QUANTITATIVE RESEARCH · TECHNICAL NOTE

From Splines to Repo Curves

A Progressive Journey Through Piecewise Polynomial Theory
and its Application to Repurchase Rate Term Structures

v1.0 · April 2026

Contents

Part I

Historical and Intuitive Foundations

1 The Draughtsman's Spline: From Wood to Mathematics

Before splines were equations, they were physical objects. Understanding their origin illuminates why the mathematical theory takes the form it does.

1.1 Shipbuilding and the flexible ruler

In 18th and 19th century shipyards, naval architects faced a practical problem: how to draw the smooth curves of a ship's hull at full scale. A hull is designed from a small number of characteristic points (the "stations"), but the full drawing requires a continuous, visually pleasing curve passing through them.

The solution was the **spline**, a thin flexible strip of wood or metal. Draughtsmen placed heavy lead weights called **ducks** at the known points, and the flexible strip naturally assumed the curve of minimum bending energy consistent with passing through those anchors. The resulting shape was smooth, harmonious, and — crucially — reproducible.

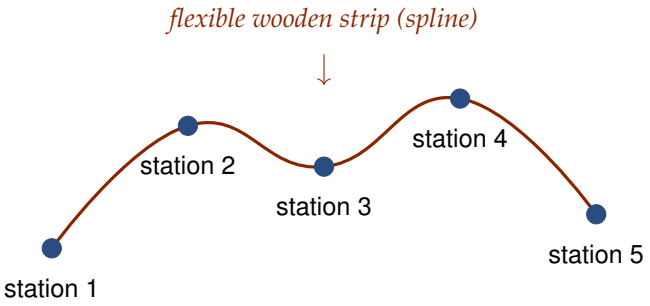

Figure 1.1 — The draughtsman's spline: a flexible strip anchored at fixed points, assuming a curve of minimum bending energy.

1.2 The physical principle

The physics of the wooden strip is governed by Euler–Bernoulli beam theory. A thin elastic beam under small deflection minimizes its elastic strain energy, which is proportional to the square of its curvature integrated along its length:

$$E_{\text{bend}} \propto \int \kappa(s)^2 \, ds \approx \int \left[y''(x)\right]^2 \, dx$$

The second approximation assumes small deflections so arc length $s \approx x$ and curvature $\kappa \approx y''$. The physical strip, subject to the constraints $y(x_i) = y_i$ at each duck, finds the function y that minimizes $\int y''(x)^2 \, dx$ — and this function, remarkably, has a clean mathematical characterization: it is a piecewise cubic polynomial.

Historical note — Isaac Schoenberg

The mathematical theory of splines was formalized by **Isaac Jacob Schoenberg** in a foundational 1946 paper, "Contributions to the problem of approximation of equidistant data by analytic functions." Schoenberg, born in Romania in 1903 and working at the University of Pennsylvania during the war years on ballistics tables, recognized that the same piecewise polynomial constructions used by draughtsmen could serve as a rigorous tool for statistical approximation and numerical analysis.

The term *spline* was borrowed directly from the shipbuilder's vocabulary. Schoenberg's genius was to see that the physical minimization principle had an elegant algebraic counterpart via what are now called **B-splines** (basis splines), introduced explicitly in his 1967 paper with Curry.

1.3 Why this matters for finance

The draughtsman's intuition carries over with remarkable fidelity to financial curve construction. We have observations at a finite set of tenors (the "stations") and we seek a smooth function passing near — not necessarily through — those points. The minimum bending energy principle becomes a formal optimization criterion, and the flexible strip becomes a vector in a functional space. The entire edifice we will build rests on this mechanical analogy.

2 What is a Spline? A First Definition

Stripped to its essence, a spline is a piecewise polynomial with prescribed smoothness at the junctions. Everything else — B-splines, natural splines, smoothing splines — is commentary on this core idea.

2.1 The formal definition

Definition 2.1 — Spline function

Let $[a, b]$ be a closed interval and let $\boldsymbol{\xi} = (\xi_0, \xi_1, \ldots, \xi_K, \xi_{K+1})$ be a strictly increasing sequence with $\xi_0 = a$ and $\xi_{K+1} = b$. The interior points $\xi_1, \ldots, \xi_K$ are called **knots**.
A function $s : [a, b] \to \mathbb{R}$ is a **spline of degree** d with knots $\boldsymbol{\xi}$ if:
1. On each subinterval $[\xi_j, \xi_{j+1}]$, s is a polynomial of degree $\leq d$;
2. $s \in C^{d-1}[a, b]$, i.e. s and its first $d - 1$ derivatives are continuous on $[a, b]$.
The set of all such splines is denoted $\mathcal{S}_d(\boldsymbol{\xi})$.

Let us unpack each element of this definition carefully, because every word matters.

2.2 The three ingredients

2.2.1 The knots ξ_j

Knots are the points where the polynomial pieces join. They partition the interval $[a, b]$ into subintervals on which the spline has a simple polynomial form. The placement and number of knots determine the spline's flexibility: more knots, more local detail; fewer knots, smoother global behavior.

2.2.2 The degree d

The degree controls how "wiggly" each polynomial piece can be. Standard choices:

Degree d	Name	Polynomial form	Continuity
0	Constant	$s(x) = c_j$	none (step)
1	Linear	$s(x) = a_j + b_j x$	C^0
2	Quadratic	$s(x) = a_j + b_j x + c_j x^2$	C^1
3	Cubic	$s(x) = a_j + b_j x + c_j x^2 + d_j x^3$	C^2

The cubic spline ($d = 3$) is by far the most common in practice. This is not arbitrary. It is the **minimum degree** at which the second derivative exists and is continuous — and the second

derivative is exactly what the minimum bending energy principle penalizes. Cubic splines are the natural mathematical expression of the draughtsman's flexible strip.

2.2.3 The continuity requirement C^{d-1}

This is what makes a spline a spline, rather than a collection of disconnected polynomial segments. At each knot ξ_j, we require that the function value agrees from left and right, and that the first $d-1$ derivatives also agree. For a cubic spline:

$$\lim_{x \to \xi_j^-} s^{(k)}(x) = \lim_{x \to \xi_j^+} s^{(k)}(x), \qquad k = 0, 1, 2.$$

The third derivative $s^{(3)}$ is generally *not* continuous at knots — this is where the "kink" in the third derivative lives, which is what distinguishes one piecewise polynomial from another.

2.3 A visual reality check

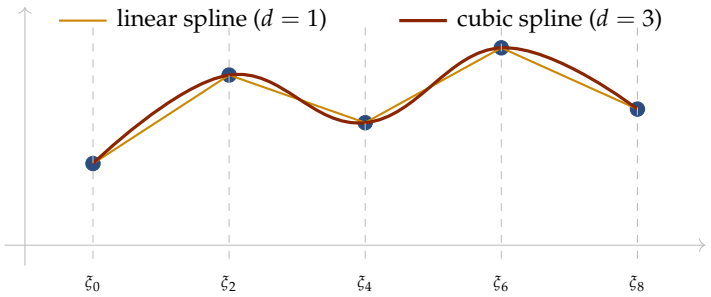

Figure 2.1 — The same knots, two different degrees. The linear spline is continuous but visibly broken; the cubic spline is smooth to the eye.

3 Why Piecewise Polynomials?

Understanding why splines work requires understanding what goes wrong with the obvious alternatives. The answer is one of the most beautiful results in classical approximation theory: Runge's phenomenon.

3.1 The naive approach and its failure

Given n data points $(x_1, y_1), \ldots, (x_n, y_n)$, the most elementary interpolation method is to fit a single polynomial of degree $n-1$:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}.$$

This *always works*: the system of equations $p(x_i) = y_i$ is exactly determined, and the solution (the Lagrange interpolating polynomial) exists and is unique. So why not simply use this?

Runge's phenomenon (1901)

Carl Runge showed that for the innocuous-looking function $f(x) = 1/(1+25x^2)$ on $[-1, 1]$, polynomial interpolation at equidistant nodes **diverges** as the number of nodes increases:

$$\sup_{x \in [-1,1]} |p_n(x) - f(x)| \to \infty \text{ as } n \to \infty.$$

The interpolating polynomial develops massive oscillations near the endpoints, growing exponentially with n. Using *more data* makes the fit *worse*.

3.2 The root cause: high-degree polynomials are rigid

Runge's phenomenon is not a pathology specific to one function. It is a structural defect of the space of high-degree polynomials. Two geometric facts explain it:

3.2.1 Non-locality

A polynomial of degree $n-1$ has n coefficients, and *every* coefficient is determined by *every* data point. Move one data point slightly and the entire polynomial reshapes globally. This is the opposite of what we want for approximation: local perturbations should have local effects.

3.2.2 Oscillation

By the fundamental theorem of algebra, a polynomial of degree $n-1$ has exactly $n-1$ roots. Its derivative has $n-2$ roots, so the polynomial has up to $n-2$ turning points. For large n, this forces extreme wiggling just to satisfy the interpolation constraints.

3.3 The spline resolution

The essential insight

Rather than using one polynomial of high degree over a wide interval, use *many polynomials of low degree*, each on a small subinterval, and glue them together smoothly.

This one idea has profound consequences:

- **Locality is restored.** Moving a data point affects only nearby polynomial pieces, not the curve at distant points.
- **Oscillation is bounded.** A cubic has at most two turning points; its behavior is controllable regardless of how many data points we have.
- **Smoothness is preserved.** The continuity conditions ensure the pieces join seamlessly, without visible breaks.
- **Approximation power is retained.** Splines with enough knots can approximate any continuous function to arbitrary accuracy (Schoenberg's density theorem).

3.4 Approximation theory results

The theoretical guarantees are precise. For a function $f \in C^4[a,b]$ and cubic spline interpolation with maximum knot spacing h:

$$\|f - s\|_\infty \leq C \cdot h^4 \cdot \|f^{(4)}\|_\infty$$

The error decreases as the fourth power of the knot spacing. Halving the spacing reduces the error by a factor of 16. This is fundamentally better than global polynomial interpolation, which can diverge entirely.

This single inequality — proved in de Boor's 1978 monograph — is the mathematical justification for why splines dominate finite-element methods, computer graphics, statistical smoothing, and financial curve construction.

Part II

Mathematical Theory of Splines

4 Knots, Degrees, and Continuity Conditions

Before we can manipulate splines computationally, we need to understand their internal structure — the algebra that governs how knots, degrees, and continuity interact.

4.1 Counting degrees of freedom

How many parameters does a spline have? This dimension count is the foundation of every practical spline algorithm.

On each of the $K+1$ subintervals $[\xi_j, \xi_{j+1}]$, a polynomial of degree d has $d+1$ coefficients. So naively the total count is $(K+1)(d+1)$ parameters.

But continuity imposes constraints. At each of the K interior knots, we require d continuity conditions (matching $s, s', \ldots, s^{(d-1)}$). This gives Kd constraints.

Dimension theorem

$$\dim \mathcal{S}_d(\boldsymbol{\xi}) = (K+1)(d+1) - Kd = K + d + 1.$$

A spline of degree d with K interior knots has $K+d+1$ free parameters.

For a cubic spline ($d=3$) with K interior knots, this gives $K+4$ degrees of freedom. If we have n data points with $K=n$ knots placed at the data, we have $n+4$ parameters and n interpolation constraints — so 4 degrees of freedom remain to be fixed by boundary conditions. This is why cubic splines require boundary conditions to be uniquely determined.

4.2 Types of boundary conditions

Name	Condition	Use case
Natural	$s''(a) = s''(b) = 0$	No information beyond range
Clamped	$s'(a), s'(b)$ specified	Known derivatives
Periodic	$s^{(k)}(a) = s^{(k)}(b)$	Closed curves, seasonal data
Not-a-knot	$s^{(3)}$ continuous at ξ_1, ξ_K	Default in MATLAB, SciPy

For financial curves, **natural boundary conditions** are almost always used at the far end of the curve, encoding the idea that we have no information to extrapolate curvature beyond the last observation.

4.3 Knot multiplicity

So far we have assumed all knots are distinct. An important extension allows a knot ξ_j to have **multiplicity** $m_j \geq 1$ (i.e. appearing m_j times in the knot sequence).

At a knot of multiplicity m, the continuity requirement drops from C^{d-1} to C^{d-m}. A cubic spline ($d=3$) with a triple knot ($m=3$) is only C^0 at that knot — the function is continuous but the first derivative jumps.

This is the key mechanism for introducing **controlled discontinuities**. As we will see, repo curves require precisely this tool: the ability to force a jump in the curve at a specific date (end of year, for example) while maintaining smoothness elsewhere.

Knot multiplicity table

Multiplicity m	Continuity at knot	Interpretation
1	C^{d-1} (full smoothness)	Standard interior knot
2	C^{d-2}	Curvature can jump
3	C^{d-3}	Slope can jump (for $d = 3$)
$d+1$	Discontinuous	Function value can jump

5 The Space of Splines as a Vector Space

The set of splines with fixed knots and degree forms a vector space. This abstraction is what allows us to work with splines computationally — any spline can be represented as a linear combination of basis functions.

5.1 Linearity of the spline space

Let $s_1, s_2 \in \mathcal{S}_d(\boldsymbol{\xi})$ and $\alpha, \beta \in \mathbb{R}$. Then:

- $\alpha s_1 + \beta s_2$ is piecewise polynomial of degree $\leq d$ on each subinterval;
- $\alpha s_1 + \beta s_2 \in C^{d-1}[a, b]$ (derivatives commute with linear combinations).

Therefore $\alpha s_1 + \beta s_2 \in \mathcal{S}_d(\boldsymbol{\xi})$. The set of splines is closed under linear combination — it is a vector space of dimension $K + d + 1$.

5.2 Why this matters

Because $\mathcal{S}_d(\boldsymbol{\xi})$ is a finite-dimensional vector space, any spline can be written as

$$s(x) = \sum_{j=1}^{K+d+1} c_j \, \phi_j(x)$$

where $\{\phi_1, \ldots, \phi_{K+d+1}\}$ is any basis of $\mathcal{S}_d(\boldsymbol{\xi})$. The problem of constructing a spline reduces to finding $K + d + 1$ real numbers — the coefficients c_j. All computation becomes linear algebra.

5.3 Candidate bases

5.3.1 The truncated power basis

The most elementary basis for cubic splines is:

$$1,\ x,\ x^2,\ x^3,\ (x - \xi_1)_+^3,\ (x - \xi_2)_+^3,\ \ldots,\ (x - \xi_K)_+^3$$

where $(u)_+^3 = \max(u, 0)^3$. This basis is conceptually simple but **catastrophically ill-conditioned** for numerical computation. The basis functions have supports that grow monotonically, making the design matrix near-singular when many knots are used.

5.3.2 The need for a better basis

A good basis for splines should have:

1. **Local support.** Each basis function nonzero only on a small portion.
2. **Positivity.** Basis functions nonnegative to avoid cancellation errors.
3. **Partition of unity.** Basis functions should sum to 1.

4. **Numerical stability.** Resulting linear systems well-conditioned.

These four properties uniquely characterize a very special basis, invented by Schoenberg and developed by Curry, Cox, and de Boor: the **B-spline basis**.

6 The B-Spline Basis

B-splines are the computational backbone of every modern spline algorithm. They are defined recursively, have compact support, form a partition of unity, and make the numerical linear algebra tractable.

6.1 The Cox–de Boor recursion

Extend the knot sequence on both sides: introduce auxiliary knots $\xi_{-d}, \dots, \xi_{-1}$ at or below a, and $\xi_{K+2}, \dots, \xi_{K+d+1}$ at or above b.

Definition 6.1 — B-splines (Cox–de Boor, 1972)

Degree 0:

$$B_{j,0}(x) = \begin{cases} 1 & \text{if } \xi_j \le x < \xi_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

Degree $d \ge 1$ (recursively):

$$B_{j,d}(x) = \frac{x - \xi_j}{\xi_{j+d} - \xi_j} B_{j,d-1}(x) + \frac{\xi_{j+d+1} - x}{\xi_{j+d+1} - \xi_{j+1}} B_{j+1,d-1}(x).$$

This recursion, though it looks intimidating, admits an intuitive reading: each $B_{j,d}$ is a weighted average of two B-splines of lower degree, with weights that interpolate linearly across the support.

6.2 Properties of B-splines

Theorem 6.1 — Fundamental properties

For the degree-d B-spline $B_{j,d}$:

1. **Local support:** $B_{j,d}(x) = 0$ for $x \notin [\xi_j, \xi_{j+d+1}]$. Each B-spline is nonzero on exactly $d+1$ consecutive subintervals.
2. **Positivity:** $B_{j,d}(x) \ge 0$ for all x.
3. **Partition of unity:** $\sum_j B_{j,d}(x) = 1$ for all $x \in [a, b]$.
4. **Smoothness:** $B_{j,d} \in C^{d-1}[a, b]$ at simple knots.
5. **Basis property:** The set $\{B_{j,d}\}_{j=-d}^{K}$ forms a basis of $\mathcal{S}_d(\boldsymbol{\xi})$.

6.3 Visual intuition

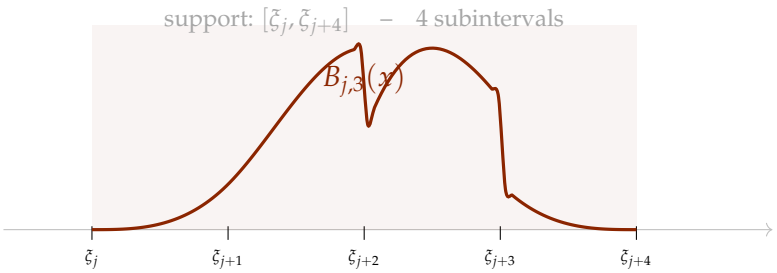

Figure 6.1 — A cubic B-spline. It is a smooth bump, nonzero on exactly four consecutive subintervals, achieving its maximum near the middle.

6.4 The B-spline representation in practice

Any cubic spline $s \in \mathcal{S}_3(\boldsymbol{\xi})$ can be written uniquely as:

$$s(x) = \sum_{j=-3}^{K} c_j \, B_{j,3}(x).$$

The coefficients $\{c_j\}$ are called the **B-spline coefficients** or **control points**. In computer graphics, the polygon connecting the points (ξ_j^*, c_j), where ξ_j^* is the average of the d knots in the support, is called the control polygon. The spline curve visually "follows" this polygon, pulled toward it by the bell-shaped B-splines.

6.5 Why this basis is the right one

The local support property is the single most important feature. When we evaluate $s(x)$ for a specific x, only at most $d+1$ coefficients contribute. This means:

- Evaluating $s(x)$ at n points costs $O(n \cdot d)$, not $O(n \cdot K)$.
- The design matrix $B_{ij} = B_{j,d}(x_i)$ is **banded**, with bandwidth $d+1$.
- Linear systems involving this matrix solve in $O(K)$ time rather than $O(K^3)$.
- Changing one coefficient c_j changes $s(x)$ only on $[\xi_j, \xi_{j+d+1}]$.

These properties make B-splines the computational foundation of every serious spline library: SciPy, R's `splines` package, MATLAB's Curve Fitting Toolbox, and every CAD system.

7 Natural Cubic Splines and Reinsch's Theorem

The connection between the draughtsman's wooden strip and modern smoothing comes from a remarkable theorem: among all smooth functions interpolating given data, the natural cubic spline minimizes the bending energy.

7.1 The natural cubic spline

Definition 7.1

A **natural cubic spline** on $[a, b]$ is a cubic spline s satisfying the natural boundary conditions:

$$s''(a) = s''(b) = 0.$$

Equivalently, s is linear on $(-\infty, a]$ and on $[b, \infty)$ when extended beyond the interval.

The term "natural" is physically apt: a wooden spline with no applied moment at its ends has zero curvature there (moment is proportional to curvature for small deflections), and extends as a straight line beyond the last duck.

7.2 The variational characterization

Consider the problem: among all functions f with $f(x_i) = y_i$ for $i = 1, \dots, n$, find the one that minimizes

$$J(f) = \int_a^b \left[f''(x)\right]^2 \, dx.$$

This is the total bending energy of a wooden strip forced through the given points.

Theorem 7.1 — Holladay (1957)

Let $x_1 < x_2 < \cdots < x_n$ be distinct points in $[a, b]$ and let $y_1, \dots, y_n \in \mathbb{R}$. The unique minimizer of $J(f)$ over $f \in H^2[a, b]$ subject to $f(x_i) = y_i$ is the natural cubic spline interpolant with knots $x_1, \dots, x_n$.

Proof sketch. Let s be the natural cubic spline interpolant and let f be any other interpolant. Write $f = s + h$ where h vanishes at each x_i. Then

$$J(f) = \int_a^b (s'' + h'')^2 \, dx = J(s) + 2\int_a^b s''h'' \, dx + J(h).$$

The cross term can be integrated by parts twice:

$$\int_a^b s''h'' \, dx = \left[s''h'\right]_a^b - \int_a^b s'''h' \, dx.$$

The boundary term vanishes by the natural conditions $s''(a) = s''(b) = 0$. Integration by parts again, noting that s''' is piecewise constant with jumps at the knots and $h(x_i) = 0$, shows the remaining integral vanishes. Thus $J(f) = J(s) + J(h) \geq J(s)$, with equality iff $h \equiv 0$. $\square$

7.3 Reinsch's extension to smoothing

Holladay's theorem handles exact interpolation. But in practice, data is noisy and interpolation is undesirable — we want a *smooth approximation*, not a curve forced through every observation. Christian Reinsch, in a 1967 paper, extended the result in a way that would transform statistics.

Theorem 7.2 — Reinsch (1967)

Consider the problem: minimize

$$J_\lambda(f) = \sum_{i=1}^{n} w_i \left(y_i - f(x_i)\right)^2 + \lambda \int_a^b \left[f''(x)\right]^2 \, dx$$

over $f \in H^2[a, b]$, where $\lambda > 0$ and $w_i > 0$. The unique minimizer is a **natural cubic spline with knots at the data points** $x_1, \dots, x_n$.

This theorem is extraordinary. It says that the infinite-dimensional optimization problem — minimize over all twice-differentiable functions — has a *finite-dimensional* solution whose structure we know in advance.

7.4 The smoothing parameter

The parameter λ controls the balance between two competing goals:

- **Data fidelity** (first term): curve should pass close to observations.
- **Smoothness** (second term): curve should have small bending energy.

Limiting cases

λ	**Behavior**
$\lambda \to 0$	Interpolation: curve passes through every data point
$\lambda \to \infty$	Linear regression: $f'' \equiv 0$, so f is a straight line
$\lambda = \lambda^*$	Best bias-variance trade-off (GCV-selected)

We have now arrived, through careful progression, at the core of modern smoothing theory. Reinsch's theorem is the theoretical keystone: it tells us what the right function class is. The remaining chapters concern how to compute the answer efficiently and how to adapt it to specific structural features — particularly the discontinuities and local variations of repo curves.

Part III

From Interpolation to Smoothing

8 The Problem with Interpolation

Financial data is noisy. Treating every observation as exact leads to overfitting, instability, and curves that do not extrapolate or interpolate usefully. This chapter formalizes why we must smooth rather than interpolate.

8.1 The statistical model

Assume observations follow the model:

$$y_i = f(x_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$$

where f is the unknown smooth function of interest and ε_i is observation noise. An interpolant $\hat{f}_{\text{int}}$ satisfying $\hat{f}_{\text{int}}(x_i) = y_i$ necessarily captures both signal and noise. At any given data point:

$$\hat{f}_{\text{int}}(x_i) - f(x_i) = y_i - f(x_i) = \varepsilon_i,$$

with variance σ_i^2. The interpolant is an *unbiased* but extremely *variable* estimator of f.

8.2 The bias-variance decomposition

The mean squared error of any estimator $\hat{f}$ at a point x decomposes as:

$$\mathbb{E}\left[(\hat{f}(x) - f(x))^2\right] = \underbrace{\left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2}_{\text{bias}^2} + \underbrace{\text{Var}(\hat{f}(x))}_{\text{variance}}.$$

Interpolation has zero bias but maximum variance. Fitting a straight line has minimum variance but huge bias (unless f is actually linear). The optimal estimator lies somewhere in between — and the parameter λ in the smoothing spline problem parameterizes exactly this trade-off.

8.3 Overfitting in the spline context

Suppose we fit a cubic spline with K knots to n data points. The spline has $K + 4$ free parameters. The "effective complexity" grows with K:

Knots K	DoF	Typical behavior
0	4	A single cubic polynomial — underfits
$n/4$	$\approx n/4 + 4$	Captures major trends — good balance
n	$n + 4$	Interpolates every point — overfits

Choosing K is painful. Too few and we miss structure; too many and we chase noise.

8.4 The smoothing spline escape

Reinsch's theorem offers an elegant way out. Instead of choosing K (a discrete, hard decision), we place knots at *every* data point and control complexity via the continuous parameter λ.

Effective degrees of freedom

For a smoothing spline with parameter λ, the effective degrees of freedom is

$$\text{edf}(\lambda) = \text{tr}(\mathbf{H}_\lambda)$$

where $\mathbf{H}_\lambda$ is the hat matrix: $\hat{\mathbf{y}} = \mathbf{H}_\lambda \mathbf{y}$. edf varies continuously from 2 (straight line, $\lambda = \infty$) to n (interpolation, $\lambda = 0$), giving a principled way to measure and control complexity.

9 Smoothing Splines: The Variational Principle

We now assemble the complete machinery for smoothing splines — the problem formulation, the solution structure, the hat matrix, and the selection of the smoothing parameter.

9.1 The general problem

Smoothing spline problem

Given data (x_i, y_i, w_i) for $i = 1, \ldots, n$, find f minimizing:

$$J_\lambda(f) = \sum_{i=1}^{n} w_i (y_i - f(x_i))^2 + \lambda \int_a^b (f''(x))^2 \, dx$$

over $f \in H^2[a, b]$, with $\lambda > 0$.

By Reinsch's theorem, the solution is a natural cubic spline with knots at the distinct x_i. Expand in the B-spline basis: $\hat{f}(x) = \sum_j c_j B_j(x)$. The problem becomes finite-dimensional.

9.2 Matrix formulation

Let $\mathbf{B}$ be the $n \times m$ matrix with $B_{ij} = B_j(x_i)$, $\mathbf{W} = \text{diag}(w_1, \ldots, w_n)$. Define the penalty matrix:

$$\mathbf{\Omega}_{jk} = \int_a^b B_j''(x) B_k''(x) \, dx.$$

Then $\int (f'')^2 \, dx = \mathbf{c}^T \mathbf{\Omega} \mathbf{c}$ and the minimization becomes:

$$\min_{\mathbf{c}} \, (\mathbf{y} - \mathbf{B}\mathbf{c})^T \mathbf{W} (\mathbf{y} - \mathbf{B}\mathbf{c}) + \lambda \, \mathbf{c}^T \mathbf{\Omega} \mathbf{c}.$$

Setting the gradient to zero gives the normal equations:

$$(\mathbf{B}^T \mathbf{W} \mathbf{B} + \lambda \mathbf{\Omega}) \mathbf{c} = \mathbf{B}^T \mathbf{W} \mathbf{y}.$$

Closed-form solution

$$\hat{\mathbf{c}}_\lambda = (\mathbf{B}^T \mathbf{W} \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^T \mathbf{W} \mathbf{y}$$
$$\hat{\mathbf{y}}_\lambda = \mathbf{H}_\lambda \mathbf{y}, \quad \mathbf{H}_\lambda = \mathbf{B}(\mathbf{B}^T \mathbf{W} \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^T \mathbf{W}.$$

The matrix $\mathbf{H}_\lambda$ is the **smoother matrix** or **hat matrix** — it maps observations to fitted values linearly.

9.3 Selecting λ: generalized cross-validation

The optimal λ should minimize prediction error. Leave-one-out cross-validation:

$$\text{LOOCV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - [\mathbf{H}_\lambda]_{ii}}\right)^2.$$

A more stable variant, introduced by Craven and Wahba in 1979, is **generalized cross-validation**:

$$\text{GCV}(\lambda) = \frac{n\,\|\mathbf{y} - \hat{\mathbf{y}}_\lambda\|^2}{(n - \text{tr}(\mathbf{H}_\lambda))^2}.$$

GCV replaces individual leverages $[\mathbf{H}_\lambda]_{ii}$ with their average $\text{tr}(\mathbf{H}_\lambda)/n$, yielding a rotation-invariant criterion that is numerically more stable and nearly as good asymptotically.

9.4 Limitations

Classical smoothing splines have two practical drawbacks:

- **Computational cost.** The basis dimension equals the number of distinct data points. Expensive for large n.
- **Global smoothness.** A single λ controls smoothness uniformly across the interval. If the underlying function has regions of rapid change alongside smooth regions, no single λ is optimal everywhere.

Both issues are addressed by the next refinement: **P-splines**.

10 P-Splines: The Eilers-Marx Innovation

In 1996, Paul Eilers and Brian Marx proposed a simplification of smoothing splines that is at once more elegant, more computationally efficient, and more flexible. P-splines have become the workhorse of modern non-parametric regression.

10.1 The key insight

Classical smoothing splines place a knot at every data point and penalize the integrated squared second derivative. Eilers and Marx observed two simplifications:

1. Use a **moderate** number of equally-spaced knots (say 20–40) — fewer than data points — without losing approximation power.
2. The integrated squared second derivative $\int (f'')^2 dx$ can be replaced with a simple **second-difference penalty on the B-spline coefficients**: $\sum_j (\Delta^2 c_j)^2$.

The result is dramatically simpler: a moderate-sized B-spline basis plus a trivial finite-difference penalty, controlled by a single parameter λ.

10.2 Formulation

Definition 10.1 — The P-spline

Let $B_1, \ldots, B_m$ be B-splines of degree d on equally-spaced knots, with m moderate ($m \sim$ 20–50). Let $\mathbf{D}_q$ denote the q-th order difference matrix.
The P-spline fit minimizes:

$$\min_{\mathbf{c}} \; \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{B}\mathbf{c})\|^2 + \lambda\,\|\mathbf{D}_q\mathbf{c}\|^2.$$

> The penalty order q is typically 2 or 3.

The difference matrix $\mathbf{D}_2$ produces second differences:

$$(\mathbf{D}_2\mathbf{c})_j = c_{j+2} - 2c_{j+1} + c_j, \qquad j = 1, \ldots, m-2.$$

Minimizing $\|\mathbf{D}_2\mathbf{c}\|^2$ pushes consecutive triples toward lying on a straight line, encouraging smoothness of the underlying spline. In the limit $\lambda \to \infty$, the coefficients are exactly linear in j and the P-spline collapses to the best-fitting line.

10.3 Why it works so well

Advantages of P-splines over classical smoothing splines

1. **Decoupling of basis and penalty.** Basis size m controls maximum flexibility; λ controls actual smoothness. We can be generous with m without overfitting risk.
2. **Simple penalty.** $\mathbf{D}_q^T\mathbf{D}_q$ is trivially computed — banded, sparse, and constant. No numerical integration needed.
3. **Automatic boundary behavior.** With second-order penalty, fit naturally extends linearly beyond data — mimicking natural boundary condition.
4. **Computational efficiency.** With $m \ll n$ knots, system size is $m \times m$, independent of sample size.
5. **Easy extensions.** Additive models, generalized responses, adaptive penalties, and tensor products trivially accommodated.

10.4 The closed form

The first-order condition yields:

$$(\mathbf{B}^T\mathbf{W}\mathbf{B} + \lambda\mathbf{D}_q^T\mathbf{D}_q)\hat{\mathbf{c}} = \mathbf{B}^T\mathbf{W}\mathbf{y}.$$

The matrix on the left is symmetric, positive-definite, and banded with bandwidth $\max(d+1, q)$. Cholesky factorization in $O(m)$, evaluation at any point in $O(d)$ via B-spline local support.

10.5 The effective degrees of freedom

The hat matrix is $\mathbf{H}_\lambda = \mathbf{B}(\mathbf{B}^T\mathbf{W}\mathbf{B} + \lambda\mathbf{D}_q^T\mathbf{D}_q)^{-1}\mathbf{B}^T\mathbf{W}$ and $\text{edf}(\lambda) = \text{tr}(\mathbf{H}_\lambda)$. For typical financial applications, edf in the range 5–20 gives visually pleasing fits; values above 30 usually indicate overfitting. GCV extends immediately:

$$\text{GCV}(\lambda) = \frac{n\,\|\mathbf{W}^{1/2}(\mathbf{y} - \hat{\mathbf{y}}_\lambda)\|^2}{(n - \text{edf}(\lambda))^2}.$$

11 Adaptive P-Splines for Local Features

Uniform smoothness is often wrong. Adaptive P-splines let the smoothing parameter vary across the domain, so that smooth regions are heavily penalized and rough regions are lightly penalized — the right behavior for repo curves.

11.1 The motivation

Consider a function that is smooth on most of its domain but has a localized region of rapid variation (a "kink", a bump, or a steep gradient). Fitting with a single global λ forces a choice:

- Small λ — captures the local feature but oversmooths elsewhere.
- Large λ — smooth globally but misses the local feature.

The solution is to let λ depend on position: $\lambda(x)$. Where the true function is rough, use small λ; where smooth, use large λ.

11.2 The formulation

Definition 11.1 — Adaptive P-spline (Lang–Brezger, 2004)

Replace the constant λ with a vector of local penalties $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m-q})$. The adaptive P-spline solves:

$$\min_{\mathbf{c}} \ \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{B}\mathbf{c})\|^2 + \sum_{j=1}^{m-q} \tau_j (\Delta^q c_j)^2.$$

Equivalently,

$$\min_{\mathbf{c}} \ \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{B}\mathbf{c})\|^2 + \mathbf{c}^T \mathbf{D}_q^T \mathbf{T} \mathbf{D}_q \mathbf{c}$$

where $\mathbf{T} = \operatorname{diag}(\tau_1, \ldots, \tau_{m-q})$.

11.3 Estimating the local penalties

We cannot estimate each τ_j from data alone — too many parameters. Instead we impose a higher-level prior: the τ_j vary smoothly with j. Two approaches dominate:

11.3.1 Bayesian hierarchical approach

Treat τ_j as random with an inverse-gamma prior, estimate jointly via MCMC or variational methods. Original Lang–Brezger formulation.

11.3.2 Penalized penalty approach

Add a second-level penalty on the τ_j:

$$\min_{\mathbf{c},\boldsymbol{\tau}} \ \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{B}\mathbf{c})\|^2 + \sum_j \tau_j (\Delta^q c_j)^2 + \nu \sum_j (\Delta^r \log \tau_j)^2.$$

The log-transform keeps $\tau_j > 0$. Parameter ν controls how smoothly the penalties are allowed to vary. Deterministic and efficient.

11.4 When to use adaptive versus standard P-splines

Situation	Recommended approach
Uniformly smooth function	Standard P-spline
Smooth with isolated features	Adaptive P-spline
Known discontinuities	P-spline + explicit jump basis
Smooth trend + localized stress + known calendar jumps	**Adaptive P-spline + jump basis**

The last row is where we are headed. A repo curve simultaneously exhibits broad smooth trends (GC term structure), localized features (specialness on particular tenors), and scheduled discontinuities (turn effects at quarter-end). No single tool suffices; we need all three in combination.

Part IV

Application to Repo Curves

12 Anatomy of a Repo Curve

Before choosing a fitting method, we must understand the object being fit. Repo curves have a rich microstructure that constrains the class of models that can represent them faithfully.

12.1 What is a repo rate?

A repurchase agreement is a short-term collateralized loan: the borrower sells a security at time t_0 and agrees to buy it back at time $t_0 + \tau$ at a pre-agreed higher price. The implied interest rate on this transaction is the **repo rate** $r(\tau)$.

The **repo curve** is the function $\tau \mapsto r(\tau)$ for a given collateral and settlement date. In practice we observe rates at a discrete set of tenors: overnight, one week, one month, three months, six months, one year.

12.2 The decomposition

Structural decomposition of the repo rate

$$r_i(\tau) = r_{\text{GC}}(\tau) + s_i(\tau) + \sum_k \delta_k(\tau) \cdot \phi_k(\tau)$$

- $r_{\text{GC}}(\tau)$: the **general collateral** rate — the rate for financing against any eligible security.
- $s_i(\tau)$: the **specialness** of security i — the discount applied when security i is in high demand.
- $\phi_k(\tau)$: indicator for whether the repo period crosses calendar date t_k (end of month, quarter, year).
- $\delta_k(\tau)$: the **jump amplitude** associated with crossing date t_k.

12.3 The smooth component: general collateral

The GC curve $r_{\text{GC}}(\tau)$ behaves much like a money-market yield curve: a smooth function of tenor, driven by central bank policy, term premia, and market expectations. It is the only component well-suited to traditional smoothing methods.

Standard shapes include upward sloping in normal conditions, downward sloping near end of hiking cycles, and hump-shaped when short-term tensions drive up 1M–3M rates relative to overnight and 1Y.

12.4 The specialness component

Specialness $s_i(\tau)$ reflects security-specific demand — investors shorting a bond must borrow it via repo, driving the repo rate *down* for that security. Characteristic patterns:

- **Concentrated in short tenors:** acute for overnight and one-week, dissipating beyond one month.
- **Episodic:** a security becomes special when actively shorted (newly-issued benchmark bonds, squeezes).
- **Localized in time:** specialness evolves day by day.

12.5 Calendar jumps: turn effects

The most structural feature of repo curves is the presence of **calendar jumps**. When a repo period spans a regulatory reporting date — especially year-end — banks face balance-sheet constraints that make financing more expensive. The result is a *discontinuous* jump for tenors that include the turn date.

Turn date	Typical (EUR)	Typical (USD)
End of month	1–5 bp	2–8 bp
End of quarter	5–25 bp	10–40 bp
End of year	20–80 bp	40–150 bp
Major events (IMM, BCE)	variable	variable

Year-end turns in stress periods (end-2018 in USD, end-2022 in EUR) have exceeded 300 basis points. These are not smooth features. They are genuine discontinuities in the mathematical sense — the repo rate as a function of tenor jumps abruptly when the tenor crosses the turn threshold.

12.6 The full picture

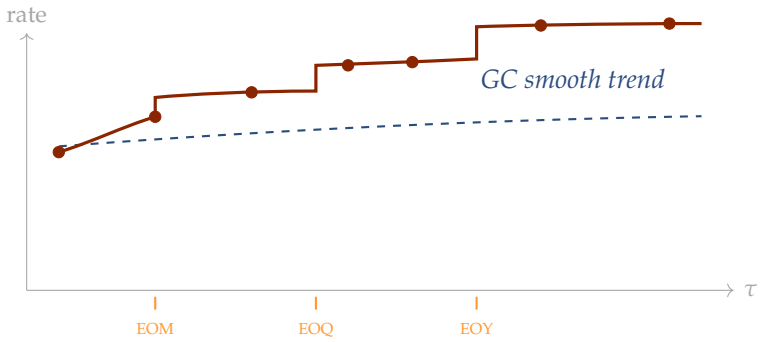

Figure 12.1 — A stylized repo curve. The smooth GC trend (blue dashed) is masked by calendar jumps at month-, quarter-, and year-end (orange ticks). A naive smoother would either smear the jumps or chase them.

13 Why Standard Smoothers Fail

Each standard smoothing method has a structural incompatibility with some feature of repo curves. A careful diagnosis of these failures motivates the specific design of the adaptive P-spline with jump basis.

13.1 Nelson-Siegel and Svensson

The Nelson-Siegel model represents yields as:

$$y(\tau) = \beta_0 + \beta_1 \frac{1 - e^{-\tau/\lambda}}{\tau/\lambda} + \beta_2 \left[\frac{1 - e^{-\tau/\lambda}}{\tau/\lambda} - e^{-\tau/\lambda} \right].$$

This three-factor model works brilliantly for government yield curves because those curves are driven by three factors (level, slope, curvature) with smooth dynamics. It fails for repo curves because:

- The basis functions are C^∞, incapable of representing jumps.
- Only a single hump is allowed; multiple localized features cannot be captured.
- The parameters are global; a local stress at overnight distorts the entire curve.
- Specialness patterns are simply not expressible in the Nelson-Siegel basis.

13.2 Global polynomial fitting

Fitting a single polynomial (degree p) fails for the same reason Runge's phenomenon exists: polynomials cannot simultaneously achieve local flexibility and global stability. High-degree polynomials oscillate; low-degree polynomials cannot capture the range of shapes that repo curves exhibit.

13.3 Classical smoothing splines

Reinsch's smoothing spline with a single λ confronts a different problem: *non-adaptivity*.

If we choose λ to respect the smoothness of the GC trend, jumps at turn dates get smeared across several days — the fit creeps up gradually rather than jumping. This produces visibly wrong behavior: fitted rates for tenors just before year-end are biased high, and just after, biased low.

Conversely, if we choose λ small enough to capture jumps, the fit overreacts to noise in smooth regions, producing spurious wiggles between tenors.

The core diagnostic

A single smoothness parameter cannot simultaneously accommodate:
1. Broad smooth trends (wanting large λ),
2. Known discontinuities (wanting small λ at specific points),
3. Localized stress features (wanting variable λ).

Any single-parameter smoother will fail on at least one of these axes.

13.4 LOESS and local regression

LOESS fits a local polynomial at each query point using kernel-weighted nearby observations. It handles localized features well but struggles with jumps: the local kernel blurs the discontinuity over a bandwidth-wide window.

Adaptive-bandwidth LOESS can narrow the kernel at stress points, but this is still fundamentally a smoothing operation — it cannot represent a true discontinuity. And LOESS has no parametric representation, making it inconvenient for pricing (where we need derivatives and integrals of the curve).

13.5 Monotone / shape-constrained splines

Some curve-building methods impose monotonicity or convexity. These are inappropriate for repo curves: turn effects and specialness genuinely produce non-monotone behavior. Imposing such constraints amounts to ignoring the most important features of the data.

13.6 Summary of failure modes

Method	Smooth trend	Jumps	Local features
Nelson-Siegel	restricted	55	55
Global polynomial	unstable	55	55
Standard P-spline	51	55	limited
Classical smoothing spline	51	55	limited
LOESS	51	blurred	51
Adaptive P-spline	51	55	51
Adaptive P-spline + jumps	**51**	**51**	**51**

The final row is the combination we now construct: adaptive P-splines handle the smooth trend and localized stress; explicit jump basis functions handle the calendar jumps.

14 The Adaptive P-Spline with Jump Components

The complete model for repo curves combines three ingredients: a B-spline basis for the smooth trend, adaptive penalties for local features, and explicit indicator functions for calendar jumps. This chapter assembles the full architecture.

14.1 The complete model

Full specification

The fitted repo rate at tenor τ is:

$$\hat{r}(\tau) = \underbrace{\sum_{j=1}^{m} c_j B_j(\tau)}_{\text{smooth component}} + \underbrace{\sum_{k=1}^{M} \delta_k \phi_k(\tau)}_{\text{jump component}} .$$

The coefficients $\mathbf{c}$ and jump amplitudes δ are estimated jointly by minimizing:

$$J(\mathbf{c}, \delta) = \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{B}\mathbf{c} - \mathbf{\Phi}\delta)\|^2 + \mathbf{c}^T \mathbf{P}_{\text{adapt}} \mathbf{c} + \mu \|\delta\|^2.$$

where $\mathbf{P}_{\text{adapt}} = \mathbf{D}_q^T \mathbf{T} \mathbf{D}_q$ is the adaptive penalty and $\mu > 0$ is a ridge parameter stabilizing the jump estimates.

14.2 Component-by-component justification

14.2.1 Why a B-spline basis?

B-splines have compact support, are computationally efficient, and form a natural basis for the smooth trend. Moderate m (20–30 knots for an O/N-to-1Y curve) gives enough flexibility without overparameterization.

14.2.2 Why adaptive penalties?

The smoothness of the GC trend varies along the curve. Near overnight, liquidity is high and the curve is well-pinned by data; in the 1M–6M region, fewer data points are available and more smoothness is desirable. Adaptive penalties encode this via local τ_j weights.

14.2.3 Why an explicit jump basis?

Turn effects are *known in advance*. We know the exact dates (end of quarter, end of year). Including them as explicit indicator functions ϕ_k lets the model represent discontinuities exactly, without compromising the smoothness of the underlying trend.

Crucially, $\phi_k(\tau) = 1$ if the repo period $[t_0, t_0 + \tau]$ crosses the turn date t_k, and 0 otherwise. This is a step function with a single discontinuity, which no B-spline basis of finite degree can represent.

14.2.4 Why a ridge penalty on δ?

Some turns may be "inactive" on a given day — if no data point straddles the turn cleanly, the jump is poorly identified. A small ridge $\mu\|\delta\|^2$ shrinks unidentified jumps toward zero, regularizing without biasing well-identified jumps.

14.3 Matrix form and solution

Define the augmented design matrix and parameter vector:

$$\mathbf{X} = [\mathbf{B} \mid \mathbf{\Phi}], \qquad \boldsymbol{\theta} = \begin{pmatrix} \mathbf{c} \\ \boldsymbol{\delta} \end{pmatrix}.$$

Define the block-diagonal penalty matrix:

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{\text{adapt}} & \mathbf{0} \\ \mathbf{0} & \mu\mathbf{I} \end{pmatrix}.$$

Then the normal equations are:

$$(\mathbf{X}^T\mathbf{W}\mathbf{X} + \mathbf{P})\hat{\boldsymbol{\theta}} = \mathbf{X}^T\mathbf{W}\mathbf{y}.$$

A single sparse symmetric positive-definite system, solvable in $O((m+M)^3)$ time — milliseconds for repo curves.

14.4 Selecting the penalties

Three tuning parameters appear: the adaptive penalty vector $\boldsymbol{\tau}$ (summarized by level λ_0 and adaptation strength ν), and the jump ridge μ.

- λ_0 is chosen by GCV or REML on a held-out grid.
- ν is typically fixed at a moderate value (e.g. $\nu = 1$) unless there is strong prior information.
- μ is set small — typically 10^{-4} to 10^{-2} — to regularize without biasing the fit.

14.5 Diagnostics

After fitting, several checks ensure the model has behaved sensibly:

1. **Residuals:** small and apparently random, no structure suggesting unmodeled features.
2. **Effective degrees of freedom:** sensible range (typically 8–15 for a 20-point repo curve).
3. **Jump estimates:** consistent with market intuition (positive for year-end in stress, small for normal end-of-month).
4. **Stability across days:** fitted $\mathbf{c}$ and δ should evolve smoothly day-to-day.

14.6 Extensions

The architecture accommodates natural extensions:

- **Multi-currency joint fitting:** share the jump structure across currencies with different GC components.
- **Time series smoothing:** impose smoothness of $\mathbf{c}_t$ and δ_t across calendar dates t, yielding a 2D adaptive P-spline.
- **Forward curve extraction:** differentiate the fit analytically to obtain instantaneous forward rates.
- **No-arbitrage constraints:** impose that forward rates remain non-negative or bounded.

15 Summary and Path Forward

We have traveled from shipbuilding in the 18th century to the frontier of non-parametric regression, arriving at a specific architecture for repo curves. This final chapter consolidates the logic and charts the next steps.

15.1 The logical spine

The chain of reasoning, stripped to essentials:

1. **Physical origin.** A flexible strip minimizes bending energy.
2. **Mathematical abstraction.** This minimization yields piecewise cubic polynomials with C^2 continuity — the natural cubic spline.
3. **Structural theory.** Splines form a finite-dimensional vector space; B-splines are a computationally optimal basis.
4. **Statistical extension.** Reinsch's theorem generalizes interpolation to smoothing: adding a penalty term yields the smoothing spline, whose solution is still a natural cubic spline.
5. **Computational simplification.** Eilers–Marx P-splines replace the integral penalty with a finite-difference penalty on B-spline coefficients.
6. **Local adaptation.** Allowing the penalty to vary across the domain (adaptive P-splines) accommodates non-uniform smoothness.
7. **Jump augmentation.** Adding explicit indicator functions to the basis accommodates known discontinuities.
8. **Application.** Repo curves exhibit smooth trend + local stress + known calendar jumps — exactly the three features the architecture addresses.

15.2 Why this method is right

The verdict

The adaptive P-spline with explicit jump basis is the minimal architecture that faithfully represents the three structural features of a repo curve:

1. It is not more complex than necessary — each component addresses a specific feature.
2. Each component is mathematically principled, with a clear rationale derived from the data structure.
3. It yields a compact parametric representation useful for pricing and risk.
4. It is computationally efficient, solvable in milliseconds on standard hardware.
5. It is diagnosable — we can test whether each component is doing its job.

15.3 What comes next

With the theoretical foundation in place, the next phase is implementation and calibration:

- **Data pipeline:** assemble a clean time series of repo observations across tenors, with attention to data quality filters.
- **Knot placement:** determine the knot grid for the B-spline basis. Equal spacing if basis is large enough; otherwise, density should reflect data density.
- **Calendar inventory:** compile turn dates over the curve horizon.
- **Penalty calibration:** run GCV / REML on historical data to select the baseline λ_0; calibrate ridge μ to avoid excessive shrinkage of large jumps.
- **Backtesting:** compare fits to held-out observations, assess jump recovery in known stress episodes (year-end 2022, March 2020, etc.).
- **Production integration:** embed in pricing, risk, and P&L systems with sufficient diagnostics to

detect fit failures in real time.

15.4 Closing reflection

The path from a shipbuilder's flexible wooden strip to an adaptive P-spline with jump components traverses more than two centuries of mathematical development. The journey is instructive: each step was motivated by a concrete limitation of the previous one, and each resolution preserved what worked while generalizing what did not.

This is the character of sound quantitative work. We inherit a toolbox, understand why each tool was invented and what it cannot do, and — when we encounter a problem that exceeds the capability of any existing tool — we compose new tools from the old, with full awareness of the structural assumptions that underpin each piece.

The adaptive P-spline with jump basis is not a novel invention; every component has been in the statistical literature for years. What matters is the *fit* of the architecture to the structure of repo data. That fit is the result of understanding both the mathematics and the market — and it is this understanding that transfers to the next problem we face.

Selected references: Schoenberg (1946); Holladay (1957); Reinsch (1967); de Boor (1978); Eilers & Marx (1996); Craven & Wahba (1979); Lang & Brezger (2004); Wahba (1990) Spline Models for Observational Data*; Wood (2017)* Generalized Additive Models: An Introduction with R.