Tension Spline Algorithm for Building Commodity Forward Curves

Jake C. Fowler

THIS DOCUMENT IS CURRENTLY WORK IN PROGRESS

1 Introduction

1.1 Reason for Interpolating Commodity Forward Curves

The aim of a curve interpolation routine is to take a collection of traded forward prices, and tranform these into a forward curve of homogenous granularity. Typically the derived curve can constructed to be in a granularity higher than what is traded in the market. But why would anyone want to derive interpolated forward prices given that we cannot actually trade at these prices? Often physical positions and structured deals will have a volume profile which is at a higher granularity than forward market prices. For example natural gas storage which can be withdrawn quickly over one or two weeks. If such a deal were to be valued using traded contracts, and the Q1 price is the highest granularity of such traded contracts, then such a deal would be undervalued.

The interpolated forward curve should consist of prices, prior to bid-offer and any other adjustment, at which we are willing to mark positions and execute trades for volume profiles at higher granularity to the traded forward market.

Next we need to think about the functional form of this interpolation.

TODO: example of why we don't want to interpolate piecewise flat. TODO: replicate high-granularity positions by rolling forward price hedges.

2 Deriving the Algorithm

2.1 Functional Form

The base of the algorithm is a spline, which by definition is made up of piecewise polynomial functions.

$$p(t) = \begin{cases} p_1(t) & \text{for} \quad t \in [t_0, t_1) \\ p_2(t) & \text{for} \quad t \in [t_1, t_2) \\ \vdots & \\ p_{n-1}(t) & \text{for} \quad t \in [t_{n-2}, t_{n-1}) \\ p_n(t) & \text{for} \quad t \in [t_{n-1}, t_n] \end{cases} \tag{1}$$

Where $t_0 < t_1 < \ldots < t_{n-1} < t_n$ are the boundary points between the polynomials which make up the spline. In the context of building a forward curve, the variable t is defined as the time until start of delivery of a forward contract.

The boundary points are chosen to be start of the input forward prices. It is also assumed that the input forward prices are not for delivery periods which overlap with any other input. Gaps between input forward contracts are permitted, in which case a boundary point will exit for the start of the gap.

$$\begin{aligned} p_i(t) = & \frac{z_{i-1} \sinh(\tau_i(t_i - t)) + z_i \sinh(\tau_i(t - t_{i-1}))}{\tau_i^2 \sinh(\tau h_i)} \\ & + \frac{(y_{i-1} - z_{i-1}/\tau_i^2)(t_i - t) + (y_i - z_i/\tau_i^2)(t - t_{i-1})}{h_i} \end{aligned} \tag{2}$$

Where $h_i = t_i - t_{i-1}$. $z_i = p''(t_i)$ and $y_i = p(t_i)$, i.e. the (as yet unknown) value of the function at the boundary points.

The curve fitting algorithm essentially involves solving for the parameters z_i, and y_i for $i = 0 \ldots n$.

In many cases the spline described above is not sufficient to derive a forward curve which shows strong price seasonality, especially when this seasonality cannot be directly observed in the traded forward prices. An example of this is the day-of-week seasonality for gas and power prices, which generally are lower at the weekend when demand is lower. As such the function form is as follows:

$$f(t) = (p(t) + S_{add}(t)) S_{mult}(t) \tag{3}$$

Where the forward price for the period starting delivery at time t is given by $f(t)$, which consists of $p(t)$ adjusted by two arbitrary seasonal adjustment functions $S_{add}(t)$ an additive adjustment, and $S_{mult}(t)$ a multiplicative adjustment.

2.2 Constraints

2.2.1 Polynomial Boundary Point Constraints

As usual with splines, constraints are put in place that adjascent polynomials have equal value, first derivative, and second derivatives at the boundary points.

2.2.2 Polynomial Value Boundary Point Equality

To make $p(t)$ continuous we need to constrain $p_i(t_{i-1}) = p_{i-1}(t_{i-1})$. Evaluating both of these:

$$\begin{aligned} p_i(t_{i-1}) = \frac{z_{i-1}\sinh(\tau_i h_i) + z_i \sinh(0)}{\tau_i^2 \sinh(\tau_i h_i)} &+ \frac{(y_{i-1} - z_{i-1}/\tau_i^2)h_i}{h_i} \\ &= \frac{z_{i-1}}{\tau_i^2} + y_{i-1} - \frac{z_{i-1}}{\tau_i^2} \\ &= y_{i-1} \end{aligned} \tag{4}$$

$$\begin{aligned} p_{i-1}(t_{i-1}) = \frac{z_{i-2}\sinh(0) + z_{i-1}\sinh(\tau_{i-1}h_{i-1})}{\tau_{i-1}^2 \sinh(\tau_{i-1}h_{i-1})} &+ \frac{(y_{i-1} - z_{i-1}/\tau_{i-1}^2)h_{i-1}}{h_{i-1}} \\ &= \frac{z_{i-1}}{\tau_{i-1}^2} + y_{i-1} - \frac{z_{i-1}}{\tau_{i-1}^2} \\ &= y_{i-1} \end{aligned}$$

Hence, by construction, $p(t)$ is always continuous with value y_{i-1} at the boundary between p_i and p_{i-1}

2.2.3 Polynomial First Derivative Boundary Point Equality

This is to constrain $p'_i(t_{i-1}) = p'_{i-1}(t_{i-1})$. First finding the expression for $p'_i(t)$:

$$\begin{aligned} p_i(t) = &\frac{z_{i-1}\sinh(\tau_i(t_i - t)) + z_i \sinh(\tau_i(t - t_{i-1}))}{\tau_i^2 \sinh(\tau h_i)} \\ &+ \frac{(y_{i-1} - z_{i-1}/\tau_i^2)(t_i - t) + (y_i - z_i/\tau_i^2)(t - t_{i-1})}{h_i} \end{aligned} \tag{5}$$

$$\begin{aligned} p'_i(t) = &\frac{-z_{i-1}\cosh(\tau_i(t_i - t)) + z_i \cosh(\tau_i(t - t_{i-1}))}{\tau_i \sinh(\tau_i h_i)} \\ &+ \frac{y_i - y_{i-1} + (z_{i-1} - z_i)/\tau_i^2}{h_i} \end{aligned} \tag{6}$$

For clarity, rearranging this to highlight the linearity with respect to the parameters:

$$p_i'(t) = z_i\left(\frac{\cosh(\tau_i(t-t_{i-1}))}{\tau_i\sinh(\tau_i h_i)} - \frac{1}{h_i\tau_i^2}\right) + z_{i-1}\left(\frac{1}{h_i\tau_i^2} - \frac{\cosh(\tau_i(t_i-t))}{\tau_i\sinh(\tau_i h_i)}\right) + y_i\frac{1}{h_i} - y_{i-1}\frac{1}{h_i} \tag{7}$$

Evaluating this about the boundary points:

$$p_i'(t_{i-1}) = z_i\left(\frac{1}{\tau_i\sinh(\tau_i h_i)} - \frac{1}{h_i\tau_i^2}\right) + z_{i-1}\left(\frac{1}{h_i\tau_i^2} - \frac{\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)}\right) + y_i\frac{1}{h_i} - y_{i-1}\frac{1}{h_i} \tag{8}$$

$$p_{i-1}'(t_{i-1}) = z_{i-1}\left(\frac{\cosh(\tau_{i-1}h_{i-1})}{\tau_{i-1}\sinh(\tau_{i-1}h_{i-1})} - \frac{1}{h_{i-1}\tau_{i-1}^2}\right) + z_{i-2}\left(\frac{1}{h_{i-1}\tau_{i-1}^2} - \frac{1}{\tau_{i-1}\sinh(\tau_{i-1}h_{i-1})}\right) + y_{i-1}\frac{1}{h_{i-1}} - y_{i-2}\frac{1}{h_{i-1}} \tag{9}$$

Setting these equal:

$$\begin{aligned}0 = z_i&\left(\frac{1}{\tau_i\sinh(\tau_i h_i)} - \frac{1}{h_i\tau_i^2}\right)\\ &+ z_{i-1}\left(\frac{1}{h_i\tau_i^2} - \frac{\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)} - \frac{\cosh(\tau_{i-1}h_{i-1})}{\tau_{i-1}\sinh(\tau_{i-1}h_{i-1})} + \frac{1}{h_{i-1}\tau_{i-1}^2}\right)\\ &- z_{i-2}\left(\frac{1}{h_{i-1}\tau_{i-1}^2} - \frac{1}{\tau_{i-1}\sinh(\tau_{i-1}h_{i-1})}\right)\\ &+ y_i\frac{1}{h_i} - y_{i-1}\left(\frac{1}{h_i} + \frac{1}{h_{i-1}}\right) + y_{i-2}\frac{1}{h_{i-1}}\end{aligned} \tag{10}$$

This constraint should be held for $i = 2\dots n$.

The above three equation should hold for the boundary points $t \in \{t_1, t_2, \dots, t_{n-2}, t_{n-1}\}$.

2.2.4 Forward Price Constraint

The most important constraints is that the derived forward curve averages back to the input traded forward prices. The market inputs to the forward curve model are traded forward prices F_i. Setting this equal to the average of the derived smooth curve:

$$F_j = \frac{\sum_{t\in T_j}(p(t) + S_{add}(t))S_{mult}(t)w(t)D(t)}{\sum_{t\in T_j} w(t)D(t)} \tag{11}$$

Where $D(t)$ is the discount factor from the settlement date of delivery period t. $w(t)$ is a weighting function and T_i is the set of all delivery start times for the delivery periods at the granularity of the curve being built. The weighting function has two meanings from a busines perspective.

- The volume of commodity delivered in each period. For example, an off-peak power forward contract in the UK delivers over 12 hours in on weekdays, and 24 hours on weekends, hence $w(t)$ would equal double for t representing weekends compared to $w(t)$ when t represents a weekday delivery. Clock changes can also cause the total volume delivered over a day in a fixed time zone to vary due to hours lost or gained. Hence $w(t)$ can be used to account for this.

- For swaps which only fix on certain days (usually business days) $w(t)$ can be used to account for this by returning the number of fixing days in the period starting at t. For example if deriving a a monthly curve $w(t)$ would evaluate to the number of fixing days in the month starting at t.

Equation 11 can be transformed into an equation linear on the parameters of the piecewise polynomial by substituting in the polynomial representation of $p(t)$:

$$\sum_i \sum_{t \in T_j \cap [t_{i-1}, t_i)} p_i(t) S_{mult}(t) w(t) D(t) = F_j \sum_{t \in T_j} w(t) D(t) - \sum_{t \in T_i} S_{add}(t) S_{mult}(t) w(t) D(t) \tag{12}$$

Substituting in for $p_i(t)$:

$$\sum_i \sum_{t \in T_i \cap [t_{i-1}, t_i)} \Bigg(\frac{z_{i-1} \sinh(\tau_i(t_i - t)) + z_i \sinh(\tau_i(t - t_{i-1}))}{\tau_i^2 \sinh(\tau_i h_i)}$$
$$+ \frac{(y_{i-1} - z_{i-1}/\tau_i^2)(t_i - t) + (y_i - z_i/\tau_i^2)(t - t_{i-1})}{h_i} + S_{add}(t) \Bigg) S_{mult}(t) w(t) D(t)$$
$$= F_i \sum_{t \in T_i} w(t) D(t) - \sum_{t \in T_i} S_{add}(t) S_{mult}(t) w(t) D(t) \tag{13}$$

Rearranging again gives a form linear with respect to the unknown polynomial coefficients z_i, z_{-1}, y_i and y_{i-1}.

$$
\begin{aligned}
\sum_i \Bigg(z_i & \sum_{t\in T_j\cap[t_{i-1},t_i)} \left(\frac{\sinh(\tau_i(t-t_{i-1}))}{\tau_i^2 \sinh(\tau_i h_i)} - \frac{t-t_{i-1}}{\tau_i^2 h_i} \right) S_{mult}(t) w(t) D(t) \\
& + z_{i-1} \sum_{t\in T_j\cap[t_{i-1},t_i)} \left(\frac{\sinh(\tau_i(t_i-t))}{\tau_i^2 \sinh(\tau_i h_i)} - \frac{t_i-t}{\tau_i^2 h_i} \right) S_{mult}(t) w(t) D(t) \\
& \quad + y_i \sum_{t\in T_j\cap[t_{i-1},t_i)} \frac{(t-t_{i-1})}{h_i} S_{mult}(t) w(t) D(t) \\
& \quad + y_{i-1} \sum_{t\in T_j\cap[t_{i-1},t_i)} \frac{(t_i-t)}{h_i} S_{mult}(t) w(t) D(t) \Bigg) \\
& \qquad = F_j \sum_{t\in T_i} w(t) D(t) - \sum_{t\in T_i} S_{add}(t) S_{mult}(t) w(t) D(t) \qquad (14)
\end{aligned}
$$

This constraint should be held for $j = 1 \ldots n$.

2.2.5 Matrix Form of Constraints

Start with forward price constraint as less lags (probably)

$$
\alpha_i^j = \qquad (15)
$$

Superscript is for contract, supersci

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha_0^1 & \beta_0^1 & \gamma_0^1 & \delta_0^1 & \alpha_1^1 & \beta_1^1 & \ldots & 0 & 0 & \alpha_n^1 & \beta_n^1 & \gamma_n^1 & \delta_n^1 \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha_0^2 & \beta_0^2 & \gamma_0^2 & \delta_0^2 & \alpha_1^2 & \beta_1^2 & \ldots & 0 & 0 & \alpha_n^2 & \beta_n^2 & \gamma_n^2 & \delta_n^2 \\
\vdots & & & & & & \ddots & & & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha_0^n & \beta_0^n & \gamma_0^n & \delta_0^n & \alpha_1^n & \beta_1^n & \ldots & 0 & 0 & \alpha_n^n & \beta_n^n & \gamma_n^n & \delta_n^n \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix} z_0 \\ y_0 \\ z_1 \\ y_1 \\ \vdots \\ z_{n-1} \\ y_{n-1} \\ z_n \\ y_n \end{bmatrix}
=
\begin{bmatrix} 0 \\ f_i \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

2.3 Smoothness Criteria

We want to find the solution which minimises the following.

$$
\int_{t_0}^{t_n} \left(p''(t)^2 + \tau_i^2 p'(t)^2 \right) dt = \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} p_i''(t)^2 + \tau_i^2 p'(t)^2 dt \qquad (16)
$$

The integral over $p''(t)^2$ penalises the change in curve direction seen in oscillations. The $p'(t)^2$ term penalises total curve length of oscillations. A reason for this being weighted by τ_i^2 can be seen as follows. In the case of high tension parameter, the pieceswise function will be virtually linear, hence $p''(t)^2$ will be tiny, so there is no point in penalising for this. Rather, we would rather penalise for the amplitute of zigzagging which can be seen when interpolating with a piecewise linear function.

Note that the final forward curve is actually read off discrete points from the piecewise function, hence the penalty function is more appropriately represented as a summation of $p''(t)^2 + \tau_i^2 p'(t)^2$ over these discrete points, rather than an integral. However, as shown below, the integral can be evaluated to a convenient form, and given we are usually interpolating to a high granularity, the spacing of the discrete points will be small, so the integral will be a close approximation of a summation.

See Appendix A for the evaluation of this integral to the following.

$$\begin{aligned}
&= z_i^2\left(\frac{\cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} - \frac{1}{\tau_i^2 h_i}\right) + z_{i-1}^2\left(\frac{\cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} - \frac{1}{\tau_i^2 h_i}\right) \\
&\qquad + y_i^2\frac{\tau_i^2}{h_i} + y_{i-1}^2\frac{\tau_i^2}{h_i} + z_i z_{i-1} 2\left(\frac{1}{\tau_i^2 h_i} - \frac{1}{\tau_i \sinh(\tau_i h_i)}\right) \\
&\qquad\qquad - y_i y_{i-1}\frac{2\tau_i^2}{h_i}
\end{aligned} \tag{17}$$

Writing this in symmetric matrix form.

$$= \sum_{i=1}^{n} \begin{bmatrix} z_{i-1} \\ y_{i-1} \\ z_i \\ y_i \end{bmatrix}^T \begin{bmatrix} \left(\frac{\cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} - \frac{1}{\tau_i^2 h_i}\right) & 0 & \left(\frac{1}{\tau_i^2 h_i} - \frac{1}{\tau_i \sinh(\tau_i h_i)}\right) & 0 \\ 0 & \frac{\tau_i^2}{h_i} & 0 & -\frac{\tau_i^2}{h_i} \\ \left(\frac{1}{\tau_i^2 h_i} - \frac{1}{\tau_i \sinh(\tau_i h_i)}\right) & 0 & \left(\frac{\cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} - \frac{1}{\tau_i^2 h_i}\right) & 0 \\ 0 & -\frac{\tau_i^2}{h_i} & 0 & \frac{\tau_i^2}{h_i} \end{bmatrix} \begin{bmatrix} z_{i-1} \\ y_{i-1} \\ z_i \\ y_i \end{bmatrix} \tag{18}$$

This can be changed from a sum to a single quadratic form.

$$= \begin{bmatrix} z_0 \\ y_0 \\ z_1 \\ y_1 \\ \vdots \\ z_{n-1} \\ y_{n-1} \\ z_n \\ y_n \end{bmatrix}^T \mathbf{H} \begin{bmatrix} z_0 \\ y_0 \\ z_1 \\ y_1 \\ \vdots \\ z_{n-1} \\ y_{n-1} \\ z_n \\ y_n \end{bmatrix} \tag{19}$$

$$
\mathbf{H} = \begin{bmatrix}
\left(\frac{\cosh(\tau_1 h_1)}{\tau_1 \sinh(\tau_1 h_1)} - \frac{1}{\tau_1^2 h_1}\right) & 0 & \left(\frac{1}{\tau_1^2 h_1} - \frac{1}{\tau_1 \sinh(\tau_1 h_1)}\right) & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & \frac{\tau_1^2}{h_1} & 0 & -\frac{\tau_1^2}{h_1} & 0 & 0 & \dots & 0 & 0 \\
\left(\frac{1}{\tau_1^2 h_1} - \frac{1}{\tau_1 \sinh(\tau_1 h_1)}\right) & 0 & \left(\frac{\cosh(\tau_1 h_1)}{\tau_1 \sinh(\tau_1 h_1)} - \frac{1}{\tau_1^2 h_1}\right) + \left(\frac{\cosh(\tau_2 h_2)}{\tau_2 \sinh(\tau_2 h_2)} - \frac{1}{\tau_2^2 h_2}\right) & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & -\frac{\tau_1^2}{h_1} & 0 & \frac{\tau_1^2}{h_1} + \frac{\tau_2^2}{h_2} & 0 & 0 & \dots & 0 & 0 \\
\vdots & & & & \ddots & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0
\end{bmatrix} \tag{20}
$$

3 Minimisation Problem

The sections above show that finding the maximum smoothness curve comes down to finding the coefficients, vector $\mathbf{x}$, which minimises $\mathbf{x^T H x}$, subject to the linear constraints $\mathbf{Ax} = \mathbf{b}$. This problem is well suited to the method of Lagrange multipliers for which we first define the vector λ and Lagrangian function $\mathcal{L}(\mathbf{x}, \lambda)$.

$$
\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{4n-2} \\ \lambda_{4n-1} \end{bmatrix} \tag{21}
$$

$$
\mathcal{L}(\mathbf{x}, \lambda) = \mathbf{x^T H x} + \lambda^{\mathbf{T}}(\mathbf{Ax} - \mathbf{b}) \tag{22}
$$

As $\mathbf{x^T H x}$ is a convex function, the minima $\min\limits_{x,\lambda} \mathcal{L}(\mathbf{x}, \lambda)$ is found as the solution where the partial derivatives of $\mathcal{L}(\mathbf{x}, \lambda)$ with respect to $\mathbf{x}$ and λ are zero.

$$
\frac{\mathcal{L}(\mathbf{x}, \lambda)}{\partial \mathbf{x}} = \mathbf{2Hx} + \mathbf{A^T}\lambda = 0 \tag{23}
$$

$$
\frac{\mathcal{L}(\mathbf{x}, \lambda)}{\partial \lambda} = \mathbf{Ax} - \mathbf{b} = 0 \tag{24}
$$

These can be arranged into a single linear system:

$$
\begin{bmatrix} \mathbf{2H} & \mathbf{A^T} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x_{min}} \\ \lambda_{\mathbf{min}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} \tag{25}
$$

Hence the vector of spline coefficients for the maximum smoothness curve, $\mathbf{x_{min}}$, can be found by solving this system.

The forward curve can then be calculated be evaluating the spline using the $\mathbf{x_{min}}$ parameters and then finally evaluating function (TODO add reference).

4 Future Work

4.1 Collinear Inputs

Any redundancy seen in the input market data (e.g. contracts for Q-1, Jan, Feb and Mar delivery all present) will result in matrix TODO above being singular, hence the linear system cannot be solved. Theoretically redundant input forward prices should be consistent and arbitrage free. However, real-world market frictions such as bid-offer spread, transaction costs and limited liquidity mean that small theoretically arbitragable inconsistencies between collinear forward prices are likely to be seen.

If the curves are built using a highly automated curve building tool this will be a problem. This section presents some ideas on how to tackle this.

4.1.1 Preprocess Input Contracts

It could make sense to discard a suffient set of input contracts such that the linear system is solvable. Firstly the criteria for which contracts are kept needs to be defined. As we are looking to build a curve with increased granularity this could be to prioritise contracts with shorter delivery period. Alternatively, the criteria could be based on liquidity.

Once the criteria has been defined, the input forward prices and shaping factors would be ordered, starting with the most favourable to be retained. The inputs would then be looped through, a matrix constructed containing the volume profiles, and then the rank of this matrix calculated. If the addition of any input does not increase the rank of the volume matrix, then this input is collinear with inputs that have more favourable selection criteria, hence should be discarded. Some additional thought would likely yield a more efficient algorithm which does not involve fully recalculating the rank after the addition of each input contract. For example a rank calculation which makes use of intermediate results in prior rank calculations.

One disadvantage of this approach is that market price information is being lost. The criteria for discarding contracts could lead prices which aren't good representations of the actual liquid prices being used.

4.1.2 Numerical Minimisation

An approach which uses and retains all input market data will need to account for the fact that the interpolated curve will not exactly average back to all input forward prices. The obvious approach is to solve for a curve which minimises the weighted sum of squared differences between input forward prices and the averaged interpolated curve. However, this problem is complicated by the linear system already having rank less than the number of unknowns being solved for, which is why the maximum smoothness solution is choosen from the infinite available solutions. We need to find an objective function which accounts for minimising both the squared difference to inputs contracts and maximises smoothness by minimising integral TODO equation reference. A simple objective function would be the weighted average of these two terms. A numerical minimisation routine would then be used to find the spline parameters which minimised this objective function. It's likely that the derivative of this objective function has an analytical form, hence a form of gradient descent could be used.

One problem with this approach is that the weighting in the objetive function needs to be arbitrarily specified to determine the relative importance of smoothness and closeness to input contract prices.

Another complication is that that the first derivative continuity constraints must be exactly met for a solution to be acceptable. This leaves us in the unusual situation of having a linear system where some constraints must be exactly met, whilst for other linear constraints we are happy with the least squares solution. As a starting point to solve this problem, the set of coefficient vectors which exactly obey the first derivative continuity can be found by finding the nullspace of a matrix containing these constraints.

4.1.3 Two Step Solution

To avoid the arbitrary weigting factor in the objective function in the approach described above one could consider that minimising the squared difference of interpolated curve to the input forward prices is more important than maximising the smoothness. A two-step solution would be to first find the general solution form (arbitrary solution and basis of the nullspace) which minimised the squared difference to market prices using an SVD. As a second step, a specific solution (as coefficients of the nullspace basis) which then maximises the smoothness is then solved for.

4.2 Alternative Tension Spline

There are alternative tension splines to the hyperbolic spline used in this document, for example the Cardinal Spline. The hyperbolic tension spline was chosen over the Cardinal Spline for no particularly reason. It is likely that there are pros and cons of the hyperbolic and Cardinal splines over each other for use in constructing commodity forward curves. A future project would be to implement a similar algorithm to the one in this document, but using the Cardinal Spline, and comparing the practicalities of each approach.

Appendices

A Maximum Smoothness Integral

This section evaluated the integral used in the maximum smoothness criteria. First writing the squares as multiples, and splitting the integral into two:

$$\int_{t_{i-1}}^{t_i} p_i''(t)^2 + \tau_i^2 p_i'(t)^2 dt = \int_{t_{i-1}}^{t_i} p_i''(t)p_i''(t)dt + \int_{t_{i-1}}^{t_i} \tau_i^2 p_i'(t)p_i'(t)dt \tag{26}$$

The first integral on the right-hand-side can be evaluated using integration by parts.

$$\int_{t_{i-1}}^{t_i} p_i''(t)p_i''(t)dt = p_i''(t_i)p_i'(t_i) - p_i''(t_{i-1})p_i'(t_{i-1}) - \int_{t_{i-1}}^{t_i} p_i'''(t)p_i'(t_{i-1})dt \tag{27}$$

Substituing this into the first equation in this appendix and combining the last two integrals into a single integral.

$$\int_{t_{i-1}}^{t_i} p_i''(t)^2+\tau_i^2 p'(t)^2 dt = p_i''(t_i)p_i'(t_i)-p_i''(t_{i-1})p_i'(t_{i-1})-\int_{t_{i-1}}^{t_i} \big(p_i'''(t)-\tau_i^2 p_i'(t)\big)p_i'(t)dt \tag{28}$$

Next, focussing on the $p_i'''(t) - \tau_i^2 p_i'(t)$ term. It has previously been shown that the first derivative is as follows:

$$p_i'(t) = \frac{-z_{i-1}\cosh(\tau_i(t_i - t)) + z_i \cosh(\tau_i(t - t_{i-1}))}{\tau_i \sinh(\tau_i h_i)} + \frac{y_i - y_{i-1} + (z_{i-1} - z_i)/\tau_i^2}{h_i} \tag{29}$$

Differentiating:

$$p_i''(t) = \frac{z_{i-1}\sinh(\tau_i(t_i - t)) + z_i \sinh(\tau_i(t - t_{i-1}))}{\sinh(\tau_i h_i)} \tag{30}$$

Differentiating again:

$$p_i'''(t) = \frac{-z_{i-1}\tau_i \cosh(\tau_i(t_i - t)) + z_i\tau_i \cosh(\tau_i(t - t_{i-1}))}{\sinh(\tau_i h_i)} \tag{31}$$

Using the above it can be seen that:

$$p_i'''(t) - \tau_i^2 p_i'(t) = -\frac{(y_i - y_{i-1})\tau_i^2 + z_{i-1} - z_i}{h_i} \tag{32}$$

Crucially this is independent of t, hence can be used to simply the equation.

$$\int_{t_{i-1}}^{t_i} \left(p_i'''(t) - \tau_i^2 p_i'(t)\right) p_i'(t) dt = -\frac{(y_i - y_{i-1})\tau_i^2 + z_{i-1} - z_i}{h_i} \int_{t_{i-1}}^{t_i} p_i'(t) dt$$
$$= -\frac{(y_i - y_{i-1})\tau_i^2 + z_{i-1} - z_i}{h_i} \left(p_i(t_i) - p_i(t_{i-1})\right) \quad (33)$$

Putting these results together.

$$\int_{t_{i-1}}^{t_i} p_i''(t)^2 + \tau_i^2 p'(t)^2 dt = p_i''(t_i) p_i'(t_i) - p_i''(t_{i-1}) p_i'(t_{i-1}) + \frac{(y_i - y_{i-1})\tau_i^2 + z_{i-1} - z_i}{h_i} \left(p_i(t_i) - p_i(t_{i-1})\right) \quad (34)$$

Evaluating the above involves substituting in for p_i and it's first two derivatives at the integral boundary points t_i and t_{i-1}. By construction we already know $p_i(t_i) = y_i$, $p_i(t_{i-1}) = y_{i-1}$, $p_i''(t_i) = z_i$ and $p_{i-1}''(t_{i-1}) = z_{i-1}$. The expressions for the first derivatives can be simplied by subsituting $\cosh(0) = 1$ and $t_i - t_{i-1} = h_i$.

$$p_i'(t_i) = \frac{z_i \cosh(\tau_i h_i) - z_{i-1}}{\tau_i \sinh(\tau_i h_i)} + \frac{y_i - y_{i-1} + (z_{i-1} - z_i)/\tau_i^2}{h_i} \quad (35)$$

$$p_i'(t_{i-1}) = \frac{z_i - z_{i-1} \cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} + \frac{y_i - y_{i-1} + (z_{i-1} - z_i)/\tau_i^2}{h_i} \quad (36)$$

Substituting these in:

$$\int_{t_{i-1}}^{t_i} p_i''(t)^2 + \tau_i^2 p'(t)^2 dt = z_i \left(\frac{z_i \cosh(\tau_i h_i) - z_{i-1}}{\tau_i \sinh(\tau_i h_i)} + \frac{y_i - y_{i-1} + (z_{i-1} - z_i)/\tau_i^2}{h_i} \right)$$
$$- z_{i-1} \left(\frac{z_i - z_{i-1} \cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} + \frac{y_i - y_{i-1} + (z_{i-1} - z_i)/\tau_i^2}{h_i} \right)$$
$$+ \frac{(y_i - y_{i-1})^2 \tau_i^2 + (y_i - y_{i-1})(z_{i-1} - z_i)}{h_i} \quad (37)$$

Rearranging.

$$= z_i \left(\frac{z_i \cosh(\tau_i h_i) - z_{i-1}}{\tau_i \sinh(\tau_i h_i)} \right) - z_{i-1} \left(\frac{z_i - z_{i-1} \cosh(\tau_i h_i)}{\tau_i \sinh(\tau_i h_i)} \right)$$
$$- \frac{(y_i - y_{i-1})(z_{i-1} - z_i) + (z_{i-1} - z_i)^2/\tau_i^2}{h_i}$$
$$+ \frac{(y_i - y_{i-1})^2 \tau_i^2 + (y_i - y_{i-1})(z_{i-1} - z_i)}{h_i} \quad (38)$$

Cancelling terms.

$$= z_i\left(\frac{z_i\cosh(\tau_i h_i) - z_{i-1}}{\tau_i\sinh(\tau_i h_i)}\right) - z_{i-1}\left(\frac{z_i - z_{i-1}\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)}\right) \\ + \frac{(y_i - y_{i-1})^2\tau_i^2 - (z_{i-1} - z_i)^2/\tau_i^2}{h_i} \tag{39}$$

Rearranging.

$$= (z_i^2 + z_{i-1}^2)\frac{\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)} - z_i z_{i-1}\frac{2}{\tau_i\sinh(\tau_i h_i)} \\ + \frac{(y_i - y_{i-1})^2\tau_i^2 - (z_{i-1} - z_i)^2/\tau_i^2}{h_i} \tag{40}$$

Multiplying out the squared differences.

$$= (z_i^2 + z_{i-1}^2)\frac{\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)} - z_i z_{i-1}\frac{2}{\tau_i\sinh(\tau_i h_i)} \\ + \frac{(y_i^2 + y_{i-1}^2 - 2y_i y_{i-1})\tau_i^2 - (z_i^2 + z_{i-1}^2 - 2z_i z_{i-1})/\tau_i^2}{h_i} \tag{41}$$

Rearranging again so it can be seen that this is a quadratic form.

$$= z_i^2\left(\frac{\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)} - \frac{1}{\tau_i^2 h_i}\right) + z_{i-1}^2\left(\frac{\cosh(\tau_i h_i)}{\tau_i\sinh(\tau_i h_i)} - \frac{1}{\tau_i^2 h_i}\right) \\ + y_i^2\frac{\tau_i^2}{h_i} + y_{i-1}^2\frac{\tau_i^2}{h_i} + z_i z_{i-1}2\left(\frac{1}{\tau_i^2 h_i} - \frac{1}{\tau_i\sinh(\tau_i h_i)}\right) \\ - y_i y_{i-1}\frac{2\tau_i^2}{h_i} \tag{42}$$