PHY604 Lecture 5

September 9, 2025

Today’s lecture:

- Finish discussing Numerical Integration

- Begin discussing interpolation
  - Lagrange Interpolation
  - Cubic splines

Choosing an integration method (Newman Sec. 5.7)

- Trapezoid method:
    - Trivial to program
    - Equally spaced points, often true of experimental data
    - Good choice for poorly behaved data (noisy, singularities)
    - Adaptive method gives guaranteed accuracy level
    - Not very accurate for given number of points
- Romberg integration:
    - Equally spaced points, often true of experimental data
    - Guaranteed accuracy level
    - Potentially high accuracy for small number of points
    - Will not work well for noisy of pathological data/integrands
- Gaussian Quadrature
    - Potentially high accuracy for small number of points
    - Simple to program (weights and roots tabulated)
    - Will not work well for noisy of pathological data/integrands
    - Need to have data on specific, unequally-spaced grid

Today’s lecture:

- Finish discussing Numerical Integration
- Begin discussing interpolation
    - Lagrange Interpolation

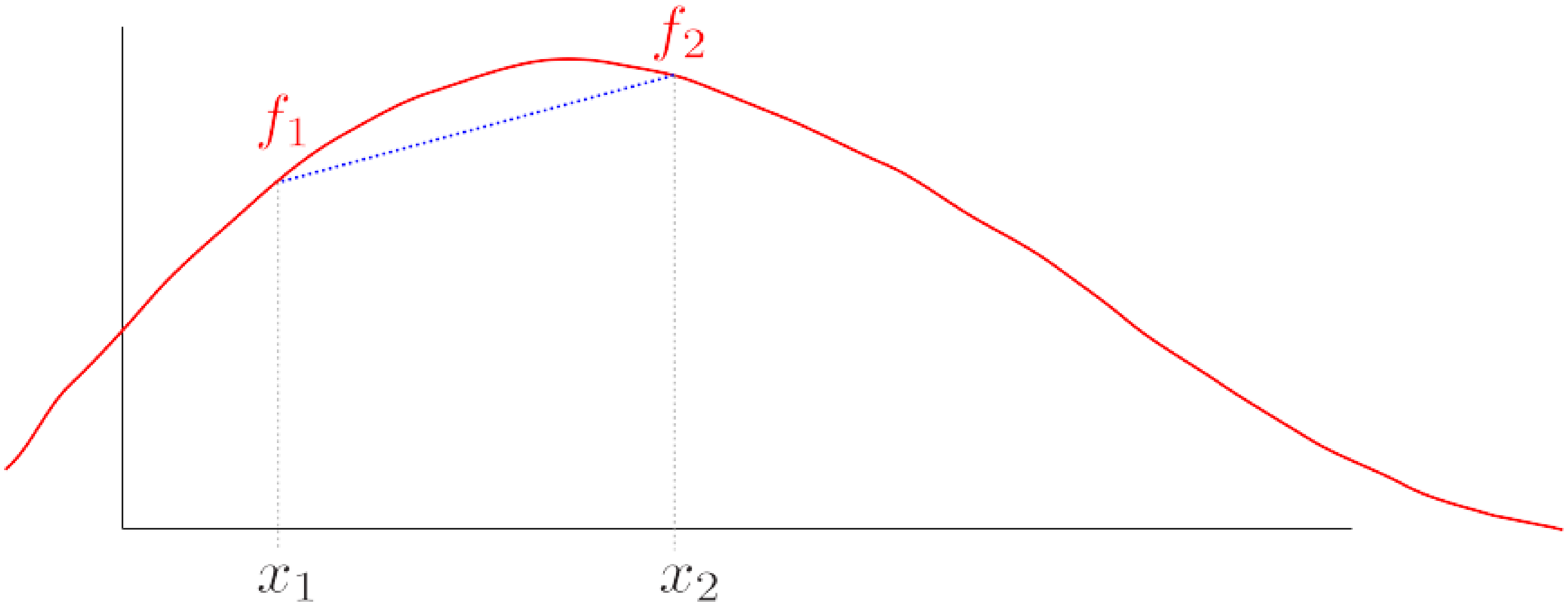
Interpolation (see Pang Ch. 2)

- Interpolation is needed when we want to infer some local information from a set of incomplete or discrete data
  - E.g., experimental data or from computational simulations

- Many different types of interpolation based on assumptions about and requirements for the data
  - Some ensure no new extrema are introduced
  - Some match derivatives at end points
  - Need to balance number of points used against pathologies (e.g., oscillations)

- Interpolations and fitting are different!
  - *Interpolation* seeks to fill in missing information in some small region of the whole dataset
  - *Fitting* a function to the data seeks to produce a model (guided by physical intuition) so you can learn more about the global behavior of your data

Linear interpolation:
Draw a line between two points



$$f(x) = \frac{f_2 - f_1}{x_2 - x_1}(x - x_1) + f_1$$

Errors in linear interpolation

- Exact value at *x*: $f(x) = f_i + \frac{x - x_i}{x_{i+1} - x_i}(f_{i+1} - f_i) + \Delta f(x)$

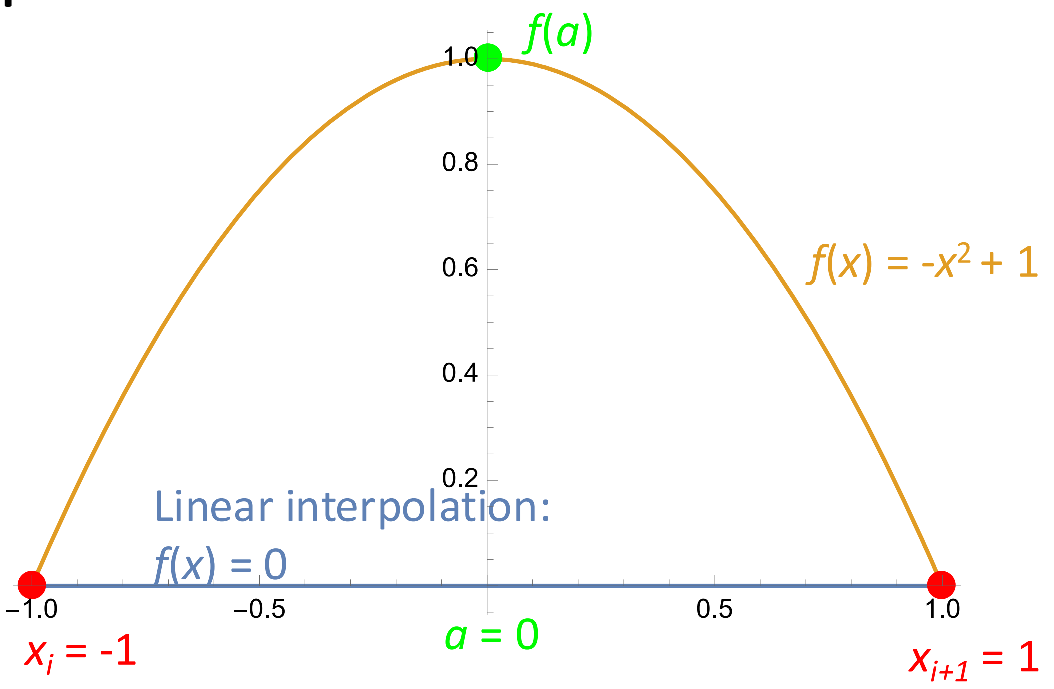
  Linear interpolant

- What is $\Delta f(x)$?
  - Consider point x = *a* where *a* is in $[x_i, x_{i+1}]$
  - Fit a quadratic to the function at x_i, *a*, x_{i+1}

$$\Delta f(x) = \frac{f''(x)}{2}(x - x_i)(x - x_{i+1})\bigg|_{x=a}$$

- As long as *f* is smooth in the region $[x_i, x_{i+1}]$
- Error of order: $\mathcal{O}(\Delta x^2)$
- Max error: $|\Delta f(x)| \leq \frac{\max[|\Delta f''(x)|]}{8}(x_{i+1} - x_i)^2$

Simple example of errors in linear interpolation:



$$\Delta f(a) = \frac{-2}{2}(-1)(1) = 1$$

- General case: Fit a parabola as we did for Simpson's rule

General approach for interpolation schemes

- Continuous curve is constructed from given discrete set of data
- Interpolated value is read off the curve
- The more points, the higher order the curve can be
- One way to achieve higher-order interpolation is through Lagrange interpolation

Lagrange interpolation

- General method for building a single polynomial that goes through all the points (alternate formulations exist)

- Given n points: x_0, x_1, ... , $x_{n\text{-}1}$, with associated function values: f_0, f_1, ..., $f_{n\text{-}1}$
    - Construct basis functions: $$l_i(x) = \prod_{j=0, i \neq j}^{n-1} \frac{x - x_j}{x_i - x_j}$$
    - Note basis function l_i is 0 at all x_j except for x_i (where it is one)
    - Function value at x is: $$f(x) = \sum_{i=0}^{n-1} l_i(x) f_i$$

Example: Quadratic Lagrange polynomial

- Three points: (x_0,f_0), (x_1,f_1), (x_2,f_2)
- Three basis functions:

$$l_0 = \frac{x - x_1}{x_0 - x_1}\frac{x - x_2}{x_0 - x_2} = \frac{(x - x_1)(x - x_2)}{2\Delta x^2}$$

$$l_1 = \frac{x - x_0}{x_1 - x_0}\frac{x - x_2}{x_1 - x_2} = -\frac{(x - x_0)(x - x_2)}{\Delta x^2}$$

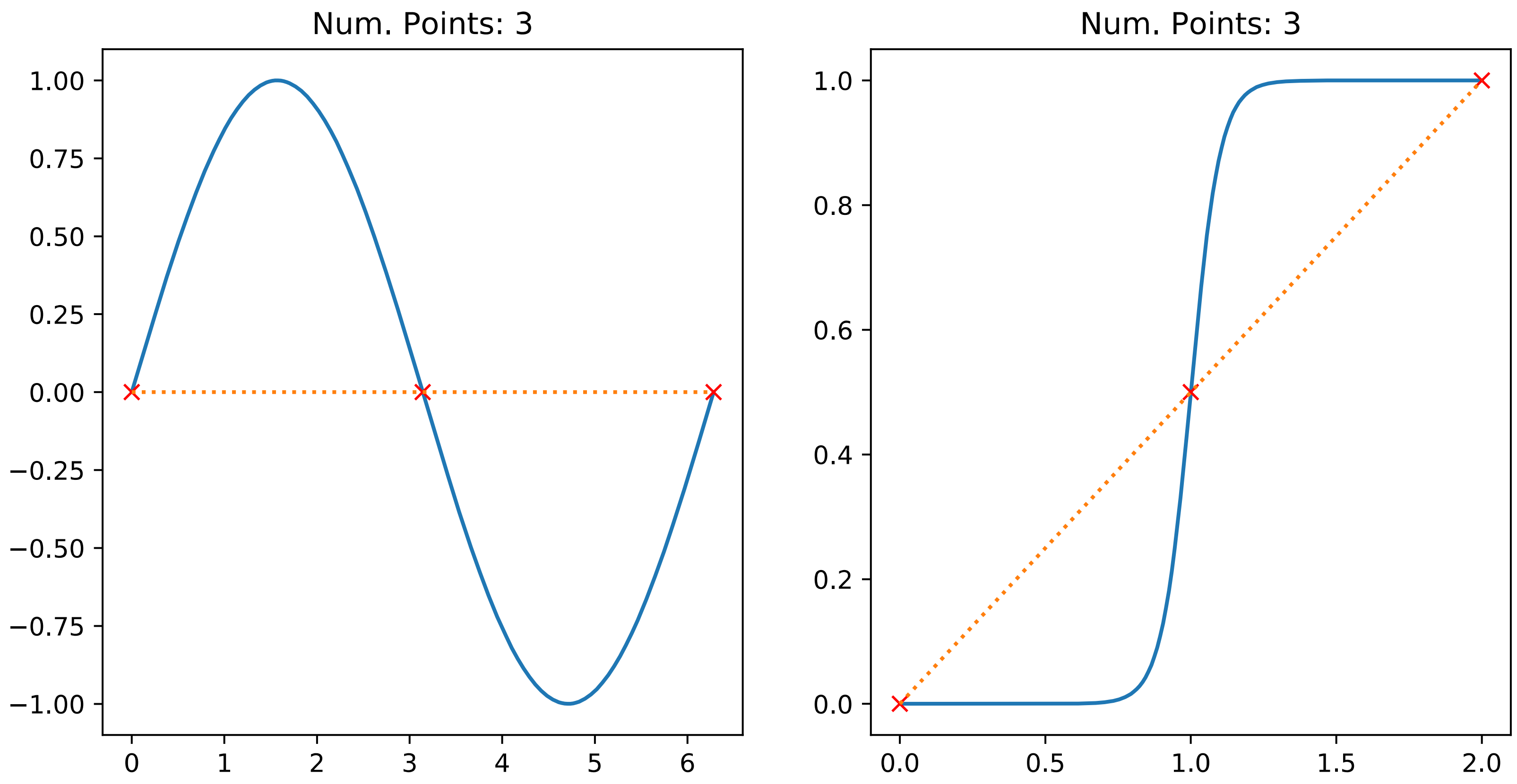
$$l_2 = \frac{x - x_0}{x_2 - x_0}\frac{x - x_1}{x_2 - x_1} = \frac{(x - x_0)(x - x_1)}{2\Delta x^2}$$

- Polynomial:

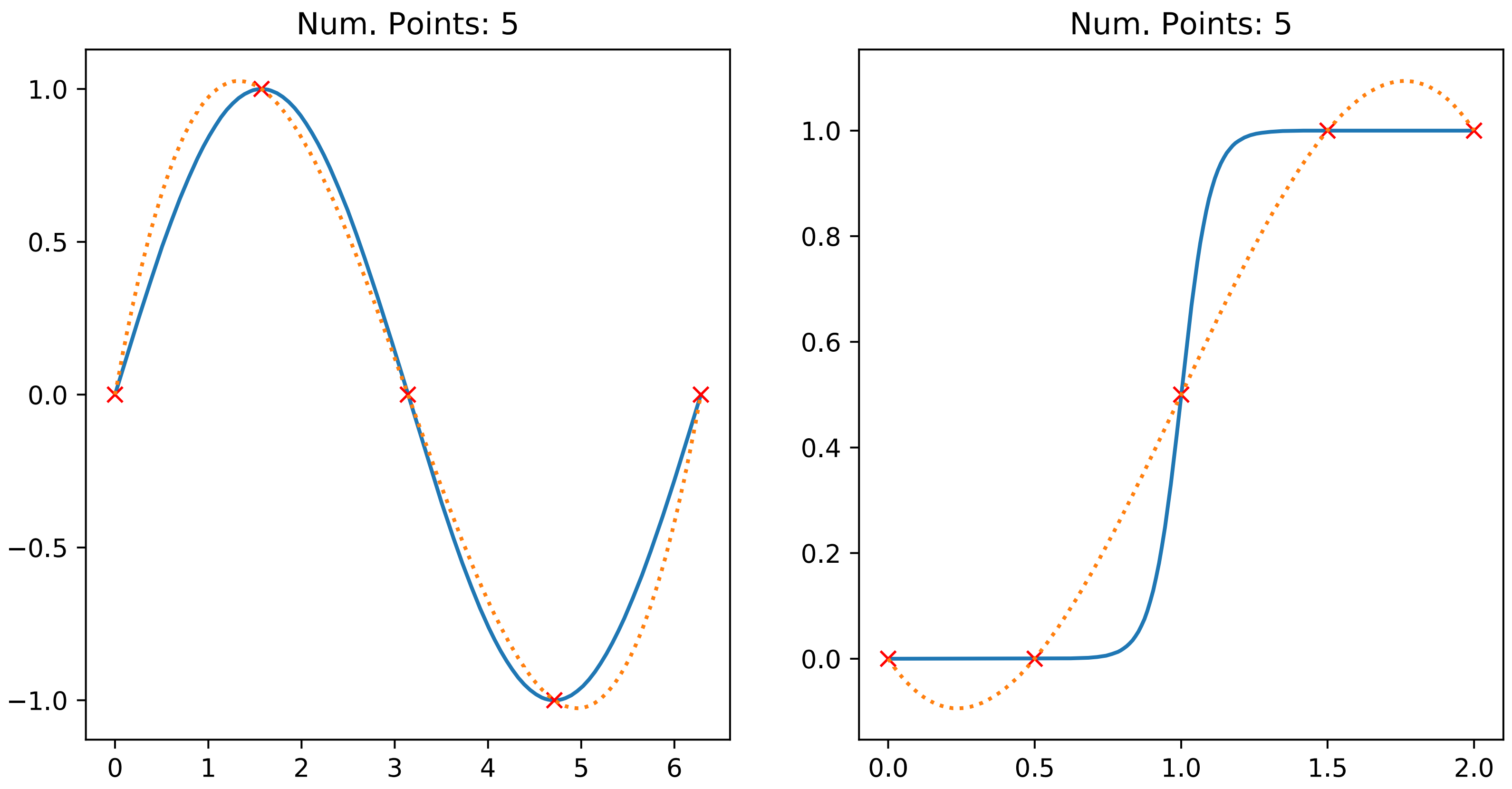
$$f(x) = f_0\frac{(x - x_1)(x - x_2)}{2\Delta x^2} - f_1\frac{(x - x_0)(x - x_2)}{\Delta x^2} + f_2\frac{(x - x_0)(x - x_1)}{2\Delta x^2}$$

D

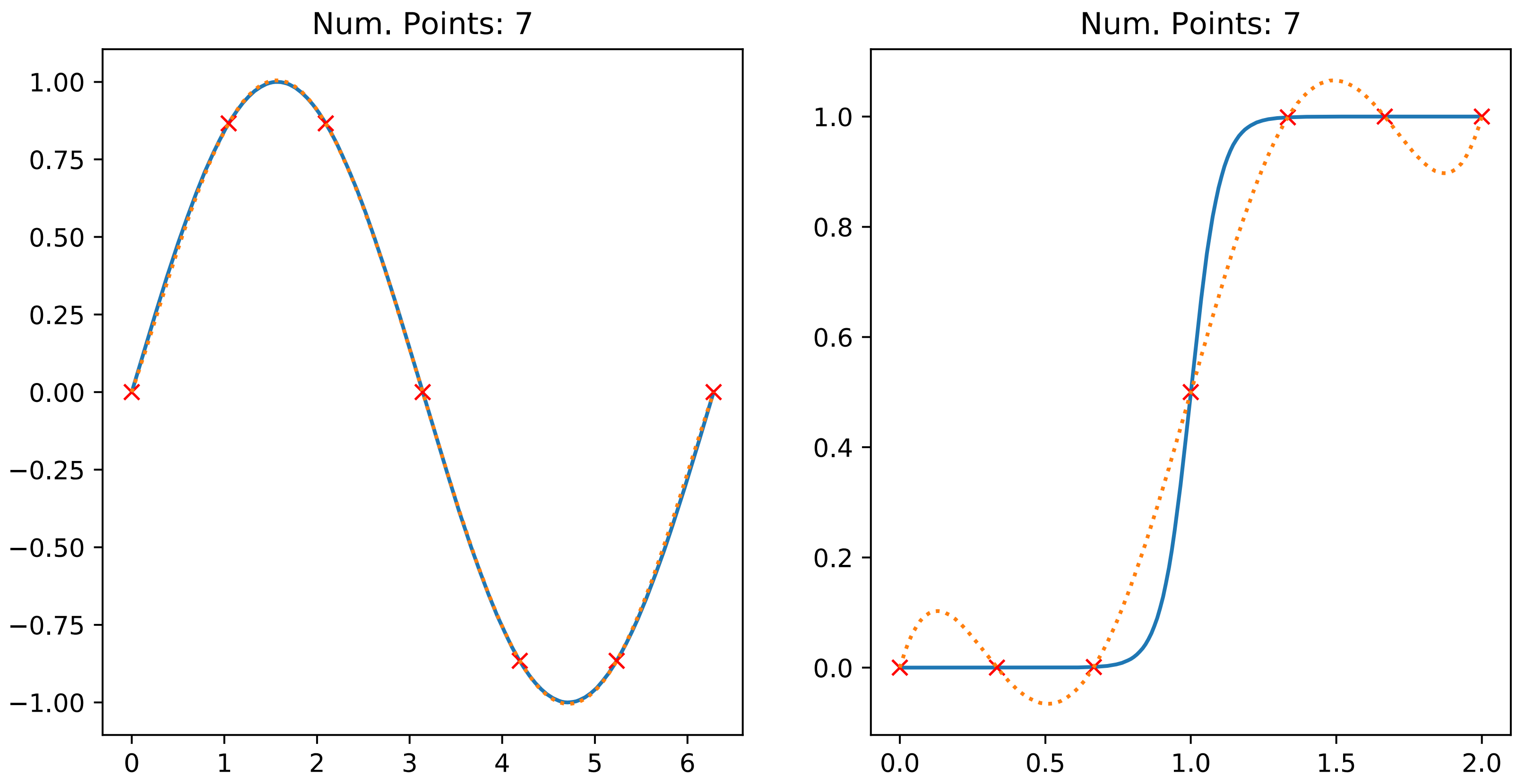
Example: Lagrange Interpolation of two functions



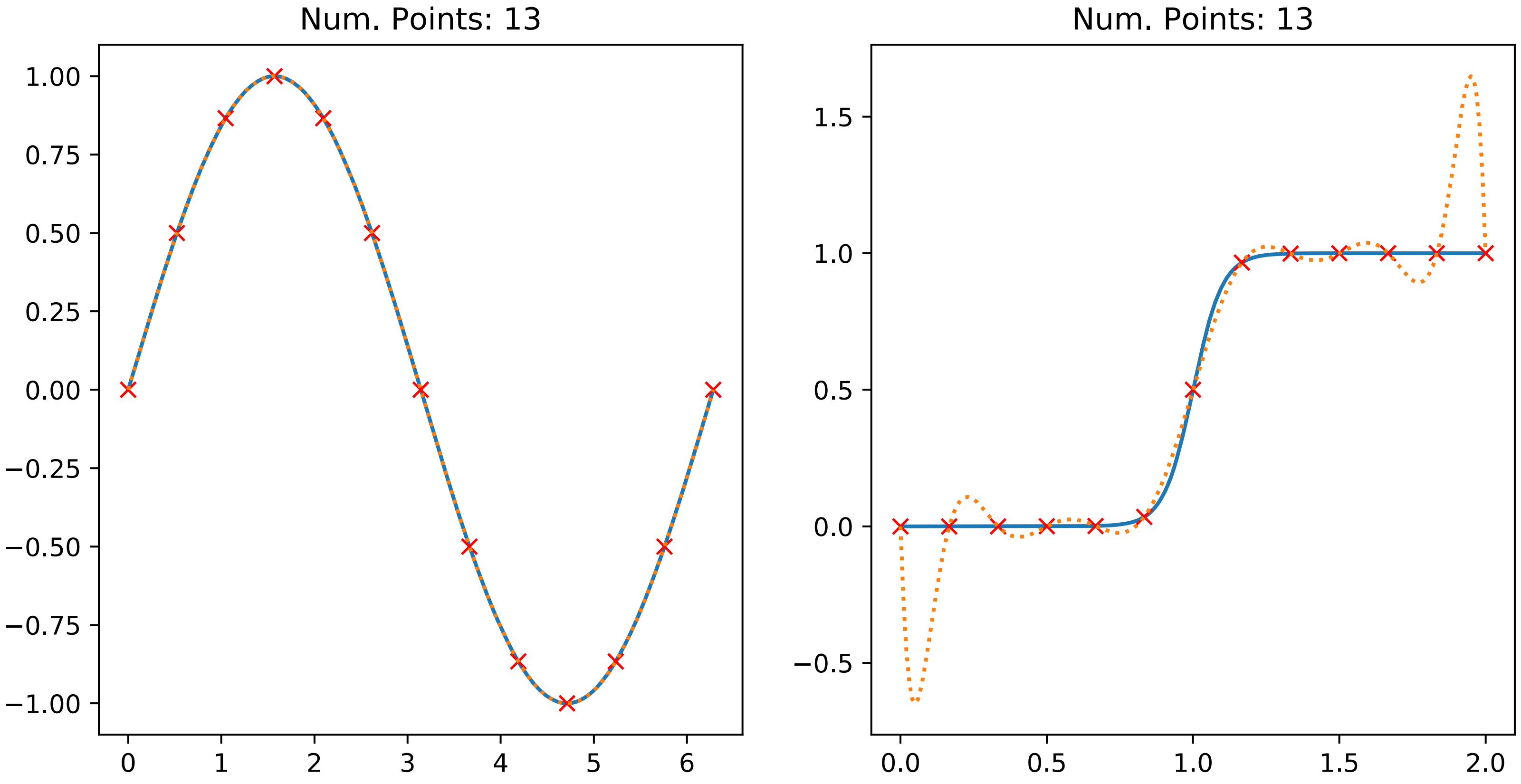
Example: Lagrange Interpolation of two functions



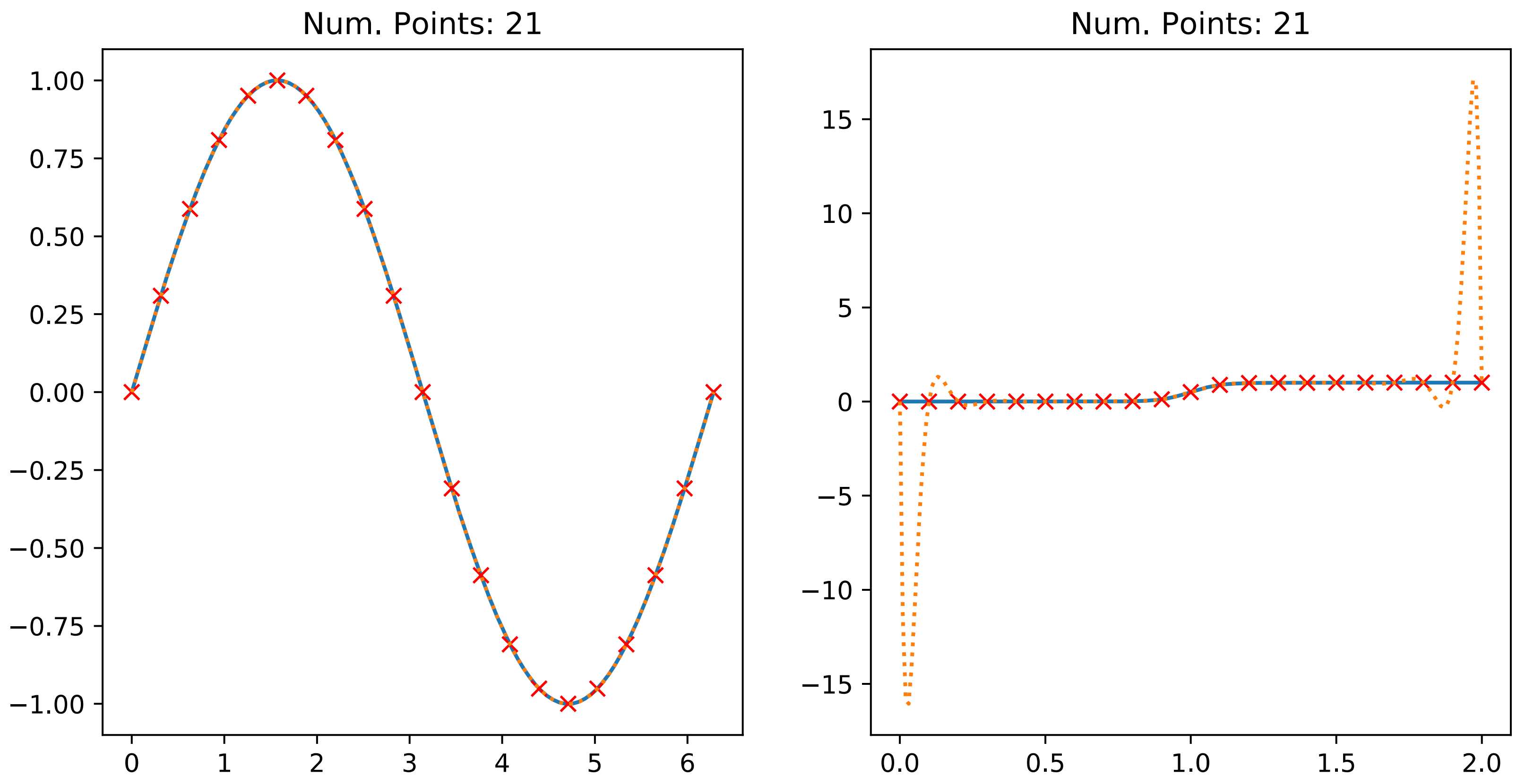
Example: Lagrange Interpolation of two functions



Example: Lagrange Interpolation of two functions



Example: Lagrange Interpolation of two functions

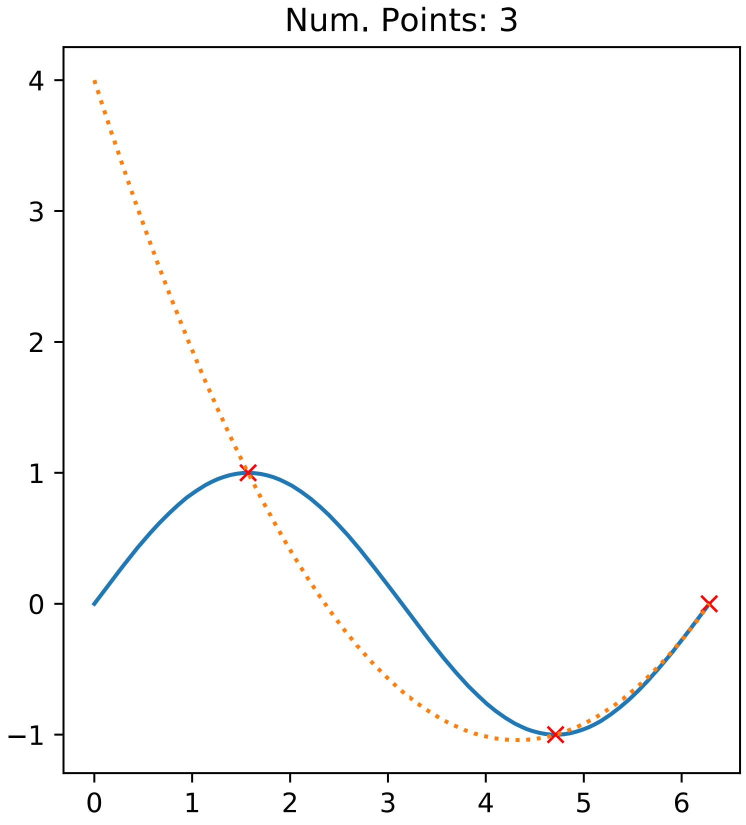


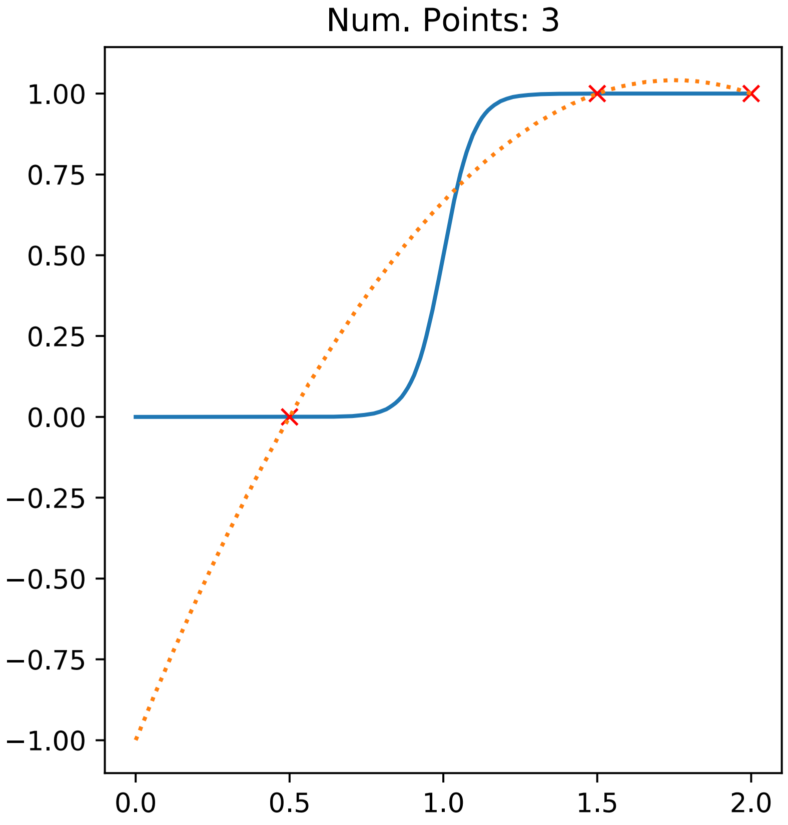
Example: Lagrange Interpolation of two functions **MORE IS NOT ALWAYS BETTER**

- For the hyperbolic tangent case, increasing the number of points beyond a certain limit increases the error
  - Runge phenomena: Oscillations at the edges of the interval
  - Increasing the number of points causes a divergence in the error
- Can do better by varying the spacing of the interpolating points
  - e.g., Chebyshev polynomial roots are concentrated toward the end of the interval
  - Chebyshev polynomial spacing is usually (almost always) convergent with the number of interpolating points

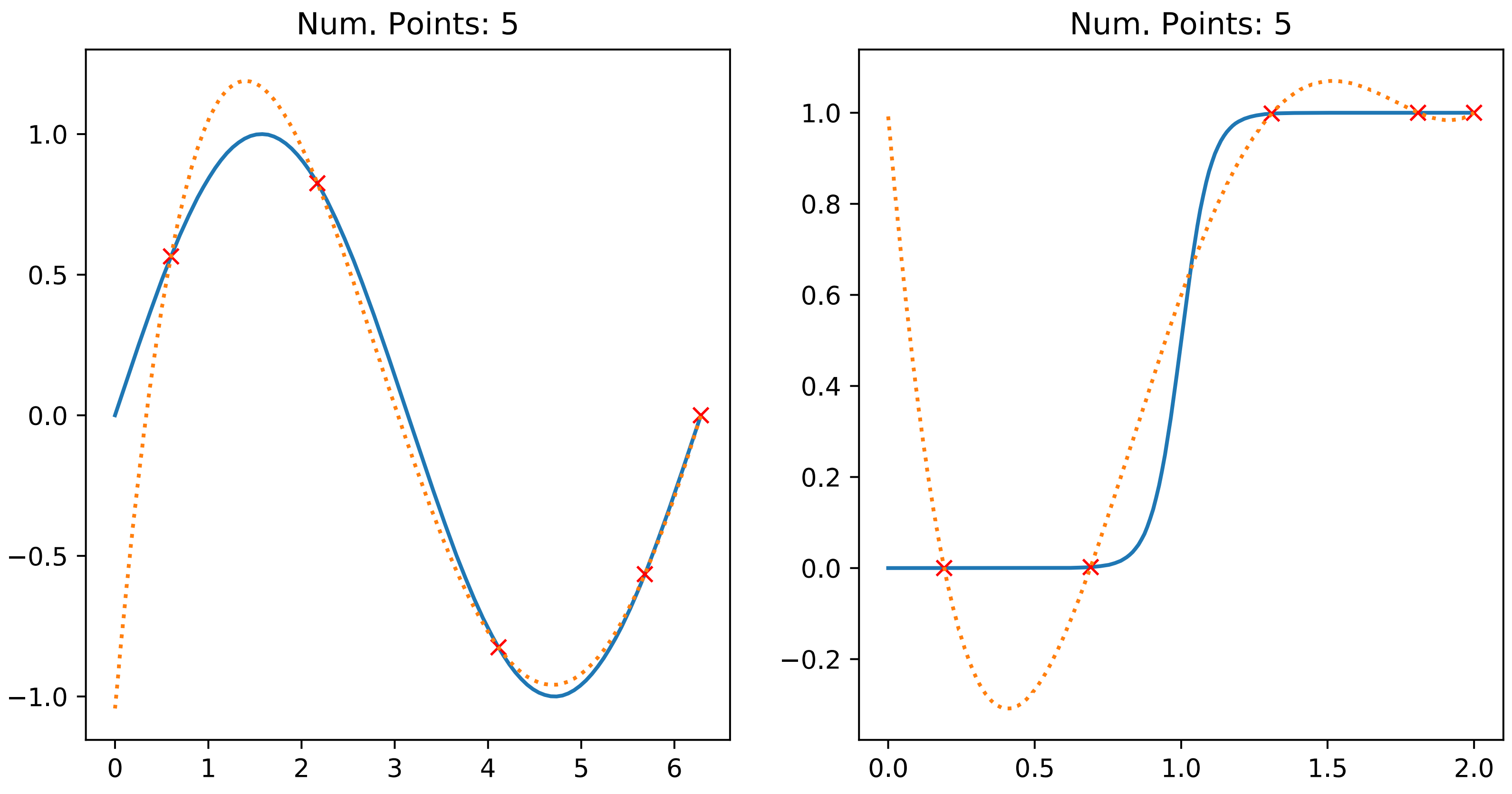
$$x_k = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2k+1}{2n}\pi\right), \quad k = 0, ..., n-1$$

Example: Lagrange Interpolation of two functions with Chebyshev nodes

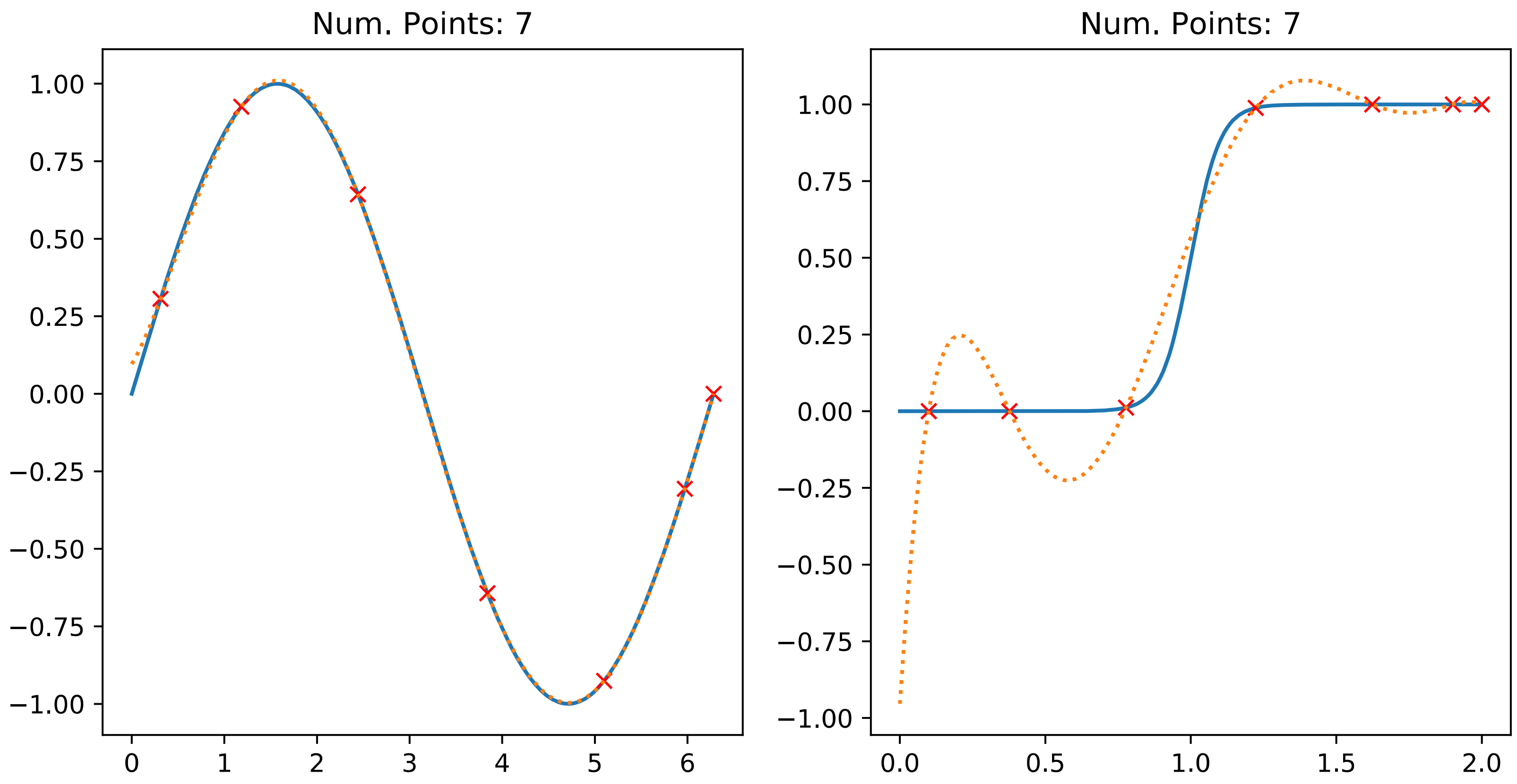




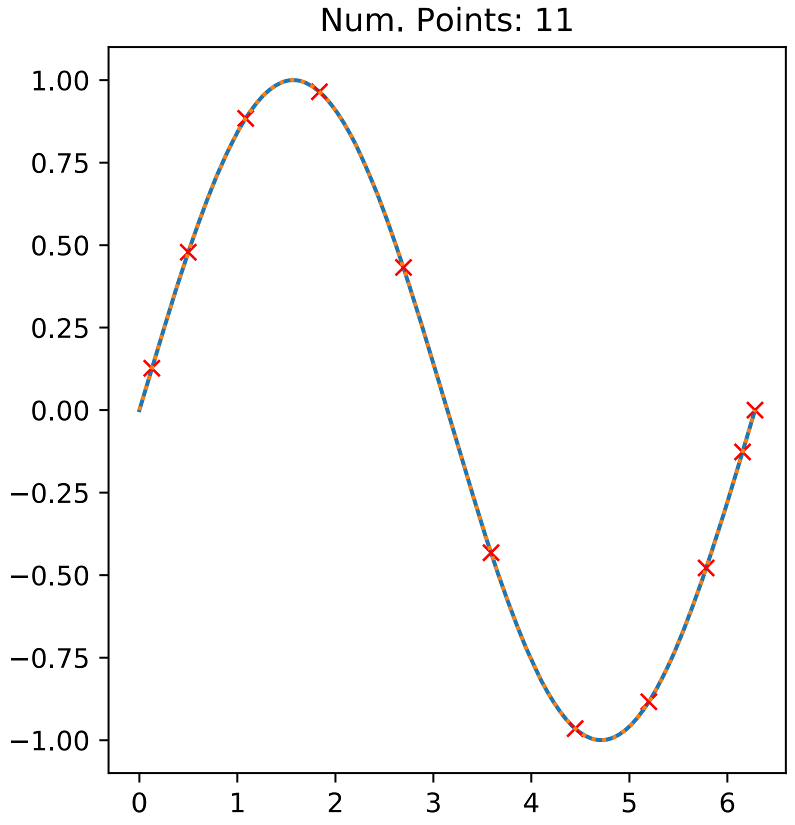
Example: Lagrange Interpolation of two functions with Chebyshev nodes

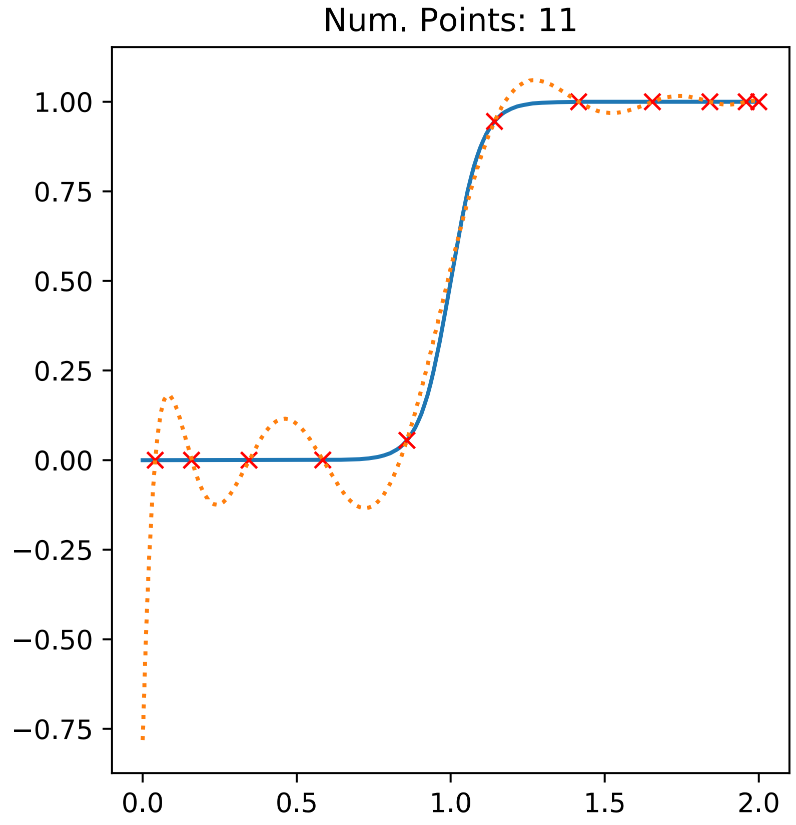


Example: Lagrange Interpolation of two functions with Chebyshev nodes

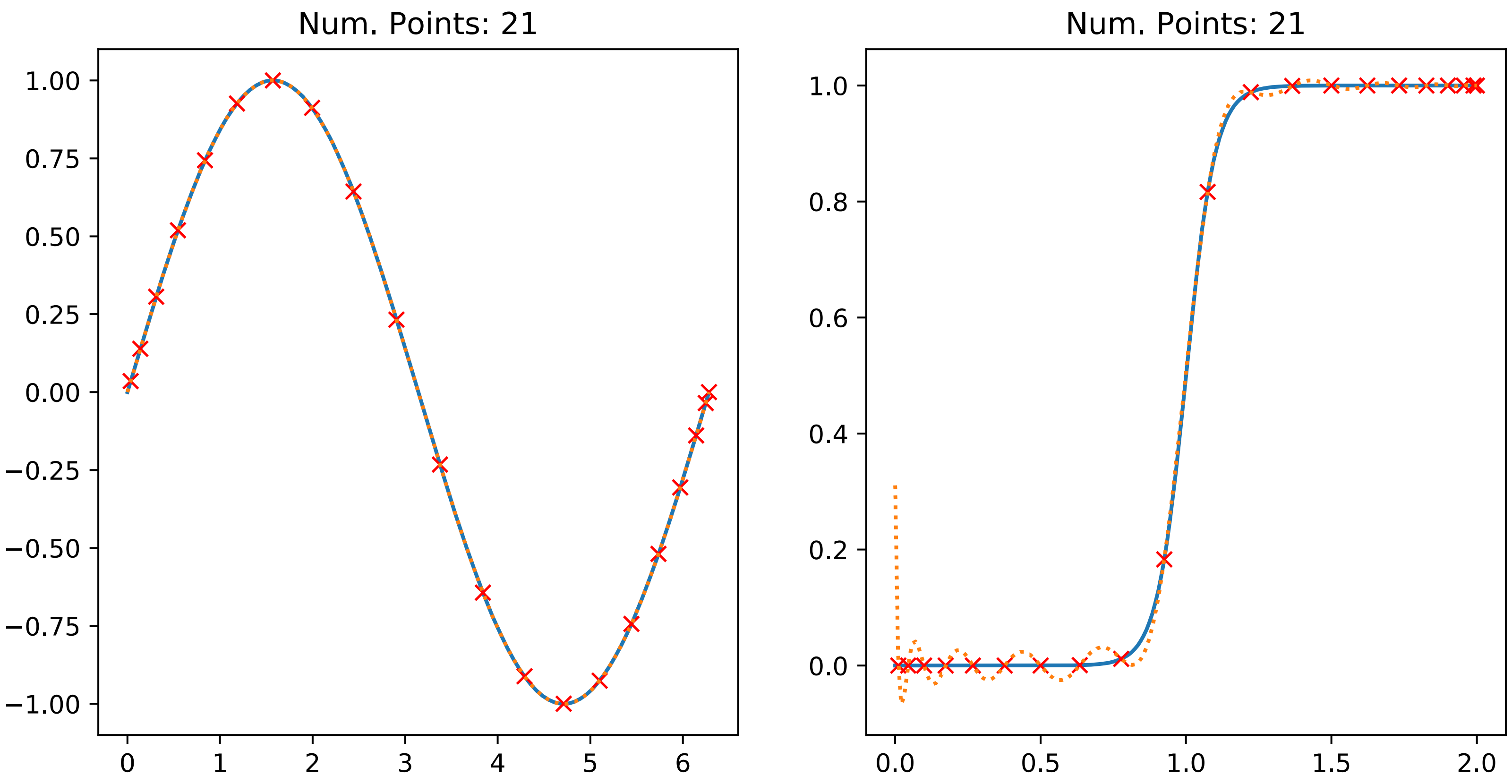


Example: Lagrange Interpolation of two functions with Chebyshev nodes

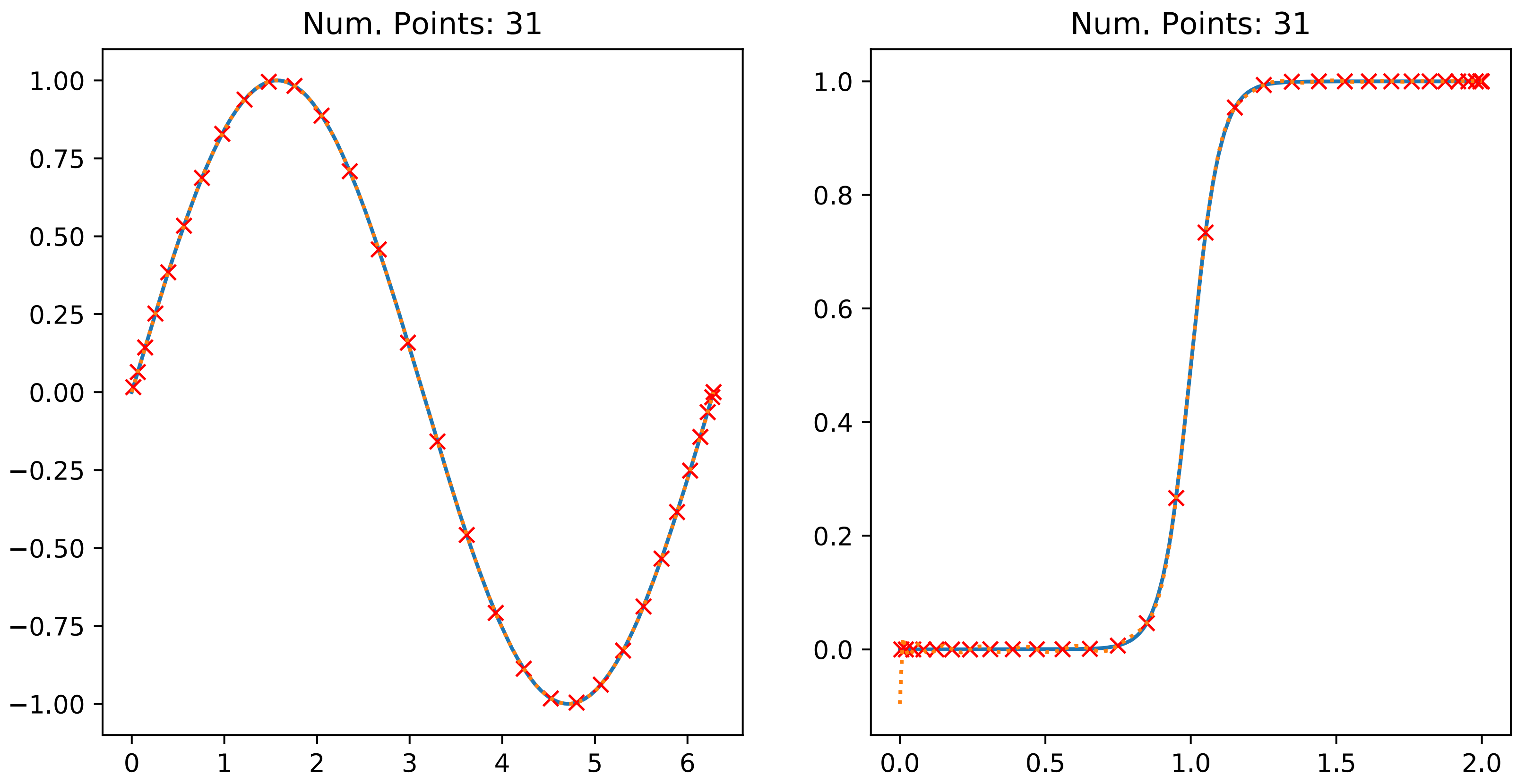




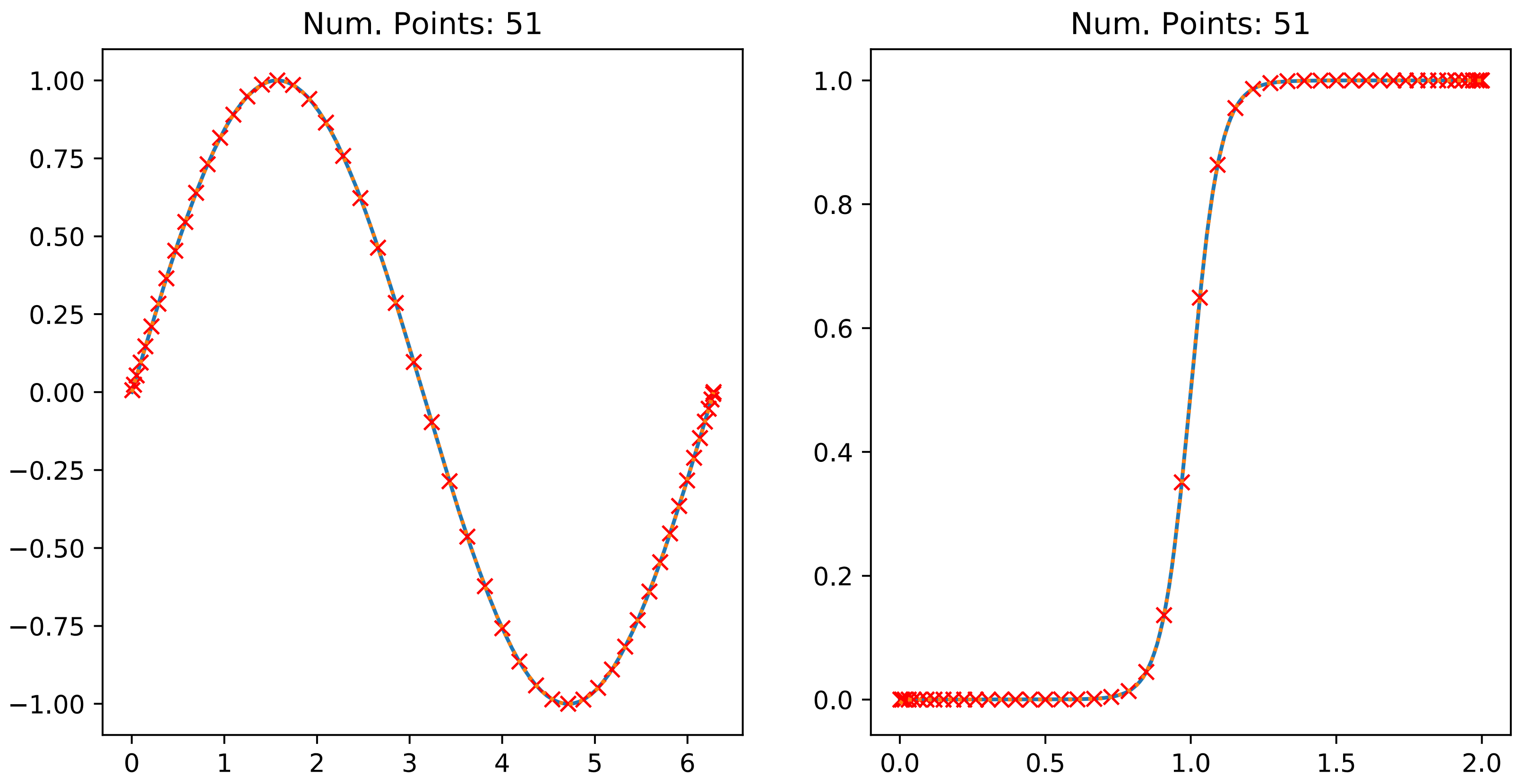
Example: Lagrange Interpolation of two functions with Chebyshev nodes



Example: Lagrange Interpolation of two functions with Chebyshev nodes



Example: Lagrange Interpolation of two functions with Chebyshev nodes



Today’s lecture:

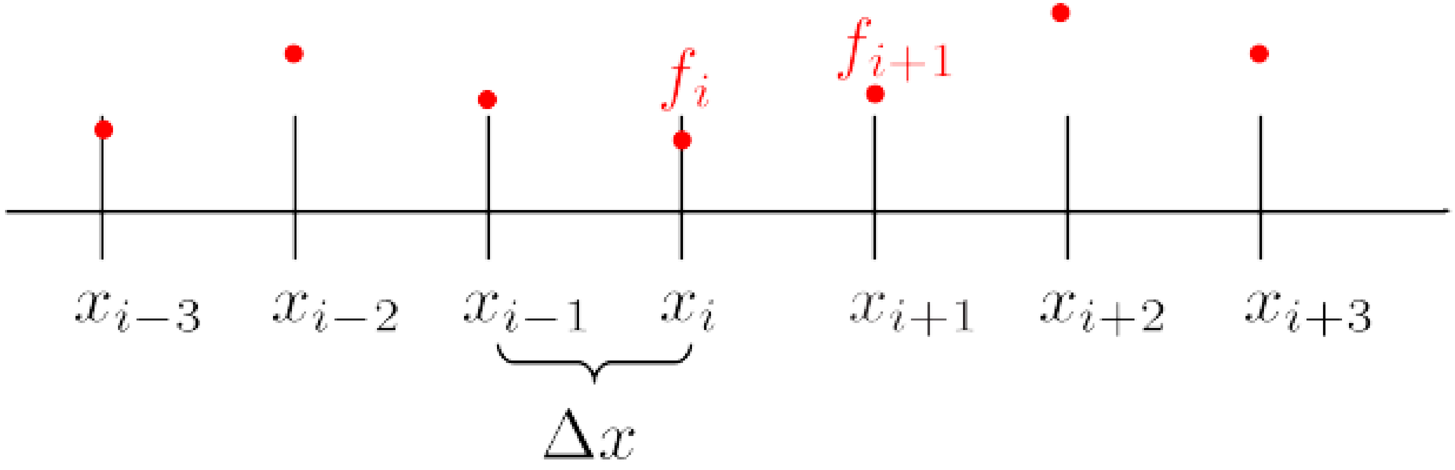
- Finish discussing Numerical Integration

- Begin discussing interpolation
  - Lagrange Interpolation
  - Cubic splines

Splines (Pang Sec. 2.4)

- So far, we've only worried about going through the specified points
- Large number of points → two distinct options:
  - Use a single high-order polynomial that passes through them all
  - Fit a (somewhat) high order polynomial to *each interval* and match all derivatives at each point—this is a spline
- Splines match the derivatives at end points of intervals
  - Piecewise splines can give a high-degree of accuracy
- Cubic spline is the most popular
  - Matches first and second derivative at each data point
  - Results in a smooth appearance
  - Avoids severe oscillations of higher-order polynomial

Splines



- We have a set of regular-spaced discrete data: $f_i=f(x_i)$ at $x_0,x_1,x_2,\ldots,x_n$
- m-th order polynomial to approximate $f(x)$ for x in $[x_i,x_{i+1}]$:

$$p_i(x) = \sum_{k=0}^{m} c_{ik} x^k$$

- Coefficients chosen so $p_i(x_i)=f_i$ and from smoothness condition: all derivatives (l) match at the endpoints

$$p_i^{(l)}(x_{i+1}) = p_{i+1}^{(l)}(x_{i+1}), \quad l = 0, 1, \ldots, m-1$$

- Except for points on the boundary of the curve

Splines: Determining the coefficients

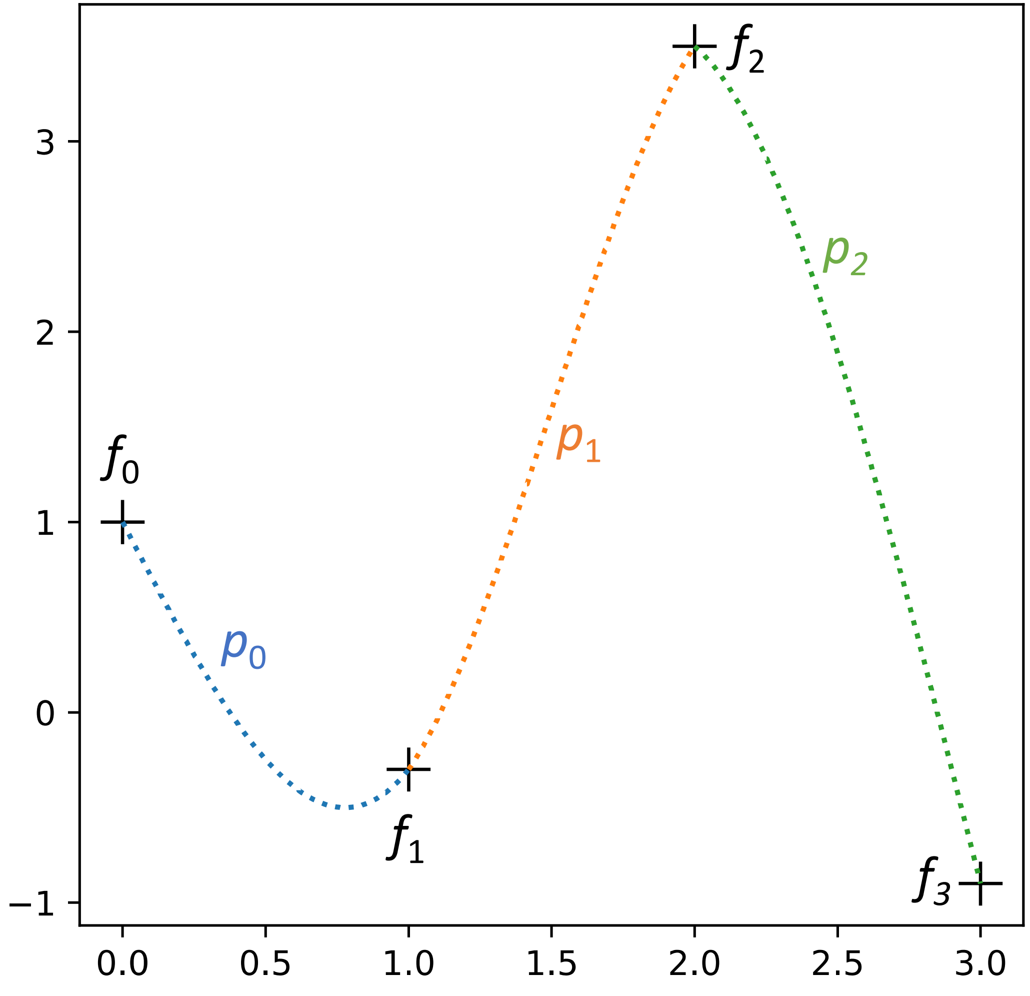
- There are n intervals; in each interval: m+1 coefficients for the polynomial
- Total: $(m+1)n$ coefficients:
  - Smoothness condition on interior points: $(m)(n-1)$ equations
  - Curve passing through interior points: $(n-1)$ equations
  - Remaining m+1 equations from imposing conditions on derivatives at end points
    - Natural spline: Setting highest-order derivative to zero at both endpoints

Most popular: Cubic splines, $m = 3$

- Easy to implement
- Produce a curve that appears to be seamless
- Avoids distortions near the edges
- Only piecewise continuous, third derivatives are discontinuous

Cubic spline example: 3 intervals

- Order: m=3, intervals: n=3, points: x = 0, 1, 2, 3
- Constraints: $(m+1)n = 12$
  - Interior point 1: $$p_0(x_1) = f_1$$ $$p_1(x_1) = f_1$$ $$p_0'(x_1) = p_1'(x_1)$$ $$p_0''(x_1) = p_1''(x_1)$$
  - Interior point 2: $$p_1(x_2) = f_2$$ $$p_2(x_2) = f_2$$ $$p_1'(x_2) = p_2'(x_2)$$ $$p_1''(x_2) = p_2''(x_2)$$



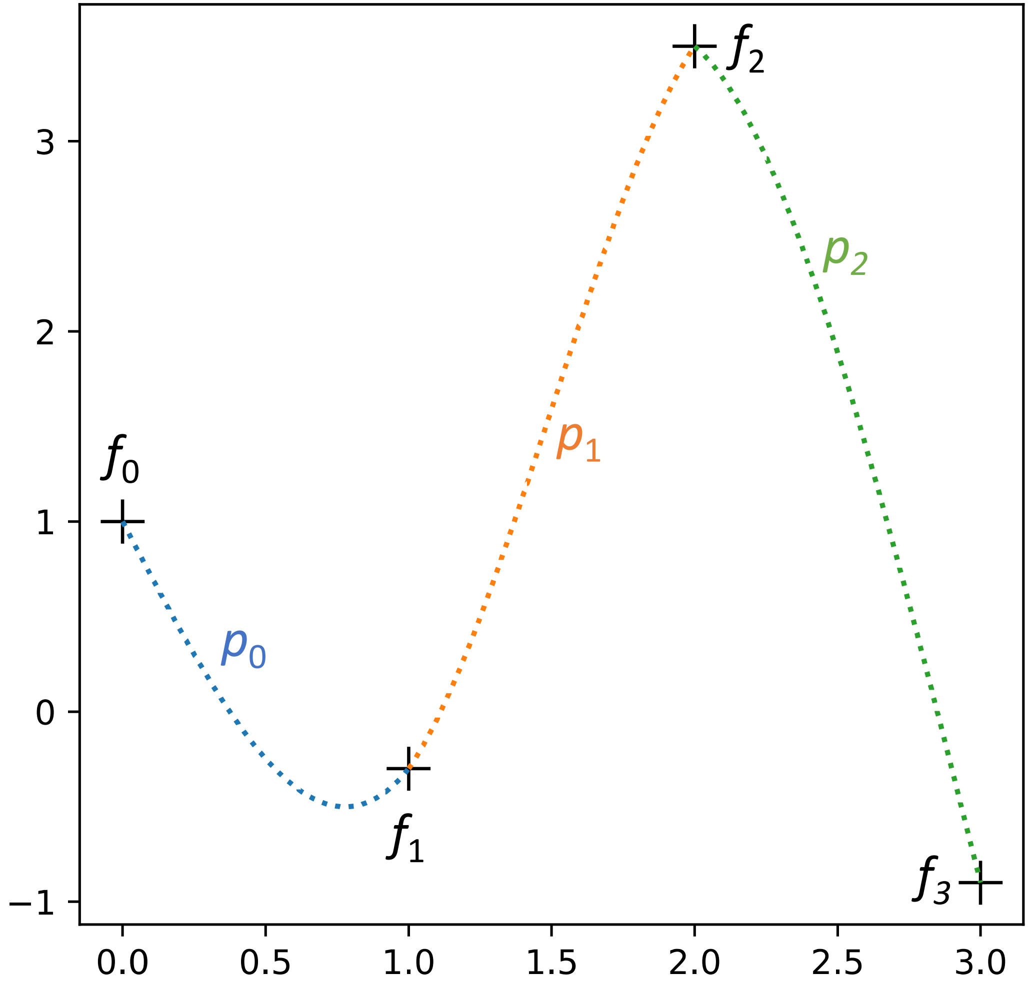
Cubic spline example: 3 intervals

- At the boundaries:

$$p_0(x_0) = f_0$$
$$p_2(x_3) = f_3$$

- Natural spline, second derivatives at the boundary set to zero

$$p_0''(x_0) = 0$$
$$p_2''(x_3) = 0$$



Now solve for the coefficients:

- Linearly interpolate the second derivative:

$$p_i''(x) = \frac{1}{\Delta x}\left[(x - x_i)p_{i+1}'' - (x - x_{i+1})p_i''\right]$$

- Integrate twice:

$$p_i(x) = \frac{1}{6\Delta x}\left\{p_{i+1}''\left[(x - x_i)^3 + 6A(x - x_i)\right] - p_i''\left[(x - x_{i+1})^3 + 6B(x - x_{i+1})\right]\right\}$$

- Impose constraints: $p_i(x_i) = f_i, \; p_i(x_{i+1}) = f_{i+1}$

Now solve for the coefficients:

$$p_i(x) = \alpha_i(x - x_i)^3 + \beta_i(x - x_{i+1})^3 + \gamma_i(x - x_i) + \eta_i(x - x_{i+1})$$

- Results:

$$\alpha_i = \frac{p''_{i+1}}{6\Delta x}, \quad \beta_i = -\frac{p''_i}{6\Delta x}, \quad \gamma_i = \frac{-p''_{i+1}\Delta x^2 + 6f_{i+1}}{6\Delta x}, \quad \eta_i = \frac{p''_i\Delta x^2 - 6f_i}{6\Delta x}$$

- For now, in terms of second derivative

- To get second derivative, use continuity condition

$$p'_{i-1}(x_i) = p'_i(x_i)$$

Now solve for the coefficients:

$$p''_{i-1}\Delta x + 4p''_i\Delta x + p''_{i+1}\Delta x = \frac{6}{\Delta x}(f_{i-1} - 2f_i + f_{i+1})$$

- Applies to all interior points
- Natural boundary conditions:

$$p''_0 = 0,\ p''_n = 0$$

- Results in a system of linear equations

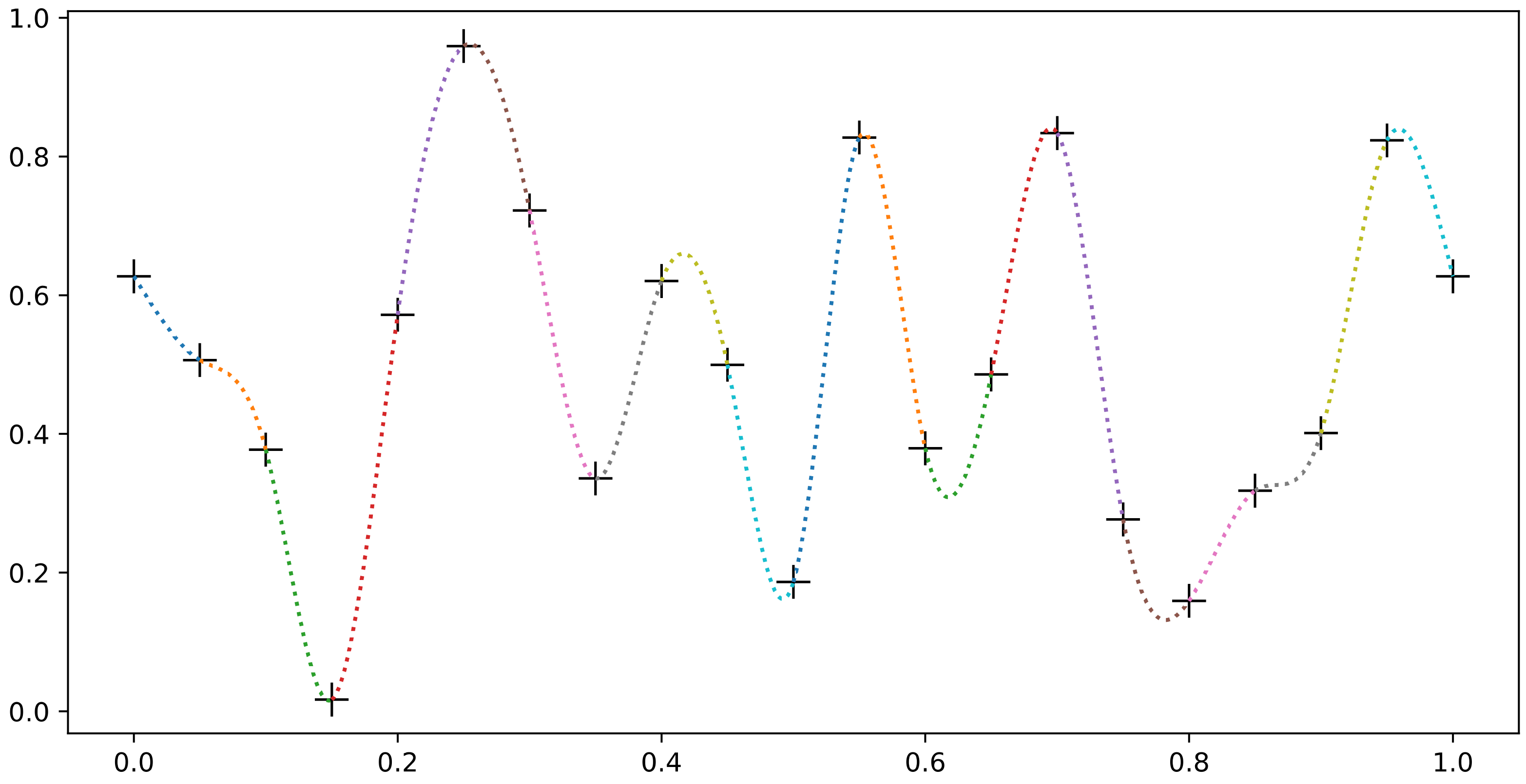
Results in system of linear equations

- Can be written as a tridiagonal matrix:

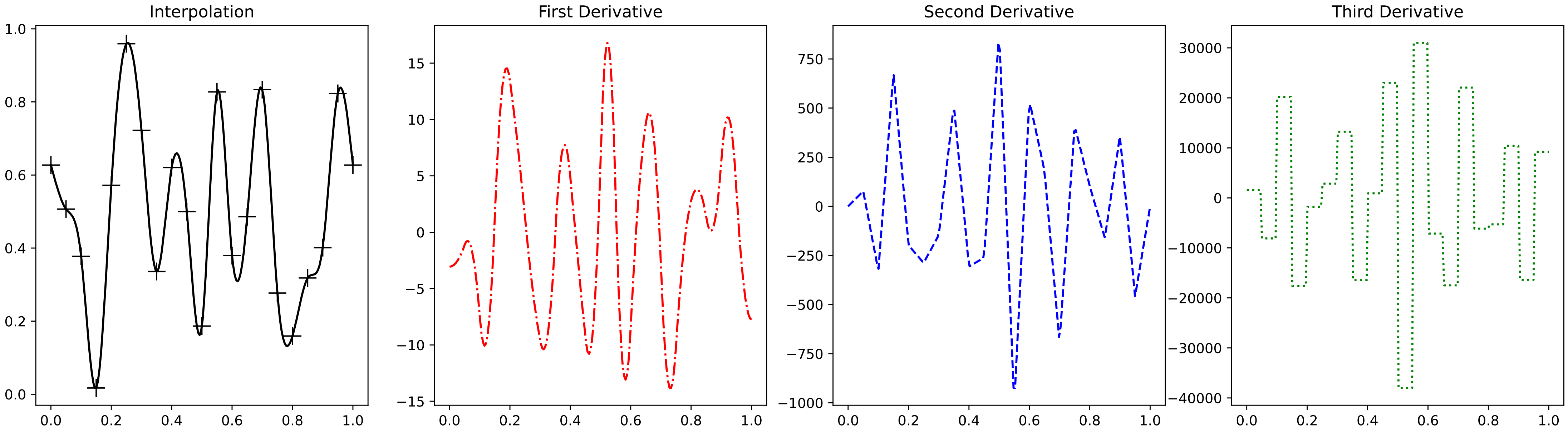
$$\begin{pmatrix} 4\Delta x & \Delta x & & & & \\ \Delta x & 4\Delta x & \Delta x & & & \\ & \Delta x & 4\Delta x & \Delta x & & \\ & & \ddots & \ddots & \ddots & \\ & & & \Delta x & 4\Delta x & \Delta x \\ & & & & \Delta x & 4\Delta x \end{pmatrix} \begin{pmatrix} p''_1 \\ p''_2 \\ p''_3 \\ \vdots \\ p''_{n-2} \\ p''_{n-1} \end{pmatrix} = \frac{6}{\Delta x} \begin{pmatrix} f_0 - 2f_1 + f_2 \\ f_1 - 2f_2 + f_3 \\ f_2 - 2f_3 + f_4 \\ \vdots \\ f_{n-3} - 2f_{n-2} + f_{n-1} \\ f_{n-2} - 2f_{n-1} + f_n \end{pmatrix}$$

- We will discuss linear algebra in a later class

Example: Cubic spline for random numbers



Example: Derivatives of cubic splines



After class tasks

- Readings:
  - Pang Section 2.1 and 3.3
  - Wikipedia article on Chebyshev nodes
  - Myths about polynomial interpolation