Lecture 2: Interpolation

J.K. Ryan@tudelft.nl

WI3097TU
Delft Institute of Applied Mathematics
Delft University of Technology

10 September 2012
Outline

1. Review

2. Lagrange Interpolation
   - Definition
   - Forming the approximation
   - Approximation properties
   - Approximation error

3. Spline Interpolation
Important tools: **Taylor’s theorem, Triangle inequality**

- How numbers are stored
- Errors
  - Roundoff error: \( \text{fl}(x) - x \)
  - Absolute error: \( |e| = |u_{\text{exact}} - u_{\text{approximate}}| \)
  - Relative error:
    
    \[
    \frac{\text{absolute error}}{|u_{\text{exact}}|} = \frac{|u_{\text{exact}} - u_{\text{approximate}}|}{|u_{\text{exact}}|}
    \]

- Landau’s \( \mathcal{O} \) symbol
Beginning of Numerics

Overall Goal: to solve

\[ \frac{d}{dx} u(x) = f(u(x)) \]

- Approximate a function (today).
- Approximate the derivative (next time).
- Approximate an integral (later).

Today’s Goal: Given a finite set of data points that represent a function, \( y = f(x) \),

\[ (x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \]

we want to form an approximation to \( y = f(x) \).
To approximate a function, one method is through interpolation.

- Lagrange interpolation.
- Cubic spline interpolation.
- ...
Interpolation/Extrapolation

Given a set of data points,

\((x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)

where \(y_n = f(x_n)\).

**Definition (Interpolation)**

For **interpolation**, we try to approximate a value in between the given data points. That is, an approximation, \(y_h \approx f(x)\), where \(x_0 \leq x \leq x_n\).

**Definition (Extrapolation)**

For **extrapolation**, we try to approximate a value outside the range of the given data points. That is, an approximation, \(y_h \approx f(x)\), where either \(x < x_0\) or \(x > x_n\).
Interpolation

Example

Given a set of data points,

\[(2, \frac{1}{2}), \quad (\frac{5}{2}, \frac{2}{5}), \quad (4, \frac{1}{4}),\]

we want to find the value at

\[x = 3, \Rightarrow \text{interpolate}\]

\[x = 5, \Rightarrow \text{extrapolate}.\]

Simplest method: use piecewise constants.
Linear interpolation is a straight line between 2 points: \((x_0, y_0), (x_1, y_1)\) where \(y_0 = f(x_0)\) and \(y_1 = f(x_1)\).

Left: We use the endpoints for the approximation.
Linear interpolation

The approximation to $f(x)$ is given by the linear interpolation formula

$$p(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1.$$  

Note: Quite similar to the point-slope formula, $y = mx + b$, where $m =$slope and $b =$y intercept.
Linear interpolation

We can right this in another way:

\[ p(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 \]

\[ = L_{0,1}(x)y_0 + L_{1,1}(x)y_1 \]

where

\[ L_{0,1}(x) = \frac{x - x_1}{x_0 - x_1} \] and \[ L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0} \]
Linear interpolation

Notice:

- Polynomial of degree 1 (with order 2)
- Constructed using 2 points
- Can relate to Linearization:

$$L(x) = f(x_0) + f'(x_0)(x - x_0)$$

At $x_0$:
\[ p(x_0) = y_0, \quad L_{0,1}(x_0) = 1, \quad L_{1,1}(x_1) = 0 \]

At $x_1$:
\[ p(x_1) = y_1, \quad L_{0,1}(x_0) = 0, \quad L_{1,1}(x_1) = 1 \]
Generalization to higher order:

Given $n + 1$ points:

$$(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$$

Is it possible to construct a polynomial of degree $n$ that interpolates $f(x)$ at those points?
**Definition (Lagrange interpolating polynomial)**

Given $n+1$ data points,

$$(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n)),$$

we can construct a polynomial of degree $n$ at those points,

$$p(x) = \sum_{k=0}^{n} f(x_k)L_{k,n}(x),$$

where $p(x)$ is the Lagrange interpolating polynomial and $L_{k,n}(x)$ are the Lagrange coefficients which are given by

$$L_{k,n}(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}.$$
Interpolation

**Theorem (Lagrange interpolation)**

If points \( x_0, x_1, \ldots, x_n \) are \( n + 1 \) distinct points and \( f(x) \) is a function whose values are given at these points, then there exists a unique polynomial, \( p(x) \), of degree at most \( n \) such that

\[
f(x_k) = p(x_k), \quad k = 0, 1, \ldots, n.
\]

The polynomial is given by

\[
p(x) = \sum_{k=0}^{n} f(x_k) L_{k,n}(x),
\]

where

\[
L_{k,n}(x) = \prod_{\substack{j = 0 \atop j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}, \quad k = 0, 1, \ldots, n.
\]
Notice:

- \( p(x) \) is a polynomial of degree \( n \).
- Constructed using \( n + 1 \) points.
- \[
    L_{k,n}(x_j) = \begin{cases} 
    0, & j \neq k \\ 
    1, & j = k 
    \end{cases}
\]
- Polynomial is unique.
- \( p(x_j) = f(x_j) \) for \( j = 0, 1, \ldots, n \).
Example

Given $x_0 = 2$, $x_1 = 2.5$, $x_2 = 4$, find an interpolating polynomial of degree 2 for $f(x) = \frac{1}{x}$.

This means our given data is

$$\left(2, \frac{1}{2}\right), \left(\frac{5}{2}, \frac{2}{5}\right), \left(4, \frac{1}{4}\right).$$

We have $3 = n + 1$ points. We should be able to construct a polynomial of degree $2 = n$. 
Interpolation

Example (continued)

In this case,

\[ p(x) = \sum_{k=0}^{n} f(x_k)L_{k,n}(x) = \sum_{k=0}^{2} f(x_k)L_{k,2}(x), \]

where

\[ L_{k,2}(x) = \prod_{\substack{j=0 \atop j \neq k}}^{2} \frac{x - x_j}{x_k - x_j}, \quad k = 0, 1, 2. \]

More specifically,

\[ L_{0,2}(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad L_{1,2}(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \]

\[ L_{2,2}(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \]
Plugging in our data:

\[ p(x) = \sum_{k=0}^{2} f(x_k) L_{k,2}(x) \]

\[ = f(x_0) L_{0,2}(x) + f(x_1) L_{1,2}(x) + f(x_2) L_{2,2}(x) \]

\[ = \frac{1}{2} L_{0,2}(x) + \frac{2}{5} L_{1,2}(x) + \frac{1}{4} L_{2,2}(x) \]

where

\[ L_{0,2}(x) = \frac{(x - \frac{5}{2})(x - 4)}{(2 - \frac{5}{2})(2 - 4)} \]

\[ L_{1,2}(x) = \frac{(x - 2)(x - 4)}{\left(\frac{5}{2} - 2\right)\left(\frac{5}{2} - 4\right)} \]

\[ L_{2,2}(x) = \frac{(x - 2) \left(x - \frac{5}{2}\right)}{(4 - 2)\left(4 - \frac{5}{2}\right)} \]
Interpolation
Example (continued)

Simplifying, we have that

\[ p(x) = (0.05x - 0.425)x + 1.15 \]

interpolates \( f(x) = \frac{1}{x} \) for \( 2 \leq x \leq 4 \).
Once we have constructed the approximation, we should always ask:

**How good is this approximation?**

Absolute Error $= |Exact - Approximate|$

$= |f(x) - p(x)|$

$= \left| \frac{1}{x} - ((0.05x - 0.425)x + 1.15) \right|$
At $x = 3$ (interpolation):

$$\text{Absolute Error} = \left| \frac{1}{3} - ((0.05(3) - 0.425)3 + 1.15) \right| = 0.0083\bar{3}$$

At $x = 5$ (extrapolation):

$$\text{Absolute Error} = \left| \frac{1}{5} - ((0.05(5) - 0.425)5 + 1.15) \right| = 0.075$$
Interpolation Error

Interpolation, $x = 3$:

$$|f(x) - p(x)| = 0.00833$$

Extrapolation, $x = 5$:

$$|f(x) - p(x)| = 0.075$$
Theorem (Interpolation error)

Given $n + 1$ distinct points in $[a, b]$, and a function with $n + 1$ continuous derivatives, $f \in C^{n+1}(a, b)$, then for each $x \in [a, b]$ there exists a $\xi \in (a, b)$ with

$$|f(x) - p(x)| = \left| (x - x_0)(x - x_1) \cdots (x - x_n) \frac{1}{(n + 1)!} f^{(n+1)}(\xi) \right|,$$

where $p(x)$ is the Lagrange interpolating polynomial of degree $n$. 

---

Interpolation // J.K. Ryan

10 September 2012 23 / 43
From the previous example: $f(x) = \frac{1}{x} \Rightarrow f^{(3)} = -\frac{6}{x^4}$

The error formula would then be:

$$|f(x) - p(x)| = \left| (x - x_0)(x - x_1)(x - x_2) \frac{1}{3!} f^{(3)}(\xi) \right|$$

$$= \left| (x - 2) \left( x - \frac{5}{2} \right) (x - 4) \frac{1}{3!} \left( -\frac{6}{\xi^4} \right) \right|$$

$$\leq \frac{1}{3!} \max_{\xi \in [2,4]} \left| \frac{6}{\xi^4} \right| \left| (x - 2) \left( x - \frac{5}{2} \right) (x - 4) \right|$$
Sometimes, it is better to use a lower order polynomial approximation, but use smaller intervals.

**Example**

Let \( f(x) = e^x \) and \( x \in [0, 1] \). Suppose we want to use a piecewise linear interpolation using points \( x_0, x_1, \ldots, x_n \).

- Find a piecewise linear interpolant for \( f(x) \).
- What should the spacing be between data to keep the error less than \( 10^{-6} \)?
Interpolation

Example

Define $x_j = 0 + jh$, $j = 0, \ldots, n$. $h$ then represents the spacing between the data,

$$h = x_{j+1} - x_j.$$

For linear interpolation, the Lagrange interpolant for $x \in [x_j, x_{j+1}]$ is given by

$$p(x) = f(x_j)L_{j,1}(x) + f(x_{j+1})L_{j+1,1}(x), \quad x \in (x_j, x_{j+1})$$

$$= f(x_j) \frac{x - x_j}{x_j - x_{j+1}} + f(x_{j+1}) \frac{x - x_{j+1}}{x_{j+1} - x_j}$$
Interpolation

Example

This gives

\[ p(x) = e^{x_j} \frac{x - x_j}{x_j - x_{j+1}} + e^{x_{j+1}} \frac{x - x_{j+1}}{x_{j+1} - x_j} \]

But \( h = x_{j+1} - x_j \)

\[ \Rightarrow p(x) = \frac{1}{h} ( (x_{j+1} - x) e^{x_j} + (x - x_j) e^{x_{j+1}} ) \]
Interpolation

Example

We want

\[ |f(x) - p(x)| \leq 10^{-6} \]

For which \( h \) does this hold?

Using our interpolation error theorem

\[ |f(x) - p(x)| \leq \left| (x - x_j)(x - x_{j+1}) \frac{1}{2} f^{(2)}(\xi) \right| \]

\[ \leq \frac{1}{2}! \left| f^{(2)}(\xi) \right| \| (x - x_j)(x - x_{j+1}) \| \]

\[ \leq \frac{1}{2} \max_{\xi \in (0,1)} e^{\xi} \left| (x - jh)(x - (j + 1)h) \right| \leq \frac{e h^2}{2 \cdot 4} \]
Interpolation

Example

We want

\[ |f(x) - p(x)| \leq 10^{-6} \]

where

\[ |f(x) - p(x)| \leq \frac{e}{8} h^2 \leq 10^{-6} \]

\[ \Rightarrow h^2 \leq 8e^{-1}10^{-6} \]

\[ \Rightarrow h \leq \left(8e^{-1}10^{-6}\right)^{1/2} \]
Lagrange Interpolation

Summary

One method of constructing an approximation to a function, \( f(x) \) is

1. Take \( n + 1 \) data points: \( x_0, x_1, \ldots, x_n \).
2. Define the Lagrange coefficients:

\[
L_{k,n}(x) = \prod_{\substack{j = 0 \atop j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}, \quad k = 0, 1, \ldots, n.
\]

3. Then the approximating polynomial is given by

\[
p(x) = \sum_{k=0}^{n} f(x_k) L_{k,n}(x)
\]

4. The error is given by

\[
|f(x) - p(x)| \leq \frac{1}{(n + 1)!} \left| \prod_{k=0}^{n} (x - x_k) \right| \max_{\xi \in (x_0, x_n)} \left| f^{(n+1)}(\xi) \right|
\]
Spline Interpolation

Another way to form a piecewise polynomial approximation is through Spline interpolation.

- Break the intervals into subintervals
  \[ \{(x_0, y_0), (x_1, y_1)\}, \{(x_1, y_1), (x_2, y_2)\}, \ldots \{(x_{n-1}, y_{n-1}), (x_n, y_n)\} \]

- Use piecewise polynomial approximation
- Use differentiability at endpoints of subintervals
Spline Interpolation
Linear interpolation

We could use piecewise linear polynomials:

**Definition**
Given a function \( f \) on \([a, b]\) and a set of nodes \( a = x_0 < x_1 < \cdots < x_n = b \), a **linear spline interpolant** is a function \( s(x) \in C[a, b] \) such that

- \( s_i(x) \) is a **linear polynomial** on the interval \([x_i, x_{i+1}]\), \( i = 0, \ldots, n - 1 \)
- \( s_i(x_i) = f(x_i) \), \( i = 0, \ldots, n \)

But this gives something like piecewise Lagrange interpolation.
What about a higher order piecewise interpolant?

**Definition**

Given a function $f$ on $[a, b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant is a function $s(x) \in C^2[a, b]$ such that

1. $s_i(x)$ is a polynomial of degree 3 on each subinterval $[x_i, x_{i+1}]$, $i = 0, \ldots, n - 1$
2. $s_i(x_i) = f(x_i)$, $i = 0, \ldots, n - 1$ and $s_i(x_{i+1}) = f(x_{i+1})$
3. $\frac{d^k}{dx^k} s_i(x_{i+1}) = \frac{d^k}{dx^k} s_{i+1}(x_{i+1})$, $i = 0, \ldots, n - 2$, $k = 0, 1, 2$
4. (One possible condition) $s_0''(x_0) = s_{n-1}''(x_n) = 0$
Cubic Spline Interpolation

This means that given equally spaced data,

\[(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\]

we can define

\[s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i\]

on \([x_i, x_{i+1}]\), where \(h = x_{i+1} - x_i\).

We must solve for the coefficients, \(a_i, b_i, c_i, d_i\).
Cubic Spline Interpolation

Step 1

Plug in $x = x_i$ into

$$s_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i, \quad i = 0, \ldots, n-1$$

$$s_i(x_i) = f(x_i)$$

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

$$= d_i$$

$$\Rightarrow d_i = f(x_i), \quad i = 0, \ldots, n - 1$$
Cubic Spline Interpolation

Step 2

Use $s''(x_{i+1}) = s''_{i+1}(x_{i+1})$ and $s''(x_0) = s''_{n-1}(x_n) = 0$ where

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

This gives

$$s''(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i = 6a_i h + 2b_i$$
$$s''_{i+1}(x_{i+1}) = 2b_{i+1}$$

(1)

$$\Rightarrow a_i = \frac{1}{3h} (b_{i+1} - b_i), \quad i = 0, \ldots, n-2$$

and

$$\Rightarrow b_0 = 0, \quad a_{n-1} = -\frac{1}{3h} b_{n-1} \quad i = 0, \ldots, n-2$$
Cubic Spline Interpolation

Step 3

Use \( s_i(x_{i+1}) = s_{i+1}(x_{i+1}) \) and \( s_{n-1}(x_n) = y_n \) Which gives,

\[
s_i(x_{i+1}) = a_i(x_{i+1} - x_i)^3 + b_i(x_{i+1} - x_i)^2 + c_i(x_{i+1} - x_i) + y_i
\]

\[= a_i h^3 + b_i h^2 + x_i h + y_i\]

\[s_{i+1}(x_{i+1}) = y_{i+1}\]

So that we have

\[
\Rightarrow c_i = \frac{1}{h}(y_{i+1} - y_i) - \frac{h}{3}(b_{i+1} + 2b_i), \quad i = 0, \ldots, n - 2
\]

and

\[
\Rightarrow c_{n-1} = \frac{1}{h}(y_n - y_{n-1}) - \frac{2h}{3}b_n,
\]
Cubic Spline Interpolation

Step 4

From \( s'_i(x_{i+1}) = s'_{i+1}(x_{i+1}) \) we obtain a linear system:

\[
\begin{align*}
  s'_i(x_{i+1}) &= \frac{2}{3} h b_{i+1} + \frac{1}{3} h b_i + \frac{1}{h} (y_{i+1} - y_i) \\
  s'_i(x_{i+1}) &= -\frac{1}{3} (b_{i+2} + 2b_{i+1}) + 1h(b_{i+1} + 2b_{i+1})
\end{align*}
\]

Which gives

\[
\Rightarrow b_{i+2} + 4b_{i+1} + b_i = \frac{3}{h^2} (y_{i+2} - 2y_{i+1} + y_i), \quad i = 0, \ldots, n-3
\]
Cubic Spline Interpolation

Algorithm

Given data,

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

we define

\[ s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \]

on \([x_i, x_{i+1}]\), where \(h_i = x_{i+1} - x_i\). To find the coefficients:

1. Solve linear system \(Mb = g\) for \(b = [b_0, b_1, \ldots, b_{n-1}]^T\)
2. Calculate \(c_i\).
3. Calculate \(a_i\).
4. Form \(s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i\) for \(x \in [x_i, x_{i+1}]\).
Cubic Spline Interpolation

Example

Given data

\[(1, 1), \left(2, \frac{1}{2}\right), \left(3, \frac{1}{3}\right), \left(4, \frac{1}{4}\right),\]

The cubic spline interpolant that approximates \( f(x) = \frac{1}{x} \) using this data is given by

\[
S(x) = \begin{cases} 
\frac{1}{12}(x^3 - 3x^2 - 4x + 18), & 1 \leq x \leq 2, \\
\frac{1}{12}(-x^3 + 9x^2 - 28x + 34), & 2 < x \leq 3, \\
\frac{1}{12}(-x + 7), & 3 < x \leq 4.
\end{cases}
\]
Cubic Spline Interpolation

How did we find this Spline interpolant?

- Piecewise polynomial intervals: \([x_0, x_1], [x_1, x_2], [x_2, x_3]\). This means we have 3 spline interpolants \(s_0(x), s_1(x), s_2(x)\).
- We know \(d_i = f(x_i) = \frac{1}{x_i}, \quad i = 0, \ldots, n - 1\).
- Boundaries: \(s_0''(x_0) = s_2''(x_3) = 0 \Rightarrow s_0''(1) = s_2''(4) = 0\)

\[\Rightarrow b_0 = 0 \quad a_2 = -\frac{1}{3} b_2\]

- Solve the system for \(b_i, \quad i = 1, 2\):

\[
\begin{bmatrix}
4 & 1 \\
1 & 4
\end{bmatrix}
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
=
\begin{bmatrix}
1 \\
\frac{1}{4}
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
=
\begin{bmatrix}
\frac{1}{4} \\
0
\end{bmatrix}
\]
Cubic Spline Interpolation

- Using $c_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{h_i}{3}(b_{i+1} + 2b_i)$:
  
  $$c_0 = -\frac{7}{12}, \quad c_1 = -\frac{1}{3}, \quad c_2 = -\frac{1}{12}$$

- Using $a_i = \frac{1}{2h_i}(b_{i+1} - b_i)$:
  
  $$a_0 = \frac{1}{12}, \quad a_1 = -\frac{1}{12}, \quad a_2 = 0$$

- Gives
  
  $$s(x) = \begin{cases} 
  s_0(x) = \frac{1}{12}(x - 1)^3 - \frac{7}{12}(x - 1) + 1, & \leq x \\
  s_1(x) = -\frac{1}{12}(x - 2)^3 + \frac{1}{4}(x - 2)^2 - \frac{1}{3}(x - 2) + \frac{1}{2}, & 2 < x \\
  s_2(x) = -\frac{1}{12}(x - 3) + \frac{1}{3}, & 3 < x
  \end{cases}$$
Material addressed

1. Review
2. Lagrange Interpolation
   - Definition
   - Forming the approximation
   - Approximation properties
   - Approximation error
3. Spline Interpolation

Material in book:
Chapter 2, Sections 1-3, 5-6

HIGHLY Recommended exercises:
All from Ch. 2.