Improvement on Neville's Algorithm

Recall : Rule #1 for less roundoff errors
→ if possible, store small numbers

Notice that $P_{1234} \simeq P_{123}, P_{234}$

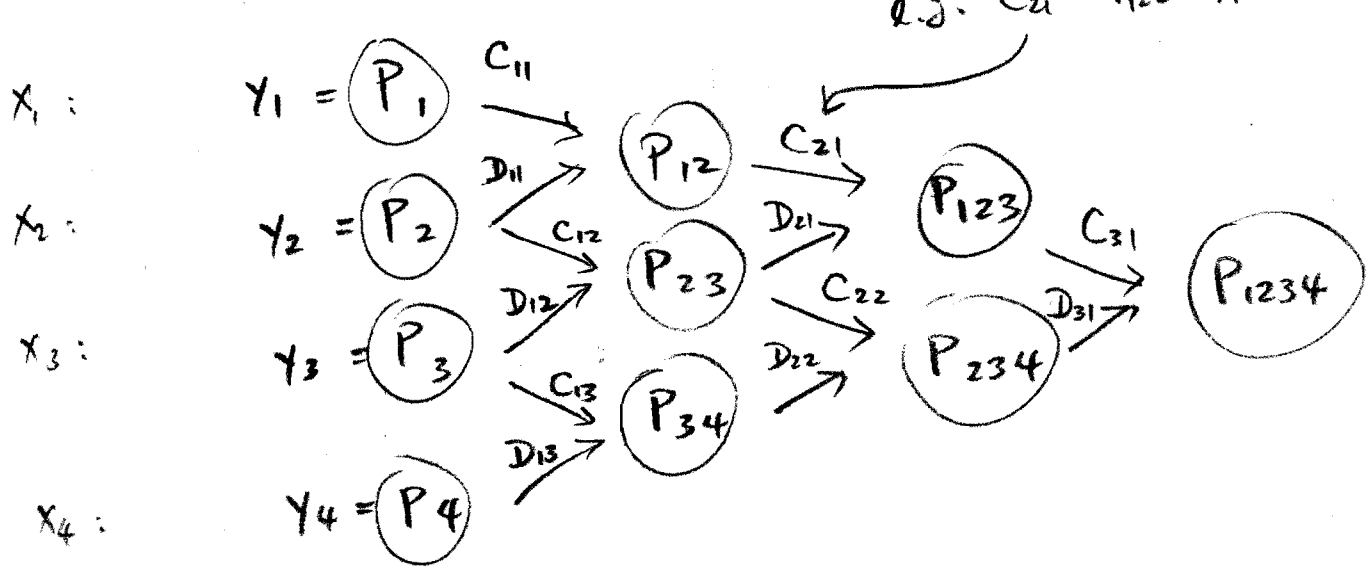
★ Let rewrite the algorithm in terms of their differences :

Define : Downward Differences :

$$C_{m,i} = P_{i\cdots(i+m)} - P_{i\cdots(i+m-1)}$$

Upward Differences :

$$D_{m,i} = P_{i\cdots(i+m)} - P_{(i+1)\cdots(i+m)}$$



(*) becomes (with I.C. $C_{0,i} = P_i$, $D_{0,i} = P_i$)

$$D_{m+1,i} = \frac{(x_{i+m+1} - x)(C_{m,i+1} - D_{m,i})}{x_i - x_{i+m+1}}$$

$$C_{m+1,i} = \frac{(x_i - x)(C_{m,i+1} - D_{m,i})}{x_i - x_{i+m+1}}$$

Usage of this revised tree:

- Generate $\{C_{ij}, D_{ij}\}$ ← ★ Note $\{C_{ij}\ D_{ij}\}$ depends on x! (tree)

★ Final best approximation

$$\longrightarrow P_{1\cdots N}(x) = \underbrace{\text{any } y_i}_{(C_0\text{'s } D_0\text{'s})} + \underbrace{\{C_{m,i}, D_{mi}\}}_{(m \geq 1)}$$

↑ all increments leading from y_i to the apex!

★ (use tree)

(I.C. $C_{0i} = P_i$, $D_{0i} = P_i$)

[Simple check] (our simple 4 pts example)

(12')

$$P_{1234} = \frac{(x - x_4)P_{123} - (x - x_1)P_{234}}{(x_1 - x_4)}$$

$C_{31} = P_{1234} - P_{123}$ downward diff.

$D_{31} = P_{1234} - P_{234}$ upward diff.

} From Graph

$$D_{31} = \frac{(x_4 - x)(C_{22} - D_{21})}{(x_1 - x_4)}$$

From Graph

$C_{22} = P_{234} - P_{23}$

$D_{21} = P_{123} - P_{23}$

$$C_{31} = \frac{(x_1 - x)(C_{22} - D_{21})}{(x_1 - x_4)}$$

$$P_{1234} - P_{234} = \frac{(x_4 - x)}{(x_1 - x_4)}\left(P_{234} - \cancel{P_{23}} - P_{123} + \cancel{P_{23}}\right)$$

$$P_{1234} - P_{123} = \frac{(x_1 - x)}{(x_1 - x_4)}(P_{234} - P_{123})$$

$$P_{1234} = \left(1 + \frac{x_4 - x}{x_1 - x_4}\right)P_{234} + \frac{(x - x_4)}{(x_1 - x_4)}P_{123}$$

$$= \frac{x_1 - \cancel{x_4} + \cancel{x_4} - x}{x_1 - x_4}P_{234} + \frac{(x - x_4)}{(x_1 - x_4)}P_{123}$$

$$P_{1234} = \frac{(x - x_4)P_{123} - (x - x_1)P_{234}}{(x_1 - x_4)} \quad \checkmark$$

Estimating the Coefficients of the Interpolating Polynomial

Newton's Divided Differences

★ the construction recursion to the Neville's Algorithm.

- given N data points : $\{(x_1, f_1), \ldots, (x_N, f_N)\}$

- Define the following set of #s iteratively,

$$f[x_k] = f_k$$

$$f[x_k\, x_{k+1}] = \frac{f[x_k] - f[x_{k+1}]}{x_k - x_{k+1}}$$

$$f[x_k\, x_{k+1} x_{k+2}] = \frac{f[x_{k+1}\, x_{k+2}] - f[x_k\, x_{k+1}]}{x_k - x_{k+2}}$$

$$f[x_k\, x_{k+1}, x_{k+2}\, x_{k+3}] = \frac{f[x_{k+1}\, x_{k+2}\, x_{k+3}] - f[x_k\, x_{k+1}\, x_{k+2}]}{x_k - x_{k+3}}$$

$$\vdots$$

★ And, the interpolating polynomial of degree (N-1) is given by :

$$\rightarrow P_{N-1}(x) = f[x_1] + f[x_1\, x_2](x - x_1) + f[x_1, x_2\, x_3](x - x_1)(x - x_2) + f[x_1, x_2\, x_3\, x_4](x - x_1)(x - x_2)(x - x_3)$$

$$\vdots$$

$$+ f[x_1 \ldots x_N](x - x_1) \cdots (x - x_{N-1})$$

The f[ ]'s can be put into a table similar to the Neville's table:

x_1	$f[x_1]$	$f[x_1 x_2]$	$f[x_1 x_2 x_3]$	$f[x_1 x_2 x_3 x_4]$
x_2	$f[x_2]$	$f[x_2 x_3]$	$f[x_2 x_3 x_4]$	
x_3	$f[x_3]$	$f[x_3 x_4]$		
x_4	$f[x_4]$			

$\longrightarrow$

fill in left to right

- the top row values are used to form $P_{N-1}(x)$.

Notes: (1) $f[x_k]$, $f[x_k \dots x_{k+i}]$ are numbers here as contrast to the Neville's Algorithm, $P_{1234}(x)$ are polynomials!

(2) $f[x_k]$, $f[x_k \dots x_{k+i}]$ form the set of coefficients for $P_{N+1}(x)$!

(3) By the Main Theorem of Polynomial Interpolation, there is only one unique $P_{N+1}(x)$ so that both will give the same interpolating polynomial for the given set of data points.

usage: - Lagrange / Neville → for evaluating $P_{N+1}(\bar{x})$ at a particular $\bar{x}$ explicitly.
(with C's & D's, better in dealing with round off errors but requires calculation of table for each $\bar{x}$)
- Newton's Divided Difference → to calculate coefficients for $P_{N+1}(x)$ once and for all (less computations with multiple $\bar{x}$'s.)

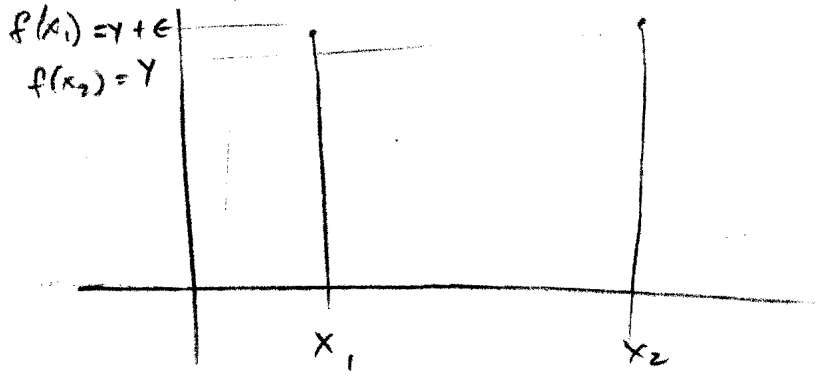
Example on the inaccuracy using the coefficients in interpolating polynomials

two points $N=2$:

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

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

problem situation :



Y large
ε small

$f(x_1) - f(x_2)$ = small number ϵ from the subtraction of two large #s !

→ significant lost of precision possible !

☆ ☆ → since, the original Lagrange's formula & the up/downward improvement do not have this problem.

— See Press Sec. 3.5

One can rewrite (*) :

$$\begin{cases} D_{i,l+1} = \dfrac{x_i - x}{x_i - x_{i+l+1}} \left(D_{i+1,l} - U_{i,l} \right) \\ U_{i,l+1} = \dfrac{x_{i+l+1} - x}{x_i - x_{i+l+1}} \left(D_{i+1,l} - U_{i,l} \right) \end{cases}$$

Interpolation w/ rational functions

$$R_{i,(i+1)\dots(i+m)}(x) = \frac{P_M(x)}{Q_N(x)} = \frac{p_0 + p_1 x + \dots + p_M x^M}{1 + q_1 x + \dots + q_N x^N}$$

↑ Rational function passing $(m+1)$ pts

↑ $M+N+1$ unknown

In order to approx R with $\frac{P}{Q}$, we need

$$m+1 = M+N+1 \; !$$

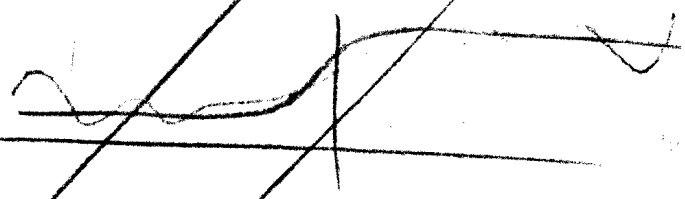
☆ The derivation of recursion relation is analogous to (*) for polynomials, check Wong / Press.

Note: Rational interpolation is used sometime to model function with poles, where regular polynomial function might fail.

$$f(x) = \frac{g(x)}{(x-x_0)^m}, \text{ where } g(x) \text{ is nonsingular!}$$

☆☆ higher order ≠ more accurate!

P.92 DeVries



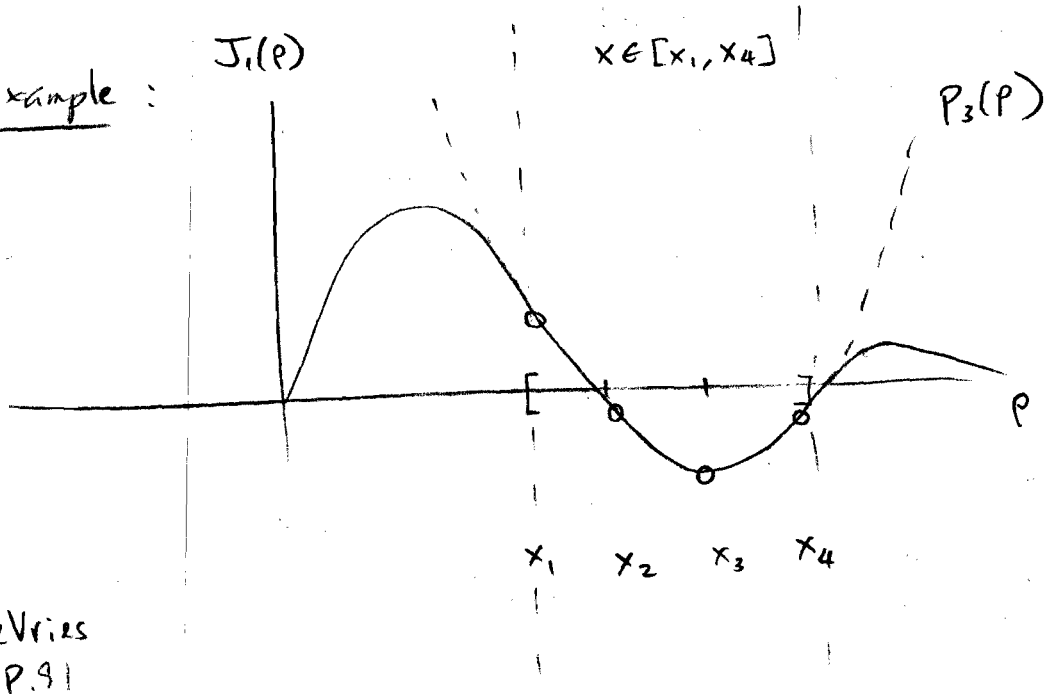
- $P_N(x)$ will in general have N bumps
- might give spurious oscillations.

Issues with Interpolation (extrapolation)

- extrapolation : $x \notin [x_1, x_N]$ (outside of data range)

☆ near is ok but evaluating $P_{N-1}(x)$ far outside $[x_1, x_N]$ is dangerous !

Example :



DeVries P.91

- Runge's phenomenon : - $P_N(x)$ with larger N (a higher order polynomial) does not $\Rightarrow$ higher accuracy !
  - $P_N(x)$ (with larger N) tends to have more oscillations .

15'

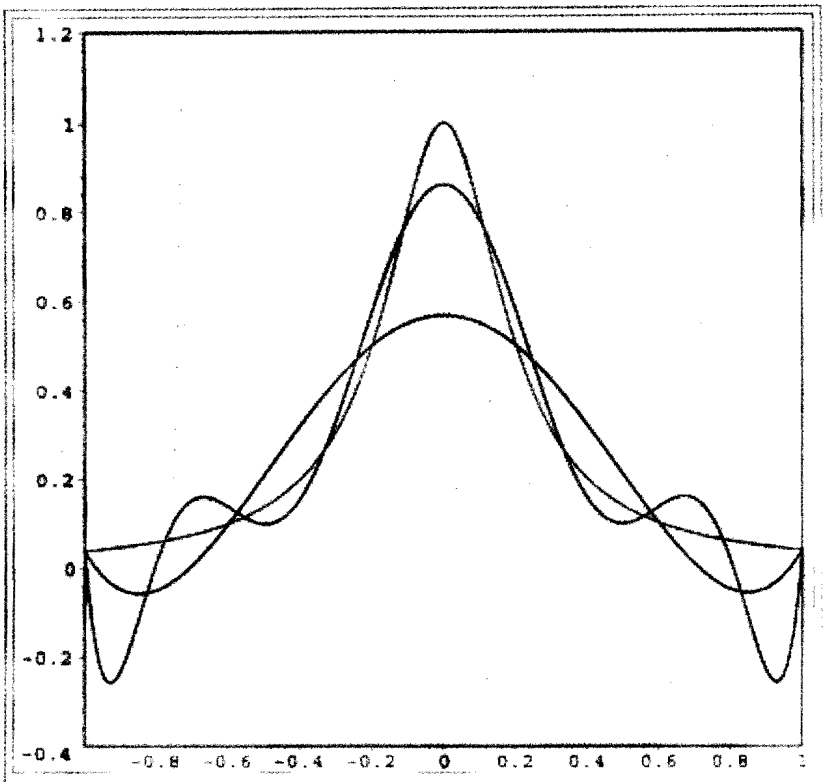
Runge's phenomenon

From Wikipedia, the free encyclopedia

In the mathematical field of numerical analysis, **Runge's phenomenon** is a problem that occurs when using polynomial interpolation with polynomials of high degree. It was discovered by Carl David Tolmé Runge when exploring the behaviour of errors when using polynomial interpolation to approximate certain functions (Runge 1901).

Contents

- 1 Problem
  - 1.1 Reason
- 2 Mitigations to the problem of Runge's phenomenon
- 3 See also
- 4 References



Phenomenon in a nutshell
The red curve is the Runge function.
The blue curve is a 5th-order interpolating polynomial (using six equally-spaced interpolating points).
The green curve is a 9th-order interpolating polynomial (using ten equally-spaced interpolating points).
At the interpolating points, the error between the function and the interpolating polynomial is (by definition) zero. Between the interpolating points (especially in the region close to the endpoints 1 and −1), the error between the function and the interpolating polynomial gets worse for higher-order polynomials.

Problem

Consider the function:

$$f(x) = \frac{1}{1 + 25x^2}.$$

Runge found that if this function is interpolated at equidistant points x_i between −1 and 1 such that:

$$x_i = -1 + (i-1)\frac{2}{n}, \qquad i \in \{1, 2, \ldots, n+1\}$$

with a polynomial $P_n(x)$ of degree $\leq n$, the resulting interpolation oscillates toward the end of the interval, i.e. close to −1 and 1. It can even be proven that the interpolation error tends toward infinity when the degree of the polynomial increases:

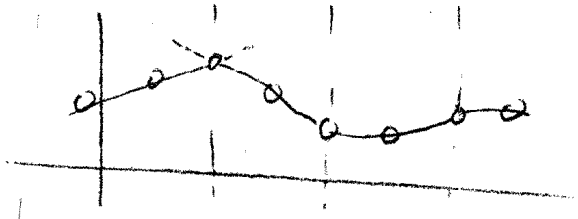
$$\lim_{n \to \infty} \left(\max_{-1 \leq x \leq 1} |f(x) - P_n(x)| \right) = \infty.$$

- These oscillations might have nothing to do with the actual function!

- Obviously, adding more pts near x (the evaluation pt) will help, but one should not interpolate the whole data range globally.

★ Recall that interpolation reflects local behavior ...

★ So,
- one should interpolate a section of the data points $[x_i, x_{i+k}]$ with $k < 4$ or 5 pts.
- the full range is a piece-wise spline of these smaller sectional interpolations.



$P_2(x)$ $\bar{P}_2(x)$ ...

★ But, $P_2(x)$ & $\bar{P}_2(x)$ won't be smooth across sections.
(still continuous by construction)

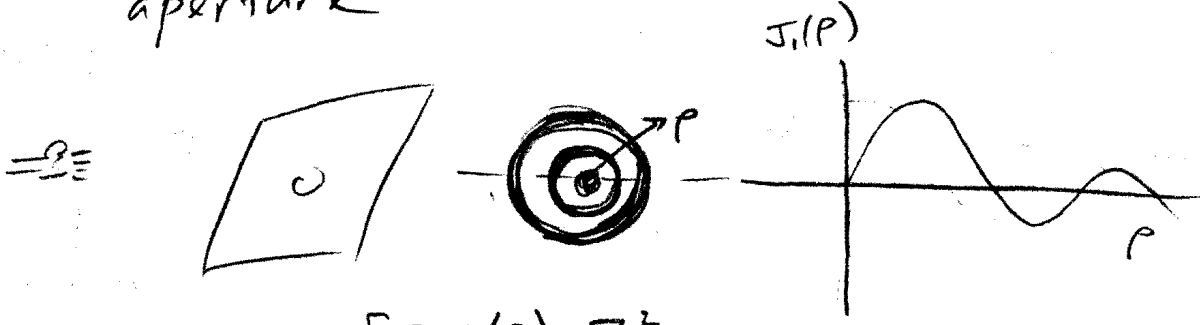
Hermite Interpolation

☆ higher order with distance info DOES NOT help!

→ But, if we know $f'(x_i)$ also, then, we can interpolate / extrapolate with higher order.

An interesting example is the Bessel function

- Diffraction pattern for circular aperture



$$I = I_0 \left[\frac{2J_1(\rho)}{\rho} \right]^2$$

Airy pattern

- ϕ electric potential for circular domains.

Identities:

$$\frac{d}{dx}\left(x^m J_m(x)\right) = x^m J_{m-1}(x) \qquad (1)$$

$$J_{m+1}(x) = \frac{2m}{x} J_m(x) - J_{m-1}(x) \qquad (2)$$

$(1) \rightarrow \quad m x^{m-1} J_m(x) + x^m J_m'(x) = x^m J_{m-1}(x)$

$$J_m'(x) = J_{m-1}(x) - \frac{m}{x} J_m(x)$$

$m=0$: $\quad J_0'(x) = J_{-1}(x)$

From (2): $\quad J_1(x) = -J_{-1}(x)$

$$\Rightarrow \boxed{J_0'(x) = -J_1(x)}$$

$m=1$: $\quad J_1'(x) = J_0(x) - \frac{J_1(x)}{x}$

From (2): $\quad J_2(x) = \frac{2}{x} J_1(x) - J_0(x)$

$$\rightarrow \frac{J_1(x)}{x} = \frac{J_2(x)}{2} + \frac{J_0(x)}{2}$$

$$\Rightarrow \boxed{J_1'(x) = \frac{J_0(x) - J_2(x)}{2}}$$