# Introduction to Computational Mathematics (AMATH 242/CS 371)

**University of Waterloo**
**Winter 2019**

# *Interpolation*

Results of experiments are usually described by discrete data. Working with these data and using them in case we need derivative is not easy. In this case it is good to define a continuous function that not only follows the trend but also passes through the given points. This continuous function interpolates all the data points and we can use it to determine the value of output at some inputs which are not in the set of given data. These points could be either between data or even beyond the domain. We can also integrate, differentiate and plot the obtained continuous function

# Interpolation

**Problem:** $(n+1)$ discrete points $\{(x_i, f_i)\}_{i=0}^{n}$ are given with $x_i \neq x_j$ for $i \neq j$.
**Objective:** defining a continuous function that interpolates the data: $y(x_i) = f_i$ for $0 \leq i \leq n$
One useful and well-known class of functions mapping the set of real-numbers to itself is the algebraic polynomials:

$$y_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

One reason for its importance is that they uniformly approximate continuous functions. By this we mean that given any function, defined and continuous on a closed and bounded interval, there exists a polynomial that is as "close" to the given function as desired. This result is expressed precisely in the Weierstrass Approximation Theorem.

## Weierstrass Approximation Theorem

If $f(x)$ is a continuous function on $[a, b]$ for each $\epsilon > 0$, $\exists$ a polynomial $y(x)$ such that $|f(x) - y(x)| < \epsilon$ for all $x \in [a, b]$.

Another reason to choose polynomial function for interpolation is that they are easy to be integrated and differentiated.
To find the interpolation function, we should find the unknown parameters i.e. coefficient of $y_n(x)$.

**Monomial Basis:**

To interpolate $n + 1$ data points using polynomials, we denote $\mathbb{Y}_n$ as a set of all polynomials of degree at most $n$. With addition and scalar multiplication defined, $\mathbb{Y}_n$ forms a vector space with basis $\phi_j = x^j$, $j = 0, \cdots, n$ for which a given polynomial $y_n \in \mathbb{Y}_n$ has the form $y_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$.

In the monomial base, $\vec{a}$ of coefficients of the polynomial interpolating the data points $(x_i, f_i)$, $i = 1, 2, \cdots, n$ is given by the $(n+1) \times (n+1)$ linear system $V\vec{a} = \vec{f}$

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}$$

A matrix in this form, whose columns are successive powers of some independent variable $x$ is called **Vandermond Matrix**.

**Proposition:**

$$\begin{aligned} det(V) \quad &= (x_1 - x_0) \\ &(x_2 - x_0)(x_2 - x_1) \\ &\vdots \\ &(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1}) \\ &= \prod_{0 \le i \le j \le n} (x_j - x_i) \end{aligned}$$

**Question:** When does the resulted linear system have a unique solution? To have the unique solution, $det(V) \neq 0$. So for $i \neq j$, if $x_i \neq x_j$ then $det(V) \neq 0$ and we can always obtain a polynomial that interpolates the given points:

**Theorem**

The interpolating polynomial $y_n(x)$ exists and is unique

**Example:**

• To illustrate the monomial basis, determine the polynomial of degree two interpolating (-2,-27), (0,-1), (1,0).

$y_2(x) = a_0 + a_1 x + a_2 x^2$

$$\begin{pmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -27 \\ -1 \\ 0 \end{pmatrix}$$

Solving by G.E gives $\vec{a} = \begin{pmatrix} -1 \\ 5 \\ 4 \end{pmatrix}$ and $y_2(x) = -1 + 5x - 4x^2$

Challenges in solving $V\vec{a} = \vec{f}$:

- Computational complexity to solve the linear system using standard solver (G.E) is of order $O(n^3)$, solvers with complexity of order $O(n^2)$ use other polynomial representation.
- V is a very ill-conditioned matrix, since $\kappa_2(V)$ grows faster than exponentially as a function of n.

# *Interpolation*

By using different basis, both the conditioning of the linear system and the amount of computational work required to solve the linear system can be improved. A change of basis still gives the same unique interpolation polynomial:

$$p_n = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$
$$q_n = a'_0 + a'_1 x + a'_2 x^2 + \cdots + a'_n x^n$$

Then, $p_n - q_n = (a_0 - a'_0) + (a_1 - a'_1)x + \cdots + (a_n - a'_n)x^n$ which is of order at most $n$ and is zero at $(n+1$ data points. But a polynomial of degree $n$ has at most $n$ zeros unless it is the zero polynomial. So, $p_n - q_n = 0 \Rightarrow p_n = q_n$. What does change is the representation of that polynomial in a different basis.

Conditioning of monomial basis can be improved by shifting and scaling the independent variable: $\phi_j(x) = \left(\frac{x-c}{d}\right)^j$, where for example $c = \frac{x_1 + x_n}{2}$ and $d = \frac{x_n - x_1}{2}$. So, the new independent variable lies in [-1,1]. This transformation also helps avoid overflow and harmful underflow in computing the entries of the basis matrix or evaluating the resulting polynomial. However, the monomial basis even after shifting is poorly conditioned, and we must seek alternatives.

**Lagrange Interpolation:**

As monomial basis is not very efficient, we need a stronger and more efficient alternative.

**Problem:** Determining a polynomial of degree one that passes through distinct points $(x_0, y_0)$ and $(x_1, y_1)$.

This polynomial is in the form of $y_1(x) = a_0 L_0 + a_1 L_1$. Defining the functions $L_0 = \dfrac{x - x_1}{x_0 - x_1}$ and $L_1 = \dfrac{x - x_0}{x_1 - x_0}$ and considering $y_1(x_0) = f_0$ and $y_1(x_1) = f_1$, $y_1(x)$ takes the form $y_1(x) = L_0(x)f_0 + L_1(x)f_1$ and is called linear Lagrange polynomial and $L_0(x)$ and $L_1(x)$ are Lagrange basis. Note that $L_0(x_0) = 1$, $L_1(x_1) = 1$, $L_0(x_1) = 0$ and $L_1(x_0) = 0$.

**Example:**

• Determine the linear Lagrange interpolating polynomial that passes through (2,4) and (5,1).

$L_0(x) = \dfrac{5 - x}{3}$, $L_1(x) = \dfrac{x - 2}{3}$ so $y(x) = 6 - x$.

To generalize the concept of linear interpolation, consider the construction of a polynomial of degree at most (n) that passes through (n+1) points $(x_0, f_0), (x_1, f_1), \cdots, (x_n, f_n)$. In this case, we first construct $L_{ni}(x)$, $i = 0, 1, \cdots, n$ with the property that $L_{ni}(x_i) = 1$, $L_{ni}(x_j) = 1, \ i \neq j$. To satisfy $L_{ni}(x_j) = 0, \ i \neq j$ numerator of $L_{ni}(x)$ should contain $(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)$. To satisfy $L_{ni}(x_i) = 1$, denominator of $L_{ni}(x)$ must be the same but should be evaluated at $x_i$

$$L_{ni}(x) = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

## Theorem

If $x_0, x_1, \cdots, x_n$ are (n+1) distinct numbers and $f$ is a function whose values are given at these numbers, then a unique polynomial $y_n(x)$ of degree at most (n) exists with $f(x_i) = y_n(x_i)$, $i = 0, \cdots, n$. This polynomial is given by

$$y_n(x) = L_{n0}(x)f_0 + L_{n1}(x)f_1 + \cdots + L_{nn}(x)f_n = \sum_{i=0}^{n} L_{ni} f_i$$

where for each (i)

$$L_{ni}(x) = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)} = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

This is called Lagrange polynomial and for simplicity we show $L_{ni}$ by $L_i$.

**Lagrange Basis:** Monomial basis or standard basis is the basis to form $\mathbb{Y}_n(x)$ as a vector space that contains all polynomial of degree at most (n): $\mathbb{Y}_n(x) = \{y_n(x) | y_n(x) \text{is a polynomial of degree} \leq n\}$. Lagrange polynomials form a different basis for vector space $\mathbb{Y}_n(x)$: $B' = \{L_0, L_1, \cdots Ł_N\}$ and we can write any polynomial $y_n(x)$ as a linear combination of the Lagrange polynomials: $y_n(x) = \sum_{i=0}^{n} L_i(x) f_i$.

**Example:**

• (a). Use $x_0 = 2$, $x_1 = 2.75$ and $x_2 = 4$ to find the second Lagrange interpolating polynomial for $f(x) = \frac{1}{x}$

(b). Use this polynomial to approximate $f(3) = \frac{1}{3}$.

(a). $L_0 = \frac{2}{3}(x - 2.75)(x - 4)$, $L_1 = -\frac{16}{15}(x - 2)(x - 4)$,
$L_2 = \frac{2}{5}(x - 2)(x - 2.75)$. Then $y_2(x) = \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}$.
(b). $f(3) \approx y_2(3) \approx 0.329$.

**Theorem**

Suppose $x_0, x_1, \cdots, x_n$ are distinct numbers in the interval $[a, b]$ and $f$ is a continuous and differentiable function. Then for each $x$ in $[a, b]$, $\exists$, $c$ (unknown) between $x_0, x_1, \cdots, x_n$ and hence in $(a, b)$ with

$$f(x) = y_n(x) + \frac{f^{n+1}(c)}{(n+1)!}(x - x_0) \cdots (x - x_n)$$

where $y_n(x)$ is Lagrange interpolating polynomial.

**Example:**

• In the previous example, determine the error form for Lagrange polynomial and the maximum error when the polynomial is used to approximate f(x) for $x \in [2, 4]$.

Error$=\frac{f(3)(c)}{3!}(x - x_0)(x - x_1)(x - x_2) =$ and $|\text{Error}| \leq \frac{9}{256}$.

**Hermite Interpolation:** If we have the values of a function and its derivative at interpolation points, then interpolating polynomial is **Hermite Polynomial**.
Hermite polynomial agrees with $f$ at $x_0, x_1, \cdots, x_n$ and also with $f'$. Then Hermite polynomial has the same shape as the function at $(x_i, f_i)$ in the sense that the tangent lines to the polynomial and function agree.

**Definition:** Given $\{(x_i, f_i, f_i')\}_{i=0}^n$, the Hermite interpolating polynomial is the polynomial $y(x)$ of degree $2n+1$ which satisfies $y(x_i) = f_i$ and $y'(x_i) = f_i'$. Since there are $2n+2$ conditions, there must be $2n+2$ polynomial coefficients in the minimal degree interpolating polynomial and hence $y(x)$ has degree $2n+1$:

$$H_{2n+1}(x) = \sum_{i=0}^{n} f(x_i) H_{n,i}(x) + \sum_{i=0}^{n} f'(x_i) \hat{H}_{n,i}(x)$$

where $H_{n,i} = \left[1 - 2(x - x_i)L'_{ni}(x_i)\right] L^2_{ni}(x)$ and $\hat{H}_{n,i} = (x - x_i)L^2_{ni}(x)$ and $L_{ni}(x)$ is the i-th Lagrange coefficient polynomial of degree (n). Moreover,

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n + 2)!} f^{2n+2}(c), \ c \in (a, b)$$

**Example:**
Use the Hermite polynomial that agrees with the given data to find an approximation of $f(1.5)$.

| $k$ | $x_k$ | $f(x_k)$ | $f'(x_k)$ |
|---|---|---|---|
| 0 | 1.3 | 0.6200860 | -0.5220232 |
| 1 | 1.6 | 0.4554022 | -0.5698959 |
| 2 | 1.9 | 0.2818186 | -0.5811571 |

$H_5(1.5) = 0.5118277$

**Second method to find Hermite polynomial for two points(cubic Hermite):** Similar to the idea of the Lagrange form, we can write the Hermite polynomial in a form that makes solving for the polynomial coefficient easier.

$$y(x) = a + b(x - x_0) + c(x - x_0)^2 + d(x - x_0)^2(x - x_1)$$
$$y'(x) = b + 2c(x - x_0) + 2d(x - x_0)(x - x_1) + d(x - x_0)^2$$

$$y(x_0) = a, \quad y(x_1) = a + b(x_1 - x_0) + c(x_1 - x_0)^2$$
$$y'(x_0) = b, \quad y'(x_1) = b + 2c(x_1 - x_0) + d(x_1 - x_0)^2$$

$\Longrightarrow$

$$c = \frac{1}{(x_1 - x_0)^2}\left[f_1 - f_0 - f_0'(x_1 - x_0)\right]$$
$$d = \frac{1}{(x_1 - x_0)^2}\left[f_1' - f_0' - \frac{2}{(x_1 - x_0)}\left(f_1 - f_0 - f_0'(x_1 - x_0)\right)\right]$$

Basis for this example (having two points) is $\{1, (x - x_0), (x - x_0)^2, (x - x_0)^2(x - x_1)\}$. This basis is chosen such that the calculations are simplified.

**Piecewise Polynomial Interpolation:** One drawback of polynomial interpolating is that high-degree polynomial can oscillate. To solve this issue, called Runge's phenomenon, we can divide the whole interval into a collection of subintervals and generally construct different approximating polynomial on each sub-interval. This is called piecewise polynomial approximation. Other approaches to fix the issue are least-square fitting in which data points are close to y(x) instead of on y(x), and choosing interpolating points which are distributed more densely towards the edges. In piecewise linear interpolation, each interval consists of two consequent points and a line segment interpolates the function in each interval, since each interval contains two points on $I_i = [x_{i-1}, x_i]$. We define

$$y_i(x) = \frac{x - x_i}{x_{i-1} - x_i} f_{i-1} + \frac{x - x_i}{x_i - x_{i-1}} f_i, \quad 1 \leq i \leq n.$$

**Problem of this interpolation:** Resulted function is not smooth at the interpolation points. An alternative procedure is using a piecewise polynomial of Hermite types. So, if $f$ and $f'$ are known at $x_0, x_1, \cdots, x_n$ a cubic Hermite polynomial can be used on $[x_0, x_1]$, $[x_1, x_2], \cdots, [x_{n-1}, x_n]$ to obtain a function that has a continuous derivative on $[x_0, x_n]$. So, $H_3$ should be computed. As Lagrange polynomials are of first degree, $H_3$ can be found easily. However, to use Hermite piecewise polynomials for general interpolation, we need to know thee derivative of the function which is frequently unavailable. So, it is good to find piecewise polynomials that require no specific derivative information, except perhaps at the endpoints of the interval on which the function is being approximated.

**Cubic Spline:** The most common piecewise-polynomial approximation uses cubic polynomials between each successive pair of nodes and is called **cubic spline interpolation**.

**Definition.** Given a function f defined on [a,b] and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant $S$ for f is a function that satisfies the following conditions:

- $S(x)$ is a piecewise polynomial of degree $k$ in each interval $I_i = [x_{i-1}, x_i]$ and $S_i(x)$ is defined as the restriction of $S(x)$ to $I_i,\ 1 \leq i \leq n$
- $S_i(x_{i-1}) = f_{i-1}$ and $S_i(x_i) = f_i$ in $I_i,\ 1 \leq i \leq n$ so $S_i(x_i) = S_{i-1}(x_i)$
- For each interior point $j$, there are $(k-1)$ smoothness conditions

$$S_j'(x_j) = S_{j+1}'(x_j)$$
$$S''_j(x_j) = S''_{j+1}(x_j)$$
$$\vdots$$
$$S_j^{(k-1)}(x_j) = S_{j+1}^{(k-1)}(x_j)$$

where $0 < j < n$. Under this definition, there will be $n$ interval and $(k+1)$ condition for each polynomial. In total, this gives $n(k+1)$ unknowns. From the second condition, we have $2n$ interpolation condition and $(k-1)(n-1)$ smoothness condition. So, in total $2n + kn - k - n + 1$ conditions exist. Thus, we need to impose $(k-1)$ extra conditions which are supplied by boundary conditions at $x_0$ to $x_n$.

Three possible boundary conditions:

- Free boundary: $S''(x_0) = 0$ and $S''(x_n) = 0$. A cubic spline with this boundary condition is known as a natural cubic spline
- Clamped boundary: we specify the first derivatives at the ends by choosing constants $f_0'$ and $f_n'$. Then $S'(x_0) = f_0'$ and $S'(x_n) = f_n'$
- Periodic boundary: If $f_0 = f_n$, we impose that the first and second derivative of the first and last polynomials are equal to $x_0$ and $x_n$. So, $S_1'(x_0) = S_n'(x_n)$ and $S''_1(x_0) = S''_n(x_n)$

By imposing these boundary conditions, we produce $(nk + n) \times (nk + n)$ linear system that may be uniquely solved for the coefficients of $S_i(x), \quad 1 \leq i \leq n$.

In general clamped boundary condition leads to more accurate approximation because they include more information about the function. However, to use this boundary condition the values of derivative at the endpoints should be provided.

**Example:**

• Construct a natural cubic spline that passes through the points (1,2), (2,3) and (3,5).

This spline consists of two cubics. The first one is for [1,2] and the second one is for [2,3]. So, 8 constants should be determined which need 8 conditions.

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3 & x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3 & x \in [1,2] \end{cases}$$

**Construction of a cubic spline:** a spline defined on an interval that is divided into $n$ subintervals will require determining $4n$ constants. To construct the cubic spline interpolant for a given function $f$ , the conditions in the definition are applied to the cubic polynomials

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

For each $j = 0, \cdots, n-1$, $S_j(x_j) = a_j = f(x_j)$, so $a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3$ for $j = 0, \cdots, n-1$. If we define $x_{j+1} - x_j = h_j$ for $j = 0, \cdots, n-1$ and $a_n = f(x_n)$, then

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \qquad (1)$$

for $j = 0, \cdots, n-1$. In a similar way, define $b_n = S'(x_n)$ and observe that $S_j'(x) = b_j + 2c_j(x_{j+1} - x_j) + 3d_j(x_{j+1} - x_j)^2$. As $S_j'(x_j) = b_j$ for $j = 0, \cdots, n-1$ then

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \qquad (2)$$

for $j = 0, \cdots, n-1$. Another relation between the coefficients of $S_j$ is obtained by defining $c_n = \dfrac{S''(x_n)}{2}$ then

$$c_{j+1} = c_j + 3d_j h_j \qquad (3)$$

for $j = 0, \cdots, n-1$ and comes from the smoothing condition. Defining $d_j$ from (3) and substituting in (2) and (1) gives

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \qquad (*)$$

and

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \qquad (**)$$

Finding $b_j$ from $(*)$ yields $b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})$ and therefore $b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j)$. Substituting in $(**)$ and reducing the index by one gives:

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}) \qquad (***)$$

for $j = 1, \cdots, n-1$. $a_j$ and $h_j$ are given and $c_j$ are unknown parameters. By knowing $c_j$, we then find $b_j$ $b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})$

and $d_j$ from (3) easily. Note that to obtain $c_j$ we need to use the imposed boundary condition.

**Example:**

• Use the data points $(0, 1), (1, e), (2, e^2)$ and $(3, e^3)$ to form a natural spline S(x) that approximates $f(x) = e^x$.

• Find clamped splines that passes through the points (1,2), (2,3) and (3,5) with $S'(1) = 2$ and $S'(3) = 1$.