Methods in Computational Science – Function approximation (ch.9)

Johan Hoffman

Function approximation in Banach spaces

$f_N \approx f$, based either on an explicit function $f : X \to Y$, an implicit model $R(f) = 0$, or a sampled data set $\{X_i, Y_i\}_{i=1}^{M}$ with $f(X_i) \approx Y_i$. If f is a function in a Banach space V and $V_N \subset V$ is a finite dimensional subspace spanned by the basis $\{\phi_j\}_{j=1}^{N} \subset V$, then we can choose an approximation of the following form,

$$f(x) \approx f_N(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x),$$

with $\alpha = (\alpha_1, ..., \alpha_N)^T$ the coordinates of f_N in the basis $\{\phi_j\}_{j=1}^{N}$. Once we have selected a suitable basis $\{\phi_j\}_{j=1}^{N}$ the approximation problem comes down to the determination of $\alpha \in R^N$.

Function approximation in Banach spaces

$$f(x) \approx f_N(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x),$$

with $\alpha = (\alpha_1, ..., \alpha_N)^T$ the coordinates of f_N in the basis $\{\phi_j\}_{j=1}^N$. Once we have selected a suitable basis $\{\phi_j\}_{j=1}^N$ the approximation problem comes down to the determination of $\alpha \in R^N$.

Since the Banach space V is a normed vector space we can seek the function f_N which minimizes the approximation error $f - f_N$ with respect to the norm in V,

$$\min_{f_N \in V_N} \|f - f_N\|_V.$$

Polynomial vector spaces

Example 9.1. $\mathcal{P}^q(I)$ is the vector space of real polynomial functions of at most order q on an interval $I \subset R$, with the vector space operations of pointwise addition and scalar multiplication

$$(p+r)(x) = p(x) + r(x), \quad (\alpha p)(x) = \alpha p(x),$$

for $p, r \in \mathcal{P}^q(I)$ and $\alpha \in R$. The dimension of $\mathcal{P}^q(I)$ is $q+1$, and every $p \in \mathcal{P}^q(I)$ can be expressed in the *monomial* basis $\{x^i\}_{i=0}^q$ as

$$p(x) = \sum_{i=0}^{q} a_i x^i, \quad a_i \in R.$$

Polynomial vector spaces

An alternative basis for $\mathcal{P}^q(I)$ is $\{(x-c)^i\}_{i=0}^q$ which gives the finite *power series*,

$$p(x) = \sum_{i=0}^{q} a_i (x-c)^i = a_0 + a_1(x-c) + ... + a_q(x-c)^q,$$

with the center $c \in I$. By Taylor's theorem and the mean value theorem, we can express any function $f \in C^{k+1}(I)$ as the kth order *Taylor polynomial* with the $k+1$ order Lagrange error term, for ξ a real number in between x and c, a power series of the form

$$f(x) = f(c) + \sum_{n=1}^{k} \frac{1}{n!} \frac{d^n f}{dx^n}(c)(x-c)^n + \frac{1}{(n+1)!} \frac{d^{k+1} f}{dx^{k+1}}(\xi)(x-c)^{k+1}.$$

Interpolation

The *interpolant* $\pi_N f \in V_N$ is an approximation of $f \in V$ constructed to be exact in a set of *nodes* $\{x_i\}_{i=1}^N$ in the domain X. Assuming that pointwise evaluation $f(x_i)$ is well defined for all $x_i \in X$, we can construct the interpolant from

$$f(x_i) = \pi_N f(x_i) = \sum_{j=1}^{N} \alpha_j \phi_j(x_i), \quad i = 1, ..., N.$$

$$A\alpha = b,$$

where $a_{ij} = \phi_j(x_i)$ and $b_i = f(x_i)$, from which we can compute the vector of coordinates $\alpha = (\alpha_1, ..., \alpha_N)^T$. If pointwise evaluation of $f(x_i)$ is not well defined for functions in V, as may be the case for discontinuous functions, it can be replaced by a well defined average of the function over a subdomain of X that contains the node x_i, referred to as *quasi-interpolation*.

Interpolation with nodal basis

Of specific interest is the case when $\{\phi_i\}_{i=1}^N$ is a *nodal basis*, defined by the condition that $\phi_j(x_i) = \delta_{ij}$ for all nodes x_i, where δ_{ij} is the Kronecker delta function. For a nodal basis the matrix A becomes an identity matrix, so that $\alpha_j = f(x_j)$ and we can express the interpolant as

$$\pi_N f(x) = \sum_{j=1}^{N} f(x_j)\phi_j(x).$$

The polynomial nodal basis

$$\pi_N f(x) = \sum_{j=1}^{N} f(x_j)\phi_j(x)$$

Let us denote by $\{\lambda_i\}_{i=0}^{q}$ the nodal basis for $\mathcal{P}^q(I)$, defined by

$$\lambda_i(x_j) = \delta_{ij},$$

such that any $p \in \mathcal{P}^q(I)$ can be expressed as

$$p(x) = \sum_{i=0}^{q} p(x_i)\lambda_i(x).$$

Interpolation with Lagrange basis

We refer to the nodal basis $\{\lambda_i\}_{i=0}^q$ as the *Lagrange polynomials*, or the *Lagrange basis*,

$$\lambda_i(x) = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_q)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_q)} = \prod_{i=0,\, i\neq j}^{q} \frac{x - x_j}{x_i - x_j},$$

with $\lambda_0 = 1$ for $q = 0$. For a continuous function $f \in \mathcal{C}(I)$, we define the *polynomial interpolant*

$$\pi_q f(x) = \sum_{i=0}^{q} f(x_i)\lambda_i(x), \quad x \in I,$$

Theorem 9.2. *If $f \in C^{q+1}(I)$ is approximated by the polynomial interpolant $\pi_q f \in \mathcal{P}^q(I)$ with the set of nodes $\{x_i\}_{i=0}^q \subset I$, then the interpolation error can be estimated for each $x \in I$ as*

$$|f(x) - \pi_q f(x)| \leq \left| \frac{(x - x_0)\cdots(x - x_q)}{(q+1)!} \right| \max_{\xi \in I} |D^{q+1} f(\xi)|.$$

Interpolation with Lagrange basis

Proof. Here we only give the proof for $q = 0$ and $q = 1$, starting with the case $q = 0$ for which

$$\pi_0 f(x) = f(x_0).$$

By the mean value theorem,

$$f(x) - \pi_0 f(x) = f(x) - f(x_0) = f'(\xi)(x - x_0),$$

for some $\xi \in [x_0, x]$, which proves the result for $q = 0$, since

$$|f(x) - \pi_0 f(x)| = |f'(\xi)(x - x_0)| \leq |x - x_0| \max_{\xi \in I} |f'(\xi)|.$$

Interpolation with Lagrange basis

For $q = 1$, we have

$$\pi_1 f(x) = f(x_0)\lambda_0(x) + f(x_1)\lambda_1(x),$$

and by Taylor's theorem,

$$f(x_i) = f(x) + f'(x)(x_i - x) + \frac{1}{2} f''(\xi_i)(x_i - x)^2,$$

for $i = 1, 2$, with ξ_i a point between x and x_i. Therefore,

$$f(x) - \pi_1 f(x) = -\frac{1}{2}(f''(\xi_0)(x_0 - x)^2 \lambda_0(x) + f''(\xi_1)(x_1 - x)^2 \lambda_1(x)),$$

since

$$\lambda_0(x) + \lambda_1(x) = 1, \quad (x_0 - x)\lambda_0(x) + (x_1 - x)\lambda_1(x) = 0,$$

from which it follows that

$$|f(x) - \pi_1 f(x)| \leq \frac{1}{2}\left(\frac{(x - x_0)^2(x_1 - x)}{x_1 - x_0} + \frac{(x_1 - x)^2(x - x_0)}{x_1 - x_0}\right) \max_{\xi \in I} |f''(\xi)|,$$

that proves the result for $q = 1$.

Interpolation with Lagrange basis

Theorem 9.3. *If the function $f \in C^2(I)$ is approximated by the linear interpolant $\pi_1 f \in \mathcal{P}^1(I)$ in the interval $I = [x_0, x_1]$, then for each $x \in I$,*

$$|f'(x) - (\pi_1 f)'(x)| \leq \frac{(x - x_0)^2 + (x - x_1)^2}{2(x_1 - x_0)} \max_{\xi \in I} |f''(\xi)|.$$

Proof. If we take the derivative of the interpolant,

$$(\pi_1 f)'(x) = f(x_0)\lambda_0'(x) + f(x_1)\lambda_1'(x),$$

then the result follows from the observation that

$$\lambda_0'(x) + \lambda_1'(x) = 0, \quad \lambda_0'(x)(x_0 - x) + \lambda_1'(x)(x_1 - x) = 1.$$

Piecewise polynomials

We now introduce *piecewise polynomials* defined over a partition of the interval $I = [a, b]$,

$$a = x_0 < x_1 < \cdots < x_{m+1} = b,$$

where we let the *mesh* $\mathcal{T}_h = \{I_i\}$ denote the set of subintervals $I_i = (x_{i-1}, x_i)$ of length $h_i = x_i - x_{i-1}$, with the piecewise constant *mesh function* $h(x) = h_i$, for $x \in I_i$. We define two vector spaces of piecewise polynomials, the discontinuous piecewise polynomials on I,

$$W_h^{(q)} = \{v : v|_{I_i} \in \mathcal{P}^q(I_i),\ i = 1, ..., m+1\},$$

and the continuous piecewise polynomials on I,

$$V_h^{(q)} = \{v \in W_h^{(q)} : v \in \mathcal{C}(I)\}.$$

The basis for $W_h^{(q)}$ is defined directly in terms of the local Lagrange *shape functions* on each subinterval, whereas for $V_h^{(q)}$ we also need to add constraints to enforce global continuity.

Piecewise polynomial basis functions

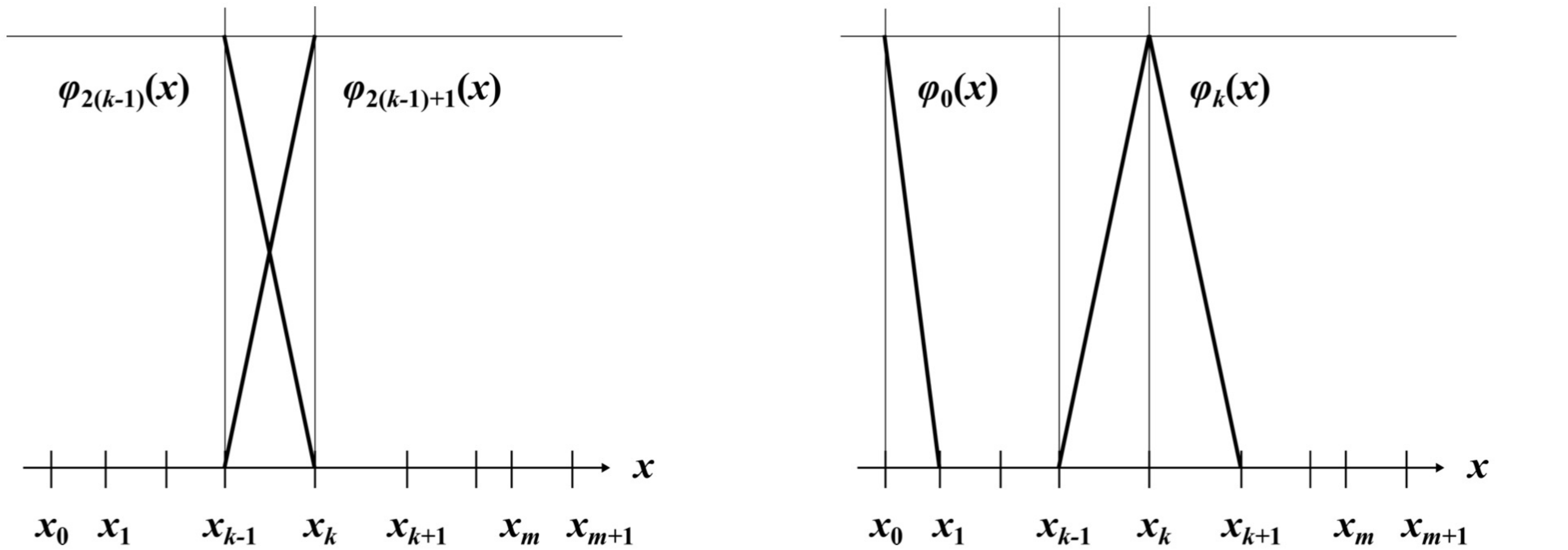


Figure 9.1. *Illustration of a mesh $\mathcal{T}_h = \{I_k\}$ with subintervals $I_k = (x_{k-1}, x_k)$, two global basis functions for $W_h^{(1)}$ (left) and two global basis functions for $V_h^{(1)}$ (right).*

Discontinuous piecewise polynomials

Example 9.4. The local linear Lagrange shape functions for subinterval I_k,

$$\lambda_{k,0}(x) = \frac{x_k - x}{h_k}, \quad \lambda_{k,1}(x) = \frac{x - x_{k-1}}{h_k},$$

define the global basis functions for $W_h^{(1)}$ by

$$\phi_i(x) = \begin{cases} \lambda_{k,j}(x), & x \in I_k, \\ 0, & x \neq I_k, \end{cases}$$

with the global index $i = 2(k-1) + j$, and local index $j = 0, 1$ for each subinterval I_k. Hence, for $x \in I_k$, any $w \in W_h^{(1)}$ can be expressed either in the global basis or the local shape functions

$$w(x) = \sum_{i=1}^{2(m+1)} \alpha_i \phi_i(x) = \sum_{j=0}^{1} \beta_{k,j} \lambda_{k,j}(x),$$

where α_i and $\beta_{k,j}$ are the global and local *degrees of freedom* (dofs), with $\alpha_{2(k-1)+j} = \beta_{k,j}$.

Continuous piecewise polynomials

$$\phi_i(x) = \begin{cases} \lambda_{k,1}(x), & x \in I_k, \\ \lambda_{k+1,0}(x), & x \in I_{k+1}, \\ 0, & x \neq I_k \cup I_{k+1}, \end{cases}$$

with the global index $i = k$ for all internal nodes x_k, $k = 1, ..., m$, and where the basis functions at the endpoints of the interval take the form of hat functions cut in half,

$$\phi_0(x) = \begin{cases} \lambda_{1,0}(x), & x \in I_1, \\ 0, & x \neq I_0, \end{cases} \qquad \phi_{m+1}(x) = \begin{cases} \lambda_{m+1,1}(x), & x \in I_{m+1}, \\ 0, & x \neq I_{m+1}. \end{cases}$$

For $x \in I_k$, a function $v \in V_h^{(1)}$ is expressed in the global basis and the local shape functions as

$$v(x) = \sum_{i=0}^{m+1} \alpha_i \phi_i(x) = \sum_{j=0}^{1} \beta_{k,j} \lambda_{k,j}(x),$$

with $\alpha_k = \beta_{k,1} = \beta_{k+1,0}$ for the internal nodes $k = 1, ..., m$, and for the endpoints $\alpha_0 = \beta_{1,0}$ and $\alpha_{m+1} = \beta_{m+1,1}$, which enforces global continuity.

Evaluation of piecewise polynomials

Algorithm 9.1. f = eval_pwp_function(alpha, x, q).

Input: global degrees of freedom **alpha**, polynomial order **q**, evaluation point **x**.

Output: piecewise polynomial function **f** evaluated in point x.

```
k = find_subinterval(x)
loc2glob = get_local_to_global_map(k)
beta = get_local_dofs(alpha, loc2glob)
lambda = eval_local_shape_functions(k, x, q)
f = scalar_product(beta, lambda)
return f
```

Piecewise polynomial basis functions

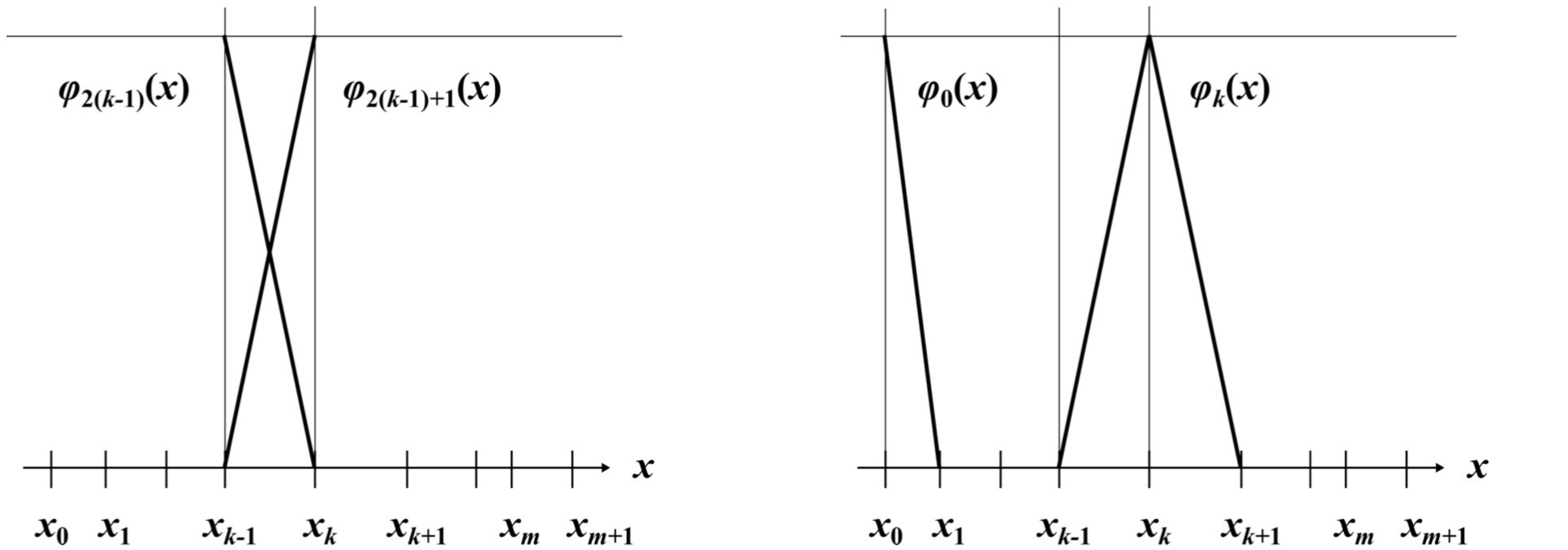


Figure 9.1. *Illustration of a mesh $\mathcal{T}_h = \{I_k\}$ with subintervals $I_k = (x_{k-1}, x_k)$, two global basis functions for $W_h^{(1)}$ (left) and two global basis functions for $V_h^{(1)}$ (right).*

Piecewise polynomial basis functions

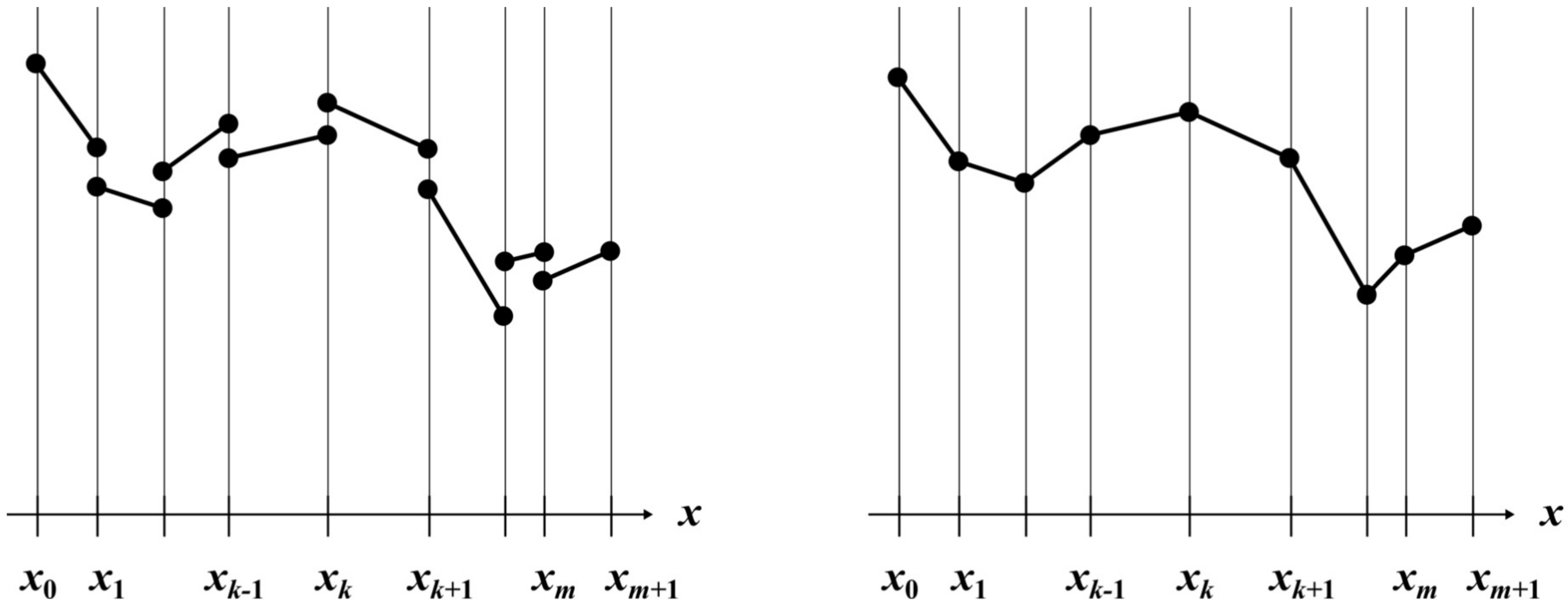


Figure 9.2. *Example of a discontinuous piecewise linear function in* $W_h^{(1)}$ *with* $2(m+1)$ *global dofs (left), and a continuous piecewise linear function in* $V_h^{(1)}$ *with* $m+2$ *global dofs (right).*

Evaluation of piecewise polynomials

For a function $f \in C(I)$ we define the global piecewise linear interpolant in $V_h^{(1)}$ as

$$\pi_h f(x) = \sum_{i=0}^{m+1} f(x_i)\phi_i(x).$$

Assuming also that $f \in C^2(I)$, the following global interpolation error estimates hold,

$$\begin{aligned}
\|f - \pi_h f\|_\infty &\leq \|hf'\|_\infty, \\
\|f - \pi_h f\|_\infty &\leq \frac{1}{8}\|h^2 f''\|_\infty, \\
\|f' - (\pi_h f)'\|_\infty &\leq \frac{1}{4}\|hf''\|_\infty,
\end{aligned}$$

as a consequence of Theorem 9.2 and Theorem 9.3.

Regression analysis

Now consider the case when we do not have access to an exact function $f(x)$, but only a set of sampled data points $\{(X_i, Y_i)\}_{i=1}^{M}$ with

$$Y_i \approx f(X_i).$$

Here f_N may be chosen as a linear combination of basis functions (9.1), so that

$$\min_{f_N \in V_N} \sum_{i=1}^{M} \|Y_i - f_N(X_i)\|^2 = \min_{\alpha \in R^N} \sum_{i=1}^{M} \|Y_i - \sum_{j=1}^{N} \alpha_j \phi_j(X_i)\|^2,$$

which corresponds to minimization of the residual $b - A\alpha \in R^M$, with $a_{ij} = \phi_j(X_i)$ and $b_i = Y_i$, a least squares problem that can be solved e.g. by forming the normal equations.

Function approximation methods

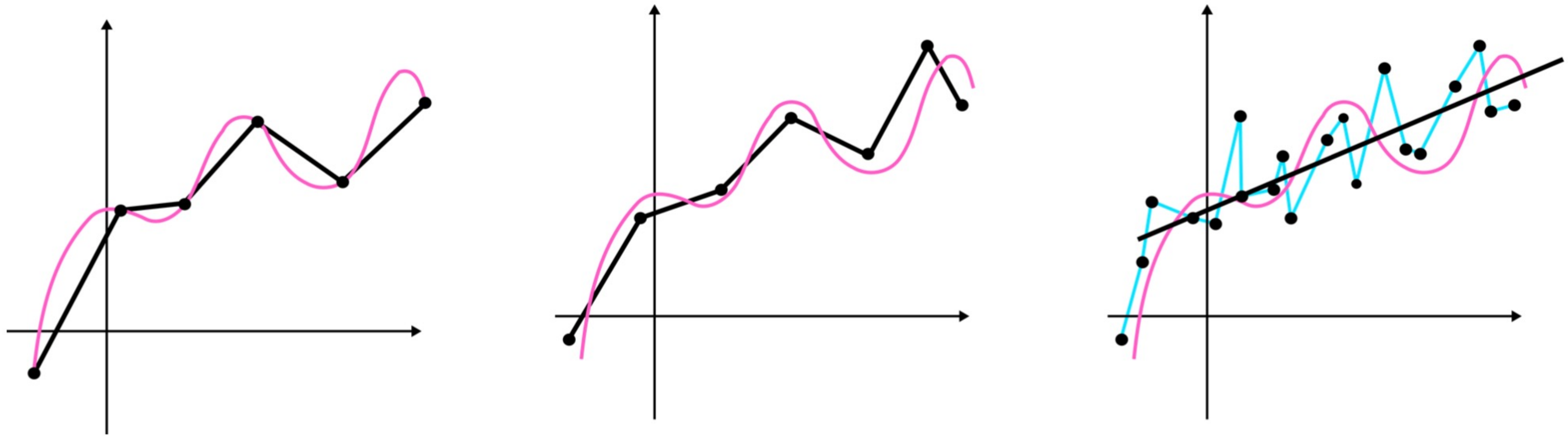
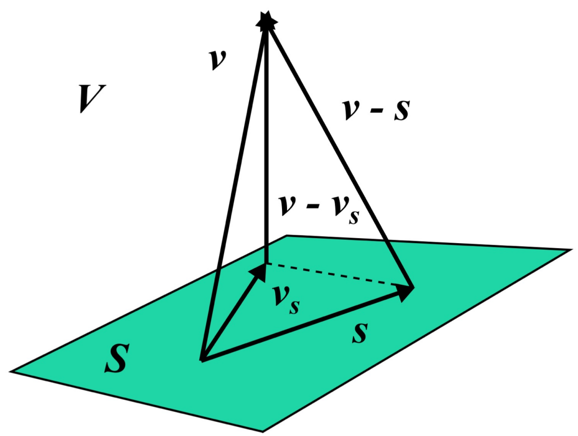


Figure 9.3. *Piecewise polynomial interpolation (left) and M dimensional error minimization (center), and overfitted interpolation compared to $N << M$ dimensional data smoothing (right).*

Projection methods

If V is a Hilbert space with inner product $(\cdot,\cdot)_V$, we can construct an approximation of $f \in V$ in a finite dimensional subspace $V_N \subset V$, in the form of the orthogonal projection onto V_N, $f_N = P_N f$. Recall that the projection error is orthogonal to V_N, $f - P_N f \in V_N^{\perp}$, or equivalently,

$$(f - P_N f, v)_V = 0, \quad \forall v \in V_N.$$



L^2 projection

Of specific interest is the Hilbert space $L^2([a,b])$, with the inner product

$$(f,g) = (f,g)_{L^2([a,b])} = \int_a^b f(x)g(x)\,dx.$$

The L^2 norm is generated by the inner product,

$$\|f\| = \|f\|_2 = (f,f)^{1/2},$$

for which the Cauchy-Schwarz inequality is satisfied by Theorem 1.5. For $f \in L^2([a,b])$, we define the L^2 *projection* $P_N f$ onto $V_N \subset L^2([a,b])$ by

$$(P_N f, v) = (f, v), \quad \forall v \in V_N.$$

L^2 projection

$$\sum_{j=1}^{N} \alpha_j (\phi_j, \phi_i) = (f, \phi_i), \quad \forall i = 1, ..., N,$$

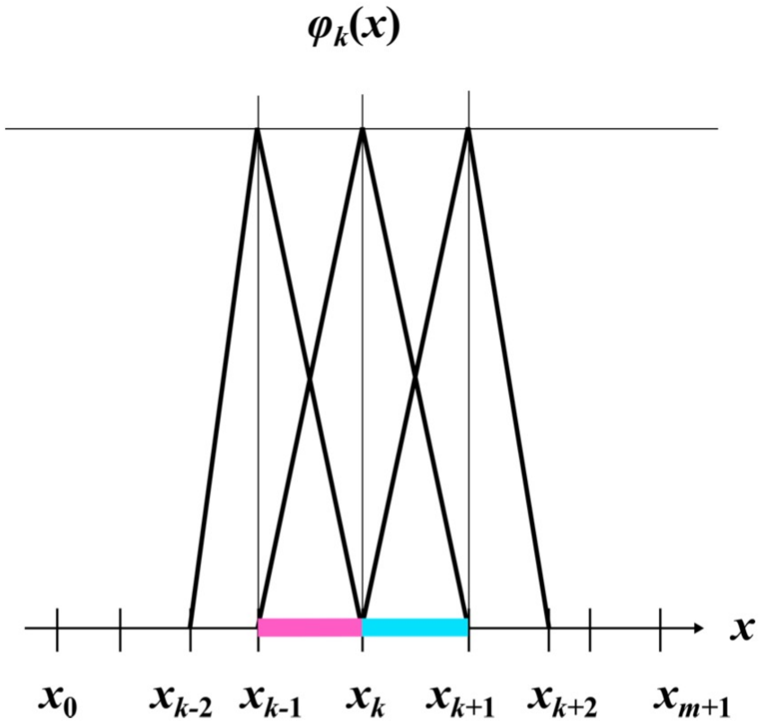
which corresponds to a matrix equation $A\alpha = b$, where $A = (a_{ij})$ and $b = (b_i)$, with

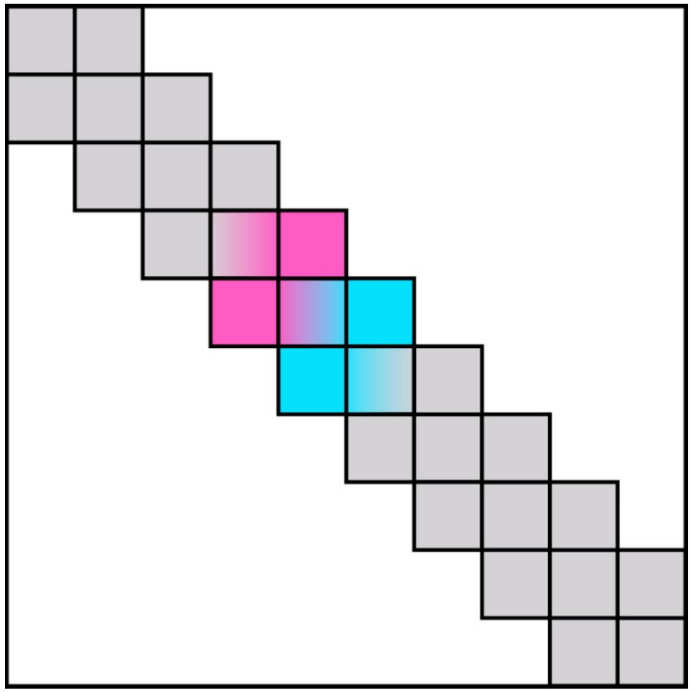
$$a_{ij} = (\phi_j, \phi_i) = \int_a^b \phi_j(x)\phi_i(x)\, dx,$$
$$b_i = (f, \phi_i) = \int_a^b f(x)\phi_i(x)\, dx.$$

From the matrix equation we compute the coordinates $\alpha \in R^N$, to get the L^2 projection

$$P_N f(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x).$$

L^2 projection





L^2 projection

Example 9.6. For a subdivision of the interval $I = [0,1]$ with m internal nodes, let

$$P_h f(x) = \sum_{j=0}^{m+1} \alpha_j \phi_j(x)$$

be the L^2 projection of $f \in L^2(I)$ onto the space of continuous piecewise linear polynomials $V_h^{(1)}$. Theorem 1.8 states that $P_h f$ is the optimal approximation in $L^2(I)$, specifically that

$$\|f - P_h f\| \leq \|f - \pi_h f\|,$$

for $\pi_h f$ the interpolant in $V_h^{(1)}$. The coordinates α_j in (9.16) are then determined from the matrix equation

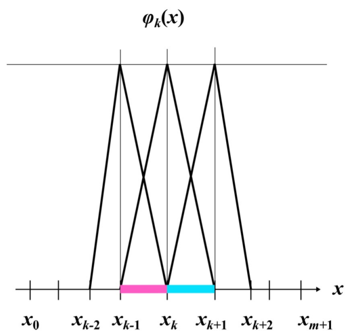
$$A\alpha = b,$$

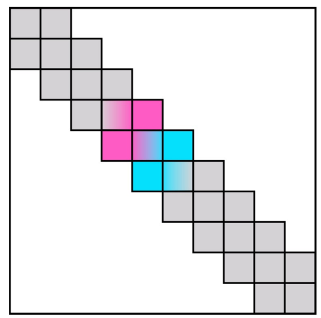
with the *mass matrix* $a_{ij} = (\phi_j, \phi_i)$, *load vector* $b_i = (f, \phi_i)$ and the *solution vector* $\alpha = (\alpha_j)$.

L^2 projection

A is a tridiagonal sparse matrix since $a_{ij} = 0$ for $|i-j| > 1$, and we can compute the components of the matrix from the definition of the basis functions (9.8),

$$
\begin{aligned}
a_{ii} = (\phi_i, \phi_i) &= \int_0^1 \phi_i^2(x)\,dx = \int_{I_i} \lambda_{i,1}^2(x)\,dx + \int_{I_{i+1}} \lambda_{i+1,0}^2(x)\,dx \\
&= \int_{I_i} \frac{(x - x_{i-1})^2}{h_i^2}\,dx + \int_{I_{i+1}} \frac{(x_{i+1} - x)^2}{h_{i+1}^2}\,dx \\
&= \frac{1}{h_i^2}\left[\frac{(x - x_{i-1})^3}{3}\right]_{x_{i-1}}^{x_i} + \frac{1}{h_{i+1}^2}\left[\frac{-(x_{i+1} - x)^3}{3}\right]_{x_i}^{x_{i+1}} = \frac{h_i}{3} + \frac{h_{i+1}}{3}.
\end{aligned}
\tag{9.17}
$$





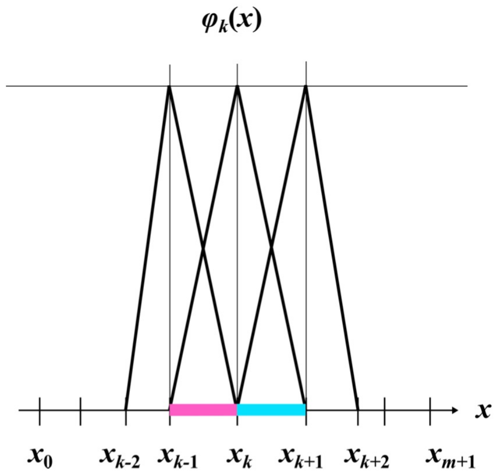
L^2 projection

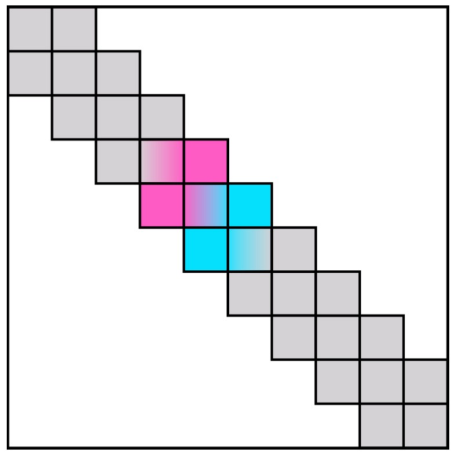
$$
\begin{aligned}
a_{ii+1} &= (\phi_i, \phi_{i+1}) = \int_0^1 \phi_i(x)\phi_{i+1}(x)\,dx = \int_{I_{i+1}} \lambda_{i+1,0}(x)\lambda_{i+1,1}(x)\,dx \\
&= \int_{I_{i+1}} \frac{(x_{i+1}-x)}{h_{i+1}}\frac{(x-x_i)}{h_{i+1}}\,dx \\
&= \frac{1}{h_{i+1}^2}\int_{I_{i+1}} (x_{i+1}x - x_{i+1}x_i - x^2 + xx_i)\,dx \\
&= \frac{1}{h_{i+1}^2}\left[\frac{x_{i+1}x^2}{2} - x_{i+1}x_i x - \frac{x^3}{3} + \frac{x^2 x_i}{2}\right]_{x_i}^{x_{i+1}} \\
&= \frac{1}{6h_{i+1}^2}(x_{i+1}^3 - 3x_{i+1}^2 x_i + 3x_{i+1}x_i^2 - x_i^3) = \frac{(x_{i+1}-x_i)^3}{6h_{i+1}^2} = \frac{h_{i+1}}{6}, \\
a_{ii-1} &= (\phi_i, \phi_{i-1}) = \int_0^1 \phi_i(x)\phi_{i-1}(x)\,dx = \ldots = \frac{h_i}{6}.
\end{aligned}
$$

Assembly algorithm

ALGORITHM 9.2. (A, b) = assemble_system(f).
Input: function **f**
Output: assembled matrix **A** and vector **b**.

```
for k=0:no_elements-1 do
    q = get_no_local_shape_functions(k)
    loc2glob = get_local_to_global_map(k)
    for i=0:q do
        b[i] = integrate_vector(f, k, i)
        for j=0:q do
            a[i,j] = integrate_matrix(k, i, j)
        end for
    end for
    add_to_global_vector(b, loc2glob)
    add_to_global_matrix(a, loc2glob)
end for
return  A, b
```





Transforms

If $\{\phi_i\}_{i=1}^N$ is an orthonormal basis for the approximation space V_N then $(\phi_j, \phi_i) = \delta_{ij}$. It follows that the coordinates in the L^2 projection are obtained as the inner product $\alpha_j = (f, \phi_j)$, without the need to solve any matrix equation. Instead the L^2 projection is given directly as

$$P_N f(x) = \sum_{j=1}^{N} (f, \phi_j)\phi_j(x),$$

which is constructed from the function $f \in L^2(I)$ by computing N inner products that project the function onto the orthonormal basis functions, corresponding to a coordinate transformation, or *transform*. An algorithm for computing the L^2 projection onto an orthonormal basis, therefore, amounts to computing the N integrals (f, ϕ_j).

Fourier series

Example 9.7 (Fourier series). Consider the subspace of the Hilbert space $L^2([0,1])$ spanned by the 2π periodic orthonormal basis $\{e_n\}_{n=-N}^{N}$, the *modes*, with

$$e_n(t) = \exp(2\pi i n t) = \cos(2\pi n t) + i\sin(2\pi n t), \quad t \in [0,1].$$

The L^2 projection of a function $f \in L^2([0,1])$ onto this subspace is the *Fourier series*

$$f(t) = \sum_{n=-N}^{N} (f, e_n) e_n(t).$$

Discrete Fourier Transform

Example 9.8 (Discrete Fourier transform). The *Discrete Fourier Transform* (DFT) acts on a complex vector $x \in C^N$ with the DFT modes $\{q_k\}_{k=0}^{N-1}$, where $q_k = (q_{k,0}, ..., q_{k,N-1})^T$ and

$$q_{k,n} = \exp(2\pi i k n/N) = W_N^{-nk},$$

with $W_N = \exp(-2\pi i/N)$. The modes satisfy an orthogonality condition

$$(q_k, q_l) = q_k^* q_l = \sum_{n=0}^{N-1} W_N^{(k-l)n} = N\delta_{kl}.$$

Wavelets

Assuming that $f(t)$ is a function of time, the Fourier series (9.21) represents the time series in terms of its modes, but the frequency content in each mode is not localized in the time domain since the modes are global sine waves over the whole time domain. In contrast, the wavelet transform is localized in both the time and frequency domains, by using short waves of finite support, referred to as *wavelets*. The wavelet basis functions are constructed by scaling and translation of the *father wavelet* (or *scaling function*) $\phi(t)$, which generates

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k),$$

the *scaling functions* that determine the scales, whereas the *mother wavelet* $\psi(t)$ generates

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k),$$

the *wavelets* which determine the frequencies, that is, differences between scales.

Wavelets

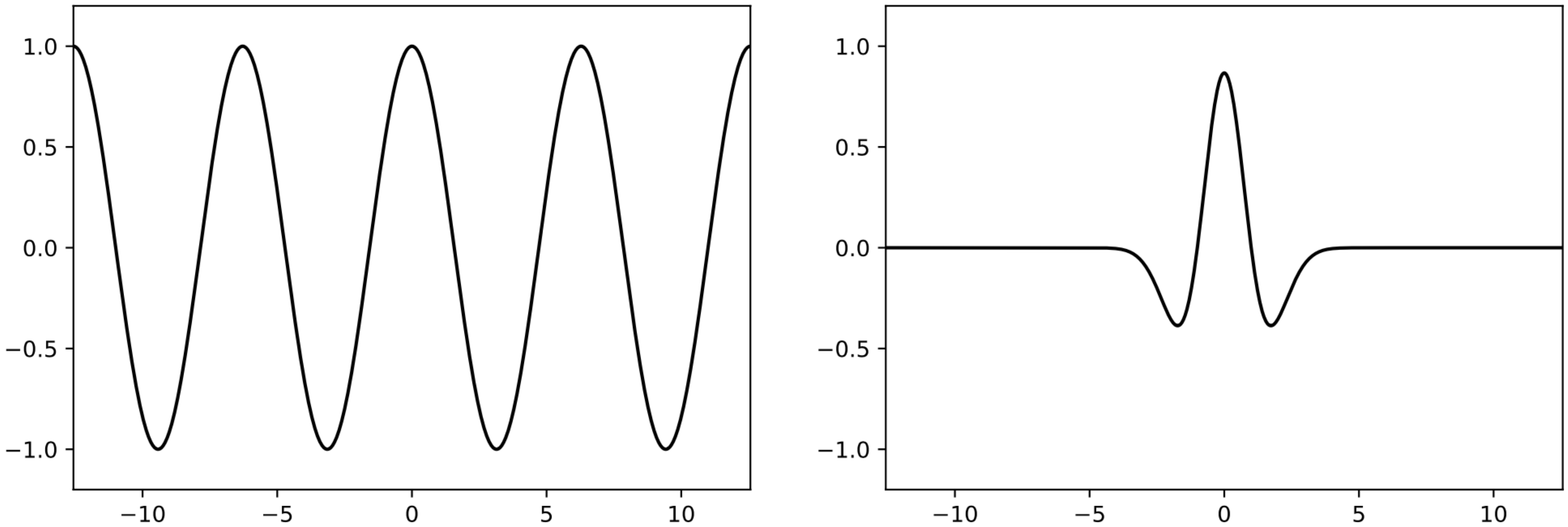
Example 9.9 (Haar wavelet basis). The *Haar wavelet basis* is generated by

$$\phi(t) = \begin{cases} 1, & t \in [0,1), \\ 0, & t \notin [0,1), \end{cases} \qquad \psi(t) = \begin{cases} 1, & t \in [0,1/2), \\ -1, & t \in [1/2,1), \\ 0, & t \notin [0,1). \end{cases}$$

Example 9.10 (Ricker wavelet). The *Ricker wavelet*, or *Mexican hat wavelet*, is constructed from the second derivative of a Gaussian function (10.2),

$$\psi(t) = \frac{2}{\sqrt{3}\pi^{1/4}}(1-t^2)\exp(-t^2/2\sigma^2).$$

Fourier basis function vs Ricker mother wavelet



Continuous wavelet transform

The wavelet space W_{j-1} is the orthogonal complement of the scale space V_{j-1} in V_j,

$$V_j = V_{j-1} \oplus W_{j-1},$$

so that starting from a coarsest scale j_0, $V_j = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus W_{j_0+2} \oplus \cdots \oplus W_{j-1}$, which forms a hierarchical decomposition, a *multiresolution analysis*, of V_j. Hence, any function $f \in V_j$ can be expressed in the orthonormal wavelet basis $\{\phi_{j_0,k}\}_{k\in Z}$ and $\{\psi_{j,k}\}_{k\in Z}$ as

$$f(t) = \sum_{k\in Z_{j_0}} (f, \phi_{j_0,k})\phi_{j_0,k}(t) + \sum_{j=j_0}^{j-1} \sum_{k\in Z_j} (f, \psi_{j,k})\psi_{j,k}(t),$$

and the coordinates in this wavelet basis represent the *continuous wavelet transform*.

Discrete wavelet transform

For a discrete function $f[n] = f(t_n)$ sampled in N points $t_0,\ldots,t_{N-1}$, the *discrete wavelet transform* (DWT) is defined as the expansion of the function in terms of a wavelet basis in the points $n = t_n$,

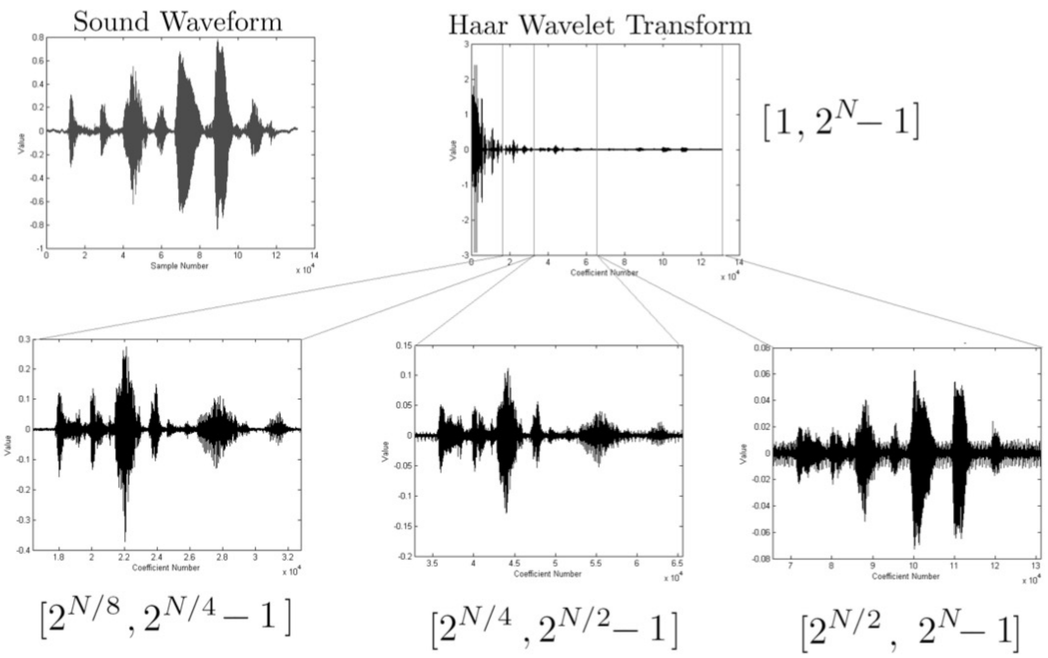
$$f[n] = \sum_{k \in Z_{j_0}} W_\phi[j_0, k]\phi_{j_0,k}[n] + \sum_{j=j_0}^{\infty} \sum_{k \in Z_j} W_\psi[j, k]\psi_{j,k}[n],$$

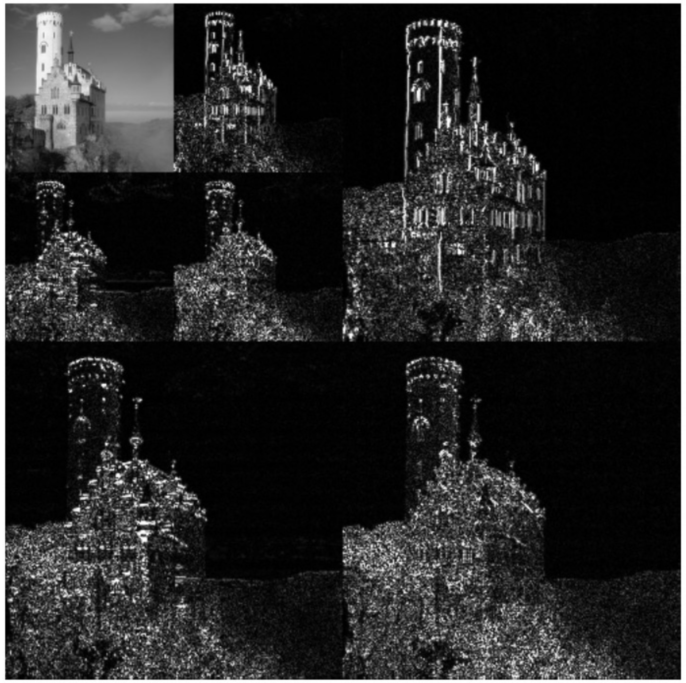
with the wavelet coefficients

$$W_\phi[j_0, k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n]\phi_{j_0,k}[n], \quad W_\psi[j, k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n]\psi_{j,k}[n], \quad j \geq j_0,$$

where $W_\phi[j_0, k]$ corresponds to a mean value over the coarsest scale j_0.

Discrete wavelet transform





Finite element method

The finite element method (FEM) is a generalization of L^2 projection to a solution method for differential equations. Hence, we need to use a Hilbert space V where not only functions but also a suitable representation of their derivatives belongs to $L^2(I)$, referred to as a *Sobolev space*.

A linear partial differential equation with solution $u \in V$ can be expressed in weak form as

$$a(u, v) = L(v), \quad \forall v \in V, \tag{9.24}$$

where $a(\cdot, \cdot) : V \times V \to R$ is a bilinear form, $L(\cdot) : V \to R$ is a linear form, and the Sobolev space V is chosen such that the bilinear and linear forms are well defined, that is,

$$|a(u, v)| < \infty, \quad |L(v)| < \infty.$$

Finite element method

We also need to prescribe boundary conditions for the endpoints of the interval I, which may either take the form of a *Dirichlet boundary condition* which is incorporated into the Sobolev space V, or a *Neumann boundary condition* that is part of the bilinear and linear forms.

In a finite element method we seek an approximate solution to the differential equation (9.24), of the form

$$U(x) = \sum_{j=1}^{N} U_j \phi_j(x).$$

Here $\{\phi_j(x)\}_{j=1}^N$ is a set of finite element basis functions that span a finite element approximation space $V_N \subset V$, and $\{U_j\}_{j=1}^N$ are the coordinates of $U \in V_N$ in that basis.

Finite element method

The approximation can be determined from the variational formulation

$$a(U, v) = L(v), \quad \forall v \in V_N,$$

which corresponds to a system of linear equations

$$A\alpha = b,$$

where $\alpha = (U_j)$ is the vector of N coordinates which determine the approximation by equation (9.25), $A = (a_{ij})$ is an $N \times N$ matrix, and $b = (b_i)$ an N vector, with

$$a_{ij} = a(\phi_j, \phi_i), \quad b_i = L(\phi_i).$$

The Poisson equation

Example 9.12. Consider the Poisson equation

$$-u''(x) = f(x), \quad x \in (0,1),$$

with the two Dirichlet boundary conditions $u(0) = 0$ and $u(1) = 0$. If we multiply both sides of the equation by a test function v and integrate over the interval $[0,1]$, we get

$$-\int_0^1 u''(x)v(x)\,dx = \int_0^1 f(x)v(x)\,dx.$$

By partial integration of the left hand side of the equation, using the boundary conditions, we are lead to the weak form of the equation: find $u \in V$ such that

$$a(u,v) = L(v), \quad v \in V,$$

with a Sobolev space V for which the boundary conditions are satisfied.

The Poisson equation

By partial integration of the left hand side of the equation, using the boundary conditions, we are lead to the weak form of the equation: find $u \in V$ such that

$$a(u, v) = L(v), \quad v \in V,$$

with a Sobolev space V for which the boundary conditions are satisfied. The bilinear and linear forms are defined by

$$a(u, v) = \int_0^1 u'(x)v'(x)\, dx, \quad L(v) = \int_0^1 f(x)v(x)\, dx,$$

and in the FEM approximation (9.25) the degrees of freedom represent the internal nodes of the mesh, since the endpoint degrees of freedom are determined by the Dirichlet boundary condition.

Analysis of the finite element method

Assume that the bilinear form $a(\cdot, \cdot)$ is symmetric, and that its associated quadratic form is positive definite. Then the bilinear form defines an inner product in the Sobolev space V, with an associated norm,

$$(v, w)_V = a(v, w), \quad v, w \in V, \quad \|v\|_V = (v, v)_V^{1/2}, \quad v \in V.$$

If the linear form $L(\cdot)$ is a bounded linear functional with respect to this norm,

$$|L(v)| \leq M \|v\|_V, \quad \forall v \in V,$$

then by Riesz representation theorem there exists a unique solution $u \in V$ to equation (9.24), identical to the Riesz representer of $L(\cdot)$. Analogously, Riesz representation theorem states that there exists a unique solution $U \in V_N$ to the finite element equation (9.26).

Analysis of the finite element method

Since the bilinear form $a(\cdot,\cdot)$ defines an inner product in V, the finite element solution $U \in V_N$ represent an orthogonal projection of the exact solution $u \in V$ to the finite element approximation space $V_N \subset V$,

$$(u - U, v)_V = 0, \quad \forall v \in V_N,$$

because for any $v \in V_N$,

$$0 = L(v) - L(v) = a(u, v) - a(U, v) = a(u - U, v) = (u - U, v)_V.$$

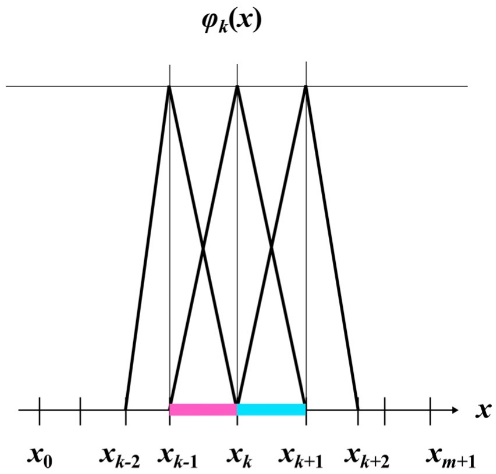
Therefore, by Theorem 1.8, it follows that the finite element solution U is the best possible approximation in V_N with respect to the norm $\|\cdot\|_V$. Specifically, the error in the finite element solution is bounded by the interpolation error,

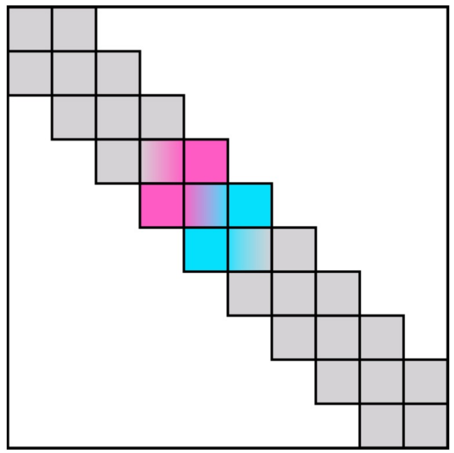
$$\|u - U\|_V \leq \|u - \pi_N u\|_N$$

FEM assembly algorithm

ALGORITHM 9.2. (A, b) = assemble_system(f).
Input: function **f**
Output: assembled matrix **A** and vector **b**.

```
for k=0:no_elements-1 do
    q = get_no_local_shape_functions(k)
    loc2glob = get_local_to_global_map(k)
    for i=0:q do
        b[i] = integrate_vector(f, k, i)
        for j=0:q do
            a[i,j] = integrate_matrix(k, i, j)
        end for
    end for
    add_to_global_vector(b, loc2glob)
    add_to_global_matrix(a, loc2glob)
end for
return A, b
```





Parallelization of the finite element method