INTERPOLATION

Raja Damanik, M.Sc.

Faculty of Computer Science
University of Indonesia

Motivation

$F: \mathbb{R} \rightarrow \mathbb{R}$ is unknown but:

$$F(0) = 1$$
$$F(3) = 7$$

Guess $F(2)$.

Motivation

$F: \mathbb{R} \to \mathbb{R}$ is unknown but:

$$F(0) = 1$$
$$F(3) = 7$$

Guess $F(2)$.

- A simple guess: $F(X) = 2X + 1$, giving $F(2) = 5$

Motivation

$F: \mathbb{R} \to \mathbb{R}$ is unknown but:

$$F(0) = 1$$
$$F(3) = 7$$

Guess $F(2)$.

- A simple guess: $F(X) = 2X + 1$, giving $F(2) = 5$
- Caution: infinitely many fitting functions!

  $F(X) = X^2 - X + 1$ $\quad$ $F(X) = X^{10} + 1 - 3X^9 + \frac{7}{3}X$ $\quad$ $F(X) = e^{X \ln 7/3}$
- Occham's razor: Choose simplest explanation over competing hypothesis.

Motivation

- Similar problem but… more data.

$F: \mathbb{R} \to \mathbb{R}$ is unknown but:

$$F(0) = 1$$
$$F(3) = 7$$
$$F(5) = 100$$
$$F(7) = 0$$

Guess $F(2)$. ☺

Interpolation

- Interpolation: method of constructing new data points within the range of a discrete set of known data points.

- Many possible settings, depends on the our knowledge of F:
  - Only know data points $\left(x_i, F(x_i)\right)$?
  - Do we know derivative F' or even F'' at point x_i?
  - Do we know if F periodic?
  - Do we know if F an even / odd function?
  - Do we know if F probably exponential? Or logarithmic?

Polynomial Interpolation (Setting)

- Let $x_0 < x_1 < \cdots < x_n$ be $n+1$ real numbers.
- Given data points

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

find a <u>polynomial</u> P that fits these $n+1$ data:

$$P(x_0) = y_0$$
$$P(x_1) = y_1$$
$$\ldots$$
$$P(x_n) = y_n$$

Polynomial: $P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$

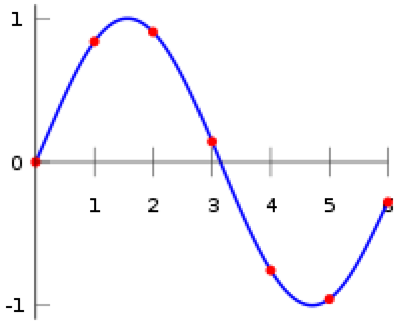
Polynomial Interpolation (Setting)

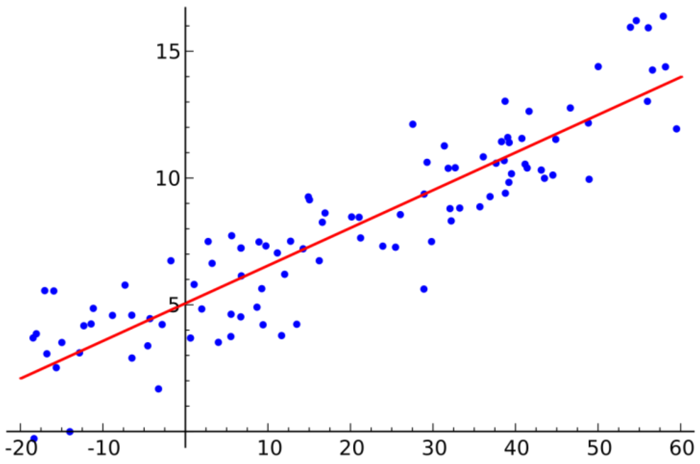
- THEOREM. To fit $n + 1$ data points, polynomial of degree n is sufficient. ☺
- We can construct a polynomial P of degree $\leq n$ that fits $n + 1$ data.

- Analogy:
  - Two points determine a line (polynomial degree 1)
  - Three points determine a quadratic function (polynomial degree 2)
  - etc

Caution

- Interpolation $\neq$ Regression
- Interpolation **<u>must perfectly fits the data</u>**, while regression may allow error.
- Also the reason why in interpolation we assume $x_0 < x_1 < \cdots < x_n$.





Interpolation Problem Setting

- Let $x_0 < x_1 < \cdots < x_n$ be $n + 1$ real numbers.
- Given data points

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

find a polynomial P that fits these $n + 1$ data.

- THEOREM: Degree n is sufficient. ☺
- We can construct a polynomial P of degree $\leq n$ that fits $n + 1$ data.

Polynomial Interpolation

- Three simple methods giving same result (mathematically):
  - Vandermonde matrix
  - Lagrange’s polynomial
  - Newton’s polynomial

Vandermonde Matrix

Given $n + 1$ data points

$$(x_i, y_i) \qquad\qquad i = 0,1, \dots, n.$$

Try to find coefficients $a_n, \dots, a_0$ in the n-degree polynomial

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

such that $P(x_i) = y_i$.

Vandermonde Matrix

Giving us linear system of $n+1$ equations

$$\begin{gathered}
a_n x_0^n + a_{n-1} x_0^{n-1} + \cdots + a_1 x_0 + a_0 = y_0 \\
a_n x_1^n + a_{n-1} x_1^{n-1} + \cdots + a_1 x_1 + a_0 = y_1 \\
\dots \\
a_n x_n^n + a_{n-1} x_n^{n-1} + \cdots + a_1 x_n + a_0 = y_n
\end{gathered}$$

and $n+1$ unknowns. Perfect!

Vandermonde Matrix

Solve linear system

$$\begin{bmatrix} x_0^n & x_0^{n-1} & \cdots & 1 \\ x_1^n & x_1^{n-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^n & x_n^{n-1} & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Vandermonde matrix is invertible iff $x_i \neq x_j$ for all $i \neq j$.

Lagrange Polynomial

Simple example:

Find a polynomial of degree 3 that fits 4 data points

$$
\begin{gathered}
(1,100) \\
(2,200) \\
(3,300) \\
(5,900)
\end{gathered}
$$

Lagrange Polynomial

- What does this polynomial do?

$$\frac{(x-2)(x-3)(x-5)}{(1-2)(1-3)(1-5)}$$

Lagrange Polynomial

- What does this polynomial do?

$$\frac{(x-2)(x-3)(x-5)}{(1-2)(1-3)(1-5)}$$

$$\frac{(x-1)(x-3)(x-5)}{(2-1)(2-3)(2-5)}$$

$$\frac{(x-1)(x-2)(x-5)}{(3-1)(3-2)(3-5)}$$

$$\frac{(x-1)(x-2)(x-3)}{(5-1)(5-2)(5-3)}$$

Lagrange Polynomial

- What does this polynomial do?

$$
100\frac{(x-2)(x-3)(x-5)}{(1-2)(1-3)(1-5)}
$$

$$
+200\frac{(x-1)(x-3)(x-5)}{(2-1)(2-3)(2-5)}
$$

$$
+300\frac{(x-1)(x-2)(x-5)}{(3-1)(3-2)(3-5)}
$$

$$
+900\frac{(x-1)(x-2)(x-3)}{(5-1)(5-2)(5-3)}
$$

Lagrange Polynomial

- Lagrange's polynomials use Lagrange basis

$$\ell_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

- Property: $\ell_i(x_i) = 1$ and $\ell_i(x_j) = 0$.
- Lagrange's polynomial $P(x)$ construction for $n + 1$ data points (x_i, y_i) for $i = 0, \dots, n$

$$P(x) = \sum_{i=0}^{n} y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Exercise

- Use Vandermonde and Lagrange Polynomial to find a quadratic polynomial that fits three data points

$$(0,1) \qquad (1,2) \qquad (2,5).$$

Newton's Polynomial

- Instead of using Lagrange's basis, use Newton's basis

$$t_0(x) = 1$$
$$t_1(x) = (x - x_0)$$
$$t_2(x) = (x - x_0)(x - x_1)$$
$$\dots$$
$$t_n(x) = (x - x_0)(x - x_1)\dots(x - x_{n-1})$$

- In general: For $i = 0,1,\dots,n$,

$$t_i(x) = \prod_{j=0}^{i-1}(x - x_j)$$

Newton’s polynomial

- Newton’s polynomial:

$$p(x) = a_0 t_0(x) + a_1 t_1(x) + \cdots + a_n t_n(x)$$

- Question: How to find coefficients $a_0, a_1, \ldots, a_n$?

Newton's polynomial

We want $p(x_i) = y_i$ for each $i = 0,1,\ldots,n$.

This gives us triangular system of linear equations $T\boldsymbol{a} = \boldsymbol{y}$ with

$$T_{ij} = \begin{cases} 0 & i < j \\ t_j(x_i) & i \geq j \end{cases}$$

and

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix}.$$

Example

- Find Newton's polynomial for data points

$$(0,1) \quad (1,2) \quad (2,5).$$

Newton's polynomial: Divided Difference

- Alternative method for computing $a_0, a_1, \dots, a_n$.

- Define function $f[\cdot]$, and we aim to compute

$$a_i = f[x_0, x_1, \dots, x_i].$$

- Divided difference $f[\cdot]$ is defined recursively:

$$f[x_i] = y_i \qquad i = 0, \dots, n$$

$$f[x_i, x_{i+1}, \dots x_{i+j}] = \frac{f[x_{i+1}, \dots, x_{i+j}] - f[x_i, x_{i+1}, \dots, x_{i+1-1}]}{x_{i+j} - x_i}$$

Hermite’s Interpolation

- Using more information than just data points: 1st, 2nd, ..., k-the derivative at certain point x_i. In the divided difference table, use $k+1$ copies of x_i.

- When computing $f[x_i, \dots, x_i]$ (where the x_i occurs $j+1$ times), use the j-th derivative at x_i.

Example

- Use Divided Difference to Compute Newton's polynomial for the data points

$$(0,1) \qquad (1,2) \qquad (2,5)$$

- What if we require the first at 1 is 0? Find the Hermite's interpolation.

Interpolation Error

- Suppose we have a function $f: \mathbb{R} \to \mathbb{R}$ and we interpolates f using $n+1$ data points $\left(x_i, f(x_i)\right)$ for $i = 0, \dots, n$.

- Let $p(x)$ be the interpolation polynomial.

- Interpolation error $e(x) = p(x) - f(x)$. One can show that

$$e(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)\,.$$

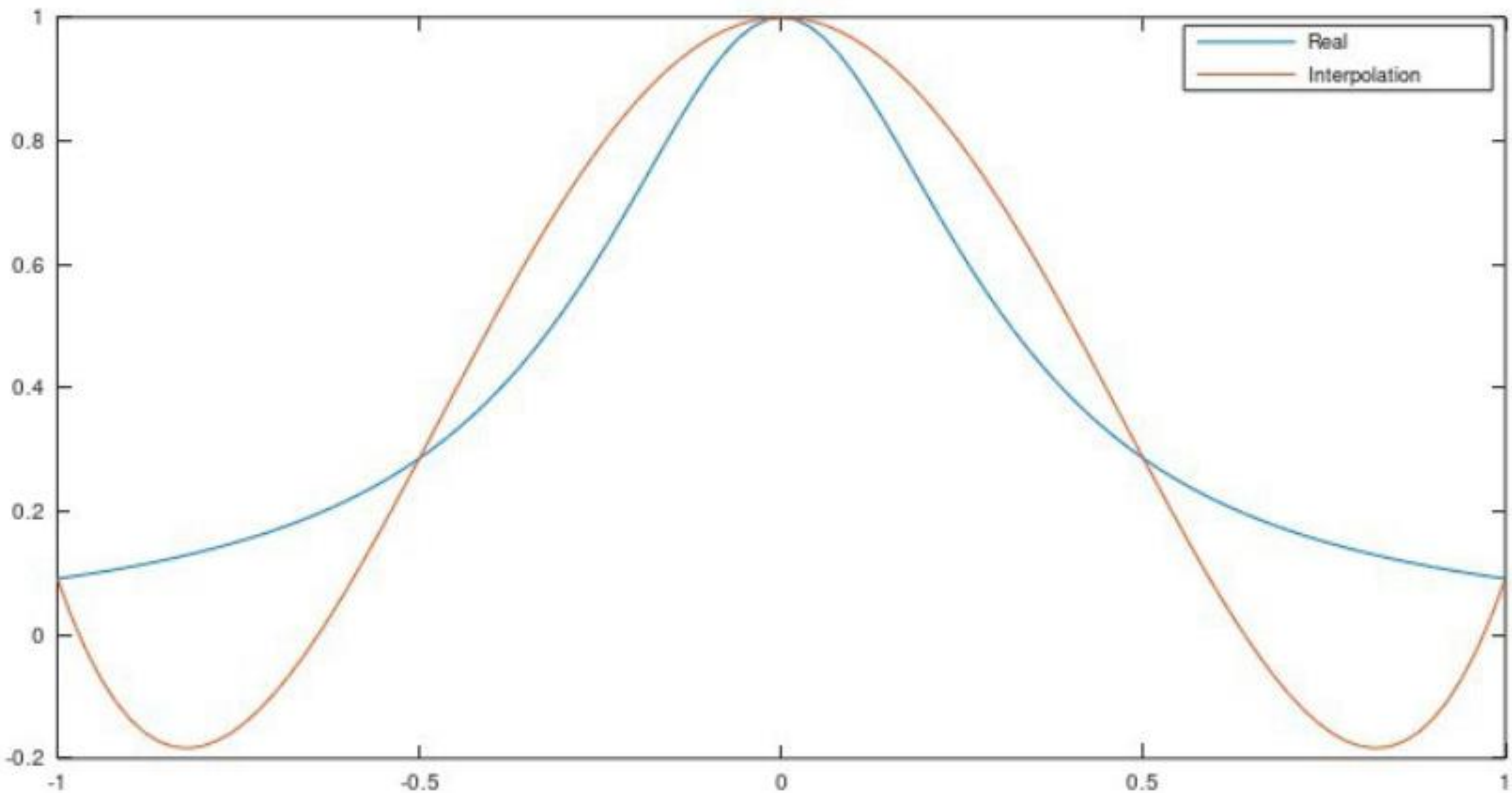
Runge's Phenomenon

- Given a function $f(x) = \frac{1}{1+x^2}$.

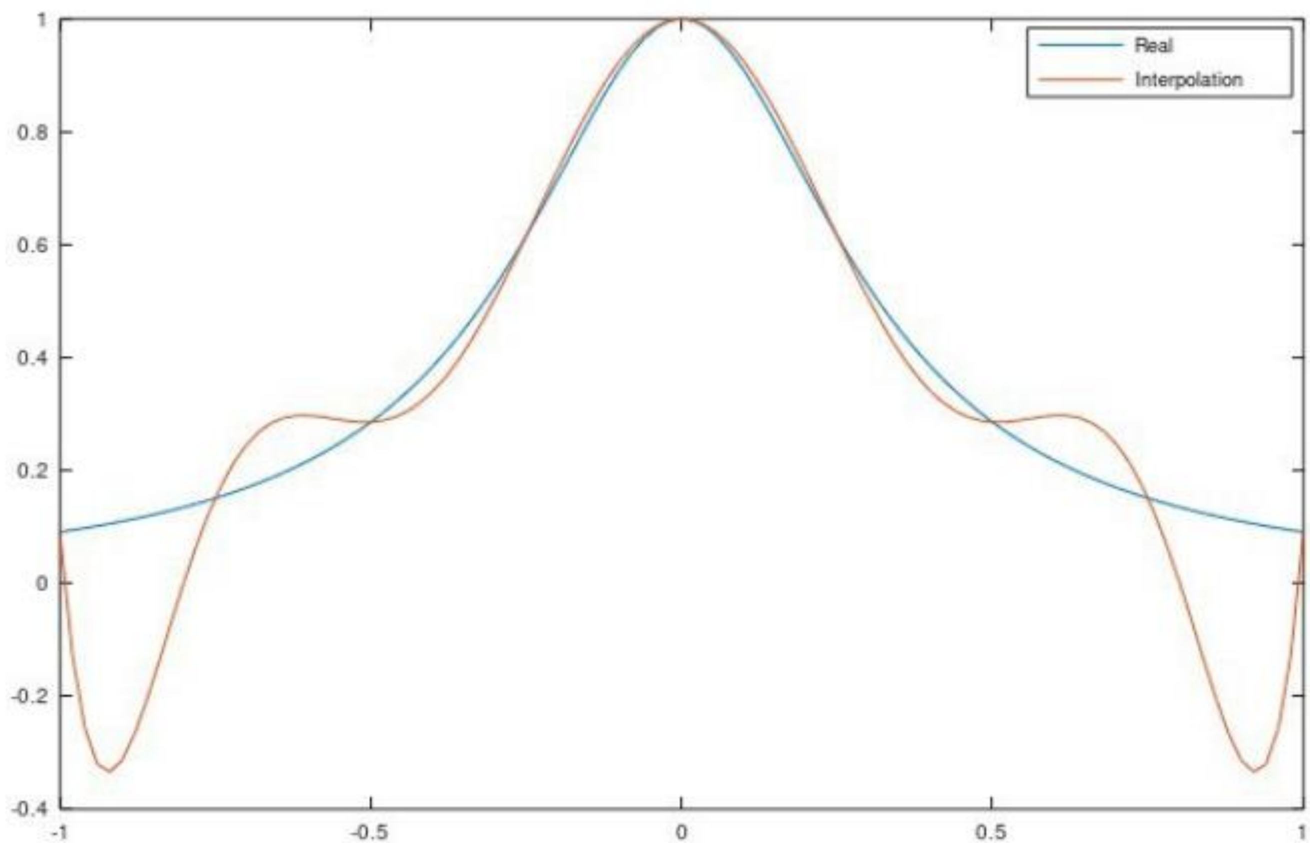
- We want to interpolate this function using points on the curve of this function at $x_0, x_1, \dots, x_n$ that are equidistant on interval $[-1,1]$.

- Increasing n, we expect: the polynomial interpolation fits the function even better.

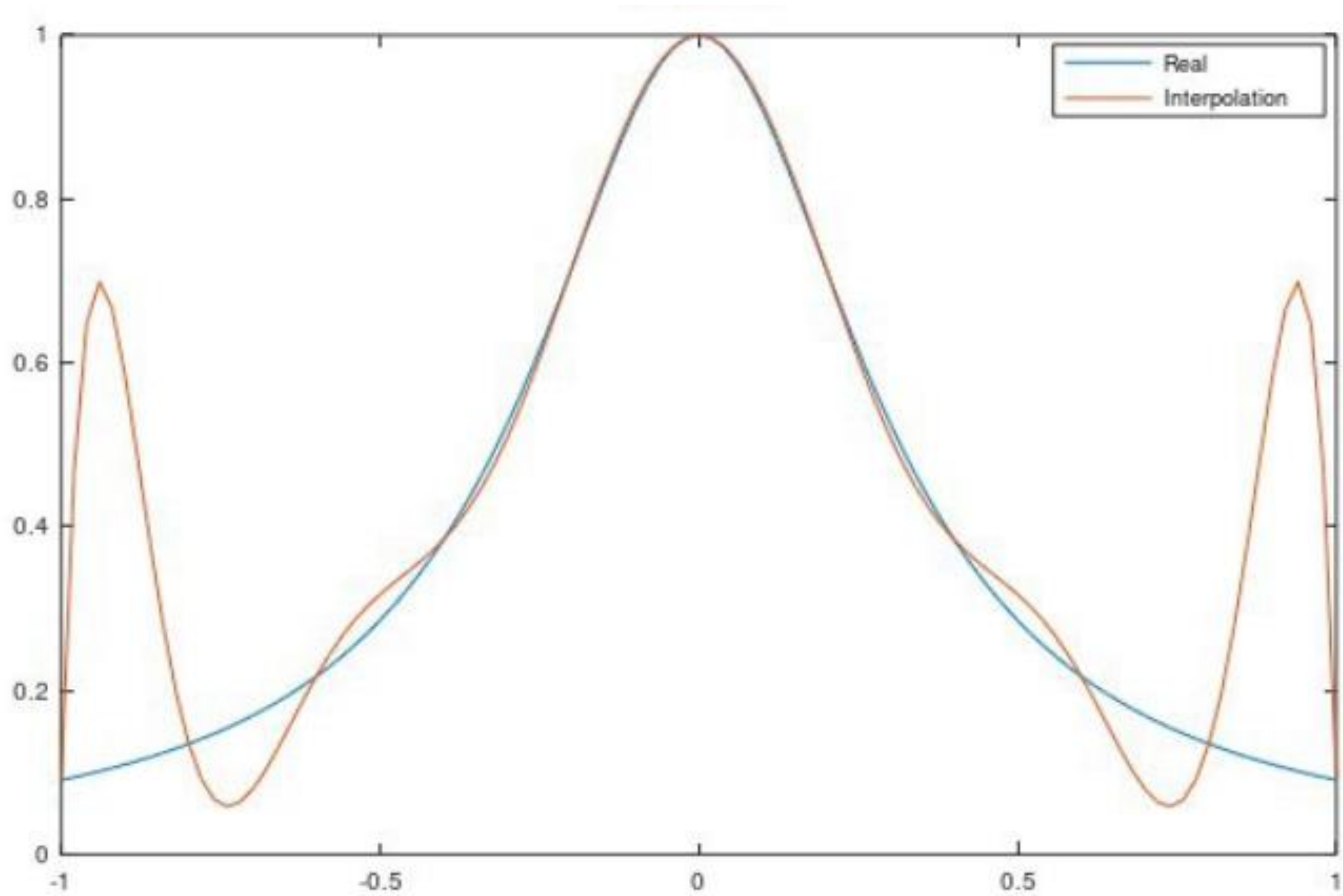
Runge’s Phenomenon



Runge’s Phenomenon



Runge’s Phenomenon



Runge’s Phenomenon

- What we want from interpolation:

More data points, more accuracy!

- This is <u>NOT TRUE</u> with polynomial interpolation.

Increasing the number data points → Increasing the degree of the polynomial → More oscillations.

SPLINE INTERPOLATION

- We use piece-wise polynomial interpolation.
- Use different low-degree polynomials for each interval:

$$s(x) = \begin{cases} s_0(x) & x_0 \leq x \leq x_1 \\ s_1(x) & x_1 \leq x \leq x_2 \\ & \vdots \\ s_{n-1}(x) & x_{n-1} \leq x \leq x_n \end{cases}$$

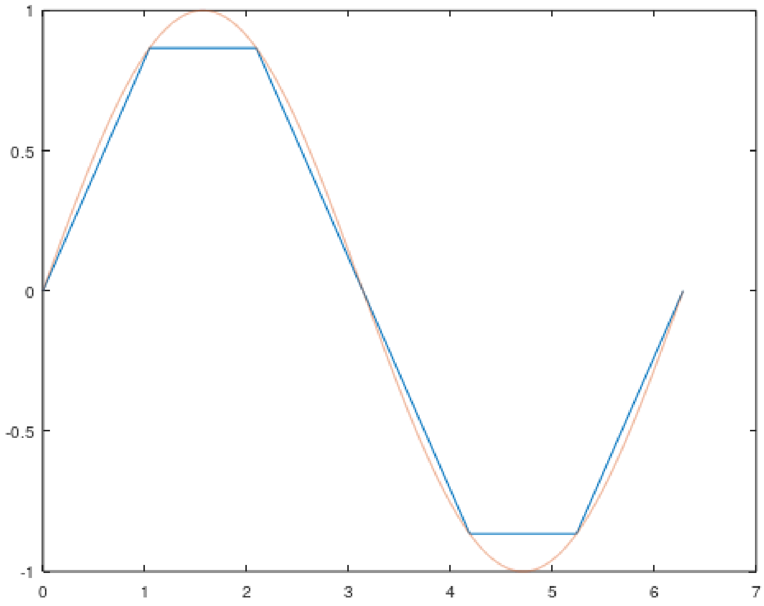
- Where $s_i(x)$ are polynomials of low-degree.

Linear Spline Interpolation

- $s_i(x)$ is a a linear function that passes through (x_i, y_i) and (y_i, y_{i+1}):

$$s_i(x) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) + y_i$$

- Efficient to compute! ☺
- No more Runge phenomenon: more data points will give more accuracy.
- $s(x)$ is continuous, but not smooth: does not look so convincing as a hypothesis of why the data points are observed!
- Can we make it smoother?



Quadratic Spline Interpolation

- Function is smoother by requiring $s'(x)$ is continous!
- Instead, we will study cubic spline which will give an even smoother function and nice structure. ☺
- Quadratic spline can be derived in similar & simpler fashion.

Cubic Spline Interpolation

- For $n + 1$ points, there will be n polynomials $s_0(x), s_1(x), \dots, s_{n-1}(x)$ with

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

- There will be $4n$ unknowns:

$$a_0, b_0, c_0, d_0, \dots, a_{n-1}, b_{n-1}, c_{n-1}, d_{n-1}$$

- We will need $4n$ requirements to determine these conditiones.

Cubic Spline Interpolation

Given $n+1$ data points $(x_0, y_0), \ldots, (x_n, y_n)$.

Requirements for $s_0(x), s_1(x), \ldots, s_{n-1}(x)$ as cubic spline:

1. Interpolate the data points:

   For $i = 0,1,\ldots,n-1$ $\quad s_i(x_i) = y_i$

2. Continuity of $s(x)$:

   For $i = 0,1,\ldots,n-1$ $\quad s_i(x_{i+1}) = y_{i+1}$

3. Continuity of $s'(x)$, to make the function smooth:

   For $i = 0,1,\ldots,n-2$ $\quad s_i'(x_{i+1}) = s_{i+1}'(x_{i+1})$

4. Continuity of $s''(x)$, to make the inflection smooth:

   For $i = 0,1,\ldots,n-2$ $\quad s_i''(x_{i+1}) = s_{i+1}''(x_{i+1})$

Cubic Spline Interpolation

- So far we only have
$$n + n + (n-1) + (n-1) = 4n - 2$$
conditions.
- Need two more conditions. There are some variations:
  - End-slope spline
  - Periodic spline
  - Not-a-knot spline
  - Natural spline: $s_0''(x_0) = 0$ and $s_{n-1}''(x_n) = 0$
- MATLAB / Octave uses not-a-knot spline.
- We will consider natural spline for the 2 additional conditions to determine $4n$ unknowns.

Cubic Spline Interpolation

- Using condition (1), we obtain

$$d_i = y_i \text{ for } i = 0,1, \dots, n-1$$

Cubic Spline Interpolation

- We introduce new variables:
  - $h_i := x_{i+1} - x_i$
  - σ_i for $i = 0,1, \dots, n$ which stands for $s_i''(x_i)$, except $\sigma_0 = \sigma_n = 0$ by natural spline condition

- Goal: We want to parameterize a_i, b_i, c_i over σ_i, and focus on finding the unknowns $\sigma_1, \dots, \sigma_{n-1}$.

Cubic Spline Interpolation

- Note that

$$s_i''(x) = 6a_i(x - x_i) + 2b_i$$

- By definition of σ_i:

$$\sigma_i = s_i''(x_i) = 2b_i$$

$$b_i = \frac{\sigma_i}{2} \qquad \text{for } i = 0,1, \dots, n-1.$$

Cubic Spline Interpolation

- Note that

$$s_i''(x) = 6a_i(x - x_i) + 2b_i$$

- By condition (4) $s_i''(x_{i+1}) = s_{i+1}''(x_{i+1})$, we obtain:

$$6a_i h_i + 2b_i = 2b_{i+1}$$

$$a_i = \frac{\sigma_{i+1} - \sigma_i}{6h_i} \qquad \text{for } i = 0,1, \dots, n-1.$$

Cubic Spline Interpolation

- Note that

$$s_i(x_i) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

- By condition (2) $s_i(x_{i+1}) = y_{i+1}$, we obtain:

$$a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i = y_{i+1}$$

and by substituting a_i, b_i, d_i, we get

$$\frac{\sigma_{i+1} - \sigma_i}{6h_i} h_i^3 + \frac{\sigma_i}{2} h_i^2 + c_i h_i + y_i = y_{i+1}$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - h_i \left(\frac{2\sigma_i + \sigma_{i+1}}{6}\right) \quad \text{for } i = 0,1, \ldots, n-1.$$

Cubic Spline Interpolation

- Note that

$$s_i'(x_i) = 3a_i(x-x_i)^2 + 2b_i(x-x_i) + c_i$$

- By condition (3) $s_i'(x_{i+1}) = s_{i+1}'(x_{i+1})$, we obtain:

$$3a_ih_i^2 + 2b_ih_i + c_i = c_{i+1}$$

and by substituting a_i, b_i, c_i, we get

$$\frac{\sigma_{i+1}-\sigma_i}{2h_i}h_i^2 + \sigma_ih_i + \frac{y_{i+1}-y_i}{h_i} - h_i\left(\frac{2\sigma_i+\sigma_{i+1}}{6}\right) = \frac{y_{i+2}-y_{i+1}}{h_{i+1}} - h_{i+1}\left(\frac{2\sigma_{i+1}+\sigma_{i+2}}{6}\right)$$

$$\frac{h_i}{6}\sigma_i + \frac{h_i+h_{i+1}}{3}\sigma_{i+1} + \frac{h_{i+1}}{6}\sigma_{i+2} = \frac{y_{i+2}-y_{i+1}}{h_{i+1}} - \frac{y_{i+1}-y_i}{h_i}$$

Cubic Spline Interpolation

- System equation of $n-1$ unknowns $\sigma_1, \dots, \sigma_{n-1}$ and $n-1$ equations

$$\left(\frac{h_i}{6}\right)\sigma_i + \left(\frac{h_i+h_{i+1}}{3}\right)\sigma_{i+1} + \left(\frac{h_{i+1}}{6}\right)\sigma_{i+2} = \frac{y_{i+2}-y_{i+1}}{h_{i+1}} - \frac{y_{i+1}-y_i}{h_i}$$

for $i = 0,1, \dots, n-2$, where $\sigma_0 = \sigma_n = 0.$

Cubic Spline Interpolation

- System equation of $n-1$ unknowns $\sigma_1, \ldots, \sigma_{n-1}$ and $n-1$ equations

$$\left(\frac{h_i}{h_i+h_{i+1}}\right)\sigma_i + 2\sigma_{i+1} + \left(\frac{h_{i+1}}{h_i+h_{i+1}}\right)\sigma_{i+2} = \frac{6\left(\frac{y_{i+2}-y_{i+1}}{h_{i+1}} - \frac{y_{i+1}-y_i}{h_i}\right)}{h_i+h_{i+1}}$$

for $i = 0,1,\ldots,n-2$, where $\sigma_0 = \sigma_n = 0$.

Cubic Spline Interpolation

Solve the following tridiagonal system:

$$\begin{pmatrix} 2 & \lambda_0 & & & & \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & & \ddots & & \\ & & & \mu_{n-3} & 2 & \lambda_{n-3} \\ & & & & \mu_{n-2} & 2 \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \\ \vdots \\ \sigma_{n-1} \end{pmatrix} = \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-2} \end{pmatrix}$$

$$\mu_i = \frac{h_i}{h_i + h_{i+1}} \qquad \lambda_i = 1 - \mu_i \qquad \gamma_i = \frac{6\left(\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}\right)}{h_i + h_{i+1}}$$

More interpolation…

- Interpolation is a very rich mathematical subject with ubiquitous application, such as in Machine Learning, Geographical Information System, Computer Aided Design, etc.
- More interesting interpolation to explore:
  - Bézier curve (used in font design and PDF drawing)
  - Multivariate interpolation (in 3D)
- Have any idea on applying interpolation to some problems? Let’s discuss!

THANK YOU

Do not hesitate to ask, if you have any question.