Uniswap V3 Pricing Review for Lenders

Abstract

With the creation of Uniswap, thorough stochastic pricing analysis has been reviewed by Bardoscia and Milionis describing it’s spot pricing dynamics. Since it’s release, only a few protocols have been created to address leveraged perpetual options using Uniswap V3 pricing dynamics, effectively creating a “loan” based on backed collateral for position holders. Below is a review of simulation using Mathematica to generate empirical risk profiles for V3 positions with standard techniques in Stochastic Calculus.

Stochastic Calculus Review

Ito’s Lemma

Single Variable Ito’s Lemma

Out[136]=

14.2.3 Ito’s Lemma

The central tool in stochastic differential equations is **Ito’s lemma**, which basically says that a smooth function of an Ito process is itself an Ito process.

THEOREM 14.2.1 *Suppose that* $f : R \to R$ *is twice continuously differentiable*[1] *and that* $dX = a_t\,dt + b_t\,dW$. *Then* $f(X)$ *is the Ito process*

$$f(X_t) = f(X_0) + \int_0^t f'(X_s)\,a_s\,ds + \int_0^t f'(X_s)\,b_s\,dW + \frac{1}{2}\int_0^t f''(X_s)\,b_s^2\,ds$$

for $t \geq 0$.

In differential form, Ito’s lemma becomes

$$df(X) = f'(X)\,a\,dt + f'(X)\,b\,dW + \frac{1}{2}\,f''(X)\,b^2\,dt. \tag{14.9}$$

In[49]:=
```
SVIto[F_, var_, stocs_] := Sum[i[[2]] * D[F, var] * 𝕕i[[1]], {i, {{t, α}, {W, β}}}] +
    1/2 β^2 * D[F, {var, 2}] * 𝕕t /. stocs // Simplify
```

Multivariable Ito’s Lemma

Out[50]=

THEOREM 14.2.3 *Let $W_1, W_2, \ldots, W_m$ be Wiener processes and let $X \equiv (X_1, X_2, \ldots, X_m)$ be a vector process. Suppose that $f : R^m \to R$ is twice continuously differentiable and X_i is an Ito process with $dX_i = a_i\, dt + b_i\, dW_i$. Then $df(X)$ is the following Ito process,*

$$df(X) = \sum_{i=1}^{m} f_i(X)\, dX_i + \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{m} f_{ik}(X)\, dX_i\, dX_k,$$

with the following multiplication table:

$\times$	dW_i	dt
dW_k	$\rho_{ik}\, dt$	0
dt	0	0

Here, ρ_{ik} denotes the correlation between dW_i and dW_k.

In[51]:=
```
MVIto[F_, vars_, stocs_] :=
  Sum[D[F, {var}] * 𝕕var, {var, vars}] +
    Sum[1/2 D[F, pairs[[1]], pairs[[2]]] * 𝕕pairs[[1]] * 𝕕pairs[[2]],
     {pairs, Subsets[vars, {2}] ~Join~ Map[{#, #} &, vars]}] /. stocs // Simplify;
```

Sanity Checks

The Single Variable Ito should correctly return GBM.

In[52]:=
```
SVIto @@ {ⅇ^S, S, {𝕕S → μ 𝕕t + σ 𝕕W, α → μ , β → σ}} // (#/ⅇ^S) & // Simplify
```

Out[52]= $\left(\mu + \frac{\sigma^2}{2}\right) dt + \sigma\, dW$

The Multi-variable Ito should correctly return the Forward Contract process.

In[53]:=
```
MVIto @@ {S * ⅇ^(y (T-t)), {S, t}, {𝕕S → μ S 𝕕t + σ S 𝕕W}}
```

Out[53]= $\frac{1}{2} e^{(-t+T)\, y}\, S\, (-2 + y\, dt)\, ((y - \mu)\, dt - \sigma\, dW)$

In[54]:=

The Multi-variable Ito should correctly return the Log Normal process.

```
In[55]:= MVIto @@ { Log[S], {S, t}, {dS → μ S dt + σ S dW}} //
           Expand //
         (# /. {(dt)^2 → 0, (dW)^2 → dt , dt dW → 0}) & //
          Simplify
```

Out[55]= $\left(\mu - \frac{\sigma^2}{2}\right) dt + \sigma\, dW$

Geometric Brownian Motion

Geometric Brownian Motion is a process that assumes random percent changes. Rather than random step changes, this has features where the value is not negative and is modeled in lots of natural processes.

In[56]:= diffGBM = $dS \to S\left(\mu + \frac{\sigma^2}{2}\right) dt + S\,\sigma\, dW$;

Ornstien-Uhlenbeck

Ornstien-Uhlenbeck processes drive to the mean μ as time goes on. This is in effect a mean reverting process model.

In[57]:= diffOrnstienUhlenbeck = $dX \to \kappa\,(\mu - X)\, dt + \sigma\, dW$;

In[58]:= SVIto[X^2, X, {α → −κ * X, β → σ}]

Out[58]= $\left(-2 X^2 \kappa + \sigma^2\right) dt + 2 X \sigma\, dW$

```
In[59]:= MVIto[X^2, {X, t}, {dX → - κ * X dt + σ dW}] //
           Expand //
         (# /. {(dt)^2 → 0, (dW)^2 → dt , dt dW → 0}) & //
          Simplify
```

Out[59]= $\left(-2 X^2 \kappa + \sigma^2\right) dt + 2 X \sigma\, dW$

Uniswap V3 Stochastic Analysis

Value of a Uniswap V3 Position

Uniswap State Equations

Pricing Derivations for Impermanent Loss

In[66]:=
```
ethDailyVol = 0.0034;
ethMeanYearly = 0.1;
currentPrice = 1628;
lowerBound = 1600;
upperBound = 1700;
initialValue = 10 000;
```

In[109]:=
```
currentLiquidityParams =
  Liquidity[lowerBound, upperBound, currentPrice, initialValue] // N
```

Out[109]=
```
{{x → 4.37661, y → 2874.87, L → 8249.71}}
```

In[110]:=
```
originalValue =
   x * p + y /. tokensGivenLiquidity /. {pa → lower, pb → higher, p → startPrice};
currentValue = x * p + y /. tokensGivenLiquidity /. {pa → lower, pb → higher};
```

$$\text{IL} = \frac{\text{currentValue} - \text{originalValue}}{\text{originalValue}};$$

```
humanReadable = {lower → pa, higher → pb , startPrice → p0};
```

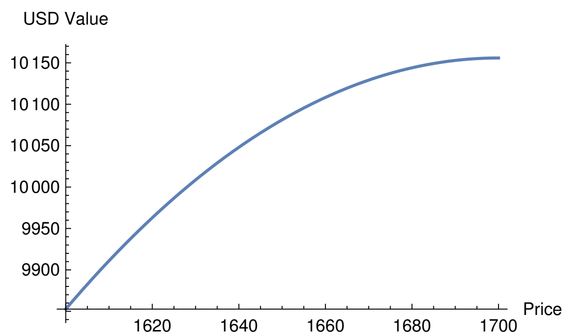
In[77]:=
```
IL /. humanReadable // Simplify;
```

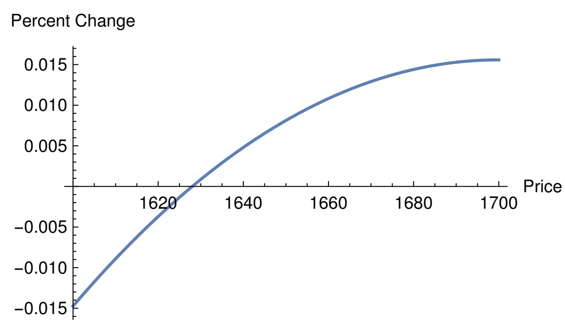
Plotting to Understand Value Curves

In[78]:=
```
valueCurve = currentValue  /. currentLiquidityParams;
```

In[79]:=
```
GraphicsGrid[{{
   Plot[valueCurve /.
     {lower → lowerBound, higher → upperBound , startPrice → currentPrice},
    {p, lowerBound, upperBound}, AxesLabel → {"Price", "USD Value"}],
   Plot[
    IL /. {lower → lowerBound, higher → upperBound , startPrice → currentPrice},
    {p, lowerBound, upperBound}, AxesLabel → {"Price", "Percent Change"}]
  }}]
```

Out[79]=





Pricing With IL

In[80]:=
```
resultIL = MVIto @@ {IL, {p, t}, {(diffGBM /. {S → p})}} //
     Expand //
    (# /. {(dt)^2 → 0, (dW)^2 → dt , dt dW → 0}) & //
   Simplify;
resultIL /. humanReadable
```

Out[81]=
$$\frac{dW\left(4\,p\,\sigma-4\,\sqrt{p}\,\sigma\,\sqrt{p_b}\right)+dt\left(2\,p\left(2\,\mu+\sigma^2\right)-\sqrt{p}\left(4\,\mu+\sigma^2\right)\sqrt{p_b}\right)}{4\left(p_0+\left(-2\,\sqrt{p_0}+\sqrt{p_a}\right)\sqrt{p_b}\right)}$$

In[82]:=
```
preprocessIL = resultIL /. currentLiquidityParams[[1]] /.
    {p → p[t], W → W[t], lower → lowerBound, higher → upperBound , μ → ethMeanYearly/365,
     σ → ethDailyVol, startPrice → currentPrice} // N // Simplify
```

Out[82]=
$$0.000228404\,dt\,\sqrt{p[t]}+0.0028049\,dW[t]\,\sqrt{p[t]}-5.59742\times10^{-6}\,dt\,p[t]-0.0000680288\,dW[t]\times p[t]$$

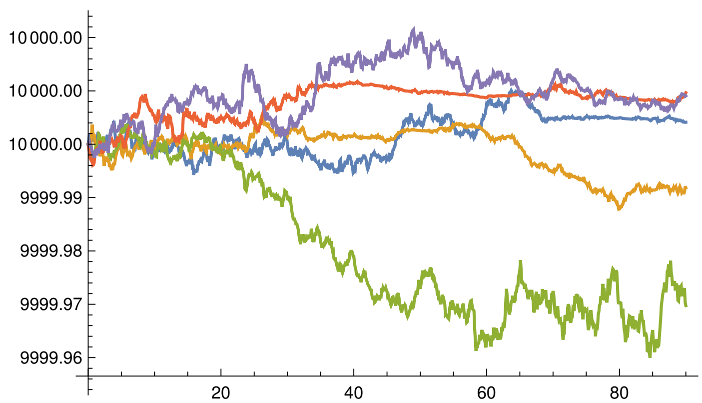
In[83]:=
```
procIL = ItoProcess[dV[t] == preprocessIL,
   V[t], {V, 10000}, {t, 0}, {W ≈ WienerProcess[], p ≈
     GeometricBrownianMotionProcess[ethMeanYearly/365, ethDailyVol, currentPrice]}];
fsIL = RandomFunction[procIL, {0, 90}, 5];
```

In[85]:=
```
Mean[fsIL[90]]
```

Out[85]= 10000.

In[108]:=
```
Show[{
  ListLinePlot[fsIL, FillingStyle → Axis]
 }]
```

Out[108]=



Pricing with Position Value

In[129]:=
```
preprocessPV =
  currentValue /. {lower → lowerBound, higher → upperBound , p → p[t]} /.
   currentLiquidityParams;
```

In[130]:=
```
preprocessPV
```

Out[130]=

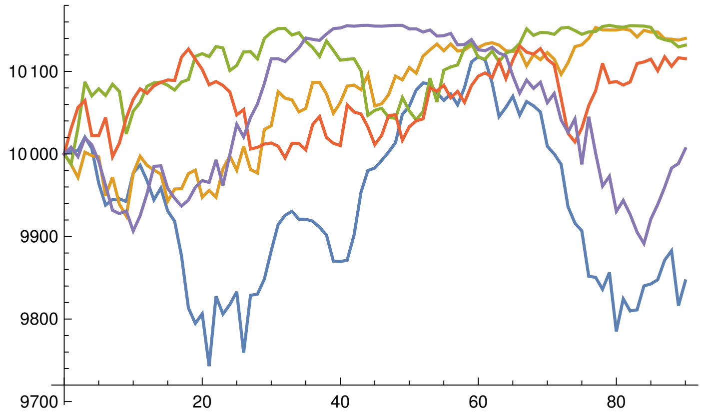
$$\left\{8249.71\left(-40+\sqrt{p[t]}\right)+200.085\left(10\sqrt{17}-\sqrt{p[t]}\right)\sqrt{p[t]}\right\}$$

In[131]:=
```
procPV = TransformedProcess[preprocessPV, {p ≈
     GeometricBrownianMotionProcess[ethMeanYearly/365, ethDailyVol, currentPrice]},
   t];
fsPV = RandomFunction[procPV, {0, 90, 1}, 5];
```

In[134]:=
```
Mean[fsPV[90]]
```
Out[134]=

10 048.

In[135]:=
```
ListLinePlot[fsPV, FillingStyle → Axis]
```
Out[135]=



Further Research

Out[94]//TableForm=

Impermanent Loss in Uniswap V3	https://lambert-guillaume.medium.com/an-analysis-of-impermanent-loss-in-uniswap-bfbfebbefed2
Uniswap Liquidity V3 Math	http://atiselsts.github.io/pdfs/uniswap-v3-liquidity-

Perpetual Lending Stochastic Analysis

Mean-Reverting Additional Value Term

TBD