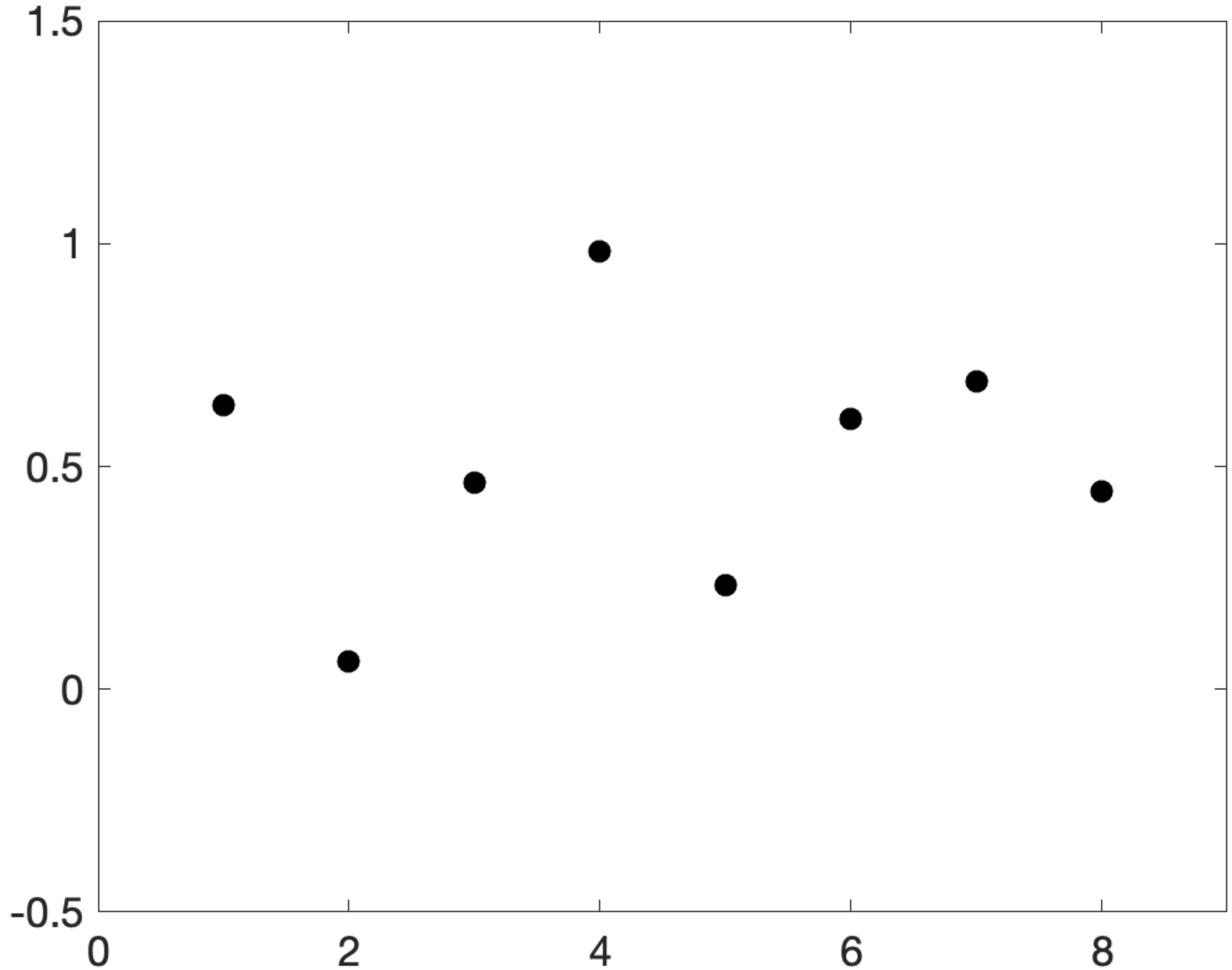
Polynomial Interpolation

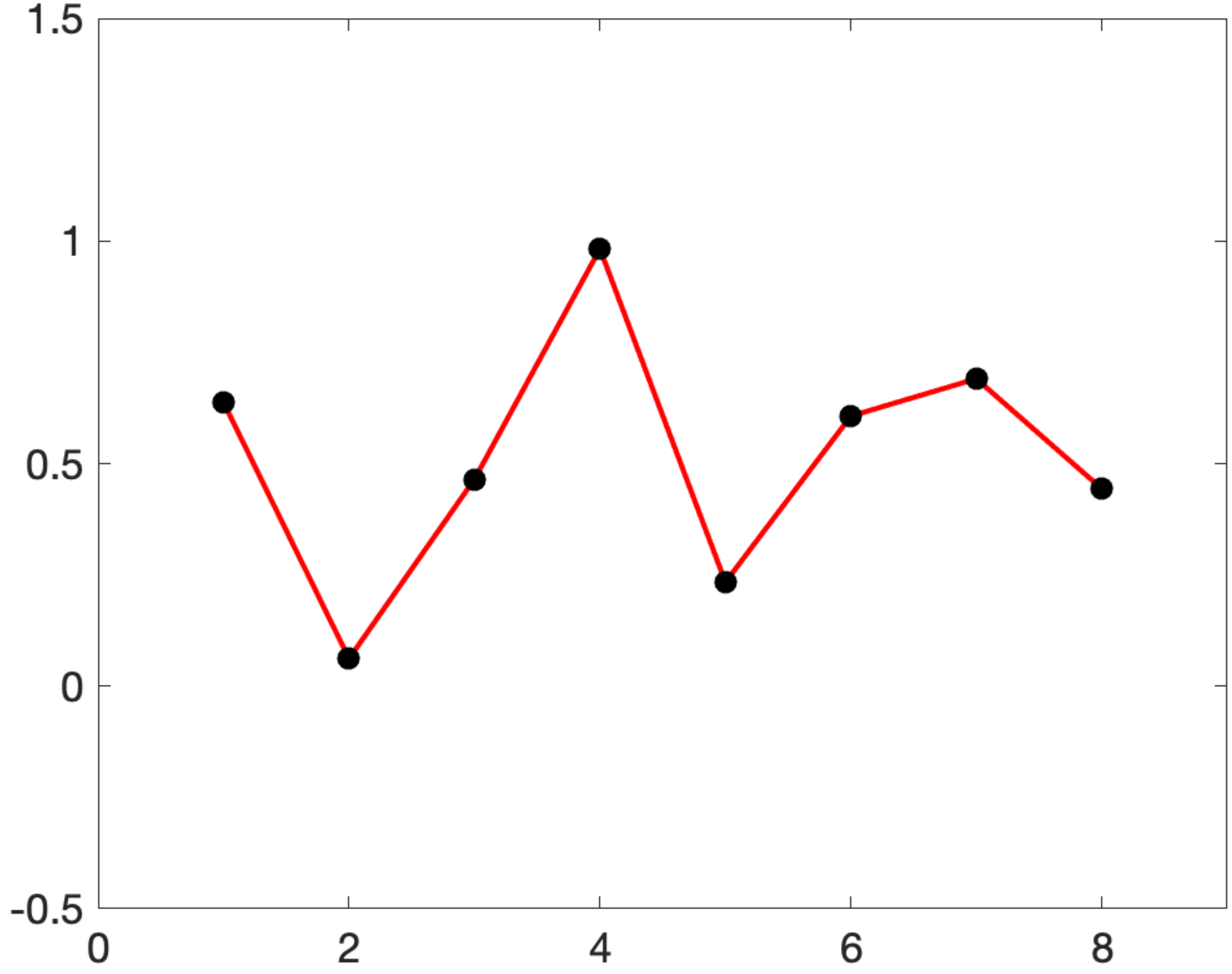
Single polynomial interpolation

1. Vandermonde matrix systems
2. Lagrange Polynomials
3. Barycentric formula

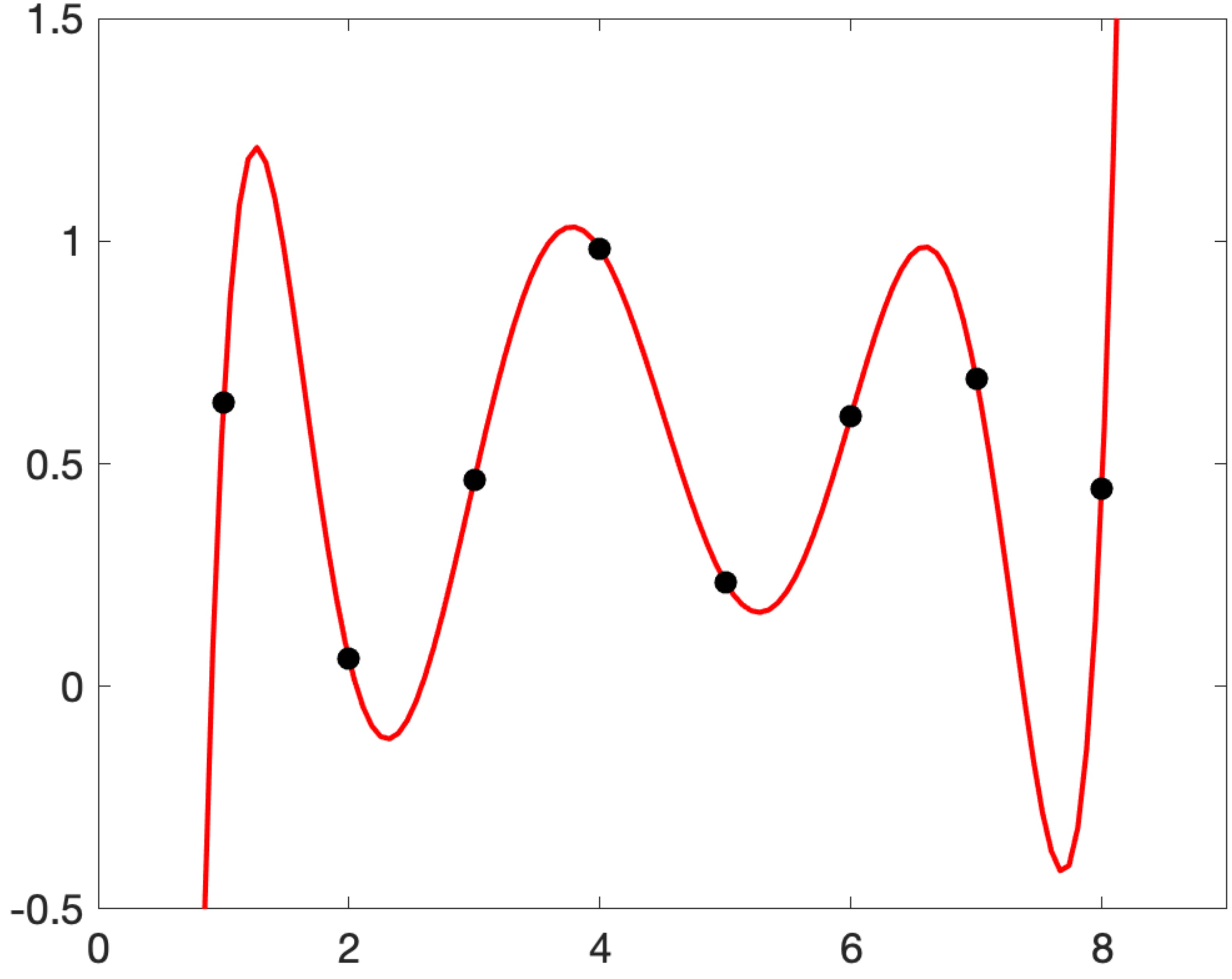
Basic Problem



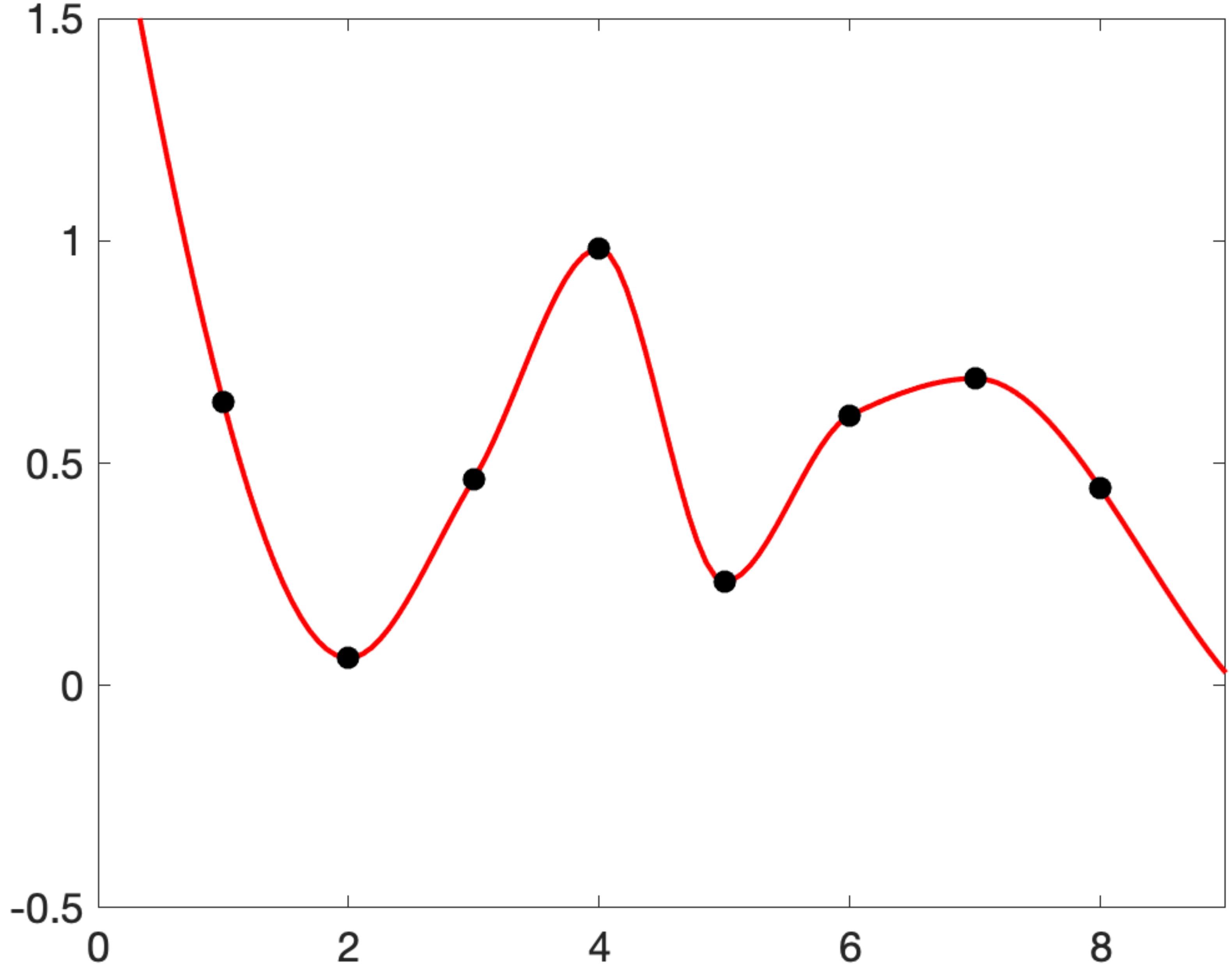
Piecewise linear



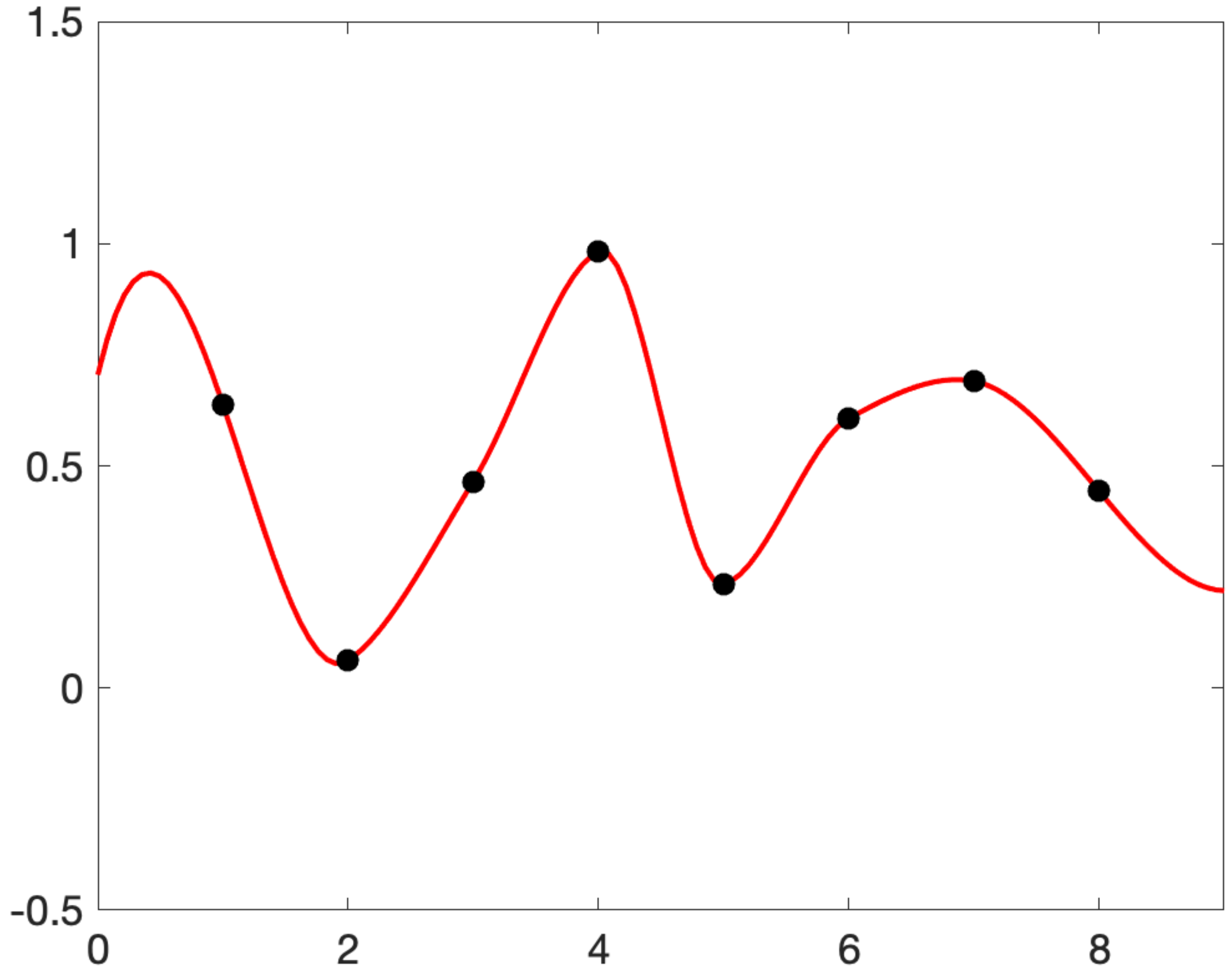
Polynomial Interpolation



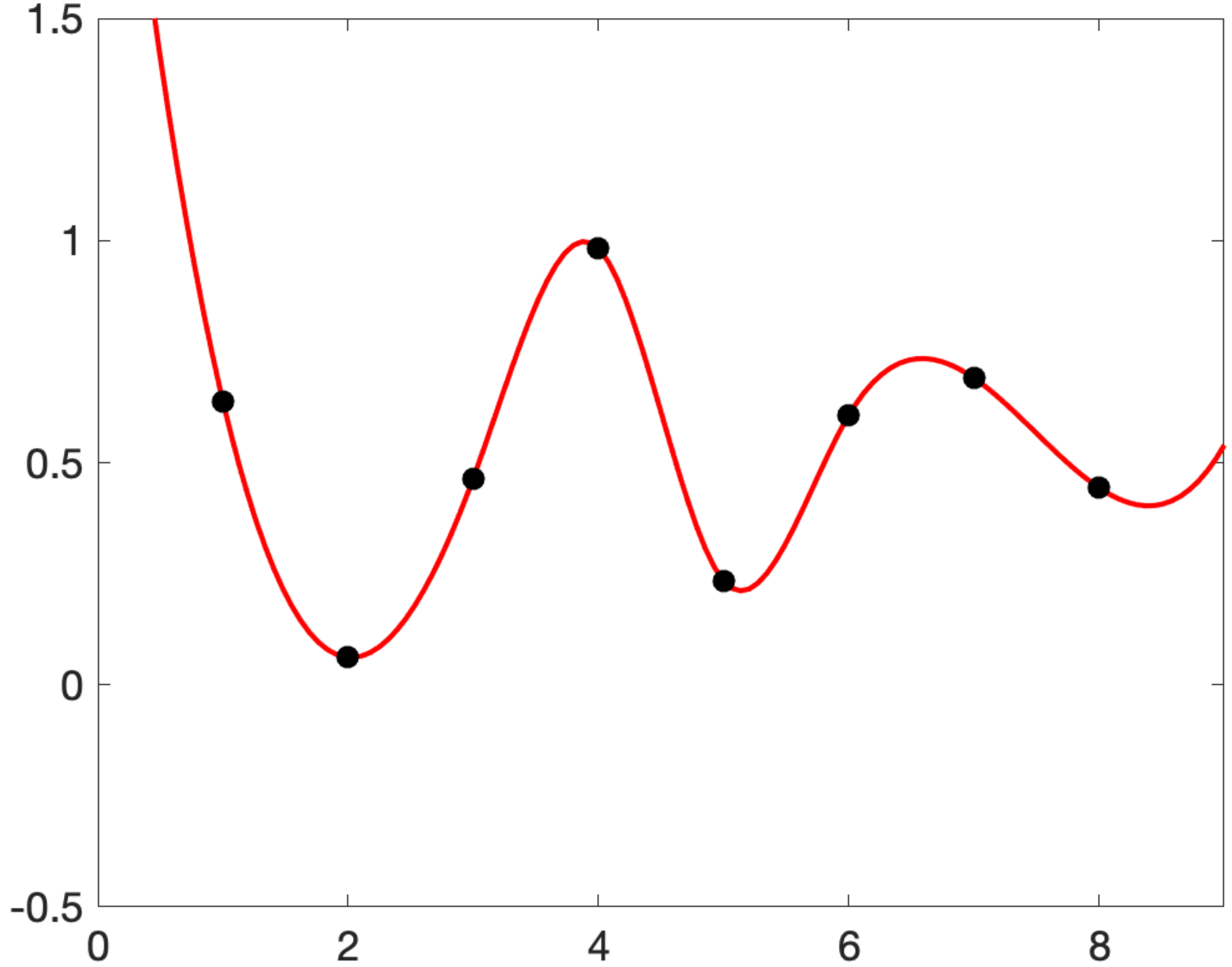
Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)



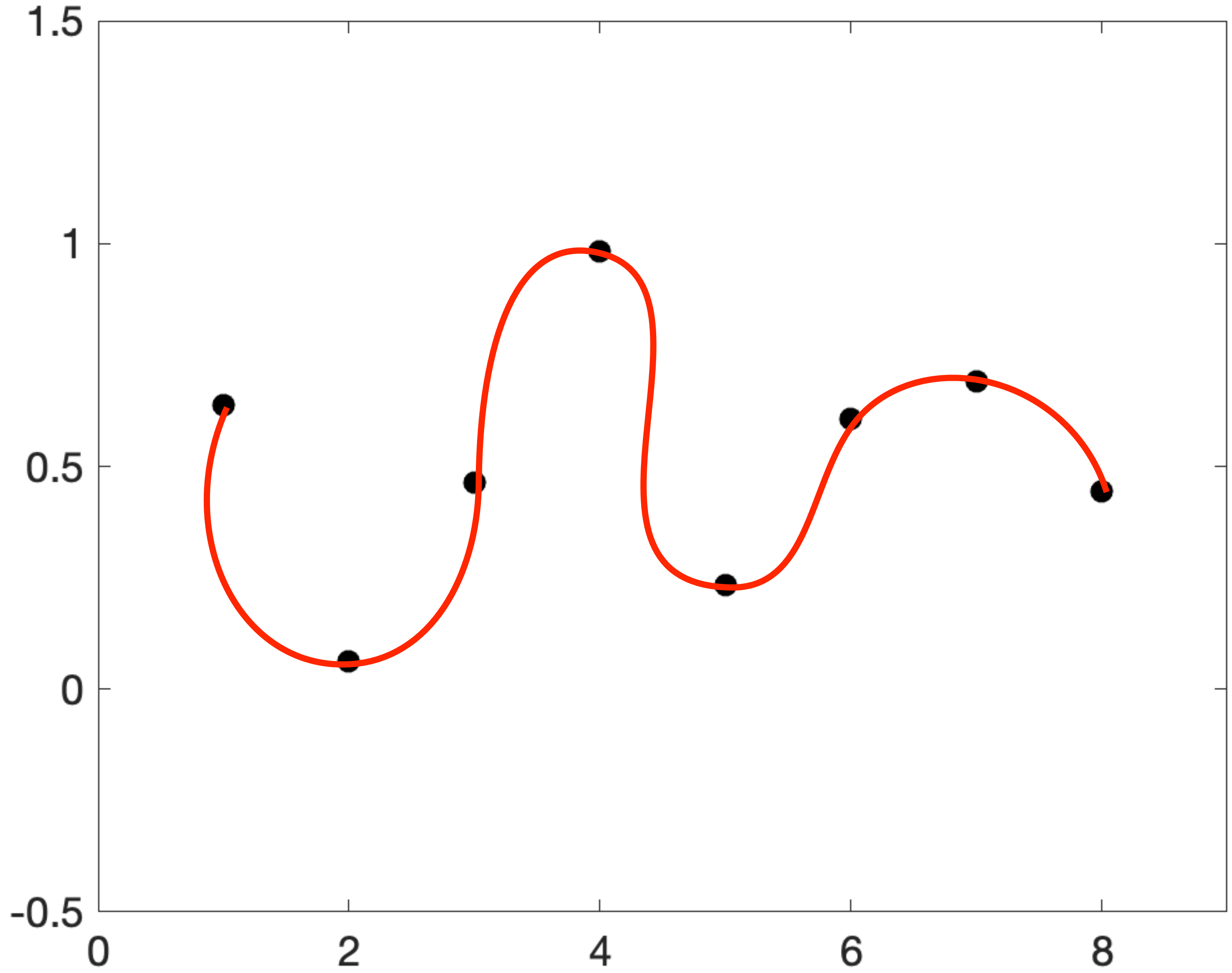
Akima PCHIP



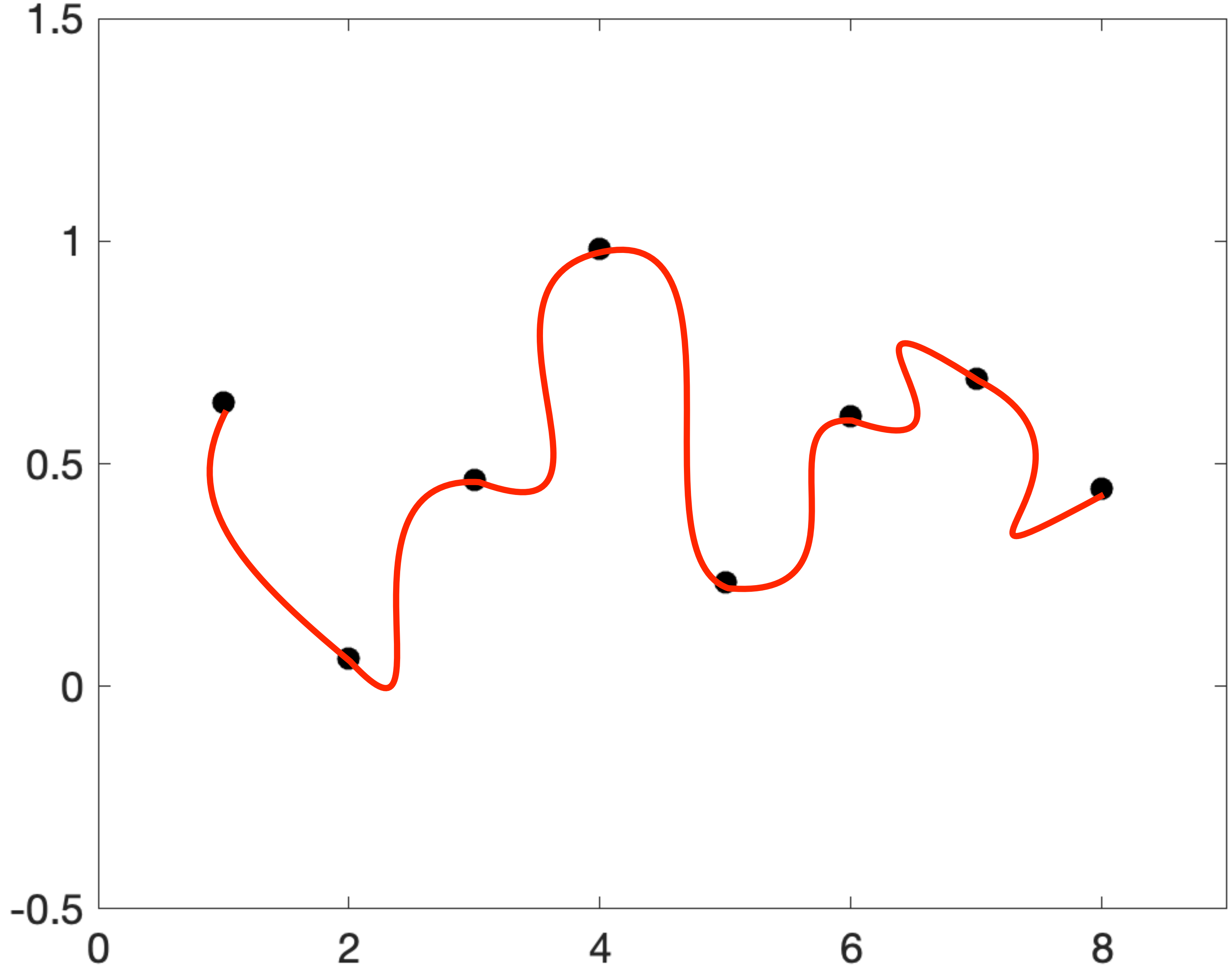
Spline interpolation



Smooth curve



Beziér Curve



Polynomial interpolation

- The data comes from a function, but you do not have an analytical form for the function (approximation theory)
- The data comes from experimental measurements (e.g. thermodynamic data)
- Construct a smooth geometric curve or surface through the data (computer graphics, automobile design)
- Numerical solutions to ordinary and partial differential equations almost all rely on polynomial approximations to functions.

Polynomial interpolation problem

Given a set of data points (x_i, y_i), for $i = 0, 1, 2, \ldots, n$, find an polynomial

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots a_1 x + a_0$$

that *interpolates* the data, i.e.

$$P(x_i) = y_i, \quad i = 0, 1, 2, \ldots, n$$

Questions : Is there only one polynomial? Does it have to be of degree n?

Vandermonde matrix system

Use the interpolation condition :

Each of $n+1$ data points (x_i, y_i) must then satisfy the condition that

$$P_n(x_i) = a_n x_i^n + a_{n-1} x_i^{n-1} + \ldots + a_1 x_i + a_0 = y_i$$

Such a system is called a *Vandermonde* system and can be written succinctly as

$$V\mathbf{a} = \mathbf{y}, \quad V \in R^{(n+1)\times(n+1)}$$

where the solution $\mathbf{a}$ contains the coefficients a_0, a_1, $\ldots$, a_n.

Vandermonde matrix system

The Vandermonde system matrix system :

$$\underbrace{\begin{bmatrix} x_0^n & x_0^{n-1} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}}$$

If we can invert the matrix, we can solve the system and find the unique polynomial that interpolates our data.

Inverting the Vandermonde matrix

How do we know we can solve this system?

Consider the 2×2 system for interpolating a line :

$$\begin{bmatrix} x_0 & 1 \\ x_1 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}$$

This system will have a unique solution if

$$\det(V) = (x_0 - x_1) \neq 0$$

$$P_1(x) = \left(\frac{y_0 - y_1}{x_0 - x_1}\right) x + \frac{y_1 x_0 - x_1 y_0}{x_0 - x_1}$$

$$m \qquad\qquad\qquad b$$

Inverting the Vandermonde system

Consider the 3×3 system for interpolating a second degree polynomial :

$$\begin{bmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

Again, this system will have a unique solution if

$$\det(V) = (x_0 - x_1)(x_0 - x_2)(x_1 - x_2) \neq 0$$

or that the x_i's are distinct.

Inverting the Vandermonde system

In general, the Vandermonde system will be invertible if

$$\det(V) = \prod_{0 \le i < j \le n} (x_i - x_j) \neq 0$$

or if the x_i's are distinct.

This also demonstrates that the polynomial that interpolates n+1 distinct points is *unique*, and in theory at least, the coefficients can be written down as

$$\mathbf{a} = V^{-1}\mathbf{y}$$

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

Uniform approximation

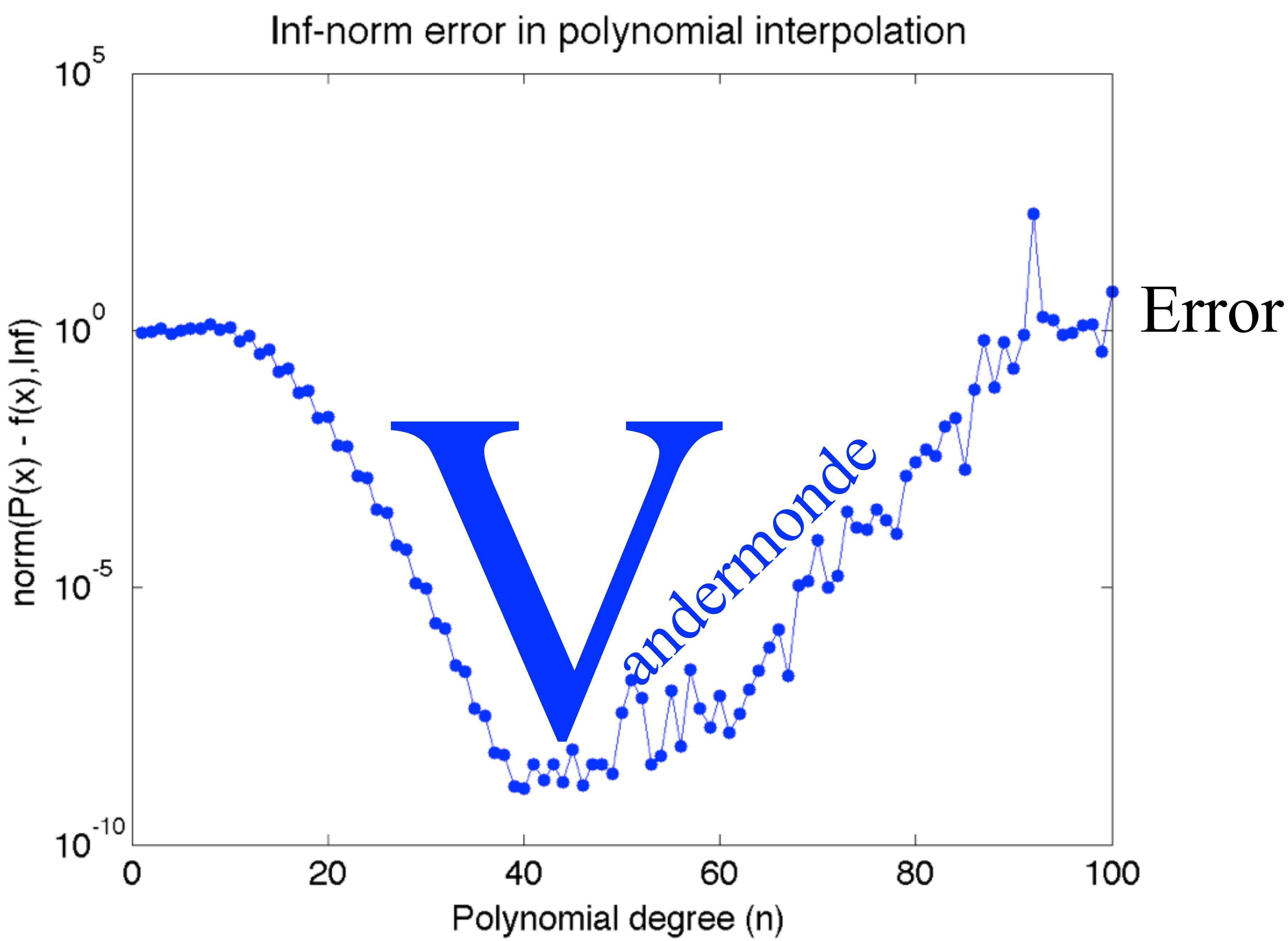
Weierstrauss Approximation Theorem. Let f be continuous on the closed interval $[a, b]$. Given any $\epsilon > 0$, there exists a polynomial P such that

$$\|f - P\|_\infty \equiv \max_{x \in [a,b]} |f(x) - P(x)| < \epsilon$$

One obvious example is the Taylor series polynomials.

$$f(x) = P_n(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

Vandermonde Matrix system



Vandermonde system

$$\underbrace{\begin{bmatrix} x_0^n & x_0^{n-1} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}}$$

We solved $V\mathbf{a} = \mathbf{y}$ to get the coefficients of the polynomial

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

Disadvantages to solving the Vandermonde system?

- Requires a linear solve (expensive)
- Potentially numerically ill-conditioned for large N.

Lagrange Polynomials

There are *explicit* (does not require a linear solve) ways of finding an interpolating polynomial through a given set of data points.

Given a set of data points (x_i, y_i), $i = 1, \ldots N+1$, suppose we had a set of polynomials $\ell_j(x)$ that satisfied

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

(Such a set of polynomials exist; the coefficients $\mathbf{a}_j$ for the j^{th} polynomial $\ell_j(x)$ *could* be found by solving the Vandermonde system $V\mathbf{a}_j = \mathbf{e}_j$, where $\mathbf{e}_j$ is the j^{th} column of the identity matrix. The $\mathbf{a}_j$ appear in the j^{th} column of V^{-1}.

Lagrange Polynomials

These polynomials are called the *Lagrange Interpolating Polynomials.*

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

and allow us to explicitly write down the polynomial that interpolates the data

$$P_n(x) = \sum_{j=0}^{n} \ell_j(x) y_j$$

Check : $P_n(x_i) = y_i$ (by construction). The n^{th} degree interpolating polynomial through $n+1$ points is unique, so we must have the same polynomial as was found by solving Vandermonde system

Lagrange basis functions

The Lagrange basis functions can be easily computed :

Let

$$\ell_j(x) = a \prod_{k=0,\ k\neq j}^{n} (x - x_k)$$

We want $\ell_j(x_j) = 1$, so we set

$$a = \frac{1}{\prod_{k=0,\ k\neq j}^{n}(x_j - x_k)}$$

and we have an explicit formula for the interpolating polynomial.

Lagrange Formulation

The Lagrange form of the interpolating polynomial is given by

$$P_n(x) = \sum_{j=0}^{n} \ell_j(x) y_j$$

where

$$\ell_j(x) = \frac{\prod_{k=0,\ k\neq j}^{n} (x - x_k)}{\prod_{k=0,\ k\neq j}^{n} (x_j - x_k)}$$

Example - Fitting a quadratic

Find the parabola that fits through 3 data points :

$$(-1, 2), \quad (0, 3), \quad (2, -7)$$

$$\begin{aligned}
\ell_0(x) &= \frac{(x-0)(x-2)}{(-1-0)(-1-2)} &= \frac{1}{3}x^2 - \frac{2}{3}x \\
\ell_1(x) &= \frac{(x+1)(x-2)}{(0+1)(0-2)} &= \frac{1}{2}x^2 + \frac{1}{2}x + 1 \\
\ell_2(x) &= \frac{(x+1)(x-0)}{(2+1)(2-0)} &= \frac{1}{6}x^2 + \frac{1}{6}x
\end{aligned}$$

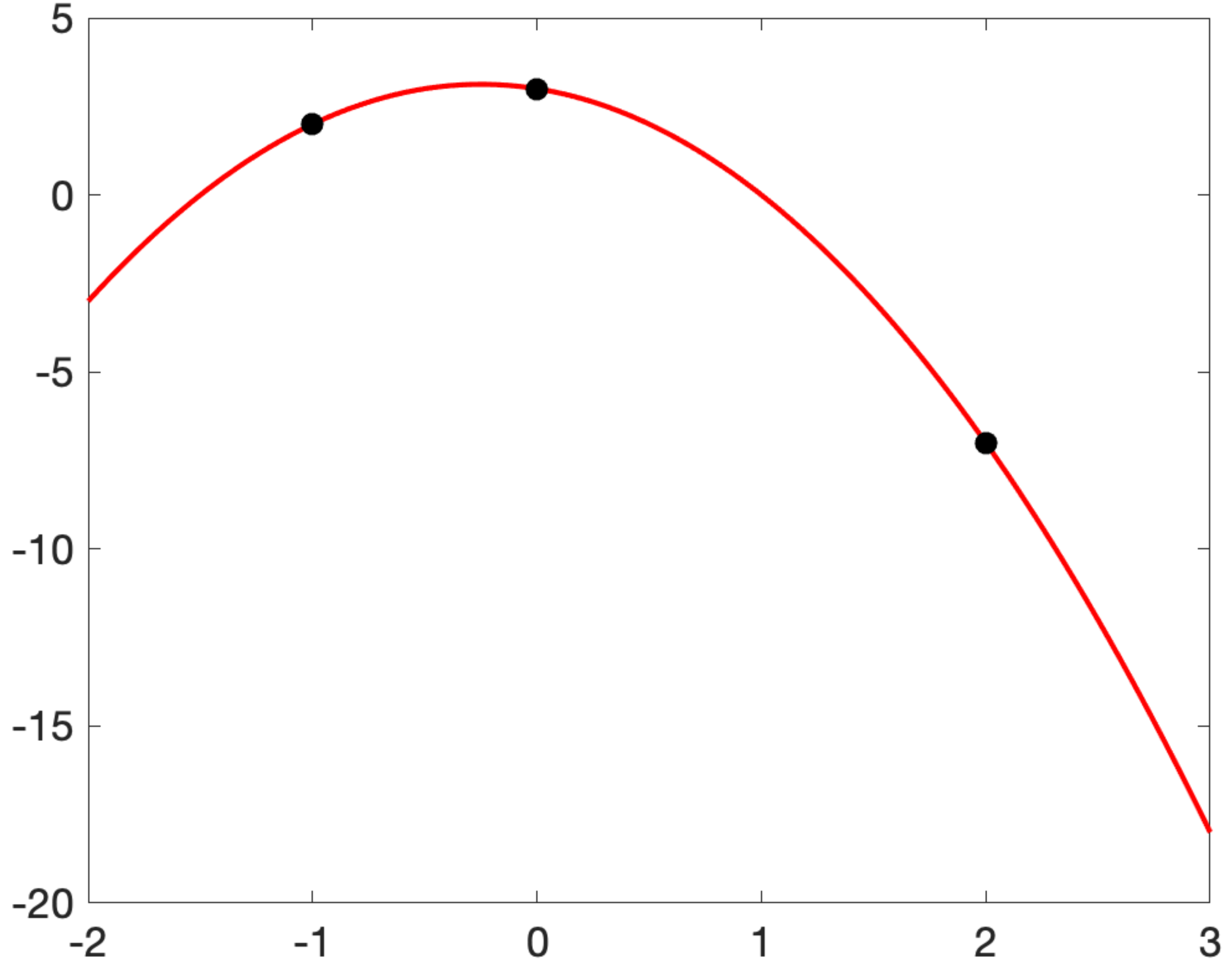
Check that $\ell_i(x_i) = 1$ and that $\ell_i(x_j) = 0,\ i \neq j$.

Check

The interpolating polynomial is then

$$P_2(x) = 2\ell_0(x) + 3\ell_1(x) - 7\ell_2(x) = \underline{-2x^2 - x + 3}$$

Parabolic fit



$$P_2(x) = 2\ell_0(x) + 3\ell_1(x) - 7\ell_2(x) = -2x^2 - x + 3$$

Lagrange vs. Vandermonde

The Vandermonde matrix system solves for the *coefficients* of the polynomial, expressed as

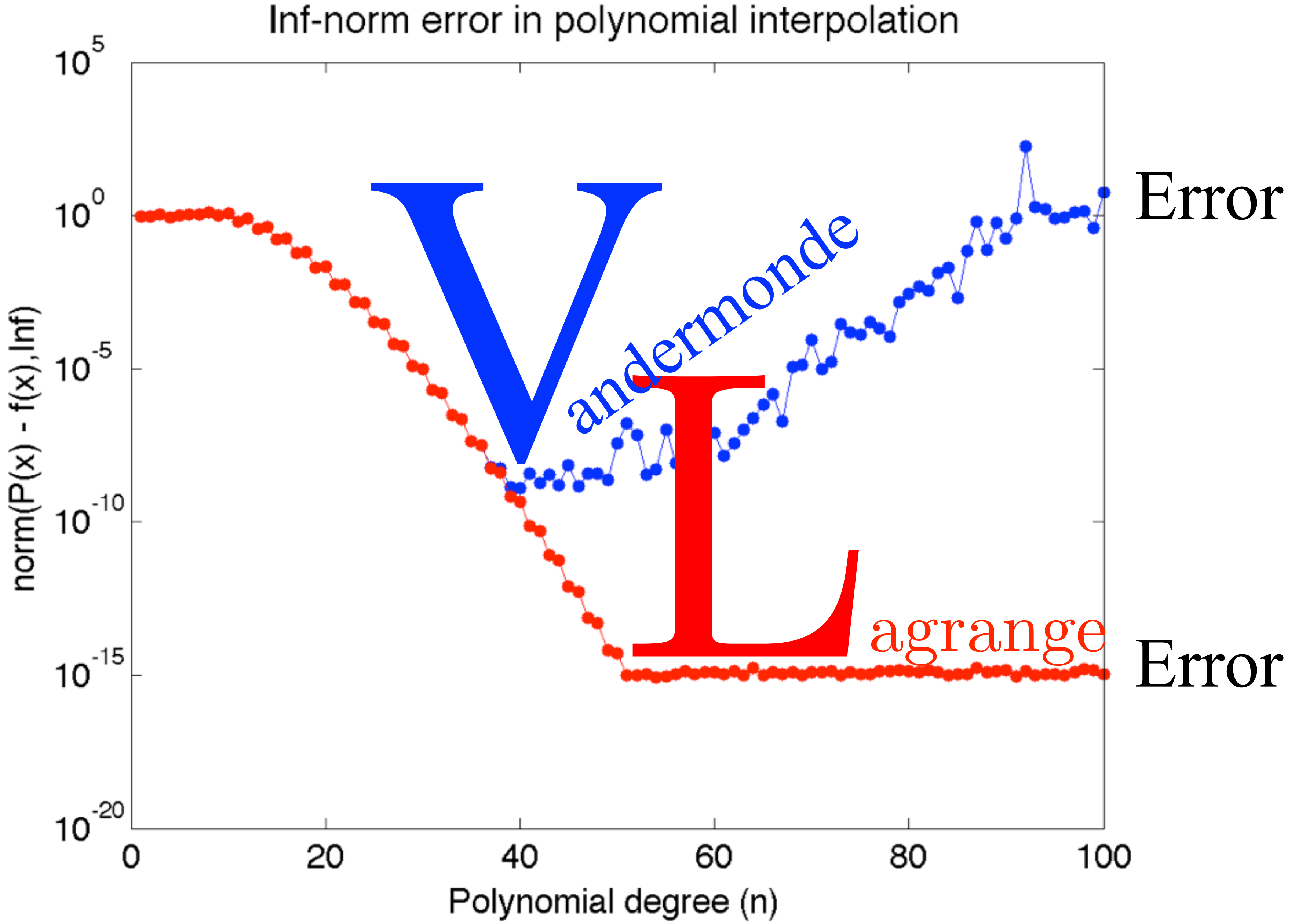
$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

The Lagrange formula returns $P_n(x)$ in the form

$$P_n(x) = \sum_{j=0}^{n} \ell_j(x) y_j$$

Both solve for the same interpolating polynomial, but the Lagrange formula is an explicit form, and produces much better error results.

Lagrange Interpolation



Lagrange interpolation is much more stable then inverting the Vandermonde system (but it is slow...)

Polynomial evaluation

If we use the Vandermonde matrix form, we should evaluate the polynomial using *Horner's Method*. This is implemented in the Matlab function `polyval`. This will be $\mathcal{O}(n)$ operations.

If we use the Lagrange Formula directly, we evaluate n degree n polynomials. This would be an $\mathcal{O}(n^2)$ operations.

What we need is an $\mathcal{O}(n)$ method for evaluating the Lagrange Forumula. The *Barycentric* formula does this for us.

Lagrange Formula evaluation

$$P_n(x) = \sum_{j=0}^{n} \ell_j(x) y_j$$

This requires that we evaluate $n + 1$ polynomials $\ell_j(x)$. We can write each $\ell_j(x)$ as

$$\ell_j(x) \equiv \omega_j \left(\frac{\prod_{k=0}^{n}(x - x_k)}{x - x_j} \right) \equiv \ell(x) \frac{\omega_j}{x - x_j}$$

where

Coefficient designed to force $\ell_j(x_j) = 1$

$$\ell(x) \equiv \prod_{k=0}^{n} (x - x_k)$$

The idea is to evaluate $\ell(x)$ only once per polynomial evaluation.

Barycentric interpolation formula

Define $$\ell(x) = \prod_{k=0}^{n} (x - x_k)$$

This is not a Lagrange polynomial

Define $$\omega_j = \frac{1}{\prod_{k=0,k\neq j}^{n} (x_j - x_k)}$$

Denominator in the Lagrange Polynomial

Then $$\ell_j(x) = \ell(x) \frac{\omega_j}{x - x_j}$$

This is the jth Lagrange polynomial

and $$P_n(x) = \ell(x) \sum_{j=0}^{n} \frac{\omega_j}{x - x_j} y_j$$

First Barycentric Form

Barycentric interpolation formula

We have that
$$\sum_{j=0}^{n} \ell_j(x) = 1$$

We then have that
$$\sum_{j=0}^{n} \ell_j(x) = \sum_{j=0}^{n} \ell(x) \frac{\omega_j}{x - x_j} = \ell(x) \sum_{j=0}^{n} \frac{\omega_j}{x - x_j} = 1$$

or
$$\ell(x) = \frac{1}{\sum_{j=0}^{n} \frac{\omega_j}{x - x_j}}$$

which leads to the form

$$P_n(x) = \frac{\sum_{j=0}^{n} \frac{\omega_j}{x - xj} y_j}{\sum_{j=0}^{n} \frac{\omega_j}{x - x_j}}$$

Second (true) Barycentric Form