Math 340 • Fall 2026 • Victor Matveev

Polynomial Interpolation Error

A big advantage of Newton's divided difference method of writing down an interpolating polynomial is the ease with which we can analyze its error. In fact, the remainder (error) of interpolation looks exactly like an extra interpolation term obtained by adding one more point:

$$f(x) = \underbrace{f[x_1] + f[x_1,x_2](x-x_1) + \ldots + f[x_1,\ldots,x_n](x-x_1)\ldots(x-x_{n-1})}_{P_{n-1}(x)} + \underbrace{f[x_1,\ldots,x_n,x](x-x_1)\ldots(x-x_n)}_{\text{Remainder / Error, } R_{n-1}(x)}$$

The **proof** of this remainder formula is remarkably simple: consider $\boldsymbol{x}$ in the above expression as another interpolation node, $\boldsymbol{x \equiv x_{n+1}}$; then the above is simply a formula for $P_n(x)$ -- an interpolant matching one extra point of $f(x)$ -- which exists for any value of $\boldsymbol{x \equiv x_{n+1}}$. Therefore, we have $\boldsymbol{f(x_{n+1}) = P_n(x_{n+1}) = P_{n-1}(x_{n+1}) + R_{n-1}(x_{n+1})}$

It can also be elegantly proven that each of the coefficients above is related to a derivative of $f(x)$, highlighting the analogy between interpolating polynomial and a "distributed" form of Taylor expansion:

$$\exists c \in \text{Interval containing all } \{x_k\}_{k=1}^n, \text{ such that } f[x_1,\ldots,x_n] = \frac{f^{(n-1)}(c)}{(n-1)!}$$

For the remainder, $\boldsymbol{R_{n-1}(x) = f[x_1, \ldots, x_n, x](x-x_1)\ldots(x-x_n)}$, this implies that

$$\exists c \in \text{Interval containing all } \{x_k\}_{k=1}^n \text{ and } x, \text{ such that } R_{n-1}(x) = \frac{f^{(n)}(c)}{n!}(x-x_1)\ldots(x-x_n)$$

Even though we don't know where the argument value $\boldsymbol{c}$ resides, the latter expression is useful when finding the upper bound on the error. Let's consider an example to show how to use the remainder formula to determine error bounds:

EXAMPLE: Construct a **linear** approximation of $\boldsymbol{f(x) = e^x}$ using $\{x_k\} = \{0, 1\}$. Examine its error over the interval [0, 1], in several ways:

1) Compute the error in the middle, at $x = \frac{1}{2}$ (since we expect the error to be largest half-way between the nodes)

2) Find the exact maximal error of $P_1(x)$ on the interval [0, 1].

3) Analyze the error bound using the remainder formula above.

First, let's find the interpolating polynomial (which is a secant line in this case of just two interpolation nodes).

x_k	$f[x_k]$	$f[x_{k-1},x_k]$
0	1	
1	e	$\frac{e-1}{1-0} = e-1$

Thus, our interpolating line is: $f(x) \sim P_1(x) = f[x_1] + f[x_1,x_2](x-x_1) = 1 + (e-1)x$

1) The remainder at $x = 1/2$ then equals: $\left|R_1\left(\frac{1}{2}\right)\right| = \left|f\left(\frac{1}{2}\right) - P_1\left(\frac{1}{2}\right)\right| = \left|e^{1/2} - \left(1 + \frac{e-1}{2}\right)\right| \approx \mathbf{0.2104}$

2) Find the exact maximum of interpolation error, $|R_1| \equiv |f(x) - P_1(x)|$, by setting its derivative to zero:

$$R_1'(x) = f'(x) - P_1'(x) = e^x - (e-1) = 0 \;\rightarrow\; e^x = e-1 \;\rightarrow\; x^* = \ln(e-1) \approx 0.54132$$

$$\max_{[0,1]} |R_1(x)| = |R_1(x^*)| = \left|e^{x^*} - 1 - (e-1)x^*\right| \approx \mathbf{0.212} \quad \textbf{Very close to the error in the middle written above!}$$

It is interesting that the actual maximal error is achieved slightly to the right of the mid-point (see figure on page 2).

3) Finally, let's compare this with the upper bound on the error obtained using the relationship between Newton's divided differences and the function derivatives:

$$|R_1(x)| = \left|\frac{f''(c)}{2!}(x-x_1)(x-x_2)\right|$$

$$\leq \max_{[0,1]}\left|\frac{f''(c)}{2!}\right| \cdot \max_{[0,1]}|(x-x_1)(x-x_2)|$$

$$= \max_{[0,1]}\left|\frac{e^c}{2}\right| \cdot \max_{[0,1]}|x(x-1)| = \frac{e}{2}\left|\frac{1}{2}\left(\frac{1}{2}-1\right)\right| = \frac{e}{8} \approx \mathbf{0.34}$$

Here we used the fact that the parabola $x^2 - x$ achieves its highest absolute value exactly in the middle, at $x = \frac{1}{2}$.

Note that the first term (the derivative of exponential) is the largest at the right end of the interval, where the error is actually reduced by the proximity to the interpolation point.

As expected, the error bound is an over-estimate of the true maximal error obtained above: $0.34 > 0.212$.

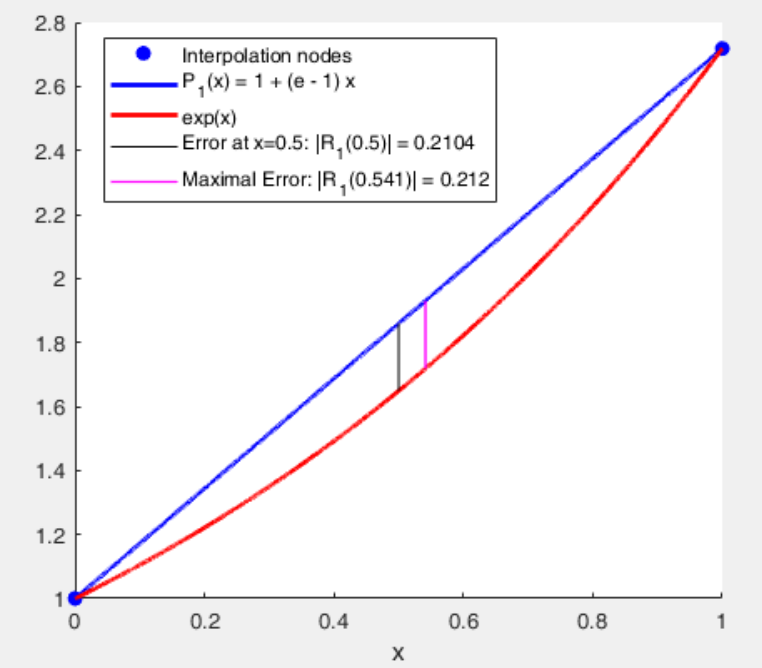
Finally, let's compare the maximal true error with the 1st degree **Taylor expansion** at $x = 0$:

$$f(x) = \exp(x) \approx T_1(x) = 1 + x$$

The maximal error of this approximation is of course achieved at $x = 1$: $Error[T_1] = |f(1) - T_1(1)| = e - 2 \approx 0.72.$

This is significantly larger than the error of the polynomial interpolant, but we could have done better by expanding around the midpoint instead: $T_1(x) = e^{\frac{1}{2}} + e^{\frac{1}{2}}\left(x - \frac{1}{2}\right) = e^{\frac{1}{2}}\left(x + \frac{1}{2}\right)$. In that case, the error would still be higher than for P_1: $Error[T_1] = |f(1) - T_1(1)| = \left|e - \frac{3}{2}e^{\frac{1}{2}}\right| \approx 0.2452 > 0.212$

Figure 1: Linear interpolation of e^x using nodes {0, 1}. The largest error is slightly to the right of the mid-point, $x = \frac{1}{2}$.



Conclusion 1: Polynomial interpolation may provide a more accurate approximation than the Taylor expansion, if the goal is to approximate the function over an entire interval.

Conclusion 2: The exact error formula is of course preferable, but the error bound approach gives a simple way to find the "worst-case scenario" error over the **entire** interval, with minimal extra computation.

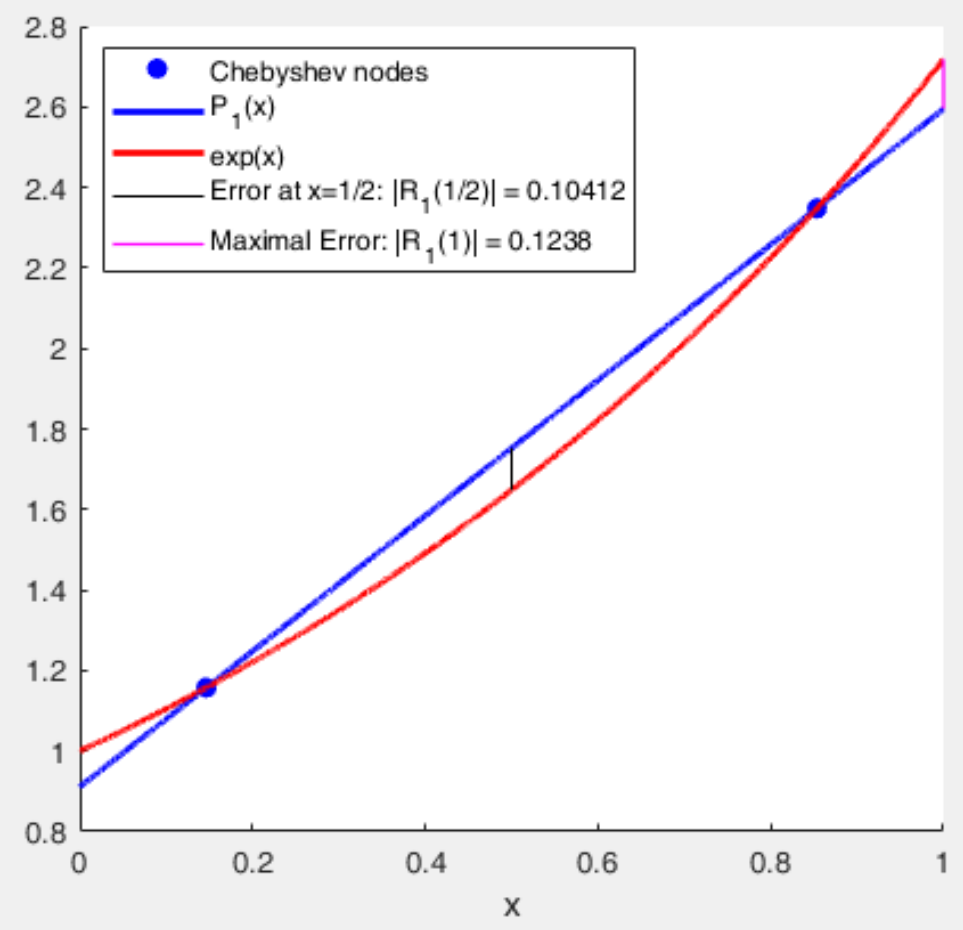
On the next slide, we will see that our choice of the two interpolation nodes in this figure is a bit silly, given our goals.

Near-Optimal Interpolation Node Choice: Chebyshev Points

From the figure above, it is clear that endpoints (and, more generally, equidistant points) are far from an optimal choice.

In fact, we will prove in class that the basis function part of the error bound, $\max_{[0,1]}|\boldsymbol{\psi}(\boldsymbol{x})| = \max_{[0,1]}|(\boldsymbol{x}-\boldsymbol{x_1})(\boldsymbol{x}-\boldsymbol{x_2})|$, is minimized over the interval [0, 1] when $\boldsymbol{x_{1,2}} = \frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{1} \pm \mathbf{cos}\frac{\boldsymbol{\pi}}{\mathbf{4}}\right) = \frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{1} \pm \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\right)$.

We get a much better fit with this choice, shown in **Figure 2** below:



Parenthetically, we note that the maximal error in this case is achieved at the endpoint $x = 1$, not in the middle of the domain. In principle, we could further reduce the maximal error by moving the nodes slightly to the right, until the error is the same at both endpoints and at the interior point where it attains its local maximum.