Polynomial Interpolation

Math 426

University of Alaska Fairbanks

October ~~28~~ 30, 2020 (boo!)

$x(i) = 0$ $\qquad b(i)$

for $j = 1 : i-1$

$\quad x(i) = x(i) - L(i,j) \cdot x(j)$

end

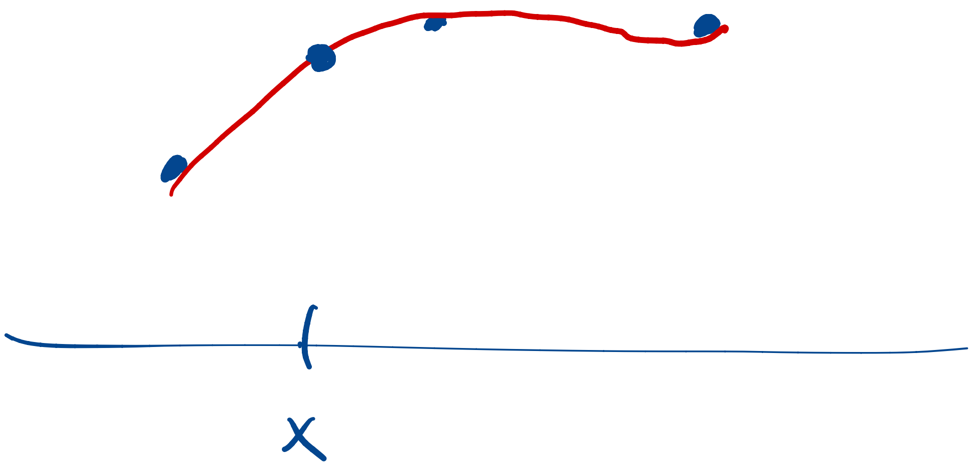
$x(i) = (x(i) + b(i)) \,/\, L(i,i)$

$$((10^{-20} + 1) + (-1)) = \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix}$$

$$(10^{-20} + (1 + (-1))) = 10^{-20}$$

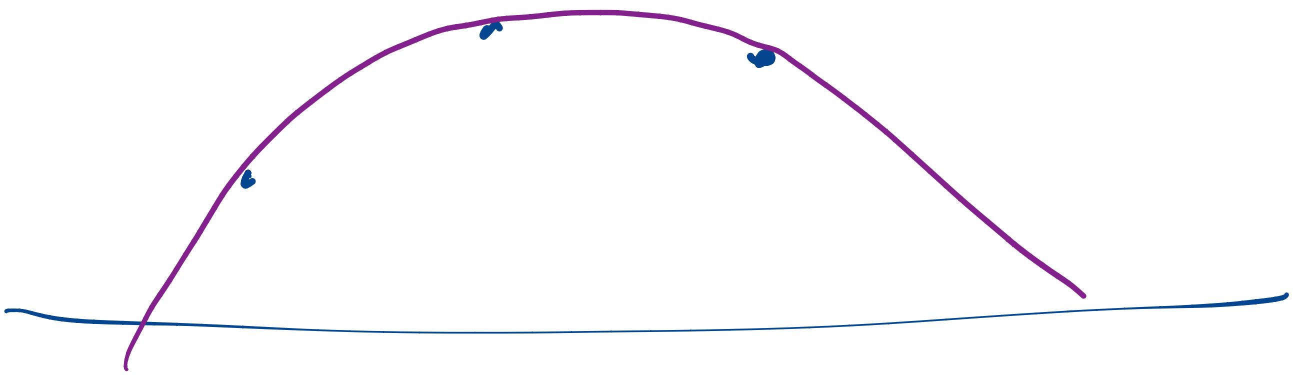
Interpolation

We have data $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ that are samples of a function $y = f(x)$ and wish to estimate $f(x)$ for $x \neq x_0, \ldots, x_n$.



Polynomial Interpolation

We have data $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ that are samples of a function $y = f(x)$ and wish to estimate $f(x)$ for $x \neq x_0, \ldots, x_n$.



Polynomial Interpolation

We have data $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ that are samples of a function $y = f(x)$ and wish to estimate $f(x)$ for $x \neq x_0, \ldots, x_n$.

One approach: find a polynomial $p(x)$ with $p(x_k) = y_k$, $k = 0, \ldots, n$.

Polynomial Interpolation

We have data $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ that are samples of a function $y = f(x)$ and wish to estimate $f(x)$ for $x \neq x_0, \ldots, x_n$.

One approach: find a polynomial $p(x)$ with $p(x_k) = y_k$, $k = 0, \ldots, n$.

What order?

$$p(x) = c_m x^m + c_{m-1} x^{m-1} + \cdots + c_1 x + c_0$$

Polynomial Interpolation

We have data $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ that are samples of a function $y = f(x)$ and wish to estimate $f(x)$ for $x \neq x_0, \dots, x_n$.

One approach: find a polynomial $p(x)$ with $p(x_k) = y_k$, $k = 0, \dots, n$.

What order?

$$p(x) = c_m x^m + c_{m-1} x^{m-1} + \cdots + c_1 x + c_0$$

$m + 1$ coefficeints and $n + 1$ conditions $p(x_k) = y_k$, so n^{th} order is appropriate.

Polynomial Interpolation

$$p(x) = c_n x^n + \cdots + c_1 x + c_0$$

Polynomial Interpolation

$$p(x) = c_n x^n + \cdots + c_1 x + c_0$$

Equations to solve:

$$\begin{aligned}
c_0 + c_1 x_0 + \cdots + c_n x_0^n &= y_0 \\
c_0 + c_1 x_1 + \cdots + c_n x_1^n &= y_1 \\
\vdots \qquad\qquad &= \vdots \\
c_0 + c_1 x_n + \cdots + c_n x_n^n &= y_n
\end{aligned}$$

Polynomial Interpolation

$$p(x) = c_n x^n + \cdots + c_1 x + c_0$$

(x_k, y_k)

$0 \le k \le n$

$p(x_k) = y_k$

Equations to solve:

$$\begin{aligned} c_0 + c_1 x_0 + \cdots + c_n x_0^n &= y_0 \\ c_0 + c_1 x_1 + \cdots + c_n x_1^n &= y_1 \\ \vdots \quad &= \vdots \\ c_0 + c_1 x_n + \cdots + c_n x_n^n &= y_n \end{aligned}$$

$n^2 + O(n)$

Matrix form

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$\frac{2}{3} n^3 + O(n^2)$

Polynomial Interpolation

$$p(x) = c_n x^n + \cdots + c_1 x + c_0$$

Equations to solve:

$$\begin{aligned} c_0 + c_1 x_0 + \cdots + c_n x_0^n &= y_0 \\ c_0 + c_1 x_1 + \cdots + c_n x_1^n &= y_1 \\ \vdots \quad &= \vdots \\ c_0 + c_1 x_n + \cdots + c_n x_n^n &= y_n \end{aligned}$$

Matrix form

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

The matrix is a **Vandermonde** matrix.

Matlab Convention

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

vs.

$$\begin{pmatrix} x_0^n & \cdots & x_0^2 & x_0 & 1 \\ x_1^n & \cdots & x_1^2 & x_1 & 1 \\ \vdots & \ddots & & \vdots & \\ x_n^n & \cdots & x_n^2 & x_n & 1 \end{pmatrix} \begin{pmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Matlab Convention

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

vs.

$$\begin{pmatrix} x_0^n & \cdots & x_0^2 & x_0 & 1 \\ x_1^n & \cdots & x_1^2 & x_1 & 1 \\ \vdots & \ddots & & \vdots & \\ x_n^n & \cdots & x_n^2 & x_n & 1 \end{pmatrix} \begin{pmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$5x^2 - 7x + 2$$

```
c = [5,-7,2]; x=[1,2,3]; polyval(c,x)
```

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix:

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Solution of system:

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Solution of system:$2/3n^3 + O(n^2)$

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Solution of system:$2/3n^3 + O(n^2)$

Evaluation:

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$$

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Solution of system:$2/3n^3 + O(n^2)$

Evaluation:

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$$

$n-1$ additions, $0+1+2+\cdots+n = n(n+1)/2$ multiplications

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Solution of system: $2/3n^3 + O(n^2)$

Evaluation:

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$$

$n + 1$ additions, $0 + 1 + 2 + \cdots + n = n(n+1)/2$ multiplications

$n^2/2 + O(n)$ operations.

Computational Effort

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Construction of matrix: $n(n-1) = n^2 + O(n)$

Solution of system:$2/3n^3 + O(n^2)$

Evaluation:

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$$

$n-1$ additions, $0+1+2+\cdots+n = n(n+1)/2$ multiplications

$n^2/2 + O(n)$ operations.

Aim: Do better!

Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_0(x) = \frac{(x - x_1)\cdots(x - x_n)}{(x_0 - x_1)\cdots(x_0 - x_n)}$$

$\varphi_0(x_1) = 0$; $\varphi_0(x_k) = 0 \quad k \neq 0$

$$\varphi_0(x_0) = \frac{(x_0 - x_1)\cdots(x_0 - x_n)}{(x_0 - x_1)\cdots(x_0 - x_n)}$$

$= ($

Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_0(x) = \frac{(x - x_1)\cdots(x - x_n)}{(x_0 - x_1)\cdots(x_0 - x_n)}$$

$$\phi_1(x) = \frac{(x - x_0)(x - x_2)\cdots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_0 - x_n)}$$

$$\frac{(x - x_1)}{(x_1 - x_1)}$$

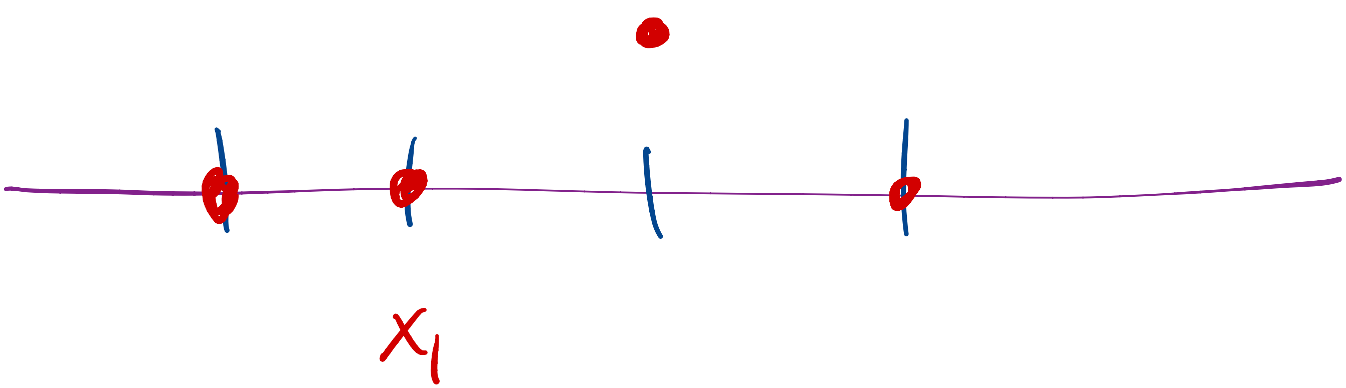
Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_0(x) = \frac{(x - x_1)\cdots(x - x_n)}{(x_0 - x_1)\cdots(x_0 - x_n)}$$

$$\phi_1(x) = \frac{(x - x_0)(x - x_2)\cdots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_0 - x_n)}$$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$



Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_0(x) = \frac{(x - x_1)\cdots(x - x_n)}{(x_0 - x_1)\cdots(x_0 - x_n)}$$

$$\phi_1(x) = \frac{(x - x_0)(x - x_2)\cdots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_0 - x_n)}$$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$$\phi_k(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

$$p(x_1) = y_0 \phi_0(x_1) + y_1 \phi_1(x_1) + \cdots + y_n \phi_n(x_n)$$

Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

$$p(x_1) = y_0\phi_0(x_1) + y_1\phi_1(x_1) + \cdots + y_n\phi_n(x_n)$$

No work!

Lagrange Form

Data points: $(x_0, y_0), \ldots, (x_n, y_n)$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

$$p(x_1) = y_0\phi_0(x_1) + y_1\phi_1(x_1) + \cdots + y_n\phi_n(x_n)$$

No work!There has to be a catch.

Polynomial Evaluation Revisted

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}.$$

$n = 3$

$(n+1)(4n + O(1))$

$4n^2 + O(n)$

$n-1$ mults

$2n$ subtractions

n divisions

$4n + O(1)$ operations.

Polynomial Evaluation Revisted

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$2n$ subtractions, $2(n-1)$ multiplications, 1 division: $4n + O(1)$ operations.

Polynomial Evaluation Revisted

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

$$\phi_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$2n$ subtractions, $2(n-1)$ multiplications, 1 division: $4n + O(1)$ operations. There are $n+1$ of these! Total cost: $4n^2 + O(n)$.

Horner's Rule

$$p(x) = c_3x^3 + c_2x^2 + c_1x + c_0$$

Horner's Rule

$$p(x) = c_3x^3 + c_2x^2 + c_1x + c_0$$

$$((c_3x + c_2)x + c_1)x + c_0$$

$$c_3x^2 + c_2x + c_1$$

$$(c_3x + c_2)x + c_1$$

Horner's Rule

$$p(x) = c_3 x^3 + c_2 x^2 + c_1 x + c_0$$

$$((c_3 x + c_2)x + c_1)x + c_0$$

Evaluating a polynomial expressed like this can be done in $2n$ operations.

polyval(c, x)

Horner's Rule

$$p(x) = c_3x^3 + c_2x^2 + c_1x + c_0$$

$$((c_3x + c_2)x + c_1)x + c_0$$

Evaluating a polynomial expressed like this can be done in $2n$ operations.

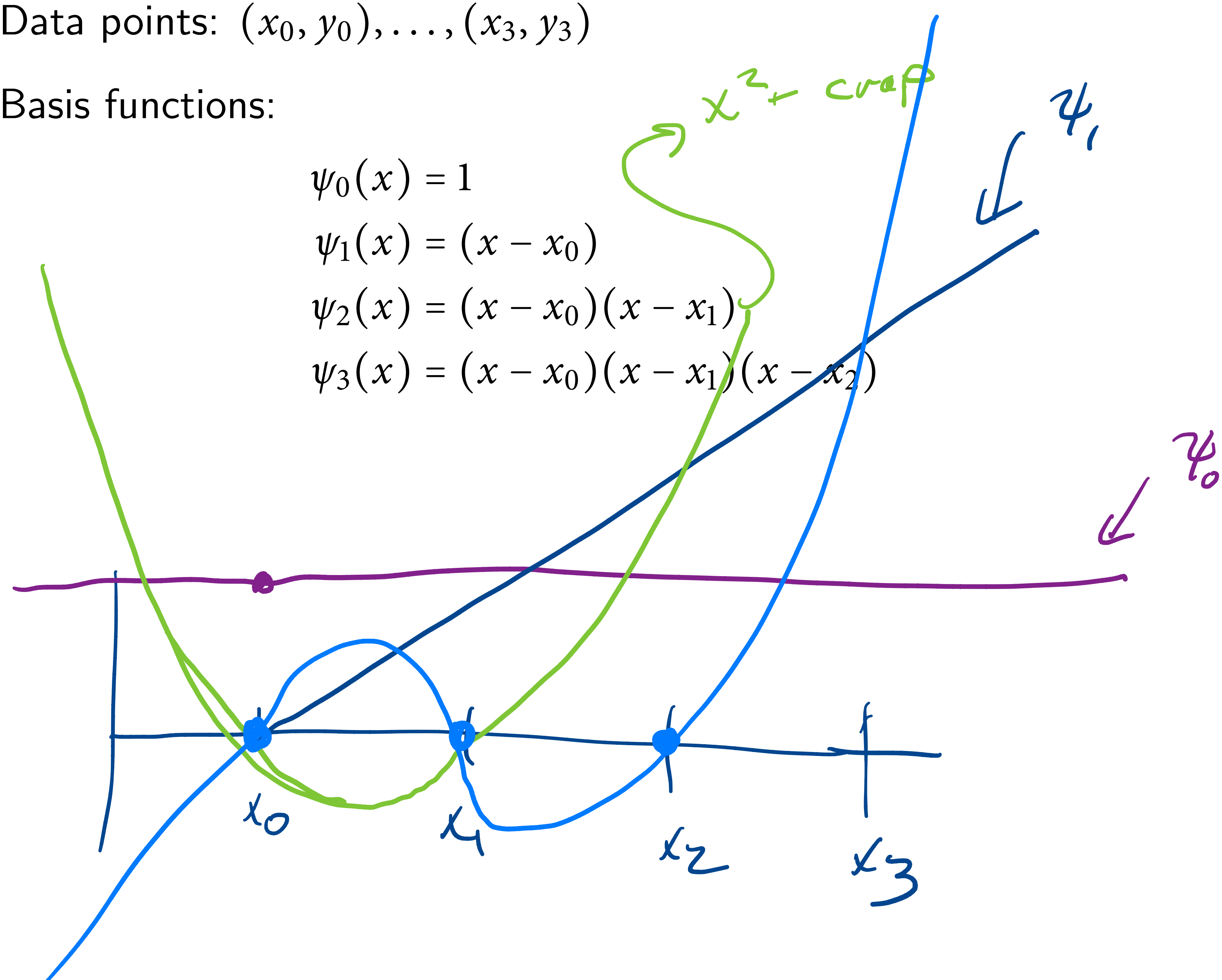
Even if we can evaluate each ϕ_k in $2n$ operations, we still would require $2n^2 + O(n)$ to evaluate p.

Newton Form

Data points: $(x_0, y_0), \ldots, (x_3, y_3)$

Basis functions:

$$\psi_0(x) = 1$$
$$\psi_1(x) = (x - x_0)$$
$$\psi_2(x) = (x - x_0)(x - x_1)$$
$$\psi_3(x) = (x - x_0)(x - x_1)(x - x_2)$$



Newton Form

Data points: $(x_0, y_0), \ldots, (x_3, y_3)$

Basis functions:

$$
\begin{aligned}
\psi_0(x) &= 1 \\
\psi_1(x) &= (x - x_0) \\
\psi_2(x) &= (x - x_0)(x - x_1) \\
\psi_3(x) &= (x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

Key point:

$$
\begin{aligned}
\psi_1(x_0) &= 0 \\
\psi_2(x_0) = \psi_2(x_1) &= 0 \\
\psi_3(x_0) = \psi_3(x_1) = \psi_3(x_2) &= 0.
\end{aligned}
$$

$$\psi_k(x_k) \neq 0$$

Newton Form

Data points: $(x_0, y_0), \ldots, (x_3, y_3)$

Basis functions: $\psi_k(x_j) = 0$ if $j < k$.

Newton Form

Data points: $(x_0, y_0), \ldots, (x_3, y_3)$

$p(x_k) = y_k$

Basis functions: $\psi_k(x_j) = 0$ if $j < k$.

$$p(x) = c_0\psi_0(x) + c_1\psi_1(x) + c_2\psi_2(x) + c_3\psi_3(x)$$

$$p(x_1) = c_0\psi_0(x_1) + c_1\psi_1(x_1) + c_2\psi_2(x_1) + c_3\psi_3(x_1) = y_1$$

$$\begin{pmatrix} \psi_0(x_0) & \psi_1(x_0) & \psi_2(x_0) & \psi_3(x_0) \\ \psi_0(x_1) & \psi_1(x_1) & \psi_2(x_1) & \psi_3(x_1) \\ \psi_0(x_2) & \psi_1(x_2) & \psi_2(x_2) & \psi_3(x_2) \\ \psi_0(x_3) & \psi_1(x_3) & \psi_2(x_3) & \psi_3(x_3) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$O(n^2)$ $n^2 + O(1)$ $O(n^3)$

Newton Form

Data points: $(x_0, y_0), \ldots, (x_3, y_3)$

Basis functions: $\psi_k(x_j) = 0$ if $j < k$.

$$p(x) = c_0\psi_0(x) + c_1\psi_1(x) + c_2\psi_2(x) + c_3\psi_3(x)$$

$$\begin{pmatrix} \psi_0(x_0) & \psi_1(x_0) & \psi_2(x_0) & \psi_3(x_0) \\ \psi_0(x_1) & \psi_1(x_1) & \psi_2(x_1) & \psi_3(x_1) \\ \psi_0(x_2) & \psi_1(x_2) & \psi_2(x_2) & \psi_3(x_2) \\ \psi_0(x_3) & \psi_1(x_3) & \psi_2(x_3) & \psi_3(x_3) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\begin{pmatrix} \psi_0(x_0) & 0 & 0 & 0 \\ \psi_0(x_1) & \psi_1(x_1) & 0 & 0 \\ \psi_0(x_2) & \psi_1(x_2) & \psi_2(x_2) & 0 \\ \psi_0(x_3) & \psi_1(x_3) & \psi_2(x_3) & \psi_3(x_3) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Newton Form Cost

Basis functions:

$$
\begin{aligned}
\psi_0(x) &= 1\\
\psi_1(x) &= (x - x_0)\\
\psi_2(x) &= (x - x_0)(x - x_1)\\
\psi_3(x) &= (x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

$$
\begin{pmatrix}
\psi_0(x_0) & 0 & 0 & 0\\
\psi_0(x_1) & \psi_1(x_1) & 0 & 0\\
\psi_0(x_2) & \psi_1(x_2) & \psi_2(x_2) & 0\\
\psi_0(x_3) & \psi_1(x_3) & \psi_2(x_3) & \psi_3(x_3)
\end{pmatrix}
\begin{pmatrix} c_0\\ c_1\\ c_2\\ c_3 \end{pmatrix}
=
\begin{pmatrix} y_0\\ y_1\\ y_2\\ y_3 \end{pmatrix}
$$

Newton Form Cost

Basis functions:

$$\begin{aligned}\psi_0(x) &= 1\\ \psi_1(x) &= (x - x_0)\\ \psi_2(x) &= (x - x_0)(x - x_1)\\ \psi_3(x) &= (x - x_0)(x - x_1)(x - x_2)\end{aligned}$$

$$\begin{pmatrix} \psi_0(x_0) & 0 & 0 & 0 \\ \psi_0(x_1) & \psi_1(x_1) & 0 & 0 \\ \psi_0(x_2) & \psi_1(x_2) & \psi_2(x_2) & 0 \\ \psi_0(x_3) & \psi_1(x_3) & \psi_2(x_3) & \psi_3(x_3) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Solution:

Newton Form Cost

Basis functions:

$$
\begin{aligned}
\psi_0(x) &= 1 \\
\psi_1(x) &= (x - x_0) \\
\psi_2(x) &= (x - x_0)(x - x_1) \\
\psi_3(x) &= (x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

Construction $O(n^2)$

$$
\begin{pmatrix}
\psi_0(x_0) & 0 & 0 & 0 \\
\psi_0(x_1) & \psi_1(x_1) & 0 & 0 \\
\psi_0(x_2) & \psi_1(x_2) & \psi_2(x_2) & 0 \\
\psi_0(x_3) & \psi_1(x_3) & \psi_2(x_3) & \psi_3(x_3)
\end{pmatrix}
\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}
=
\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}
$$

Solution: $n^2 + O(n)$.

Newton Form Evaluation Cost

Basis functions:

$$\psi_0(x) = 1$$
$$\psi_1(x) = (x - x_0)$$
$$\psi_2(x) = (x - x_0)(x - x_1)$$
$$\psi_3(x) = (x - x_0)(x - x_1)(x - x_2)$$

$$p(x) = c_0\psi_0(x) + c_1\psi_1(x) + c_2\psi_2(x) + c_3\psi_3(x)$$

Newton Form Evaluation Cost

Basis functions:

$$
\begin{aligned}
\psi_0(x) &= 1 \\
\psi_1(x) &= (x - x_0) \\
\psi_2(x) &= (x - x_0)(x - x_1) \\
\psi_3(x) &= (x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

$$p(x) = c_0\psi_0(x) + c_1\psi_1(x) + c_2\psi_2(x) + c_3\psi_3(x)$$

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$$

$$p(x) = c_1 + (x - x_0)\big(c_1 + c_2(x - x_1) + c_3(x - x_1)(x - x_2)\big)$$

$$= c_1 + (x - x_0)\big(c_1 + (x - x_1)(c_2 + c_3(x - x_3))\big)$$

Newton Form Evaluation Cost

Basis functions:

$$\psi_0(x) = 1$$
$$\psi_1(x) = (x - x_0)$$
$$\psi_2(x) = (x - x_0)(x - x_1)$$
$$\psi_3(x) = (x - x_0)(x - x_1)(x - x_2)$$

$$p(x) = c_0\psi_0(x) + c_1\psi_1(x) + c_2\psi_2(x) + c_3\psi_3(x)$$

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$$

$$p(x) = c_0 + (x - x_0)(c_1 + (x - x_1)(c_2 + (x - x_2)c_3))$$

Newton Form Evaluation Cost

Basis functions:

$$\begin{aligned}\psi_0(x) &= 1\\ \psi_1(x) &= (x - x_0)\\ \psi_2(x) &= (x - x_0)(x - x_1)\\ \psi_3(x) &= (x - x_0)(x - x_1)(x - x_2)\end{aligned}$$

$$p(x) = c_0\psi_0(x) + c_1\psi_1(x) + c_2\psi_2(x) + c_3\psi_3(x)$$

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$$

$$p(x) = c_0 + (x - x_0)(c_1 + (x - x_1)(c_2 + (x - x_2)c_3))$$

Evaluation: $3n$ operations

Summary of Costs

	Vandermonde	Lagrange	Newton
Construction	$n^2 + O(n)$	0	$n^2 + O(n)$
Solution	$(2/3)n^3 + O(n^2)$	0	$n^2 + O(n)$
Evaluation	$2n$	$2n^2 + O(n)$	$3n$

$2n^2 + O(n)$ $2n^2 + O(n)$