Lecture of:
26 Feb 2013

Newton interpolation also facilitates easy evaluation:

$$
\begin{aligned}
\text{e.g. } p(x) &= c_0 \\
&\quad + c_1(x-x_0) \\
&\quad + c_2(x-x_0)(x-x_1) \\
&\quad + c_3(x-x_0)(x-x_1)(x-x_2) \\
&\quad + c_4(x-x_0)(x-x_1)(x-x_2)(x-x_3) \\
&= \underbrace{c_0 + (x-x_0)\underbrace{\left[c_1 + (x-x_1)\underbrace{\left[c_2 + (x-x_2)\underbrace{\left[c_3 + (x-x_3)\underbrace{c_4}_{Q_4(x)}\right]}_{Q_3(x)}\right]}_{Q_2(x)}\right]}_{Q_1(x)}}_{Q_0(x)}
\end{aligned}
$$

Recursively:

$$
\begin{aligned}
Q_n(x) &= c_n \\
Q_{n-1}(x) &= c_{n-1} + (x - x_{n-1})Q_n(x)
\end{aligned} \tag{7}
$$

The value of $P(x) = Q_0(x)$ can be evaluated (in linear time) by iterating this recurrence n times. We also have:

$$
\begin{aligned}
Q_{n-1}(x) &= c_{n-1} + (x - x_{n-1})Q_n(x) \\
\Rightarrow Q'_{n-1}(x) &= Q_n(x) + (x - x_{n-1})Q'_n(x)
\end{aligned} \tag{8}
$$

Thus, if we iterate the recurrence in equation (8) along with the recurrence of equation (7), we can ultimately compute the value of the derivative $P'(x) = Q'_0(x)$ in linear time as well.

Accuracy and interpolation error

We saw three methods for polynomial interpolation (Vandermonde, Lagrange, Newton). It is important to understand that all 3 methods compute (outside of any discrepancies due to machine precision) the same exact interpolant $P(x)$ just following different paths which may be better or worse from a computational perspective.

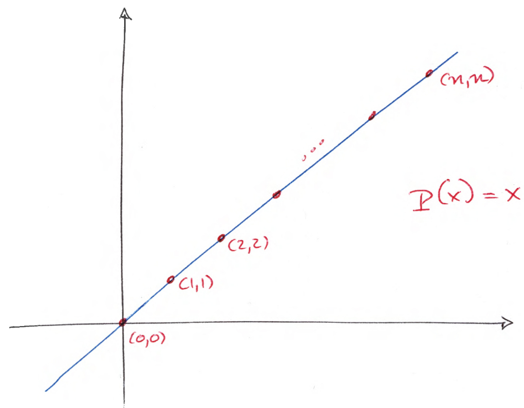
The question, however, remains:

- How accurate is this interpolation? Or, in other words:
- How close is $P(x)$ to the *real* function $f(x)$ whose plot the data points (x_i, y_i) were collected from?

Consider interpolating through the following points

$$(x_0, y_0) = (0, 0), (x_1, y_1) = (1, 1), \ldots, (x_n, y_n) = (n, n)$$

which are sampled from the plot of the straight line $f(x) = x$.



Since $f(x) = x$ is a 1-st order polynomial, any of our interpolation methods would reconstruct it exactly. Using Lagrange polynomials, $P(x)(= x)$ is written as:

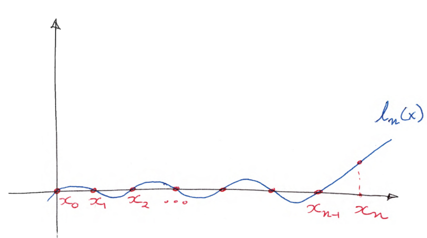
$$P(x) = \sum_{i=0}^{n} y_i l_i(x)$$

Let us "shift" y_n by a small amount δ. The new value is $y_n^* = y_n + \delta$. The updated interpolant $P^*(x)$ then becomes:

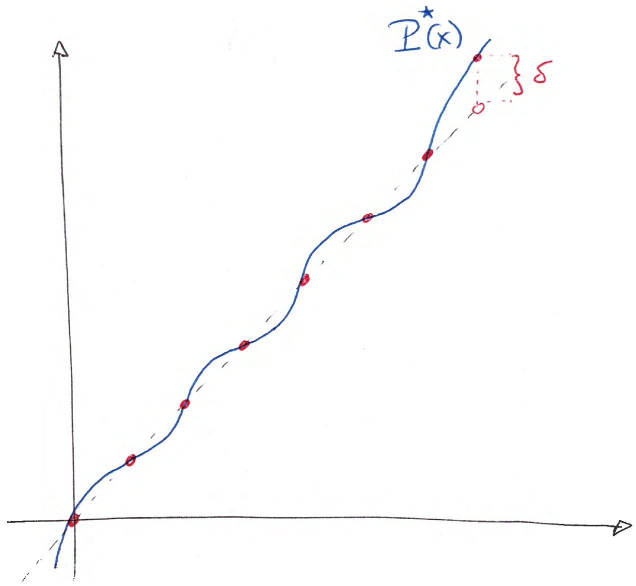
$$P^*(x) = \sum_{i=0}^{n-1} y_i l_i(x) + y_n^* l_n(x)$$

Thus, this shift in the value of y_n by δ would change our computed interpolant by $P^*(x) - P(x) = \delta \cdot l_n(x)$.

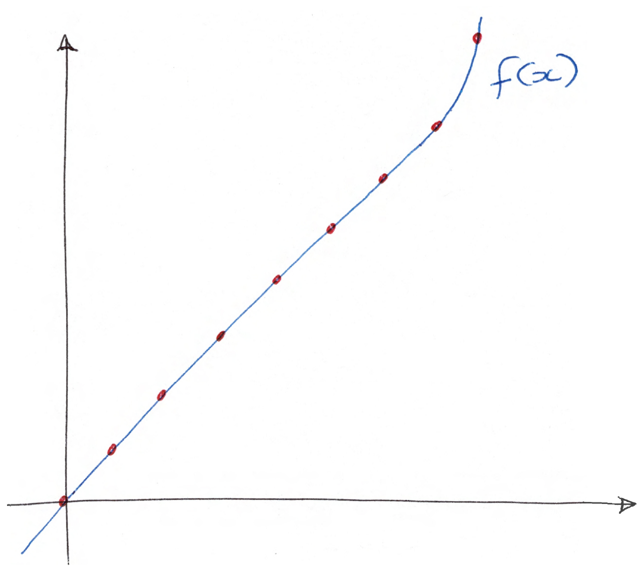
Note that $l_n(x)$ is a function that *oscillates* through zero several times:



Thus, P^* looks like:



What we observe is that a *local* change in y-values caused a *global* (and drastic) change in $P(x)$. Perhaps the *real* function f would have exhibited a more graceful and localized change, e.g.:



We will use the following theorem to compare the *real* function f being sampled and the reconstructed interpolant $P(x)$.

Theorem Let:

- $x_1 < x_2 < \cdots < x_{n-1} < x_n$
- $y_k = f(x_k)$ $k = 1, 2, \ldots, n$ where f is a function which is n-times differentiable with continuous derivatives.
- $P(x)$ is a polynomial that interpolates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Then for any $x \in (x_1, x_n)$ there exists a $\theta = \theta(x) \in (x_1, x_n)$ such that

$$f(x) - P(x) = \frac{f^{(n)}(\theta)}{n!}(x - x_1)(x - x_2) \ldots (x - x_n)$$

This theorem may be difficult to apply directly, since:

- θ is not known.
- θ changes with x.
- The nth derivative $f^{(n)}(x)$ may not be fully known.

However, we can use it to derive a conservative bound:

Theorem: If $M = \max\limits_{x \in [x_1, x_n]} |f^{(n)}(x)|$ and $h = \max\limits_{1 \leq i \leq n-1} |x_{i+1} - x_i|$, then

$$|f(x) - P(x)| \leq \frac{Mh^n}{4n}$$

for all $x \in [x_1, x_n]$.

How good is this, especially when we keep adding more and more data points (e.g. $n \to \infty$ and $h \to 0$)? This really depends on the higher order derivatives of $f(x)$. For example,

$$f(x) = \sin(x),\ x \in [0, 2\pi]$$

All derivatives of f are $\pm \sin(x)$ or $\pm \cos(x)$. Thus, $|f^{(k)}(x)| \leq 1$ for any k. In this case, $M = 1$, and as we add more (and denser) data points, we have

$$|f(x) - P(x)| \leq \frac{Mh^n}{4n} \xrightarrow[n \to \infty]{h \to 0} 0$$

For some functions, however, the values of $|f^{(k)}(x)|$ grow vastly as $k \to 0$ (i.e. when we introduce additional points), e.g.:

$$f(x) = \frac{1}{x} \Rightarrow |f^{(n)}(x)| = n!\frac{1}{x^{n+1}},\ x \in (0.5, 1)$$

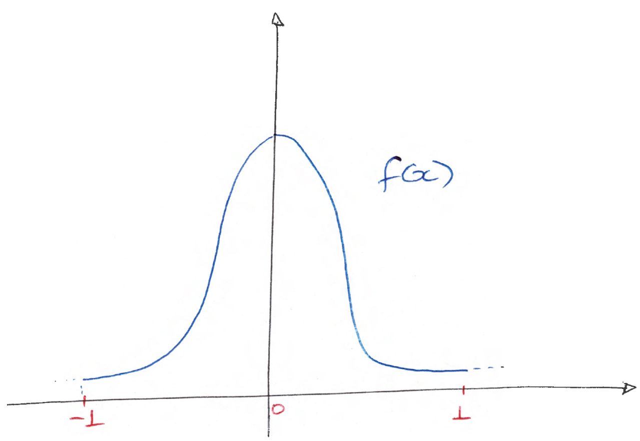
$$M_n = \max_{x \in (0.5,1)} |f^{(n)}(x)| = n! \cdot 2^n$$

in this case, as $n \to \infty$:

$$\frac{M_n h^n}{f_n} = \frac{n! 2^n h^n}{4n} \to \infty$$

Another commonly cited counter-example is Runge's function:

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



Approximating this with a 5-degree polynomial yields: