Moment matching for Asian Options 02-22-17

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.

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

Purpose

My last previous notebook on this topic was *Moment matching for Asian Options 01-22-11*, which consolidated several pieces of earlier work. As I worked through that notebook, I realized that advances in Mathematica technology suggested a rework of the old calculations.

Included are

1) Demonstration that arithmetic average price is approximated by log normal distribution
2) Moment matching for a continuously sampled average
3) Moment matching for a discretely sampled average

Background

Asian options are financial derivatives that depend on a time average of the price. The archetype is an option with a simple put or call payoff on the arithmetic average of periodically sampled prices.

$$A(T) = \frac{1}{n+1}\sum_{i=0}^{n} S(t_i) \tag{1}$$

where T = n Δt and Δt is a constant interval of time, say one day.

For the example of a European style call option on this average, the fair value is given by

$$C(K,\ T,\ t) = \mathbb{E}_Q\left[e^{-r(T-t)} \max(A(T) - K,\ 0)\right] \tag{2}$$

an Asian call option with fixed strike. There are various Asian option types depending on how the price is sampled, how the average is constructed, special payoff features and more. However, central to the valuation of Asian options is knowledge is how the average A(T) is distributed.

Within the Black Scholes framework, stock prices are assumed to be lognormally distributed. Thus, A(T) is an arithmetic sum of lognormally distributed random variables, specifically

$$A(T) = \frac{1}{n+1}\left(S_0 + S_0\, e^{\left(r-\frac{\sigma^2}{2}\right)\Delta t + \epsilon_1\, \sigma \sqrt{\Delta t}} + \ldots + S_0\, e^{\left(r-\frac{\sigma^2}{2}\right) i\, \Delta t + (\epsilon_1 + \ldots\, \epsilon_i)\, \sigma \sqrt{i\, \Delta t}} + \ldots\right) \tag{3}$$

This distribution for this sum has no closed form and must be approximated. It is observed that A(T) is reasonably approximated by a lognormal distribution. If the parameters of the approximating distribution can be obtained, then the expectation in (2) can be evaluated in closed form, just like plain vanilla options.

The parameters of the approximating lognormal distribution for A are determined by matching the moments of the assumed lognormal distribution to the moments calculated from (3), a procedure called moment matching. Since a lognormal distribution is determined by two parameters, only two moments need be estimated.

In this notebook, I illustrate how *Mathematica* facilitates can be used to facilitate some of the calculations associated with moment matching. In my work as a quant, I generated many *Mathematica* notebooks on Asian options.

1 The distribution A(T) is approximated by a lognormal distribution

I start by using Monte Carlo simulation to illustrate how A(T) is reasonably approximated by a lognormal distribution. The assumed stock dynamics are

$$\frac{\mathbf{dS}_t}{S_t} = \mu\, \mathbf{dt} + \sigma\, \mathbf{dz}_t$$

$$S_t = S_0 \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\mathbf{dt} + \sigma\, z_t\right]$$

where z_t is a standard Wiener process.

I use Monte Carlo simulation to illustrate how A(T) is distributed. In this notebook, I'll take advantage of *GeometricBrownianMotionProcess,* which was introduced into Mathematica in 2012. Simulating using RandomFunction is actually quite fast. I make some comments about simulation speed in Appendix C.

In[19]:=

```
Module[{GBM, info},
 GBM = GeometricBrownianMotionProcess[μ, σ, S_0];
 info = TraditionalForm /@ {Refine[PDF[GBM[t], S_t], S_t > 0],
    Mean[GBM[t]], Variance[GBM[t]], CovarianceFunction[GBM, s, t]};
 info = Join[{Stl /@ {"PDF", "mean", "variance", "covariance"}}, {info}];
 Grid[Transpose@info, Frame → All]]
```

Out[19]=

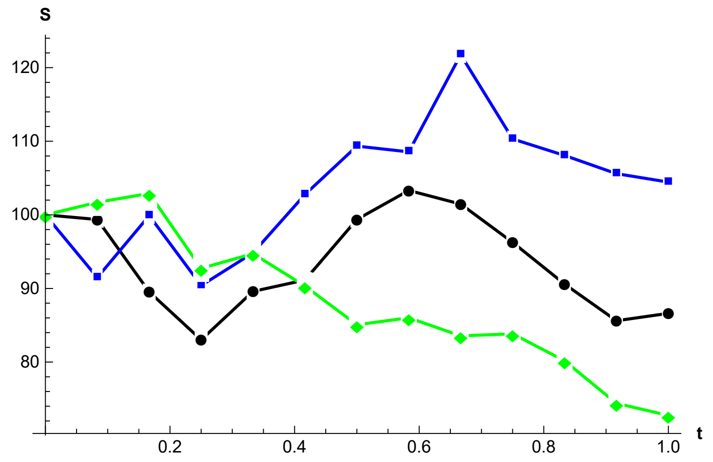
PDF	$\exp\left(-\frac{1}{2\sigma^2 t}\left(\log(S_t)-\log(S_0)-t\left(\mu-\frac{\sigma^2}{2}\right)\right)^2\right)\Big/\left(\sqrt{2\pi}\,\sigma\sqrt{t}\,S_t\right)$
mean	$S_0\, e^{t\left(\mu-\frac{\sigma^2}{2}\right)+\frac{\sigma^2 t}{2}}$
variance	$\left(e^{\sigma^2 t}-1\right) e^{2\left(\log(S_0)+t\left(\mu-\frac{\sigma^2}{2}\right)\right)+\sigma^2 t}$
covariance	$S_0^2\, e^{\mu\,(s+t)}\left(e^{\sigma^2 \min(s,t)}-1\right)$

I generate some representative price paths for nominal parameters (1 year, sampled each month).

In[20]:=

```
Module[{S0 = 100, r = 0, q = 0, σ = 0.2, T = 1.0, ΔT = 1/12, pathList, GBM},
 SeedRandom[123];
 GBM = GeometricBrownianMotionProcess[(r - q), σ, S0];
 pathList = RandomFunction[GBM, {0, T, ΔT}, 3];
 ListLinePlot[pathList, AxesLabel → {Stl["t"], Stl["S"]},
  PlotStyle → {Black, Blue, Green}, PlotMarkers → Automatic]]
```

Out[20]=

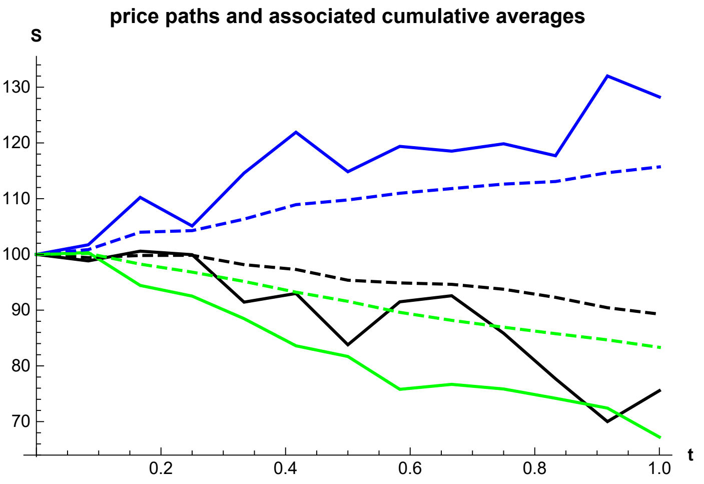


Here are some paths together with their corresponding averages.

In[21]:=

```
Module[{S0 = 100, r = 0, q = 0, σ = 0.2, T = 1.0,
  ΔT = 1/12, nPaths = 3, pathList, averagePathList, GBM, lab},
 GBM = GeometricBrownianMotionProcess[(r - q), σ, S0];
 SeedRandom[123];
 pathList = Table[RandomFunction[GBM, {0, T, ΔT}], {nPaths}];
 averagePathList = CumulativeAverage /@ pathList;
 (* The TemporalData object returned by RandomFunction is sometimes inconvenient
   The following statement converts it to a simple list of time-value pairs *)
 pathList = Flatten[Normal /@ pathList, 1];
 lab = Stl["price paths and associated cumulative averages"];
 ListLinePlot[Join[pathList, averagePathList], AxesLabel → {Stl["t"], Stl["S"]},
  PlotStyle → {Black, Blue, Green, Directive[Black, Dashed],
    Directive[Blue, Dashed], Directive[Green, Dashed]}, PlotLabel → lab]]
```

Out[21]=



In[1]:=

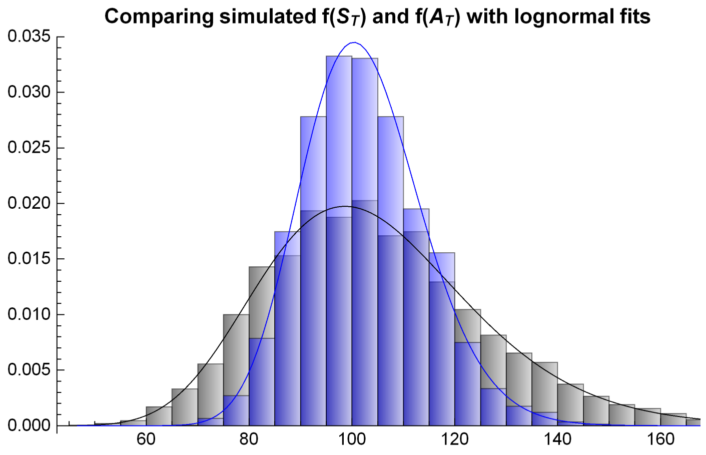
```
Clear[CumulativeAverage];
(* This is illustrative. It could be made more elegant or efficient. *)
CumulativeAverage[path_] :=
 Module[{dat, times, prices, sums, cumAve},
  dat = Normal[path]⟦1⟧;
  {times, prices} = {dat⟦All, 1⟧, dat⟦All, 2⟧};
  sums = Accumulate[prices];
  cumAve = Table[sums⟦i⟧/i, {i, 1, Length[sums]}];
  Transpose[{times, cumAve}] ]
```

I compare the simulated distributions for A(T) and S(T). Of course, S(T) is lognormal by construction, but A(T) is also well-fitted by that distribution.

In[22]:=

```
Module[{S0 = 100, r = 0.05, q = 0, σ = 0.2, T = 1.0, ΔT = 1/12,
  nPaths = 4000, seed = 123, pathList, fST, fAT, 𝒟ST, 𝒟AT, GBM, fittedPDFs},
 GBM = GeometricBrownianMotionProcess[(r - q), σ, S0];
 SeedRandom[seed];
 pathList = Table[RandomFunction[GBM, {0, T, ΔT}], {nPaths}];
 pathList = Flatten[Normal /@ pathList, 1];
 (* ST is the final price in the path *)
 fST = (#[[-1, 2]]) & /@ pathList;
 (* AT is the mean of the price path *)
 fAT = (Mean[#[[All, 2]]]) & /@ pathList;
 (* fitting S(T) and A(T) to lognormal distributions *)
 𝒟ST = EstimatedDistribution[fST, LogNormalDistribution[μμ, σσ]];
 𝒟AT = EstimatedDistribution[fAT, LogNormalDistribution[μμ, σσ]];
 fittedPDFs = Module[{Smin, Smax, curve},
   {Smin, Smax} = MinMax[fST];
   curve[1] = ({#, PDF[𝒟ST, #]}) & /@ Range[Smin, Smax, 1];
   curve[1] = {Black, Line[curve[1]]};
   curve[2] = ({#, PDF[𝒟AT, #]}) & /@ Range[Smin, Smax, 1];
   curve[2] = {Blue, Line[curve[2]]};
   {curve[1], curve[2]}];
 Histogram[{fST, fAT}, 40, "PDF",
  ChartElementFunction → "FadingRectangle", ChartStyle → {Black, Blue},
  PlotLabel → Style["Comparing simulated f(ST) and f(AT) with lognormal fits",
    Bold], Epilog → fittedPDFs]]
```

Out[22]=



2 Moment matching for a continuously sampled average

Although discrete sampling of a price path is of the most practical interest, it is useful to derive expressions for the moments of the continuously sampled average

$$A_{\text{cont}}(T) = \frac{1}{T-t}\int_t^T S_u \, dt \tag{4}$$

If I take t = 0, for the sake of simplicity, the first moment is

$$\begin{aligned} M_1 &= \mathbb{E}[A_{\text{cont}}(T)] \\ &= \mathbb{E}\left[\frac{1}{T}\int_0^T S_0\, e^{\left(r-\frac{\sigma^2}{2}\right)u+\sigma z_u}\, du\right] \\ &= \frac{S_0}{T}\,\frac{e^{r T}-1}{r} \end{aligned} \tag{5}$$

This result is derived in detail in Appendix A.

I define a function for this moment.

```
In[23]:= Clear[M1Continuous];
         M1Continuous[S0_, r_, T_] :=
           (-1 + ⅇ^(r T)) S0
           ─────────────────
                 r T
```

Similarly, the second moment is

$$\begin{aligned} M_2 &= \mathbb{E}\left[A_{\text{cont}}(T)^2\right] \\ &= \left(\frac{S_0}{T}\right)^2 \mathbb{E}\left[\int_0^T\int_0^T e^{\left(r-\frac{\sigma^2}{2}\right)u+\sigma z_u}\, e^{\left(r-\frac{\sigma^2}{2}\right)v+\sigma z_v}\, dv\, du\right] \\ &= \left(\frac{S_0}{T}\right)^2 \left(\left(2\left(\left(1-2 e^{r T}+e^{T\left(2 r+\sigma^2\right)}\right) r-\left(-1+e^{r T}\right)\sigma^2\right)\right) \Big/ \left(r\left(r+\sigma^2\right)\left(2 r+\sigma^2\right)\right)\right) \end{aligned} \tag{6}$$

```
In[25]:= Clear[M2Continuous];
         M2Continuous[S0_, r_, σ_, T_] :=
           (S0/T)^2 ((2 ((1 - 2 ⅇ^(r T) + ⅇ^(T (2 r+σ^2))) r - (-1 + ⅇ^(r T)) σ^2)) / (r (r + σ^2) (2 r + σ^2)))
```

The moments of a lognormal distribution with parameters *μ*A and *σ*A are

```
In[27]:= w2[1] = Module[{𝒟},
           𝒟 = LogNormalDistribution[μA, σA];
           {M1 == Moment[𝒟, 1], M2 == Moment[𝒟, 2]}]

Out[27]= {M1 == ⅇ^(μA+σA^2/2), M2 == ⅇ^(2 μA+2 σA^2)}
```

In[28]:=

```
w2[2] = Solve[w2[1], {μA, σA}] /. {C[1] → 0, C[2] → -0}
```

Out[28]=

$$\left\{\left\{\mu A \to \text{Log}\left[-\frac{\text{M1}^2}{\sqrt{\text{M2}}}\right],\ \sigma A \to -\sqrt{2}\ \sqrt{\text{Log}\left[-\frac{\sqrt{\text{M2}}}{\text{M1}}\right]}\right\},\right.$$

$$\left\{\mu A \to \text{Log}\left[-\frac{\text{M1}^2}{\sqrt{\text{M2}}}\right],\ \sigma A \to \sqrt{2}\ \sqrt{\text{Log}\left[-\frac{\sqrt{\text{M2}}}{\text{M1}}\right]}\right\},$$

$$\left.\left\{\mu A \to \text{Log}\left[\frac{\text{M1}^2}{\sqrt{\text{M2}}}\right],\ \sigma A \to -\sqrt{2}\ \sqrt{\text{Log}\left[\frac{\sqrt{\text{M2}}}{\text{M1}}\right]}\right\},\ \left\{\mu A \to \text{Log}\left[\frac{\text{M1}^2}{\sqrt{\text{M2}}}\right],\ \sigma A \to \sqrt{2}\ \sqrt{\text{Log}\left[\frac{\sqrt{\text{M2}}}{\text{M1}}\right]}\right\}\right\}$$

The financially relevant branch is the fourth one

In[29]:=

```
w2[3] = w2[2][[4]]
```

Out[29]=

$$\left\{\mu A \to \text{Log}\left[\frac{\text{M1}^2}{\sqrt{\text{M2}}}\right],\ \sigma A \to \sqrt{2}\ \sqrt{\text{Log}\left[\frac{\sqrt{\text{M2}}}{\text{M1}}\right]}\right\}$$

I need to relate the parameters of this distribution to the calculated moments.

In[34]:=

```
w2[4] = {M1 == Moment[LogNormalDistribution[μA , σA], 1],
   M2 == Moment[LogNormalDistribution[μA , σA ], 2]} // Simplify
```

Out[34]=

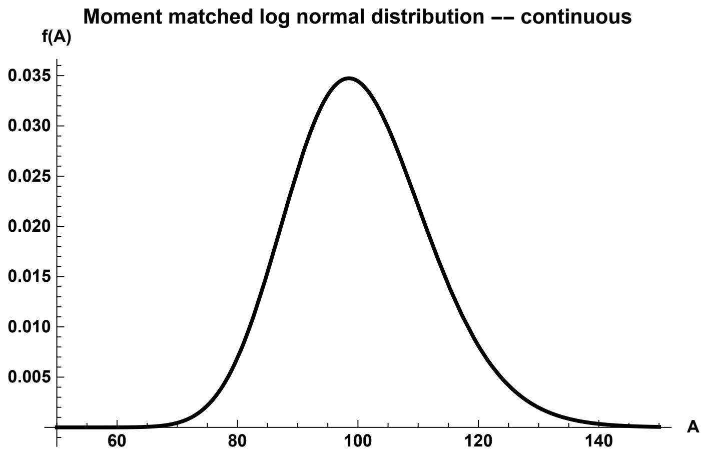
$$\left\{e^{\mu A+\frac{\sigma A^2}{2}} == \text{M1},\ e^{2\left(\mu A+\sigma A^2\right)} == \text{M2}\right\}$$

For example,

In[36]:=

```
Module[{S0 = 100, r = 0.01, q = 0, σ = 0.2, T = 1.0, μA, σA, M1, M2, 𝒟},
 M1 = M1Continuous[S0, r, T];
 M2 = M2Continuous[S0, r, σ, T];
 μA = 1/2 (4 Log[M1] - Log[M2]);
 σA = √(-2 Log[M1] + Log[M2]);
 𝒟 = LogNormalDistribution[μA, σA];
 Plot[PDF[𝒟, A], {A, 50, 150},
  PlotStyle → Directive[Thick, Black],
  PlotLabel → Stl["Moment matched log normal distribution -- continuous"],
  AxesLabel → {Stl["A"], Stl["f(A)"]}]]
```

Out[36]=



I check this moment matched distribution by comparing against a simulated distribution for monthly sampling.

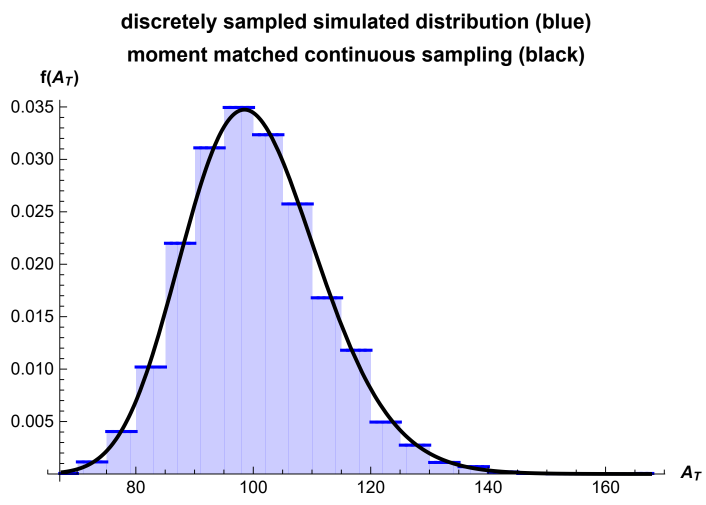
In[37]:=

```
Module[{S0 = 100, r = 0.01, q = 0, σ = 0.2, T = 1.0, ΔT = 1/12, nPaths = 4000,
  seed = 123, pathList, fAT, ℋ, M1, M2, μA, σA, 𝒟, GBM, curve, G},
 GBM = GeometricBrownianMotionProcess[(r - q), σ, S0];
 SeedRandom[seed];
 pathList = Table[RandomFunction[GBM, {0, T, ΔT}], {nPaths}];
 pathList = Flatten[Normal /@ pathList, 1];
 (* The monthly sampled simulated A(T) *)
 fAT = (Mean[#[[All, 2]]]) & /@ pathList;
 ℋ = HistogramDistribution[fAT];

 (* The continuously sampled case for the same parameters *)
 M1 = M1Continuous[S0, r, T];
 M2 = M2Continuous[S0, r, σ, T];
 μA = 1/2 (4 Log[M1] - Log[M2]);
 σA = Sqrt[-2 Log[M1] + Log[M2]];
 𝒟 = LogNormalDistribution[μA, σA];

 G[1] = DiscretePlot[PDF[ℋ, A],
   {A, Min[fAT], Max[fAT]}, ExtentSize → Full, PlotStyle → Blue,
   PlotLabel → Stl["discretely sampled simulated distribution (blue)\nmoment matched
       continuous sampling (black)"], AxesLabel → {Stl["A_T"], Stl["f(A_T)"]}];
 G[2] = Plot[PDF[𝒟, A], {A, Min[fAT], Max[fAT]}];
 Show[{G[1], G[2]}]]
```

Out[37]=



Agreement is remarkable, considering there were only 12 monthly samples per price path.

3 Moment matching for a discretely sampled average

I now consider the case of a discrete average with sample time $\Delta t = \frac{T}{n}$. See Appendix B for details of the moment calculation.

$$
\begin{aligned}
\mathbf{M1} &= \mathbb{E}[A] \\
&= \mathbb{E}\left[\frac{1}{n+1}\sum_{i=0}^{n} S_i\right] \\
&= \frac{S_0}{(1+n)}\,\frac{\left(-1+e^{r\,\Delta\mathrm{t}+n\,r\,\Delta\mathrm{t}}\right)}{\left(-1+e^{r\,\Delta\mathrm{t}}\right)}
\end{aligned}
$$

In[38]:=

```
Clear[M1Discrete];
M1Discrete[S0_, r_, Δt_, n_] := S0/(n + 1) (Exp[(n + 1) r Δt] - 1)/(Exp[r Δt] - 1)
```

The discrete second moment is

$$
\begin{aligned}
\mathbb{E}\left[A^2\right] &= \mathbb{E}\left[\left(\frac{1}{n+1}\sum_{i=0}^{n} S_i\right)\left(\frac{1}{n+1}\sum_{j=0}^{n} S_j\right)\right] \\
&= \left(\left(-1-x+2\,x^{1+n}+x\,y+x^2\,y-2\,x^{3+n}\,y-x^{2+2\,n}\,y^{1+n}+x^{3+2\,n}\,y^{1+n}-x^{3+2\,n}\,y^{2+n}+x^{4+2\,n}\,y^{2+n}\right)S_0^2\right)\Big/ \\
&\qquad \left((1+n)^2\,(-1+x)\,(-1+x\,y)\left(-1+x^2\,y\right)\right)\Big]
\end{aligned}
$$

where $x = e^{r\,t}$ and y = $e^{\sigma^2\,\Delta\mathrm{t}}$. See Appendix B for the calculation

In[40]:=

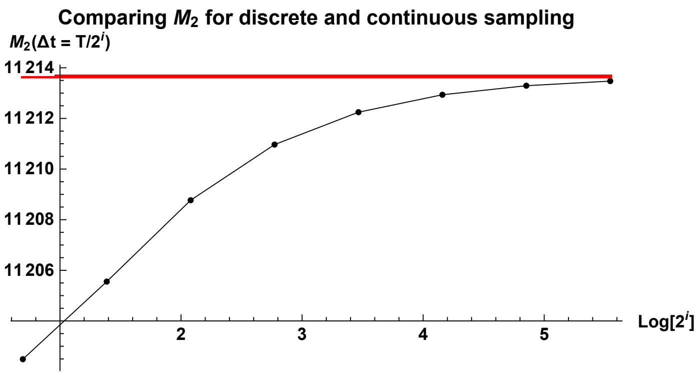
```
Clear[M2Discrete];
M2Discrete[S0_, r_, σ_, T_, n_] :=
 Module[{x, y, Δt = T/n, doubleSum, M2},
  x = Exp[r Δt];
  y = Exp[σ^2 Δt];
  doubleSum =
   (-1 - x + 2 x^(1+n) + x y + x^2 y - 2 x^(3+n) y - x^(2+2 n) y^(1+n) + x^(3+2 n) y^(1+n) - x^(3+2 n) y^(2+n) + x^(4+2 n) y^(2+n)) /
    ((-1 + x) (-1 + x y) (-1 + x^2 y));
  M2 = (S0/(n + 1))^2 doubleSum]
```

I compare the discrete sampling result with continuous sampling.

In[43]:=

```
Module[{S0 = 100, r = 0.1, σ = 0.2, T = 1, nDoub = 8, vals, M2Cont},
 M2Cont = M2Continuous[S0, r, σ, T];
 vals = Table[{Log[2^i], M2Discrete[S0, r, σ, T, 2^i]}, {i, 1, nDoub}];
 Graphics[{{Line[vals], {PointSize[0.01], Point /@ vals},
   {Directive[Thick, Red], Line[{{Log[2], M2Cont}, {Log[2^nDoub], M2Cont}}]}}},
  Axes → Automatic, AspectRatio → 0.5,
  PlotLabel → Stl["Comparing M2 for discrete and continuous sampling"],
  AxesLabel → {Style["Log[2^i]", Bold], Style["M2(Δt = T/2^i)", Bold]}]]
```

Out[43]=



The black line represents M_2 for discrete sampling. The sampling interval is $\Delta t = \frac{T}{n}$, with n = 2^i . The abscissa is scaled logarithmically so that 8 doublings of the sample frequency can be accommodated. The red line represents M_2 for continuous sampling. For a sampling frequency of 2^8 = 256, the discrete value for M_2 is quite close to the result for continuous sampling.

Appendix A - Moment calculations for continuously sampled average

First moment calculation: The symbols 𝔼 (for expectation) and Int (for integration) are abstract operators that I manipulate with rules

In[44]:=

$$m1[1] = \mathbb{E}\left[\frac{1}{T}\text{Int}[A[u], \{u, 0, T\}]\right] \;/.\; A[u] \to S_0\,\text{Exp}\left[\left(r - \frac{\sigma^2}{2}\right) u + \sigma\, z[u]\right]$$

Out[44]=

$$\mathbb{E}\left[\frac{\text{Int}\left[e^{u\left(r-\frac{\sigma^2}{2}\right)+\sigma z[u]}\, S_0, \{u, 0, T\}\right]}{T}\right]$$

Interchange expectation and integration operators

In[45]:=

```
m1[2] = m1[1] /. 𝔼[ a_. Int[b_, c_]] → a Int[𝔼[b], c]
```

Out[45]=

$$\frac{1}{T}\text{Int}\left[\mathbb{E}\left[e^{u\left(r-\frac{\sigma^2}{2}\right)+\sigma z[u]}\, S_0\right], \{u, 0, T\}\right]$$

Make the expectation explicit

In[46]:=
```
m1[3] = m1[2] /. 𝔼[a_] → Expectation[a, z ≈ WienerProcess[0, 1]]
```

Out[46]=
$$\frac{\text{Int}\left[e^{r\,u} S_0,\ \{u,\ 0,\ T\}\right]}{T}$$

Perform the integral

In[47]:=
```
m1[4] = m1[3] /. Int → Integrate
```

Out[47]=
$$\frac{\left(-1+e^{r\,T}\right) S_0}{r\,T}$$

Second moment calculation

In[48]:=
```
m2[1] =
  𝔼[1/T² Int[A[u] A[v], {u, 0, T}, {v, 0, T}]] /. A[x_] → S₀ Exp[(r - σ²/2) x + σ z[x]]
```

Out[48]=
$$\mathbb{E}\left[\frac{1}{T^2}\text{Int}\left[e^{u\left(r-\frac{\sigma^2}{2}\right)+v\left(r-\frac{\sigma^2}{2}\right)+\sigma\, z[u]+\sigma\, z[v]}\, S_0^2,\ \{u,\ 0,\ T\},\ \{v,\ 0,\ T\}\right]\right]$$

In[49]:=
```
m2[2] = m2[1] /. 𝔼[ a_. Int[b_, c_, d_]] → a Int[𝔼[b], c, d] // Simplify
```

Out[49]=
$$\frac{1}{T^2}\text{Int}\left[\mathbb{E}\left[e^{\frac{1}{2}(u+v)\left(2\,r-\sigma^2\right)+\sigma\, z[u]+\sigma\, z[v]}\, S_0^2\right],\ \{u,\ 0,\ T\},\ \{v,\ 0,\ T\}\right]$$

The expectation calculation is simplified if I invoke an identity

In[50]:=
```
m2[3] =
  m2[2] /. 𝔼[c_. Exp[a_. + b_]] /; FreeQ[a, z] → c Exp[a] Exp[𝔼[b²]/2] // ExpandAll
```

Out[50]=
$$\frac{1}{T^2}\text{Int}\left[e^{r\,u+r\,v-\frac{u\,\sigma^2}{2}-\frac{v\,\sigma^2}{2}+\frac{1}{2}\mathbb{E}\left[\sigma^2\, z[u]^2+2\,\sigma^2\, z[u]\, z[v]+\sigma^2\, z[v]^2\right]}\, S_0^2,\ \{u,\ 0,\ T\},\ \{v,\ 0,\ T\}\right]$$

Distribute the expectation

In[51]:=
```
m2[4] = m2[3] //. 𝔼[a_ + b_] → 𝔼[a] + 𝔼[b]
```

Out[51]=
$$\frac{1}{T^2}\text{Int}\left[e^{r\,u+r\,v-\frac{u\,\sigma^2}{2}-\frac{v\,\sigma^2}{2}+\frac{1}{2}\left(\mathbb{E}\left[\sigma^2\, z[u]^2\right]+\mathbb{E}\left[2\,\sigma^2\, z[u]\, z[v]\right]+\mathbb{E}\left[\sigma^2\, z[v]^2\right]\right)}\, S_0^2,\ \{u,\ 0,\ T\},\ \{v,\ 0,\ T\}\right]$$

Perform the expectation

In[52]:=

```
m2[5] = m2[4] /. 𝔼[a_ z[x_]^2] → Expectation[a z[x]^2, z ≈ WienerProcess[0, 1]] /.
   𝔼[a_ z[x_] z[y_]] → a CovarianceFunction[ WienerProcess[0, 1], x, y] // Simplify
```

Out[52]=

$$\frac{1}{T^2}\text{Int}\left[e^{r\,(u+v)+\sigma^2\,\text{Min}[u,v]}\,S_0^2,\ \{u,0,T\},\ \{v,0,T\}\right]$$

Mathematica takes a loong time to value such an integral. I speed things up by spitting the v integration into two parts 0 < v < u and u < v < T

In[53]:=

```
part[1] = (m2[5] /.
   Int[a_, {u, 0, T}, {v, 0, T}] :> Integrate[(a /. Min[u, v] → v), {v, 0, u}])
```

Out[53]=

$$\frac{e^{r\,u}\left(-1+e^{u\,(r+\sigma^2)}\right)S_0^2}{T^2\left(r+\sigma^2\right)}$$

In[54]:=

```
part[2] = (m2[5] /.
   Int[a_, {u, 0, T}, {v, 0, T}] :> Integrate[(a /. Min[u, v] → u), {v, u, T}])
```

Out[54]=

$$\frac{e^{u\,\sigma^2}\left(-e^{2\,r\,u}+e^{r\,(T+u)}\right)S_0^2}{r\,T^2}$$

Finally,

In[55]:=

```
m2[6] = Integrate[part[1] + part[2], {u, 0, T}]
```

Out[55]=

$$\left(2\left(\left(1-2\,e^{r\,T}+e^{T\,(2\,r+\sigma^2)}\right)r-\left(-1+e^{r\,T}\right)\sigma^2\right)S_0^2\right)\Big/\left(r\,T^2\left(r+\sigma^2\right)\left(2\,r+\sigma^2\right)\right)$$

Appendix B - Moment calculations for discretely sampled average

$$\begin{aligned}
\mathbf{M1} &= \mathbb{E}[A]\\
&= \mathbb{E}\left[\frac{1}{n+1}\sum_{i=0}^{n} S_i\right]\\
&= \frac{S_0}{n}\,\frac{\left(e^{(n+1)\,r\,\Delta t}-1\right)}{\left(e^{r\,\Delta t}-1\right)}
\end{aligned}$$

Define the expression

In[58]:=

$$\mathbf{m1B[1]} = \mathbb{E}\left[\frac{1}{n+1}\,\mathrm{Sm}[S_i,\ \{i,\ 0,\ n\}]\right] \text{ /. } S_i \to S_0\,\mathrm{Exp}\left[\left(r-\frac{\sigma^2}{2}\right) i\,\Delta t+\sigma\,z[i\,\Delta t]\right]$$

Out[58]=

$$\mathbb{E}\left[\frac{1}{1+n}\mathrm{Sm}\left[e^{i\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+\sigma\,z[i\,\Delta t]}\,S_0,\ \{i,\ 0,\ n\}\right]\right]$$

Interchange expectation and summation

In[59]:=

$$\mathbf{m1B[2]} = \mathbf{m1B[1]} \text{ /. } \mathbb{E}[a\_.\,\mathrm{Sm}[b\_,\ c\_]] \to a\,\mathrm{Sm}[\mathbb{E}[b],\ c]$$

Out[59]=

$$\frac{1}{1+n}\mathrm{Sm}\left[\mathbb{E}\left[e^{i\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+\sigma\,z[i\,\Delta t]}\,S_0\right],\ \{i,\ 0,\ n\}\right]$$

Make the expectation explicit

In[60]:=

$$\mathbf{m1B[3]} = \mathbf{m1B[2]} \text{ /. } \mathbb{E}[a\_] :> \mathrm{Expectation}[a,\ z \approx \mathrm{WienerProcess}[0,\ 1]]$$

Out[60]=

$$\frac{\mathrm{Sm}\left[e^{i\,r\,\Delta t}\,S_0,\ \{i,\ 0,\ n\}\right]}{1+n}$$

Evaluate the sum

In[61]:=

$$\mathbf{m1B[4]} = \mathbf{m1B[3]} \text{ /. } \mathrm{Sm} \to \mathrm{Sum}$$

Out[61]=

$$\frac{\left(-1+e^{r\,\Delta t+n\,r\,\Delta t}\right)\,S_0}{\left(-1+e^{r\,\Delta t}\right)\,(1+n)}$$

Second moment calculation.

In[62]:=

$$\mathbf{m2B[1]} = \mathbb{E}\left[\left(\mathrm{Sm}\left[\frac{S_i}{n+1}\,\frac{S_j}{n+1},\ \{i,\ 0,\ n\},\ \{j,\ 0,\ n\}\right]\right)\right]$$

Out[62]=

$$\mathbb{E}\left[\mathrm{Sm}\left[\frac{S_i\,S_j}{(1+n)^2},\ \{i,\ 0,\ n\},\ \{j,\ 0,\ n\}\right]\right]$$

In[63]:=

$$\mathbf{m2B[2]} = \mathbf{m2B[1]} \text{ /. } S_{i\_} \to S_0\,\mathrm{Exp}\left[\left(r-\frac{\sigma^2}{2}\right) i\,\Delta t+\sigma\,z[i\,\Delta t]\right]$$

Out[63]=

$$\mathbb{E}\left[\mathrm{Sm}\left[\frac{1}{(1+n)^2}e^{i\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+j\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+\sigma\,z[i\,\Delta t]+\sigma\,z[j\,\Delta t]}\,S_0^2,\ \{i,\ 0,\ n\},\ \{j,\ 0,\ n\}\right]\right]$$

In[64]:=

$$\mathbf{m2B[3]} = \mathbf{m2B[2]} \text{ /. } \mathbb{E}[\mathrm{Sm}[a\_,\ b\_,\ c\_]] \to \mathrm{Sm}[\mathbb{E}[a],\ b,\ c]$$

Out[64]=

$$\mathrm{Sm}\left[\mathbb{E}\left[\frac{1}{(1+n)^2}e^{i\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+j\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+\sigma\,z[i\,\Delta t]+\sigma\,z[j\,\Delta t]}\,S_0^2\right],\ \{i,\ 0,\ n\},\ \{j,\ 0,\ n\}\right]$$

In[65]:=
```
m2B[4] = m2B[3] /. 𝔼[a_ Exp[b_ + c_]] /; FreeQ[c, z] → a Exp[c] 𝔼[Exp[b]]
```

Out[65]=

$$\text{Sm}\left[\frac{1}{(1+n)^2} e^{i\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)+j\,\Delta t\,\left(r-\frac{\sigma^2}{2}\right)}\, S_0^2\, \mathbb{E}\left[e^{\sigma\, z[i\,\Delta t]+\sigma\, z[j\,\Delta t]}\right],\ \{i,0,n\},\ \{j,0,n\}\right]$$

In[66]:=
```
m2B[5] = m2B[4] /. 𝔼[Exp[a_]] → Exp[𝔼[a²/2]] // ExpandAll
```

Out[66]=

$$\text{Sm}\left[\left(e^{i\, r\,\Delta t+j\, r\,\Delta t-\frac{1}{2} i\,\Delta t\,\sigma^2-\frac{1}{2} j\,\Delta t\,\sigma^2+\mathbb{E}\left[\frac{1}{2}\sigma^2 z[i\,\Delta t]^2+\sigma^2 z[i\,\Delta t]\, z[j\,\Delta t]+\frac{1}{2}\sigma^2 z[j\,\Delta t]^2\right]}\, S_0^2\right)\Big/\left(1+2\,n+n^2\right),\ \{i,0,n\},\ \{j,0,n\}\right]$$

In[67]:=
```
m2B[6] = m2B[5] //. 𝔼[a_ + b_] → 𝔼[a] + 𝔼[b]
```

Out[67]=

$$\text{Sm}\left[\left(e^{i\, r\,\Delta t+j\, r\,\Delta t-\frac{1}{2} i\,\Delta t\,\sigma^2-\frac{1}{2} j\,\Delta t\,\sigma^2+\mathbb{E}\left[\frac{1}{2}\sigma^2 z[i\,\Delta t]^2\right]+\mathbb{E}\left[\sigma^2 z[i\,\Delta t]\, z[j\,\Delta t]\right]+\mathbb{E}\left[\frac{1}{2}\sigma^2 z[j\,\Delta t]^2\right]}\, S_0^2\right)\Big/\left(1+2\,n+n^2\right),\ \{i,0,n\},\ \{j,0,n\}\right]$$

In[68]:=
```
m2B[7] =
 m2B[6] /. 𝔼[a_ z[i_ Δt]²] → Expectation[a z[i Δt]², z ≈ WienerProcess[0, 1]] /.
   𝔼[a_ z[x_] z[y_]] → a CovarianceFunction[ WienerProcess[0, 1], x, y] // Simplify
```

Out[68]=

$$\text{Sm}\left[\frac{e^{(i+j)\, r\,\Delta t+\sigma^2\,\text{Min}[i\,\Delta t,j\,\Delta t]}\, S_0^2}{(1+n)^2},\ \{i,0,n\},\ \{j,0,n\}\right]$$

It’s worthwhile to simplify the notation before performing the integration

In[69]:=
```
rRule = Sol[x == Exp[r Δt], r] /. C[1] → 0
```

Out[69]=

$$r\to\frac{\text{Log}[x]}{\Delta t}$$

In[70]:=
```
σRule = Solve[y == Exp[σ² Δt], σ]⟦2, 1⟧ /. C[1] → 0
```

Out[70]=

$$\sigma\to\frac{\sqrt{\text{Log}[y]}}{\sqrt{\Delta t}}$$

In[71]:=
```
m2B[8]  = m2B[7] /. Min[i Δt, j Δt] → Δt Min[i, j] /. rRule /. σRule // Simplify
```

Out[71]=

$$\text{Sm}\left[\frac{x^{i+j}\, y^{\text{Min}[i,j]}\, S_0^2}{(1+n)^2},\ \{i,0,n\},\ \{j,0,n\}\right]$$

For some fun, try calculating this sum by hand. *Pricing european average rate currency options,* Edward Levy (1992). In the late 90’s it took a couple of days to establish confidence in Levy’s results. Mathematica calculates the sum in a few seconds.

In[72]:=

```
m2B[9] = m2B[8] /. Sm → Sum
```

Out[72]=

$$\begin{cases} \frac{S_0^2}{(1+n)^2} & n == 0 \\ \left(\left(-1 - x + 2\,x^{1+n} + x\,y + x^2\,y - 2\,x^{3+n}\,y - x^{2+2\,n}\,y^{1+n} + x^{3+2\,n}\,y^{1+n} - x^{3+2\,n}\,y^{2+n} + x^{4+2\,n}\,y^{2+n}\right) S_0^2\right) \Big/ \left((1+n)^2\,(-1+x)\,(-1+x\,y)\,(-1+x^2\,y)\right) & \text{True} \end{cases}$$

In[73]:=

```
m2B[10] = Refine[m2B[9], n > 0]
```

Out[73]=

$$\left(\left(-1 - x + 2\,x^{1+n} + x\,y + x^2\,y - 2\,x^{3+n}\,y - x^{2+2\,n}\,y^{1+n} + x^{3+2\,n}\,y^{1+n} - x^{3+2\,n}\,y^{2+n} + x^{4+2\,n}\,y^{2+n}\right) S_0^2\right) \Big/ \left((1+n)^2\,(-1+x)\,(-1+x\,y)\,(-1+x^2\,y)\right)$$

Appendix C Simulation speed

Just out of curiosity, I perform a simple speed test that compares simulation using RandomFunction of GeometricBrownianMotionProcess to methods I have used in the past. RandomFunction actually seems to be faster than the path generators I used in the past.

In[74]:=

```
Module[{S0 = 100, r = 0.02, q = 0, σ = 0.2,
  T = 1.0, ΔT = 1/52, nPaths = 100000, pathList, GBM},
 SeedRandom[123];
 GBM = GeometricBrownianMotionProcess[(r - q), σ, S0];
 pathList = Timing[RandomFunction[GBM, {0, T, ΔT}, nPaths]];
 pathList 〚1〛]
```

Out[74]=

```
0.1875
```

In[75]:=

```
GBMPathCompiled =
  Compile[{{S0, _Real}, {drift, _Real}, {diff, _Real}, {nSteps, _Integer}},
   FoldList[(#1 drift Exp[ diff #2]) &, S0,
    RandomVariate[NormalDistribution[0, 1], nSteps]]];
```

In[76]:=

```
Module[{S0 = 100, r = 0.02, q = 0, σ = 0.2, T = 1.0,
  ΔT = 1/52, nSteps = 52, nPaths = 100000, drift, diff, pathList, GBM},
 SeedRandom[123];
 drift = Exp[(r - q - σ^2/2) ΔT];
 diff = σ √ΔT;
 pathList = Timing[Table[GBMPathCompiled[S0, drift, diff, nSteps], {nPaths}]];
 pathList 〚1〛]
```

Out[76]=

```
0.46875
```

In[15]:=

```
Clear[GBMPath];
GBMPath[S0_, drift_, diff_, nSteps_] :=
 FoldList[(#1 drift Exp[ diff #2]) &,
  S0, RandomVariate[NormalDistribution[0, 1], nSteps]]
```

In[77]:=

```
Module[{S0 = 100, r = 0.02, q = 0, σ = 0.2, T = 1.0,
  ΔT = 1/52, nSteps = 52, nPaths = 100 000, drift, diff, pathList, GBM},
 SeedRandom[123];
 drift = Exp[(r - q - σ^2/2) ΔT];
 diff = σ √ΔT ;
 pathList = Timing[Table[GBMPath[S0, drift, diff, nSteps], {nPaths}]];
 pathList 〚1〛 ]
```

Out[77]=

```
6.375
```