QuantMinds International

Intercontinential O2 London, November 17, 2025

Rough volatility workshop

Lecture 2: Rough volatility models

Jim Gatheral
Department of Mathematics

Baruch COLLEGE
The City University of New York

Outline of Lecture 2

- The forward variance curve
- Change of measure
- The rough Bergomi model
- The rough Heston model
- The quadratic rough Heston model

A simplifying assumption

- We will set interest rates and dividends to zero (unless we specifically say otherwise).
  - It is typically easy to reintroduce nonzero rates and dividends - but of course everything get's more complicated.
- With this assumption
  - The stock price process is a (local) martingale, and so has zero drift.
  - The drift of the stock price process under $\mathbb{P}$ is the *equity risk premium* - the extra return that investors require for holding the risky stock.

Stochastic volatility

- Under the pricing measure $\mathbb{Q}$, stochastic volatility models take the form

$$\frac{dS_t}{S_t} = \sqrt{V_t}\, dZ_t$$

with $V_t = \frac{d}{dt}\langle \log S \rangle_t$.

- Thus, the stock price process and the quadratic variation process are both assumed continuous.

  - There are no jumps!
- To ensure no-arbitrage, the stock price S is modeled as a positive semimartingale.
  - This excludes for example fractional Brownian motion with $H \neq \frac{1}{2}$.
    - If $H > 1/2$, quadratic variation (QV) vanishes; if $H < 1/2$, QV is infinite!

Forward variance models

- In classical models, such as Black-Scholes and classical Heston, the volatility process is modeled directly.
  - However V_t is not observable.
- Bergomi and Guyon][3], suggested that it is natural to model forward variances

$$\xi_t(u) = \mathbb{E}_t^{\mathbb{Q}}\left[V_u\right], \quad u > t.$$

- The forward variances, being conditional expectations under $\mathbb{Q}$, are tradable.
  - Not only in principle, but in practice, as forward starting variance swaps.

- The forward variance processes are modeled (in the single-factor case) as

$$d\xi_t(u) = \eta_t(u;\omega)\, dW_t,$$

  where as before $d\langle W, Z\rangle_t = \rho\, dt$.

- The $\mathbb{R}_{\geqslant 0}$-valued stochastic process $\eta_t(u;\omega)$ is progressively measurable for all $u > 0$.
  - Conventionally, it is assumed that η is adapted to the filtration generated by W. In other words, the variance process depends only on the history of the variance process.
  - A truly path-dependent model would have η adapted to the filtration jointly generated by W and Z.

- If V is continuous and uniformly integrable, we can recover V_t from $\xi_t(u)$ as $V_t = \lim_{u \downarrow t} \xi_t(u)$.
  - For our purposes, $V_t = \xi_t(t)$.
- The initial conditions of a typical forward variance model are the initial stock price S_t and the initial forward variance curve $\xi_t(u)_{u>t}$.

Forward variance curve models and perfect hedging

- As noted by [El Euch and Rosenbaum][7] , models written in forward variance form are explicitly Markovian in the asset price S_t and the (infinite-dimensional) forward variance curve ξ_t.
  - European payoffs V may be perfectly hedged.
  - The delta-hedging strategy involves holding $\partial_S V$ in the asset and $\delta_\xi V$ in forward variance contracts where δ_ξ denotes the Fréchet derivative of V with respect to the forward variance curve.

Example: The Heston model

- The classical Heston model reads:

$$dV_t = -\lambda\left(V_t - \bar{V}\right) dt + \nu\, \sqrt{V_t}\, dW_t.$$

- The forward variance curve is easily computed as the solution of an ODE:

$$\xi_t(u) := \mathbb{E}_t\left[V_u\right] = (V_t - \bar{V})\, e^{\lambda\,(u-t)} + \bar{V}.$$

- Thus, in forward variance form, classical Heston model reads:

$$d\xi_t(u) = \nu\, e^{\lambda\,(u-t)}\, \sqrt{V_t}\, dW_t.$$

- - The classical Heston model generates a term structure of volatility skew $\mathcal{S}(\tau)$ that is something like

$$\mathcal{S}(\tau) \sim \frac{1}{\lambda\tau}\left\{1 - \frac{1-e^{-\lambda\tau}}{\lambda\tau}\right\}.$$

Example: The Bergomi model

- In Lecture 1, we demonstrated that smiles typically scale as a power-law.
- Partially motivated by this, Bergomi introduced the n-factor Bergomi variance curve model:

(1)

$$\xi_t(u) = \xi_0(u)\; \exp\left\{\sum_{i=1}^n \eta_i\, \int_0^t e^{-\kappa_i\,(u-s)}\, dW_s^{(i)} + \text{ drift }\right\}.$$

- The Bergomi model generates a term structure of volatility skew $\mathcal{S}(\tau)$ that is something like

$$\mathcal{S}(\tau) \sim \sum_i \frac{1}{\kappa_i\,\tau}\left\{1 - \frac{1-e^{-\kappa_i\,\tau}}{\kappa_i\,\tau}\right\}.$$

- This functional form is related to the term structure of the autocorrelation function.
  - Which is in turn driven by the exponential kernel in the exponent in (1).
- To achieve a decent fit to the observed volatility surface, and to control the forward smile, we need at least two factors.
  - In the two-factor case, there are 8 parameters.
- When calibrating, we find that the two-factor Bergomi model is already over-parameterized. Any combination of parameters that gives a roughly $1/\sqrt{T}$ ATM skew fits well enough.
  - Moreover, the calibrated correlations between the Brownian increments $dW_s^{(i)}$ tend to be high.

ATM skew in the Bergomi model

- The Bergomi model generates a term structure of volatility skew $\mathcal{S}(\tau)$ that is something like

$$\mathcal{S}(\tau) = \sum_i \frac{1}{\kappa_i\,\tau}\left\{1 - \frac{1-e^{-\kappa_i\,\tau}}{\kappa_i\,\tau}\right\}.$$

- This functional form is related to the term structure of the autocorrelation function.
  - Which is in turn driven by the exponential kernel in the exponent in (1).

Tinkering with the Bergomi model

- Empirically, $\mathcal{S}(\tau) \sim \tau^{-\alpha}$ for some α.

- It's tempting to replace the exponential kernels in (1) with a power-law kernel.
- This would give a model of the form

$$\xi_t(u) = \xi_0(u)\,\exp\left\{\eta \int_0^t \frac{dW_s}{(u-s)^\gamma} + \text{ drift }\right\}$$

  which looks similar to

$$\xi_t(u) = \xi_0(u)\,\exp\left\{\eta\, W_t^H + \text{ drift }\right\}$$

  where W_t^H is fractional Brownian motion.

Representations of fBm

There are infinitely many possible representations of fBm in terms of Brownian motion. For example, with $\gamma = \frac{1}{2} - H$,

> **Mandelbrot-Van Ness**
>
> $$W_t^H = C_H \left\{ \int_{-\infty}^t \frac{dW_s}{(t-s)^\gamma} - \int_{-\infty}^0 \frac{dW_s}{(-s)^\gamma} \right\}.$$

where the choice

$$C_H = \sqrt{\frac{2\,H\,\Gamma(3/2-H)}{\Gamma(H+1/2)\,\Gamma(2-2\,H)}}$$

ensures that

$$\mathbb{E}\left[W_t^H\,W_s^H\right] = \frac{1}{2}\left\{t^{2H} + s^{2H} - |t-s|^{2H}\right\}.$$

The RFSV model

In Lecture 1, our analysis of realized variance data suggested the following model for volatility under the real (or historical or physical) measure $\mathbb{P}$:

$$\log\sigma_u - \log\sigma_t = \nu\,\left(W_u^H - W_t^H\right), \quad u > t.$$

Then, with the Mandelbrot-Van Ness representation of fBm,

$$\begin{aligned}\log V_u - \log V_t &= 2\,\nu\,C_H \left\{ \int_t^u \frac{1}{(u-s)^\gamma}\,dW_s^{\mathbb{P}} \right.\\ &\quad \left. + \int_{-\infty}^t \left[\frac{1}{(u-s)^\gamma} - \frac{1}{(t-s)^\gamma}\right]\,dW_s^{\mathbb{P}} \right\}\\ &=: 2\,\nu\,C_H\,\left[M_t(u) + Z_t(u)\right].\end{aligned}$$

- Note that $\mathbb{E}_t^{\mathbb{P}}\left[M_t(u)\right] = 0$ and $Z_t(u)$ is $\mathcal{F}_t$-measurable.
  - To price options, it would seem that we would need to know $\mathcal{F}_t$, the entire history of the Brownian motion W_s for $s < t$!

Pricing under $\mathbb{P}$

- Let $\tilde\eta = 2\,\nu\,C_H$ with $\tilde\eta = \eta\,\sqrt{2H}$. Then

$$\begin{aligned}V_u &= V_t \exp\left\{\tilde\eta \int_t^u \frac{1}{(u-s)^\gamma}\,dW_s^{\mathbb{P}} + 2\,\nu\,C_H\,Z_t(u)\right\}\\ &= \mathbb{E}_t^{\mathbb{P}}\left[V_u\right]\,\mathcal{E}\left(\tilde\eta \int_t^u \frac{1}{(u-s)^\gamma}\,dW_s^{\mathbb{P}}\right).\end{aligned}$$

  - The conditional distribution of V_u depends on $\mathcal{F}_t$ only through the variance forecasts $\mathbb{E}_t^{\mathbb{P}}\left[V_u\right]$,
- The last equality is the key:
  - To price options, one does not need to know $\mathcal{F}_t$, the entire history of the Brownian motion $W_s^{\mathbb{P}}$ for $s < t$.

Pricing under $\mathbb{Q}$

- Our model under $\mathbb{P}$ reads:

$$V_u = \mathbb{E}_t^{\mathbb{P}}\left[V_u\right]\,\mathcal{E}\left(\tilde\eta \int_t^u \frac{1}{(u-s)^\gamma}\,dW_s^{\mathbb{P}}\right).$$

- Consider some general change of measure

$$dW_s^{\mathbb{P}} = dW_s^{\mathbb{Q}} + \lambda_s\,ds,$$

  where $\{\lambda_s : s > t\}$ has a natural interpretation as the price of volatility risk. We may then write

$$V_u = \mathbb{E}_t^{\mathbb{P}}\left[V_u\right]\,\mathcal{E}\left(\tilde\eta \int_t^u \frac{1}{(u-s)^\gamma}\,dW_s^{\mathbb{P}}\right)\exp\left\{\tilde\eta \int_t^u \frac{\lambda_s}{(u-s)^\gamma}\,ds\right\}.$$

- Although the conditional distribution of V_u under $\mathbb{P}$ is lognormal, it will not be lognormal in general under $\mathbb{Q}$.
  - The upward sloping smile in VIX options means λ_s cannot be deterministic in this picture.

The rough Bergomi model

Let's nevertheless consider the simplest change of measure

$$dW_s^{\mathbb{P}} = dW_s^{\mathbb{Q}} + \lambda(s)\,ds,$$

where $\lambda(s)$ is a deterministic function of s. Then from (2), we would have

$$\begin{aligned}V_u &= \mathbb{E}_t^{\mathbb{P}}\left[V_u\right]\,\mathcal{E}\left(\eta\,\tilde W_t^{\mathbb{Q}}(u)\right)\exp\left\{\tilde\eta \int_t^u \frac{1}{(u-s)^\gamma}\,\lambda(s)\,ds\right\}\\ &= \xi_t(u)\,\mathcal{E}\left(\eta\,\tilde W_t^{\mathbb{Q}}(u)\right) \qquad (1)\end{aligned}$$

where the forward variances $\xi_t(u) = \mathbb{E}_t^{\mathbb{Q}}\left[V_u\right]$ are (at least in principle) tradable and observed in the market.

- $\xi_t(u)$ is the product of two terms:
- $\mathbb{E}_t^{\mathbb{P}}\left[V_u\right]$ which depends on the historical path $\{W_s, s < t\}$ of the Brownian motion

- a term which depends on the price of risk $\lambda(s)$.

Features of the rough Bergomi model

- The rBergomi model is a non-Markovian generalization of the Bergomi model:

$$\mathbb{E}\left[V_u | \mathcal{F}_t\right] \neq \mathbb{E}[V_u|V_t].$$

  - The rBergomi model is Markovian in the (infinite-dimensional) state vector $\mathbb{E}^{\mathbb{Q}}\left[V_u | \mathcal{F}_t\right] = \xi_t(u)$.

- We have achieved our earlier aim of replacing the exponential kernels in the Bergomi model with a power-law kernel.

- We may therefore expect that the rBergomi model will generate a realistic term structure of ATM volatility skew.

Re-interpretation of the conventional Bergomi model

- A conventional n-factor Bergomi model is not self-consistent for an arbitrary choice of the initial forward variance curve $\xi_t(u)$.

- $\xi_t(u) = \mathbb{E}_t\left[V_u\right]$ should be consistent with the assumed dynamics.

- Viewed from the perspective of the fractional Bergomi model however:
  - The initial curve $\xi_t(u)$ reflects the history $\{W_s; s < t\}$ of the driving Brownian motion up to time t.
  - The exponential kernels in the exponent of the conventional Bergomi model approximate more realistic power-law kernels.

- The conventional two-factor Bergomi model is then justified in practice as a tractable Markovian engineering approximation to a more realistic fractional Bergomi model.

The variance contract and the log-contract.

Under zero interest rates and dividends,applying Itô's Lemma, path-by-path

$$\log\left(\frac{S_T}{S_t}\right) = \int_t^T d\log(S_u) = \int_t^T \frac{dS_u}{S_u} - \frac{1}{2}\int_t^T V_u\, du.$$

- The second term on the RHS is immediately recognizable as half the quadratic variation $\langle X\rangle_T$ over the interval $[0, T]$.

- Thus, the value of the variance contract

$$M_t(T) := \mathbb{E}_t\left[\int_t^T V_u\, du\right] = -2\,\mathbb{E}_t\left[\log\frac{S_T}{S_t}\right],$$

  may be expressed in terms of the fair value of the log-contract.

Robust valuation of the variance contract

- The log-contract may be valued using the Carr-Madan spanning formula as the *log-strip* of options that gives rise to the VIX formula.

- In principle, we need to know the prices of Europeans with all possible strikes for a given expiration T.
  - In practice, we only have a finite number of strike prices listed per expiration.
- One way to estimate the value of such swaps would be to fit a parameterization such as SVI or one of the Vola Dyanamics curves, interpolating and extrapolating to fill in all the other strikes.

- We will now show how to estimate the value of the variance contract robustly with not too much dependence on the interpolation/extrapolation method.

A cool formula

- Write $\Sigma(k,T) = \Sigma$ for short and define

$$d_\pm = -\frac{k}{\sqrt{\Sigma}} \pm \frac{\sqrt{\Sigma}}{2}$$

- Further define the inverse functions $g_\pm(z) = d_\pm^{-1}(z)$.
  - Intuitively, z measures the log-moneyness of an option in implied standard deviations.
- Then,

$$M_t(T) = -2\,\mathbb{E}_t\left[\log\frac{S_T}{S_t}\right] = \int\limits_{-\infty}^{\infty} dz\, N'(z)\,\Sigma\left(g_-(z)\right).$$

- To see this formula is plausible, it is obviously correct in the flat-volatility Black-Scholes case.

Estimating the forward variance curve in practice

- With the above formulae, it's easy to see how to get the forward variance curve in principle.

- Let's now do this in practice.

- Again, we consider the SPX volatility surface as of 15-Feb-2023.

The SPX volatility surface as of 15-Feb-2023

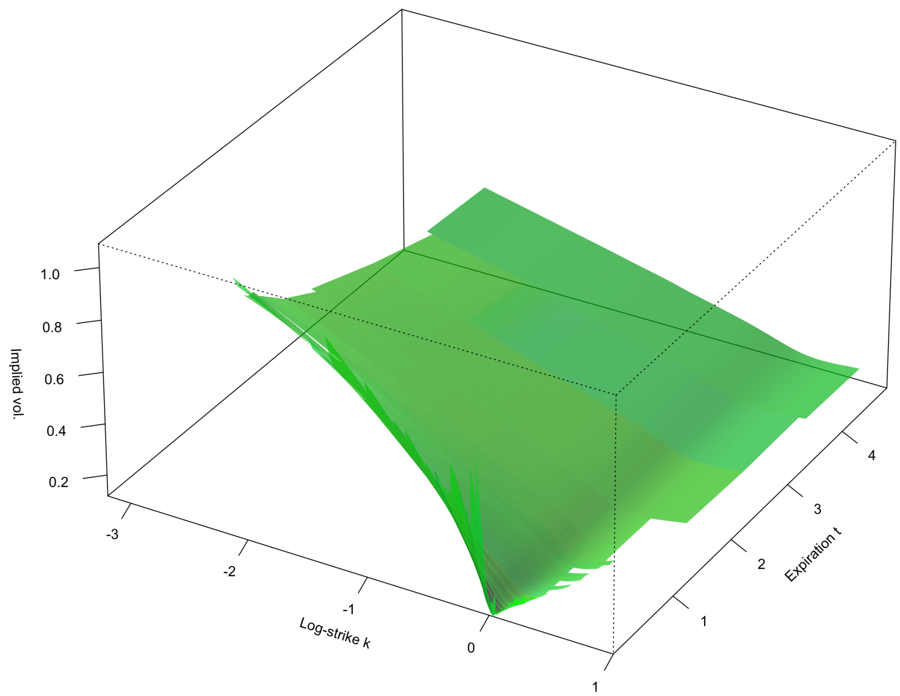

Figure 1. The SPX volatility surface as of 15-Feb-2023 (data from OptionMetrics via WRDS).

Set up the environment

In [1]:
```
# Download .zip from GitHub
release_url <- "https://github.com/jgatheral/RoughVolatilityWorkshop2025/raw/main/QM_2025.zip'
local_zip <- "QM_2025.zip"

download.file(release_url, local_zip, mode = "wb")
unzip(local_zip)
```

In [2]:
```
library(repr)
library(colorspace)
library(stinepack)
library(foreach)
library(doParallel)
```

```
Loading required package: iterators

Loading required package: parallel
```

Some R-code

In [3]:
```
source("black_scholes.R")
source("black_formula.R")
source("fwd_var_curve.R")
source("fukasawa_robust.R")
source("gamma_kernel.R")
source("hybrid_bss.R")
source("lewis.R")
source("plot_ivols.R")
```

```
source("plot_ivols_mc.R")
source("rough_heston_pade.R")
```

Load volatility smiles from 15-Feb-2023

In [4]:
```
load("spx_vix_vols_20230215.rData")

ivolData <- spxIvols20230215
head(ivolData)
```

A data.frame: 6 × 7

	Expiry	Texp	Strike	Bid	Ask	Fwd	CallMid
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	20230216	0.002737851	1000	NA	7.793085	4146.742	NA
2	20230216	0.002737851	1200	NA	6.813266	4146.742	NA
3	20230216	0.002737851	1400	NA	5.987566	4146.742	NA
4	20230216	0.002737851	1600	NA	5.273554	4146.742	NA
5	20230216	0.002737851	1800	NA	4.644049	4146.742	NA
6	20230216	0.002737851	2000	NA	4.080578	4146.742	NA

In [5]:
```
options(repr.plot.width=10,repr.plot.height=7,repr.plot.res=150)

res.plot <- plotIvols(ivolData)
```

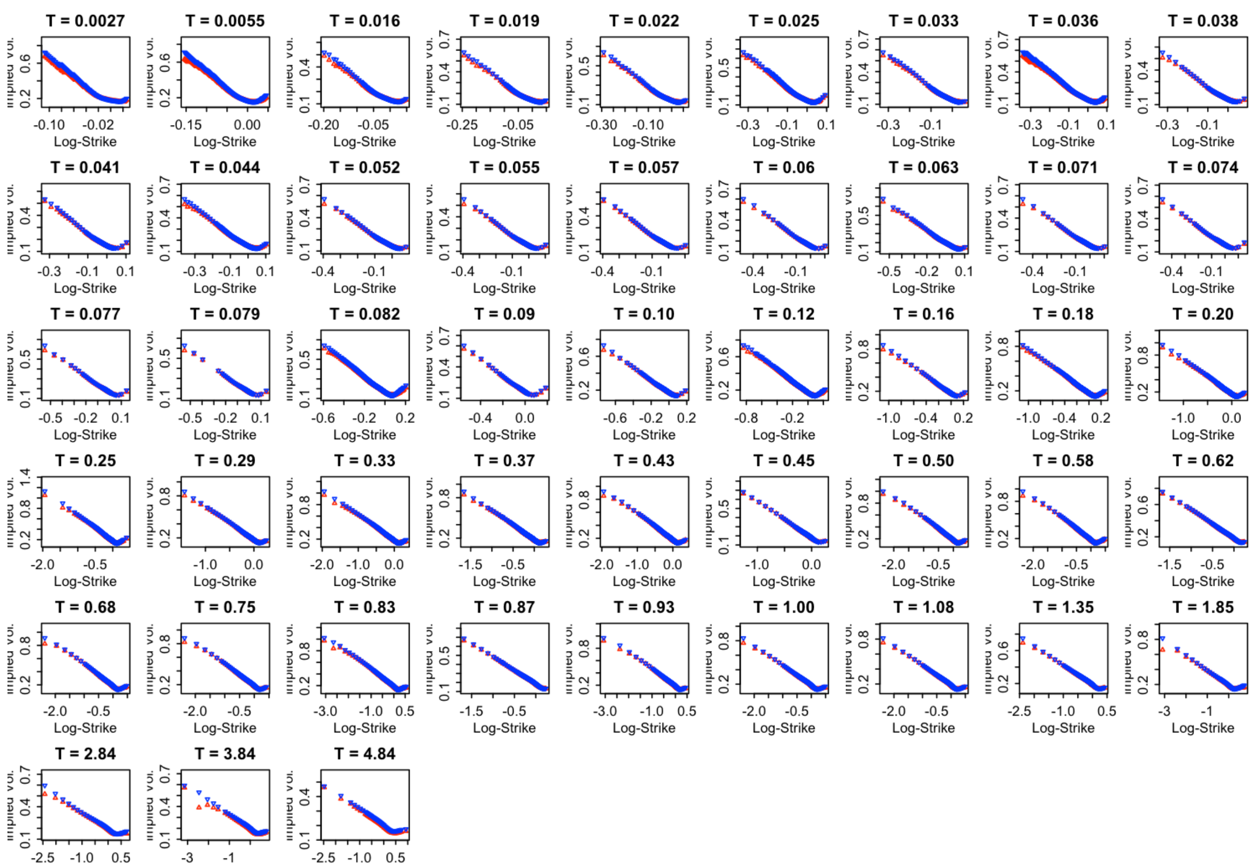

Figure 2: SPX smiles as of February 15, 2023.

Set up nice colors

In [6]:
```
my.col <- sequential_hcl(5, palette="Batlow")
bl <- "royalblue"
rd <- "red2"
pk <- "hotpink1"
gr <- "green4"
br <-"brown"
pu <- "purple"
or <- "orange"
```

Robust estimation of the variance contract

In [7]:
```
expiries <- res.plot$expiries

(vs <- varSwap.Robust(ivolData)$vs.mid)
```

0.036529328507355 · 0.0317776298748159 · 0.019801436839558 · 0.0216205797598485 · 0.0239817142815479 · 0.0260070933624724 · 0.0230480008871306 · 0.0242269111404731 · 0.0254217621840437 · 0.0262608820432924 · 0.0272812740956352 · 0.0251873317922458 · 0.0270816758954462 · 0.0277890020550951 · 0.0283828078588884 · 0.0307845544704758 · 0.0288281365209749 · 0.0326234863276105 · 0.033036923041284 · 0.0333085545046597 · 0.0333045778294113 · 0.0321547119979983 · 0.0372856381939817 · 0.0368526562667083 · 0.0384015885110832 · 0.0382728883769698 · 0.0389762987224796 · 0.0422129823407889 · 0.041766324573276 · 0.0449034147120543 · 0.0453292487617555 · 0.0467388646631131 · 0.046384306264404 · 0.0493899123530733 · 0.0506716850375437 · 0.0509751148828293 · 0.0525838020688622 · 0.0536616949197811 · 0.0552555914782415 · 0.0528434116744946 · 0.0544748623553668 · 0.0544544038189155 · 0.0549311335723502 · 0.0552749722502056 · 0.0571945373055973 · 0.0567880111485679 · 0.0571112166824289 · 0.059465108018572

In [8]:
```
plot(expiries,vs,type="b",pch=20,lwd=2,col=pk,ylab="Swap values",xlab="Maturity (years)",log='
```

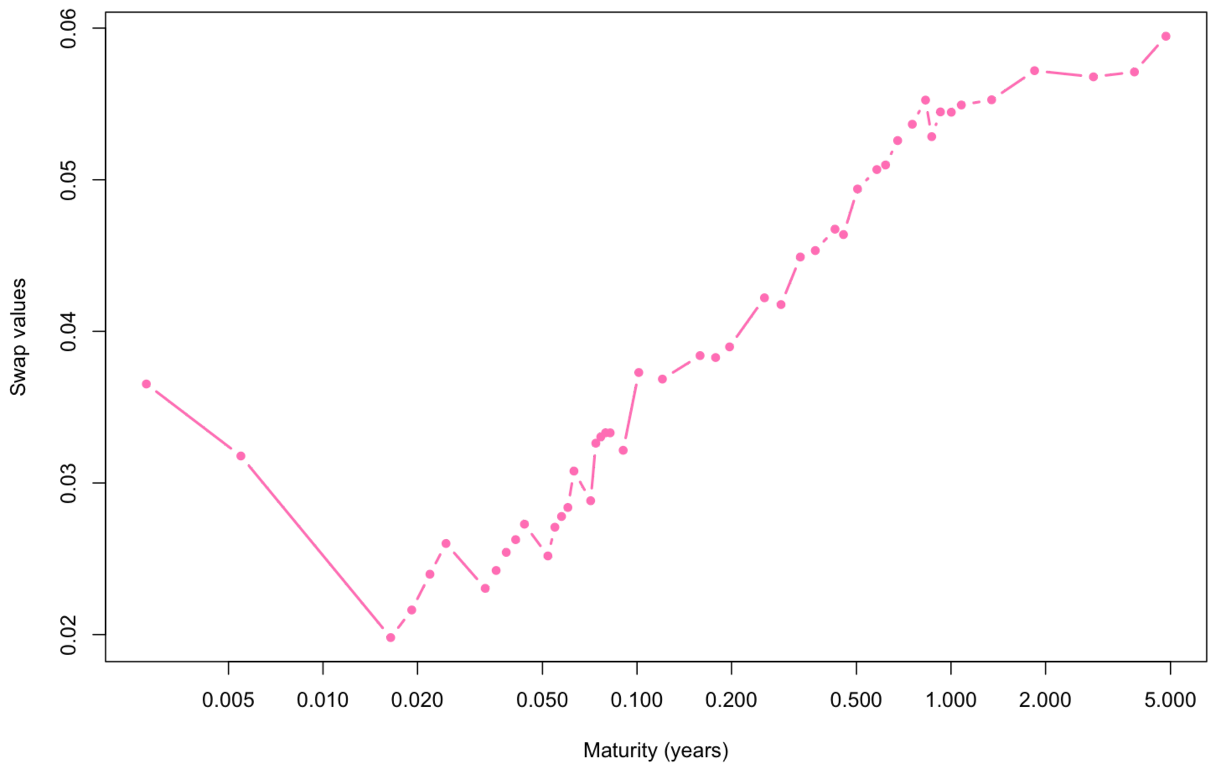

Figure 3: Log-linear plot of the forward variance curve.

The forward variance curve from the variance curve

- By defition, $\xi_t(u) = \mathbb{E}_t\left[V_u\right]$.
- Recall that

$$M_t(T) = \int_t^T \xi_t(u)\,du.$$

- Differentiating wrt T gives

$$\xi_t(T) = \partial_T M_t(T).$$

Exact smooth forward curve construction

- In 2019, Baruch MFE student Rick Cao implemented (beautifully) a beautiful paper of [Filipović and Willems][5]
- That paper presents a non-parametric method to estimate the discount curve from market quotes, that reproduces the market quotes perfectly and has maximal smoothness in the sense that it minimizes the L^2-norm of the forward curve.
- We apply this method to the variance swap curve.
  - The resulting forward variance curve is piecewise quadratic.
- Warning: The resulting forward variance curve is not guaranteed to be positive - though this does not seems to matter in practice.

Adding a bid-offer spread to smooth the curve

- In practice, `w.in` is not known exactly but only up to some bid-offer spread.
  - Moreover some expirations have more strikes than others and we get a better estimate.
- We input this bid-offer volatility spread using the `eps` parameter.

Why the forward variance curve should be smooth

- If there are two forward variance curves that are consistent with the data, the smoother one is better.
- To see why, consider trading forward variance swaps around discontinuities in the forward curve!

In [9]:
```
xi.curve.smooth
```

```
function (expiries, w.in, xi = TRUE, eps = 0)
{
    phi <- function(tau) {
        function(x) {
            min <- pmin(x, tau)
            return(1 - min^3/6 + x * tau * (2 + min)/2)
        }
    }
    phi.deri <- function(tau) {
        function(x) {
            min <- pmin(x, tau)
            return(tau - min^2/2 + tau * min)
        }
    }
    n <- length(expiries)
    c <- diag(n)
    A <- array(dim = c(n, n))
    for (i in seq(1, n)) {
        A[i, ] <- phi(expiries[i])(expiries)
    }
    A.inv <- solve(A)
    obj.1 <- function(err.vec) {
        v <- w.in + 2 * sqrt(w.in) * err.vec * sqrt(expiries)
        return(t(v) %*% A.inv %*% v)
    }
    res.optim <- optim(rep(0, n), obj.1, method = "L-BFGS-B",
        lower = rep(-eps, n), upper = rep(eps, n))
    err.vec <- res.optim$par
    w.in.1 <- w.in + 2 * sqrt(w.in) * err.vec * sqrt(expiries)
    Z <- A.inv %*% w.in.1
    curve.raw <- function(x) {
        sum.curve <- 0
        sum.curve.deri <- 0
        for (i in seq(1, n)) {
            sum.curve <- sum.curve + Z[i] * phi(expiries[i])(x)
            sum.curve.deri <- sum.curve.deri + Z[i] * phi.deri(expiries[i])(x)
        }
        if (xi) {
            return(sum.curve.deri)
        }
        else {
            return(sum.curve)
        }
    }
    xi.curve.out <- Vectorize(curve.raw)
    fit.errs <- sqrt(w.in.1/expiries) - sqrt(w.in/expiries)
    return(list(xi.curve = xi.curve.out, fit.errs = fit.errs,
        w.out = w.in.1))
}
```

In [10]:
```
w.in <- vs*expiries
xi.smooth <- xi.curve.smooth(expiries, w.in, eps = .006)
xi.smooth.c <- xi.smooth$xi.curve
xi.smooth.w.out <- xi.smooth$w.out
```

In [11]:
```
curve(xi.smooth.c,from=0,to=3,col=bl,lty=1,lwd=2,xlab="Maturity",ylab="Forward variance",n=10(
```

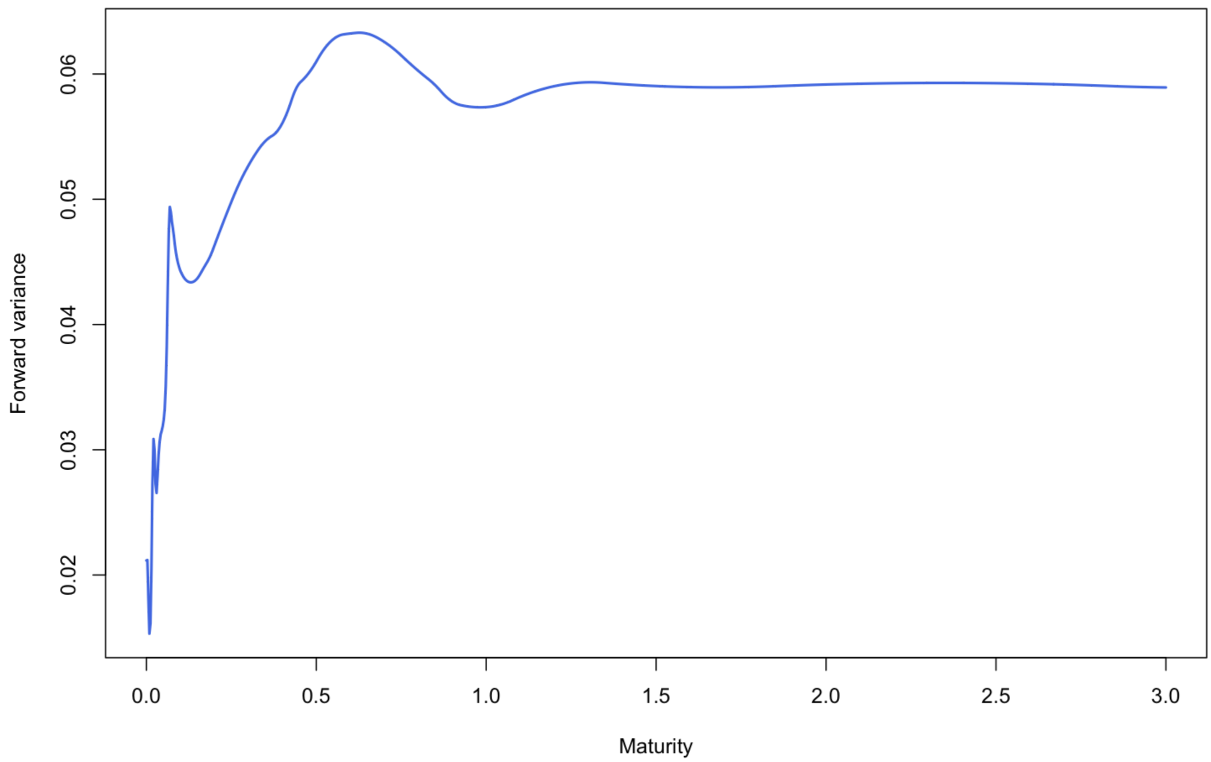

Figure 4: Smooth approximation to the forward variance curve.

What does exact interpolation mean?

- Output variance swaps exactly match input variance swaps (up to the bid-offer spread).

In [12]:
```
plot(expiries,w.in/expiries,col=pk,lwd=2,pch=20,xlab="Maturity",ylab="Variance rate",type="b"
points(expiries,xi.smooth.w.out/expiries,col=bl,pch=20,lwd=2,type="b")
```

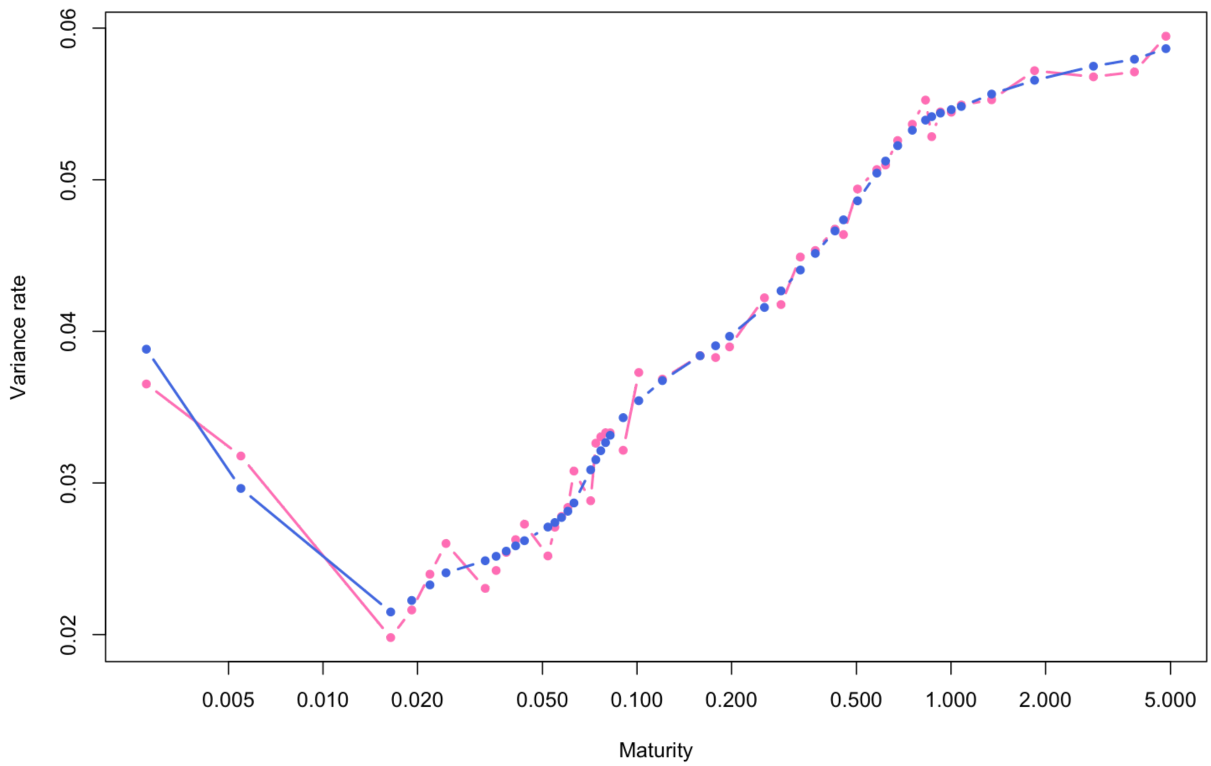

Figure 5: Blue circles are input variance swaps; pink circles are output variance swaps; green dots are from variance swaps with bid/offer spread.

Instantiate the forward variance curve

We can speed things up by instantiating the curve:

In [13]:
```
xiCurveObj <- CurveSmoothBuilder(expiries,vs*expiries,eps=.006)
xiCurveObj$fitCurve()
xi.curve.fast <- xiCurveObj$getForwardVarCurve()
```

In [14]:
```
save(xiCurveObj,file="xi20230215.rData")
```

Note: Object-oriented programming in R. We can even save the object for future use!

Plot the fast instantiated forward variance curve

In [15]:
```
curve(xi.smooth.c,from=0,to=3,col=bl,lty=1,lwd=2,xlab="Maturity",ylab="Forward variance")
curve(xi.curve.fast,from=0,to=3,col=pk,lwd=3,add=T,lty=2)
```

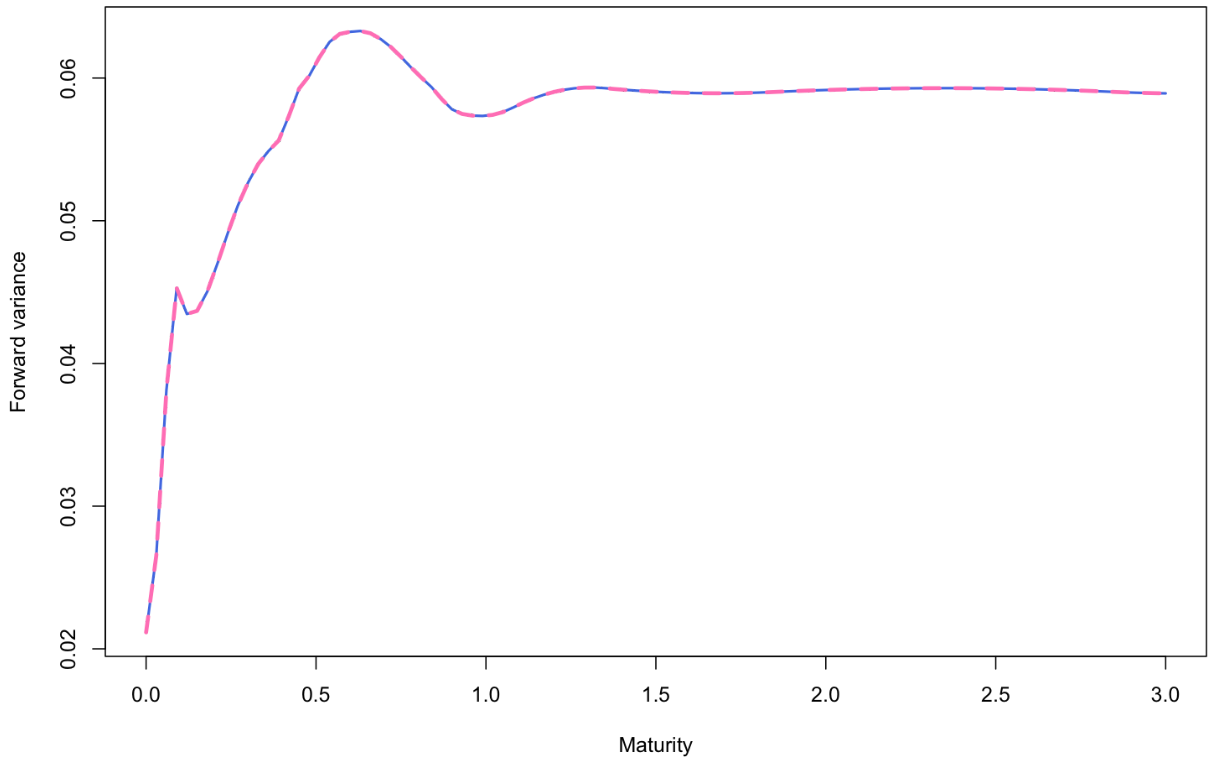

Figure 6: Original and fast (instantiated) forward variance curve.

Using forward variance curves in practice

- Forward variance models, rough volatility models in particular, take the forward variance curve as given.
- The forward variance curve is estimated from the variance swap curve (by differencing for example).
  - Variance swap estimates depend on the extrapolation methodology.
- A forward variance model that takes the estimated forward variance curve as input will not generate the same variance swap values as the estimates.
  - In particular, the smile extrapolation will be different.
- In practice therefore, we iterate on the forward variance curve so as to match market and model ATM volatilities.

The stock price process

- The observed anticorrelation between price moves and volatility moves may be modeled naturally by anticorrelating the Brownian motion W that drives the volatility process with the Brownian motion driving the price process.
- Thus

$$\frac{dS_t}{S_t} = \sqrt{V_t}\, dZ_t$$

with

$$dZ_t = \rho\, dW_t + \sqrt{1-\rho^2}\, dW_t^{\perp}$$

where ρ is the correlation between volatility moves and price moves.

Hybrid simulation of BSS processes

- The Rough Bergomi variance process is a special case of a Brownian Semistationary (BSS) process.
- [Bennedsen, Lunde and Pakkanen][3] show how to simulate such processes more efficiently.
- [McCrickerd and Pakkanen][10] show how to increase the efficiency of the hybrid scheme with variance reduction.
  - Moreover, they provide a sample Jupyter notebook!
- An improved version of their idea is roughly as follows:

$$\begin{aligned}\int_t^u \frac{dW_s}{(u-s)^\gamma} &= \int_0^\tau \frac{dW_s}{s^\gamma}\\ &\approx \sum_{k=1}^{\kappa} \int_{t_{k-1}}^{t_k} \frac{dW_s}{s^\gamma} + \sum_{k=\kappa+1}^{n} b_k \int_{t_{k+1}}^{t_k} dW_s\\ &= \sum_{k=1}^{\kappa} \int_{t_{k-1}}^{t_k} \frac{dW_s}{s^\gamma} + \sum_{k=\kappa+1}^{n} b_k\, Z_k \sqrt{\Delta}\end{aligned}$$

where $\gamma = \frac{1}{2} - H$, $\Delta = (u-t)/n$, $t_k = k\,\Delta$,

- $$b_k^2 = \mathrm{var}\left[\int_{t_{k+1}}^{t_k} \frac{dW_s}{s^\gamma}\right] = \int_{t_{k+1}}^{t_k} \frac{ds}{s^{2\gamma}} = \frac{1}{2H}\left\{t_k^{2H} - t_{k-1}^{2H}\right\}$$

and the Z_k are iid $N(0,1)$ random variables.

- The choice $\kappa = 0$ corresponds to the Euler scheme (or Riemann sum scheme), which performs relatively poorly.
- The choice $\kappa = 1$ works well in practice.
  - The idea is to not only match the variance at each step (which the Riemann sum scheme does) but also the covariance

$$\mathrm{cov}\left[\int_0^\Delta \frac{dW_s}{s^\gamma}, \int_0^\Delta dW_s\right] = \int_0^\Delta \frac{ds}{s^\gamma} = \frac{\Delta^{1-\gamma}}{1-\gamma} = \frac{\Delta^{H+\frac{1}{2}}}{H+\frac{1}{2}}.$$

  - We simulate another normal random variable to achieve this variance and covariance.

R-implementation of the hybrid scheme

The following function simulates the Riemann-Liouville process

$$\tilde{W}_t^H := \sqrt{2H} \int_0^t (t-s)^{\alpha-1}\, dW_s,$$

where $\alpha = H + \frac{1}{2}$.

In [16]:
```
WtildeRL.sim
```

```
function (params, hybrid = T)
function(W, Wperp) {
    library(stats)
    steps <- dim(W)[1]
    N <- dim(W)[2]
    stopifnot(dim(Wperp) == c(steps, N))
    dt <- 1/steps
    wp <- Vectorize(wRL(params))
    sqrt.dt <- sqrt(dt)
    tj <- (1:steps) * dt
    wpj <- c(0, wp(tj))
    bstar <- sqrt(diff(wpj)/dt)
    cstar <- cov1RL(params)(dt)
    rhostar <- cstar/(bstar[1] * bstar[2])
    rhobarstar <- sqrt(1 - rhostar^2)
    f <- function(n) {
        Wr <- W[steps:1, n]
        Y.Euler <- convolve(bstar, Wr, type = "open")[1:steps]
        Y.Correct <- bstar[1] * ((rhostar - 1) * W[, n] + rhobarstar *
            Wperp[, n])
        return((Y.Correct * isTRUE(hybrid) + Y.Euler) * sqrt(dt))
    }
    Wtilde <- sapply(1:N, f)
    return(Wtilde)
}
```

The following code uses the same Riemann-Liouville process for each expiry.

In [17]:
```
hybridSchemeRL.S
```

```
function (params, xi)
function(paths, steps, expiries) {
    eta <- params$eta
    H <- params$al - 1/2
    rho <- params$rho
    N <- paths
    W <- matrix(rnorm(N * steps), nrow = steps, ncol = N)
    Wperp <- matrix(rnorm(N * steps), nrow = steps, ncol = N)
    Zperp <- matrix(rnorm(N * steps), nrow = steps, ncol = N)
    Z <- rho * W + sqrt(1 - rho * rho) * Zperp
    Wtilde <- WtildeRL.sim(params)(W, Wperp)
    sim <- function(expiry) {
        dt <- expiry/steps
        ti <- (1:steps) * dt
        xi.t <- xi(ti)
        v1 <- xi.t * exp(eta * expiry^H * Wtilde - 1/2 * eta^2 *
            ti^(2 * H))
        v0 <- rep(xi(0), N)
        v <- rbind(v0, v1[-steps, ])
        logs <- apply(sqrt(v * dt) * Z - v/2 * dt, 2, sum)
        s <- exp(logs)
        return(s)
    }
    st <- t(sapply(expiries, sim))
    return(st)
}
```

Run the hybrid BSS scheme

We will use R parallel processing functionality.

In [18]:
```
paths <- 1e5
steps <- 200
```

Parameters are from the fit to SPX smiles as of 14-Aug-2013 reported in

In [19]:
```
params.rBergomi <- list(al=0.55, eta=2.3, rho=-0.9)
xiCurve <- xi.curve.fast # We use the instantiated curve for speed
```

In [20]:
```
t0<-proc.time()

#number of iterations
iters<- max(1,floor(paths/1000))

#setup parallel backend
cl.num <- detectCores() # This number is 8 on my MacBook Pro
cl<-makeCluster(cl.num)
registerDoParallel(cl)

#loop
ls <- foreach(icount(iters),.packages = "stinepack") %dopar% {
        hybridSchemeRL.S(params.rBergomi,xiCurve)(paths=1000, steps=steps, expiries=expiries)
        }

stopCluster(cl)
mcMatrix2013 <- do.call(cbind, ls) #Bind all of the submatrices into one big matrix
```

```
print(proc.time()- t0)
```

```
   user  system elapsed 
  0.173   0.075   8.341 
```

Plot actual and rough Bergomi (2013) smiles

In [21]:
```
system.time(res.plot2013 <- plotIvolsMC(ivolData,mcMatrix=mcMatrix2013))
```

```
   user  system elapsed 
 48.377   0.139  48.526 
```

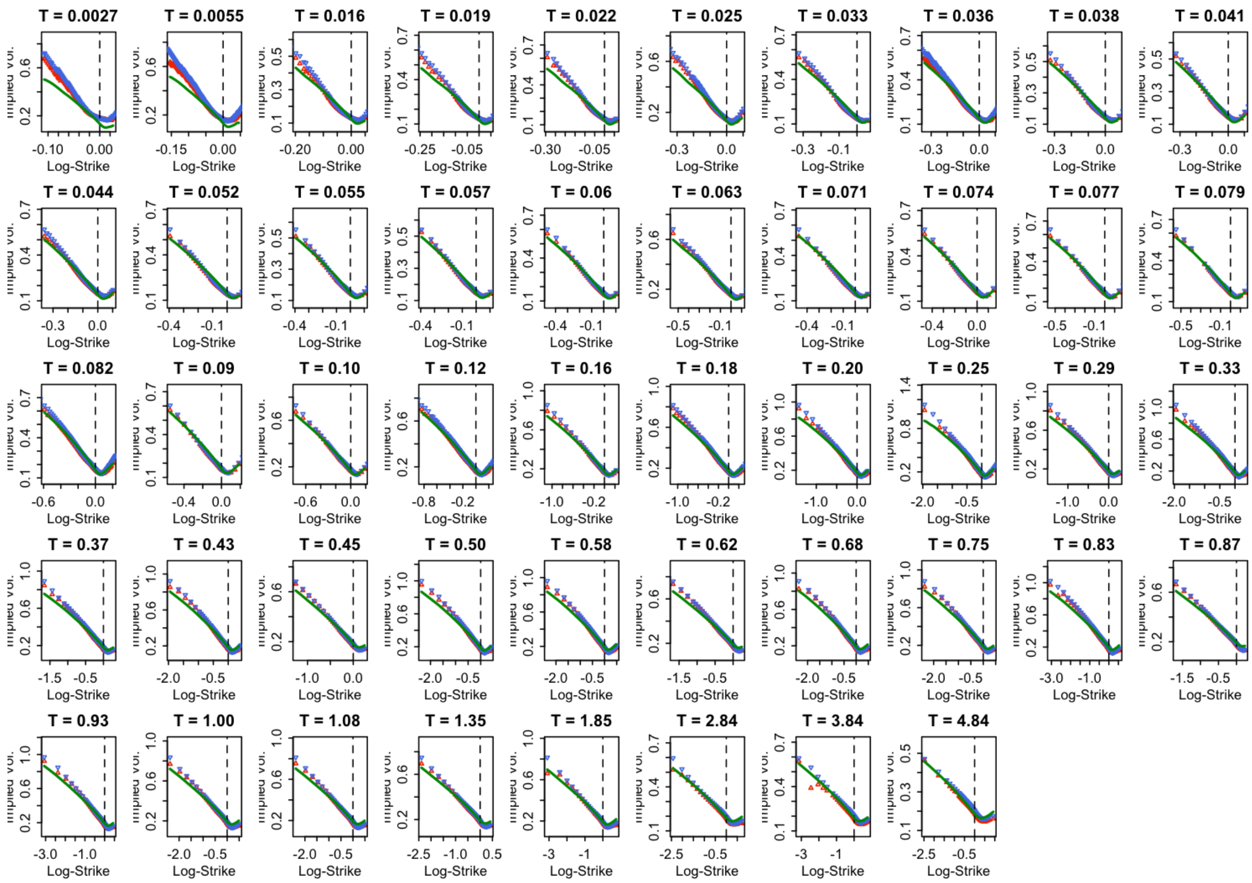

Figure 7: Rough Bergomi smiles (green) with parameters from 2013 superimposed on February 15, 2023 SPX smiles.

Plot a selection of actual and rough Bergomi smiles

In [22]:
```
res.plot2013.6 <- plotIvolsMC(ivolData,mcMatrix=mcMatrix2013,slices= c(2,10,21,28,34,42))
```

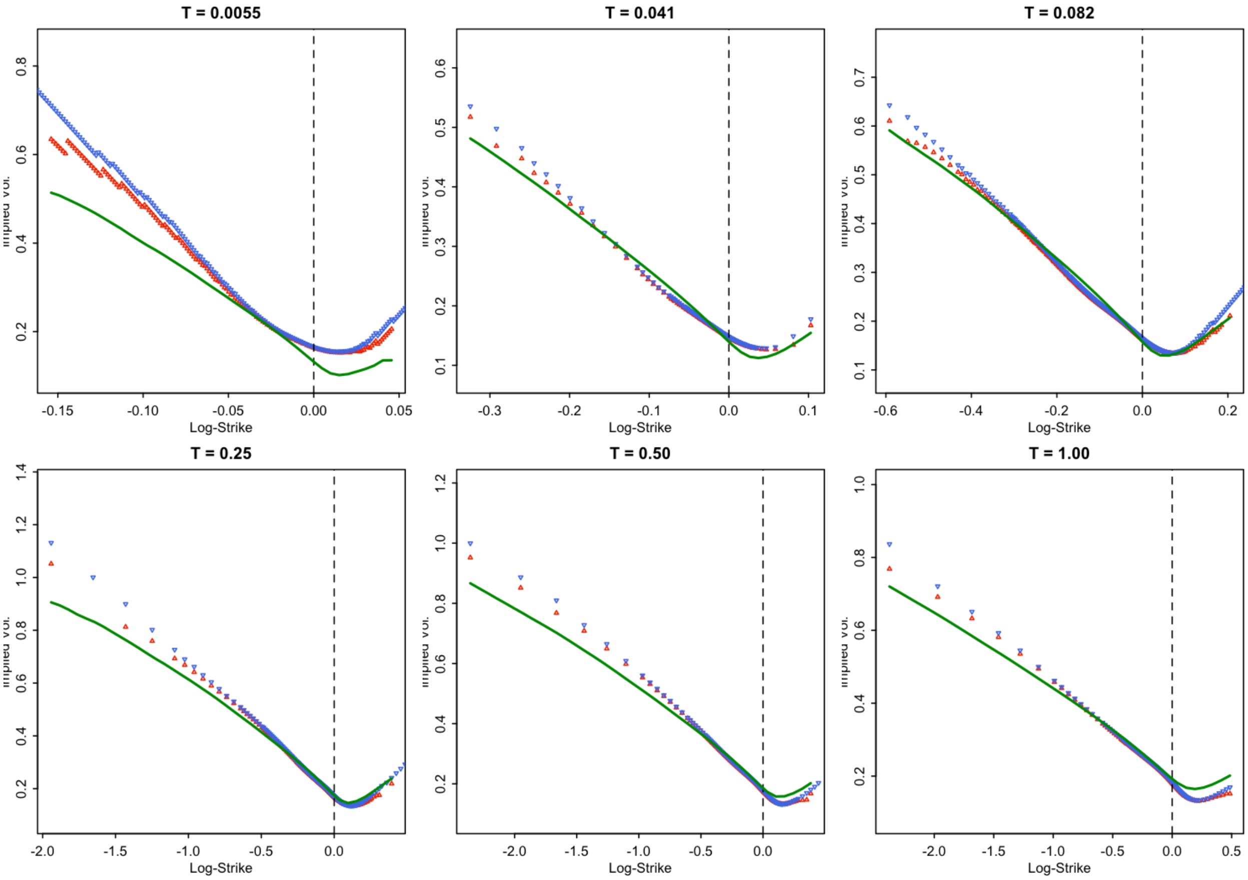

Figure 8: Six rough Bergomi smiles (green) with parameters from 2013 superimposed on February 15, 2023 SPX smiles.

Comments on Figures 7 and 8

- Considering that we are using parameters from 2013, the rough Bergomi smiles look pretty good.
  - Rough Bergomi parameters seem to be remarkably stable!
- If simulation were fast enough, we could just iterate on these parameters to find the best fit to observed option prices.
  - The BSS scheme is not yet fast enough, at least in my R implementation.

Plot the ATM skew vs ATM rough Bergomi skew

In [23]:
```
plot(res.plot2013$expiries,res.plot2013$atmSkew,pch=20,col=bl,xlab="Expiry",ylab="ATM skew")
lines(res.plot2013$expiries,res.plot2013$atmSkewMC,col=gr,lwd=2)
```

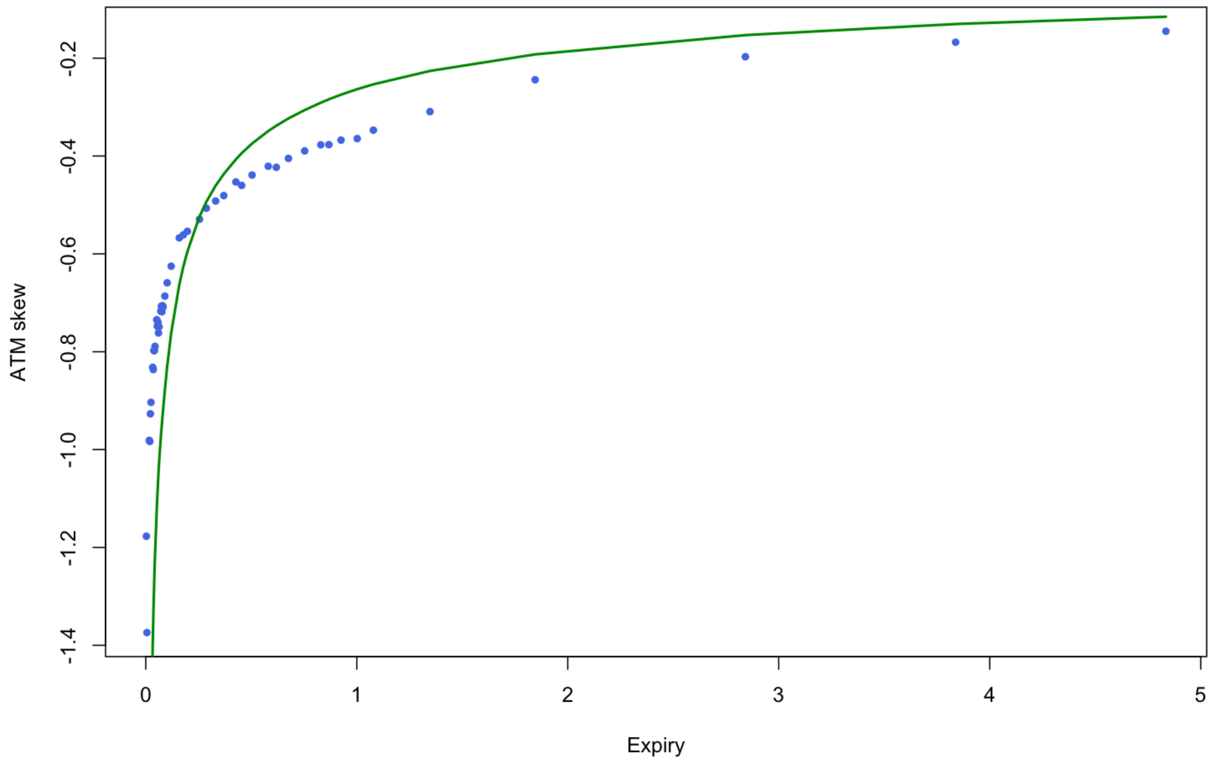

Figure 9: Termstructure of skew seems to be off. Maybe we can do better?

Guessing rBergomi model parameters

- The rBergomi model has only three parameters: H, η and ρ.
- The model parameters H, η and ρ have very direct interpretations:
  - H controls the decay of ATM skew $\mathcal{S}(\tau)$ for very short expirations.
  - The product $\rho\eta$ sets the level of the ATM skew for longer expirations.
    - Keeping $\rho\eta$ constant but decreasing ρ (so as to make it more negative) pushes the minimum of each smile towards higher strikes.
- So we can guess parameters in practice.
  - A couple of examples of the results of guessing are given in [Bayer, Friz and Gatheral][1].

Log-log plot of rough Bergomi ATM skew for various H

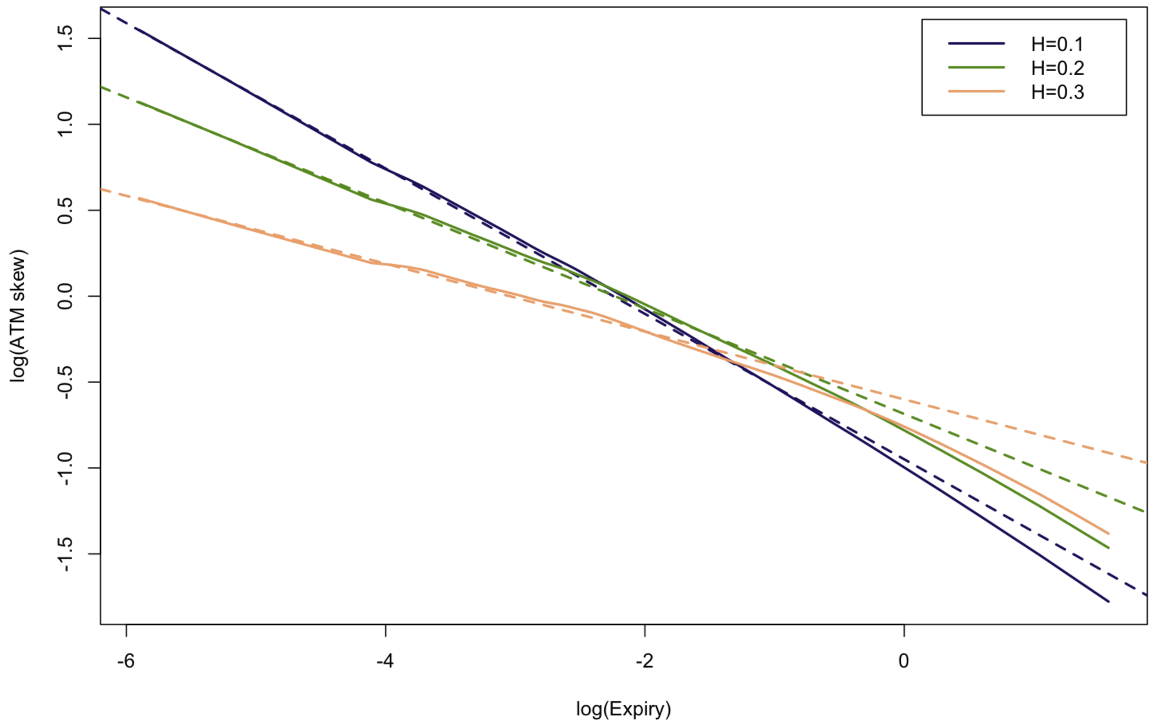

Figure 10: log(ATM skew) for $H \in \{0.1, 0.2, 0.3\}$ together with linear fits to the first ten points.

Estimate H from term structure of skew

- We see that for short expirations, the rough Bergomi skew is almost a perfect power-law
  - $S(\tau) \sim \tau^{\alpha-1}$ with $\alpha = H + 1/2$.
- So let's estimate the slope of the empirical ATM skew!

Estimate the power-law

In [24]:
```
res.plot2013 <- plotIvolsMC(ivolData,mcMatrix=mcMatrix2013,plot=F)

summary(fit.lm2013 <- lm(log(-res.plot2013$atmSkew[2:20])~log(res.plot2013$expiries[2:20])))
```

```
Call:
lm(formula = log(-res.plot2013$atmSkew[2:20]) ~ log(res.plot2013$expiries[2:20]))

Residuals:
      Min        1Q    Median        3Q       Max 
-0.039155 -0.014162 -0.008391  0.017412  0.045638 

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)                      -0.979911   0.028417  -34.48  < 2e-16 ***
log(res.plot2013$expiries[2:20]) -0.240428   0.008568  -28.06 1.11e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02454 on 17 degrees of freedom
Multiple R-squared:  0.9789,	Adjusted R-squared:  0.9776 
F-statistic: 787.5 on 1 and 17 DF,  p-value: 1.113e-15
```

Graph the fit

In [25]:
```
plot(log(res.plot2013$expiries),log(-res.plot2013$atmSkew),pch=20,col=bl,xlab="Expiry",ylab="/
abline(fit.lm2013,col=rd,lwd=2)
```

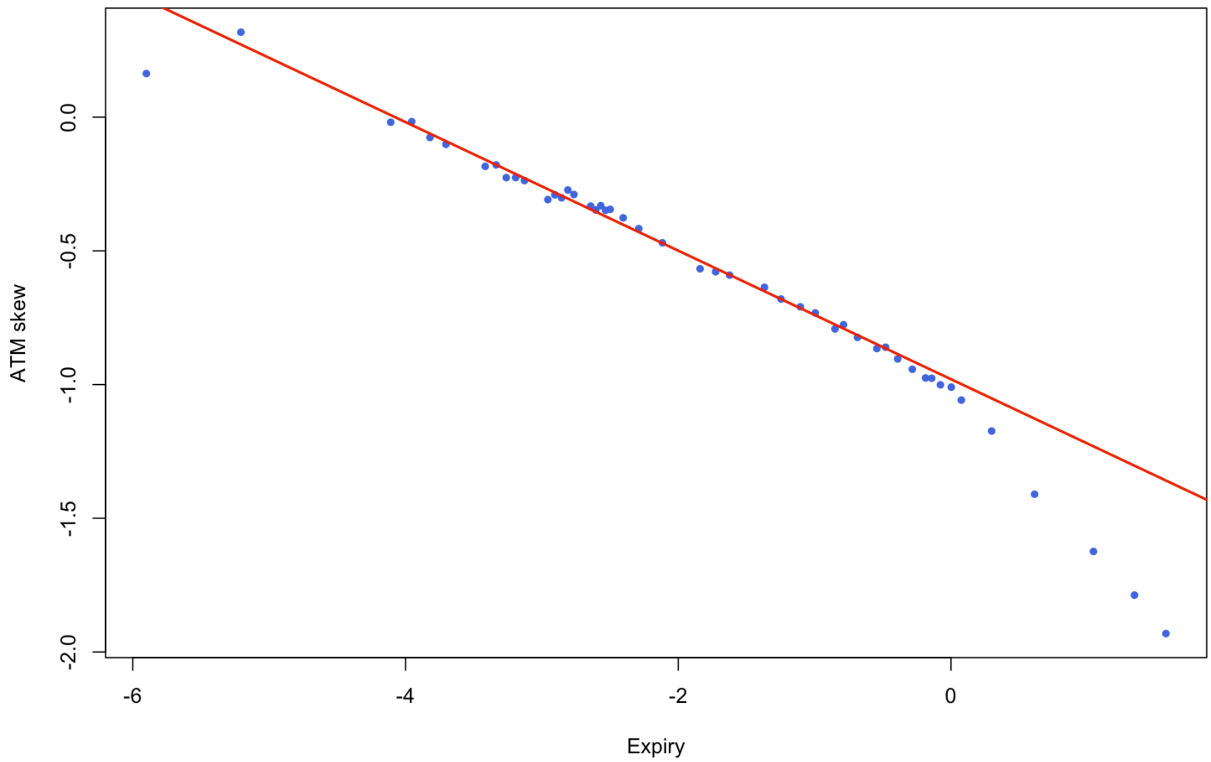

Figure 10: Blue points are empirical skews; the red line is a linear fit to the first 20 points (excluding the first).

Run the rough Bergomi code with $H = 0.26$

- And playing with η and ρ a little...

In [26]:
```
params.rBergomi.skew <- list(al=1-0.24, eta=1.8, rho=-0.8)
xiCurve <- xi.curve.fast # We use the instantiated curve for speed

t0<-proc.time()

#number of iterations
iters<- max(1,floor(paths/1000))

#setup parallel backend
cl.num <- detectCores() # This number is 8 on my MacBook Pro
cl<-makeCluster(cl.num)
registerDoParallel(cl)

#loop
ls <- foreach(icount(iters),.packages = "stinepack") %dopar% {
        hybridSchemeRL.S(params.rBergomi.skew,xiCurve)(paths=1000, steps=steps, expiries=expi
        }

stopCluster(cl)
mcMatrix2023 <- do.call(cbind, ls) #Bind all of the submatrices into one big matrix
```

```
print(proc.time()- t0)
```

```
 user  system elapsed 
0.145   0.079   8.418 
```

In [27]:
```
res.plot2023 <- plotIvolsMC(ivolData,mcMatrix=mcMatrix2023)
```

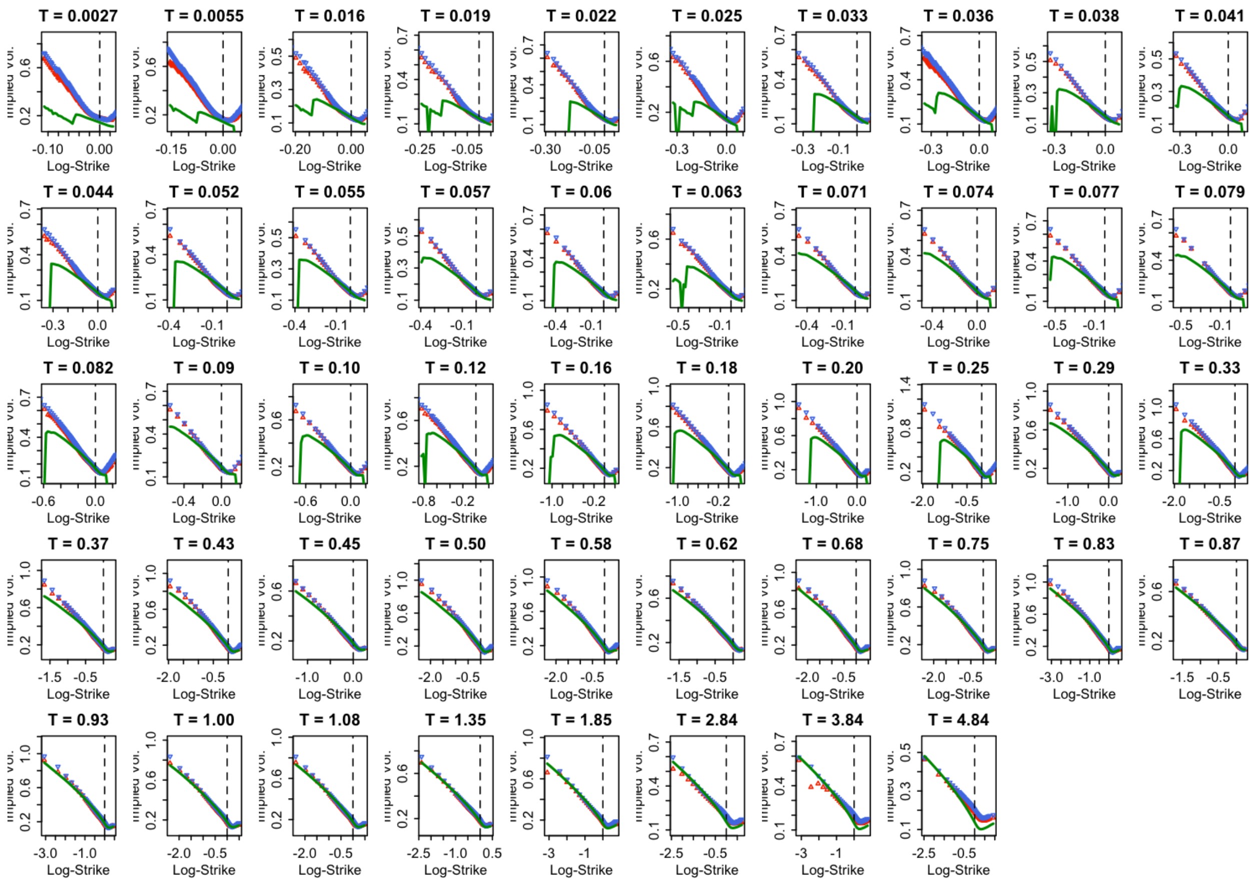

Figure 11: Blue points are empirical skews; the green lines are from the rough Bergomi simulation.

Check the fit

In [28]:
```
plot(log(res.plot2023$expiries),log(-res.plot2023$atmSkew),pch=20,col=bl,xlab="log(Expiry)",y
lines(log(res.plot2023$expiries),log(-res.plot2023$atmSkewMC),col=gr,lwd=2)
```

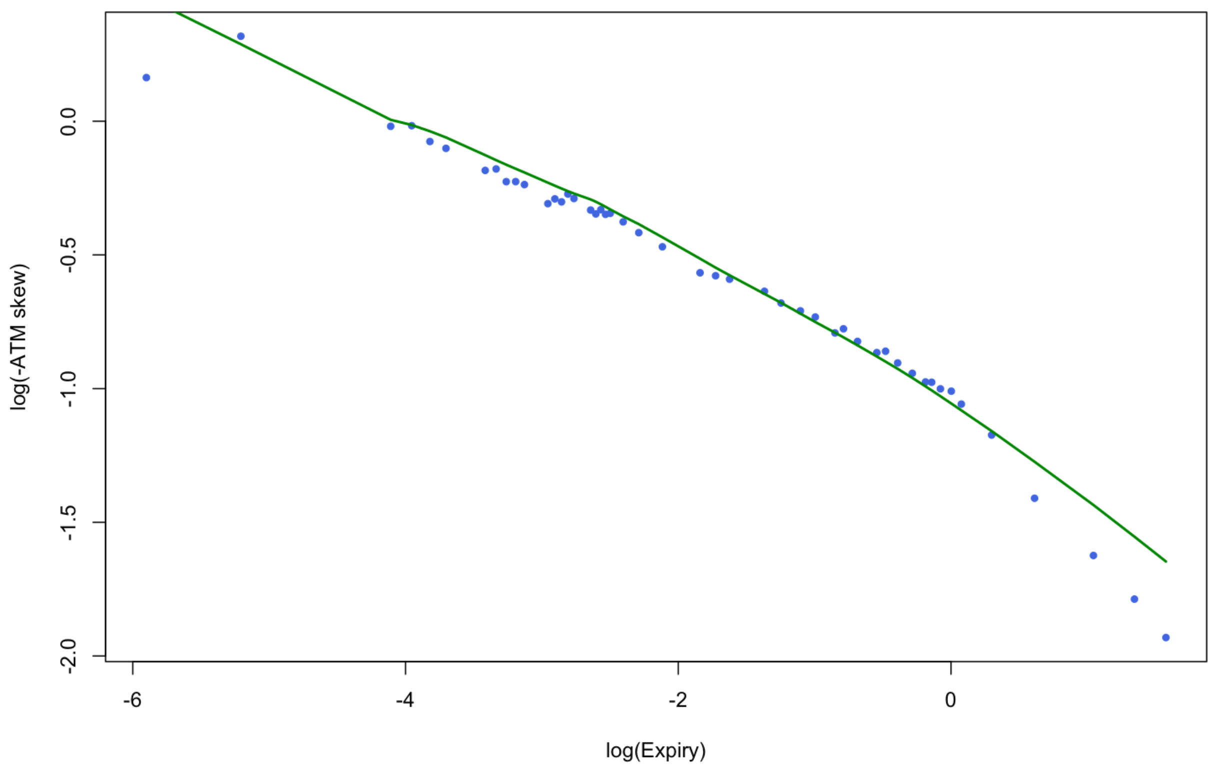

Figure 12: Blue points are empirical skews; the green line is from the rough Bergomi simulation.

Check the six expirations, comparing with the 2013 guess

In [29]:
```
res.plot2023.6 <- plotIvolsMC2(ivolData,mcMatrix=mcMatrix2023,mcMatrix2=mcMatrix2013,slices= (
```

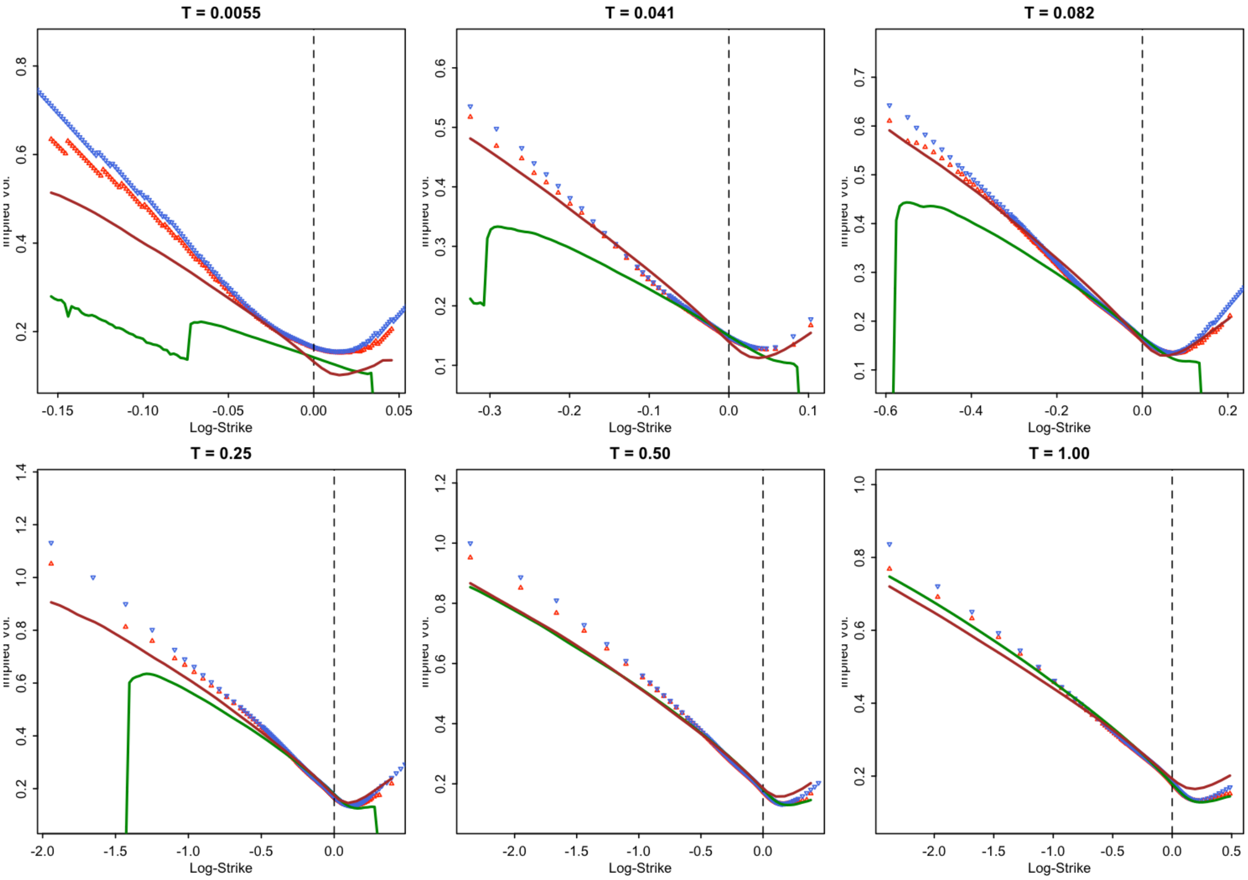

Figure 13: Six rough Bergomi smiles. Green is with parameters $H = 0.26, \eta = 1.8, \rho = -0.8$; Brown is with parameters from 2013.

Calibration using machine learning

- In a very well-cited paper, [Horvath et al.][7] showed how to calibrate the rough Bergomi model to the volatility surface using machine learning.
  - A neural network is trained to map the shape of the volatility surface to model parameters.

H from VIX options and futures

- Rather than brute-force fitting a rough volatility model to the volatility surface, following [Jacquier, Martini and Muguruza], one can try to fix H from the term structure of the convexity adjustment between the variance contract and VIX futures.
- Once the Volterra process $\tilde{W}$ has been simulated for this H, iterating on the parameters η and ρ to fit the observed volatility surface is relatively fast.

The distribution of VIX future payoffs

- Denote the terminal value of the VIX futures by $\sqrt{\zeta(T)}$. Then, by definition (see Chapter 11 of [The Volatility Surface][6] for more details),

$$\zeta(T) = \frac{1}{\Delta} \int_T^{T+\Delta} \mathbb{E}_T \left[V_u \right] \, du.$$

where Δ is one month.

- In the rough Bergomi model,

$$V_u = \xi_t(u)\,\mathcal{E}\left(\eta\,\sqrt{2\,H}\,\int_t^u \frac{dW_s}{(u-s)^\gamma}\right)$$

  with $\gamma = 1/2 - H$ so V_u is lognormal.

The lognormal approximation under rough Bergomi

- Under rough Bergomi, the VIX payoff and its square $\zeta(T)$ should be approximately lognormally distributed.
  - The quality of this approximation was confirmed by [Jacquier, Martini and Muguruza][8].
  - In that case, the terminal distribution of $\zeta(T)$ is completely determined by $\mathbb{E}\left[\,\zeta(T)|\,\mathcal{F}_t\right]$ and $\mathrm{var}[\log\zeta(T)|\mathcal{F}_t]$.

- Obviously

$$\mathbb{E}_t\left[\zeta(T)\right] = \frac{1}{\Delta}\,\int_T^{T+\Delta}\,\xi_t(u)\,du.$$

  - Recall that forward variances $\xi_t(u)$ may be estimated from variance swaps which can themselves be proxied by the log-strip (see Chapter 11 of [The Volatility Surface][5] again).
    - Alternatively they may be estimated from linear strips of VIX options.

Approximating the conditional variance of $\zeta(T)$ under rough Bergomi

- To estimate the conditional variance of $\zeta(T)$, we approximate the arithmetic mean by the geometric mean as follows:

$$\zeta(T) \approx \exp\left\{\frac{1}{\Delta}\,\int_T^{T+\Delta}\,\mathbb{E}[\log V_u|\mathcal{F}_T]\,du\right\}.$$

Let $y_u = \log V_u$ and recall that $\gamma = \frac{1}{2} - H$. Apart from $\mathcal{F}_t$ measurable terms (abbreviated as ``drift''), we have

$$\begin{aligned}
\int_T^{T+\Delta}\mathbb{E}_t\left[y_u\right]\,du &= \eta\sqrt{2H}\int_t^T \frac{dW_s}{(u-s)^\gamma}du + \mathrm{drift} \\
&= \eta\sqrt{2H}\int_t^T\int_T^{T+\Delta}\frac{du}{(u-s)^\gamma}dW_s + \mathrm{drift} \\
&= \eta\frac{\sqrt{2H}}{1-\gamma}\int_t^T\left[(T+\Delta-s)^{1-\gamma} - (T-s)^{1-\gamma}\right]dW_s + \mathrm{drift}.
\end{aligned}$$

This gives

$$\begin{aligned}
\mathrm{var}[\log\zeta(T)|\mathcal{F}_t] &\approx \frac{\eta^2}{\Delta^2}\,\frac{2H}{(H+1/2)^2}\,\int_t^T\left[(T+\Delta-s)^{1/2+H} - (T-s)^{1/2+H}\right]^2 ds \\
&= \eta^2\,(T-t)^{2\,H}\,f^H\left(\frac{\Delta}{T-t}\right)
\end{aligned}$$

where

$$f^H(\theta) = \frac{2H}{(H+1/2)^2}\,\frac{1}{\theta^2}\int_0^1 \left[(1+\theta-x)^{1/2+H} - (1-x)^{1/2+H}\right]^2 dx.$$

The approximate fair value of VIX futures

- In [Bayer, Friz and Gatheral][1], we chose to study the term structure of VVIX (the VIX of VIX).
- It is more natural to follow [Jacquier, Martini and Muguruza][9] and approximate the fair value of VIX futures.
- Under the lognormal approximation, the fair value of the T-maturity VIX future is given by

$$\mathbb{E}\left[\sqrt{\zeta(T)}\middle|\,\mathcal{F}_t\right] = \sqrt{\mathbb{E}\left[\zeta(T)|\,\mathcal{F}_t\right]}\,\exp\left\{-\frac{1}{8}\,\mathrm{var}[\log\zeta(T)|\mathcal{F}_t]\right\}.$$

Load VIX option data

In [30]:
```
# load("vixIvols20230215.rData")
vixVolData <- vixIvols20230215
head(vixVolData)
```

A data.frame: 6 × 7

	Expiry	Texp	Strike	Bid	Ask	Fwd	CallMid
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	20230222	0.01916496	10.0	NA	2.086124	20.19517	NA
2	20230222	0.01916496	10.5	NA	1.952121	20.19517	NA
3	20230222	0.01916496	11.0	NA	1.995202	20.19517	NA
4	20230222	0.01916496	11.5	NA	1.862784	20.19517	NA
5	20230222	0.01916496	12.0	NA	1.735648	20.19517	NA
6	20230222	0.01916496	12.5	NA	1.613290	20.19517	NA

In [31]:
```
plotVIX <- plotIvols(vixVolData)
```

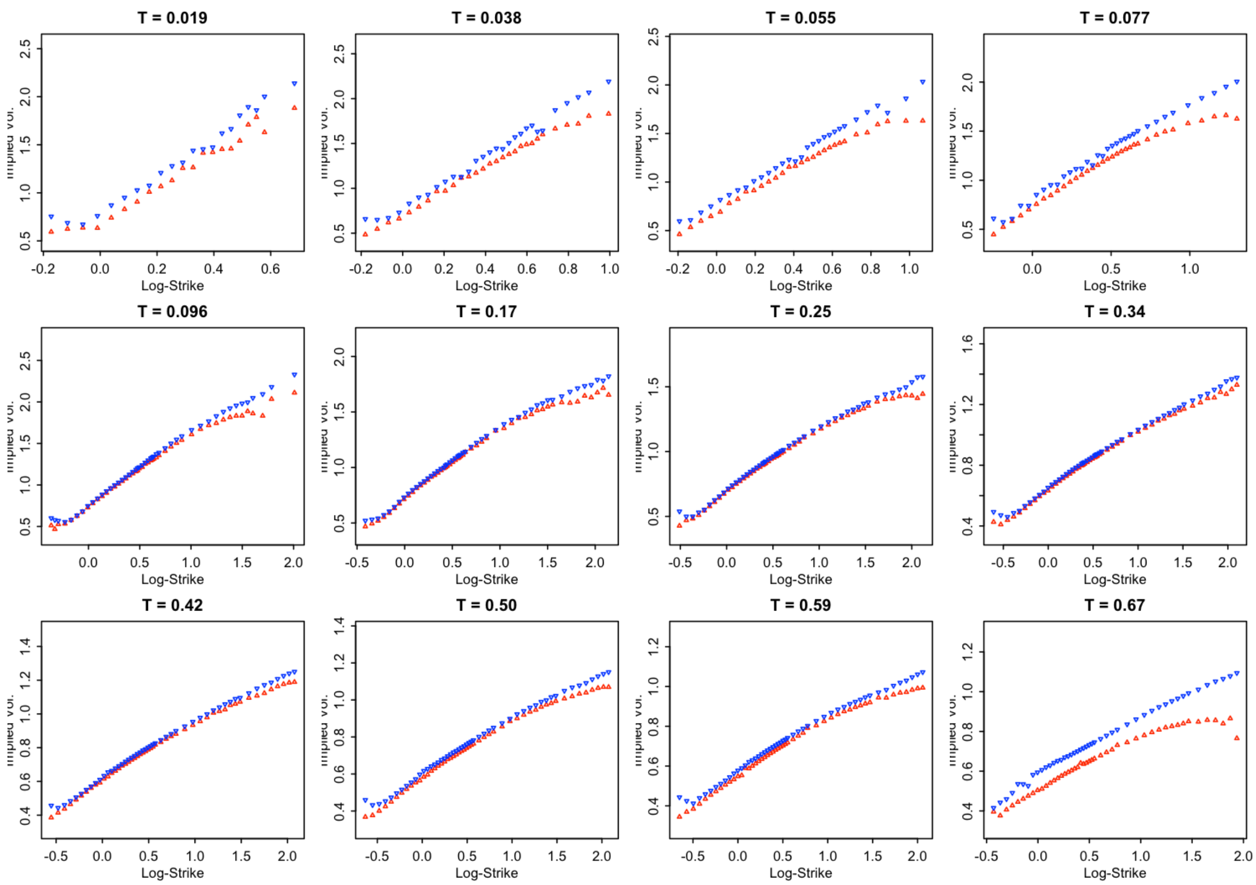

Figure 14: VIX smiles as of February 15, 2023.

VIX futures from put-call parity

In [32]:
```
t.VIX <- plotVIX$expiries
(f.VIX <- unique(vixVolData$Fwd))
```

20.1951741855249 · 20.3550062990049 · 20.6097782540332 · 20.468796922825 · 20.1261648745547 · 21.1644803229044 · 21.6400670851342 · 22.0432737879408 · 22.5994173344876 · 22.608388198057 · 23.0021994712784 · 23.0946832691182

In [33]:
```
plot(t.VIX,f.VIX,pch=20,col=bl,cex=2,type="b")
```

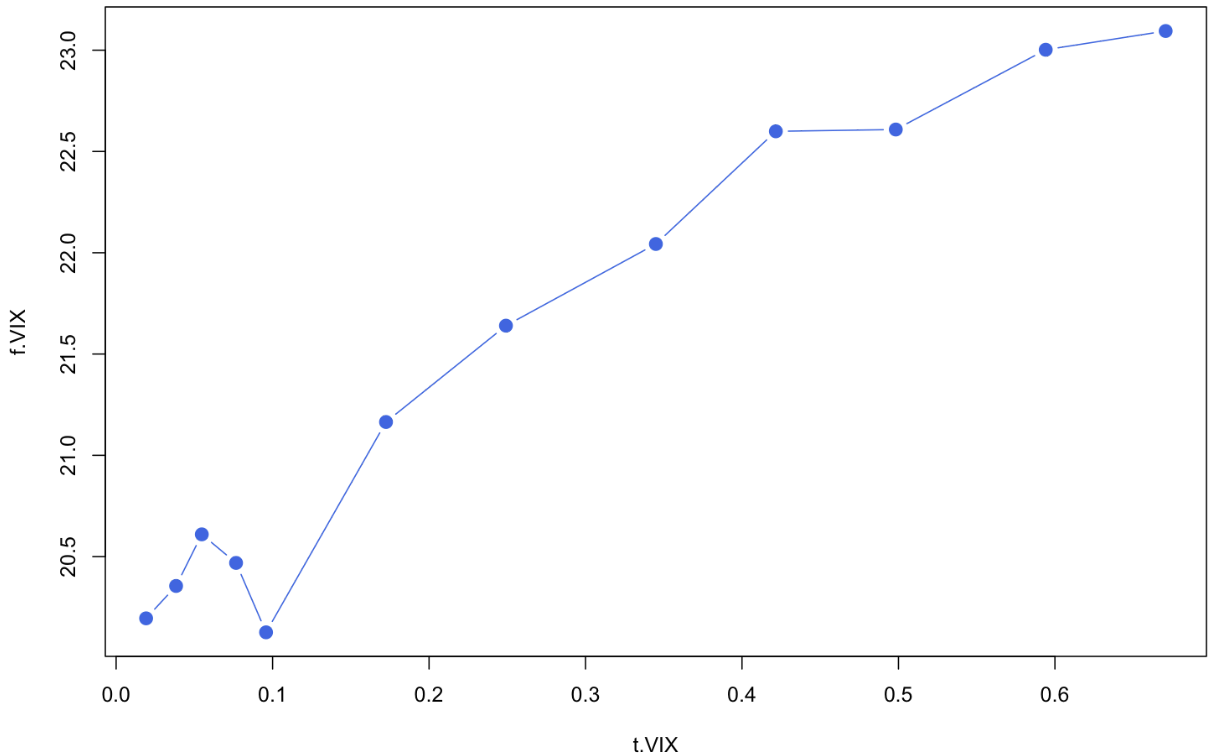

Figure 15: The VIX futures curve from put-call parity

$\mathbb{E}_t\left[\zeta(T)\right]$ from VIX option data

- We can span the payoff of a forward starting variance swap $\zeta(T)=\mathbb{E}_T\left[\int_T^{T+\Delta}V_s\,ds\right]$ using VIX options.
- From the Carr-Madan spannng formula,

$$\mathbb{E}_t\left[\zeta(T)\right]=F_{VIX}^2+2\int_0^{F_{VIX}}P(K)\,dK+2\int_{F_{VIX}}^{\infty}C(K)\,dK.$$

- We need to interpolate and extrapolate out-of-the-money option prices to get the *convexity adjustment*.

Interpolation and extrapolation of VIX smiles

- We perform
  - Monotonic spline interpolation of mid-vols.
  - Extrapolation at constant level.
- Then we integrate the Black-Scholes formula with these vols.

In [34]:
```
vix2 <- function(ivolData)function(slice){

  bidVols <- as.numeric(ivolData$Bid)
  askVols <- as.numeric(ivolData$Ask)
  expDates <- unique(ivolData$Texp)
```

```
  #############################################################################################
  # Interpolate and extrapolate vols for this slice at requested output points

  t <- expDates[slice]
  texp <- ivolData$Texp
  bidVol <- bidVols[texp==t]
  askVol <- askVols[texp==t]
  midVol <- (bidVol+askVol)/2
  f <- (ivolData$Fwd[texp==t])[1]
  k <- log(ivolData$Strike[texp==t]/f) # Plot vs log-strike
  include <- !is.na(bidVol)
  kmin <- min(k[include])
  kmax <- max(k[include])

  # Compute and store interpolated and extrapolated vols
  kIn <- k[!is.na(midVol)]
  volIn <- midVol[!is.na(midVol)]
  volInterp <- function(kout){
    if (kout < kmin){res <- midVol[which(k==kmin)] }
    else if( kout > kmax){res <- midVol[which(k==kmax)] }
    else res <- stinterp(x=kIn,y=volIn, kout)$y
    return(res)
  }
  vixVol <- function(x){sapply(x,volInterp)}
  # Now we use the vectorized function vixVol to compute the convexity adjustemnt
  cTilde <- function(y){exp(y)*BSFormula(1, exp(y), t, r=0, vixVol(y))}
  pTilde <- function(y){exp(y)*BSFormulaPut(1, exp(y), t, r=0, vixVol(y))}
  callIntegral <- integrate(cTilde,lower=0,upper=10)$value
  putIntegral <- integrate(pTilde,lower=-10,upper=0)$value
  res <- f^2*(1+2*(callIntegral+putIntegral))
  return(res)
}
```

In [35]:

```
(e.VIX2 <- Vectorize(vix2(vixVolData))(1:12)/10^4)
```

0.0414436421593004 · 0.0429423797669813 · 0.0447453688499278 · 0.0455250592633412 · 0.0454742634194872 · 0.0540602065324636 · 0.0593273396600168 · 0.0639020315523995 · 0.0687341512776171 · 0.0698957062653717 · 0.0744212383905677 · 0.0767311559552205

Functions to compute the convexity adjustment

In [36]:

```
etaNu <- function(nu,h){
    cH2 <- gamma(3/2-h)/(gamma(h+1/2)*gamma(2-2*h))
    return(2*nu*sqrt(cH2))
    }

fH <- function(theta,h){
    integ <- function(x){((1+theta-x)^(h+1/2)-(1-x)^(h+1/2))^2}
    tmp <- integrate(integ, lower=0,upper=1)$value
    return(tmp/theta^2*(2*h)/(h+1/2)^2)
    }

varLogPsi <- function(tau, delta, h, nu){etaNu(nu,h)^2*tau^(2*h)*fH(delta/tau,h)}

convAdj.raw <- function(tau, delta, h, nu){exp(-varLogPsi(tau, delta, h, nu)/8)}
convAdj <- Vectorize(convAdj.raw,vectorize.args="tau")
```

Function to compute VIX futures given H and ν

In [37]:

```
e.VIX <- function(paramvec)function(t.VIX,e.VIX2){

    H <- paramvec[1]
    eta <- paramvec[2]
```

```
    conv.adj <- convAdj(t.VIX, delta=1/12, H, eta)
    return(sqrt(e.VIX2) * conv.adj * 100)
}
```

Fix dates for optimization

In [38]:
```
eVIX <- function(paramvec){

    e.VIX <- e.VIX(paramvec)(t.VIX,e.VIX2)
    return(e.VIX)

}

eVIX(c(.2,.6))
```

20.2091390147171 · 20.457329893908 · 20.7977286657498 · 20.8784895162704 · 20.7892775905226 · 22.389710052101 · 23.2257302822626 · 23.8581403960556 · 24.5673339020038 · 24.6142080585121 · 25.2124237273954 · 25.4624172549922

Compare with actual futures curve.

In [39]:
```
f.VIX
```

20.1951741855249 · 20.3550062990049 · 20.6097782540332 · 20.468796922825 · 20.1261648745547 · 21.1644803229044 · 21.6400670851342 · 22.0432737879408 · 22.5994173344876 · 22.608388198057 · 23.0021994712784 · 23.0946832691182

Optimize

In [40]:
```
obj <- function(paramvec){

    eVIX.model <- eVIX(paramvec)
    return(sum((eVIX.model-f.VIX)^2)*1e6)
}
```

In [41]:
```
(res.20230215.VIX <- optim(c(.3,.15),obj,method="L-BFGS-B",lower=c(0.0001,0.01),upper=c(.49,1(
```

$par	0.185399648790267 · 0.916948255714096
$value	337483.626861749
$counts	**function:** 28 **gradient:** 28
$convergence	0
$message	'CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH'

Plot actual and fitted VIX futures

In [42]:
```
plot(t.VIX,f.VIX,pch=20,col=bl,cex=2,ylim=c(20,23.5))
points(t.VIX,eVIX(res.20230215.VIX$par),pch=20,col=pk,cex=1)
```

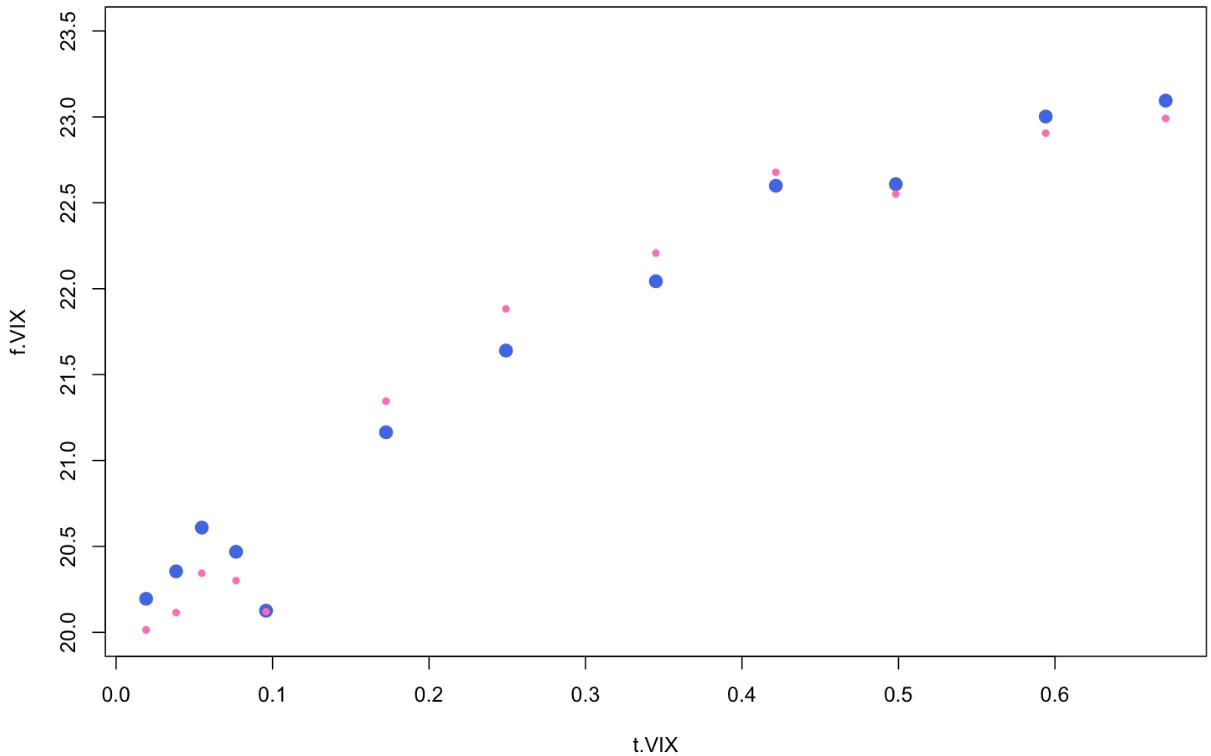

Figure 16: Fit of the rough Bergomi VIX approximation to the observed VIX futures curve. VIX formula in blue; actual VIX futures in pink.

Summary

- The VIX estimate gives $H = 0.185$.

NOw draw smiles with $H = 0.185$

In [43]:
```
params.rBergomi.VIX <- list(al=0.685, eta=1.7, rho=-0.8)
xiCurve <- xi.curve.fast # We use the instantiated curve for speed
```

In [44]:
```
t0<-proc.time()

#number of iterations
iters<- max(1,floor(paths/1000))

#setup parallel backend
cl.num <- detectCores() # This number is 8 on my MacBook Pro
cl<-makeCluster(cl.num)
registerDoParallel(cl)

#loop
ls <- foreach(icount(iters),.packages = "stinepack") %dopar% {
        hybridSchemeRL.S(params.rBergomi.VIX,xiCurve)(paths=1000, steps=steps, expiries=expir
        }

stopCluster(cl)
mcMatrix.VIX <- do.call(cbind, ls) #Bind all of the submatrices into one big matrix

print(proc.time()- t0)
```

```
   user  system elapsed 
  0.144   0.068   8.432 
```

Plot actual and rough Bergomi smiles

In [45]:
```
res.plot.VIX <- plotIvolsMC(ivolData,mcMatrix=mcMatrix.VIX,plot=F)
```

In [46]:
```
plot(res.plot.VIX$expiries,res.plot.VIX$atmSkew,pch=20,col=bl,xlab="Expiry",ylab="ATM skew")
lines(res.plot.VIX$expiries,res.plot.VIX$atmSkewMC,col=gr,lwd=2)
```

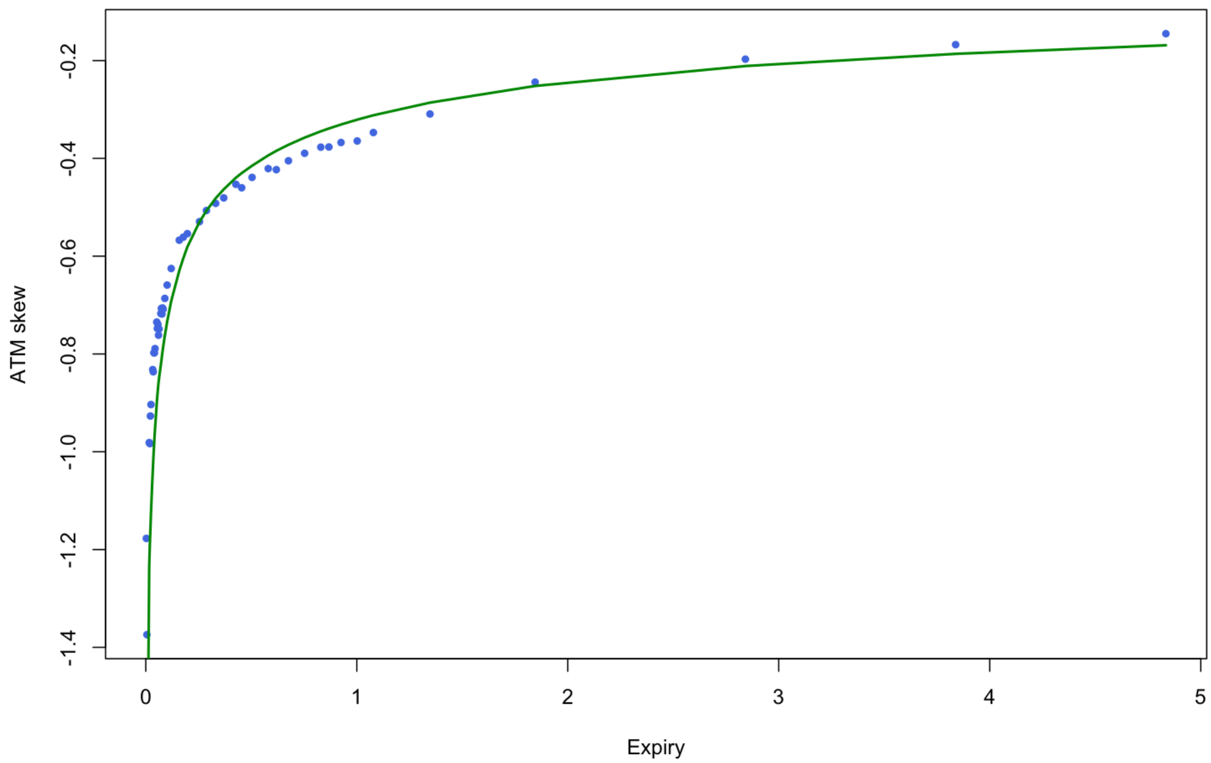

Figure 17: Actual (in blue) vs fitted (in green) SPX ATM skew.

Compare smiles with the two choices of H

In [47]:
```
res.plot4 <- plotIvolsMC2(ivolData,mcMatrix=mcMatrix.VIX,mcMatrix2=mcMatrix2023,,slices= c(2,
```

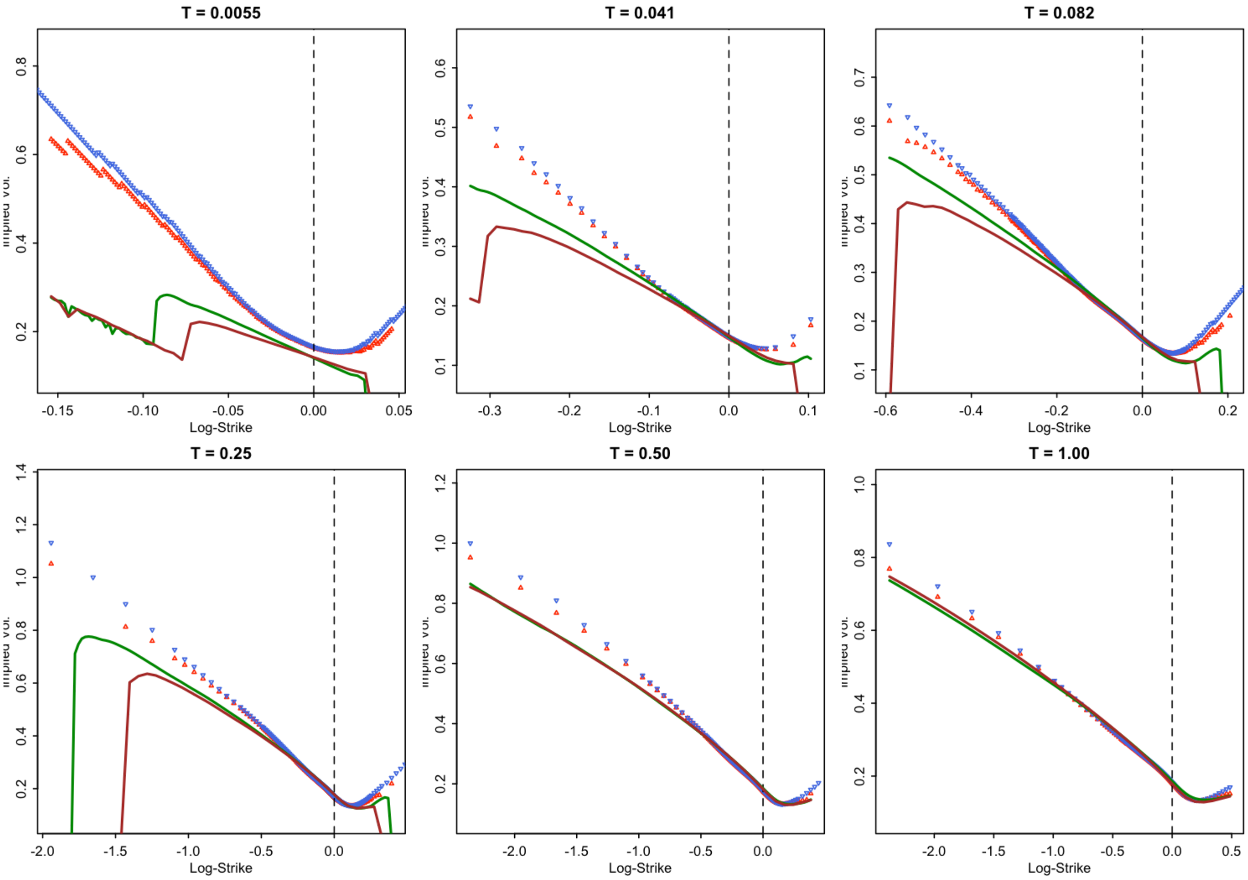

Figure 18: Six rough Bergomi smiles. Green is with parameters `params.rBergomi.VIX`; brown s with parameters `params.rBergomi.skew`.

Rough Bergomi parameters under $\mathbb{P}$ and under $\mathbb{Q}$

- We might wonder whether implied model parameters are consistent with historical parameters.
- It is shown in [Bayer, Friz and Gatheral][2] that the volatility of volatility parameter η in the rough Bergomi model and the volatility of volatility ν in the historical time series should be related as follows.

$$\tilde{\eta} := \eta\,\sqrt{2\,H} = 2\,\nu\,C_H$$

with

$$C_H = \sqrt{\frac{2\,H\,\Gamma(3/2-H)}{\Gamma(H+1/2)\,\Gamma(2-2\,H)}}.$$

Parameter estimates under $\mathbb{Q}$

In Section 5.2 of [Bayer, Friz and Gatheral][1], parameter guesses for the SPX implied volatility surface on two particular dates in history are given as follows:

Date	H	η	$\tilde{\eta}$
February 4, 2010	0.07	1.9	0.7109
August 14, 2013	0.05	2.3	0.7273

- Estimates of $\tilde{\eta}$ seem more stable than estimates of η and H separately.
- We observe the same phenomenon when estimating ν and H from historical RV data.
  - Estimates of the product $\nu\,\sqrt{H}$ are more stable than estimates of the two parameters separately.

Parameter estimates under $\mathbb{P}$

- From our analysis of the SPX realized variance time series in Lecture 1, we estimated

$$H \approx 0.166, \quad \nu \approx 0.302.$$

- Plugging these estimates into the formula (from above)

$$\tilde{\eta}_1 = 2\,\nu\,\sqrt{\frac{2\,H\,\Gamma(3/2-H)}{\Gamma(H+1/2)\,\Gamma(2-2\,H)}} \approx 0.268.$$

In [48]:
```
h.est <- 0.166
nu.est <- 0.302
(nu.tilde <- 2*nu.est*sqrt(2*h.est*gamma(3/2-h.est)/gamma(h.est+1/2)*gamma(2-2*h.est)))
```

0.268425542130751

- Seemingly inconsistent with the implied estimate of around 0.72.

In [49]:
```
nu.tilde*252^h.est
```

0.672134717914495

- However, the historical estimate is in daily terms and the implied estimate in annualized terms.
- To convert, we need to multiply the historical estimate by the annualization factor $(252)^H$, to get

$$\tilde{\eta} \approx \tilde{\eta}_1 \times (252)^H = 0.67.$$

  - Historical and implied estimates are consistent.

Rough volatility and long memory

- In [Bennedsen, Lunde and Pakkanen][4], the authors show how we can both have our cake and eat it by choosing different kernels.
- In particular, with appropriate choices of γ and β the kernel

$$\kappa(\tau) = \frac{1}{\tau^\gamma\,(1+\tau)^\beta}$$

  generates a model that exhibits both rough volatility and power-law decay of the autocorrelation function.
  - That is rough volatility plus long memory.

- Models with with more parameters may of course also fit the volatility surface better.

Forecasting the variance swap curve

In [Bayer, Friz and Gatheral][1], we show how to forecast the whole variance swap curve using the variance forecasting formula.

- We show consistency between the volatility forecast under $\mathbb{P}$ and the forward variance curve (under $\mathbb{Q}$) around two of the most dramatic events:
  - The collapse of Lehman Brothers, and
  - The Flash Crash.

Features of the rough Bergomi model

- In Lecture 1, scaling properties of the time series of historical volatility suggested a natural non-Markovian stochastic volatility model under $\mathbb{P}$.

- The simplest specification of $\frac{d\mathbb{Q}}{d\mathbb{P}}$ gives the rough Bergomi model, a non-Markovian generalization of the Bergomi model.

  - The history of the Brownian motion $\{W_s, s < t\}$ required for pricing is encoded in the forward variance curve, which is observed in the market.
  - Efficient computations are possible using the hybrid BSS scheme.
- Rough Bergomi is easy to simulate using the hybrid-BSS scheme.

- Rough Bergomi is a lognormal model and thus has reasonable dynamics.

- However, rough Bergomi gives flat VIX smiles.

More rough volatility models

This form suggests many other rough volatility models of the form

$$\begin{aligned} \frac{dS_t}{S_t} &= \sqrt{\xi_t(t)}\, dZ_t \\ d\xi_t(u) &= \lambda(\xi)\, \kappa(u-t)\, dW_t \end{aligned}$$

where both the function λ and the kernel κ depend on the model.

- As long as $\kappa(\tau) \sim \tau^{-\gamma}$ as $\tau \to 0$, the model will be rough in the sense that sample paths of instantaneous variance will be Hölder continuous with exponent $H = \frac{1}{2} - \gamma$.

The rough Heston model

By considering the limit of a simple Hawkes process-based model of order flow, [Jaisson and Rosenbaum][9] and [El Euch, Fukasawa and Rosenbaum][5] derive a rough Heston model. The equation for variance in this model takes the form

$$V_u = \theta_t(u) - \frac{1}{\Gamma(\alpha)} \int_t^u (u-s)^{\alpha-1} \lambda\, V_s\, ds + \frac{1}{\Gamma(\alpha)} \int_t^u (u-s)^{\alpha-1} \nu \sqrt{V_s}\, dW_s.$$

where $\alpha = H + \frac{1}{2}$.

- $H \in (0, \frac{1}{2}]$ is the Hurst exponent of the volatility, $\lambda > 0$ is the mean reversion parameter, $\eta > 0$ is the volatility of volatility parameter.

- The function θ is assumed to be continuous and represents a time-dependent mean reversion level.

- The rough Heston model generalizes the classical Heston model which is recovered when $H = 1/2$.

Forward variance in the rough Heston model ($\lambda = 0$)

- We will consider only the special case $\lambda = 0$. In this case, $\xi_t(u) = \mathbb{E}_t\left[V_u\right] = \theta_t(u)$.
- It follows that

$$V_u = \xi_t(u) + \frac{\nu}{\Gamma(\alpha)} \int_t^u (u-s)^{\alpha-1} \sqrt{V_s}\, dW_s.$$

- Also

$$V_u = \xi_{t+h}(u) + \frac{\nu}{\Gamma(\alpha)} \int_{t+h}^u (u-s)^{\alpha-1} \sqrt{V_s}\, dW_s.$$

The rough Heston model with $\lambda = 0$ in forward variance form

Subtracting these two equations gives

$$\xi_{t+h}(u) - \xi_t(u) = \frac{\nu}{\Gamma(\alpha)} \int_t^{t+h} (u-s)^{\alpha-1} \sqrt{V_s}\, dW_s.$$

Taking the limit $h \to 0$, we obtain

$$d\xi_t(u) = \frac{\nu}{\Gamma(\alpha)}\, (u-t)^{\alpha-1}\, \sqrt{V_t}\, dW_t,$$

the rough Heston model in forward variance form.

The rough Heston model in forward variance form ($\lambda \geq 0$)

Let $\phi(\tau) = \frac{1}{\Gamma(\alpha)}\, \tau^{\alpha-1}$ and $\kappa(\tau) = \tau^{\alpha-1}\, E_{\alpha,\alpha}(-\lambda \tau^\alpha)$. Then

$$\phi - \kappa = \lambda\, (\phi \star \kappa).$$

That is, ϕ is the λ-resolvent of κ. To check this, take Laplace transforms:

$$\frac{1}{p^\alpha} - \frac{1}{p^\alpha + \lambda} = \lambda\, \frac{1}{p^\alpha}\, \frac{1}{p^\alpha + \lambda}.$$

The rough Heston model in forward variance form ($\lambda \geq 0$)

Consider once again the erough Heston model of El Euch and Rosenbaum:

$$V_u = \theta_t(u) - \frac{\lambda}{\Gamma(\alpha)} \int_t^u (u-s)^{\alpha-1}\, V_s\, ds + \frac{1}{\Gamma(\alpha)} \int_t^u (u-s)^{\alpha-1} \nu \sqrt{V_s}\, dW_s.$$

Write this formally as:

$$V = \theta - \lambda\, (\phi \star V) + \nu\, (\phi \star \sqrt{V}\, dW).$$

Convolve with κ to get

$$(\kappa \star V) = (\kappa \star \theta) - \lambda\, (\kappa \star \phi \star V) + \nu\, (\kappa \star \phi \star \sqrt{V}\, dW).$$

But $\phi - \kappa = \lambda\,(\phi \star \kappa)$ so

$$(\kappa \star V) = (\kappa \star \theta) - ((\phi - \kappa) \star V) + \frac{\nu}{\lambda}\,((\phi - \kappa) \star \sqrt{V}\,dW).$$

Thus

$$0 = -\lambda\,(\kappa \star \theta) + \lambda\,(\phi \star V) - \nu\,(\phi \star \sqrt{V}\,dW) + (\kappa \star \sqrt{V}\,dW).$$

Add to the first equation to get

$$V = \theta - \lambda\,(\kappa \star \theta) + \nu\,(\kappa \star \sqrt{V}\,dW).$$

Thus, the drift term is eliminated and we must have $\xi = \theta - \lambda\,(\kappa \star \theta)$.

The rough Heston model in forward variance form ($\lambda \geq 0$)

Writing out the last equation in full,

$$V_u = \xi_t(u) + \nu \int_t^u \kappa(u-s)\,\sqrt{V_s}\,dW_s.$$

With the same argument as before, we get the rough Heston model in forward variance form:

$$d\xi_t(u) = \nu\,\kappa(u-t)\,\sqrt{V_t}\,dW_t,$$

with $\kappa(\tau) = \tau^{\alpha-1}\,E_{\alpha,\alpha}(-\lambda\tau^\alpha)$.

Non-Markovianity of the rough Heston model

- Note that the limit $u \to t$ of the rough Heston model makes no sense.
  - This reflects the fact that the rough Heston model is not Markovian.
    - There is no SDE for V_t and no corresponding PDE.
  - On the other hand, we can write an SDE for each $\xi_t(u)$, $u > t$.
    - We can even apply Itô's Lemma!
- The rough Heston model is Markovian in the infinite-dimensional forward variance curve $\xi_t(u),\ u > t$.

Features of the rough Heston model

- The rough Heston model, as we will see in Lecture 3, is very tractable, at least with $\lambda = 0$.
  - Arguably more tractable than the classical Heston model.
- The rough Heston model arises as a limit of a simple Hawkes process-based model of order flow.
- The model is harder to simulate than rough Bergomi, but not so hard (see Lecture 4).
- However, since the model is affine, dynamics are not consistent with observation.
  - VIX smiles are negatively sloped, totally inconsistent with observation!

Dynamics of the volatility surface: Model dependence

- All rough stochastic volatility models have essentially the same implications for the shape of the volatility surface.
- At first it might therefore seem that it would be hard to differentiate between models.

- That would certainly be the case if we were to confine our attention to the shape of the volatility surface today.

- If instead we were to study the dynamics of the volatility skew – in particular, how the observed volatility skew depends on the overall level of volatility, we would be able to differentiate between models.
- As explained in [The Volatility Surface][9] , we expect the ATM volatility skew to be roughly independent of the ATM volatility in a lognormal model such as rough Bergomi.
- In Figure 4, we see how the ATM skew varies with ATM volatility under rough Bergomi and rough Heston, and compare with empirical estimates.

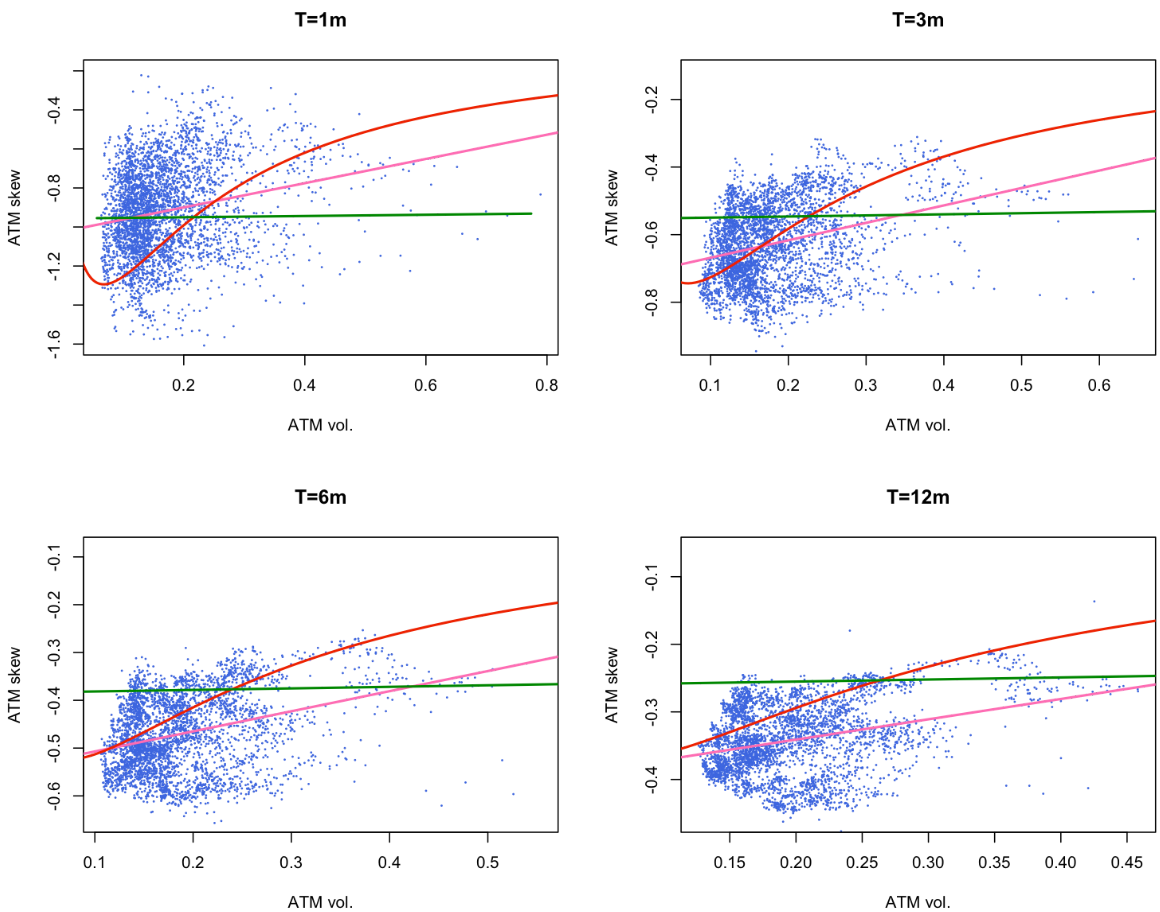

Figure 19: Blue points are empirical 3-month ATM volatilities and skews (from Sep-2008 to today); a regression line in pink; the green line is the rough Bergomi computation with the above parameters; the red line is rough Heston.

The quadratic rough Heston model

The quadratic rough Heston (QRH) model of [QRH][9] may be written as

$$\begin{aligned}\frac{dS_t}{S_t} &= -\sqrt{V_t}\,dW_t,\\ V_t &= Y_t^2 + c,\end{aligned} \tag{2}$$

where $c \geq 0$, and

$$Y_t = \bar{Y} + \int_{-\infty}^{t} \kappa(t-s)\,\sqrt{V_s}\,dW_s = \bar{Y} - \int_{-\infty}^{t} \kappa(t-s)\,\frac{dS_s}{S_s} \tag{3}$$

is a weighted average of historical returns and $\kappa(\cdot)$ a kernel function.

Dynamics in martingale form

Taking the conditional expectation of Y_u for $u > t$ gives,

$$Y_u = y_t(u) + \int_{t}^{u} \kappa(u-s)\,\sqrt{V_s}\,dW_s \tag{4}$$

where

$$y_t(u) := \mathbb{E}_t\left[Y_u\right] = \bar{Y} + \int_{-\infty}^{t} \kappa(u-s)\,\sqrt{V_s}\,dW_s.$$

Then, $y_\cdot(u)$ is a martingale and

$$dy_t(u) = \kappa(u-t)\,\sqrt{V_t}\,dW_t.$$

The forward variance curve

From the model definition, for $u > t$, $V_u = y_u(u)^2 + c$ so applying Itô's formula, we have

$$\begin{aligned} \xi_t(u) &= \mathbb{E}_t\left[y_u(u)^2\right] + c \\ &= y_t(u)^2 + \mathbb{E}_t\left[\int_t^u d\langle y_\cdot(u)\rangle_s\right] + c \\ &= y_t(u)^2 + \int_t^u \xi_t(s)\,\kappa(u-s)^2\,ds + c. \qquad (5) \end{aligned}$$

Alternatively,

$$y_t(u)^2 = \xi_t(u) - \int_t^u \xi_t(s)\,\kappa(u-s)^2\,ds - c,$$

- Thus $y_t(u)$ may be easily imputed from the forward variance curve.

The QR Heston forward variance curve

- From the model definition , for $u > t$, $V_u = (Z_u - b)^2 + c = y_u(u)^2 + c$ so applying Itô's Formula,

```
\begin{align}
\xi_t(u):=\eet{V_u} &= \eet{y_u(u)^2}+c \nonumber\\
&= y_t(u)^2 + \eet{\int_t^u\,d\angl{y_{\bullet}(u)}_s}+c\nonumber\\
&= y_t(u)^2 +\int_t^u\,\xi_t(s)\,\kappa(u-s)^2\,ds+c.
\label{eq:xitu}
\end{align}
```

- Alternatively,

$$y_t(u)^2 = \xi_t(u) - \int_t^u \xi_t(s)\,\kappa(u-s)^2\,ds - c,$$

  so, in principle, $y_t(u)$ may be easily imputed from the forward variance curve.

$\xi_t(u)$ from $y_t(u)$

- We thus have a Wiener-Hopf equation for $\xi_t(u)$ whose solution may be written as

$$\xi_t(u) = y_t(u)^2 + c + \int_t^u K(u-s)\,\left[y_t(s)^2 + c\right]\,ds.$$

- The Laplace transform (denoted as $\mathcal{L}$) of the resolvent kernel K is given by

$$\mathcal{L}[K] = \frac{\mathcal{L}[\kappa^2]}{1 - \mathcal{L}[\kappa^2]}.$$

Dynamics of forward variance

- Using that $\xi_\bullet(u)$ is a martingale, we may write the QR Heston model in forward variance form:

$$\begin{aligned} d\xi_t(u) &= 2\,y_t(u)\,dy_t(u) + 2\int_t^u y_t(s)\,dy_t(s)\,K(u-s)\,ds \\ &= 2\,y_t(u)\,\kappa(u-t)\,\sqrt{V_t}\,dW_t + 2\int_t^u y_t(s)\,\kappa(s-t)\,K(u-s)\,ds\,\sqrt{V_t}\,dW_t \\ &= -2\left\{\kappa(u-t)\,y_t(u) + \int_t^u y_t(s)\,\kappa(s-t)\,K(u-s)\,ds\right\}\,\frac{dS_t}{S_t}. \end{aligned}$$

- We see that dS_t and $d\xi_t(u)$ are usually (but not aalways) anti-correlated in the QR Heston model.

Features of the quadratic rough Heston model

- Like the rough Heston model, the QR Heston model has a microstructural foundation.
  - The ZHawkes model of [Blanc et al.][2] which is in turn related to QUARCH (quadratic ARCH).
- Dynamics are approximately lognormal, consistent with observation.
- VIX smiles are positively sloped, again consistent with observation.
- Like the other models, very parsimonious.
  - S_t and $\xi_t(\cdot)$ are state variables.
  - Parameters are c and the parameters of the chosen kernel, such as H and η.

Summary

- We have presented the three most popular rough volatility models.
  - In all of these models, S_t and $\xi_t(\cdot)$ are state variables.
- We showed how to estimate the forward variance curve.

-- We formulated all three models in forward variance form.

- The models are all very parsimonious.
  - All of them fit the SPX surface remarkably well.
- The rough Bergomi model is easy to simulate, with reasonable dynamics, but generates flat VIX smiles.
  - We showed how to estimate rough Bergomi parameters, and checked the resulting smiles.
- Only the QR Heston model generates positively-sloped VIX smiles.

References

1. ^ Christian Bayer, Peter Friz and Jim Gatheral, Pricing under rough volatility, *Quantitative Finance* **16**(6) 887-904 (2016).
2. ^ Pierre Blanc, Jonathan Donier, and Jean-Philippe Bouchaud, uadratic Hawkes processes for financial prices, *Quantitative Finance* **17**(2), 171-188 (2017).
3. ^ Mikkel Bennedsen, Asger Lunde, and Mikko S. Pakkanen, Hybrid scheme for Brownian semistationary processes, *Finance and Stochastics* **21**(4) 931-965, (2017).
4. ^ Mikkel Bennedsen, Asger Lunde, and Mikko S. Pakkanen, Decoupling the short-and long-term behavior of stochastic volatility, *Journal of Financial Econometrics*, **20**(5) 961-1006, (2021).
5. ^ Omar El Euch, Masaaki Fukasawa, and Mathieu Rosenbaum, The microstructural foundations of leverage effect and rough volatility, *Finance and Stochastics* **22** 3241-280 (2018).
6. ^Jim Gatheral, *The Volatility Surface: A Practitioner's Guide* , John Wiley and Sons, Hoboken, NJ (2006).
7. ^Blanka Horvath, Aitor Muguruza and Mehdi Tomas, Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models, *Quantitative Finance* **21**(1) 11-27 (2021).
8. ^ Antoine Jacquier, Claude Martini, and Aitor Muguruza, On VIX Futures in the rough Bergomi model, *Quantitative Finance* **18**(1) 45-61 (2018).
9. ^ Thibault Jaisson and Mathieu Rosenbaum, Rough fractional diffusions as scaling limits of nearly unstable heavy tailed Hawkes processes, *The Annals of Applied Probability* **26**(5) 2860-2882 (2016).
10. ^ Ryan McCrickerd and Mikko S Pakkanen, Turbocharging Monte Carlo Pricing for the Rough Bergomi Model, *Quantitative Finance* **18**(11) 1877-1886 (2018).