CS137 – Introduction to Scientific Computing

Lecture 13 – More on Interpolation

Felix Kwok

Stanford University

High-Degree Polynomial Interpolation

Last time we showed that, for Lagrangian interpolation,

$$f(x) = p_n(x) + \frac{\Delta_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi),$$

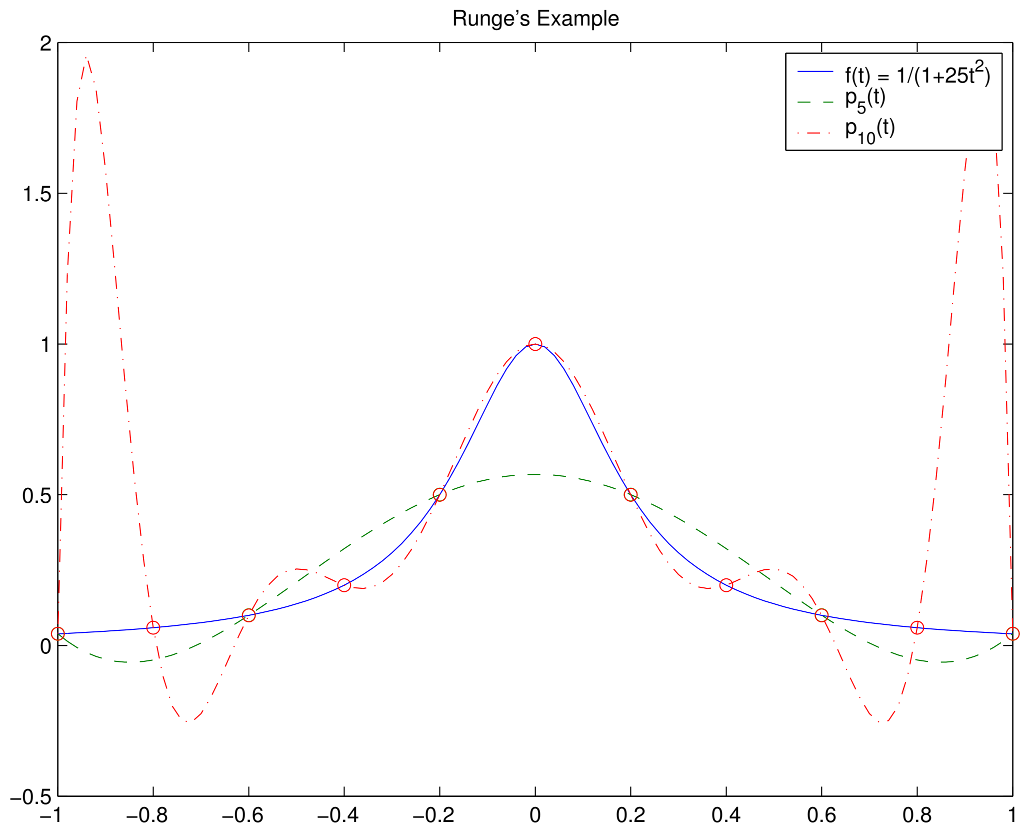
where $\Delta_n(x) = \prod_{i=0}^{n+1}(x - x_i)$. This does *not* mean that $p_n(x) \to f(x)$ as $n \to \infty$ because

1. $f^{(n+1)}(\xi)$ may not be nicely bounded,
2. $\Delta_{n+1}(x)$ can grow; in particular, the product can be large near end points, since $(x - x_i)$ is large for most i

Runge's Example

$$f(x) = \frac{1}{1+25x^2}, \qquad x \in [-1, 1]$$

Suppose we pick equally spaced nodes.



Runge's Example

- Highly oscillatory (Typical for high-degree polynomials)
- Non-convergence near the end-points

Possible solutions:

1. Use more nodes near the end-points
   $\Longrightarrow$ Chebyshev Polynomials
2. Divide into subintervals, and use a different low-degree polynomial for each subinterval
   $\Longrightarrow$ Splines

Chebyshev Polynomials

$$T_n(x) = \begin{cases} \cos(n \cos^{-1}(x)), & |x| \leq 1 \\ (\operatorname{sgn}(x))^n \cosh(n \cosh^{-1}(|x|)), & |x| \geq 1 \end{cases}$$

Examples:

- $T_0(x) = 1$
- $T_1(x) = x$
- $T_2(x) = \cos(2\cos^{-1} x) = 2\cos^2(\cos^{-1} x) - 1 = 2x^2 - 1$
- $T_3(x) = \cos(3\cos^{-1} x) = 4\cos^3(\cos^{-1} x) - 3\cos(\cos^{-1} x) = 4x^3 - 3x$

Chebyshev Polynomials

In general,

$$\cos(A) + \cos(B) = 2\cos\left(\frac{A+B}{2}\right)\cos\left(\frac{A-B}{2}\right)$$

so

$$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos n\theta\cos\theta$$

implies

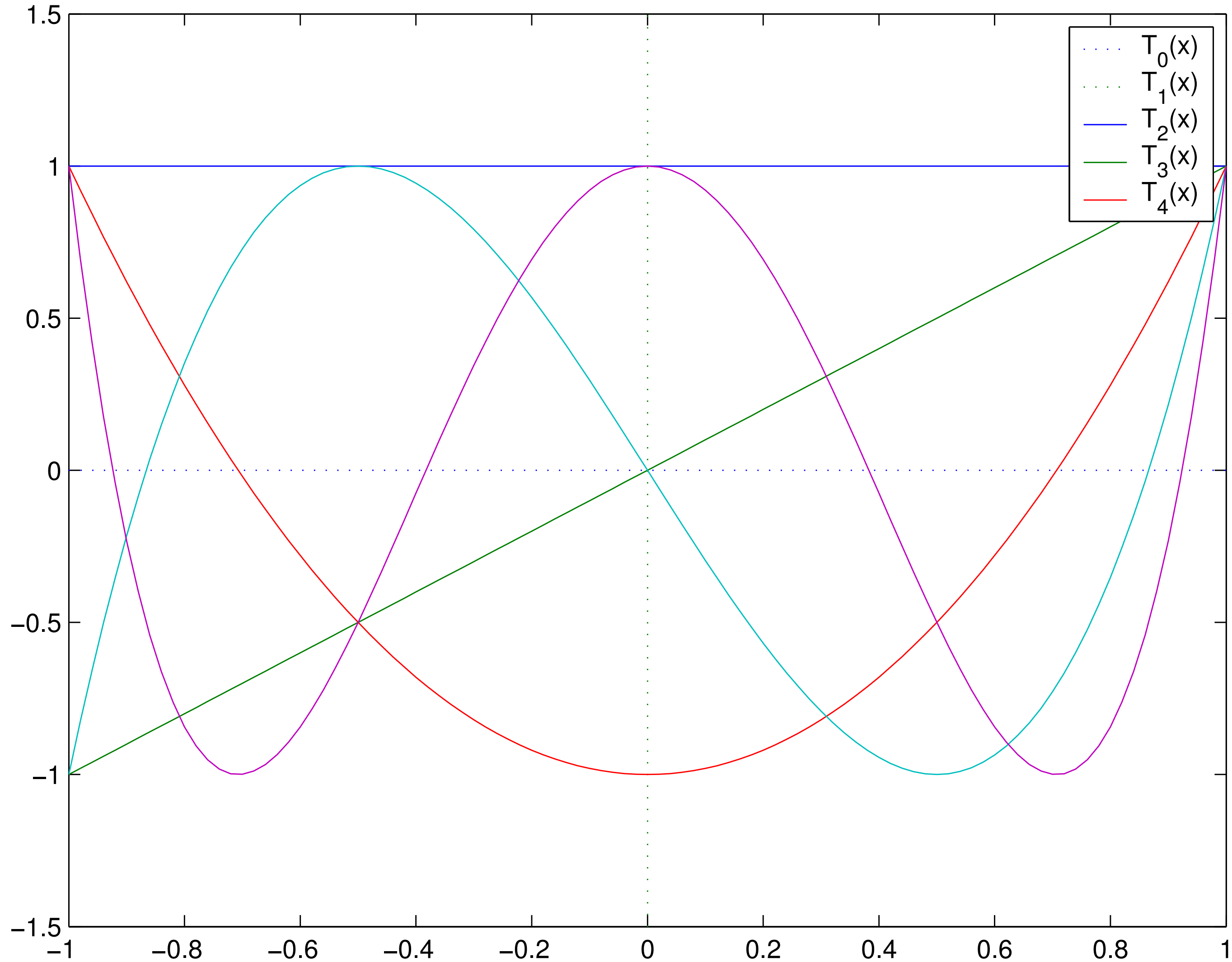
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Note: Leading coefficient of $T_n(x)$ is 2^{n-1}.

Properties of $T_n(x)$

1. $|T_n(x)| \leq 1$ for $|x| \leq 1$
2. Maximum modulus attained at $t_j = cos(j\pi/n)$, $j = 0, \ldots, n$:
$$T_n(t_j) = (-1)^j.$$
3. $T_n(x)$ is a degree-n polynomial $\implies$ n roots:
$$\cos(n\theta) = 0 \implies \theta_j = \frac{(2j-1)\pi}{2n}, \; j = 1, \ldots, n$$
So $T_n(x_j) = 0$, $x_j = \cos\left(\frac{(2j-1)\pi}{2n}\right)$, i.e. all n roots lie within $[-1, 1]$.
4. All roots are distinct $\implies$ alternating signs.

Chebyshev Polynomials



Unequally Spaced Nodes for Interpolation

Recall

$$f(x) = p_{n-1}(x) + \frac{\Delta_n(x)}{n!} f^{(n)}(\xi).$$

Suppose we are allowed to evaluate $f(x)$ at n different points within the interval $[-1, 1]$ for the purpose of interpolation, and we want to pick the nodes to minimize $\Delta_n(x)$.

Answer: Choose x_j to be the zeros of $T_n(x)$! Then

$$\hat{\Delta}_n(x) = 2^{-n+1} T_n(x)$$

(since Δ_n is monic).

Optimality of $T_n(x)$

Claim: Let $\Gamma_n(x) = x^n + \cdots$ (monic polynomial). Then

$$\max_{-1\le x\le 1} |\Gamma_n(x)| \ge \max_{-1\le x\le 1} |\hat{\Delta}_n(x)|.$$

Proof: Suppose, on the contrary, that

$$\max_{-1\le x\le 1} |\Gamma_n(x)| < \max_{-1\le x\le 1} |\hat{\Delta}_n(x)|.$$

Define $D(x) = \hat{\Delta}_n(x) - \Gamma_n(x)$. Then for $t_j = \cos(j\pi/n)$, $j = 0, \ldots, n$, we have

$$|\Gamma_n(t_j)| < \max_{-1\le x\le 1} |\hat{\Delta}_n(x)| = |\hat{\Delta}_n(t_j)|.$$

Optimality of $T_n(x)$

Thus, $D(t_j)$ has the same sign as $\hat{\Delta}_n(t_j)$, i.e.

$$D(t_j) \begin{cases} > 0 & \text{for } j \text{ even,} \\ < 0 & \text{for } j \text{ odd,} \end{cases}$$

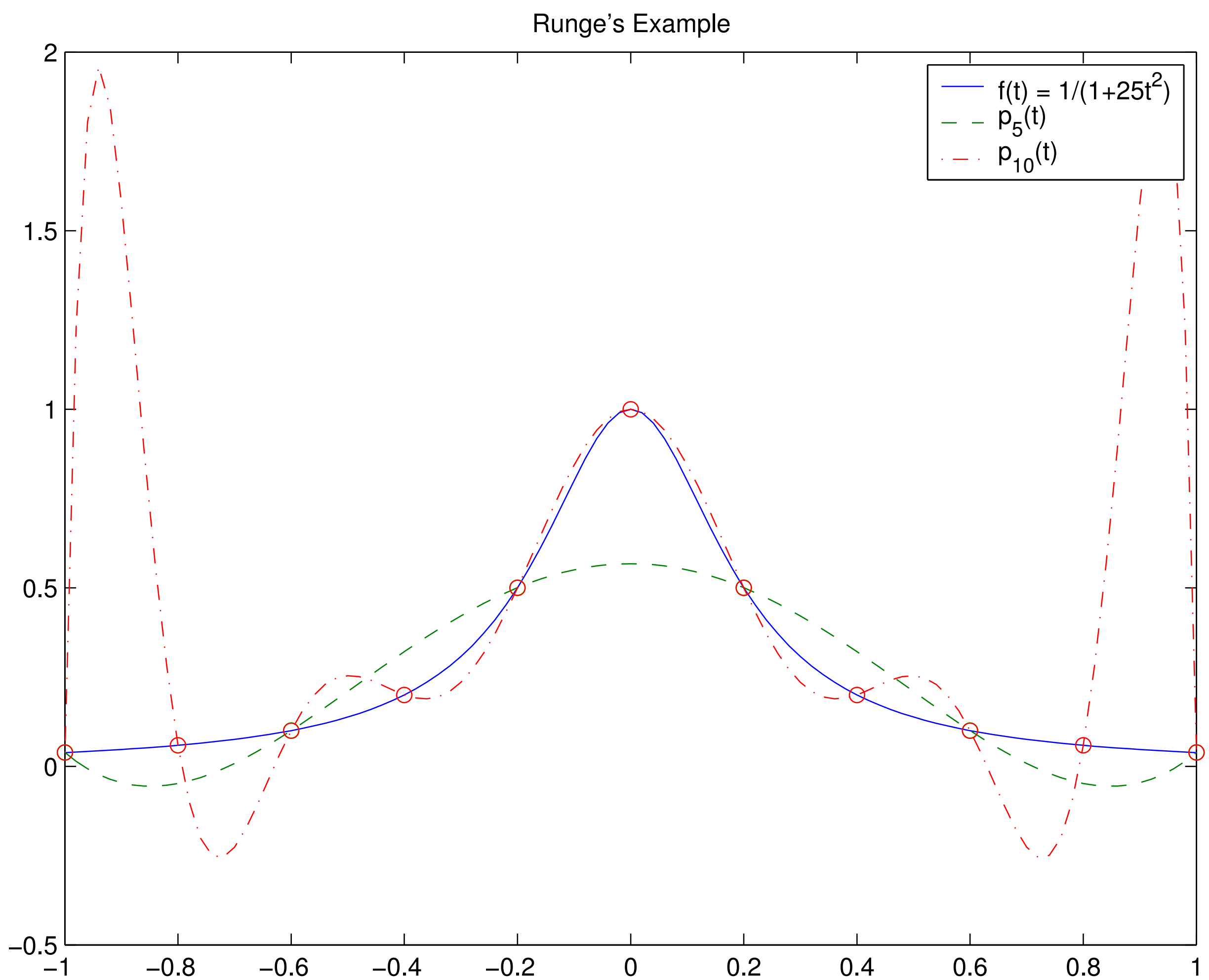
$j = 0, \ldots, n$. There are n sign changes between $t_0 = 1$ and $t_n = -1$, so $D(x)$ has at least n zeros. But since $\Gamma_n(x)$ and $\hat{\Delta}_n(x)$ are both monic, we have

$$\begin{aligned} D(x) &= \hat{\Delta}_n(x) - \Gamma_n(x) \\ &= (x^n + \cdots) - (x^n + \cdots) = cx^{n-1} + \cdots \end{aligned}$$

has degree at most $n - 1 \implies$ Contradiction!

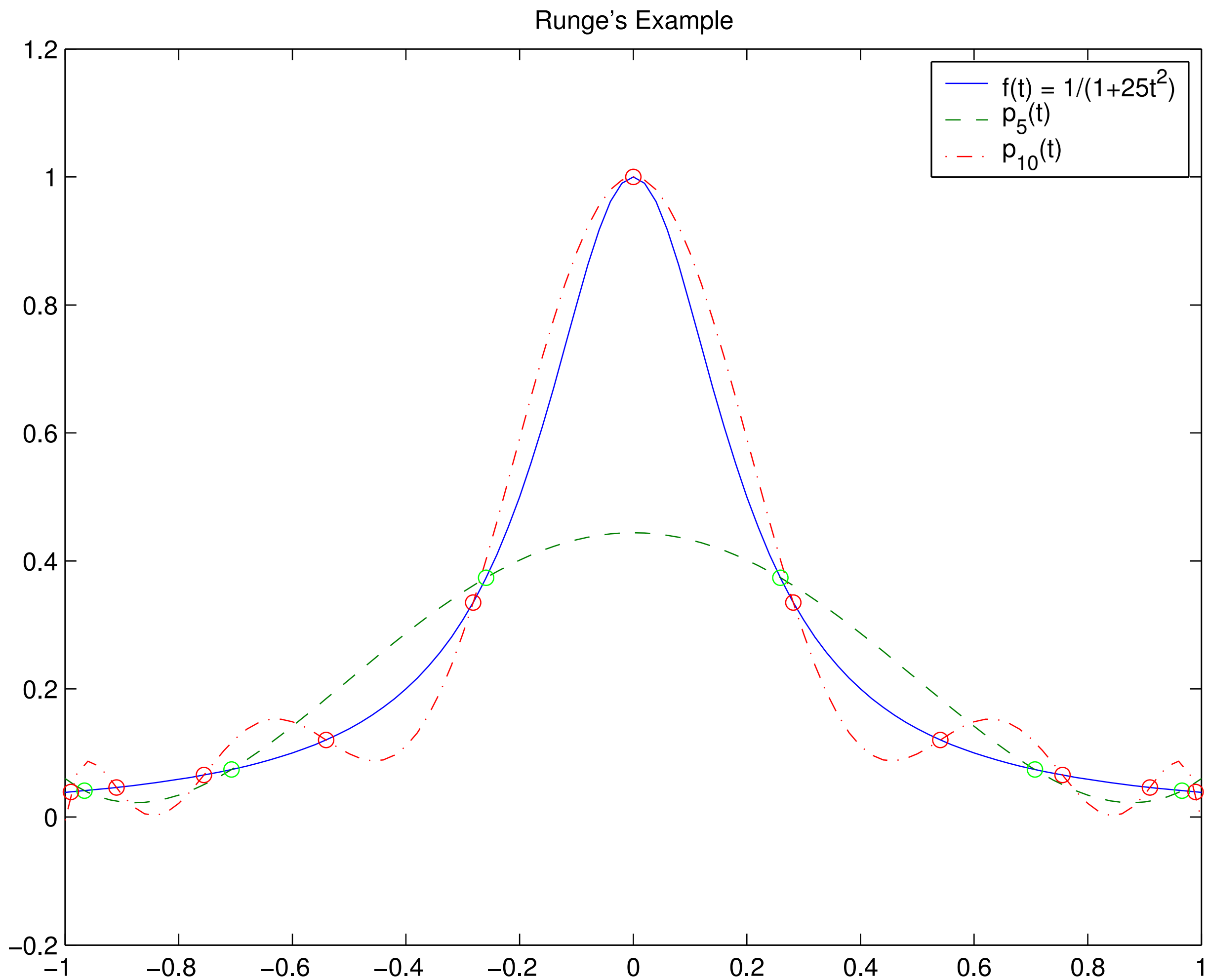
Runge's Example Revisited

- Evenly spaced nodes



Runge's Example Revisited

- Chebyshev nodes

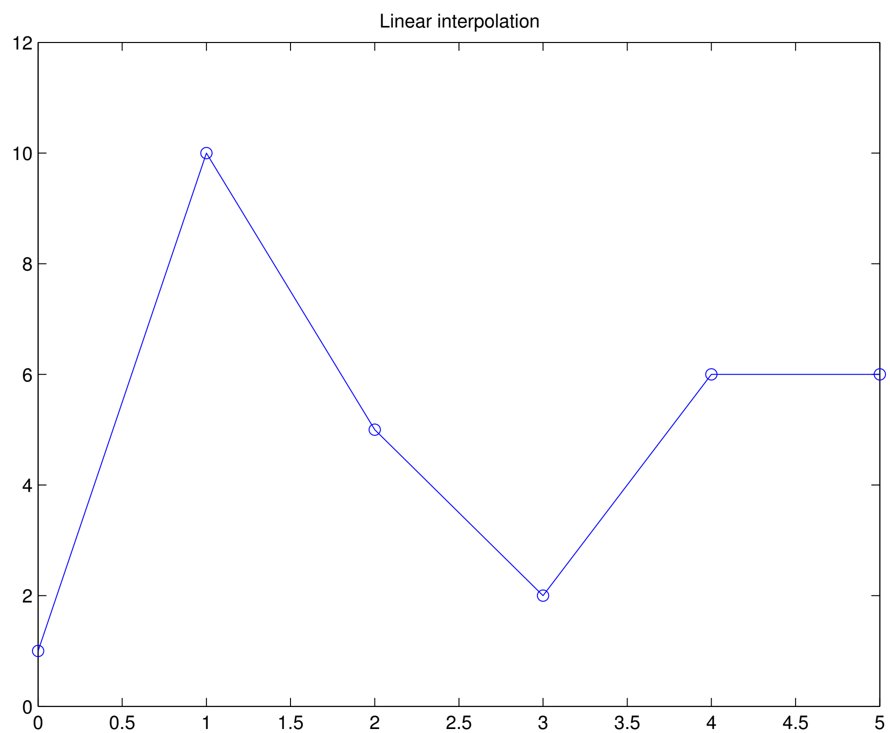


Remarks on Chebyshev Interpolation

1. Can be proved to converge for sufficiently smooth underlying functions
2. For intervals other than $[-1, 1]$, use a change of variables:
   $$x = \frac{(a+b) + (b-a)s}{2},$$
   This brings $x \in [a, b]$ to $s \in [-1, 1]$.
3. High-degree interpolating polynomials still contain "wiggles", may be unphysical.

Piecewise polynomial interpolation

- Basic Idea: Instead of fitting all data with the same polynomial, use different polynomials for each interval $I_j = [x_{j-1}, x_j]$.
- Example: piecewise linear



Quadratic Splines

- For high-order piecewise polynomials, require continuity of derivatives
- Example: piecewise quadratics. Given $x_0 < \cdots < x_n$ and $y_0, \ldots, y_n$, we require

$$p_j(x) = a_j + b_j(x - x_{j-1}) + c_j(x - x_{j-1})^2$$

$$p_j(x_{j-1}) = y_{j-1}$$
$$p_j(x_j) = y_j$$
$$p_j'(x_j) = p_{j+1}'(x_j)$$

- Number of unknowns: $3n$
- Number of constraints: $n + n + (n-1) = 3n - 1$
- Prescribe intial/final condition: $p_1'(x_0) = y_0'$ or $p_n'(x_n) = y_n'$

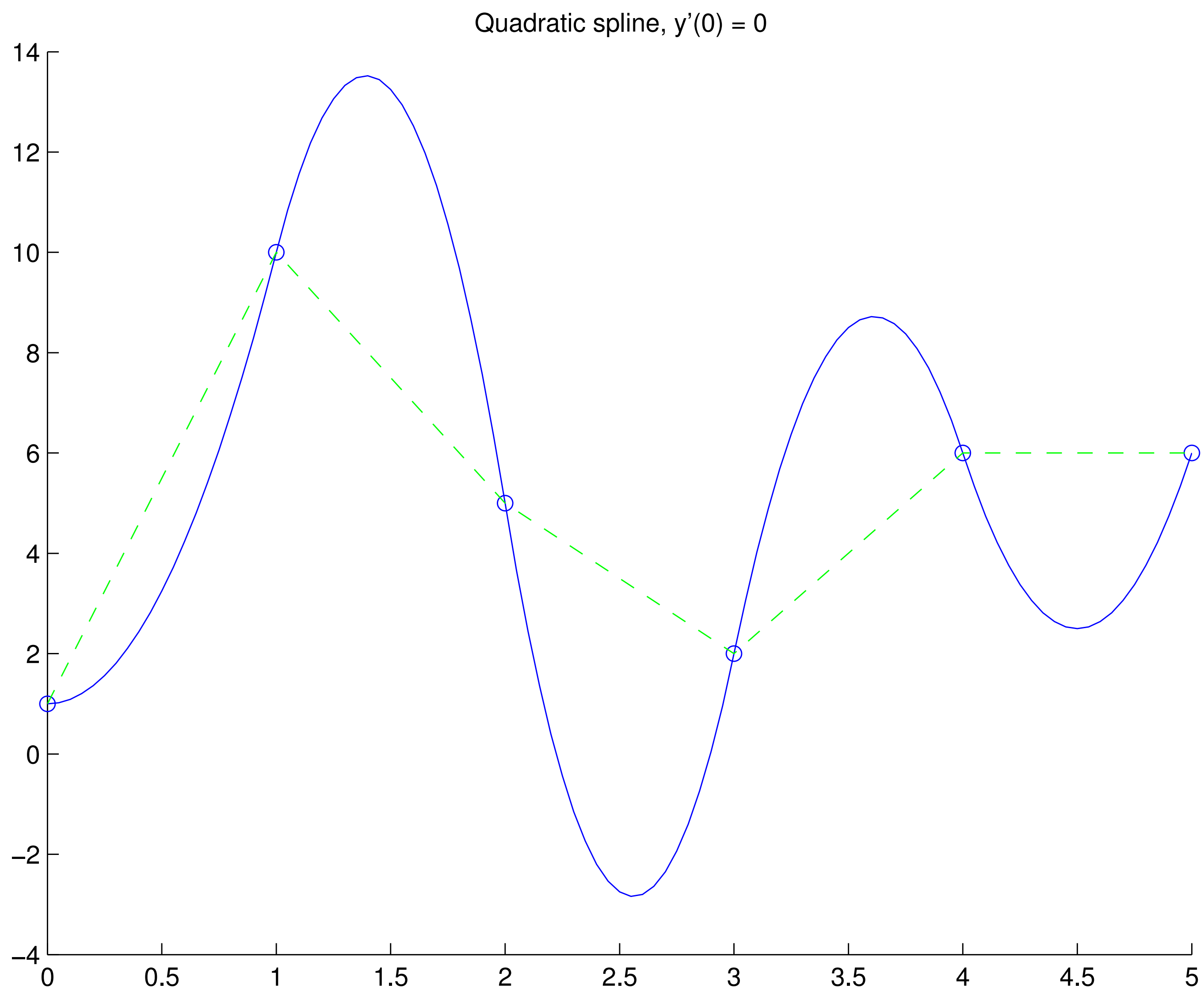
Quadratic Splines

- Linear system: assuming $h_j = x_j - x_{j-1}$,

$$
\begin{aligned}
p_j(x_{j-1}) &= a_j = y_{j-1} \\
p_j(x_j) &= a_j + b_j h_j + c_j h_j^2 = y_j \\
&\implies b_j h_j + c_j h_j^2 = y_j - y_{j-1} \\
p_j'(x_j) &= b_j + 2c_j h_j \\
&= b_{j+1} = p_{j+1}'(x_j) \\
p_1'(x_0) &= y_0'
\end{aligned}
$$

- Sparse matrix of size $2n-1$
- Interpolant is continuously differentiable
- Discontinuous second derivatives

Quadratic Splines



Cubic Splines

- Formulation:

$$p_j(x) = a_j + b_j(x - x_{j-1}) + c_j(x - x_{j-1})^2 + d_j(x - x_{j-1})^3$$

$$p_j(x_{j-1}) = y_{j-1}$$

$$p_j(x_j) = y_j$$

$$p'_j(x_j) = p'_{j+1}(x_j)$$

$$p''_j(x_j) = p''_{j+1}(x_j)$$

- Twice continuously differentiable
- $4n$ unknowns, $4n - 2$ constraints

Natural Cubic Splines

- Set $p_1''(x_0) = p_n''(x_n) = 0$
- The natural cubic spline has “minimum curvature, i.e. it minimizes

$$\int_{x_0}^{x_n} |S''(x)|^2 dx,$$

  over all cubic splines $S(x)$.
- Can set up linear system the same way as in the quadratic spline, but we can do better; the trick is to find the right basis.

Natural Cubic Splines

$$p_j(x) = a_j + b_j(x - x_{j-1}) + c_j(x - x_{j-1})^2 + d_j(x - x_{j-1})^3$$

Suppose we know the nodal curvature $M_j := p_j''(x_j)$ as well as the nodal values y_j. Then we can write

$$\begin{aligned} y_{j-1} &= a_j \\ y_j &= a_j + b_j h_j + c_i h_j^2 + d_j h_j^3 \\ M_{j-1} &= 2c_j \\ M_j &= 2c_j + 6d_j h_j \end{aligned}$$

Natural Cubic Splines

We can solve for the coefficients easily:

$$a_j = y_{j-1}$$

$$b_j = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(2M_{j-1} + M_j)$$

$$c_j = \frac{1}{2}M_{j-1}$$

$$d_j = \frac{1}{6h_j}(M_j - M_{j-1})$$

To solve for the M_j, enforce continuity condition

$$p_j'(x_j) = p_{j+1}'(x_j)$$

Natural Cubic Splines

$$
\begin{aligned}
p_j'(x_j) &= b_j + 2c_jh_j + 3d_jh_j^2 \\
&= \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(2M_{j-1} + M_j) + M_{j-1}h_j + \frac{h_j}{2}(M_j - M_j \\
&= \frac{h_j}{6}M_{j-1} + \frac{h_j}{3}M_j + \frac{1}{h_j}(y_j - y_{j-1})
\end{aligned}
$$

$$
p_{j+1}'(x_j) = b_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6}(2M_j + M_{j+1})
$$

Rearrange and get

$$
\frac{h_j}{6}M_{j-1} + \frac{h_j + h_{j+1}}{3}M_j + \frac{h_{j+1}}{6}M_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j}.
$$

Natural Cubic Splines

$$\frac{h_j}{6}M_{j-1} + \frac{h_j + h_{j+1}}{3}M_j + \frac{h_{j+1}}{6}M_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j}.$$

- Only M_{j-1}, M_j, M_{j+1} involved in equation $\Longrightarrow$ tridiagonal
- Symmetric, diagonally dominant $\Longrightarrow$ positive definite
- Use banded Cholesky $\Longrightarrow$ $O(n)$ solve

Natural Cubic Splines

$$\begin{bmatrix} \alpha_2 & \beta_2 & & & & \\ \beta_2 & \alpha_3 & \beta_3 & & & \\ & \beta_3 & \ddots & \ddots & & \\ & & \ddots & \ddots & \beta_{n-2} \\ & & & \beta_{n-2} & \alpha_{n-1} \end{bmatrix} \begin{pmatrix} M_2 \\ M_3 \\ \vdots \\ \vdots \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} \delta_2 \\ \delta_3 \\ \vdots \\ \vdots \\ \delta_{n-1} \end{pmatrix}$$

where

$$\alpha_j = \frac{h_j + h_{j+1}}{3}, \quad \beta_j = \frac{h_{j+1}}{6}, \quad \delta_j = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j}.$$

Natural Cubic Splines