spline

Goal: Provide a"compact" approximated representation of f.
1st approach: Do a piecewise linear interpolation.
2nd approach: Iteratively insert new values at midpoints by averaging.
3rd approach: Subdivision (curve fitting)
Exact representation

$$f = \sum_{i,j,k} \langle f, \psi_{i,j,k} \rangle \psi_{i,j,k}$$

Approximate representation
From now on, our function will be defined on $[a, b]$ and continuous. That is, the natural class to deal with is

$$X := C[a, b] := \{f : [a, b] \to \mathbb{R} : f \text{ is continuous}\}$$

(a big linear space of functions)
There are too many functions in X to represent, so we will select much smaller class F in X. Let F be a small space of 'terrific' functions, i.e., a finite dimension space. For $h \in F$, we are talking about exact representation and for $f \in X - F$, we will only approximate them. In other words, we try to associate f with an approximation g in F. There are two issues in here:
1. How to represent g?
2. Whether this approximation is good or not?

Main question Given $f \in X$, find $g \in F$ such that

$$f \approx g$$

Representation of functions in an n-dimensional function space
Now, we will focus on the space F.
We will assume that F is a finite dimensional subspace with dimension n and $W = \{w_1, w_2, \cdots, w_n\}$ is a basis of F. With this basis, we can construct a synthesis map.

$$W : \mathbb{R}^n \longrightarrow F : \quad \begin{bmatrix} a(1) \\ \vdots \\ a(n) \end{bmatrix} \mapsto \sum_{j=1}^{n} a(j) w_j$$

We want to understand what is the relationship between the coefficients and the properties of the function obtained in this way from this synthesis

operator.
Can we do analysis? Yes
Let $\Lambda := \{\lambda_1, \cdots, \lambda_n\}$ be a collection of n linear functionals. And an analysis map is defined by

$$\Lambda' f := [\lambda_1 f, \cdots, \lambda_n f]$$

Theses linear functionals are linearly independent, but in what sense? The best way to realize this is that the following matrix

$$\Lambda' W := \begin{bmatrix} \lambda_1 w_1 & \cdots & \lambda_n w_1 \\ \vdots & \lambda_i w_j & \vdots \\ \lambda_n w_1 & \cdots & \lambda_n w_n \end{bmatrix}$$

should be invertible.

Definition We say that Λ is total for F,
if $\forall f \in F, \lambda_j f = 0$ for all $j = 1, \cdots, n$ then $f = 0$.
Λ is total for $F \iff \Lambda' W$ should be invertible.

Now, for a given $f \in X$, $W\Lambda' f = \sum_{j=1}^{n} \lambda_j f w_j$ is an approximation of f. In the best circumstances, we will have perfect reconstruction:

$$W\Lambda' = Id \text{ on} F$$

$$; W\Lambda' f = \sum_{j=1}^{n} \lambda_j f w_j = f \text{ for } f \in F$$

In here, W is a synthesis operator and Λ' is an analysis operator. But,in general, it is not going to happen because you will enter with linear functionals and a basis of your choices and you will have to refer that you don't have perfect reconstruction. How to fix this?
Example

$$F = span\{1, t\} =: \Pi_1 \quad (i.e. W = \{1, t\})$$

$$\Lambda = \{\delta_0, \delta_1\}$$

If $f(t) = a + bt$, then $f \neq W\Lambda' f = a + (a + b)t$
Remedy

(1) Stick with Λ, change W to be $\{1 - t, t\}$

(2) Stick with W, change Λ to be $\{\delta_0, \delta_1 - \delta_0\}$

(3) Use magic (manipulate matrices)

$$f = W(\Lambda' W)^{-1}\Lambda' f \quad \forall f \in \Pi_1$$

Theorem If $W\Lambda'$ is invertible, then

$$P := W(\Lambda' W)^{-1}\Lambda' = Id \text{ on} F$$

There are two different ways to understand this.
One way to understand this is split $W(\Lambda'W)^{-1}\Lambda'$ into W and $(\Lambda'W)^{-1}\Lambda'$ and consider $(\Lambda'W)^{-1}\Lambda'$ as the right decomposition of the basis W (i.e., the basis W is important here) . The other way to understand this is split $W(\Lambda'W)^{-1}\Lambda'$ into $W(\Lambda'W)^{-1}$ and Λ' and consider $W(\Lambda'W)^{-1}$ as the right basis of the linear functionals Λ' (i.e., the linear functional Λ' is important here).
proof
Let $f \in F$

$$g := W(\Lambda'W)^{-1}\Lambda f$$

It is enough to show $\Lambda' f = \Lambda' g$.

$$\Lambda' g = \Lambda' W(\Lambda' W)^{-1}\Lambda f = Id\Lambda' f = \Lambda' f$$

$\square$

Properties

(1) $Pf = f \quad \forall f \in \Pi_1$ (projector)

(2) $||Pf||_{\infty,[a,b]} \leq C||f||_{\infty,[a,b]}$

Let's P extend to a map on X. The requirement for the extension is

$$||Pf||_{\infty,[a,b]} \leq C||f||_{\infty,[a,b]}, \quad \forall f \in X$$

$$dist_{\infty,[a,b]}(f, F) := min\{||f - g||_{\infty,[a,b]} | g \in F\}$$

Theorem If $P : C[a, b] \to F$ is a bounded linear projector, then for all $f \in X$

$$||f - Pf||_{\infty,[a,b]} \leq (1 + C)dist_{\infty,[a,b]}(f, F)$$

where C is the bound of the projector.
Proof Let $f \in X, \quad g \in F$

$$\begin{aligned}
||f - Pf||_{\infty,[a,b]} &= ||(f-g) + (g - Pf)||_{\infty,[a,b]} \\
&\leq ||f-g||_{\infty,[a,b]} + ||g - Pf||_{\infty,[a,b]} \\
&= ||f-g||_{\infty,[a,b]} + ||P(g-f)||_{\infty,[a,b]} \\
&\leq (1+C)||f-g||_{\infty,[a,b]}
\end{aligned}$$

□

We would like to know about B-splines now.
Definition: A knot is a either of finite or infinite sequence of points in the real line;

$$\cdots t_0 \leq t_1 \leq t_2 \cdots$$

A order is a local degree of polynomials in the splines. We are going to build a different synthesis system or different basis depending on the order. A typical element in the system would be B-splines.
The ith B-spline order 1 for the knot sequence is defined by

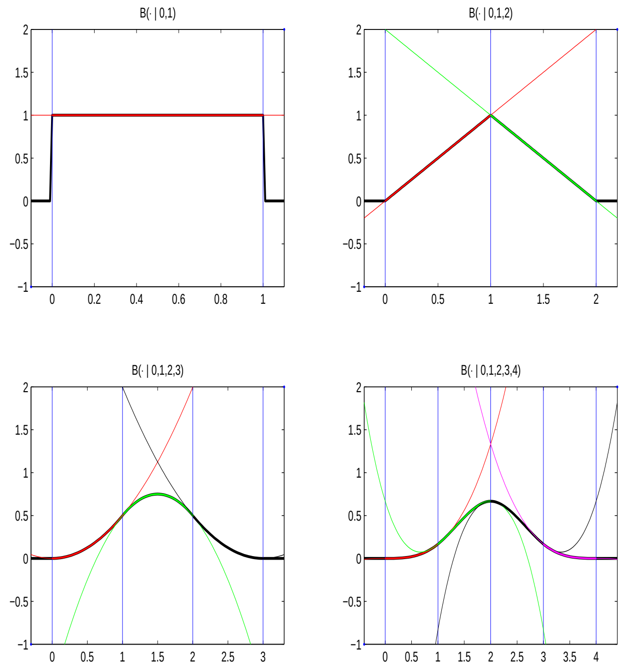
$$B_{i,1}(t) := \begin{cases} 1, & t_i \leq t < t_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

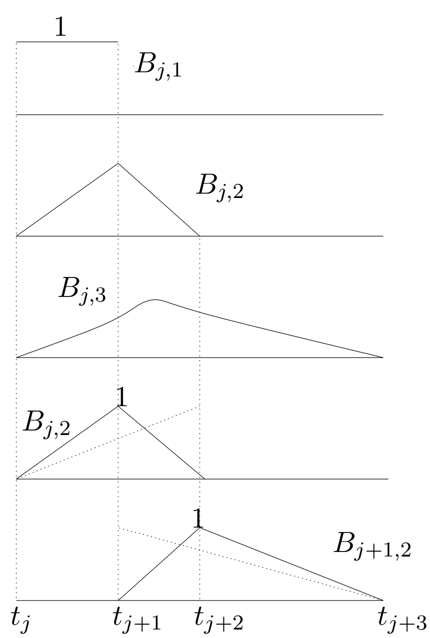
And, the ith B-spline order $k > 1$ for the knot sequence is defined by the following recurrence relation:

$$\begin{aligned}
B_{i,1}(t) &:= \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \left(1 - \frac{t - t_{i+1}}{t_{i+k} - t_{i+1}}\right) B_{i+1,k-1}(t) \\
&= w_{i,k}(t) B_{i,k-1}(t) + \left(1 - w_{i+1,k}(t)\right) B_{i+1,k-1}(t)
\end{aligned}$$

where

$$w_{i,k}(t) := \frac{t - t_i}{t_{i+k-1} - t_i} \text{ (a linear polynomial)}$$


B(· | 0,1)
B(· | 0,1,2)
B(· | 0,1,2,3)
B(· | 0,1,2,3,4)


1
$B_{j,1}$
$B_{j,2}$
$B_{j,3}$
1
$B_{j,2}$
1
$B_{j+1,2}$
t_j
t_{j+1}
t_{j+2}
t_{j+3}

Note: Each higher order B-spline is smoother that the individual lower order B-splines that contributed to the recurrence equation.

$$\Pi_n := \{f | f^{(n+1)} = 0\}$$

Properties of B-splines

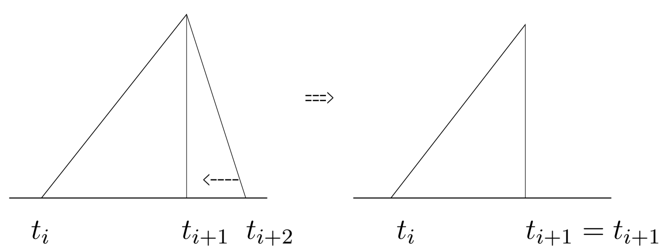
(1) $suppB_{i,k} = (t_i, t_{i+k})$ The end points are excluded since $B_{i,k}$ is vanished at the end points.

(2)$B_{i,k} > 0, \quad \forall t \in (t_i, t_{i+k})$
since $w_{i,k}$ and $1 - w_{i+1,k}$ are positive in the $suppB_{i,k-1}$ and $suppB_{i+1,k-1}$, respectively.

(3) $B_{i,k}|_{(t_i,t_{i+1})} \in \Pi_{k-1}$

(4) If $t_{i-1} < t_i < t_{i+1}$, then $B_{i,k}$ is k-2 times differentiable at t_i

We lose smoothness because of the multiplicity. See the following example



The B-spline order 2 whose knots are t_{i-1}, t_{i-1}, and t_{i-1} is not continuous in the right figure.
Why do we need multiplicity? First of all, it gives us little bit more generality and second the generality is very useful if we introduce splines over finite interval. We will see that later.

Now, we want to use B-splines in order to synthesize other functions. That is, B-splines are used to approximate functions. The questions are how to assign good coefficients for a given function and how to manipulate the sum. In order to approximate a function, we are fixing the order k and the knot sequence.
Remark: Directly from the recurrence relation,

$$\sum_j a_j B_{j,k} = \sum_j (a_j w_{j,k} + a_{j-1}(1 - w_{j,k})) B_{j,k-1}$$

By using the recurrence relation, we can push down our representation from splines of high order to splines of lower order, but the coefficients become more complicated because they are multiplied by linear polynomials.

Theorem (Marsden's identity) For any $\tau, t \in \mathbb{R}$

$$\sum_j \psi_{j,k}(\tau) B_{j,k}(t) = (t-\tau)^{k-1}$$

with

$$\psi_{j,k}(\tau) := (t_{j+1} - \tau) \cdots (t_{j+k-1} - \tau),$$

Proof

$$a_j := \psi_{j,k}(\tau)$$

Then,

$$a_j = a_j w_{j,k}(t) + a_{j-1}(1 - w_{j,k}(t))$$

$$= (t_{i+k-1} - \tau) w_{j,k}(t) + (t_i - \tau)(1 - w_{j,k}(t)) \psi_{j,k-1}(\tau) = (t-\tau)\psi_{j,k-1}(\tau)$$

$$\therefore \quad \sum_j \psi_{j,k}(\tau) B_{j,k}(t) = (t-\tau) \sum_j \psi_{j,k-1}(\tau) B_{j,k-1}(t) = \cdots = (t-\tau)^{k-1}$$

by induction $\square$

In here, the final outcome is a polynomial. This means we can span a polynomial by a linear combinations of B-splines which are piecewise polynomials. It is not hard to see that we can get a polynomials of the form $(t-\tau)^i$, $i \leq k-1$ for any τ. Directly from the above equations, we get

$$\sum_j \frac{\psi_{j,k}(\tau)}{(k-1)!} B_{j,k} = \frac{(t-\tau)^{k-1}}{(k-1)!}$$

Let's differentiate this equation i times with respect to τ. Then

$$\sum_j \frac{(-D)^i \psi_{j,k}(\tau)}{(k-1)!} B_{j,k} = \frac{(t-\tau)^{k-i-1}}{(k-i-1)!}$$

And, for any polynomial p with degree $\leq k-1$, we have

$$p(t) = \sum_{i=0}^{k-1} p^{(k-i-1)}(\tau) \frac{(t-\tau)^{k-i-1}}{(k-i-1)!} \quad \text{by the Taylor's Theorem}$$

Hence, we found a way to express any given polynomial up to degree $k-1$ as a linear combinations of B-splines and it shows the way to expand any polynomial as a linear combinations of B-splines. It would be a major force in terms of understanding how to approximate functions with B-splines.

We are trying to understand how to form linear combinations of B-spines, i.e., what kind functions we are going to get when we assign a certain coefficients or conversely, if we want to obtain a certain function as a linear combination, how to choose the coefficients.
The set of all polynomial functions of degree $< k$ is denoted by

$$\Pi_{<k} \quad \text{with } dim(\Pi_{<k}) = k$$

Then, for any $p \in \Pi_{<k}$,

$$p = \sum_j \lambda_{j,k} p B_{j,k}$$

where

$$\lambda_{j,k} p := \sum_{i=1}^{k} \frac{(-D)^{i-1}\psi_{j,k}(\tau)}{(k-1)!} D^{k-i} p(\tau)$$

Now, for any f such that $f^{(k-1)}$ is bounded,

$$\lambda_{j,k} f := \sum_{i=1}^{k} \frac{(-D)^{i-1}\psi_{j,k}(\tau)}{(k-1)!} D^{k-i} f(\tau)$$

Theorem If $\tau \in [t_i, t_{i+k})$, then

$$\lambda_{j,k} B_{j',k} = \begin{cases} 1, & \text{if } j' = j, \\ 0, & \text{otherwise.} \end{cases}$$

Proof: Assume that $\tau \in [t_l, t_{l+1}) \subset [t_i, t_{i+k})$.

$$B_{j',k}|_{[t_l,t_{l+1})} = 0 \text{ if } j' < l-k+1 \text{ if } j' > l$$

$$\therefore \quad \lambda_{j,k} B_{j',k} = 0 \text{ if } j' < l-k+1 \text{ if } j' > l$$

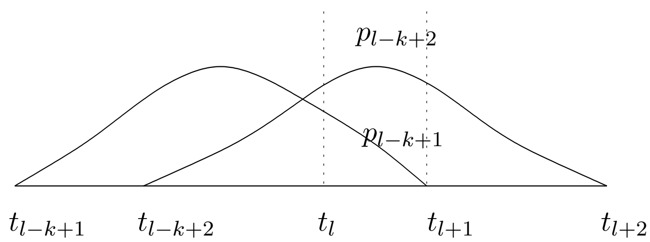
Now, for each of the remaining j', let $p_{j'}$ be the polynomial which agrees with $B_{j',k}$ on $[t_l, t_{l+1})$. Then

$$\lambda_{j,k} B_{j',k} \quad = \quad \sum_{i=1}^{k} \frac{(-D)^{i-1}\psi_{j,k}(\tau)}{(k-1)!} D^{k-i} B_{j',k}(\tau)$$

$$
\begin{aligned}
&= \sum_{i=1}^{k} \frac{(-D)^{i-1}\psi_{j,k}(\tau)}{(k-1)!} D^{k-i} p_{j'}(\tau) \ (\because \quad \tau \in [t_l, t_{l+1})) \\
&= \lambda_{j,k} p_{j'}
\end{aligned}
$$

$$
\therefore \ p_{j'} = \sum_j \lambda_{j,k} p_{j'} B_{j,k} = \sum_{j=l-k+1}^{l} \lambda_{j,k} p_{j'} B_{j,k}
$$

since $\lambda_{j,k} p_{j'} = \lambda_{j,k} B_{j',k} = 0$ if $j' < l-k+1$ if $j' > l$



On the other hand, $\forall p \in \Pi_{<k,[t_l,t_{l+1})}$,

$$
p = \sum_j \lambda_{j,k} p B_{j,k} = \sum_{j=l-k+1}^{l} \lambda_{j,k} p \, p_j \text{ on } [t_l, t_{l+1})
$$

by the definition of p_j That is $\{p_{l-k+1}, \cdots, p_l\}$ is a basis of $\Pi_{<k,[t_l,t_{l+1})}$. That is, $p_{l-k+1}, \cdots, p_l$ are linearly independent. Hence

$$
\begin{aligned}
& p_{j'} = \sum_{j=l-k+1}^{l} \lambda_{j,k} p_{j'} B_{j,k} \\
\Longrightarrow \quad & p_{j'} = \sum_{j=l-k+1}^{l} \lambda_{j,k} p_{j'} p_j \text{ on } [t_l, t_{l+1}) \\
\Longrightarrow \quad & \lambda_{j,k} p_{j'} = \begin{cases} 1, & \text{if } j' = j, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}
$$

$$
\therefore \ \lambda_{j,k} B_{j',k} = \begin{cases} 1, & \text{if } j' = j, \\ 0, & \text{otherwise.} \end{cases}
$$

□

So,

$$
\sum_j \left(\lambda_{j,k} B_{j',k}\right) B_{j,k} = B_{j',k}
$$

Conclusion: If $f = \sum_j a_j B_{j,k}$, then

$$a_j = \lambda_{j,k} f$$

So if we know the output, we can find the coefficients. We know to do the decomposition in a way that would give us the right coefficients for the synthesis.
Finally, we want to insert the notion of a projector. This is the hardest part of the entire building up of the spline theory, finding the inversion, finding the right decomposition for the synthesis based on the B-splines.
Projector
Input: a smoothie function f

$$f \approx Pf := \sum_j \lambda_{j,k} f B_{j,k}$$

$$P^2 f = Pf$$

A spline of order k with knot sequence $\underline{t}$ is a linear combination of the B-splines.

$$\$_{k,\underline{t}} := span_j\{B_{j,k}\}$$

We are trying to understand this space as a source of approximation for more general functions.

$$\lambda_{j,k} f := \sum_{\nu=1}^{k} \frac{(-D)^{\nu-1}\psi_{j,k}(\tau_j)}{(k-1)!} D^{k-\nu} f(\tau_j)$$

Note: $\frac{(-D)^{\nu-1}\psi_{j,k}(\tau_j)}{(k-1)!}$ are numbers that depend only on the know sequence $\underline{t}$ and τ_j and the information about the function f is recored $f(\tau_j)$. The information is very local information that consists of function values, derivative values and some fixed point τ_j. What do we know about the linear functional $\lambda_{j,k}$? There are three facts:
(1) If p is a polynomial of degree$< k$, then

$$\sum_j \lambda_{j,k} p B_{j,k} = p$$

(2)

$$\lambda_{j,k} B_{j',k} = \begin{cases} 1, & \text{if } j' = j, \\ 0, & \text{otherwise.} \end{cases}$$

(3) If $P : f \mapsto \sum_j \lambda_{j,k} f B_{j,k}$, then

$$Pf = f \quad \forall f \in \$_{k,\underline{t}}$$

Question: What should be a good choice for τ_j?

$$f = 1 \Longrightarrow \lambda_{j,k} f = 1$$

$$f \in \Pi_1 \Longrightarrow \lambda_{j,k} f = f(\tau_j) + \frac{(-D)^{k-2}\psi_{j,k}(\tau_j)}{(k-1)!} f'(\tau_j)$$

In here,

$$\frac{(-D)^{k-2}\psi_{j,k}(\tau_j)}{(k-1)!} = 0 \text{ if } \tau_j = \frac{t_{j+1} + \cdots + t_{j+k-1}}{k-1} =: t_j^*$$

$$\therefore \quad f = \sum_j f(t_j^*) B_{j,k}$$

For a given f, we want to find suitable coefficients and to analyze the error or for a given coefficients, we want to understand how the coefficients are encoding properties of the curve. These are two related issues, but they are not entirely same. From now on, we will assume $f, f', \cdots, f^{(m)}$ are continuous everywhere.

$$Qf := \sum_j f(t_{j,k}^*) B_{j,k} \text{ where } t_{j,k}^* = \frac{t_{j+1} + \cdots + t_{j+k-1}}{k-1}$$

Then $Qf = f, \quad \forall f \in \Pi_1$

$$\|f\|_{m,[a,b]} := sup_{t \in [a,b]} \left| f^{(m)}(t) \right|$$

Fact: If $p \in \Pi_{m-1}$, then

$$\|f - p\|_{0,[a,b]} \leq C_m (b-a)^m \|f\|_{m,[a,b]}$$

We are trying to approximate f on $[a, b]$ by p, and our ability to do that is related the quality of the approximation, namely the size of the error which is related to the size of the interval. Essentially, it measures how bad or good the function f is. Here is the error analysis for m=2. We are going to approximate f as good as we can do that in the interval $[t_{j-k+1}, t_{j+k}]$ by a linear polynomial. Note that the B-splines for Qf is compactly supported and by the fact, there is a polynomial p_t for $m = 2$. Now,

$$\begin{aligned} |f(t) - Qf(t)| &= |f(t) - p_t(t) - Q(f - P_t)(t)| \\ &\quad (\because \ p_t \in \Pi_1, \text{ and } Qp_t = p_t) \\ &\leq |f(t) - p_t(t)| + |Q(f - p_t)(t)| \end{aligned}$$

And,

$$\begin{aligned} |Q(f - p_t)(t)| &\leq \sum_j \left|(f - p_t)(t^*_{j,k})B_{j,k}\right| \\ &\leq \|f - p_t\|_{0,[t_{j-k+1},t_{j+k}]} \sum_j B_{j,k} \\ &\leq \|f - p_t\|_{0,[t_{j-k+1},t_{j+k}]} \end{aligned}$$

$$\therefore \quad |f(t) - Qf(t)| \leq 2C_2\|f - p_t\|_{0,[t_{j-k+1},t_{j+k}]}(t_{j-k+1} - t_{j+k})^2$$

$$\therefore \quad |f(t) - Qf(t)| \leq C\|f\|_{2,[t_{j-k+1},t_{j+k}]}(t_{j-k+1} - t_{j+k})^2$$

You are going to achieve nothing by changing the k. Your ability to suppress error is by making the knot sequence denser and dancer. More than that, the error reflects local smoothness of f.
Why $m = 2$? There are two reasons. One is that f is not that smooth and the other is the scheme is not good enough. The scheme only reproduces linear polynomials.
Here is the 2nd scheme.

$$Pf := \sum_j \lambda_{j,k}(f)B_{j,k}$$

Then $Pf = f, \quad \forall f \in \Pi_{k-1}$ and similarly, we can get

$$|f(t) - Pf(t)| \leq C\|f\|_{k,[t_{j-k+1},t_{j+k}]}(t_{j+k} - t_{j-k+1})^k$$

The only reason not to use the scheme is not because of the error estimate, but because the computing the coefficients is out of reach. It involves the derivatives of f and usually we don't have that information. But you will never ask for an error estimates better than this. You will ask for an error estimate compatible to this.

Properties

1 $B_{j,k} \geq 0$ and $suppB_{j,k} = (t_j, t_{j+k})$

2 $\forall s \in \$_{k,\underline{t}}, \forall j$,

$$s|_{(t_j,t_{j+k})} \in \Pi_{<k}$$

3 Smoothness:

$t_{j-1} < t_j = \cdots = t_{j+m-1} < t_{j+m}$ (m multiplicity), then every $s \in \$_{k,\underline{t}}$ is $(k-m-1)$ times differentiable at t_j

4 Truncated power of order k:

$$T_j(t) := \begin{cases} 0, & t \leq t_j, \\ (t-t_j)^{k-1}, & \text{otherwise.} \end{cases}$$

We can build the spline space as a linear combination of truncated powers. This is the one of the ways to define splines. If we can spanned the truncated powers, we essentially proved that every function with the piecewise polynomial structure is going to be spanned by the B-splines. So we need to show that the B-splines can span the truncated powers.

$$T_i(t) = \sum_{j=i}^{\infty} \psi_{j,k}(\tau) B_{j,k}$$

where τ is one of the knots.

$$(\cdot - t_j)_+^{k-1} := T_j$$

Theorem Given $\underline{t}$ and k, consider a function s with the following properties:

(1) $s|_{(t_j,t_{j+k})} \in \Pi_{<k}$

(2) s is $k-m-1$ differentiable at t_j where $t_{j-1} < t_j = \cdots = t_{j+m-1} < t_{j+m}$

then

$$s \in \$_{k,\underline{t}}$$

Subdivision

$$\underline{t} := (t_0, \cdots, t_N)$$

Let's define a new knot sequence.

$$\bar{t} := \underline{t} \cup \{x\}$$

where the location x may be overlapped with an old knot or may be strictly between two old knots. Then

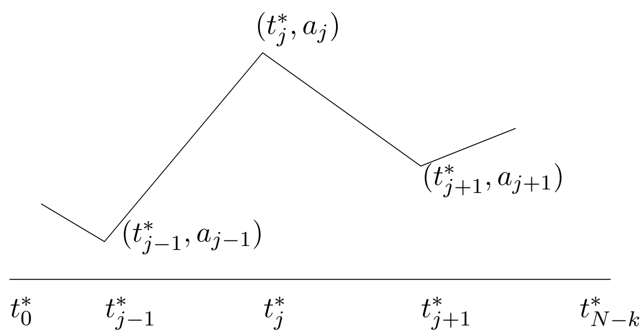
$$\$_{k,\underline{t}} \subset \$_{k,\bar{t}}$$

Now, let

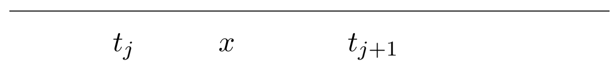
$$s = \sum_{j=0}^{N-k} a_j B_{j,k,\underline{t}} = \sum_{j=0}^{N-k+1} \bar{a}_j B_{j,k,\bar{t}}$$

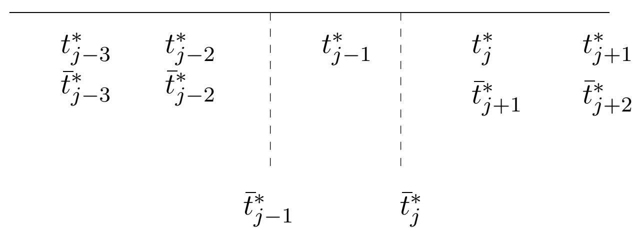
We want to know the relation between a_j and $\bar{a}_j$.
We will use the dual knot sequence $\{t_j^*\}$ in order build what we call the control line or the control polygon of the spline by building a piecewise linear picture where at the dual knot we put a point a_j (control point)

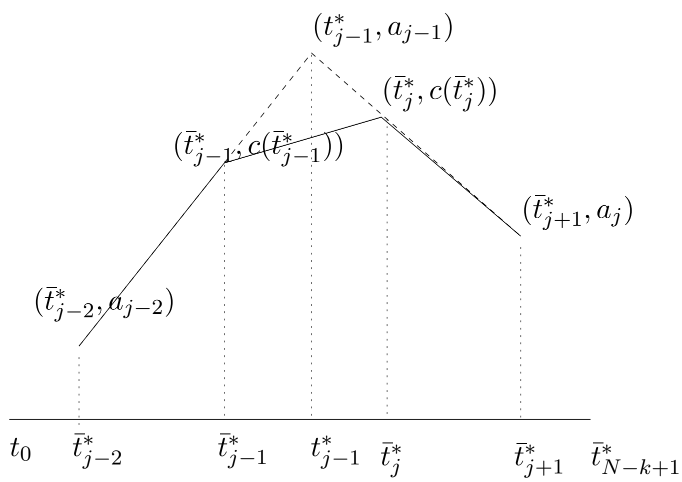
$$t_j^* := \frac{t_{j+1} + \cdots + t_{j+k-1}}{k-1}$$



We want to understand first of all between the control line from the coefficients of $\sum_j a_j B_{j,k,\underline{t}}$ and the control line from the coefficients of $\sum_j \bar{a}_j B_{j,k,\bar{t}}$
Example k=3.
Let's assume that we insert the knot x in between t_j and t_{j+1}. Then in the control polygon, we lose t_j^* and we gain $\bar{t_{j-1}}^*$ and $\bar{t_j}^*$ since k=3.







That is, the control polygon of the spline function s is a picewise linear function that connects the points $(t^*_j, a_j), \quad j = 0, \cdots, N-k$.
$c(\bar{t}^*_j)$ is obtained by the linear interpolation between the two adjacent control points in the old control sequence.
So if you can figure out where your new knots are, then you know all the new coefficients for free. And we can repeat the process. The new control sequence is representing the same spline s only with respect to another knot sequence. If we do the refinement a few hundred times, then we will be able to get very nice approximation in terms of piecewise linear curve to the spline.

Example K=4, i.e., cubic spline

If we insert a knot x between t_j and t_{j+1}, then the new coefficients will be

$$\bar{a}_i = \begin{cases} a_i, & i \leq j-3, \\ a_{i-1}, & i \geq j+1 \end{cases}$$

And, $\bar{a}_{j-2}$, $\bar{a}_{j-1}$, and $\bar{a}_j$ are obtained by the linear interpolation between the two adjacent control points in the old control sequence.

Conclusions.
Spline curves are shape preserving. Let

$$f = \sum_j a_j B_{j,k}$$

(1) If $a_j \geq 0$ for all j, then $f \geq 0$ everywhere.

(2) If $a_j \leq a_{j+1} \leq \cdots$ for all j, then f is nondecreasing.

(3) If the control polygon is convex (or concave), then f is so.

Fact If $\underline{t} \subseteq \mathbb{Z}$, then

$$B_{j,k,\underline{t}} = B_k(\cdot - j)$$

where $B_k = B_1 * \cdots * B_1$ (k times).
Now, we want to elaborate little bit more about the idea of knot insertion in this context.

$$\underline{t} = \mathbb{Z}$$

$$\bar{\underline{t}} = \mathbb{Z}/2$$

(i.e., inserting globally in order to preserve the equidistance). Then

$$B_{j,k,\bar{\underline{t}}} = B_k(2 \cdot -j)$$

On the other hand, because of the refinability of B_k,

$$f = \sum_j a_j B_k(\cdot - j) = \sum_j \bar{a_j} B_k(\cdot/2 - j)$$

where

$$\bar{a} = \bar{h}_0 * (a \uparrow)$$

This is the subdivision algorithm. Subdivision is a global attempt to take f which is represented only in terms of its coefficients. In other words, you don't know the information of ϕ, but you know the mask h of ϕ where

$\sum_j h(j) = 2$

Approximation by splines

$$f \approx \sum_j \lambda_{j,k}(f) B_{j,k}$$

These dual functionals requires not only function values but also derivatives values. If they are given, this is the ideal approximation. If they are not given, there are two major ideas:
(1) Interpolation
(2) Least squares
Let's assume $\underline{\tau} := (\tau_i)_{i=1}^N$ is given. We want to recover f based on the

$$\Lambda' f = \begin{bmatrix} f(\tau_1) \\ \vdots \\ f(\tau_N) \end{bmatrix}$$

The question is if we have N values, how many knots we are going to use. In interpolation, it is going to be compatible to N and in least squares, it is going to be significantly smaller than N.
Interpolation
Suppose the knot sequence $\underline{t} := t_0 \leq \cdots \leq t_N$ and the order k are given. Then there are $N + 1 - k$ Select interpolation points

$$\tau_1 < \cdots < \tau_n \quad \text{where n = N + 1 - k}$$

Given, $(f(\tau_i))_{i=1}^n$, we want to find $g \in \$_{k,\underline{t}}$ such that

$$g(\tau_i) = f(\tau_i) \quad i = 1, \cdots, n$$

Put $g := \sum_{j=0}^{N-k+1} a_j B_{j,k,\underline{t}}$. We will find the coefficients by solving the following linear system:

$$\begin{bmatrix} B_{0,k}(\tau_1) & \cdots & B_{n,k}(\tau_1) \\ \vdots & B_{j,k}(\tau_i) & \vdots \\ B_{0,k}(\tau_n) & \cdots & B_{n,k}(\tau_n) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(\tau_1) \\ \vdots \\ f(\tau_n) \end{bmatrix}$$

(The above nxn matrix is called the collocation matrix.)

Theorem (Whitney-Schoenberg) The collocation matrix in spline interpolation, based on the B-spline representation, is invertible if and only if its diagonal is positive.

Now, the question is where to place the interpolation points. If one of them is given, then we will choose the other one relative to the ones that given. If none of them is given, then we will put them uniformly space.

(1) If $\underline{t}$ is given, choose $\underline{\tau}$ as the dual knots; $\tau_i = t_i^*$

(2) If $\underline{\tau}$ is given, then we append $\underline{\tau}^*$ with τ_1 and τ_k, each with multiplicity k.

Example k=3, $\underline{\tau} = 0 < .5 < 1.5 < 2 < 3 < 4 < 4.5 < 5$

Then a good choice is

$$0, 0, 0, 1, 1 + 3/4, 2 + 1/2, 3 + 1/2, 4 + 1/4, 5, 5, 5$$

$$Pf := g$$

(1) P is a linear map

(2) P is a projector; If $f \in \$_{k,\underline{t}}$, then $Pf = f$

Question: How good Pf is as an approximation to f?

$$||f - Pf||_{\infty,I} := \sup_{t \in I} |f(t) - Pf(t)|$$

$$||P|| := \text{ the smallest number such that } ||Ph||_{\infty,I} \leq ||P|| \; ||h||_{\infty,I} \quad \forall h$$

$$\begin{aligned} ||f - Pf||_{\infty,I} &= ||f - g + g - Pf||_{\infty,I} \quad g \in \$_{k,\underline{t}} \\ &\leq ||f - g||_{\infty,I} + ||P(f - g)||_{\infty,I} \\ &\leq ||f - g||_{\infty,I} + ||P|| \; ||f - g||_{\infty,I} \\ &= (1 + ||P||) ||f - g||_{\infty,I} \end{aligned}$$

Least squares

Let $\Lambda = [\delta_{x_j}]_{j=0}^n$. Then the data which we have is

$$\Lambda' f := \begin{bmatrix} f(\tau_1) \\ \vdots \\ f(\tau_n) \end{bmatrix}$$

The two facts might be valid. One of them is that n is so large to do interpolation. Another one is that our measurement is not accurate if there is noisy in the measurement which is not negligible. If there is noisy in the measurement, we do not want to interpolate.

Given some $f : I \longrightarrow \mathbb{R}$ and $\Lambda' f$, we want to find $g \in G := \{g | g : I \longrightarrow \mathbb{R}\}$ so that

$$||\Lambda'(f - g)||_{l_2}$$

is minimized where $dimG << n$.

$$||\Lambda'(f - g)||_{l_2} := \left[\sum_{i=1}^{n}(f(\tau_i) - g(\tau_i))^2\right]^{1/2}$$

Let $W = \{g_1, \cdots, g_m\}$ be a basis for G. Then we want to find $a_1, \cdots, a_m$ such that

$$\begin{bmatrix} g_1(\tau_1) & \cdots & g_m(\tau_1) \\ \vdots & g_j(\tau_i) & \vdots \\ g_1(\tau_n) & \cdots & g_m(\tau_n) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} f(\tau_1) \\ \vdots \\ f(\tau_n) \end{bmatrix}$$

By solving this linear system, we will get an approximation $g = \sum_{j=1}^{m} a_m g_m$.

$$A := \Lambda' W = \begin{bmatrix} g_1(\tau_1) & \cdots & g_m(\tau_1) \\ \vdots & g_j(\tau_i) & \vdots \\ g_1(\tau_n) & \cdots & g_m(\tau_n) \end{bmatrix}, \quad a := \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}, \quad b := \begin{bmatrix} f(\tau_1) \\ \vdots \\ f(\tau_n) \end{bmatrix}$$

Then we have

$$Aa = b \Longrightarrow A'Aa = A'b \text{ (normal equation)}$$

$$\Longrightarrow a = (A'A)^{-1}A'b$$

Recall $L_2([a, b])$.

$$\langle f, g \rangle := \int_a^b f(t)\overline{g(t)}dt$$

$$||f||_2 := \left(\int_a^b |f(t)|^2 dt\right)^{1/2}$$

Let F be a big space and G be a small space with a basis $W = \{g_1, \cdots, g_m\}$.
Fact I: If you found $g^* \in G$ such that $(f - g^*) \perp g \quad \forall g \in G$, then

$$||f - g^*||_2 \leq ||f - g||_2, \quad \forall g \in G \backslash \{g^*\}$$

$(f \perp g \iff \langle f, g \rangle = 0\ )$
That is , g^* is the unique minimizer.
Proof If $f_1, f_2 \in F$ and $f_1 \perp f_2$, then

$$||f_1 + f_2||_2^2 = ||f_1||_2^2 + ||f_2||_2^2$$

Suppose $g \in G \backslash \{g^*\}$, then

$$\begin{aligned} ||f - g||_2^2 &= ||(f - g^*) + (g^* - g)||_2^2 \\ &= ||f - g^*||_2^2 + ||g^* - g||_2^2 \quad (\because\ (f - g^*) \perp (g^* - g)) \\ &> ||f - g^*||_2^2 \quad (\because\ g^* \neq g) \end{aligned}$$

$\square$

Fact II: How to find such g^*? Put

$$g^* = Wa := \sum_{i=1}^{m} a_i g_i$$

In order to find g^*, we need

$$\langle f - g^*, g_j \rangle = 0, \quad \forall g_j \in W$$

$$\iff \langle g^*, g_j \rangle = \langle f, g_j \rangle, \quad \forall g_j \in W$$

That is, find a such that

$$\begin{bmatrix} \langle g_1, g_1 \rangle & \cdots & \langle g_m, g_1 \rangle \\ \vdots & \langle g_j, g_i \rangle & \vdots \\ \langle g_1, g_m \rangle & \cdots & \langle g_m, g_m \rangle \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \langle f, g_1 \rangle \\ \vdots \\ \langle f, g_m \rangle \end{bmatrix} - - - -(1)$$

Now, suppose we can not compute the continuous inner product $\langle f, g_k \rangle$, then what to do?
Assume that f is given in terms of

$$\tilde{f} := \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_M) \end{bmatrix}$$

$$\langle f, g \rangle' := \sum_{j=1}^{M} f(x_j) g(x_j)$$

$$||f||_{l_2} := \left(\sum_{j=1}^{M} |f(x_j)|^2 \right)^{1/2}$$

With this new inner product, solve the linear system (1) to get g^*. Then $||f - g^*||_{l_2}$ is minimal.
Now, consider G with a basis $W = (B_{j,k})_{j=0}^{N-k}$. Then for a given $f : [a,b] \to \mathbb{R}$, least squares produces a spline $Pf \in G$. It is now known that Pf is best in tersmof minimizing in L_2 error.
Properities of P.

(1) P is linear

(2) P is a projector ($\because$ $||Pg - g||_2 = 0$)

(3) P is bounded in maximal norm sense;

$$||Pf||_{\infty,[a,b]} \leq C||f||_{\infty,[a,b]}$$