

Lecture 09 - C

§4.2,6.1 Interpolation Error

T. Takahashi

Department of Mathematics and Statistics
University of New Mexico



We will answer these questions:

- How well does interpolation approximate a function?
- When does polynomial interpolation lead to disaster?
- What are Chebyshev nodes and how are they related to interpolation?
- What are splines? In what way are they advantageous over standard polynomial interpolation?



Quantifying Error for Interpolation

Question

How can we quantify how well of an interpolation $p(x)$ approximates the unknown function $f(x)$?

In polynomial interpolation, we assume that the data points (x_i, y_i) come from some unknown continuous function $f(x)$

Simple answer: We can choose our own function $f(x)$ and create our data points $e(x) = f(x) - p_n(x)$.



Quantifying Error for Interpolation

Strategy

Strategy for quantifying error :

- 1 Choose a function $f(x)$
- 2 Choose $n + 1$ values of x to get the x_i
- 3 Generate data points (x_i, y_i) where $y_i = f(x_i)$
- 4 Interpolate (x_i, y_i) to get our polynomial $p(x)$
- 5 Calculate the error $e(x) = f(x) - p(x)$

Definition: Sampling

Extracting the value of $y = f(x)$ at finite discrete points is called **sampling**.

Question

How do choose the sample points?

Evenly Distributed Sample Points

- To get $n + 1$ equally spread out over $[a, b]$, we first break up $[a, b]$ into n subintervals

$$h = \frac{b - a}{n}$$

- The first point is always $x_0 = a$ and the last is always $x_{n+1} = b$
- The points in between are given by $x_i = x_0 + ih$

Method: Equispaced Discretization

- ▶ **Given:** Interval $[a, b]$, Number of points $n + 1$
- ▶ **Output:** Set of equally spaced out points $\{x_i\}$
- ▶ **Idea:** Use the constant space $h = (b - a)/n$ to determine the points.

- In Matlab, this is simply

```
>> xx = linspace(a,b,n+1);
```

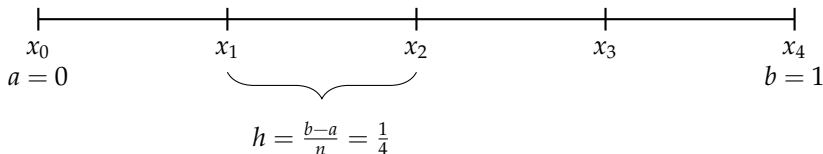


Discretization Example

Example

Discretize the interval $[0, 1]$ into equispaced 5 points.

- Left-end $a = 0$ and right-end $b = 1$ are included as part of the 5 points.
- Number of subintervals is $n = 5 - 1 = 4$.
- Can be helpful to draw the line interval



- Based on the drawing, our discretization is $\{0, 0.25, 0.5, 0.75, 1.0\}$
- We can also use the formula $x_k = x_0 + h k = 0 + 0.25 k$



Original vs. Interpolation

- Let $f(x)$ be our test function sampled on $[a, b]$
- Let $g(x)$ be our interpolation using data from $(a, f(a)), \dots, (b, f(b))$

Visual Error

Plot $f(x)$ and $g(x)$ together and see the difference.

Max Discrepancy

We may define

$$E = \max_{x \in (a,b)} |f(x) - p(x)|$$



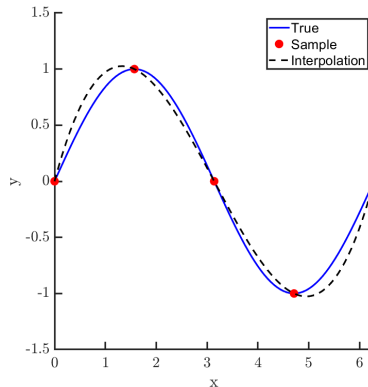
Interpolate $y = \sin(x)$ with $n = 4$

- Consider sampling $f(x) = \sin(x)$ over $[0, 2\pi]$ with $n = 4$

- Interpolation is roughly

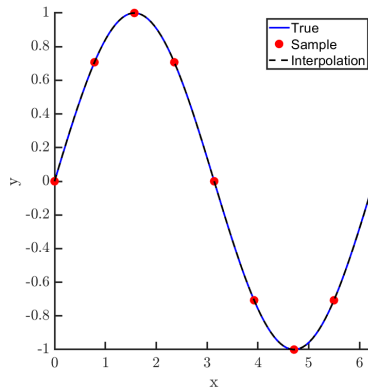
$$p(x) \approx 1.697x - 0.8106x^2 + 0.086x^3$$

- There is visual difference between the true function (blue curve) and the interpolation (dashed curve)



Interpolate $y = \sin(x)$ with $n = 8$

- Same interpolation with
 $n = 8$
- No visual error on $[0, \pi]$!
- $n = 8$ looks better than $n = 4$



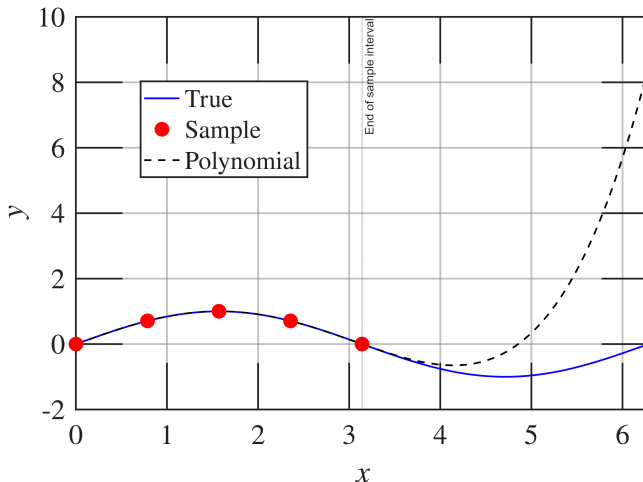
- Logic says

increase $n \Rightarrow$ use more information $\Rightarrow$ decrease error



Beyond the Sample Points

- We sampled $\sin(x)$ only on $[0, \pi]$.
- What happens if we evaluate the polynomial beyond π ?



Interpolation vs. Extrapolation

Interpolation

Estimate values **inside** the sampled interval:

$$x \in [x_0, x_n].$$

Extrapolation

Estimate values **outside** the sampled interval:

$$x < x_0 \quad \text{or} \quad x > x_n.$$

- The same polynomial can be evaluated in either region.
- A good fit inside the interval does not guarantee accuracy outside it.



Interpolation vs. Extrapolation

Interpolation

Estimate values **inside** the sampled interval:

$$x \in [x_0, x_n].$$

Extrapolation

Estimate values **outside** the sampled interval:

$$x < x_0 \quad \text{or} \quad x > x_n.$$

- The same polynomial can be evaluated in either region.
- A good fit inside the interval does not guarantee accuracy outside it.

Key Idea

The polynomial approximation $p(x) \approx f(x)$ is only valid in the data range

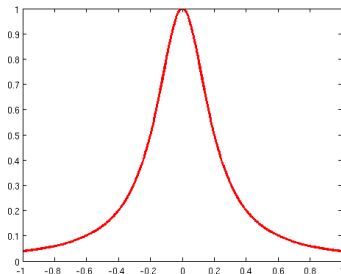
Case Study: Runge Function

Consider **Runge** function

$$f(x) = \frac{1}{1 + 25x^2}$$

Properties:

- no singularities - *good*
- bounded - *good*
- smooth - *good*
- We write this as $f \in C_b^\infty(\mathbb{R})$

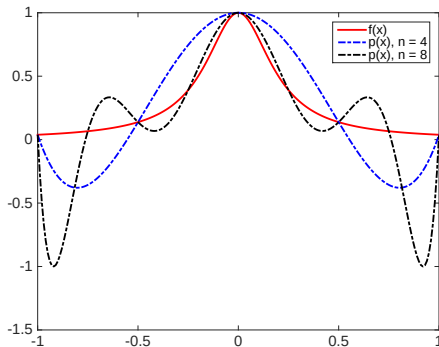


Question

Runge's function is very nice on paper. How well does equispaced interpolation capture the function?

Case Study: Runge Function

Interpolation results



Answer: Not very well

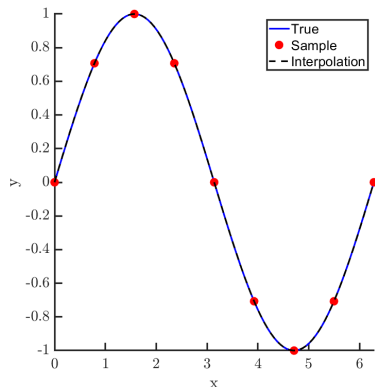
- $p_4(x)$ using 5 nodes looks bad
- $p_8(x)$ using 9 nodes *looks even worse*
- $p_n(x)$ will continue to worsen as n is increased



Case Study: Runge Function

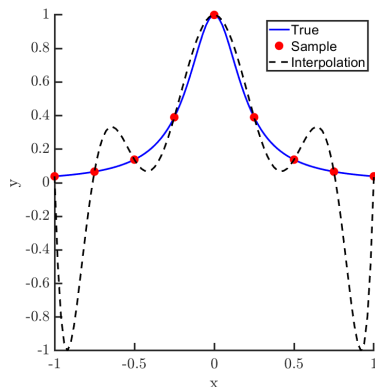
What went wrong?

$$f(x) = \sin(x)$$



No visible error

$$f(x) = \frac{1}{1 + 25x^2}$$



Very inaccurate



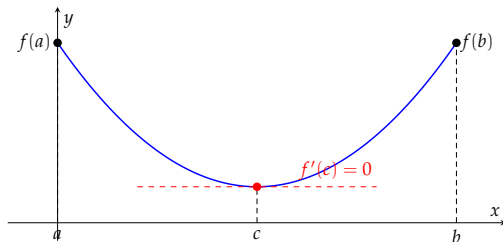
Interpolation Error



Rolle's Theorem

Theorem

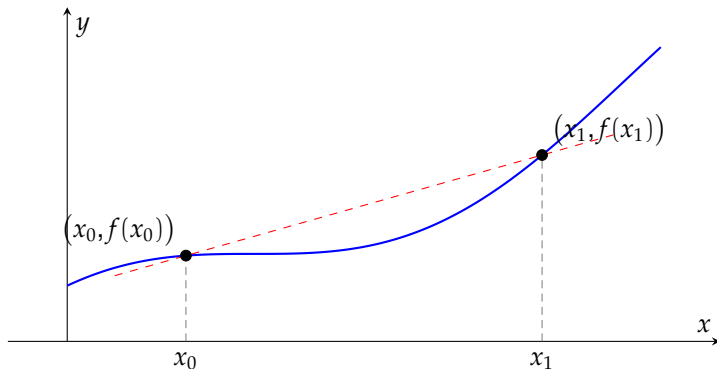
If $f(a) = f(b)$ then $\exists c \in (a, b)$ such that $f'(c) = 0$.



Key Idea

If $f(x)$ returns to the same value, then at some point, it must have turned around.

Two-Point Interpolation



Interpolation

Let $f(x)$ be the actual function. Let $p_1(x)$ be the two-point interpolation of $f(x)$.



Some analysis...

Interpolation Error

$$e(x) = f(x) - p_1(x)$$

Consider this function

$$g(t) := e(t) - \frac{(t - x_0)(t - x_1)}{(x - x_0)(x - x_1)}e(x)$$

Notice that

- $g(x_0) = 0$
- $g(x_1) = 0$
- $g(x) = 0$



Using Rolle's Theorem

Consider this function

$$g(t) := e(t) - \frac{(t - x_0)(t - x_1)}{(x - x_0)(x - x_1)}e(x)$$

Since $g(t) = 0$ at $t = x_0, x_1, x$

- That implies $g'(t)$ is zero at **two** points
- That implies $g''(t)$ is zero at **one** point; call it c
- Then

$$\begin{aligned} 0 &= g''(c) = e''(c) - 2 \frac{e(x)}{(x - x_0)(x - x_1)} \\ &= f''(c) - 2 \frac{e(x)}{(x - x_0)(x - x_1)} \\ e(x) &= \frac{(x - x_0)(x - x_1)}{2} f''(c) \end{aligned}$$

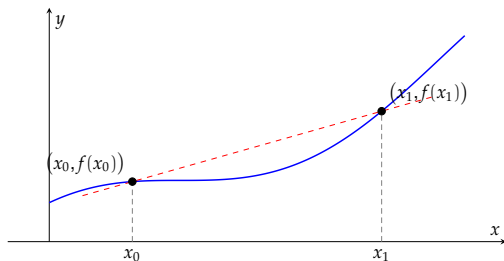


Error Formula for Two Points

Result

Suppose $f(x)$ is approximated by an interpolation $p_1(x)$ at two points x_0 and x_1 . The error is given by

$$e(x) = f(x) - p_1(x) = \frac{(x - x_0)(x - x_1)}{2} f''(c)$$



And if there were 3 points...

Question

What's the error for a three-point interpolation?

- For 3 points, the error is still

$$e(x) = f(x) - p_2(x)$$

- For this case, we let

$$g(t) := e(t) - \frac{(t - x_0)(t - x_1)(t - x_2)}{(x - x_0)(x - x_1)(x - x_2)} e(x)$$

- Then we take 3 derivatives and find that

$$e(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{6} f'''(c)$$



And if there were 3 points...

Question

What's the error for a three-point interpolation?

- For 3 points, the error is still

$$e(x) = f(x) - p_2(x)$$

- For this case, we let

$$g(t) := e(t) - \frac{(t - x_0)(t - x_1)(t - x_2)}{(x - x_0)(x - x_1)(x - x_2)} e(x)$$

- Then we take 3 derivatives and find that

$$e(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{6} f'''(c)$$

- This is generalizable to any number of points!



Analysis of Interpolation Error: General Result

Theorem: Interpolation Error I

If $p_n(x)$ is the (at most) n degree polynomial interpolating $f(x)$ at $n + 1$ distinct points and if $f^{(n+1)}$ is continuous, then

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

Proof in the book NMC 7th page 181.

- At a glance, it looks like the error decreases like $O(\frac{1}{n!})$ (that's extremely fast!)
- So why did the error Runge's function grow?
- That last term $\prod_{i=0}^n (x - x_i)$ doesn't depend on f . Just how large is it?



Interpolation error formula

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$



Interpolation error formula

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

Key Idea

The product term

$$\prod_{i=0}^n (x - x_i)$$

depends only on the location of the interpolation points.



Interpolation error formula

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

Key Idea

The product term

$$\prod_{i=0}^n (x - x_i)$$

depends only on the location of the interpolation points.

- Each factor $(x - x_i)$ is the distance from x to the node x_i
- So the error is small when these distances are small



Equally spaced nodes on a fixed interval

Question

What if assume the points x_i are equispaced:

$$x_i = a + ih, \quad i = 0, \dots, n$$

with spacing h .



Equally spaced nodes on a fixed interval

Question

What if assume the points x_i are equispaced:

$$x_i = a + ih, \quad i = 0, \dots, n$$

with spacing h .

If $x = a + th$. Then

$$x - x_i = (a + th) - (a + ih) = h(t - i).$$



Equally spaced nodes on a fixed interval

Question

What if assume the points x_i are equispaced:

$$x_i = a + ih, \quad i = 0, \dots, n$$

with spacing h .

If $x = a + th$. Then

$$x - x_i = (a + th) - (a + ih) = h(t - i).$$

Therefore,

$$\prod_{i=0}^n (x - x_i) = \prod_{i=0}^n h(t - i) = h^{n+1} \underbrace{\prod_{i=0}^n (t - i)}_{\text{grows like } n!}.$$



Theorem: Bounding Lemma

Suppose x_i are equispaced in $[a, b]$ for $i = 0, \dots, n$. Then

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

Proof in the book NMC 7th page 182.

Key Idea

The error terms has factorial in the numerator and denominator which wind up canceling

$$e(x) = \frac{f^{(n+1)}(c)}{(n+1)!} \prod_{i=0}^n (x - x_i) \leq \frac{f^{(n+1)}(c) h^{n+1}}{4} \frac{n!}{(n+1)!} = \frac{f^{(n+1)}(c) h^{n+1}}{4(n+1)}$$

Analysis of Interpolation Error: Equispaced Points

Error Bound

Assuming interpolation points x_i are equispaced:

$$e(x) \leq \frac{f^{(n+1)}(c)h^{n+1}}{4(n+1)}$$

Theorem: Interpolation Error II

Let $|f^{(n+1)}(x)| \leq M$, then with the above,

$$|e(x)| = |f(x) - p_n(x)| \leq \frac{Mh^{n+1}}{4(n+1)}$$

What is $\lim_{h \rightarrow 0} \frac{Mh^{n+1}}{4(n+1)}$?



The Takeaway

The error for interpolation with $n + 1$ nodes can be estimated with

$$|e_n(x)| = |f(x) - p_n(x)| \leq \frac{Mh^{n+1}}{4(n+1)},$$

where $|f^{(n+1)}(x)| \leq M$.

- If a function $f(x)$ can be turned into 0 with enough derivatives, then it can be perfectly represented by polynomial interpolation (zero error).
- If a function's upper bound gets larger with each derivative, then there is no guarantee on the polynomial error will decrease with n .
- Runge function is hard to interpolate because the $\max(f^{(n)})$ gets larger with respect to n



Why Sine is not an Issue

Question

Why was interpolation so effective for $y = \sin(x)$?

- Take some derivatives:

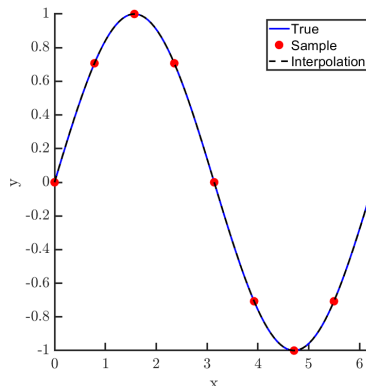
$$y' = \cos x$$

$$y'' = -\sin x$$

$$y''' = -\cos x$$

$$y^{(4)} = \sin x$$

- The derivatives are always **bounded** between 1 and -1



Runge's Function

Question

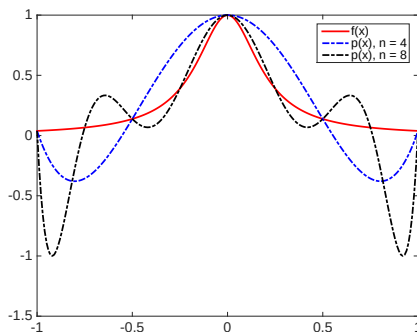
Why was interpolation so bad for $y = \frac{1}{1+x^2}$?

- The issue is **not the function values** — it is the **derivatives**
- Higher derivatives become **very large** near the endpoints
- So the constant M in the error bound

$$|f^{(n+1)}(x)| \leq M$$

is **large**

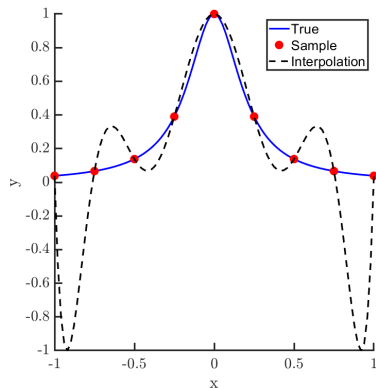
- Even if h is small, a large $M \Rightarrow$ large error



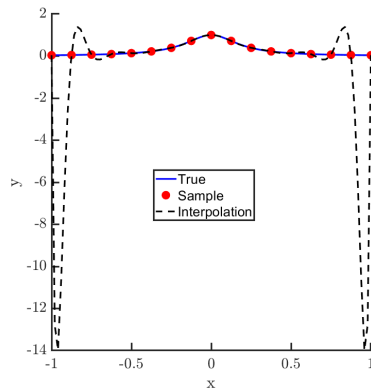
Chebyshev Nodes



Issue at the Endpoints



Interpolation with $n = 8$



Interpolation with $n = 16$

In Runge's function, the worst error is observed near the endpoints.

We have two options:

- 1 Move the nodes to the problematic areas
- 2 Restrict the maximum order of the polynomial (splines)

Option #1: Cluster the nodes towards the endpoints

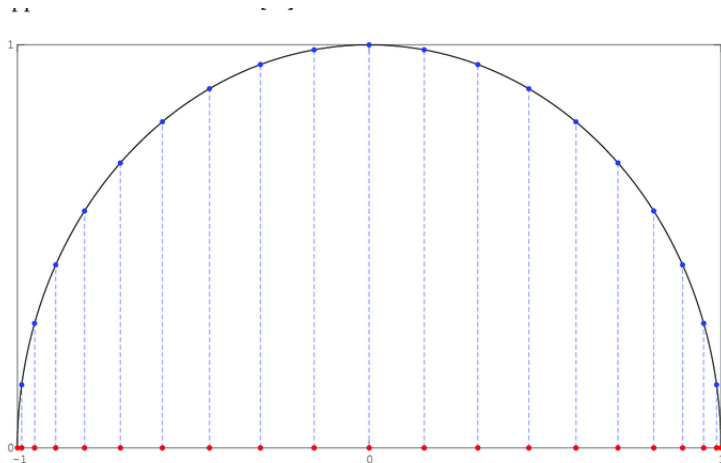
Chebyshev Nodes in $[-1, 1]$

$$x_i = \cos \left(\pi \frac{2i+1}{2n+2} \right), \quad i = 0, \dots, n$$

Option #2: Piecewise polynomials... we discuss later



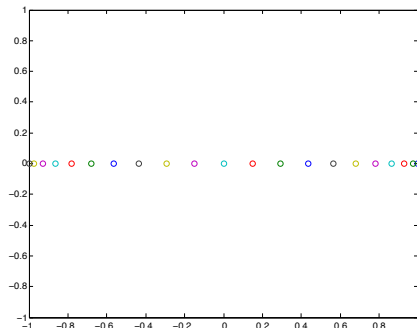
Chebyshev Nodes



- Nodes from equidistant points on a circle are projected down to the x -axis.



Chebyshev Nodes

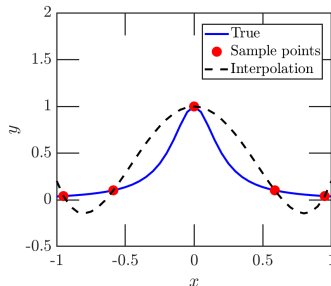


Key Idea

- With equispaced interpolation, the error is the worst at the endpoints.
- Chebyshev lets us systematically create nodes clustered at the endpoints.

Chebyshev Nodes

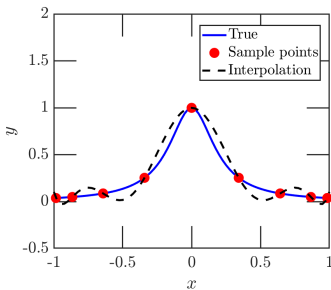
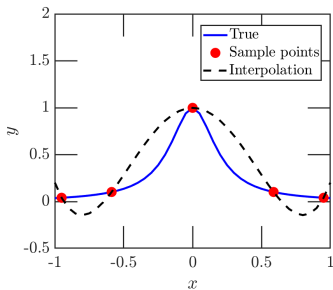
High degree polynomials using equispaced points suffer from many oscillations



Method: Chebyshev Sampling

- **Given:** Interval $[a, b]$, number of points $n + 1$
- **Output:** Set of points (x_i) on $[a, b]$ clustered at endpoints
- **Idea:** Clustering at the ends reduces the error of interpolation

Is Chebyshev the solution?



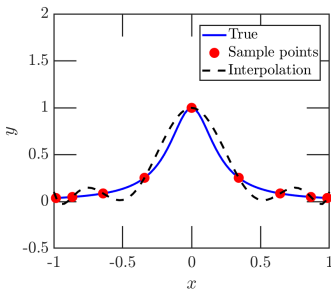
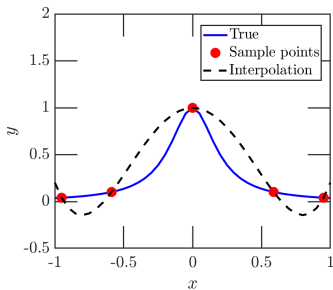
- The figures above shows interpolation with $n = 4$ and $n = 8$.

Question

Will Chebyshev get better as n increases?

Or will it encounter trouble like equispaced interpolation?

Is Chebyshev the solution?



- The figures above shows interpolation with $n = 4$ and $n = 8$.

Question

Will Chebyshev get better as n increases?
Or will it encounter trouble like equispaced interpolation?

- Homework!