 1, 0]}, {Thickness[0.0075], RGBColor[0, 0, 1]}},
```

```
PlotLegend → {"SABR", "LogNormal", "Normal"}, LegendPosition → {1, -1}]]
```

```
essai={0.00839392231009634146607715351865,
  0.00849766540128174151833623956135, 0.00823611158810586994311161461057}
```

done

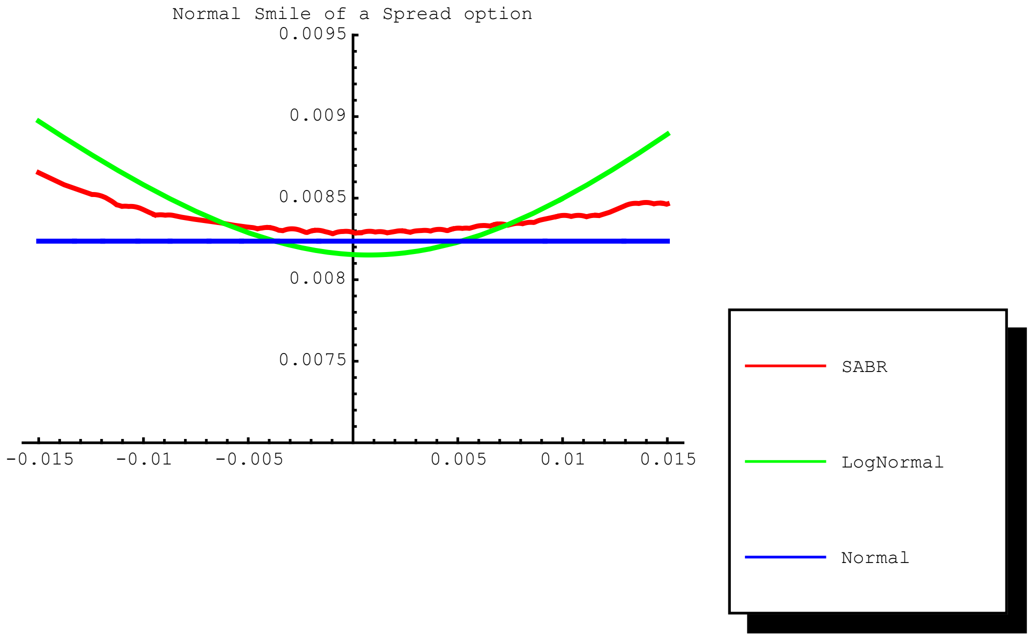


- Graphics -

InverseFunction

done

```
essai={0.0083939223100963414660771535l865,
  0.0084976654012817415183362395613 5, 0.0082361115881058699431116146105 7}
```



- Graphics -

```
legenlst = LegendreCoeffs[30];;
Type1SpreadOption[0.05, 0.05, 0, 0, 0.2, 0.19, 0.8, -0.01, 1, -1, 0, 0, 1, legenlst]
```

0.000184464

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5, nu1 = 0.2,
   f2 = 0.045, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, corr = 0.8, nu2 = 0.18,
   k = 0.01, tex = 1, nlegen, ninteg, nprix, prixlst0, prixlst, Func1, Func2,
   xlen, k0 = 0.01, normalprice, corrSABRImplicit, corrLOGLNImplicit},
  μ1 = 0;
  μ2 = 0;
  σ1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
  σ2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
  corr = 0.8;
  α1 = 1;
  α2 = -1;
  nlegen = 40;
  ninteg = 60;
  nprix = 8;
  xlen = 3;
  legenlst = LegendreCoeffs[nlegen];
  normalprice = NormalSpreadOption[f1, f2, k0, σ1 f1 √tex , σ2 f2 √tex , corr];
  Print["normal price=", normalprice];
  corrLOGLNImplicit =
   corr1 /. FindRoot[Type1SpreadOption[f1, f2, μ1, μ2, σ1, σ2, corr1, -k0,
       α1, α2, 0, 0, tex, legenlst] == normalprice, {corr1, -0.999, +.999}];
  Print["Correlation Implicit pour LogNorm:", corrLOGLNImplicit];
]
```

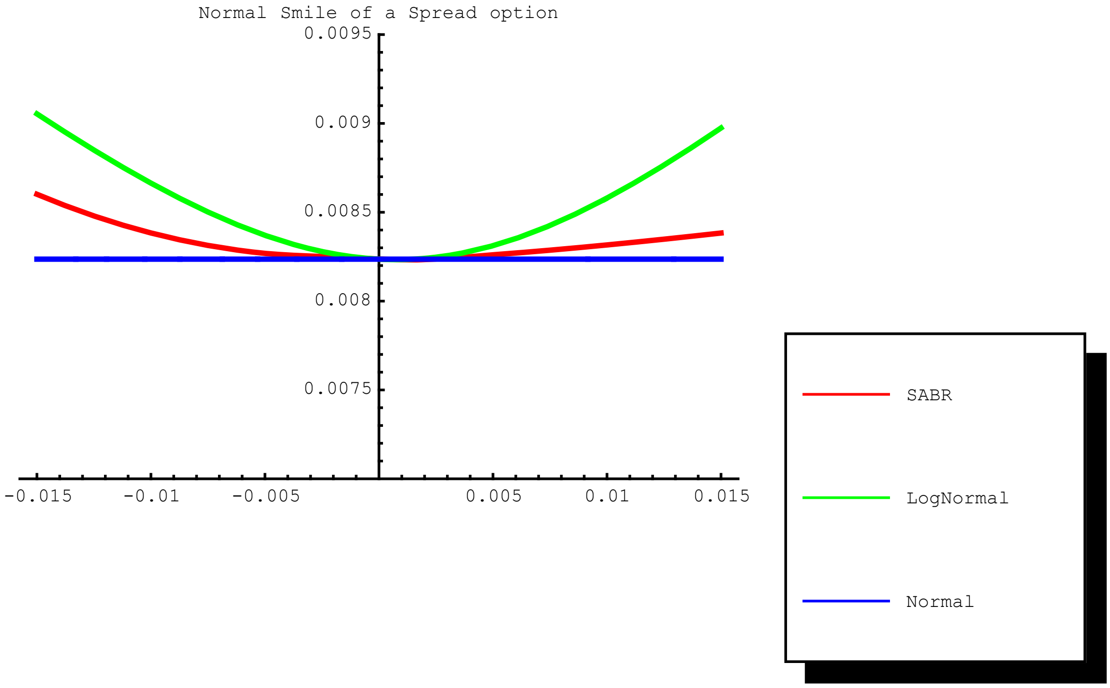
```
normal price=0.00137328

ReplaceAll::reps :
 {FindRoot[Type1SpreadOption[f1$18436, f2$18436, μ1, μ2, σ1, σ2, corr1, -k0$18436, α1, α2, <<4>>] ==
     normalprice$18436, {corr1, -0.999, +0.999}]} is neither a list of replacement rules
   nor a valid dispatch table, and so cannot be used for replacing. More…

ReplaceAll::reps :
 {FindRoot[Type1SpreadOption[f1$18436, f2$18436, μ1, μ2, σ1, σ2, corr1, -k0$18436, α1, α2, <<4>>] ==
     normalprice$18436, {corr1, -0.999, +0.999}]} is neither a list of replacement rules
   nor a valid dispatch table, and so cannot be used for replacing. More…

ReplaceAll::reps :
 {FindRoot[Type1SpreadOption[f1$18436, f2$18436, μ1, μ2, σ1, σ2, corr1, -k0$18436, α1, α2, <<4>>] ==
     normalprice$18436, {corr1, -0.999, +0.999}]} is neither a list of replacement rules
   nor a valid dispatch table, and so cannot be used for replacing. More…

General::stop : Further output of ReplaceAll::reps will be suppressed during this calculation. More…

Correlation Implicit pour LogNorm:
 corr1 /.FindRoot[Type1SpreadOption[f1$18436, f2$18436, μ1, μ2, σ1, σ2, corr1, -k0$18436,
     α1, α2, 0, 0, tex$18436, legenlst] == normalprice$18436, {corr1, -0.999, +0.999}]
```

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5, nu1 = 0.2,
   f2 = 0.045, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, corr = 0.8, nu2 = 0.18,
   k = 0.01, tex = 1, nlegen, ninteg, nprix, prixlst0, prixlst, Func1, Func2,
   xlen, k0 = 0.0, normalprice, corrSABRImplicit, corrLOGLNImplicit},
  μ1 = 0;
  μ2 = 0;
  σ1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
  σ2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
```

```
corr = 0.8;
α1 = 1;
α2 = -1;
nlegen = 40;
ninteg = 60;
nprix = 8;
xlen = 3;
legenlst = LegendreCoeffs[nlegen];
Func1 = Interpolation[
  Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
   Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
Func2 = Interpolation[Map[
   ({#, FromGaussTODistribution[f2, #, tex, alpha2, beta2, rho2, nu2]}) &,
   Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
normalprice = NormalSpreadOption[f1, f2, k0, σ1 f1 √tex, σ2 f2 √tex, corr];
Print["normal price=", normalprice];
corrSABRImplicit =
 corr1 /. FindRoot[SpreadOption[Func1, Func2, σ1, σ2, corr1, k0, tex, legenlst] ==
    normalprice, {corr1, -0.999, +.999}];
Print["Correlation Implicit pour SABR:", corrSABRImplicit];
prixlst0 = Table[0.02 (i / (nprix + 1) - 0.5), {i, 1, nprix}];
prixlst =
 Interpolation[Map[({#, NormalCallOptionImplicitVol[f1 - f2, SpreadOption[Func1,
        Func2, σ1, σ2, corrSABRImplicit, #, tex, legenlst], #, tex]}) &,
   Table[0.03 (i / (nprix + 1) - 0.5), {i, 1, nprix}]]];
deltalnvol = NormalCallOptionImplicitVol[f1 - f2, normalprice, k0, tex] -
  NormalCallOptionImplicitVol[f1 - f2, Type1SpreadOption[f1, f2,
    μ1, μ2, σ1, σ2, corr, -k0, α1, α2, 0, 0, tex, legenlst], k0, tex];
Print["delta vol Ln=", deltalnvol];
Print["done"];
Plot[{
  prixlst[k1],
  NormalCallOptionImplicitVol[f1 - f2, Type1SpreadOption[f1, f2, μ1, μ2,
     σ1, σ2, corr, -k1, α1, α2, 0, 0, tex, legenlst], k1, tex] + deltalnvol,
  NormalCallOptionImplicitVol[f1 - f2, NormalSpreadOption[f1,
    f2, k1, σ1 f1 √tex, σ2 f2 √tex, corr], k1, tex]
 },
 {k1, -0.015, 0.015}, PlotRange → {0.007, 0.0095},
 PlotLabel → StringJoin[ "Normal Smile of a Spread option"],
 PlotStyle → {{Thickness[0.0075], RGBColor[1, 0, 0]},
   {Thickness[0.0075], RGBColor[0, 1, 0]}, {Thickness[0.0075], RGBColor[0, 0, 1]}},
 PlotLegend → {"SABR", "LogNormal", "Normal"}, LegendPosition → {1, -1}]
]
```

```
normal price=0.00637328

Correlation Implicit pour SABR:0.802659
```

```
delta vol Ln=0.0000821660024635025610581622448

done
```




```
- Graphics -
```

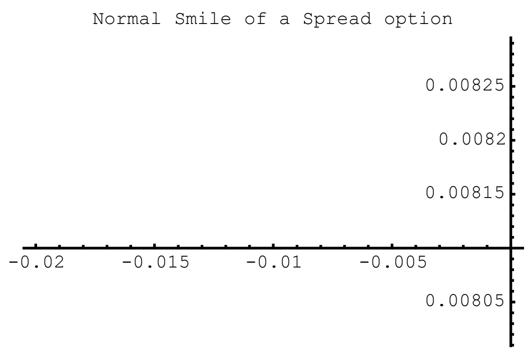
corr1

```
corr1
```

? Type1SpreadOption

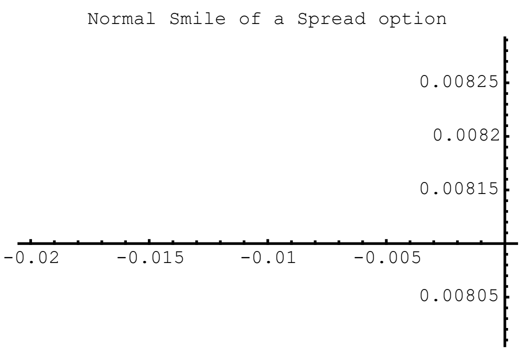
```
Global`Type1SpreadOption

Type1SpreadOption[L1_, L2_, μ1_, μ2_, σ1_, σ2_, ρ12_, α0_, α1_, α2_, β0_, β1_, T_, lst_] :=
 LN2GenericSpreadOptionType1[α0, α1, α2, L1, L2, μ1, μ2, σ1, σ2, ρ12, α0, α1, α2, β0, β1, T, lst]
```

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5, nu1 = 0.2,
  f2 = 0.045, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, corr = 0.8, nu2 = 0.18,
  k = 0.01, tex = 1, nlegen, ninteg, Func1, Func2, xlen, k0 = 0.01, ll},
 μ1 = 0;
 μ2 = 0;
 σ1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
 σ2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
 corr = 0.8;
 α1 = 1;
 α2 = -1;
 nlegen = 40;
 ninteg = 40;
 xlen = 6;
 legenlst = LegendreCoeffs[nlegen];
 Func1 = Interpolation[
   Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
    Table[xlen ( i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 Func2 = Interpolation[Map[
    ({#, FromGaussTODistribution[f2, #, tex, alpha2, beta2, rho2, nu2]}) &,
    Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 ll = Map[({#, NormalCallOptionImplicitVol[f1 - f2, SpreadOption[Func1, Func2,
        corr, #, tex, legenlst], #, tex]}) &, Table[-0.02 (i / 25), {i, 1, 25}]];
 ListPlot[ll, PlotLabel → StringJoin[ "Normal Smile of a Spread option"],
  PlotStyle → {{Thickness[0.0075], RGBColor[1, 0, 0]},
    {Thickness[0.0075], RGBColor[0, 1, 0]}, {Thickness[0.0075], RGBColor[0, 0, 1]}}] ]
```



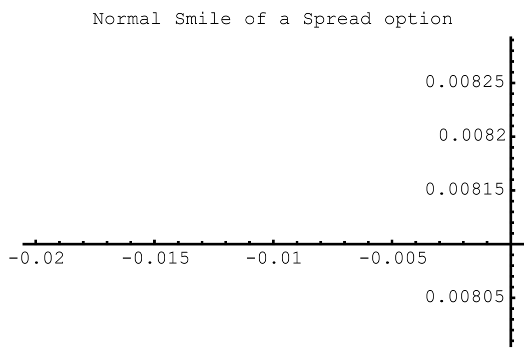
- Graphics -

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5, nu1 = 0.2,
  f2 = 0.045, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, corr = 0.8, nu2 = 0.18,
  k = 0.01, tex = 1, nlegen, ninteg, Func1, Func2, xlen, k0 = 0.01, ll},
 μ1 = 0;
 μ2 = 0;
 σ1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
 σ2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
 corr = 0.8;
 α1 = 1;
 α2 = -1;
 nlegen = 40;
 ninteg = 40;
 xlen = 6;
 legenlst = LegendreCoeffs[nlegen];
 Func1 = Interpolation[
   Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
    Table[xlen ( i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 Func2 = Interpolation[Map[
    ({#, FromGaussTODistribution[f2, #, tex, alpha2, beta2, rho2, nu2]}) &,
    Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 ll = Map[({#, NormalCallOptionImplicitVol[f1 - f2, SpreadOption[Func1, Func2, σ1, σ2,
       corr, #, tex, legenlst], #, tex]}) &, Table[-0.02 (i / 25), {i, 1, 25}]];
 ListPlot[ll, PlotLabel → StringJoin[ "Normal Smile of a Spread option"],
  PlotStyle → {{Thickness[0.0075], RGBColor[1, 0, 0]},
    {Thickness[0.0075], RGBColor[0, 1, 0]}, {Thickness[0.0075], RGBColor[0, 0, 1]}}] ]
```



- Graphics -

```
Module[{f1 = 0.05, alpha1 = 0.11, beta1 = 0.7, rho1 = -0.5, nu1 = 0.2,
  f2 = 0.045, alpha2 = 0.112, beta2 = 0.7, rho2 = -0.5, corr = 0.8, nu2 = 0.18,
  k = 0.01, tex = 1, nlegen, ninteg, Func1, Func2, xlen, k0 = 0.01, ll},
 μ1 = 0;
 μ2 = 0;
 σ1 = ImpVolSABR[f1, f1 0.9999, tex, alpha1, beta1, rho1, nu1];
 σ2 = ImpVolSABR[f2, f1 0.9999, tex, alpha2, beta2, rho2, nu2];
 corr = 0.8;
 α1 = 1;
 α2 = -1;
 nlegen = 40;
 ninteg = 40;
 xlen = 7;
 legenlst = LegendreCoeffs[nlegen];
 Func1 = Interpolation[
   Map[({#, FromGaussTODistribution[f1, #, tex, alpha1, beta1, rho1, nu1]}) &,
    Table[xlen ( i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 Func2 = Interpolation[Map[
    ({#, FromGaussTODistribution[f2, #, tex, alpha2, beta2, rho2, nu2]}) &,
    Table[xlen (i / (ninteg + 1) - 0.5), {i, 1, ninteg}]]];
 ll = Map[({#, NormalCallOptionImplicitVol[f1 - f2, SpreadOption[Func1, Func2,
        corr, #, tex, legenlst], #, tex]}) &, Table[-0.02 (i / 25), {i, 1, 25}]];
 ListPlot[ll, PlotLabel → StringJoin[ "Normal Smile of a Spread option"],
  PlotStyle → {{Thickness[0.0075], RGBColor[1, 0, 0]},
    {Thickness[0.0075], RGBColor[0, 1, 0]}, {Thickness[0.0075], RGBColor[0, 0, 1]}}] ]
```



- Graphics -

Shiftedlog Model

```
d2[x_] := Log[(f + alpha) / (alpha + x)] / (sigma Sqrt[T]) - 1 / 2 sigma Sqrt[T]

d2inv[x_, f_, sigm_, alpha_, T_] := (f + alpha) Exp[-1 / 2 sigm^2 T - sigm Sqrt[T] x] - alpha

phi[x_] := Exp[-x^2 / 2] / Sqrt[2 Pi]

gaussianmix[corr_, x_, y_] := corr x + Sqrt[1 - corr^2] y

gaussiantodistribution[f_, sigma_, alpha_, t_, x_] := d2inv[-x, f, sigma, alpha, t]
```

$$\text{LimitInverse[a\_, x\_, f\_, sigma\_, t\_, corr\_] :=} \frac{\text{sigma}^2\ \text{t} - 2\ \text{corr0 sigma}\ \sqrt{\text{t}}\ \text{x} + 2\ \text{Log}\left[\frac{\text{a}}{\text{f}}\right]}{2\ \sqrt{1 - \text{corr}^2}\ \text{sigma}\ \sqrt{\text{t}}}$$

```
payoff[f1_, f2_, k_] := If[f1 - f2 ≥ k, 1., 0.0]

integrand[x_, y_, k_, t0_] :=
 payoff[gaussiantodistribution[f01, sigma01, alpha1, t0, x],
  gaussiantodistribution[f02, sigma02, alpha1, t0, gaussianmix[corr0, x, y]], k]

LimitS2[S1_, k_] := (k - S1) / (-1)

prix[k_, n_] := Module[{c = HermiteCoeffs[n], sum, i, j, f1, f2},
  sum = 0;
  Do[
   x0 = c[[i, 1]] Sqrt[2];
   f1 = gaussiantodistribution[f01, sigma01, t0, x0];
   Do[
    y0 = c[[j, 1]] Sqrt[2];
    f2 = gaussiantodistribution[f02, sigma02, t0, gaussianmix[corr0, x0, y0]];
    sum += payoff[f1, f2, k] × c[[i, 2]] × c[[j, 2]] ;
    , {j, 1, n}], {i, 1, n}];
  sum / Pi
 ]

prix5[k_, n1_, n2_] := Module[
  {c = HermiteCoeffs[n1], c2 = LaguerreCoeffs[n2], sum, sum0, i, j, f1, f2, a, x0, y0},
  sum = 0;
  Do[
   x0 = c[[i, 1]] Sqrt[2];
   f1 = gaussiantodistribution[f01, sigma01, t0, x0];
   a = LimitS2[f1, k];
   za = LimitInverse[a, x0, f02, sigma02, t0, corr0];
   Print["f1=", f1, " a=", a, " za=", za];
   sum0 = 0;
   Plot[Exp[-y0^2 / 2] / Sqrt[2 Pi], {y0, -6, za}, PlotRange → All] ×
    sum0 = c[[i, 2]] × NIntegrate[Exp[-y0^2 / 2] / Sqrt[2 Pi], {y0, -Infinity, za}];
   sum01 = c[[i, 2]] × CoeffBasedIntegrate[(Exp[-(# - za)^2 / 2 + #] / Sqrt[2 Pi]) &, c2];
   Print["exact integrale=", sum0, " Laguerre approx=", sum01];
   sum += sum0;
   , {i, 1, n1}];
  sum / Sqrt[Pi]
 ]
```

```
prix8[k_, n_] :=
 Module[{c = HermiteCoeffs[n], c2 = LaguerreCoeffs[n], sum, i, j, f1, f2, a},
  sum = 0;
  Do[
   x0 = c[[i, 1]] Sqrt[2];
   f1 = gaussiantodistribution[f01, sigma01, t0, x0];
   a = LimitS2[f1, k];
   za = LimitInverse[a, x0, f02, sigma02, t0, corr0];
   (* Print["f1=",f1," a=",a," za=",za]; *)
   Do[
    y0 = za - c2[[j, 1]] ;
    f2 = gaussiantodistribution[f02, sigma02, t0, gaussianmix[corr0, x0, y0]];
    (* Print["f2=",f2];*)
    sum += payoff[f1, f2, k] Exp[-y0^2 / 2 + c2[[j, 1]]] c[[i, 2]] × c2[[j, 2]] ;
    , {j, 1, n}], {i, 1, n}];
  sum / (Pi Sqrt[2])
 ]
```

```
f01 = 0.06;
sigma01 = 0.15;
f02 = 0.06;
sigma02 = 0.15;
t0 = 5.0;
corr0 = 0.0;
alpha01 = 0.1;
```

```
NIntegrate[integrand[x, y, 0.0015] × NormDens[x] × NormDens[y],
 {x, -5, +5}, {y, -5, +5}, WorkingPrecision → 20, GaussPoints → 32]
```

```
NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following:
   singularity, value of the integration being 0, oscillatory integrand, or insufficient
   WorkingPrecision. If your integrand is oscillatory try using the option Method->Oscillatory in
   NIntegrate. More…

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 13 recursive bisections
   in y near {x, y} = {-0.00183105, -0.000610352}. More…
```

```
0.4996
```

```
f01 = 0.06; sigma01 = 0.15; f02 = 0.06; sigma02 = 0.15; t0 = 5.0; corr0 = 0.0;
prix[0.0015, 50]
```

```
$Aborted
```

```
f01 = 0.06; sigma01 = 0.15; f02 = 0.06; sigma02 = 0.15; t0 = 5.0; corr0 = 0.0;
prix8[0.0015, 10]
```

CMS with SABR

```
DateStrip[nYear_, nperYear_, initialstub_] :=
 Module[{i, deltaT = 1 / nperYear, nbpayments = nYear * nperYear},
  Table[{i * deltaT + initialstub, deltaT}, {i, 1, nbpayments}]]
```

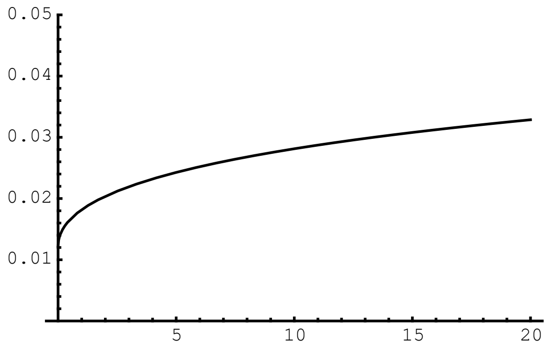
```
ds = DateStrip[10, 4, 0.1]
```

$\left\{\left\{0.35, \frac{1}{4}\right\}, \left\{0.6, \frac{1}{4}\right\}, \left\{0.85, \frac{1}{4}\right\}, \left\{1.1, \frac{1}{4}\right\}, \left\{1.35, \frac{1}{4}\right\}, \left\{1.6, \frac{1}{4}\right\}, \left\{1.85, \frac{1}{4}\right\}, \left\{2.1, \frac{1}{4}\right\}, \left\{2.35, \frac{1}{4}\right\}, \left\{2.6, \frac{1}{4}\right\}, \left\{2.85, \frac{1}{4}\right\}, \left\{3.1, \frac{1}{4}\right\}, \left\{3.35, \frac{1}{4}\right\}, \left\{3.6, \frac{1}{4}\right\}, \left\{3.85, \frac{1}{4}\right\}, \left\{4.1, \frac{1}{4}\right\}, \left\{4.35, \frac{1}{4}\right\}, \left\{4.6, \frac{1}{4}\right\}, \left\{4.85, \frac{1}{4}\right\}, \left\{5.1, \frac{1}{4}\right\}, \left\{5.35, \frac{1}{4}\right\}, \left\{5.6, \frac{1}{4}\right\}, \left\{5.85, \frac{1}{4}\right\}, \left\{6.1, \frac{1}{4}\right\}, \left\{6.35, \frac{1}{4}\right\}, \left\{6.6, \frac{1}{4}\right\}, \left\{6.85, \frac{1}{4}\right\}, \left\{7.1, \frac{1}{4}\right\}, \left\{7.35, \frac{1}{4}\right\}, \left\{7.6, \frac{1}{4}\right\}, \left\{7.85, \frac{1}{4}\right\}, \left\{8.1, \frac{1}{4}\right\}, \left\{8.35, \frac{1}{4}\right\}, \left\{8.6, \frac{1}{4}\right\}, \left\{8.85, \frac{1}{4}\right\}, \left\{9.1, \frac{1}{4}\right\}, \left\{9.35, \frac{1}{4}\right\}, \left\{9.6, \frac{1}{4}\right\}, \left\{9.85, \frac{1}{4}\right\}, \left\{10.1, \frac{1}{4}\right\}\right\}$

```
zerocurve = (Log[1.5 + #^0.5 / 3] * 0.03) &
```

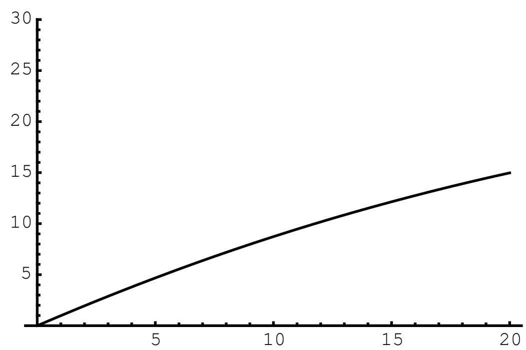
$\mathrm{Log}\left[1.5 + \frac{\#1^{0.5}}{3}\right] 0.03\ \&$

```
Plot[zerocurve[t], {t, 0.00001, 20}, PlotRange → {0, .05}]
```



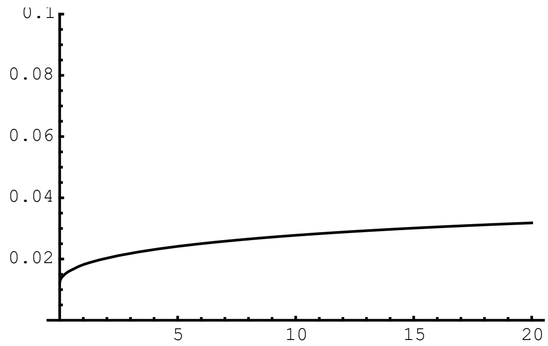
- Graphics -

```
Plot[AnnuityPrice0[SwapDateStrip[t], zerocurve], {t, 0.00001, 20}, PlotRange → {0, 30}]
```



- Graphics -

```
Plot[SwapRate[t, zerocurve], {t, 0.001, 20}, PlotRange → {0, 0.1}]
```



- Graphics -

```
CMSFunctionEval[dtstrp_, x_, T_] :=
 x / Sum[dtstrp[[i, 2]] (1 + x) ^ (dtstrp[[i, 1]] - T), {i, 1, Length[dtstrp]}]

D[CMSFunctionEval[ds, x, 12], x, x];

ZeroCouponPrice[T_, zrcppricefunction_] := zrcppricefunction[T]

Discount[T_, zrcppricefunction_] := (1 + ZeroCouponPrice[T, zrcppricefunction]) ^ (-T)

SwapDateStrip[T_] := Module[{i}, If[T < 1, {{T, T}},
   Prepend[Table[{i + T - Floor[T], 1}, {i, 1, Floor[T]}], {T - Floor[T], T - Floor[T]}]]]

SwapDateStrip[2.1]

{{1.1, 1}, {2.1, 1}, {0.1, 0.1}}

SwapRate[T_, zrcppricefunction_] := (1 - Discount[T, zrcppricefunction]) /
  AnnuityPrice0[SwapDateStrip[T], zrcppricefunction]

SwapRate[12, zerocurve]

0.0288022

AnnuityPrice0[dtstrp_, zrcppricefunction_] :=
 Module[{i}, Sum[dtstrp[[i, 2]] (1 + zrcppricefunction[dtstrp[[i, 1]]]) ^ (-dtstrp[[i, 1]]),
   {i, 1, Length[dtstrp]}]]

AnnuityPricex[dtstrp_, x_] :=
 Module[{i}, Sum[dtstrp[[i, 2]] (1 + x) ^ (-dtstrp[[i, 1]]), {i, 1, Length[dtstrp]}]]
```

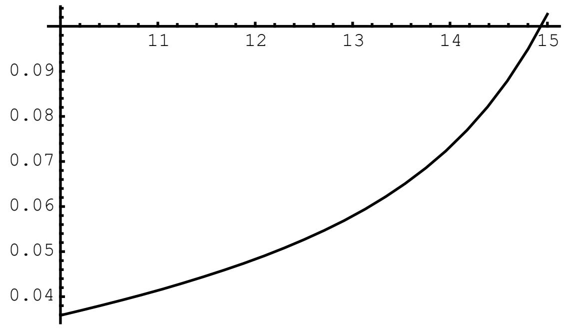
```
CMSPrice[dtstrp_, T_, zrcppricefunction_, κ_, α_, β_, ρ_, ν_] :=
 Module[{cms0, fκ, fderiveκ, call0, put0, callstrip, putstrip, legenlst, x, fderive2},
  fκ = CMSFunctionEval[dtstrp, κ, T];
  fderiveκ = D[CMSFunctionEval[dtstrp, x, T], x] /. {x → κ};
  cms0 = SwapRate[T, zrcppricefunction];
  call0 = CallNewOption[cms0, κ, T, α, β, ρ, ν];
  put0 = CallNewOption[cms0, κ, T, α, β, ρ, ν] - (κ - cms0);
  fderive2 = D[CMSFunctionEval[dtstrp, x, T], x, x] /. { x → K};
  callstrip =
   NIntegrate[fderive2 PutNewOption[cms0, K, T, α, β, ρ, ν], {K, 0.00001, κ}];
  putstrip = NIntegrate[fderive2 CallNewOption[cms0, K, T, α, β, ρ, ν], {K, κ, 1}];
  AnnuityPricex[SwapDateStrip[T], cms0]
   (fκ + fderiveκ (call0 - put0) + callstrip + putstrip)
 ]
```

```
der2 = D[CMSFunctionEval[ds, x, 12], x, x];
```

```
CMSPrice[ds, 12, zerocurve, 0.03, 0.11, 0.7, -0.5, 0.2]
```

0.0482952

```
Plot[CMSPrice[ds, K, zerocurve, 0.03, 0.11, 0.7, -0.5, 0.2], {K, 10, 15}]
```

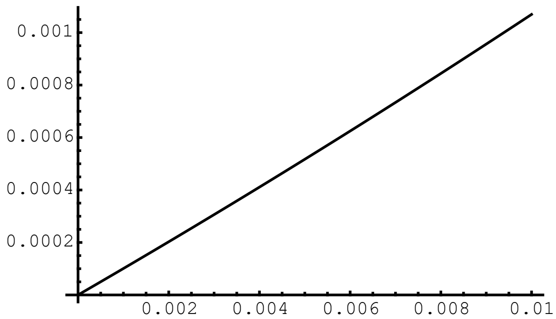


- Graphics -

```
FindRoot[D[CMSFunctionEval[ds, x, 12], x, x] == 0, {x, 0, 10}]
```

{x → −0.223056}

```
Plot[CMSFunctionEval[ds, x, 12], {x, 0, 0.01}]
```



- Graphics -

12 !

479 001 600