A Practical Introduction to Local Stochastic Volatility

for maths mortals like me

Martin Burke

June 5, 2026

Dedicated to Peter Austing and Fabrice Rouah,
who teach with intuition and clarity.

Abstract

A local-volatility model fits today's vanilla surface exactly but makes a specific, empirically wrong prediction about the *future* smile: it flattens. This note traces that defect from the start—where the skew comes from, how diffusion averages it away over a forward-start horizon, and why a stochastic-volatility process does not suffer the same fate—and then builds the cure. Local stochastic volatility (LSV) marries the two: a stochastic variance for realistic dynamics, scaled by a deterministic *leverage function* that restores the exact vanilla fit. We motivate the leverage function through Gyöngy's mimicking theorem, show why calibrating it is a fixed-point problem solved by bootstrapping forward in time, and develop the two work-horse solvers—the forward-PDE (Fokker–Planck) method and the particle (Monte-Carlo) method—in detail, with pseudocode for each. Appendices cover the supporting numerics: explicit versus implicit finite differences and the Craig–Sneyd ADI scheme, Euler–Maruyama time stepping, and kernel-weighted regression. Every smile and density is computed, not drawn by hand.

1 The trade-off that motivates everything

Two parent models sit at opposite corners. **Local volatility** (Dupire [2]) makes instantaneous volatility a deterministic function $\sigma_{\text{loc}}(t, S)$ of time and spot. By construction it reprices every vanilla on the surface; its defect is dynamic. **Stochastic volatility** (Heston, SABR [3, 4]) gives volatility a life of its own—a mean-reverting random process correlated with spot—which produces realistic dynamics, but a handful of parameters cannot bend themselves into the shape of an entire market surface.

So the choice is between *perfect fit with broken dynamics* and *good dynamics with imperfect fit*. Neither is acceptable on an exotics desk, which must be arbitrage-free against hedgeable vanillas *and* price the future-smile-sensitive product in hand. The cleanest symptom of local vol's broken dynamics is the forward smile, and that is what the rest of this note dissects.

Fixing the term "skew." Everything below is a statement about one number, so it is worth defining precisely. The *implied volatility* $\sigma_{\text{BS}}(K, T)$ is the single Black–Scholes volatility that reprices a European option of strike K and maturity T at its market value; at a fixed maturity the map $K \mapsto \sigma_{\text{BS}}(K, T)$ is the *smile*, and were Black–Scholes literally true it would be a horizontal line. The *skew* is the slope of that line. Writing log-moneyness against the forward F as $k = \ln(K/F)$, the formal definition is the at-the-money slope

$$\mathcal{S}(T) \;=\; \left.\frac{\partial\, \sigma_{\text{BS}}(K,T)}{\partial \ln(K/F)}\right|_{K=F}, \tag{1}$$

illustrated in Figure 1. For equity indices $\mathcal{S}(T) < 0$: out-of-the-money puts (low K) carry higher implied vols than out-of-the-money calls, so the curve slopes *down* to the right. A trading desk

quotes the same object discretely as the 90–110 *skew*, $\sigma_{\mathrm{BS}}(0.9F) - \sigma_{\mathrm{BS}}(1.1F)$, the figure that appears in the tables below; a steeper down-sloping smile is a larger 90–110 skew, and a flat smile is zero skew. The whole note asks how this single quantity behaves *today* versus on a future date.

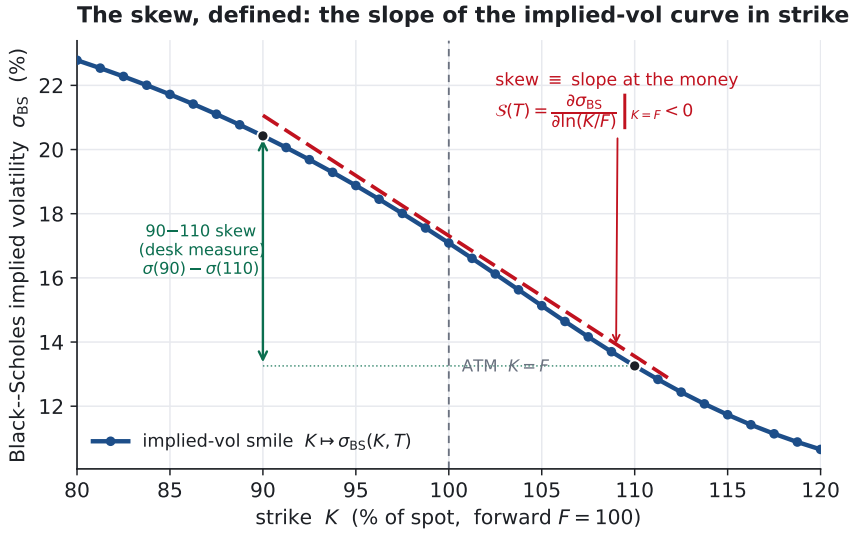

Figure 1: The skew, defined. Implied volatility $\sigma_{\mathrm{BS}}(K, T)$ is plotted against strike; the skew $\mathcal{S}(T)$ is the slope of this curve at the money (red dashed tangent), negative for equities. The desk's 90–110 measure (green) reads the same slope off discretely. The curve shown is the Monte-Carlo one-year smile of the local-volatility model used throughout (90–110 skew = 7.2 vol points).

2 Step 1 — the skew lives in the slope of one curve

Local vol has exactly one mechanism for producing a skew: the dependence of $\sigma_{\mathrm{loc}}(t, S)$ on the spot level. For an equity skew the calibrated surface is downward sloping—high instantaneous vol at low spot, low vol at high spot (Figure 2). The skew is not a feature bolted on top of the model; it *is* the shape of this curve. Whatever skew the model can ever generate, at any date, must come from spot sitting somewhere on it.

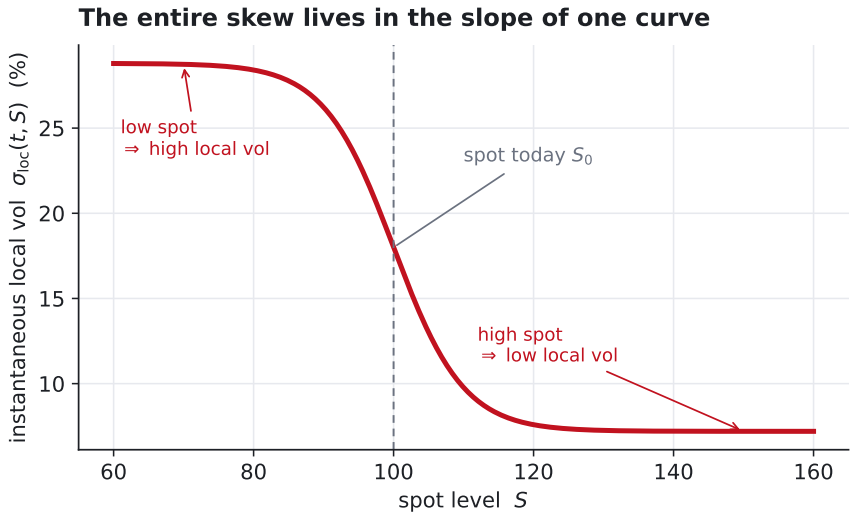

Figure 2: The local-volatility function $\sigma_{\text{loc}}(t, S)$. A downward slope in S is the sole source of the equity skew. A pure power-law (scale-invariant) slope would leave the forward smile unchanged; here the curve *saturates*, breaking scale invariance, which is what allows diffusion to average it away.

3 Step 2 — diffusion averages the level-dependence away

The forward smile is the smile seen from a future date T_1, looking forward to T_2. To form it one must condition on where spot *is* at T_1—and spot at T_1 is not a number but a distribution. Over $[0, T_1]$ the paths diffuse out from S_0, fanning into a spread of arrival levels (Figure 3).

A skew is fundamentally about the correlation between spot moves and vol moves. In local vol that correlation is manufactured purely by spot sliding along the fixed σ_{loc} curve. But by T_1 the paths are scattered across many levels, each sitting on a different part of the curve, so the sharp, deterministic level-dependence that produced a steep skew today is smeared into an average over the diffused distribution. The longer the horizon to T_1, the more diffusion, the more averaging, the weaker the residual skew.

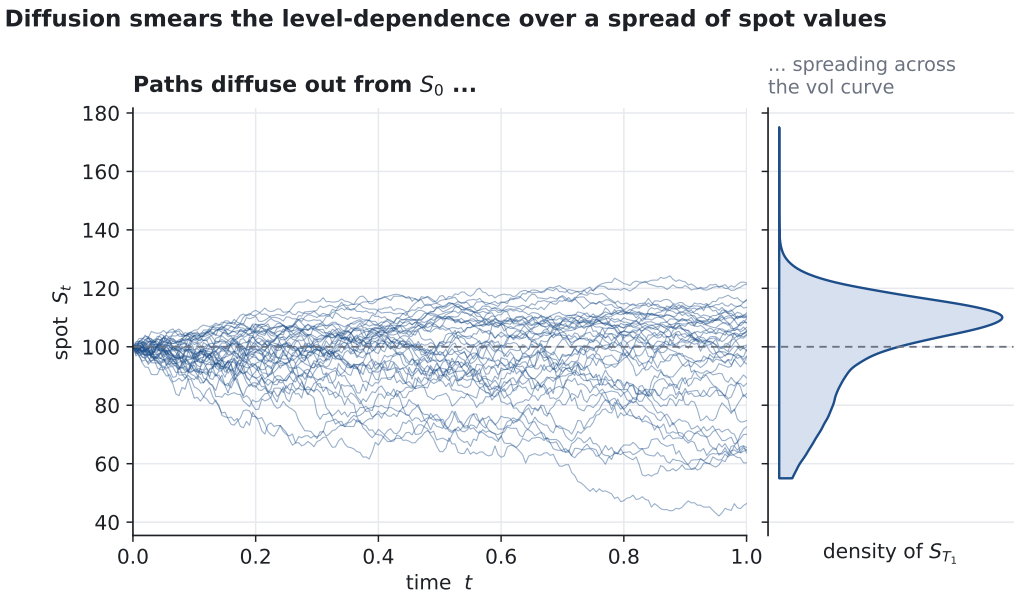

Figure 3: Spot paths diffusing out of S_0 over $[0, T_1]$ (left) and the resulting density of S_{T_1} (right). An option starting at T_1 does not see one point on the vol curve—it sees this whole spread, which dilutes the level-dependence responsible for the skew.

4 Step 3 — the forward smile comes out flat

Today's smile is read directly off the sloped curve with no diffusion in between, so it is steep. The forward smile is read off the *same* mechanism but only after spot has diffused out to T_1, which dilutes the level-dependence. The model therefore predicts a forward smile systematically flatter than the smile observed today (Figure 4). In the Monte-Carlo experiment here the 90–110 skew falls from 7.2 to 3.7 volatility points—roughly a halving—over a one-year forward start.

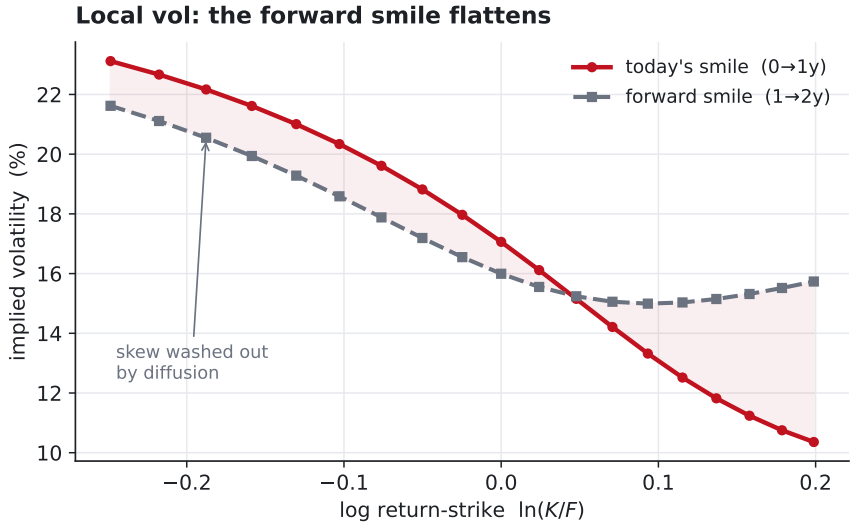

Figure 4: Local-volatility model: today's one-year smile ($0 \to 1$y) versus the one-year forward smile ($1 \to 2$y), both expressed as return smiles for a like-for-like comparison. The forward skew is markedly weaker—the defect that mis-prices forward starts, cliquets and other future-smile-sensitive products.

5 Why stochastic volatility does not flatten, and how LSV uses this

In a stochastic-volatility model the skew is generated by the volatility process itself—a fixed spot/vol correlation ρ and a vol-of-vol ξ that do not decay merely because time has passed. An option starting at T_1 still sees full-strength ρ and ξ, so it still sees a steep skew: nothing is averaged away because the skew-generating mechanism is intrinsic to the vol process rather than parasitic on spot's position along a curve. With parameters tuned to match the local-vol model's *today* skew, the stochastic-vol forward skew stays essentially put while the local-vol one collapses (Figure 5, Table 1).

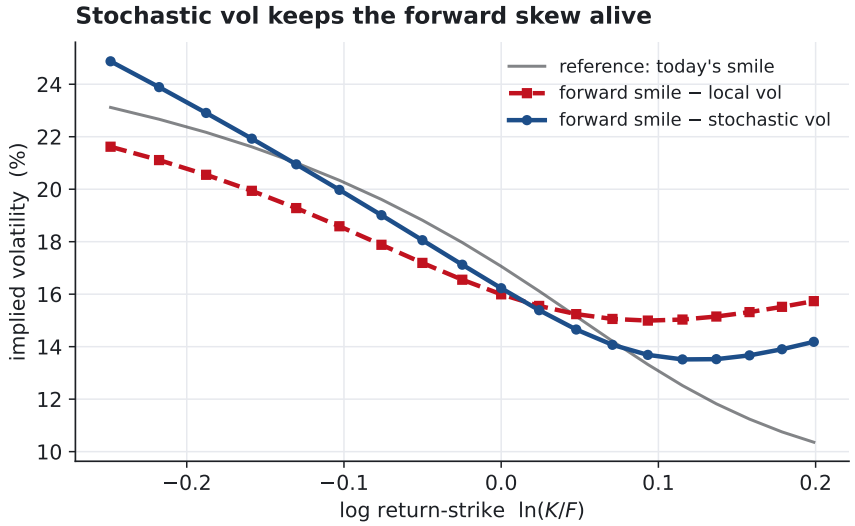

Figure 5: One-year forward smiles. Both models carry the same skew today (grey reference), but only stochastic volatility preserves it forward; the local-volatility forward smile has flattened.

	ATM vol		90–110 skew (vol pts)	
	today	forward	today	forward
Local volatility	17.1	16.0	7.2	3.7
Stochastic vol	15.9	16.2	6.9	6.4

Table 1: Monte-Carlo skew term structure. The two models are tuned to the same *today* skew; only the local-vol forward skew decays.

This is precisely the gap LSV closes. Keep a stochastic variance v_t for the dynamics and attach a deterministic *leverage function* $L(t, S)$ to the spot diffusion:

$$\mathrm{d}S_t = (r - q)\, S_t\, \mathrm{d}t + L(t, S_t)\, \sqrt{v_t}\, S_t\, \mathrm{d}W_t^S, \qquad \mathrm{d}v_t = \kappa\, (\theta - v_t)\, \mathrm{d}t + \xi\, \sqrt{v_t}\, \mathrm{d}W_t^v, \tag{2}$$

the two Brownian motions being correlated, $\mathrm{d}W_t^S\, \mathrm{d}W_t^v = \rho\, \mathrm{d}t$. The variance runs its own mean-reverting process—written here in the Heston/CIR form used throughout this note—in which θ is the long-run variance it reverts to, κ the speed of reversion, ξ the vol-of-vol, and ρ the spot/vol correlation (negative for equities, which is what tilts the joint distribution of (S, v) and generates the skew). The stochastic factor $\sqrt{v_t}$ supplies the non-decaying skew mechanism that keeps the forward smile alive; the deterministic factor $L(t, S_t)$ is a knob, one value for each (time, spot) pair, that we will turn to recover the exact vanilla fit. The only question is what L must be.

Gyöngy's mimicking theorem. The answer is a corollary of a general result of Gyöngy [5] relating a complicated Itô process to a simple *Markovian* one.[1] In a sentence: *any Itô process can be matched, marginal by marginal, by a local-volatility diffusion whose squared local volatility* $\sigma^2_{\text{eff}}(t,S)$ *equals the conditional expectation of the original instantaneous variance.* Here $\sigma_{\text{eff}}(t,S)$ is the deterministic "effective" volatility of the mimicking local-vol model—the single number that, placed at (t,S), makes spot spread at the same average rate as the original process. The two models then share every *marginal distribution of* S_t, that is, the one-dimensional distribution of spot at each individual date t taken on its own (the histogram of S_t across all paths), as opposed to the joint distribution of the entire trajectory.[2]

The intuition is that a European vanilla of maturity T depends only on the marginal density of S_T—not on the path taken, and not on whether volatility was random along the way. And that marginal feels only the *average* squared diffusion coefficient at each level of spot: when a path sits at $S_t = S$, what spreads the density there is not the particular random value of v_t but its average across *all the paths that are at* S *at time* t. That average is the conditional expectation $\mathbb{E}[v_t \mid S_t = S]$: the mean of the instantaneous variance v_t over the ensemble of paths, conditioned on their sharing the same current spot $S_t = S$—i.e. taken over just the vertical column of paths above S in Figure 6, not over the whole population. Replace the random v_t at each (t,S) by this conditional mean and the marginal distribution of S—hence every vanilla price—is unchanged (Figure 6). The projection discards the randomness of volatility but keeps precisely the information vanillas can see.

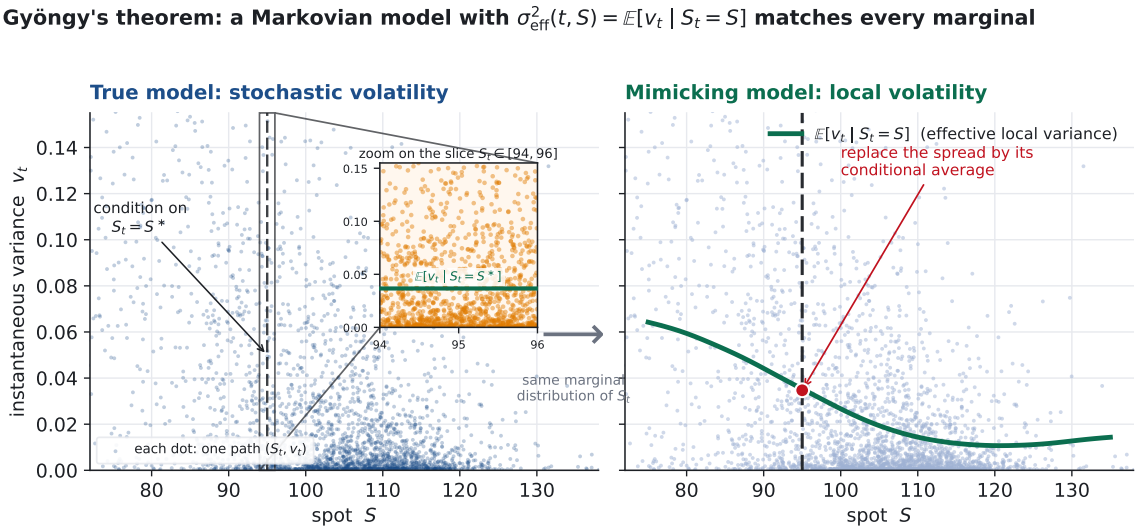

Figure 6: Gyöngy's mimicking theorem. Each point is one simulated path at the snapshot time t, plotted at its current spot S_t (horizontal) and its current *instantaneous* variance v_t (vertical). *Left:* in the true stochastic-volatility model v_t is random. The inset magnifies a thin conditioning window $S_t \in [94, 96]$ about $S^* = 95$: the paths landing there carry a whole vertical spread of variances, and their average is the conditional mean $\mathbb{E}[v_t \mid S_t = S^*]$ (green line). *Right:* doing this at every spot—replacing the spread by its conditional mean $\mathbb{E}[v_t \mid S_t = S]$ (green curve)—defines a deterministic, Markovian model. The theorem says the two share every marginal distribution of S_t, so they price all European vanillas identically.

[1] A model is *Markovian in spot* if its future evolution depends on the present only through the current spot S_t: there is no extra hidden state carrying memory. A local-volatility model, in which spot is driven by a deterministic function of time and the current spot alone, is Markovian in this sense; a stochastic-vol model is *not*, because the variance v_t is a second state variable the future also depends on. The *Markovian projection* is the operation of replacing the non-Markovian model by the Markovian one that keeps the same one-dimensional distributions—collapsing the extra state v_t by averaging over it at each spot.

[2] Matching the marginal of each S_t is far weaker than matching the full path distribution. The mimicking model gets every *vanilla* right—those depend only on a single marginal—but generally misprices path-dependent and forward-starting payoffs, which see the joint distribution across dates. That gap is exactly why the forward smile can differ between two models that share today's vanillas.

Apply the projection to the LSV diffusion (2). Its instantaneous variance is $L^2(t,S)\,v_t$, so the effective squared local volatility Gyöngy produces is $L^2(t,S)\,\mathbb{E}[v_t \mid S_t = S]$. For the model to reprice the entire vanilla surface this must equal Dupire's local volatility $\sigma_{\text{loc}}(t,S)$—the unique local-vol function already calibrated to that surface. Equating the two gives the *leverage identity*, the central equation of LSV:

$$\boxed{L^2(t,S)\,\mathbb{E}[\,v_t \mid S_t = S\,] = \sigma^2_{\text{loc}}(t,S)} \qquad \Longleftrightarrow \qquad L^2(t,S) = \frac{\sigma^2_{\text{loc}}(t,S)}{\mathbb{E}[\,v_t \mid S_t = S\,]}. \tag{3}$$

What the leverage function is. Read the identity as a ratio (Figure 7, left). The numerator $\sigma^2_{\text{loc}}(t,S)$ is the *target*: the squared local vol the market demands at (t,S). The denominator $\mathbb{E}[v_t \mid S_t = S]$ is what the bare stochastic-vol process *delivers* on average when spot is at S. Their ratio $L^2(t,S)$ is the point-by-point correction that bends one onto the other. So the leverage function is a deterministic surface over (t,S) (Figure 7, right) that scales the stochastic volatility *up* where it falls short of the surface ($L > 1$) and *down* where it overshoots ($L < 1$). Two limiting cases pin down its meaning:

- Switch the randomness off ($v_t \equiv 1$). Then $\mathbb{E}[v_t \mid S] = 1$ and $L(t,S) = \sigma_{\text{loc}}(t,S)$: the leverage function *is* the local-vol function, and LSV degenerates to pure Dupire.
- Suppose the stochastic-vol process already reproduced the surface on its own. Then $\mathbb{E}[v_t \mid S] = \sigma^2_{\text{loc}}(t,S)$ and $L \equiv 1$: no leverage is needed, and the model is pure stochastic vol.

The useful regime sits in between: L is a mild, order-one rescaling that contributes exactly the part of the skew the chosen ρ, ξ leave on the table—the bare conditional average $\mathbb{E}[v \mid S]$ is too flat to bend into the target skew by itself, and the leverage makes up the difference (Figure 7, left).

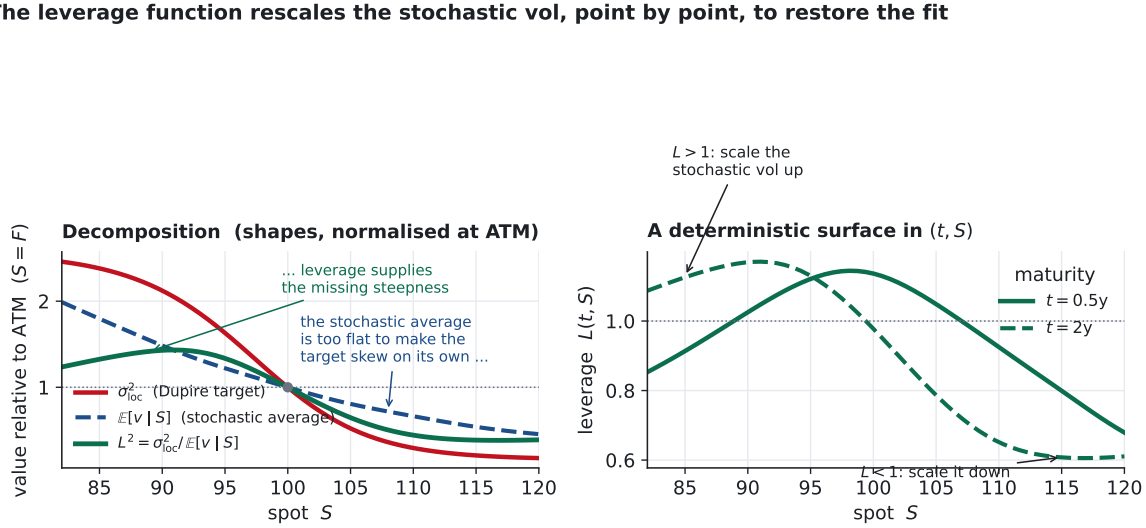

Figure 7: The leverage function. *Left:* the identity as a ratio, with the three curves normalised at the money so their shapes are comparable (in absolute size σ^2_{loc} and $\mathbb{E}[v \mid S]$ are $\sim 30\times$ smaller than L^2). The Dupire target (red) is steep; the stochastic average (blue) is too flat to make that skew alone; the leverage L^2 (green) supplies the missing steepness. *Right:* the resulting leverage $L(t,S)$ is a deterministic surface—shown here as a function of spot at two maturities—above one where the stochastic vol must be scaled up, below one where it must be scaled down. (Illustrative: σ_{loc} is a saturating (tanh) local-vol curve and $\mathbb{E}[v \mid S]$ comes from the Section-5 Heston distribution.)

Because L always restores the fit, the stochastic-vol parameters are *not* pinned down by vanillas: essentially any reasonable $(\rho, \xi, \dots)$ can be made to reprice the surface once L is solved

for. That residual freedom—a mixing weight running from pure local vol ($L = \sigma_{\text{loc}}$, no stochastic vol) to as much stochastic vol as one cares to add—is what one calibrates to the forward smile and other exotics, and it is where the model risk lives.

6 Solving the fixed point

At first glance the leverage identity (3) looks like something you can simply evaluate: take the known Dupire surface $\sigma^2_{\text{loc}}(t, S)$, divide by the denominator $\mathbb{E}[v_t \mid S_t = S]$, and read off L. It is not that simple, because the denominator is not a quantity we already have. The conditional average $\mathbb{E}[v_t \mid S_t = S]$ is a feature of the joint distribution of spot and variance (S_t, v_t), and that distribution is whatever comes out of *running the model*—evolving the SDE (2) forward. But L sits *inside* that SDE, multiplying the spot diffusion, so the distribution the model produces already depends on L. We are going in a circle: to get L we need $\mathbb{E}[v_t \mid S_t = S]$; to get that we need the joint distribution; and to get the distribution we already need L (Figure 8).

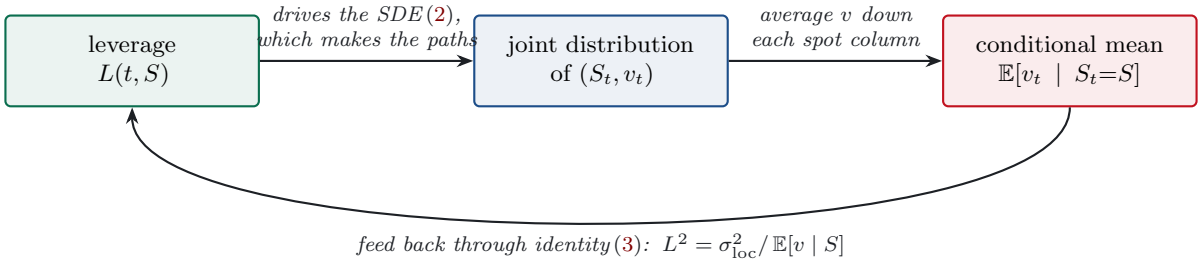

Figure 8: Why the leverage identity is a fixed point, not a formula. To compute L you need the conditional mean $\mathbb{E}[v \mid S]$; to get that you need the joint distribution of (S, v); but that distribution comes from running the model, whose spot SDE already contains L. The three quantities close into a loop, so L has to be solved for self-consistently. The escape (below) is that the loop need only be closed one time-step at a time.

This is why (3) is a *fixed-point* equation rather than an explicit formula: we are after a leverage function that reproduces itself when sent once around the loop. The technical name for an SDE of this kind—one whose coefficients depend on the *distribution* of its own solution, not merely on the current value of the state—is a *McKean–Vlasov* SDE. Two solvers dominate in practice, and neither cracks it in closed form: the *forward-PDE* (Fokker–Planck) method of Section 7, which carries the distribution as a deterministic density on a grid and evolves it with a PDE, and the *particle* (Monte-Carlo) method of Section 8, which carries it as a cloud of simulated paths and evolves them by simulation. Both escape the circularity the same way: $L(t, \cdot)$ is only ever needed to push the model from t to $t + \mathrm{d}t$, so the loop can be broken *locally in time*.

Concretely, chop time into steps of length Δt and carry the joint distribution of (S_t, v_t) forward one step at a time. At the start of a step we already hold the time-t distribution—which is all the identity needs. Read the conditional mean $\mathbb{E}[v_t \mid S_t = S]$ off it, form the leverage on that slice through the identity (3), $L(t, S) = \sigma_{\text{loc}}(t, S) \big/ \sqrt{\mathbb{E}[v_t \mid S_t = S]}$, then *freeze* this $L(t, \cdot)$ and advance every path one Euler–Maruyama step (Appendix B),

$$\begin{aligned} S_{t+\Delta t} &= S_t + (r - q)\, S_t\, \Delta t + L(t, S_t)\, \sqrt{v_t}\, S_t\, \sqrt{\Delta t}\; Z^S_t, \\ v_{t+\Delta t} &= v_t + \kappa\, (\theta - v_t)\, \Delta t + \xi\, \sqrt{v_t}\, \sqrt{\Delta t}\; Z^v_t, \end{aligned} \tag{4}$$

with Z^S_t, Z^v_t standard normals of correlation ρ. Stepping the whole distribution this way yields the distribution at $t + \Delta t$; the clock advances and the cycle repeats (Figure 9). Freezing L is exactly what breaks the loop of Figure 8: over $[t, t + \Delta t]$ the leverage is built *only* from the

already-known time-t distribution, so within the step it is an ordinary known coefficient, not an unknown depending on the step's own output. (In practice the variance is advanced by a positivity-preserving scheme—full truncation or QE—rather than the plain Euler step shown above; see Section 8.)

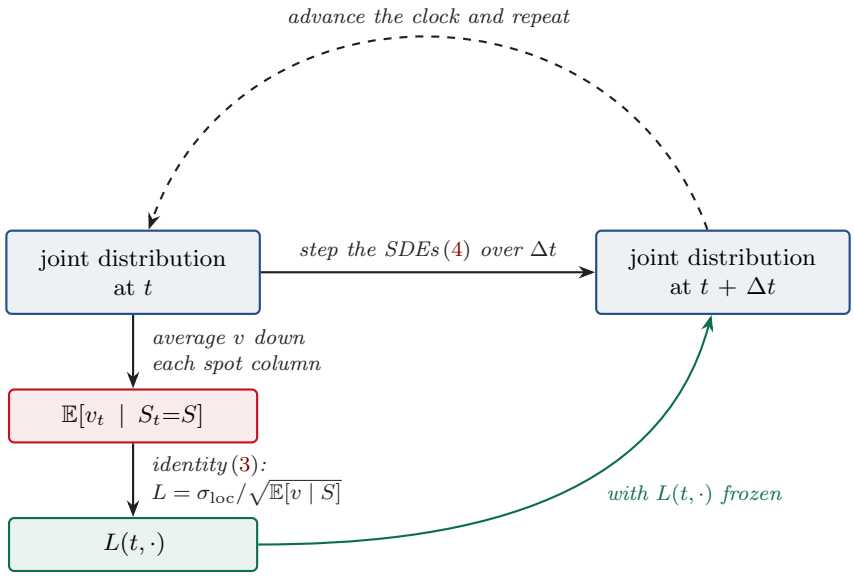

Figure 9: One step of the bootstrap. The leverage is rebuilt from the *current* distribution—average the variance down each spot column to get $\mathbb{E}[v_t \mid S_t{=}S]$, then divide it into σ^2_{loc} via the identity (3)—then *frozen* (green) and used to push the whole distribution from t to $t + \Delta t$ through the discretised SDEs (4). The new distribution feeds the next step, so the leverage is calibrated slice by slice as the clock advances.

Everything else—the bootstrap, the identity (3), the conditional expectation—is common to both solvers; they part company only in how they *represent* the advancing distribution (Figure 10), and the next two sections take each in turn.

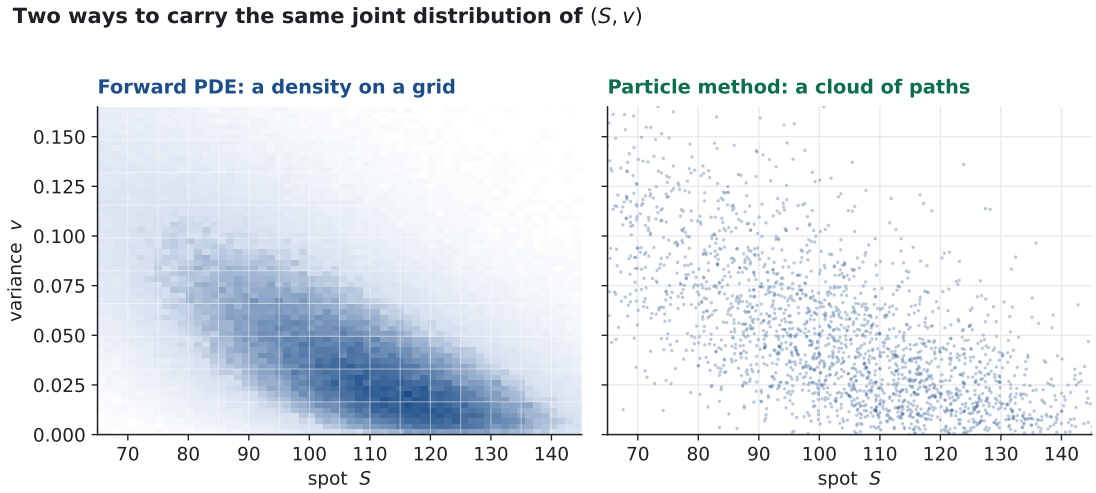

Figure 10: One snapshot of the joint distribution of (S_t, v_t) at a single, fixed time t—the very object the bootstrap of Figure 9 carries forward one step at a time—shown two ways. The downward tilt is the negative spot–variance correlation ($\rho < 0$): low spot goes with high variance. The forward-PDE method represents this distribution as a density on the left grid; the particle method represents it as the cloud of points on the right. (Illustrated on a Heston joint distribution at $t = 1$.)

7 The forward PDE (Fokker–Planck) method

Think of the joint density $p(t, S, v)$ as a sheet of probability mass laid over the (S, v) plane, and push it forward in time. It obeys the Fokker–Planck (Kolmogorov forward) equation—a conservation law stating that probability is neither created nor destroyed, only transported by the drift and spread by the diffusion:

$$\partial_t p = -\,\partial_S[\mu_S\, p] \;-\; \partial_v[\mu_v\, p] \;+\; \tfrac{1}{2}\,\partial_{SS}[L^2 v\, S^2 p] \;+\; \partial_{Sv}[\rho\,\xi\, L\, v\, S\, p] \;+\; \tfrac{1}{2}\,\partial_{vv}[\xi^2 v\, p]. \tag{5}$$

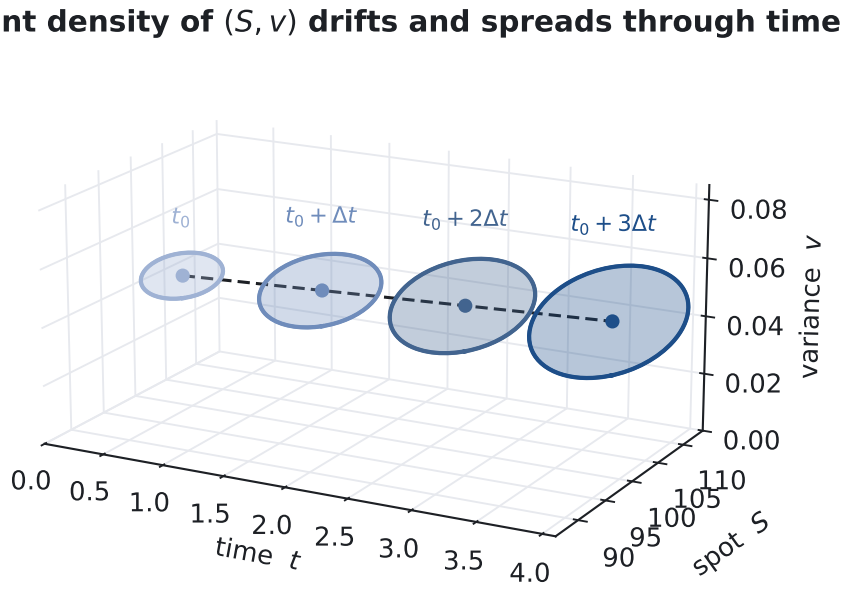

Figure 11: What equation (5) does to the density, schematically, with time as the third axis. Each tilted ellipse is a contour of the two-dimensional joint density of (S, v) at one time slice. As t advances the density *drifts*—its centre tracks the dashed line (the first-order terms transport the mass)—and *spreads*—the ellipse grows as the second-order terms diffuse the mass, the peak falling to keep total probability fixed—while its tilt is the negative spot–variance correlation. The forward-PDE solver evolves exactly this object, the density on the (S, v) grid, forward one time step at a time.

Discretise the plane into a grid and (5) becomes a large, sparse linear update that marches p one step. The pay-off is that once a slice is solved you hold the *entire* density, so the conditional expectation is a direct quadrature: for each spot column, average the variance against the density down that column (Figure 12),

$$\mathbb{E}[v_t \mid S_t = S] = \frac{\int v\, p(t, S, v)\, \mathrm{d}v}{\int p(t, S, v)\, \mathrm{d}v}. \tag{6}$$

Feed this into the leverage identity (3) to obtain $L(t, S)$ on the grid, freeze it into the coefficients of the next step, and march on.

The method is deterministic and noise-free: the density is known everywhere, so the conditional expectation and any greeks come out smooth, and the vanilla fit can be made very accurate. The cost is that this is a genuinely two-dimensional PDE. On a finite-difference grid the spatial operator couples each node to its neighbours in both S and v, and an explicit time march is throttled by stability limits, so one steps implicitly. Rather than invert the full two-dimensional system at every step, the standard choice is an *alternating-direction implicit*

(ADI) scheme—Craig–Sneyd or Hundsdorfer–Verwer—which splits each step into a tridiagonal sweep along S and another along v, carrying the mixed ∂_{Sv} term (the ρ coupling) explicitly; each step is then a handful of cheap one-dimensional solves rather than one large two-dimensional one (Appendix A sets the Craig–Sneyd scheme out in full). Even so, work scales as (spot grid) × (variance grid) × (time steps), and every extra factor (stochastic rates, a second asset) adds a whole grid dimension, so the curse of dimensionality bites quickly. The wings are the delicate part: where the density is thin the conditional expectation is a ratio of two tiny numbers, and the boundary in v near zero needs care, especially when the Feller condition[3] is violated and mass piles up at $v = 0$. (Lipton's work [6] is the standard reference.)

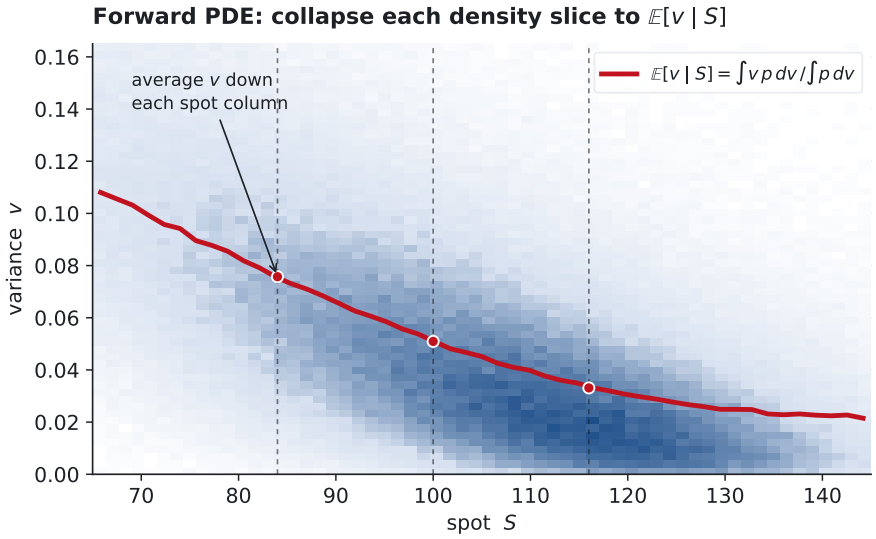

Figure 12: The forward-PDE method's defining step. Holding the full joint density, the conditional expectation $\mathbb{E}[v \mid S]$ (red) is obtained by averaging the variance down each spot column—a quadrature against the density. This curve, divided into σ^2_{loc}, gives the leverage on the grid.

Algorithm 1 collects the forward-PDE calibration as a single forward sweep: at each time level read the conditional mean off the current density, set the leverage from the identity, freeze it, and march the density one ADI step.

Input: Dupire surface $\sigma_{\text{loc}}(t, S)$; variance parameters $\kappa, \theta, \xi, \rho$; grids $\{S_i\}$, $\{v_j\}$; time levels $t_0 < \cdots < t_M$ (step Δt); floor ϵ

Output: leverage $L(t_n, S_i)$ on the grid

initialise the joint density $p(t_0, S_i, v_j)$ concentrated near (S_0, v_0)
for $n = 0, 1, \ldots, M-1$ **do**
  for each spot column S_i **do**
    $m_i \leftarrow \mathbb{E}[v \mid S_i]$ by quadrature down the column ▷ *eq.* (6)
    $L(t_n, S_i) \leftarrow \sigma_{\text{loc}}(t_n, S_i) / \sqrt{\max(m_i, \epsilon)}$ ▷ *identity, eq.* (3)
  end for
  assemble $A = A_0 + A_1 + A_2$ with $L(t_n, \cdot)$ frozen into its coefficients
  $p(t_{n+1}, \cdot) \leftarrow \textsc{CraigSneydStep}\big(p(t_n, \cdot), A, \Delta t\big)$ ▷ *ADI, App. A*
end for

Algorithm 1: Forward-PDE (Fokker–Planck) calibration of the LSV leverage

[3] For the CIR variance process $\mathrm{d}v_t = \kappa(\theta - v_t)\,\mathrm{d}t + \xi\sqrt{v_t}\,\mathrm{d}W^v_t$ the *Feller condition* is $2\kappa\theta \geq \xi^2$. When it holds the mean-reversion is strong enough relative to the vol-of-vol to repel the process from the origin, and v_t stays strictly positive; when it is violated ($2\kappa\theta < \xi^2$, common in equity calibrations with a large ξ) the origin becomes attainable and probability mass accumulates at $v = 0$. There the variance diffusion $\xi^2 v$ vanishes, so the $v = 0$ boundary must be discretised carefully to keep the scheme stable and mass-conserving.

8 The particle method

Instead of a grid, represent the joint distribution by a large cloud of simulated paths, each particle carrying its own (S, v). The cloud *is* the density—dense where probability is high, sparse where it is low. To estimate $\mathbb{E}[v \mid S = S^*]$ one does not integrate; one looks at the particles whose spot sits near S^* and averages their variances, weighting each by its closeness through a kernel K_δ (a Nadaraya–Watson regression; Appendix C explains how a kernel works),

$$\widehat{\mathbb{E}}[v_t \mid S_t = S^*] = \frac{\sum_i v_t^i \, K_\delta\big(S^* - S_t^i\big)}{\sum_i K_\delta\big(S^* - S_t^i\big)}, \tag{7}$$

illustrated in Figure 13. That weighted average sets the leverage at S^*; one then advances every particle a step with its local leverage frozen in (variance stepped by the QE or full-truncation scheme, never naive Euler—see Appendix B), and repeats. The particles interact *only* through this empirical conditional expectation—that is the McKean–Vlasov coupling made concrete, the cloud talking to itself once per step. Crucially, L is only ever evaluated where particles currently sit, so the leverage builds itself adaptively exactly where the simulation has mass: calibration and simulation are the same forward pass.

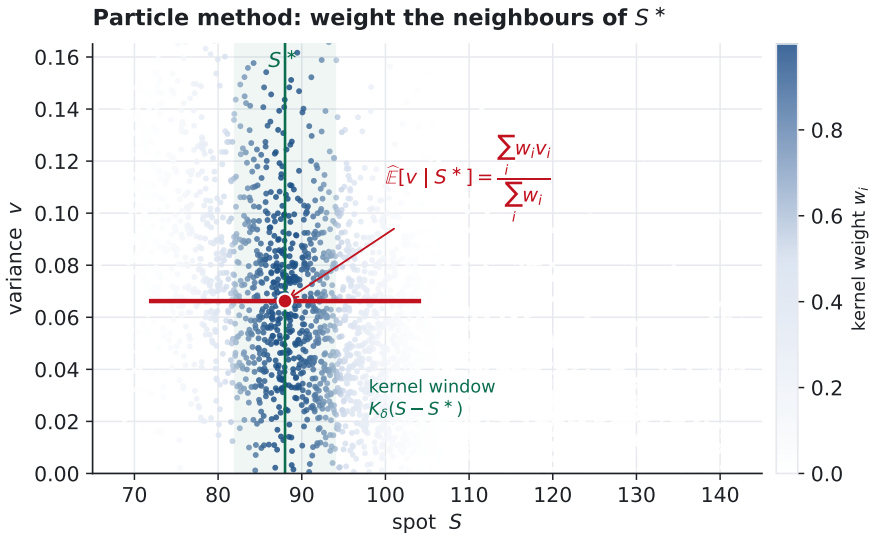

Figure 13: The particle method's defining step. To estimate $\mathbb{E}[v \mid S^*]$, particles are weighted by a kernel centred at S^* (shading = weight); the weighted average of their variances (red) is the estimate. No grid and no integral—just a local regression over the cloud.

This sidesteps the 2D PDE entirely, scales gracefully to extra factors (just give each particle more state), and is fast—which is why it is the desk favourite (Guyon & Henry-Labordère [7, 8]). The price is Monte-Carlo noise, and the bandwidth δ is a bias–variance dial (Figure 14): too small and the conditional expectation is built from a handful of neighbours and is pure noise; too large and it over-smooths, biasing the estimate toward the global mean and washing out the very spot-dependence one is trying to capture. Standard practice is a maturity- and density-dependent bandwidth scaling like $\delta_t \propto \sigma\sqrt{t}\,N^{-1/5}$ (the kernel-regression rate), sometimes in a k-nearest-neighbour form so the window adapts to local density. The wings hurt here too: sparse particles make the denominator both noisy and liable to be small, which blows the leverage up, so one floors the denominator and/or caps L. And because the estimate is biased at finite N, the calibration carries a systematic error—calibrate, then reprice the vanillas through the Monte Carlo and check the residual.

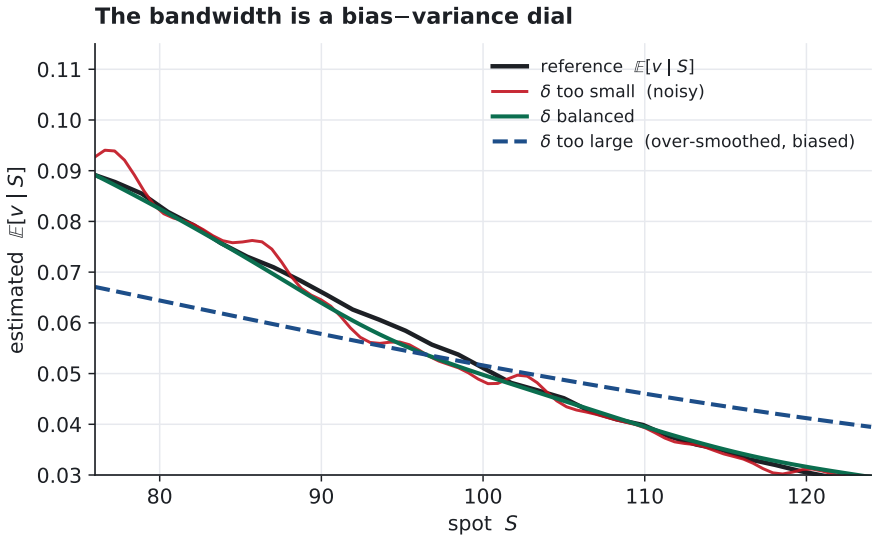

Figure 14: The bandwidth bias–variance trade-off, estimating $\mathbb{E}[v \mid S]$ from a finite particle set. Too small a δ is noisy; too large flattens the curve toward the global mean (bias); a balanced δ tracks the truth.

Algorithm 2 is the same bootstrap carried by the cloud: estimate each particle's conditional mean by kernel regression, set its local leverage, then push every particle one Euler–Maruyama step. Calibration and simulation are literally the same loop.

Input: Dupire surface $\sigma_{\text{loc}}(t, S)$; variance parameters $\kappa, \theta, \xi, \rho$; N particles; time levels $t_0 < \cdots < t_M$ (step Δt); kernel K_δ with bandwidth schedule δ_n; floor ϵ

Output: leverage $L(t_n, S)$ sampled along the cloud

initialise particles $(S^i, v^i) \leftarrow (S_0, v_0)$ for $i = 1, \ldots, N$
for $n = 0, 1, \ldots, M-1$ **do**
  for each particle $i = 1, \ldots, N$ **do**
    $\widehat{m}^{\,i} \leftarrow \widehat{\mathbb{E}}[v \mid S^i]$ by kernel regression over the cloud ▷ *eq.* (7)
    $L^i \leftarrow \sigma_{\text{loc}}(t_n, S^i) / \sqrt{\max(\widehat{m}^{\,i}, \epsilon)}$ ▷ *identity, eq.* (3)
  end for
  draw $(Z_S^i, Z_v^i)_{i=1}^N$, standard normals with corr $= \rho$
  for each particle $i = 1, \ldots, N$ **do**
    $S^i \leftarrow S^i + (r-q)S^i \Delta t + L^i \sqrt{v^i}\, S^i \sqrt{\Delta t}\, Z_S^i$ ▷ *eq.* (4)
    $v^i \leftarrow v^i + \kappa(\theta - v^i)\Delta t + \xi \sqrt{v^i} \sqrt{\Delta t}\, Z_v^i$, then truncate/QE ▷ *App. B*
  end for
end for

Algorithm 2: Particle (Monte-Carlo) calibration of the LSV leverage

	Forward PDE	Particle method
Joint distribution as	density on a grid	cloud of paths
$\mathbb{E}[v \mid S]$ via	quadrature down columns	kernel regression
Character	deterministic, noise-free	Monte-Carlo, noisy
Calibration	separate, reusable grid	fused with the pricing pass
Extra factors	adds a grid dimension	adds state per particle
Main weakness	2D cost, $v=0$ boundary	bandwidth, MC bias

Table 2: The two solvers at a glance.

Calibration versus pricing. A final practical contrast: how each solver relates calibration to pricing. The forward-PDE method is a self-contained, deterministic calibration—one forward sweep produces the whole leverage surface $L(t, S)$ tabulated on the grid, a reusable object that one then hands to a *separate* pricer (typically a Monte Carlo for the exotic in hand). The particle method instead *fuses* the two: calibration and path propagation are the same forward pass, with L built on the fly wherever the particles sit, so the product can be priced on the very paths that calibrated it. To reuse a particle-calibrated surface for other products one simply records $L(t, \cdot)$ slice by slice as the pass proceeds, recovering a gridded surface much like the PDE's.

A The Craig–Sneyd ADI scheme

Section 7 marched the Fokker–Planck equation (5) one step with an *alternating-direction implicit* (ADI) scheme. This appendix sets out one concrete member of that family—the Craig–Sneyd scheme [9]—in enough detail to implement. The obstacle it is built to dodge is the mixed ∂_{Sv} term: a fully implicit step would couple every grid node to every other through both directions at once, a large two-dimensional linear solve. ADI keeps only *one* direction implicit at a time—a cheap tridiagonal solve—and carries the mixed term explicitly.

Finite differences, explicit and implicit. Finite differencing replaces derivatives on a grid (spacing h) by differences of neighbouring values—for instance $\partial_{xx}u \approx (u_{i+1} - 2u_i + u_{i-1})/h^2$. Doing this to the spatial derivatives of a diffusion equation $\partial_t u = \partial_{xx} u$ turns it into a system of ODEs in time, $\mathrm{d}U/\mathrm{d}t = A\,U$, with A tridiagonal (each node coupled only to its two neighbours). What remains is how to step time. An *explicit* step (forward Euler) reads the new values straight off the old,

$$U^{n+1} = U^n + \Delta t\, A\, U^n, \tag{8}$$

just one matrix–vector product—but it is only *conditionally stable*: for diffusion the step is capped at $\Delta t \lesssim h^2/2$, so halving the grid spacing quarters the admissible step and a fine grid forces a huge number of tiny steps. An *implicit* step (backward Euler) instead evaluates the operator at the new level,

$$(I - \Delta t\, A)\, U^{n+1} = U^n, \tag{9}$$

which costs a linear solve per step but is *unconditionally stable*—the step is now limited by accuracy, not stability (Figure 15). In one dimension that system is tridiagonal, an $O(N)$ Thomas sweep, so stability comes almost for free; the popular Crank–Nicolson scheme is the half-and-half average ($\theta = \frac{1}{2}$), second-order accurate and stable. A finance-oriented account of all this is in Austing's *Smile Pricing Explained* [1].

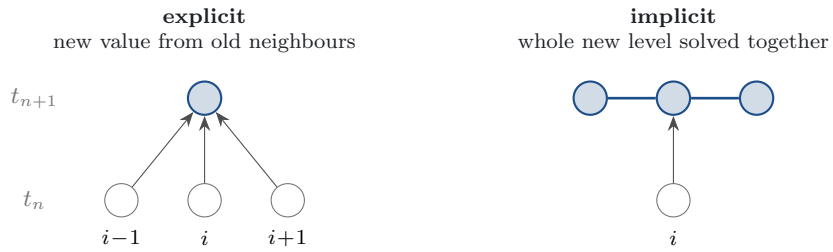

Figure 15: Explicit versus implicit time stepping (a one-dimensional diffusion). Filled blue nodes are unknowns. *Explicit:* the new value at i is an immediate formula in the old neighbours—cheap, but stable only for tiny Δt. *Implicit:* the new level is coupled (blue link) and solved as one linear system—more work per step, but stable at any Δt.

	explicit (forward Euler)	implicit (backward Euler)
Work per step	one matrix–vector product	one linear solve
Stability	conditional, $\Delta t \lesssim h^2/2$	unconditional
Step set by	stability (tiny steps)	accuracy

What breaks in two dimensions is the implicit solve: the mixed ∂_{Sv} term couples the two grid directions, so $I - \Delta t\, A$ is no longer tridiagonal but a wide-bandwidth two-dimensional system, expensive to factor. ADI is the device that keeps implicit stability while restoring the cheap tridiagonal solves.

Method of lines and operator splitting. Apply the same recipe to the Fokker–Planck equation (5): discretise the (S, v) plane (central differences in space, time left continuous) so the grid densities form a vector $U(t)$ obeying $\mathrm{d}U/\mathrm{d}t = A\,U$. In two dimensions the spatial operator splits by direction,

$$A = A_0 + A_1 + A_2, \tag{10}$$

into:

- A_1 collects the S-derivatives ($\partial_S, \partial_{SS}$: the drift and the $\frac{1}{2}L^2 v S^2$ diffusion);
- A_2 collects the v-derivatives ($\partial_v, \partial_{vv}$: the mean-reversion drift and the $\frac{1}{2}\xi^2 v$ diffusion);
- A_0 is the *mixed* term ∂_{Sv} (the $\rho\xi$ coupling).

The coefficients of A are frozen at the values implied by the leverage $L(t, \cdot)$ computed at the start of the step (Section 7), so A is constant across the step. The split is engineered around one fact: with the nodes ordered along grid lines, A_1 couples a node only to its neighbours *in* S, so A_1 is block-diagonal—one *tridiagonal* block per line of constant v—and likewise A_2 is tridiagonal along v. Only the mixed operator A_0 couples the two directions, and it is never inverted.

The scheme. Fix $\theta = \frac{1}{2}$. One step from U^n (time t_n) to U^{n+1} ($t_n + \Delta t$) runs in two passes. A *predictor*—one explicit full step, then one implicit correction per direction:

$$\begin{aligned} Y_0 &= U^n + \Delta t\, A\, U^n, \\ (I - \theta\Delta t\, A_1)\, Y_1 &= Y_0 - \theta\Delta t\, A_1\, U^n, \\ (I - \theta\Delta t\, A_2)\, Y_2 &= Y_1 - \theta\Delta t\, A_2\, U^n; \end{aligned} \tag{11}$$

then a *corrector*, which re-introduces the mixed term explicitly (the Craig–Sneyd step proper) and repeats the directional solves:

$$\begin{aligned} \tilde{Y}_0 &= Y_0 + \tfrac{1}{2}\Delta t\, A_0\, (Y_2 - U^n), \\ (I - \theta\Delta t\, A_1)\, \hat{Y}_1 &= \tilde{Y}_0 - \theta\Delta t\, A_1\, U^n, \\ (I - \theta\Delta t\, A_2)\, \hat{Y}_2 &= \hat{Y}_1 - \theta\Delta t\, A_2\, U^n, \\ U^{n+1} &= \hat{Y}_2. \end{aligned} \tag{12}$$

Read it thus. Y_0 is a cheap, fully explicit guess. Each implicit solve $(I - \theta\Delta t\, A_j)\,(\cdot) = \ldots$ then treats the stiff diffusion in *one* direction implicitly—which is what buys stability—while its right-hand side holds everything else at the old level U^n. Because $I - \theta\Delta t\, A_1$ is tridiagonal along S, that solve is one Thomas-algorithm sweep per v-line; $I - \theta\Delta t\, A_2$ is one sweep per S-line. The mixed operator A_0 enters only through matrix–vector products (in Y_0 and in the correction $\tilde{Y}_0$), so it is never factorised.

Cost, accuracy, stability. A step costs a fixed number of tridiagonal sweeps—two in each direction—plus a few sparse matrix–vector products: $O(N_S N_v)$ work for an $N_S \times N_v$ grid, linear in the number of nodes. Inverting the full two-dimensional operator directly costs far more (a dense factorisation scales as the cube of the node count, and even sparse 2D factorisations suffer heavy fill-in). The explicit predictor on its own (the *Douglas* scheme) is only first-order accurate once a mixed derivative is present; the Craig–Sneyd correction restores *second-order* accuracy in Δt at $\theta = \frac{1}{2}$. And unlike a fully explicit march—whose stable step shrinks like the *square* of the grid spacing—the scheme stays stable at the large steps one actually wants, which is the whole point of going implicit.

Extra factors (stochastic rates, a second asset) split the same way: one more direction $A_3, \dots$ and one more family of tridiagonal sweeps per step. The mixed terms then multiply, and variants such as the Modified Craig–Sneyd or Hundsdorfer–Verwer schemes handle several of them more robustly [11, 10]—but the structure is the one above.

B Euler–Maruyama time stepping

The particle method (Section 8) and the discretised step of Section 7 advance the SDEs by *Euler–Maruyama*, the stochastic cousin of the forward-Euler step (8). For a generic Itô process $\mathrm{d}X_t = a(X_t, t)\,\mathrm{d}t + b(X_t, t)\,\mathrm{d}W_t$ it reads, over a step Δt,

$$X_{t+\Delta t} = X_t + \underbrace{a(X_t, t)\,\Delta t}_{\text{drift}} + \underbrace{b(X_t, t)\,\sqrt{\Delta t}\,Z}_{\text{diffusion}}, \qquad Z \sim \mathcal{N}(0, 1). \tag{13}$$

The one wrinkle that separates it from an ordinary ODE step is the $\sqrt{\Delta t}$: a Brownian increment over Δt has variance Δt (standard deviation $\sqrt{\Delta t}$), not Δt. So over a small step the random "kick" $b\sqrt{\Delta t}\,Z$ dominates the deterministic drift $a\,\Delta t$, and halving Δt shrinks the noise only by $\sqrt{2}$ (Figure 16). The LSV update (4) is exactly this scheme applied to the pair (S_t, v_t), with the two normals correlated by ρ.

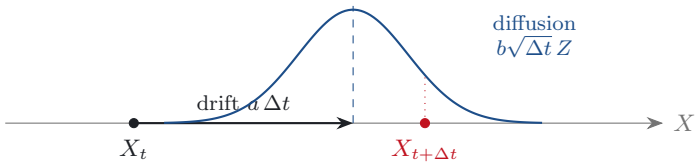

Figure 16: One Euler–Maruyama step. The deterministic drift $a\,\Delta t$ shifts the point; the diffusion then adds a Gaussian kick of size $b\sqrt{\Delta t}$ (note the $\sqrt{\Delta t}$), so the next value is a draw from the bump. Here $X_{t+\Delta t}$ is one such draw.

Accuracy. Euler–Maruyama has *strong* order $\frac{1}{2}$ (the pathwise error, matching individual trajectories, shrinks like $\sqrt{\Delta t}$) but *weak* order 1 (errors in expectations—hence option prices—shrink like Δt). For pricing it is the weak order that matters, so the scheme is first-order in practice; the Milstein scheme adds a $\frac{1}{2}\,b\,b'\,((\Delta W)^2 - \Delta t)$ correction to reach strong order 1 [12].

The variance pitfall. Applied naively to the CIR variance $\mathrm{d}v_t = \kappa(\theta - v_t)\,\mathrm{d}t + \xi\sqrt{v_t}\,\mathrm{d}W_t^v$, a Gaussian kick can push $v_{t+\Delta t}$ below zero, where $\sqrt{v}$ is undefined—the more so when the Feller condition is violated. Two standard repairs keep the scheme alive: *full truncation*, which uses $\sqrt{v^+}$ (i.e. $\sqrt{\max(v, 0)}$) in the coefficients [14], and Andersen's moment-matched *quadratic-exponential (QE)* scheme, more accurate near the origin [13]. This is what Section 8 means by "stepped by the QE or full-truncation scheme, never naive Euler". Austing [1] discusses the practical choices.

C How the kernel works

Section 8 estimates the conditional mean $\mathbb{E}[v_t \mid S_t = S^*]$ from the particle cloud with a *kernel-weighted average* (7). This appendix unpacks what the kernel is doing.

A *kernel* K_δ is just a weighting function—a smooth bump centred at zero that is largest there and decays as its argument moves away. The canonical choice is the Gaussian,

$$K_\delta(u) = \exp\Big(-\tfrac{u^2}{2\delta^2}\Big), \tag{14}$$

whose width is set by the *bandwidth* δ. Evaluated at $u = S^* - S_t^i$, it scores how close particle i's spot S_t^i sits to the query point S^*: a particle right at S^* gets weight 1, one a few bandwidths away gets a weight near 0.

The conditional mean we want is the average variance *among the paths that sit at* S^*. The cloud almost never has particles exactly at S^*, so instead of an impossible exact match we take a *soft* neighbourhood: weight every particle by its closeness $K_\delta(S^* - S_t^i)$ and form the weighted average of the variances. That is precisely (7); the denominator $\sum_i K_\delta(S^* - S_t^i)$ simply normalises the weights so they sum to one. Equivalently, it fits a constant to the variances of the nearby particles, with nearness measured by the kernel—the simplest form of a *local regression*.

The bandwidth δ is the one real choice, and it is the bias–variance dial of Figure 14. A small δ keeps only the few closest particles: the estimate follows the true $\mathbb{E}[v \mid S]$ faithfully (low bias) but, built from a handful of points, is noisy (high variance). A large δ averages over a wide window: smooth and stable (low variance) but blurred toward the global mean, flattening out the very spot-dependence we are trying to measure (high bias). A hard window—count every particle within $\pm\delta$ equally and ignore the rest—is the special case of a rectangular kernel; the smooth bump of (14) simply avoids the jitter of particles popping in and out of a sharp cutoff as S^* moves.

References

[1] P. Austing. *Smile Pricing Explained*. Palgrave Macmillan, 2014.

[2] B. Dupire. Pricing with a smile. *Risk*, 7(1):18–20, 1994.

[3] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, 1993.

[4] P. S. Hagan, D. Kumar, A. S. Lesniewski, D. E. Woodward. Managing smile risk. *Wilmott Magazine*, pages 84–108, September 2002.

[5] I. Gyöngy. Mimicking the one-dimensional marginal distributions of processes having an Itô differential. *Probability Theory and Related Fields*, 71(4):501–516, 1986.

[6] A. Lipton. The vol smile problem. *Risk*, 15(2):61–65, 2002.

[7] J. Guyon, P. Henry-Labordère. Being particular about calibration. *Risk*, 25(1):88–93, 2012.

[8] J. Guyon, P. Henry-Labordère. *Nonlinear Option Pricing*. Chapman & Hall/CRC Financial Mathematics Series, 2014.

[9] I. J. D. Craig, A. D. Sneyd. An alternating-direction implicit scheme for parabolic equations with mixed derivatives. *Computers & Mathematics with Applications*, 16(4):341–350, 1988.

[10] K. J. in 't Hout, S. Foulon. ADI finite difference schemes for option pricing in the Heston model with correlation. *International Journal of Numerical Analysis and Modeling*, 7(2):303–320, 2010.

[11] W. Hundsdorfer, J. G. Verwer. *Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations*. Springer, 2003.

[12] P. E. Kloeden, E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.

[13] L. B. G. Andersen. Simple and efficient simulation of the Heston stochastic volatility model. *Journal of Computational Finance*, 11(3):1–42, 2008.

[14] R. Lord, R. Koekkoek, D. van Dijk. A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2):177–194, 2010.