ADDITIVE NONLINEAR OPERATOR APPROXIMATION

The linear algebra concepts arising from study of linear mappings between vector spaces $\boldsymbol{f}: U \to V$, $\boldsymbol{f}(\alpha \boldsymbol{u} + \beta \boldsymbol{v}) = \alpha \boldsymbol{f}(\boldsymbol{u}) + \beta \boldsymbol{f}(\boldsymbol{v})$, are widely applicable to nonlinear functions also. The study of nonlinear approximation starts with the simplest case of approximation of a function with scalar values and arguments, $f: \mathbb{R} \to \mathbb{R}$ through additive corrections.

1. Function spaces

An immediate application of the linear algebra framework is to construct vector spaces of real functions $\mathscr{F} = (F, +, \cdot)$, with $F = \{f | f: \mathbb{R} \to \mathbb{R}\}$, and the addition and scaling operations induced from $\mathbb{R}$,

$$(\alpha f + \beta g)(t) = \alpha f(t) + g(t), f, g \in F, \alpha, \beta \in \mathbb{R}.$$

Comparing with the real vector space $(\mathbb{R}^m, +, \cdot)$ in which the analogous operation is $\alpha \boldsymbol{u} + \beta \boldsymbol{v}, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m, \alpha, \beta \in \mathbb{R}$, or componentwise

$$(\alpha \boldsymbol{u} + \beta \boldsymbol{v})_i = \alpha u_i + \beta v_i, i = 1, 2, \ldots, m,$$

the key difference that arises is the dimension of the set of vectors. Finite-dimensional vectors within $\mathbb{R}^m$ can be regarded as functions defined on a finite set $\boldsymbol{u} \Leftrightarrow u: \{1, 2, \ldots, m\} \to \mathbb{R}$, with $u(i) = u_i$. The elements of F are however functions defined on $\mathbb{R}$, a set with cardinality $\mathfrak{c} = 2^{\aleph_0}$, with $\aleph_0$ the cardinality of the naturals $\mathbb{N}$. This leads to a review of the concept of a basis for this infinite-dimensional case.

1.1. Infinite dimensional basis set

In the finite dimensional case $\boldsymbol{B} \in \mathbb{R}^{m \times m}$ constituted a basis if any vector $\boldsymbol{y} \in \mathbb{R}^m$ could be expressed uniquely as a linear combination of the column vectors of

$$\forall \boldsymbol{y} \in \mathbb{R}^m, \exists ! \boldsymbol{c} \in \mathbb{R}^m \text{ such that } \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c} = c_1 \boldsymbol{b}_1 + \cdots + c_m \boldsymbol{b}_m.$$

While the above finite sum is well defined, there is no consistent definition of an infinite sum of vectors. As a simple example, in the vector space of real numbers $\mathscr{R}_1 = (\mathbb{R}, +, \cdot)$, any finite sum of reals is well defined, for instance

$$S_n = \sum_{k=0}^{n} (-1)^k = \begin{cases} 1 & \text{if } n \text{ even} \\ 0 & \text{if } n \text{ odd} \end{cases}$$

but the limit $S_{n \to \infty}$ cannot be determined. This leads to the necessity of seeking *finite-dimensional* linear combinations to span a vector space $\mathscr{V} = (V, S, +, \cdot)$. First, define linear independence of an infinite (possibly uncountable) set of vectors $\mathscr{B} = \{v_\gamma | \gamma \in \Gamma, v_\gamma \in V\}$, where Γ is some indexing set.

DEFINITION. *The vector set $\mathscr{B} = \{v_\gamma | \gamma \in \Gamma, v_\gamma \in V\}$, is linearly independent if the only $n \in \mathbb{N}$ scalars, $x_1, \ldots, x_n \in S$, that satisfy*

$$x_1 v_{\gamma_1} + \ldots + x_n v_{\gamma_n} = 0, \gamma_i \in \Gamma \tag{1}$$

are $x_1 = 0$, $x_2 = 0, \ldots, x_n = 0$.

The important aspect of the above definition is that all finite vector subsets are linearly independent. The same approach is applied in the definition of a spanning set.

DEFINITION. *Vectors within the set $\mathscr{B} = \{v_\gamma | \gamma \in \Gamma, v_\gamma \in V\}$, span V, stated as $V = \operatorname{span}(\mathscr{B})$, if for any $u \in V$ there exist $n \in \mathbb{N}$ scalars, $x_1, \ldots, x_n \in S$, such that*

$$x_1 v_{\gamma_1} + \ldots + x_n v_{\gamma_n} = u, \gamma_i \in \Gamma. \tag{2}$$

This now allows a generally-applicable definition of basis and dimension.

DEFINITION. *The vector set $\mathscr{B} = \{v_\gamma | \gamma \in \Gamma, v_\gamma \in V\}$ is a basis for vector space $\mathscr{V} = (V, S, +, \cdot)$ if*

1. *$\mathscr{B}$ is linearly independent;*
2. *$\operatorname{span}(\mathscr{B}) = V$.*

DEFINITION. *The dimension of a vector space $\mathscr{V} = (V, S, +, \cdot)$ is the cardinality of a basis set $\mathscr{B}$, $\dim(\mathscr{V}) = |\mathscr{B}|$.*

The use of finite sums to define linear independence and bases is not overly restrictive since it can be proven that every *vector space has a basis*. The proof of this theorem is based on Zorn's lemma from set theory, and asserts exsitence of a basis, but no constructive procedure. The difficulty in practical construction of bases for infinite dimensional vector spaces is ascertained through basic examples.

Example. $\mathcal{R}_\infty$. As a generalization of $\mathcal{R}_m = (\mathbb{R}^m, \mathbb{R}, +, \cdot)$, consider the vector space of real sequences $\{x_n\}_{n\in\mathbb{N}}$ represented as a vectors with a countably infinite number of components $\boldsymbol{x} = [\ x_1\ \ x_2\ \ x_3\ \ \ldots\ ]^T$. Linear combinations are defined by

$$\alpha\boldsymbol{x} + \beta\boldsymbol{y} = [\ \alpha x_1 + \beta y_1\ \ \alpha x_2 + \beta y_2\ \ \alpha x_3 + \beta y_3\ \ \ldots\ ]^T.$$

Let $\boldsymbol{e}_i$ denote the vector of all zeros except the i^{th} position. In $\mathbb{R}^m$, the identity matrix $\boldsymbol{I} = [\ \boldsymbol{e}_1\ \ldots\ \boldsymbol{e}_m\ ]$ was a basis, but this does not generalize to $\mathbb{R}^\infty$; for example the vector $\boldsymbol{v} = [\ 1\ \ 1\ \ 1\ \ \ldots\ ]^T$ cannot be obtained by finite linear combination of the $\boldsymbol{e}_i$ vectors. In fact, there is no countable set of vectors that spans $\mathbb{R}^\infty$.

Example. $P(\mathbb{R})$. The vector space of polynomials $P(\mathbb{R}) = \{p | p(t) = c_n t^n + c_{n-1} t^{n-1} + \cdots + c_0, n \in \mathbb{N}, c_i \in \mathbb{R}, i = 0, 1, \ldots, N\}$ on the real line has an easily constructed basis, namely the set of the monomials

$$\mathcal{B}(t) = \{t^n \,|\, n \in \mathbb{N}\},$$

an infinite set with the cardinality as the naturals $|\mathcal{B}| = |\mathbb{N}| = \aleph_0$.

1.2. Alternatives to the concept of a basis

The difficulty in ascribing significance to an infinite sum of vectors $\sum_{i=1}^{\infty} \boldsymbol{v}_i$ can be resolved by endowing the vector space with additional structure, in particular a way to define convergence of the partial sums

$$\boldsymbol{s}_n = \sum_{i=1}^{n} \boldsymbol{v}_i$$

to a limit $\lim_{n\to\infty} \boldsymbol{s}_n = \boldsymbol{v}$.

Fourier series. One approach is the introduction of an inner product $(\boldsymbol{u}, \boldsymbol{v})$ and the associated norm $\|\boldsymbol{u}\| = (\boldsymbol{u}, \boldsymbol{v})^{1/2}$. A considerable advantage of this approach is that it not only allows infinite linear combinations, but also definition of orthonormal spanning sets. An example is the vector space of continuous functions defined on $[-\pi, \pi]$ with the inner product

$$(f, g) = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t)\, g(t)\, \mathrm{d}t,$$

and norm $\|f\| = (f, f)^{1/2}$. An orthonormal spanning set for $C[-\pi, \pi]$ is given by

$$\left\{\frac{1}{2}\right\} \bigcup \{\cos(nx) | n \in \mathbb{N}^+\} \bigcup \{\sin(nx) | n \in \mathbb{N}^+\}.$$

Vector spaces with an inner product are known as *Hilbert spaces*.

Taylor series. Convergence of infinite sums can be determined through a norm, without the need of an inner product. An example is the space of real-analytic functions with the inf-norm

$$\|f\|_\infty = \sup_x |f(t)|,$$

for which a spanning set is given by the monomials $\{1, t, t^2, \ldots\}$, and the infinite exapnsion

$$f(t) = \sum_{k=0}^{\infty} a_k\, (t - c)^k$$

is convergent, with coefficients given by the Taylor series

$$f(t) = f(c) + \frac{f'(c)}{1!}(t - c) + \cdots, a_k = \frac{f^{(k)}(c)}{k!}.$$

Note that orthogonality of the spanning set cannot be established, absent an inner product.

1.3. Common function spaces

Several function spaces find widespread application in scientific computation. An overview is provided in Table 1.

$B(\mathbb{R})$	bounded functions		
$C(\mathbb{R})$	continuous functions	$C^r(\mathbb{R})$	with continuous derivatives up to r
$C_c(\mathbb{R})$	with compact support	$C_c^r(\mathbb{R})$	and compact support
$C_0(\mathbb{R})$	that vanish at infinity	$C^\infty(\mathbb{R})$	smooth functions
$L^p(\mathbb{R})$	with finite p-norm	$W^{k,p}(\mathbb{R})$	Sobolev space, with norm
	$\|f\|_p = (\int_{-\infty}^{\infty} \|f(t)\|^2 \mathrm{d}t)^{1/p}$		$\|f\|_{k,p} = (\sum_{i=0}^{k} \|f^{(i)}\|_p^p)^{1/p}$

Table 1. Common vector spaces of functions

2. Interpolation

The interpolation problem seeks the representation of a function f known only through a sample data set $\mathscr{D} = \{(x_i, y_i = f(x_i)) | i = 0, \ldots, m\} \subset \mathbb{R} \times \mathbb{R}$, by an approximant $p(t)$, obtained through combination of elements from some family of computable functions, $\mathscr{B} = \{b_0, \ldots, b_n | b_k : \mathbb{R} \to \mathbb{R}\}$. The approximant $p(t)$ is an interpolant of $\mathscr{D}$ if

$$p(x_i) = f(x_i) = y_i, i = 0, \ldots, m,$$

or $p(t)$ passes through the known poles (x_i, y_i) of the function f. The objective is to use $p(t)$ thus determined to approximate the function f at other points. Assuming $x_0 < x_1 < \cdots < x_m$, evaluation of $p(t)$ at $t \in (x_0, x_m)$ is an *interpolation*, while evaluation at $t < x_0$ or $t > x_m$, is an *extrapolation*. The basic problems arising in interpolation are:

- choice of the family from which to build the approximant $p(t)$;
- choice of the combination technique;
- estimation of the error of the approximation given some knowledge of f.

◦ Algorithms for interpolation of real functions can readily be extended to more complicated objects, e.g., interpolation of matrix representations of operators. Implementation is aided by programming language polymorphism as in Julia.

2.1. Additive corrections

As to be expected, a widely used combination technique is linear combination,

$$p(t) = c_0 b_0(t) + c_1 b_1(t) + \cdots + c_n b_n(t).$$

The idea is to capture the nonlinearity of $f(t)$ through the functions $b_0(t), \ldots, b_n(t)$, while maintaining the framework of linear combinations. Sampling of $b_j(t)$ at the poles x_i of a data set $\mathscr{D}$, constructs the vectors

$$\boldsymbol{b}_j = b_j(\boldsymbol{x}) = [\ b_j(x_0) \ \ \ldots \ \ b_j(x_m)\ ]^T \in \mathbb{R}^{m+1},$$

which gathered together into a matrix leads to the formulation of the interpolation problem as

$$\boldsymbol{B}\boldsymbol{c} = \boldsymbol{y} = \boldsymbol{I}\boldsymbol{y}, \boldsymbol{B} \in \mathbb{R}^{(m+1)\times(n+1)}. \tag{3}$$

Before choosing some specific function set $\mathscr{B}$, some general observations are useful.

1. The function values $y_i = f(x_i), i = 0, \ldots, m$, are directly incorporated into the interpolation problem (3). Any estimate of the error at other points requires additional information on f. Such information can be furnished by bounds on the function values, or knowledge of its derivatives for example.
2. A solution to (3) exists if $\boldsymbol{y} \in C(\boldsymbol{B})$. Economical interpolations would use $n < m$ functions in the set $\mathscr{B}$, hopefully $n \ll m$.

2.2. Polynomial interpolation

Monomial form of interpolating polynomial. As noted above, the vector space of polynomials $P(\mathbb{R})$ has an easily constructed basis, that of the monomials $m_j(t) = t^j$ which shall be organized as a row vector of functions

$$\mathscr{M}(t) = [\ 1 \ \ t \ \ t^2 \ \ \ldots\ ].$$

With $\mathscr{M}_{n+1}(t)$ denoting the first $n+1$ monomials

$$\mathscr{M}_{n+1}(t) = [\ 1\ \ t\ \ \dots\ \ t^n\ ],$$

a polynomial of degree n is the linear combination

$$p(t) = \mathscr{M}_{n+1}(t)\,\boldsymbol{a} = [\ 1\ \ t\ \ \dots\ \ t^n\ ] \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = a_0 + a_1 t + \cdots + a_n t^n.$$

Let $\boldsymbol{M} \in \mathbb{R}^{(m+1)\times(n+1)}$ denote the matrix obtained from evaluation of the first $n+1$ monomials at the sample points $\boldsymbol{x} = [\ x_0\ \ x_1\ \ \dots\ \ x_m\ ]^T$,

$$\boldsymbol{M} = \mathscr{M}_{n+1}(\boldsymbol{x}).$$

The above notation conveys that a finite-dimensional matrix $\boldsymbol{M} \in \mathbb{R}^{(m+1)\times(n+1)}$ is obtained from evaluation of the row vector of the monomial basis function $\mathscr{M}(x)\colon \mathbb{R} \to \mathbb{R}^{n+1}$, at the column vector of sample points $\boldsymbol{x} \in \mathbb{R}^{m+1}$. The interpolation condition $p(\boldsymbol{x}) = \boldsymbol{y}$ leads to the linear system

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{y}. \tag{4}$$

For a solution to exist for arbitrary $\boldsymbol{y}$, $\boldsymbol{M}$ must be of full rank, hence $m = n$, in which case $\boldsymbol{M}$ becomes the Vandermonde matrix

$$\boldsymbol{M} = \begin{bmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^n \end{bmatrix},$$

known to be ill-conditioned. Since $\boldsymbol{M}$ is square and of full rank, (4) has a unique solution.
Finding the polynomial coefficients by solving the above linear system requires $\mathcal{O}(n^3/3)$ operations. Evaluation of the monomial form is economically accomplished in $\mathcal{O}(n)$ operations through Horner's scheme

$$p(t) = a_0 + (a_1 + \cdots + (a_{n-2} + (a_{n-1} + a_n t)\cdot t)\cdot t)\cdot t. \tag{5}$$

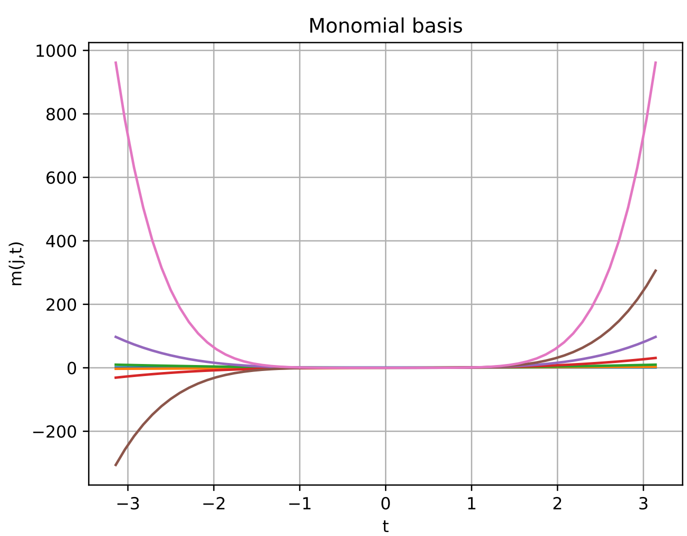
- 

Figure 1. Monomial basis over interval $[-\pi, \pi]$

- **Algorithm (Horner's scheme)**

  Input: $t \in \mathbb{R}, \boldsymbol{a} \in \mathbb{R}^{n+1}$
  Output: $p(t) = a_0 + a_1 t + \cdots + a_n t^n$
  $p = a_n$
  for $k = n-1$ downto 0

$p = a_k + p \cdot t$
end
return p

Lagrange form of interpolating polynomial. It is possible to reduce the operation count to find the interpolating polynomial by carrying out an LU decomposition of the monomial matrix $\boldsymbol{M}$,

$$\mathcal{M}_{n+1}(\boldsymbol{x}) = \boldsymbol{M} = \boldsymbol{L}\boldsymbol{U}.$$

Let $\mathcal{L}_{n+1}(t) = [\ \ell_0(t)\ \ \ell_1(t)\ \ \ldots\ \ \ell_n(t)\ ]$ denote another set of basis functions that evaluates to the identity matrix at the sample points $\boldsymbol{x}$, such that $\mathcal{L}_{n+1}(\boldsymbol{x}) = \boldsymbol{I}$,

$$\mathcal{M}_{n+1}(\boldsymbol{x}) = \boldsymbol{M} = \boldsymbol{L}\boldsymbol{U} = \boldsymbol{I}\boldsymbol{L}\boldsymbol{U} = \mathcal{L}_{n+1}(\boldsymbol{x})\,\boldsymbol{L}\boldsymbol{U}.$$

For arbitrary t, the relationship

$$\mathcal{M}_{n+1}(t) = \mathcal{L}_{n+1}(t)\,\boldsymbol{L}\boldsymbol{U},$$

describes a linear mapping between the monomials $\mathcal{M}_{n+1}(t)$ and the $\mathcal{L}_{n+1}(t)$ functions, a mapping which is invertible since $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{U}$ is of full rank

$$\mathcal{L}_{n+1}(t) = \mathcal{M}_{n+1}(t)\,\boldsymbol{U}^{-1}\boldsymbol{L}^{-1}.$$

Note that organization of bases as row vectors of functions leads to linear mappings expressed through right factors.

- The LU factorization of the Vandermonde matrix can be determined analytically, as exemplified for $n = 3$ by

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & \frac{x_0 - x_2}{x_0 - x_1} & 1 & 0 \\ 1 & \frac{x_0 - x_3}{x_0 - x_1} & \frac{(x_0 - x_3)(x_3 - x_1)}{(x_0 - x_2)(x_2 - x_1)} & 1 \end{pmatrix} \begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 & x_1^3 - x_0^3 \\ 0 & 0 & (x_0 - x_2)(x_1 - x_2) & (x_0 - x_2)(x_1 - x_2)(x_0 + x_1 + x_2) \\ 0 & 0 & 0 & -((x_0 - x_3)(x_3 - x_1)(x_3 - x_2)) \end{pmatrix}$$

- Both factors can be inverted analytically, e.g., for $n = 3$,

$$\boldsymbol{L}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \frac{x_1 - x_2}{x_0 - x_1} & \frac{x_2 - x_0}{x_0 - x_1} & 1 & 0 \\ \frac{(x_1 - x_3)(x_3 - x_2)}{(x_0 - x_1)(x_0 - x_2)} & \frac{(x_0 - x_3)(x_2 - x_3)}{(x_0 - x_1)(x_1 - x_2)} & \frac{(x_0 - x_3)(x_1 - x_3)}{(x_0 - x_2)(x_2 - x_1)} & 1 \end{pmatrix},$$

$$\boldsymbol{U}^{-1} = \begin{pmatrix} 1 & \frac{x_0}{x_0 - x_1} & -\frac{x_0 x_1}{(x_0 - x_2)(x_2 - x_1)} & \frac{x_0 x_1 x_2}{(x_0 - x_3)(x_3 - x_1)(x_3 - x_2)} \\ 0 & \frac{1}{x_1 - x_0} & \frac{x_0 + x_1}{(x_0 - x_2)(x_2 - x_1)} & -\frac{x_1 x_2 + x_0(x_1 + x_2)}{(x_0 - x_3)(x_3 - x_1)(x_3 - x_2)} \\ 0 & 0 & \frac{1}{(x_0 - x_2)(x_1 - x_2)} & \frac{x_0 + x_1 + x_2}{(x_0 - x_3)(x_3 - x_1)(x_3 - x_2)} \\ 0 & 0 & 0 & -\frac{1}{(x_0 - x_3)(x_3 - x_1)(x_3 - x_2)} \end{pmatrix}.$$

- The functions that result for $n = 3$

$$\left\{ \frac{(t - x_1)(t - x_2)(t - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)}, \frac{(t - x_0)(t - x_2)(t - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)}, \frac{(t - x_0)(t - x_1)(t - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)}, \frac{(t - x_0)(t - x_1)(t - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \right\},$$

  can be generalized as

$$\ell_i(t) = \prod_{j=0}^{n}{}' \frac{t - x_j}{x_i - x_j},$$

  known as the *Lagrange basis set*, where the prime on the product symbol skips the index $j = i$. Note that each member of the basis is a polynomial of degree n.

By construction, through the condition $\mathcal{L}_{n+1}(\boldsymbol{x}) = \boldsymbol{I}$, a Lagrange basis function evaluated at a sample point is

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

A polynomial of degree n is expressed as a linear combinations of the Lagrange basis functions by

$$p(t) = \mathscr{L}_{n+1}(t)\boldsymbol{c} = [\ \ell_0(t)\ \ \ell_1(t)\ \ \ldots\ \ \ell_n(t)\ ] \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = c_0\ell_0(t) + c_1\ell_1(t) + \ldots c_n\ell_n(t).$$

The interpolant of data $\{(x_i, y_i = f(x_i)), i = 0, 1, \ldots, n\}$ is determined through the conditions

$$p(\boldsymbol{x}) = \boldsymbol{y} = \mathscr{L}_{n+1}(\boldsymbol{x})\boldsymbol{c} = \boldsymbol{I}\boldsymbol{c} = \boldsymbol{c} \Rightarrow \boldsymbol{c} = \boldsymbol{y},$$

i.e., the linear combination coefficients are simply the sampled function values $c_i = y_i = f(x_i)$.

$$p(t) = \sum_{i=0}^{n} y_i\, \ell_i(t) = \sum_{i=0}^{n} y_i \prod_{j=0}^{n}{}' \ \frac{t - x_j}{x_i - x_j}. \tag{7}$$

Determining the linear combination coefficients may be without cost, but evaluation of the Lagrange form (7) of the interpolating polynomial requires $\mathscr{O}(n^2)$ operations, significantly more costly than the $\mathscr{O}(n)$ operations required by Horner's scheme (5)

◦ **Algorithm (Lagrange evaluation)**

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
Output: $p(t) = \sum_{i=0}^{n} y_i \prod_{j=0}^{n}{}' \ (t - x_j)/(x_i - x_j)$
$p = 0$
for $i = 0$ to n
  $w = 1$
  for $j = 0$ to n
    if $j \neq i$ then $w = w\,(t - x_j)/(x_i - x_j)$
  end
  $p = p + w \cdot y_i$
end
return p

◦

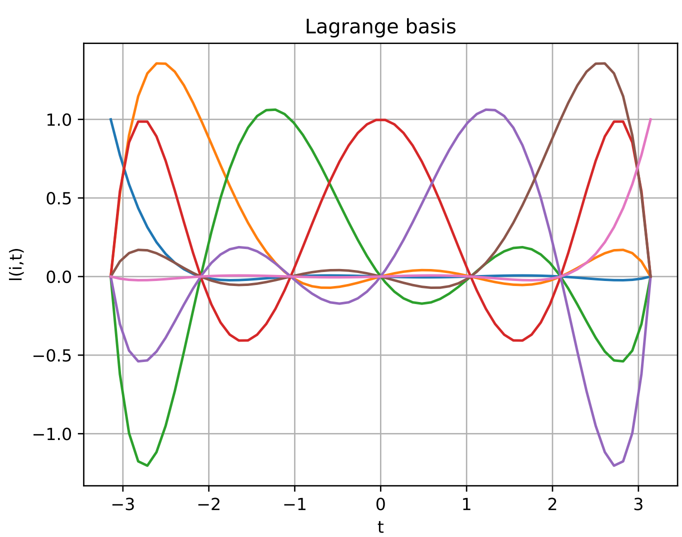


Figure 2. Lagrange basis for $n = 6$ for $\sin(x)$ over interval $[-\pi, \pi]$

A reformulation of the Lagrange basis can however reduce the operation count. Let $\ell(t) = \prod_{k=0}^{n} (t - x_k)$, and rewrite $\ell_i(t)$ as

$$\ell_i(t) = \prod_{j=0}^{n}{}' \frac{t - x_j}{x_i - x_j} = \ell(t) \frac{w_i}{t - x_i},$$

with the weights

$$w_i = \prod_{j=0}^{n}{}' \frac{1}{x_i - x_j},$$

depending only on the function sample arguments x_i, but not on the function values y_i. The interpolating polynomial is now

$$p(t) = \sum_{i=0}^{n} y_i \, \ell_i(t) = \ell(t) \sum_{i=0}^{n} y_i \frac{w_i}{t - x_i}.$$

Interpolation of the function $g(t) = 1$ would give

$$1 = \ell(t) \sum_{i=0}^{n} \frac{w_i}{t - x_i},$$

and taking the ratio yields

$$p(t) = \frac{\sum_{i=0}^{n} y_i \frac{w_i}{t - x_i}}{\sum_{i=0}^{n} \frac{w_i}{t - x_i}}, \tag{9}$$

known as the barycentric Lagrange formula (by analogy to computation of a center of mass). Evaluation of the weights w_i costs $\mathcal{O}(n^2)$ operations, but can be done once for any set of x_i. The evaluation of $p(t)$ now becomes an $\mathcal{O}(2n)$ process, comparable in cost to Horner's scheme.

◦ **Algorithm (Barycentric Lagrange evaluation)**

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
Output: $p(t) = \left(\sum_{i=0}^{n} y_i \frac{w_i}{t - x_i} \right) / \left(\sum_{i=0}^{n} \frac{w_i}{t - x_i} \right)$

```
for i = 0 to n
    w_i = 1
  for j = 0 to n
    if j ≠ i w_i = w_i / (x_i − x_j)
  end
end
q = r = 0
for i = 0 to n
   s = w_i / (t − x_i);  q = q + y_i s; r = r + s
end
p = q / r
return p
```

Newton form of interpolating polynomial. Inverting only one factor of the $\mathcal{M}_{n+1}(t) = \mathcal{L}_{n+1}(t) \boldsymbol{L} \boldsymbol{U}$ mapping yields yet another basis set $\mathcal{S}(t) = [\ N_0(t) \ \ N_1(t) \ \ N_2(t) \ \ \ldots\ ]$

$$\mathcal{M}_{n+1}(t) \boldsymbol{U}^{-1} = \mathcal{L}_{n+1}(t) \boldsymbol{L} = \mathcal{S}_{n+1}(t).$$

◦ The first four basis polynomials are

$$\left\{ 1, \frac{t - x_0}{x_1 - x_0}, \frac{(t - x_0)(t - x_1)}{(x_2 - x_0)(x_2 - x_1)}, \frac{(t - x_0)(t - x_1)(t - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \right\},$$

with $N_0(t) = 1$, and in general

$$N_k(t) = \prod_{j=0}^{k-1} \frac{t - x_j}{x_k - x_j},$$

for $k>0$.

Computation of the scaling factors $w_k = 1/\prod_{j=0}^{k-1}(x_k - x_j)$ would require $\mathcal{O}(n^2/2)$ operations, but can be avoided by redefining the basis set as $\mathcal{N}(t) = [\ n_0(t)\ \ n_1(t)\ \ n_2(t)\ \ \ldots\ ]$, with $n_0(t) = 1$, and

$$n_k(t) = \prod_{j=0}^{k-1}(t - x_j),$$

known as the *Newton basis*. As usual, the coefficients $\boldsymbol{d} \in \mathbb{R}^{n+1}$ of the linear combination of Newton polynomials

$$p(t) = \mathcal{N}_{n+1}(t)\boldsymbol{c} = [\ n_0(t)\ \ n_1(t)\ \ \ldots\ \ n_n(t)\ ]\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = d_0 n_0(t) + d_1 n_1(t) + \ldots + d_n n_n(t),$$

are determined from the interpolation conditions $p(\boldsymbol{x}) = \boldsymbol{y}$. The resulting linear system is of triangular form,

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ 1 & x_3 - x_0 & (x_3 - x_0)(x_3 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{j=0}^{n-1}(x_n - x_j) \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

and readily solved by forward substitution.

◦ The first few coefficients are

$$d_0 = y_0,\ d_1 = \frac{y_1 - d_0}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0},$$

$$d_2 = \frac{y_2 - (x_2 - x_0)d_1 - d_0}{(x_2 - x_0)(x_2 - x_1)} = \frac{y_2 - (x_2 - x_0)\dfrac{y_1 - y_0}{x_1 - x_0} - y_0}{(x_2 - x_0)(x_2 - x_1)} = \frac{\dfrac{y_2 - y_1}{x_2 - x_1} - \dfrac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}.$$

The forward substitution is efficiently expressed through the definition of divided differences

$$[y_i] = y_i,\ [y_{i+1}, y_i] = \frac{[y_{i+1}] - [y_i]}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i},\ [y_{i+2}, y_{i+1}, y_i] = \frac{[y_{i+2}, y_{i+1}] - [y_{i+1}, y_i]}{x_{i+2} - x_i},$$

or in general, the k^{th} divided difference

$$[y_{i+k}, y_{i+k-1}, \ldots, y_i] = \frac{[y_{i+k}, y_{i+k-1}, \ldots, y_{i+1}] - [y_{i+k-1}, y_{i+k-1}, \ldots, y_i]}{x_{i+k} - x_i},$$

given in terms of the $(k-1)$ divided differences. The forward substitution computations are conveniently organized in a table, useful both for hand computation and also for code implementation.

i	x_i	$[y_i]$	$[y_i, y_{i-1}]$	$[y_i, y_{i-1}, y_{i-2}]$		
0	x_0	y_0	$-$	$-$		
1	x_1	y_1	$\frac{y_1 - y_0}{x_1 - x_0}$	$-$		
2	x_2	y_2	$\frac{y_2 - y_1}{x_2 - x_1}$	$\frac{[y_2, y_1] - [y_1, y_0]}{x_2 - x_0}$		$\ddots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$			
n	x_n	y_n	$\frac{y_n - y_{n-1}}{x_n - x_{n-1}}$	$\frac{[y_n, y_{n-1}] - [y_{n-1}, y_{n-2}]}{x_n - x_{n-2}}$	$\ldots$	$\frac{[y_n, \ldots, y_1] - [y_{n-1}, \ldots, y_0]}{x_n - x_0}$

Table 2. Table of divided differences. The Newton basis coefficients $\boldsymbol{d}$ are the diagonal terms.

◦ **Algorithm (Forward substitution, Newton coefficients)**

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}$

Output: $\boldsymbol{d} \in \mathbb{R}^{n+1}$
$\boldsymbol{d} = \boldsymbol{y}$
for $i = 1$ to n
  for $j = n$ downto i
    $d_j = (d_j - d_{j-1}) \,/\, (x_j - x_{j-i})$
  end
end

The above algorithm requires only $\mathcal{O}(n)$ operations, and the Newton form of the interpolating polynomial

$$p(t) = [y_0] \cdot 1 + [y_1, y_0] \cdot (t - x_0) + [y_1, y_0] \cdot (t - x_0)(t - x_1) + \cdots + +[y_n, y_{n-1}, \ldots, y_0] \cdot (t - x_0) \cdot (t - x_1) \cdot \ldots \cdot (t - x_{n-1}),$$

can also be evaluated in $\mathcal{O}(n)$ operations

◦ **Algorithm (Newton polynomial evaluation)**

Input: $\boldsymbol{x}, \boldsymbol{d} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
Output: $p(t) = d_0 + d_1 t + \cdots + d_n t^n$
$p = d_0$; $r = 1$
for $k = 1$ to n
  $r = r \cdot (t - x_{k-1})$
  $p = p + d_k \cdot r$
end
return p

◦

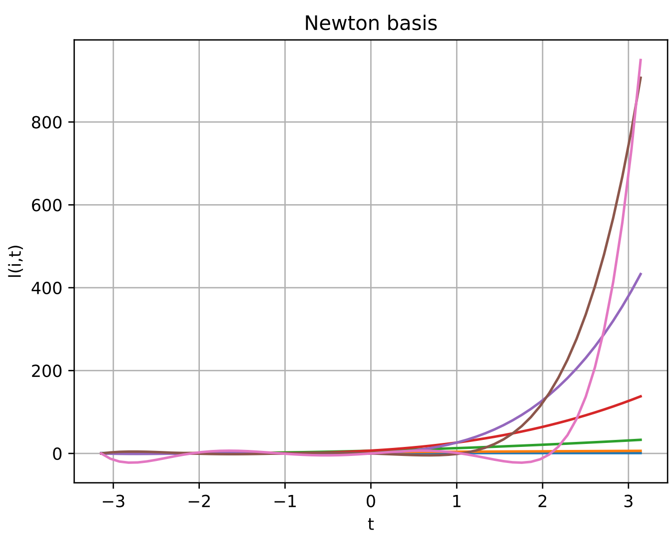


Figure 3. Newton basis for $n = 6$ for $\sin(x)$ over interval $[-\pi, \pi]$