Numerical Methods - Interpolation & Splines

Y. K. Goh

Universiti Tunku Abdul Rahman

2013
Outline

1 Introduction

2 Interpolating Polynomials
   - Existence and Uniqueness of Interpolating Polynomial
   - Lagrange Interpolating Polynomial
   - Divided Difference & Newton Interpolating Polynomials

3 Splines
Introduction

**The Problem**: Given a set of data

<table>
<thead>
<tr>
<th>$x$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$\ldots$</th>
<th>$x_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>$y_0$</td>
<td>$y_1$</td>
<td>$\ldots$</td>
<td>$y_n$</td>
</tr>
</tbody>
</table>

We can have three different scenario to ask questions:

- Can we reproduce the points **exactly** by a simple function $p$? – **Interpolation**.
- Assume the points are generated from complicated (usually means computationally expensive) function $f$, can we find a simpler function $g$ to reproduce reasonable (usually means within the full machine precision) approximation to $f$? – **Interpolation**.
- Assume points contains errors, can we reproduce the points **approximately** by a simple function, $\hat{y} = \hat{y}(x)$? – **Curve fitting**.

Depending on the strategies to treat the problem we can:

- construct interpolating polynomial of degree $m$, $p_m$.
- construct spline of degree $m$, $S$.
- construct the least square fit to the curve.

We will cover only interpolating polynomials and splines.
1 Introduction

2 Interpolating Polynomials
   - Existence and Uniqueness of Interpolating Polynomial
   - Lagrange Interpolating Polynomial
   - Divided Difference & Newton Interpolating Polynomials

3 Splines
The Problem: Given a set of data

<table>
<thead>
<tr>
<th>$x$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>...</th>
<th>$x_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>$y_0$</td>
<td>$y_1$</td>
<td>...</td>
<td>$y_n$</td>
</tr>
</tbody>
</table>

- **Interpolation** is to find a function $f$ such that reproduces the given data points exactly, i.e. $f(x_i) = y_i$, for $x_0 \leq x \leq x_n$.
- The given data points $(x_i, y_i)$ are called **nodes**.
- There is no other information in between, $x_i < x < x_{i+1}$, we let $f$ be the exact function (which is unknown to us) that generates the data.
- In most problem, we want a simple function, usually a polynomial, $p(x)$ to approximate $f$ (Weierstraß theorem).
- The main reason for using polynomial in interpolation is that the derivative and integration are easy to determine.
- Other commonly used classes of interpolation functions are rational functions and trigonometric functions (Fourier series).
**Theorem (Weierstraß Approximation Theorem)**

Suppose \( f \) is defined and continuous on \([a, b]\). For each \( \epsilon > 0 \), there exists a polynomial \( p(x) \) such that

\[
|f(x) - p(x)| < \epsilon, \quad \text{for all } x \in [a, b].
\]

**Figure:** Illustration of Weierstraß Theorem.
Outline

1 Introduction

2 Interpolating Polynomials
   - Existence and Uniqueness of Interpolating Polynomial
   - Lagrange Interpolating Polynomial
   - Divided Difference & Newton Interpolating Polynomials

3 Splines
Existence and Uniqueness

**Theorem (Existence and Uniqueness)**

Let \( \{x_i\}_{i=0}^{n} \) be \((n + 1)\) distinct points in \([a, b]\). Let \( \{y_i\}_{i=0}^{n} \) be any set of real numbers, then there exists a unique polynomial of degree \( n \), \( p(x) \in \mathcal{P}_n \), such that \( p(x_i) = y_i, \forall i \in [0, n] \).

- **Existence:** Proof by construction of \( p \in \mathcal{P}_n \) by Newton algorithm.
- **Uniqueness:** Proof by contradiction.
- **Implication:** Suppose we have \( n + 1 \) points, the theorem tell us there is one and only one polynomial of degree \( n \) that fit all the data points.
Newton’s Algorithm

We will construct a polynomial \( p \) that passes through all the \( n + 1 \) points:

1. For \( n = 0 \), we choose \( p_0(x) = y_0 \).
2. For \( n \geq 1 \), we construct the polynomial recursively with

\[
p_k(x) = p_{k-1}(x) + c(x - x_0)(x - x_1) \ldots (x - x_{k-1})
\]

where the constant \( c \) is determined from the condition \( p_k(x_k) = y_k \).

Example

Construct the Newton polynomial of degree \( \leq 2 \) that interpolates the points \((1, 2), (3, 5), (4, 8)\).

**ANSWER :**

- First point \((1, 2)\), \( p_0(x) = 2 \).
- First two points: \( p_1(x) = p_0(x) + c_1(x - 1) \). Since \( p_1(3) = 5 \), we get
  \[
  2 + 2c_1 = 5 \implies c_1 = \frac{3}{2}, \text{ ie. } p_1(x) = 2 + \frac{3}{2}(x - 1).
  \]
- Similarly, \( p_2(x) = p_1(x) + c_2(x - 1)(x - 3) \). Since \( p_2(4) = 8 \), we have
  \[
  8 = 2 + \frac{3}{2}(3) + c_2(3)(1) \implies c_2 = \frac{1}{2}, \text{ ie. } p_2(x) = 2 + \frac{3}{2}(x - 1) + \frac{1}{2}(x - 1)(x - 3).
  \]
Uniqueness of Interpolating Polynomial

The proof of the uniqueness of interpolating polynomial of degree $n$ utilised the Fundamental Theorem of Algebra.

**Theorem (Fundamental Theorem of Algebra)**

A polynomial $p(x) = a_0 + a_1x + \cdots + a_kx^k$ of degree $k$ cannot have more than $k$ roots unless $p(x) \equiv 0$.

- Let $p$ be the Newton polynomial of degree $n$ that passes through the $n + 1$ distinct points.
- Let $q$ be another distinct polynomial of degree $n$ that also passes through the same $n + 1$ points.
- Let $r(x) = q(x) - p(x)$, and which is also a polynomial of degree at most $n$.
- Since at the nodes $q(x_i) = p(x_i), 0 \leq i \leq n$, thus there are at least $n + 1$ points where $r(x)$ is zero. From Fundamental Theorem of Algebra, this can only happens when $r(x) \equiv 0$, ie. $q(x) \equiv p(x)$ which contradict with the assertion that $q(x)$ is different from $p(x)$.
Outline

1. Introduction

2. Interpolating Polynomials
   - Existence and Uniqueness of Interpolating Polynomial
   - Lagrange Interpolating Polynomial
   - Divided Difference & Newton Interpolating Polynomials

3. Splines
Cardinal Polynomial

Definition (Cardinal Polynomial)

Given a set of points \((x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\), a cardinal polynomial \(L_k(x)\) is a polynomial defined as

\[
L_k(x) = \prod_{\substack{i=0 \\
i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)} = \frac{(x - x_0) \ldots (x - x_{k-1})(x - x_{k+1}) \ldots (x - x_n)}{(x_k - x_0) \ldots (x_k - x_{k-1})(x_k - x_{k+1}) \ldots (x_k - x_n)}.
\]

Note:

- \(L_k(x_j) = \delta_{kj} = \begin{cases} 
0, & j \neq k \\
1, & j = k 
\end{cases}\)
Definition (Lagrange Interpolating Polynomial)

Given a set of points \((x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\), Lagrange polynomial is a polynomial of degree \(n\) defined as

\[
p(x) = \sum_{i=0}^{k} y_k L_k(x)
\]

Note:

- On the nodes, \(p(x_k) = y_k = f(x_k)\), ie \(p(x)\) passes through all nodes.
Example

Given the following set of data

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
<td>2</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

Find the Lagrange polynomial and estimate $f(2.5)$.

**ANSWER:**

$$
L_0(x) = \frac{(x - 3)(x - 4)}{(1 - 3)(1 - 4)} = \frac{1}{6}(x - 3)(x - 4),
$$

$$
L_1(x) = \frac{(x - 4)(x - 1)}{(3 - 4)(3 - 1)} = -\frac{1}{2}(x - 4)(x - 1),
$$

$$
L_2(x) = \frac{(x - 1)(x - 3)}{(4 - 1)(4 - 3)} = \frac{1}{3}(x - 1)(x - 3)
$$

$$
p(x) = 2\left(\frac{1}{6}(x - 3)(x - 4) - \frac{5}{2}(x - 4)(x - 1) + \frac{8}{3}(x - 1)(x - 4)\right),
$$

$$
p(2.5) = 3.8750.
$$
Error Formula for Lagrange Polynomial

Theorem

Suppose \( \{x_i\}_{i=0}^{n} \) are distinct points in the interval \([a, b]\) and \( f \in C^{n+1}[a, b] \). Then, for each \( x \) in \([a, b]\), a number \( \xi(x) \) between \( x_0, x_1, \ldots, x_n \), and hence \([a, b]\), exists with

\[
f(x) = p(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n)
\]

where \( p(x) \) is the Lagrange interpolating polynomial.

Note: There are cases where a function is not very well approximated by a highorder polynomial then a highorder polynomial is a bad choice to use for approximation.

Theorem (Upper Bound Lemma)

Suppose that \( x_i = a + ih \) for \( i = 0, 1, \ldots, n \) and that \( h = (b - a)/n \). Then for any \( x \in [a, b] \)

\[
\prod_{i=0}^{n} |x - x_i| \leq \frac{1}{4} h^{n+1} n!
\]
Theorem

Let $f$ be a function such that $f^{(n+1)}$ is continuous on $[a, b]$ and satisfies $|f^{(n+1)}(x)| \leq M$. Let $p$ be the polynomial of degree $\leq n$ that interpolates $f$ at $n + 1$ equally spaced nodes in $[a, b]$, including the endpoints. Then on $[a, b]$

$$|f(x) - p(x)| \leq \frac{1}{4(n + 1)} M h^{n+1}$$

where $h = (b - a)/n$ is the spacing between nodes.

Example

Given the data $(0, 1), (1, 2)$, and $(2, 4)$ are generated from $f(x) = 2^x$. Find the maximum error when the Lagrange polynomial that passes through all the points are used to estimate $f(1.5)$. 
Outline

1. Introduction

2. Interpolating Polynomials
   - Existence and Uniqueness of Interpolating Polynomial
   - Lagrange Interpolating Polynomial
   - Divided Difference & Newton Interpolating Polynomials

3. Splines
Recall the Newton polynomial that interpolates the \( n + 1 \) distinct nodes, \( x_0, x_1, \ldots, x_n \), with respect to a function \( f \):

\[
p(x) = x_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}).
\]

The coefficients of Newton polynomial \( p(x) \) could be found from the divided differences.

**Definition (Divided Difference)**

For an arbitrary function \( f \) over \( x_0, x_1, \ldots, x_n \) distinct nodes, we define the divided differences \( f[x_0, \ldots, x_n] \):

- \( f[x_0] = f(x_0) \)
- \( f[x_0, x_1] = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \), and recursively
- \( f[x_0, \ldots, x_n] = \frac{f[x_0, \ldots, x_{n-1}] - f[x_1, \ldots, x_n]}{x_0 - x_n} \).
Now, with the definition of divided differences, we can write the Newton’s polynomial as

\[ p(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\ldots(x - x_k). \]

**Example**

Assume that (1, 2), (3, 5) and (4, 8) sampled from a function \( f \). Compute \( f[1], f[1, 3], f[1, 3, 4] \) and, hence, write down the Newton’s polynomial.
Theorem

Let \( p \) be the Newton interpolation polynomial of degree \( n \) that interpolates a function \( F \) at a set of \( n + 1 \) distinct points, \( x_0, x_1, \ldots, x_n \in [a, b] \). If \( t \) is a point different from all the \( x_i \)'s then

\[
f(x) = p(t) + f[x_0, x_1, \ldots, x_n, t] \prod_{i=0}^{n} (t - x_i).
\]

Example

Given the data \((1, 0), (e, 1), \) and \((e^2, 2)\) are generated from \( f(x) = \ln x \). Find the maximum error when the Newton polynomial that passes through all the points are used to estimate \( f(3) \).
Forward Difference and Backward Difference

Definition (Forward Difference)

Let \( f \) be an arbitrary function and \( h \) be a positive real number. The Forward Difference Operator, \( \Delta \) is defined as:

- \( \Delta^0 f(x) = f(x) \)
- \( \Delta^1 f(x) = f(x + h) - f(x) \)
- \( \Delta^k f(x) = \Delta^{k-1} f(x + h) - \Delta^{k-1} f(x). \)

Definition (Backward Difference)

Let \( f \) be an arbitrary function and \( h \) be a positive real number. The Backward Difference Operator, \( \nabla \) is defined as:

- \( \nabla^0 f(x) = f(x) \)
- \( \nabla^1 f(x) = f(x) - f(x - h) \)
- \( \nabla^k f(x) = \nabla^{k-1} f(x) - \nabla^{k-1} f(x - h). \)
Newton’s Forward/Bacward Difference Formula

Let \( x_0, x_1, \ldots, x_n \) are arranged in an increasing order with \( h = x_{i+1} - x_i \) for \( i = 0, \ldots, n - 1 \), we have

\[
f[x_0, x_1, \ldots, x_k] = \frac{1}{k! h^k} \Delta^k f(x_0)
\]

and

\[
f[x_n, x_{n-1}, \ldots, x_{n-k}] = \frac{1}{k! h^k} \nabla^k f(x_0)
\]

Definition (Newton Forward Difference Formula)

\[
p(x) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f[x_0], \quad s = (x - x_0)/h.
\]

Definition (Newton Backward Difference Formula)

\[
p(x) = f[x_n] + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f[x_0], \quad s = (x - x_n)/h.
\]
Example

Find the Newton Forward and Backward polynomials for the following set of data: 

\((-2, -0.99532), (-1, -0.84270), (0, 0), (1, 0.84270), (2, 0.99532)\).

\(h = 1\).

<table>
<thead>
<tr>
<th>(x)</th>
<th>(f(x))</th>
<th>(\Delta f)</th>
<th>(\Delta^2 f)</th>
<th>(\Delta^3 f)</th>
<th>(\Delta^4 f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>-0.99532</td>
<td>0.15262</td>
<td>0.69008</td>
<td>-0.69008</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-0.84270</td>
<td>0.84270</td>
<td>0</td>
<td>-0.69008</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.84270</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.84270</td>
<td>-0.69008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.99532</td>
<td>0.15262</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Forward Polynomial**

\[ p(x) = -0.99532 + \frac{0.15262}{1!1!} (x+2) + \frac{0.69008}{2!1^2} (x+2)(x+1) - \frac{0.69008}{3!1^3} (x+2)(x+1)x. \]

- **Backward Polynomial**

\[ p(x) = 0.99532 + \frac{0.15262}{1!1!} (x-2) - \frac{0.69008}{2!1^2} (x-2)(x-1) + \frac{0.69008}{3!1^3} (x-2)(x-1)x. \]
Outline

1. Introduction

2. Interpolating Polynomials
   - Existence and Uniqueness of Interpolating Polynomial
   - Lagrange Interpolating Polynomial
   - Divided Difference & Newton Interpolating Polynomials

3. Splines
Spline interpolation

Definition (Spline function)

Let the interval \([a, b]\) be composed of \(n\) ordered disjoint subintervals \([x_k, x_{k+1}]\) with \(a = x_0 < x_1 < \cdots < x_n = b\). A spline function

\[ S(x) = \begin{cases} 
S_0(x), & x_0 \leq x \leq x_1 \\
S_1(x), & x_1 \leq x \leq x_2 \\
& \vdots \\
S_{n-1}(x), & x_{n-1} \leq x \leq x_n 
\end{cases} \]

is a function consists of piecewise-polynomials \(S_k(x), k = 0, 1, \ldots, n - 1\) joined together over all the subintervals with certain smoothness conditions.

- The order of the spline is the highest order of the polynomials \(S_k(x)\).
- If all the subintervals are of same length, the spline is said to be uniform.
- Commonly used splines are: natural cubic spline, B-spline and Bézier spline.
- Splines are widely used in interpolation and computer graphics.
Smoothness Conditions of Splines

**Definition (Continuity)**

A function \( f \) is continuous at some point \( s \) of its domain if the limit
\[
\lim_{x \to s} f(x) = f(s).
\]
In particular, for real function
\[
\lim_{x \to s^+} f(x) = \lim_{x \to s^-} f(x) = f(s).
\]

Usually the smoothness conditions of splines are conditions that:

- \( S \) is continuous in \([a, b]\).
- Thus, for an interior nodes \( x_k \), the connecting piecewise polynomial must be continuous, \( S_k(x_k) = S_{k+1}(x_k) \).
- In additions, to ensure the spline of degree \( m \) is smooth, the derivatives of the spline \( S', S'', \ldots, S^{(m-1)} \) are all continuous functions, except spline of degree 1.
Spline of Degree 1 and Quadratic Spline

Example

Find (i) the spline of degree 1, and (ii) quadratic spline with zero derivatives at end points for the following points: (1,2), (3,5), (4,8). Use the splines to estimate the value of \( y \) when \( x = 1.5 \).

**ANSWER:**

- \( S_k(x) = y_k + m_k(x-x_k) \)
- \( S(x) = \begin{cases} 2 + \frac{3}{2}(x-1), & x \in [1, 3] \\ 5 + 3(x-3), & x \in [3, 4] \end{cases} \)
- \( S(1.5) = S_0(1.5) = 2.75 \)
- \( Q_k(x) = a_k + b_k(x-x_k) + c_k(x-x_k)^2 \)
- \( Q(x) = \begin{cases} 2 + \frac{3}{4}(x-1)^2, & x \in [1, 3] \\ 5 + 3(x-3), & x \in [3, 4] \end{cases} \)
- \( Q(1.5) = Q_0(1.5) = 2.25 \)
Natural Cubic Spline

- Natural cubic spline is a spline function of degree 3 with the smoothness conditions:
  - For all interior nodes $1 \leq k \leq n - 1$, $S_k(x_k) = S_{k+1}(x_k)$, $S'_k(x_k) = S'_{k+1}(x_k)$ and $S''_k(x_k) = S''_{k+1}(x_k)$.
  - For the boundary nodes, $S''_0(x_0) = S''_{n-1}(x_n) = 0$.

- Note that there are all together $4n$ coefficients in the cubic spline.
- The interpolation at the nodes gives $2n$ constraints: $S_k(x_k) = y_k$ and $S_k(x_{k+1}) = y_{k+1}$ for $k \leq n - 1$.
- The continuity conditions for the derivative of spline at the interior points on $S'$ and $S''$ provides $2(n - 1)$ constraints.
- The remaining two constraints are chosen such that $S''(x_0) = S''(x_n) = 0$.

Example

Given the data points $(-2, -0.99532), (-1, -0.84270), (0, 0), (1, 0.84270), (2, 0.99532)$, plot the natural cubic spline that passes through these points.
THE END