Computational Astrophysics

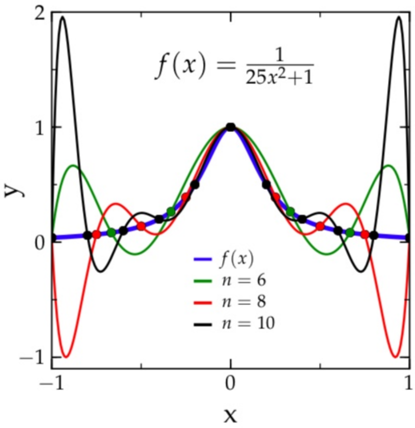
E. Larrañaga

Observatorio Astronómico Nacional
Universidad Nacional de Colombia

April 7, 2019

Outline

Lagrange Interpolation



Convergence of the Interpolated Function

Data describe a continuous function $f(x)$ on an interval $[a, b]$.

Convergence of the Interpolated Function

Data describe a continuous function $f(x)$ on an interval $[a, b]$. We expect that interpolating polynomials $p_n(x)$ with increasing degree n make the interpolation error converges to zero, i.e.

$$\lim_{n\to\infty}\left[\max_{x\in[a,b]}|f(x)-p_n(x)|\right]=0\,. \tag{1}$$

Convergence of the Interpolated Function

Data describe a continuous function $f(x)$ on an interval $[a, b]$. We expect that interpolating polynomials $p_n(x)$ with increasing degree n make the interpolation error converges to zero, i.e.

$$\lim_{n\to\infty}\left[\max_{x\in[a,b]}|f(x)-p_n(x)|\right]=0\,. \tag{1}$$

However, for most continuous functions, this is not the case !

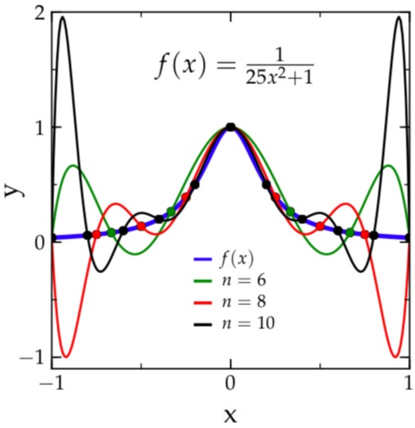
Convergence of the Interpolated Function

Data describe a continuous function $f(x)$ on an interval $[a, b]$. We expect that interpolating polynomials $p_n(x)$ with increasing degree n make the interpolation error converges to zero, i.e.

$$\lim_{n\to\infty}\left[\max_{x\in[a,b]}|f(x)-p_n(x)|\right]=0\,. \tag{1}$$

However, for most continuous functions, this is not the case ! This is called *Runge's Phenomenon*.

Lagrange Interpolation



Convergence of the Interpolated Function

While the interpolating polynomials $p_n(s)$ converge to $f(x)$ near the peak, Runge's phenomenon produces strong oscillations and divergences at the edges of the interval.

Convergence of the Interpolated Function

While the interpolating polynomials $p_n(s)$ converge to $f(x)$ near the peak, Runge's phenomenon produces strong oscillations and divergences at the edges of the interval.

A higher polynomial degree could lead to a greater error and unwanted oscillations!

Piecewise-polynomial Interpolation

High degree polynomial interpolation should only be used if one is certain that the function to be interpolated can be approximated well globally by a single polynomial.

Piecewise-polynomial Interpolation

High degree polynomial interpolation should only be used if one is certain that the function to be interpolated can be approximated well globally by a single polynomial.

If not, one can use *piecewise-polynomial* interpolation. This uses many low-degree polynomials instead of a single global polynomial.

Piecewise-polynomial Interpolation

High degree polynomial interpolation should only be used if one is certain that the function to be interpolated can be approximated well globally by a single polynomial.

If not, one can use *piecewise-polynomial* interpolation. This uses many low-degree polynomials instead of a single global polynomial.

The simplest form of piecewise polynomial interpolation is piecewise linear interpolation (connecting data points with straight lines).

Hermite Interpolation

Hermite interpolation is a form of polynomial interpolation in which it is used data points as well as derivatives of the data. It reduces unwanted oscillations (specially if it is applied in piecewise form).

Hermite Interpolation

Hermite interpolation is a form of polynomial interpolation in which it is used data points as well as derivatives of the data. It reduces unwanted oscillations (specially if it is applied in piecewise form).

Now we will consider only the simplest case of Hermite interpolation: a function and its first derivative.

Hermite Interpolation

Hermite interpolation is a form of polynomial interpolation in which it is used data points as well as derivatives of the data. It reduces unwanted oscillations (specially if it is applied in piecewise form).

Now we will consider only the simplest case of Hermite interpolation: a function and its first derivative.
Consider $n+1$ data points. We will find a polynomial satisfying the conditions

$$p(x_i) = c_{i0}\ , \quad p'(x_i) = c_{i1} \quad \text{for } i \in [0, n]\ , \tag{2}$$

where $c_{i0} = f(x_i)$ and $c_{i1} = f'(x_i)$.

Hermite Interpolation

We propose the form

$$p(x) = \sum_{i=0}^{n} c_{i0} A_i(x) + \sum_{i=0}^{n} c_{i1} B_i(x) \ , \tag{3}$$

where $A_i(x)$ and $B_i(x)$ are polynomials satisfying:

$$\begin{aligned} A_i(x_j) &= \delta_{ij} \ , \quad & B_i(x_j) &= 0 \ , \\ A_i'(x_j) &= 0 \ , \quad & B_i'(x_j) &= \delta_{ij} \ . \end{aligned} \tag{4}$$

Hermite Interpolation

A possible form of the polynomials $A_i(x)$ and $B_i(x)$ is obtained using the Lagrange interpolation terms

$$L_{nj}(x) = \prod_{\substack{j=0 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k} \ , \tag{5}$$

as

$$\begin{aligned} A_i(x) &= [1 - 2(x - x_i)L'_{ni}(x_i)]L^2_{ni}(x) \ , \\ B_i(x) &= (x - x_i)L^2_{ni}(x) \ . \end{aligned} \tag{6}$$

Hermite Interpolation

A possible form of the polynomials $A_i(x)$ and $B_i(x)$ is obtained using the Lagrange interpolation terms

$$L_{nj}(x) = \prod_{\substack{j=0 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k} \ , \tag{5}$$

as

$$\begin{aligned} A_i(x) \ &= [1 - 2(x - x_i)L'_{ni}(x_i)]L^2_{ni}(x) \ , \\ B_i(x) \ &= (x - x_i)L^2_{ni}(x) \ . \end{aligned} \tag{6}$$

Each L_{ni} is of degree n and therefore A_i and B_i are of degree $2n + 1$ (The same as the maximal degree of $p(x)$).

(see proof in `http://www.math.umd.edu/~dlevy/books/na.pdf`.)

Piecewise Cubic Hermite Interpolation

Consider a dataset such that for each point x_i we also have the point x_{i+1}, as well as the values $f(x_i)$, $f(x_{i+1})$, $f'(x_i)$, and $f'(x_{i+1})$ (These values are known or can be evaluated numerically).

Piecewise Cubic Hermite Interpolation

Consider a dataset such that for each point x_i we also have the point x_{i+1}, as well as the values $f(x_i)$, $f(x_{i+1})$, $f'(x_i)$, and $f'(x_{i+1})$ (These values are known or can be evaluated numerically).

Taking $n = 1$ in the Hermite interpolation equations we obtain A_i and B_i as polynomials of degree $2n + 1 = 3$.

Piecewise Cubic Hermite Interpolation

The cubic Hermite polynomial that interpolates $f(x)$ in $[x_i, x_{i+1}]$ is given by

$$\begin{aligned} H_3(x) =& f(x_i)\psi_0(z) + f(x_{i+1})\psi_0(1-z) \\ &+ f'(x_i)(x_{i+1} - x_i)\psi_1(z) \\ &- f'(x_{i+1})(x_{i+1} - x_i)\psi_1(1-z) \ , \end{aligned}$$

where

$$\psi_0(z) = 2z^3 - 3z^2 + 1 \ , \quad (7)$$

$$\psi_1(z) = z^3 - 2z^2 + z \ . \quad (8)$$

and

$$z = \frac{x - x_i}{x_{i+1} - x_i} \ , \quad (9)$$

Spline Interpolation

Splines are the ultimate method for piecewise polynomial interpolation. It achieves smoothness by requiring continuity at data points not only for the function values $f(x_i)$, but also for a number of its derivatives $f^{(\ell)}(x_i)$.

Spline Interpolation

Splines are the ultimate method for piecewise polynomial interpolation. It achieves smoothness by requiring continuity at data points not only for the function values $f(x_i)$, but also for a number of its derivatives $f^{(\ell)}(x_i)$.

Assuming that we know the values $f(x_i)$ at points x_i $(i \in [0, n])$, we construct piecewise polynomials of degree m on each of the n segments $[x_i, x_{i+1}]$,

$$p_i(x) = \sum_{k=0}^{m} c_{ik} x^k \ , \tag{10}$$

to approximate $f(x)$ for $x \in [x_i, x_{i+1}]$.

Spline Interpolation

m is the degree of the spline.
There are $m + 1$ coefficients c_{ik} for each i.
There are n intervals.

Spline Interpolation

m is the degree of the spline.
There are $m+1$ coefficients c_{ik} for each i.
There are n intervals.

Hence, there are $n(m+1)$ coefficients c_{ik} that are determined by the conditions:

1. requiring $(n-1)(m-1)$ smoothness conditions at non-boundary points: $p_i^{\ell}(x_{i+1}) = p_{i+1}^{\ell}(x_{i+1})$ for $\ell = 1, \cdots, m-1$,
2. requiring $2n$ interpolation conditions: $p_i(x_i) = f(x_i) = p_{i+1}(x_i)$,
3. choosing the remaining $m-1$ values of some of the $p_o^{\ell}(x_o)$ and $p_{n-1}^{\ell}(x_n)$ for $\ell = 1, \cdots, m-1$.

Cubic Natural Spline Interpolation

Consider $m = 3$: Cubic Spline or *Natural* Spline.
There are $m + 1 = 4$ coefficients c_{ik} for each i.
There are n intervals.

Cubic Natural Spline Interpolation

Consider $m = 3$: Cubic Spline or *Natural* Spline.
There are $m + 1 = 4$ coefficients c_{ik} for each i.
There are n intervals.

Hence, there are $n(m + 1) = 4n$ coefficients c_{ik} and we need $4n$ conditions to find them and we need the information of the first and second derivatives.

Cubic Natural Spline Interpolation

Consider $m = 3$: Cubic Spline or *Natural* Spline.
There are $m + 1 = 4$ coefficients c_{ik} for each i.
There are n intervals.

Hence, there are $n(m + 1) = 4n$ coefficients c_{ik} and we need $4n$ conditions to find them and we need the information of the first and second derivatives.

The first choice is to set the highest derivative to zero on both ends of the interval:

$$p_0''(x_0) = 0 \ , \ \ p_{n-1}''(x_n) = 0 \ . \tag{11}$$

Cubic Natural Spline Interpolation

Then we consider the linear interpolation of the second derivative in $[x_i, x_{i+1}]$,

$$p_i''(x) = \frac{1}{x_{i+1} - x_i}\left[(x - x_i)p_{i+1}'' - (x - x_{i+1})p_i''\right] \ , \qquad (12)$$

where $p_i'' = p_i''(x_i) = p_{i-1}''(x_i)$ and $p_{i+1}'' = p_{i+1}''(x_{i+1}) = p_i''(x_{i+1})$.

Cubic Natural Spline Interpolation

Integration of this interpolated second derivative two times and using $f_i = p_i(x_i) = f(x_i)$ gives

$$p_i(x) = \alpha_i(x - x_i)^3 + \beta_i(x - x_{i+1})^3 + \gamma_i(x - x_i) + \eta_i(x - x_{i+1}), \quad (13)$$

where

$$\alpha_i = \frac{p''_{i+1}}{6h_i} \qquad \beta_i = \frac{-p''_i}{6h_i}, \quad (14)$$

$$\gamma_i = \frac{f_{i+1}}{h_i} - \frac{h_i p''_{i+1}}{6} \qquad \eta_i = \frac{h_i p''_i}{6} - \frac{f_i}{h_i}, \quad (15)$$

with $h_i = x_{i+1} - x_i$.

Cubic Natural Spline Interpolation

Integration of this interpolated second derivative two times and using $f_i = p_i(x_i) = f(x_i)$ gives

$$p_i(x) = \alpha_i (x - x_i)^3 + \beta_i (x - x_{i+1})^3 + \gamma_i (x - x_i) + \eta_i (x - x_{i+1}) \,, \quad (13)$$

where

$$\alpha_i = \frac{p''_{i+1}}{6h_i} \qquad \beta_i = \frac{-p''_i}{6h_i} \,, \quad (14)$$

$$\gamma_i = \frac{f_{i+1}}{h_i} - \frac{h_i p''_{i+1}}{6} \qquad \eta_i = \frac{h_i p''_i}{6} - \frac{f_i}{h_i} \,, \quad (15)$$

with $h_i = x_{i+1} - x_i$.

Hence, all we need to find the spline is to know all of its second derivatives $p''_i = p''_i(x_i)$.

Cubic Natural Spline Interpolation

Now apply the condition $p'_{i-1}(x_i) = p'_i(x_i)$ to the interpolated polynomial to obtain the equation

$$h_{i-1}p''_{i-1} + 2(h_{i-1} + h_i)p''_i + h_i p''_{i+1} = 6\left(\frac{g_i}{h_i} - \frac{g_{i-1}}{h_{i-1}}\right) \ , \qquad (16)$$

where $g_i = f_{i+1} - f_i$.

Cubic Natural Spline Interpolation

Now apply the condition $p'_{i-1}(x_i) = p'_i(x_i)$ to the interpolated polynomial to obtain the equation

$$h_{i-1}p''_{i-1} + 2(h_{i-1} + h_i)p''_i + h_i p''_{i+1} = 6\left(\frac{g_i}{h_i} - \frac{g_{i-1}}{h_{i-1}}\right) \ , \qquad (16)$$

where $g_i = f_{i+1} - f_i$.

This is a linear system with $n-1$ unknowns: p''_i for $i = 1, \cdots, n-1$ (remember that $p''_0 = p''_n = 0$, as set by the natural spline condition.

Cubic Natural Spline Interpolation

Defining $d_i = 2(h_{i-1} + h_i)$ and $b_i = 6\left(\frac{g_i}{h_i} - \frac{g_{i-1}}{h_{i-1}}\right)$, we can write

$$Ap'' = b \quad \text{with } A_{ij} = \begin{cases} d_i & \text{if } i = j\,, \\ h_i & \text{if } i = j-1\,, \\ h_{i-1} & \text{if } i = j+1\,, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Cubic Natural Spline Interpolation

Defining $d_i = 2(h_{i-1} + h_i)$ and $b_i = 6\left(\frac{g_i}{h_i} - \frac{g_{i-1}}{h_{i-1}}\right)$, we can write

$$Ap'' = b \quad \text{with } A_{ij} = \begin{cases} d_i & \text{if } i = j\,, \\ h_i & \text{if } i = j-1\,, \\ h_{i-1} & \text{if } i = j+1\,, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

The coefficient matrix A_{ij} is real, symmetric, and tri-diagonal. The solution of such a linear system of equations will be presented later in the course.