Numerical Analysis: Interpolation

Computer Science, Ben-Gurion University

(slides based mostly on Prof. Ben-Shahar’s notes)

2019/2020, Fall Semester
1. Polynomial Interpolation

2. Analyzing the Error in Polynomial Interpolation

3. Optimal Placement of the Roots
Definition (a polynomial (from $\mathbb{R}$ to $\mathbb{R}$))

A polynomial, from $\mathbb{R} \rightarrow \mathbb{R}$, is a function of the form

$$P_n(x) = \sum_{j=0}^{n} a_j x^j = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

where $n$ is a nonnegative integer and $(a_j)_{j=0}^{n}$ are real numbers.
The importance of polynomials, for representing/approximating functions stems from the ease of computations of

$$x \mapsto P_n(x), \quad \frac{d}{dx} P_n(x) \quad \text{and} \quad \int P_n(x) \, dx$$

and from the fact they (uniformly) approximate continuous functions (see Theorem below)
Theorem (Weierstrass)

Every continuous function defined on a closed interval can be uniformly approximated as closely as desired by a polynomial function. More formally, let $f \in C([a, b])$. Then for every $\varepsilon > 0$ there exists a polynomial, $P_n$, such that, for every $x \in [a, b]$,

$$|f(x) - P_n(x)| < \varepsilon.$$
Polynomial Interpolation

The fundamental problem:

**Definition (the polynomial interpolation problem)**

Given \((n + 1)\) points,

\[
((x_i, y_i))_{i=0}^{n} \quad x_i \in \mathbb{R}, y_i \in \mathbb{R} \quad \forall i \in \{0, 1, \ldots, n\}
\]

such that the \(x_i\)'s are **distinct**, namely,

\[
i \neq j \Rightarrow x_i \neq x_j,
\]

find a polynomial, \(P(x)\), of **minimal degree**, such that “\(P\) interpolates the data”; i.e.

\[
P(x_i) = y_i \quad \forall i \in \{0, 1, \ldots, n\}.
\]
Example

- Assume the setting from the previous slide with the additional assumption that $y_i = c$ for every $i$, where $c \in \mathbb{R}$ is some constant.
- The zeroth-order polynomial,

$$P(x) \equiv c,$$

interpolates the data.
- Since its degree is zero, it is also minimal.
- Clearly, if $c' \in \mathbb{R}$ is some other constant, with $c \neq c'$, then the zeroth-order polynomial $Q(x) \equiv c'$ does not interpolate the data, since, for every $i$,

$$Q(x_i) = c' \neq c = y_i.$$
- It follows that $P(x) \equiv c$ is the the unique polynomial that satisfies the requirements.
Some Observations from that Example

- In the definition of polynomial interpolation, only the $x_i$’s are required to be distinct, not the $y_i$’s.
- We just saw, in that particular example, existence and uniqueness of the interpolation polynomial. As will see, the interpolation polynomial always exists and is always unique, not just in that example.
Theorem

Given \(((x_i, y_i))_{i=0}^{n}\), with \(x_i \neq x_j\) whenever \(i \neq j\), there exists a unique polynomial \(P_n\), of degree no greater than \(n\), such that

\[ P(x_i) = y_i \quad \forall i \in \{0, 1, \ldots, n\} . \]
Proof – uniqueness.

A proof by contradiction. Suppose there are two different polynomials, $P_n$ and $Q_n$, such that

$$P_n(x_i) = y_i = Q_n(x_i) \quad \forall i \in \{0, 1, \ldots, n\}.$$ 

Thus, defining a third polynomial by $S(x) = P_n(x) - Q_n(x)$ we get

$$S(x_i) = P_n(x_i) - Q_n(x_i) = 0 \quad \forall i \in \{0, 1, \ldots, n\}.$$ 

This implies that $S$ has at least $n + 1$ distinct roots. By construction, the degree of $S$ is no higher than $n$. By the fundamental theorem of algebra, a polynomial of degree $n$ has exactly $n$ roots (including complex roots, and including multiplicities), unless it is the zero polynomial. It follows that $S(x) \equiv 0$, and thus $P_n \equiv Q_n$, a contradiction. 

□
A proof by induction.

- **Base.** $n = 0$, given $(x_0, y_0)$. Choose

  $$P_0(x) \equiv C_0 \triangleq y_0.$$ 

- **Assumption.** Suppose there exists a polynomial, $P_{k-1}$, of degree no higher than $k - 1$, such that it interpolates the first $k$ points:

  $$P_{k-1}(x_i) = y_i \quad \forall i \in \{0, 1, \ldots, k - 1\}.$$ 

- **Step.** We will now build a polynomial, $P_k$, of degree no higher than $k$, that interpolates the first $k + 1$ points:

  $$P_k(x_i) = y_i \quad \forall i \in \{0, 1, \ldots, k\}.$$ 

(continue to next slide)
Proof – existence.

We do this by setting

\[
P_k(x) = P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j).
\]

Now, \( \prod_{j=0}^{k-1} (x - x_j) \) is a polynomial, of degree \( k \), that vanishes on \( (x_j)_{j=0}^{k-1} \). So \( P_k \) interpolates the first \( k \) points, and its degree is no higher than \( k \).

We still need it to interpolate \((x_k, y_k)\), the \((k + 1)\)-th point. As for \( C_k \):

\[
y_k^{\text{required}} = P_k(x_k) = P_{k-1}(x_k) + C_k \prod_{j=0}^{k-1} (x_k - x_j)
\]

\[\Rightarrow C_k = \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1} (x_k - x_j)} \neq 0\]
The proof for existence was also constructive, in the sense that it gave a concrete formulate for the interpolation polynomial:
Definition (Newton’s form of the interpolation polynomial)

The form we saw in the proof,

\[ P_k(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \ldots + C_k(x - x_0) \cdots (x - x_{k-1}), \]

is called Newton’s form of the interpolation polynomial.
Polynomial Interpolation

\[ P_k(x) = \begin{cases} 
C_0 & k = 0 \\
P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j) & k \in \{1, \ldots, n\}
\end{cases} \]

where \[ C_k = \begin{cases} 
y_0 & k = 0 \\
y_k - P_{k-1}(x_k) \prod_{j=0}^{k-1} (x_k - x_j) & k \in \{1, \ldots, n\}
\end{cases} \]

**Example (n = 0 (a single point))**

Given a single point, \((x_0, y_0)\), the interpolation polynomial is

\[ P_0(x) = C_0 = y_0 \]
Polynomial Interpolation

\[ P_k(x) = \begin{cases} 
C_0 & k = 0 \\
P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j) & k \in \{1, \ldots, n\}
\end{cases} \]

where

\[ C_k = \begin{cases} 
y_0 & k = 0 \\
\frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1} (x_k - x_j)} & k \in \{1, \ldots, n\}
\end{cases} \]

Example \((n = 1\ (\text{two points}))\)

Given two points, \(\{(x_i, y_i)\}_{i=0}^{1}\), the interpolation polynomial is (the line)

\[ P_1(x) = C_0 + C_1(x - x_0) = \frac{C_0}{y_0} + \frac{C_1}{y_1 - y_0} \cdot (x - x_0) \]

\[ P_1(x_0) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cdot (x_0 - x_0) = y_0 \]

\[ P_1(x_1) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cdot (x_1 - x_0) = y_1 \]
The Interpolation Polynomial by Solving a Linear System

Here is another way to derive the interpolation polynomial.

The $n + 1$ equations,

$$P_n(x_i) = \sum_{j=0}^{n} a_j x_i^j = \begin{bmatrix} 1 & x_i & x_i^2 & \cdots & x_i^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = y_i, \ i = 0, 1, \ldots, n$$

are linear w.r.t. the $a_i$'s:

$$\begin{bmatrix} \sum_{j=0}^{n} a_j x_0^j \\ \sum_{j=0}^{n} a_j x_1^j \\ \vdots \\ \sum_{j=0}^{n} a_j x_n^j \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$
So can solve via $\mathbf{a} = \mathbf{V}^{-1}\mathbf{y}$.

$\mathbf{V}$ is called the Vandermonde Matrix.

Assuming the $x_i$’s are distinct, we know that $\mathbf{V}$ is guaranteed (by the theorem) to be invertible.

By uniqueness, we know the solution will coincide with Newton’s form.
Problem with This Method

- **Computationally expensive.** Matrix inversion is expensive; that said, in the remainder of the course we will study practical methods to solve a linear system without inversion.

- **Ill-conditioned.** The values of \( a = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_n \end{bmatrix}^T \) might be determined inaccurately. We will later see how to characterize the condition number of a linear system.
We again emphasize that the interpolation polynomial is unique: regardless whether it is expressed via

1

\[ P_n(x) = \sum_{j=1}^{n} a_j x_i^j \]

or

2

\[ P_n(x) = C_0 + \sum_{k=1}^{n} C_k \prod_{j=0}^{k-1} (x - x_j) \]

it is the same polynomial.
Remark

- Recall that one advantage of polynomials is the ease of their evaluations.
- By inspection of $P_n(x) = \sum_{j=0}^{n} a_j x^j$, it seems that $x \mapsto P_n(x)$ requires $n$ additions and $\sum_{j=1}^{n} j = n \frac{n+1}{2}$ multiplications. This is $O(n^2)$.
- It is possible, however, to reduce the # of operations to $O(n)$, using Horner’s rule (see next slide).
Horner’s Rule (AKA Nested Polynomials)

- Set $b_n \triangleq a_n$, and then, iteratively, set $b_{k-1} \triangleq a_{k-1} + b_k x$ till obtaining $b_0 = a_0 + b_1 x$ and this is equal to $P_n(x)$.

Example

$$P_4(x) = \sum_{j=0}^{4} a_j x^j = a_0 + x (a_1 + x (a_2 + x (a_3 + a_4 x)))$$

- This gives $O(n)$ additions and $O(n)$ multiplications (instead of $O(n)$ and $O(n^2)$).
Since the interpolation polynomial is unique, if the $n + 1$ points were sampled from a polynomial of degree $n$, then we will recover that polynomial.
Example

Let $P(x) = 2x^3 + 3x^2 - 4x - 5$. Consider the following 4 sampled points.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$x_i$</th>
<th>$y_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>64</td>
</tr>
</tbody>
</table>

Reminder: Newton’s form

$$P_k(x) = \begin{cases} 
C_0 & k = 0 \\
 P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j) & k \in \{1, \ldots, n\}
\end{cases}$$

where $C_k = \begin{cases} 
y_0 & k = 0 \\
 \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1} (x_k - x_j)} & k \in \{1, \ldots, n\}
\end{cases}$

$$P_0(x) \equiv C_0 = y_0 = 0$$
Example (continued)

Let \( P(x) = 2x^3 + 3x^2 - 4x - 5 \). Consider the following 4 sampled points.

\[
\begin{array}{cccc}
  i = 0 & i = 1 & i = 2 & i = 3 \\
  x_i & -1 & 0 & 2 & 3 \\
  y_i & 0 & -5 & 15 & 64 \\
\end{array}
\]

Reminder: Newton’s form

\[
P_k(x) = \begin{cases} 
  C_0 & k = 0 \\
  P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j) & k \in \{1, \ldots, n\}
\end{cases}
\]

where \( C_k = \begin{cases} 
  y_0 & k = 0 \\
  \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1} (x_k - x_j)} & k \in \{1, \ldots, n\}
\end{cases} \)

\( P_0(x) \equiv 0 \) (saw in the previous slide)

\[
C_1 = \frac{y_1 - P_0(x_1)}{x_1 - x_0} = \frac{-5}{0+1} = -5
\]

\[
P_1(x) = P_0(x) + C_1(x - x_0) = 0 - 5(x + 1) = -5x - 5
\]
Example (continued)

Let \( P(x) = 2x^3 + 3x^2 - 4x - 5 \). Consider the following 4 sampled points.

<table>
<thead>
<tr>
<th>( i )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_i )</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>( y_i )</td>
<td>0</td>
<td>-5</td>
<td>15</td>
<td>64</td>
</tr>
</tbody>
</table>

Reminder: Newton's form

\[
P_k(x) = \begin{cases} 
C_0 & k = 0 \\
P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j) & k \in \{1, \ldots, n\} 
\end{cases}
\]

where \( C_k = \begin{cases} 
y_0 & k = 0 \\
\frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1} (x_k - x_j)} & k \in \{1, \ldots, n\} 
\end{cases} \)

\[
P_1(x) = -5x - 5 \text{ (saw in the previous slide)}
\]

\[
C_2 = \frac{y_2 - P_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} = \frac{15 - (-10 - 5)}{(2 + 1)(2)} = \frac{30}{6} = 5
\]

\[
P_2(x) = P_1(x) + C_2(x - x_0)(x - x_1) = 5x^2 - 5
\]
Example (continued)

Let \( P(x) = 2x^3 + 3x^2 - 4x - 5 \). Consider the following 4 sampled points.

| \( i \) | \( 0 \) | \( 1 \) | \( 2 \) | \( 3 \) |
|---|---|---|---|
| \( x_i \) | -1 | 0 | 2 | 3 |
| \( y_i \) | 0 | -5 | 15 | 64 |

Reminder: Newton’s form

\[
P_k(x) = \begin{cases} 
  C_0 & k = 0 \\
  P_{k-1}(x) + C_k \prod_{j=0}^{k-1} (x - x_j) & k \in \{1, \ldots, n\} 
\end{cases}
\]

where \( C_k = \begin{cases} 
  y_0 & k = 0 \\
  \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1} (x_k - x_j)} & k \in \{1, \ldots, n\} 
\end{cases} \)

\( P_2(x) = 5x^2 - 5 \) (saw in the previous slide)

\[
C_3 = \frac{y_3 - P_2(x_3)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} = \frac{64 - (5 \cdot 9 - 5)}{(3+1)(3)(3-2)} = \frac{24}{12} = 2
\]

\[
P_3(x) = P_2(x) + C_3(x - x_0)(x - x_1)(x - x_2) = 5x^2 - 5 + 2(x + 1)(x)(x - 2)
\]

\[
= 5x^2 - 5 + (2x^2 + 2x)(x - 2) = 5x^2 - 5 + 2x^3 + 2x^2 - 4x^2 - 4x
\]

\[
= 2x^3 + 3x^2 - 4x - 5 \text{ as expected.}
\]
Polynomial Interpolation

\( P_n(x) \) as a Linear combination of Basis Functions

- An alternative way to represent the interpolation polynomial is as the following linear combination of basis functions.

\[
P_n(x) = \sum_{i=0}^{n} y_i L_i(x)
\]

where \((L_i(x))_{i=0}^{n}\) are polynomials that depend on the values of \(x_i\)'s but not on the values of the \(y_i\)'s.

- Since we can’t control the \(y_i\)'s, the most general constraint on \(L_i(x)\) is:

\[
L_i(x_j) = \delta_{ij} \overset{\Delta}{=} \begin{cases} 
0 & i \neq j \\
1 & i = j 
\end{cases}
\text{(Kronecker delta)}
\]

together with the restriction that \(L_i(x)\) is a polynomial of degree no higher than \(n\).
Polynomial Interpolation

\( P_n(x) \) as a Linear combination of Basis Functions

\[
P_n(x) = \sum_{i=0}^{n} y_i L_i(x)
\]

The constraint “\( i \neq j \Rightarrow L_i(x_j) = 0 \)” implies the following form:

\[
L_i(x) = C_i \prod_{j:j \neq i} (x - x_j)
\]

where the value of \( C_i \) is found via

\[
L_i(x_i) = 1 = C_i \prod_{j:j \neq i} (x_i - x_j) \Rightarrow C_i = \prod_{j:j \neq i} (x_i - x_j)^{-1}
\]
In other words, we have just derived the family of the basis functions, $(L_i(x))_{i=0}^{n}$:

$$L_i(x) = \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j} = \left( \prod_{j=0}^{i-1} \frac{x - x_j}{x_i - x_j} \right) \left( \prod_{j=i+1}^{n} \frac{x - x_j}{x_i - x_j} \right)$$

And as mentioned earlier, the polynomial

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

satisfies

$$P_n(x_i) = y_i$$

so by uniqueness of the interpolation polynomial, it is the same polynomial we saw in other forms.
This is Lagrange’s form of the interpolation polynomial. It is often called Lagrange Approximation, but of course if the data came from a polynomial (of degree no higher than \( n \)) then it is exact.

Remark: This form is easy to construct but is expensive to evaluate.
Example

Again let \( P(x) = 2x^3 + 3x^2 - 4x - 5 \) and consider

\[
\begin{array}{c|cccc}
  x_i & i = 0 & i = 1 & i = 2 & i = 3 \\
  y_i & -1 & 0 & 2 & 3 \\
\end{array}
\]

Lagrange’s form of the interpolation polynomial is given by

\[
P_n(x) = \sum_{i=0}^{3} y_i L_i(x) = 0 \cdot L_0(x) - 5L_1(x) + 15L_2(x) + 64L_3(x)
\]

\( L_0(x) = \) don’t bother since \( y_0 = 0 \)

\[L_1(x) = \frac{(x+1)(x-2)(x-3)}{(0+1)(0-2)(0-3)} = \frac{1}{6} (x + 1)(x - 2)(x - 3)\]

\[L_2(x) = \frac{(x+1)x(x-3)}{(2+1)(2-0)(2-3)} = -\frac{1}{6} (x + 1)x(x - 3)\]

\[L_3(x) = \frac{(x+1)x(x-2)}{(3+1)(3-0)(3-2)} = \frac{1}{12} (x + 1)x(x - 2)\]

\[P_n(x) = \frac{-5}{6} (x + 1)(x - 2)(x - 3) - \frac{15}{6} (x + 1)x(x - 3) + \frac{32}{6} (x + 1)x(x - 2)\]

\[= \ldots = 2x^3 + 3x^2 - 4x - 5 \text{ as expected.}\]
The Interpolation Polynomial’s Error

- Our examples so far involved the (exact) recovering of a polynomial.
- Polynomial interpolation, however, can also be used to approximate non-polynomials.
Example (approximating sin(x))

Let \( f(x) = \sin(x) \). It is easy to evaluate \( f \) in several key points.

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( i = 0 )</th>
<th>( i = 1 )</th>
<th>( i = 2 )</th>
<th>( i = 3 )</th>
<th>( i = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_i )</td>
<td>(-\pi)</td>
<td>(-\pi/2)</td>
<td>0</td>
<td>( \pi/2 )</td>
<td>( \pi )</td>
</tr>
<tr>
<td>( y_i )</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Can now apply the tools we learned to find a polynomial, of degree \( \leq 4 \), that approximates \( \sin(x) \) on \([-\pi, \pi]\). We will get the following result:

\[
P_4(x) = \frac{-8}{3\pi^3} x^3 + \frac{8}{3\pi} x
\]

Exercise

Verify this at home.

A natural question: how good is this approximation?
Error Analysis

- More generally, how can we quantify the interpolation error?
- Can we derive an upper bound on it?
Theorem

Let \( f \in C^{n+1}[a, b] \) and let \( P_n(x) \) be the interpolation polynomial of \( f \) at nodes \((x_i)_{i=0}^n \subset [a, b]\). Then, the interpolation error at \( x \in [a, b] \) is given by

\[
E_n(x) \triangleq f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{i=0}^{n} (x - x_i)
\]

for some \( \xi \in [a, b] \).
Before the Proof...

\[ E_n(x) \triangleq f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \]

This theoretical result is of paramount importance since it lets us know, a-priori, the expected maximal error and enables us to design an interpolation polynomial to meet a given accuracy criterion.
Example

Find the degree of the interpolation polynomial, \( P_n(x) \) that will guarantee that approximation error will be \( \leq 10^{-5} \) for \( f(x) = \sin(x) \) on \([-\pi, \pi]\) (even w/o knowing the nodes, \( (x_i)_{i=1}^n \)).

Solution:

\[
|E_n(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \right| \\
\leq \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \right| (2\pi)^{n+1} \leq \frac{(2\pi)^{n+1}}{(n+1)!} \text{ required} \leq 10^{-5}
\]

By search we can find \( |E_{25}(x)| \leq M \approx 5.8 \cdot 10^{-6} \). Note, however, that by using a better placement of the nodes, we will be able to get the desired accuracy using a much smaller \( n \).
Before the proof of the theorem, we need a lemma.

**Lemma**

If \( g \in C^{k-1}[a, b] \) has \( k \) roots in \([a, b]\) then \( g^{(k-1)} \) has at least one root in \([a, b]\).

**Proof of the lemma.**

By the Mean Value Theorem:

- \( g' \) has at least \( k - 1 \) roots in \([a, b]\).
- \( g'' \) has at least \( k - 2 \) roots in \([a, b]\).

\[ \vdots \]

- \( g^{k-1} \) has at least one root in \([a, b]\) (note that \( k - (k - 1) = 1 \)).
Proof of the theorem

If \( x = x_i \), then the claim holds. We only need to prove it for the case where \( x \neq x_i, \forall i \in \{0, 1, \ldots, n\} \). Define a new function whose argument is \( t \in [a, b] \), and for which \( x \) is a constant parameter:

\[
    t \mapsto g(t; x) = \frac{E_n(t)}{f(t) - P_n(t)} - \frac{E_n(x)}{f(x) - P_n(x)} \prod_{j=0}^{n} \frac{t - x_j}{x - x_j}
\]

Note that \( t \mapsto g(t; x) \in C^{n+1}[a, b] \). This \( g \) was designed to satisfy the following property: it has \( n + 2 \) distinct roots. To see that, take \( t = x \):

\[
    g(x; x) = E_n(x) - E_n(x) \prod_{j=0}^{n} \frac{x - x_j}{x - x_j} = E_n(x) - E_n(x) = 0.
\]

Thus, \( x \) is a root of \( t \mapsto g(t; x) \). Moreover, each of the \( x_i \)'s is a root: (the proof continues on the next slide)
Analyzing the Error in Polynomial Interpolation

Proof of the theorem – continued.

\[ g(x_i; x) = E_n(x_i) - E_n(x) \prod_{j=0}^{n} \frac{x_i - x_j}{x - x_j} = 0 - E_n(x) \cdot 0 = 0. \]

Together, this means \( g \) has \( n + 2 \) distinct roots. By the lemma, \( g^{(n+1)} \) has at least one root: \( \exists \xi \in [a, b] \) such that \( g^{(n+1)}(\xi; x) = 0 \). Inspect this derivative:

\[
g^{(n+1)}(t; x) = E^{(n+1)}_n(t) - E_n(x) \frac{d^{n+1}}{dt^{n+1}} \prod_{j=0}^{n} \frac{t - x_j}{x - x_j}
\]

\[
= f^{(n+1)}(t) - \underbrace{P^{(n+1)}_n(t)}_{0} - E_n(x) \frac{1}{\prod_{j=0}^{n}(x - x_j)} \frac{d^{n+1}}{dt^{n+1}} t^{n+1}
\]

\[
= f^{(n+1)}(t) - E_n(x) \frac{(n + 1)!}{\prod_{j=0}^{n} x - x_j}
\]

\[
0 = g^{(n+1)}(\xi; x) = f^{(n+1)}(\xi) - E_n(x) \frac{(n + 1)!}{\prod_{j=0}^{n}(x - x_j)} \Rightarrow
\]

\[
E_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{j=0}^{n} (x - x_j)
\]
Connection to Taylor’s Expansion

- The expression,
  \[ E_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{j=0}^{n} (x - x_j) \]
  
  is similar to the error from Taylor’s expansion:
  \[ \frac{f^{(n+1)}(\xi)}{(n + 1)!} (x - x_0)^{n+1} \]

- Taylor’s expansion approximates a function using a single point (while here we use \(n + 1\) points). Thus, we can interpret the Taylor expansion as taking the limits of the distances between all our \(n + 1\) points to zero, collapsing them to a single point.
Bounding the Error

- We are now in a position to discuss a bound on the error.
- Clearly, the theorem is useful only if we can bound $f^{(n+1)}(x)$.
- If we can assume, or prove, a bound on $|f^{(n+1)}(x)|$ (as we could in the example with $f(x) = \sin(x)$),

\[
|f^{(n+1)}(x)| < M_{n+1} \quad \forall x \in [a, b]
\]

then

\[
|E_n(x)| = |f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n + 1)!} \max_{x \in [a, b]} \prod_{j=0}^{n} |x - x_j|
\]
Bounding the Error

- **However**, we can’t always assume or prove such a bound!
- Thus, in such cases, there is no guarantee that increasing $n$ will decrease a bound on the error. In fact, can’t even guarantee that $\lim_{n \to \infty} E_n = 0$. 
Evenly-spaced Nodes

- Often, it is convenient to sample the function at evenly-spaced points:

  \[ x_i = x_0 + i \cdot h \quad \forall i = 1, \ldots, n \]

- In which case, it is possible to derive more convenient error bounds.
Example

Consider a quadratic interpolation on \([a, b]\), where \(x_0 = a\), \(x_1 = a + h\), and \(x_2 = a + 2h = b\). By the theorem, if \(M_3\) is an upper bound on \(|f^{(3)}(x)|\),

\[
E_2(x) \leq \frac{M_3}{(3)!} \max_{x \in [a, b]} |(x - x_0)(x - x_1)(x - x_2)| \triangleq \frac{M_3}{3!} \max_{x \in [a, b]} |H(x)|
\]

\[
H(x) = (x - x_0)(x - x_1)(x - x_2) = \ldots = x^3 - x^2(x_0 + x_1 + x_2) - x(x_0x_1 + x_0x_2 + x_1x_2) - x_0x_1x_2
\]

Looking for the extremum, we set

\[
H'(x) = 3x^2 - 2x(x_0 + x_1 + x_2) - (x_0x_1 + x_0x_2 + x_1x_2) = 0.
\]

Now substitute \(x_i = x_0 + i \cdot h\) and obtain

\[
0 = 3x^2 - 6x(x_0 + h) + 3(x_0^2 + 2x_0h + \frac{2}{3}h^2)
\]

\[
x_{\text{max}} = \frac{6(x_0 + h) \pm \sqrt{36(x_0 + h)^2 - 36(x_0^2 + 2x_0h + \frac{2}{3}h^2)}}{6} = x_0 + h \pm \sqrt{\frac{1}{3}h^2}
\]

\[
= x_0 + h(1 \pm \frac{1}{\sqrt{3}})
\]

(continue next slide)
Example (continued)

Going back to $\max_{x \in [a,b]} |H(x)|$, we get

$$H_{\text{max}} = (x_{\text{max}} - x_0)(x_{\text{max}} - x_0 - h)(x_{\text{max}} - x_0 - 2h)$$

$$= h(1 \pm \frac{1}{\sqrt{3}})h(\pm \frac{1}{\sqrt{3}})h(\pm \frac{1}{\sqrt{3}} - 1)$$

$$= h^3 \frac{\sqrt{3} + 1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{\sqrt{3} - 1}{\sqrt{3}} = \frac{2}{3\sqrt{3}} h^3$$

$$\Rightarrow |E_2(x)| \leq \frac{M_3}{3!} \max_{x \in [a,b]} |H(x)| = \frac{h^3}{9\sqrt{3}} M_3$$
More generally:

**Theorem**

Let \( f \in C^{n+1}([a, b]) \) and let \( P_n(x) \) be its interpolation polynomial for the evenly-spaced nodes \( x_i = X_0 + ih, \ i = 0, 1, \ldots, n \). If

\[
|f^{(n+1)}(x)| \leq M_{n+1} \quad \forall x \in [a, b]
\]

then

\[
|E_n(x)| = |f(x) - P_n(x)| \leq O(h^{n+1})M_{n+1}
\]

We omit the proof. Some particular cases:

- \(|E_0(x)| \leq hM_1\)
- \(|E_1(x)| \leq \frac{1}{8}h^2M_2\)
- \(|E_2(x)| \leq \frac{1}{9\sqrt{3}}h^3M_3\)
- \(|E_3(x)| \leq \frac{1}{24}h^4M_4\)
Reminder: increasing the number of nodes does not guarantee convergence of $P_n(x)$ to $f(x)$.

This is true even if the points are evenly spaced, despite the fact that

$$|E_n(x)| \leq O(h^{n+1}) M_{n+1}$$

since $M_{n+1}$, the bound on $|f^{(n+1)}(x)|$, might increase faster than the decrease of $O(h^{n+1})$ so the error might, in fact, increase.

This is the notorious Runge’s phenomenon, expressed via sharp oscillations near the interval’s endpoints.
Optimal Placement of the Roots

Optimizing the Locations of the Nodes

- We saw

\[
E_n(x) \triangleq f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{i=0}^{n} (x - x_i)
\]

- In the product, \( \prod_{i=0}^{n} (x - x_i) \), both \( n \) and the locations of the nodes play a role. Thus, for a fixed \( n \), a smart choice of the locations can greatly influence the maximal error.

- If we define

\[
Q(x) = \prod_{i=0}^{n} (x - x_i)
\]

then, for a fixed \( n \), the optimal choice of \( (x_i)_{i=0}^{n} \) is the one that will minimize the maximum of \( |Q(x)| \):

\[
\max_{x \in [a,b]} |Q(x)| \Rightarrow \text{minimum}
\]
So here is the plan.

We will build a certain polynomial, of degree $n$, and will get an analytic expression for its roots.

As it will turn out, these roots will minimize $\max_{x \in [a,b]} |Q(x)|$.

This method is due to Chebyshev.

The derivation in the next slides is based on Burden and Faires’ textbook.
We will start by assuming our interval is $[-1, 1]$. This will be generalized later.

Define, for any nonnegative integer $n$,

$$T_n : [-1, 1] \to \mathbb{R} \quad T_n : x \mapsto \cos(n \arccos(x)) \quad n \geq 0$$

It is not obvious, but $T_n(x)$ is in fact a polynomial in $x$, as we now show.
\[ T_n(x) = \cos(n \arccos(x)) \]

- First,
  \[ T_0(x) = \cos(0 \arccos(x)) = \cos(0) = 1. \]
  Thus, \( T_0(x) \) is a zeroth-order polynomial.

- Second,
  \[ T_1(x) = \cos(1 \arccos(x)) = \cos(\arccos(x)) = x. \]
  Thus, \( T_1(x) \) is a first-order polynomial.

- So only need to show \( T_n \) is a polynomial for the case \( n > 1 \).
Optimal Placement of the Roots

\[ T_n(x) = \cos(n \arccos(x)) \]

Let

\[ \theta \triangleq \arccos(x) \]

Thus,

\[ \cos(\theta) = x \]

The expression becomes

\[ \hat{T}_n(\theta) \triangleq T_n(\cos(\theta)) = \cos(n \arccos(\cos(\theta))) = \cos(n\theta) \quad \theta \in [0, \pi] \]

Observe:

\[ \hat{T}_{n+1}(\theta) = \cos((n + 1)\theta) = \cos(\theta) \cos(n\theta) - \sin(\theta) \sin(n\theta) \]

\[ \hat{T}_{n-1}(\theta) = \cos((n - 1)\theta) = \cos(\theta) \cos(n\theta) + \sin(\theta) \sin(n\theta) \]

Adding

\[ \Rightarrow \hat{T}_{n+1}(\theta) = 2 \cos(\theta) \cos(n\theta) - \hat{T}_{n-1}(\theta) \]

or

\[ T_{n+1}(x) = 2x \cos(n \arccos(x)) - T_{n-1}(x) \]

\[ \Rightarrow T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \]
Optimal Placement of the Roots

Just saw:

\[
T_0(x) = 1 \\
T_1(x) = x \\
T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad n > 1
\]

It follows that \( T_n(x) \) is a polynomial of order \( n \).

These polynomials are called **Chebysehv polynomials**.
Chebyshev Polynomials

\[ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad n > 1 \]
\[ T_0(x) = 1 \]
\[ T_1(x) = x \]
\[ T_2(x) = 2x^2 - 1 \]
\[ T_3(x) = 4x^3 - 3x \]
\[ T_4(x) = 8x^4 - 8x^2 + 1 \]

\[ \vdots \]

The recurrence also implies that \( T_n(x) \) is a polynomial of degree \( n \) whose leading coefficient is \( 2^{n-1} \).
Roots of $T_n(x)$

**Theorem**

The Chebyshev polynomial $T_n(x)$ of degree $n \geq 1$ has $n$ simple roots in $[1, 1]$ at

$$
\bar{x}_k = \cos \left( \frac{2k - 1}{2n} \pi \right) \quad \text{for each } k = 1, 2, \ldots, n.
$$
Optimal Placement of the Roots

Proof.

Let

$$\bar{x}_k = \cos \left( \frac{2k - 1}{2n} \pi \right) \quad \text{for each } k = 1, 2, \ldots, n.$$  

Then,

$$T_n(\bar{x}_k) = \cos(n \arccos(\bar{x}_k)) = \cos \left( n \arccos \left( \cos \left( \frac{2k - 1}{2n} \pi \right) \right) \right)$$

$$= \cos \left( n \frac{2k - 1}{2n} \pi \right) = \cos \left( \frac{2k - 1}{2} \pi \right) = 0$$

(since we get $\frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{2}, \ldots$)

So we got $n$ such roots. Since these roots are distinct, and $T_n$ is of degree $n$, these are all its roots.
Extreme Points of $T_n(x)$

**Theorem**

The Chebyshev polynomial $T_n(x)$ of degree $n \geq 1$ assumes its absolute extrema at

$$\bar{x}_k' = \cos \left( \frac{k\pi}{n} \right) \quad \text{with} \quad T_n(\bar{x}_k') = (-1)^k \quad \text{for each} \quad k = 0, 1, \ldots, n.$$
Proof.

\[
\frac{d}{dx} T_n(x) = \frac{d}{dx} \left( \cos(n \arccos(x)) \right) = -\sin(n \arccos(x)) \frac{d}{dx} \left( n \arccos(x) \right)
\]

\[
= -n \sin(n \arccos(x)) \frac{d}{dx} \left( \arccos(x) \right)
\]

\[
= -n \sin(n \arccos(x)) \left( -\frac{1}{\sqrt{1-x^2}} \right) = \frac{n \sin(n \arccos(x))}{\sqrt{1-x^2}}
\]

For \( k = 1, 2, \ldots, n - 1 \):

\[
\frac{d}{dx} T_n(\bar{x}'_k) = \frac{n \sin(n \arccos(\cos(\frac{k\pi}{n})))}{\sqrt{1 - \left( \cos(\frac{k\pi}{n}) \right)^2}} = \frac{n \sin(k\pi)}{\sin(\frac{k\pi}{n})} = 0
\]

\( T_n \) is a polynomial of degree \( n \Rightarrow \frac{d}{dx} T_n \) is a polynomial of degree \( n - 1 \). And we just found \( n - 1 \) distinct roots of the latter, so it has no others. The only other extrema of \( T_n \) can occur at the endpoints, \( \pm 1 \). And these coincide with \( \bar{x}'_0 = -1 \) and \( \bar{x}'_n = 1 \). Finally, for \( k = 0, 1, \ldots, n \),

\[
T_n(\bar{x}_k) = \cos \left( n \arccos \left( \cos \left( \frac{k\pi}{n} \right) \right) \right) = \cos(k\pi) = (-1)^k.
\]
The Monic Chebysehv Polynomials

The Monic (polynomials whose leading coefficient is 1) Polynomials, denoted by $\tilde{T}_n$, derived from the Chebysehv Polynomials, $T_n$, by dividing the latter by $2^{n-1}$:

$$\tilde{T}_0 \equiv 1 \quad \text{and} \quad \tilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x) \quad \text{for each } n \geq 1.$$

It follows that

$$\tilde{T}_2(x) = x\tilde{T}_1(x) - \frac{1}{2} \tilde{T}_0(x) \quad \text{and}$$

$$\tilde{T}_{n+1}(x) = x\tilde{T}_n(x) - \frac{1}{4} \tilde{T}_{n-1}(x) \quad \text{for each } n \geq 2.$$
Since $\tilde{T}_n$ is just a multiple of $T_n$, the zeros of $\tilde{T}_n$ (for $n \geq 1$, occur at

$$\bar{x}_k = \cos \left( \frac{2k - 1}{2n} \pi \right) \quad \text{for each } k = 1, 2, \ldots, n,$$

and its extreme values, for $n \geq 1$, occur at

$$\bar{x}'_k = \cos \left( \frac{k}{n} \pi \right) \quad \text{for each } k = 1, 2, \ldots, n,$$

with

$$\tilde{T}_n(\bar{x}'_k) = \frac{(-1)^k}{2n-1}.$$
Let
\[ \tilde{\Pi}_n \]
denote the set of all monic polynomials of degree \( n \) on \([-1, 1]\).

As we will see, the fact that the extreme values of Chebysehv’s monic polynomials, \( \tilde{T}_n \) (for \( n \geq 1 \)), are
\[ \pm \frac{1}{2^{n-1}} , \]
will lead to an important minimization property that sets Chebysehv’s monic polynomials apart from other members of \( \tilde{\Pi}_n \).
Theorem

The polynomials of the form $\tilde{T}_n$, when $n \geq 1$, satisfy the following property:

$$\frac{1}{2^{n-1}} = \max_{x \in [-1,1]} |\tilde{T}_n(x)| \leq \max_{x \in [-1,1]} |P_n(x)|, \forall P_n \in \tilde{\Pi}_n.$$

Moreover, equality holds only if $P_n \equiv \tilde{T}_n$. 
Proof.

Let $P_n \in \tilde{\Pi}_n$. Suppose that

\[ \max_{x \in [-1,1]} |P_n(x)| \leq \frac{1}{2n-1} = \max_{x \in [-1,1]} |\tilde{T}_n(x)|. \]

Let $Q = \tilde{T}_n - P_n$. Since $\tilde{T}_n$ and $P_n$ are monic polynomials, $Q$ is a polynomial of degree $\leq n - 1$. At $\bar{x}'_k$, the extreme points of $\tilde{T}_n$, we have

\[ Q(\bar{x}'_k) = \tilde{T}_n(\bar{x}'_k) - P_n(\bar{x}'_k) = \frac{(-1)^k}{2n-1} - P_n(\bar{x}'_k). \]

However, by assumption,

\[ |P_n(\bar{x}'_k)| \leq \frac{1}{2n-1} \quad \text{for each } k = 0, 1, \ldots, n. \]

So $Q(\bar{x}'_k) \leq 0$ when $k$ is odd $Q(\bar{x}'_k) \geq 0$ when $k$ is even. By continuity of $Q$, for each $j = 0, 1, \ldots, n - 1$, $Q$ has at least one root between $\bar{x}'_j$ and $\bar{x}'_{j+1}$ $\Rightarrow$ $Q$ has at least $n$ roots in $[-1, 1]$. It follows that $Q \equiv 0$. \qed
Recall the quantity of interest is $|Q(x)| = \prod_{i=0}^{n} |x - x_i|$

$Q(x) = \prod_{i=0}^{n} (x - x_i)$ is a monic polynomial of order $n + 1$.

Based on what we just saw, the maximal value of $|Q(x)|$ is smallest when the $(x_k)^n_{k=0}$ are chosen to be roots of the Chebyshev polynomial, $\tilde{T}_{n+1}(x)$; i.e., when

$$\prod_{i=0}^{n} |x - x_i| = \tilde{T}_{n+1}(x)$$

Hence we choose

$$\bar{x}_{k+1} \cos \left( \frac{2k + 1}{2n} \pi \right)$$

for each $k = 0, 1, \ldots, n$. 
We also know that, since $\max_{x \in [-1,1]} \tilde{T}_{n+1}(x) = 2^{-n}$,

$$2^{-n} = \max_{x \in [-1,1]} \prod_{i=1}^{n+1} |x - \bar{x}_i| \leq \max_{x \in [-1,1]} \prod_{i=0}^{n} |x - x_i|$$

for any choice of $(x_i)_{i=0}^{n} \subset [-1, 1]$. 
Corollary

Let \( P_n(x) \) be the interpolation polynomial of degree \( \leq n \) with nodes at the roots of \( T_{n+1} \). Then

\[
\max_{x \in [-1,1]} |f(x) - P_n(x)| \leq \frac{1}{2^n (n + 1)!} \max_{x \in [-1,1]} |f^{(n+1)}(x)|
\]

for every \( f \in C^{(n+1)}([-1,1]) \).
Generalizing from $[-1, 1]$ to $[a, b]$

Set

$$x \mapsto \frac{1}{2}[(b - a)x + a + b]$$

to transform the numbers $\bar{x}_k$ from $[-1, 1]$ to numbers in $[a, b]$. 

Version Log

- 19/11/2019, ver 1.00.