

Lecture 9

Interpolation and Numerical Differentiation

Ying-Jer Kao

Department of Physics
National Taiwan University

September 10, 2024

Outline

- 1 Interpolation
 - Polynomial Interpolation

- 2 Numerical Differentiation
 - Finite Difference Approximations
 - Richardson's Extrapolation

Interpolation

- Discrete data sets of the form

x	x_0	x_1	x_2	\cdots	x_n
y	y_0	y_1	y_2	\cdots	y_n

are commonly involved in technical calculations.

- Is it possible to find a **simple** and **convenient** formula that reproduces the given points exactly?
- If the data set contains errors, is it possible to find a formula to **represent the data approximately** and, filters out the errors?
- If the computation of a function f is very **expensive to evaluate**, is it possible to find another function g which is simpler to evaluate and gives a reasonable approximation of f ?

Polynomial Interpolation

- Given discrete data sets of the form

x	x_0	x_1	x_2	\cdots	x_n
y	y_0	y_1	y_2	\cdots	y_n

where x_i 's form a set of $n + 1$ distinct points.

- Find a polynomial that is defined for all x , and takes on the corresponding values of y_i for each of the $n + 1$ distinct x_i 's.
- A polynomial p for which $p(x_i) = y_i$ when $0 \leq i \leq n$ is said to interpolate the table. The points x_i are called **nodes**.

Lagrange's Method

- It is always possible to construct a **unique** polynomial of degree n that passes through $n + 1$ distinct data points.
- The polynomial can be written as

$$P_n(x) = \sum_{i=0}^n l_i(x) y_i$$

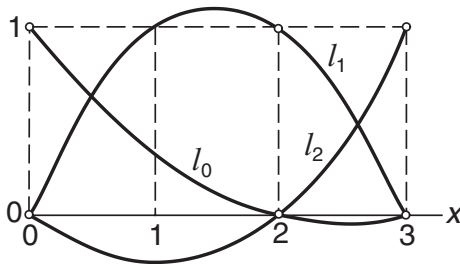
where the **cardinal functions** $l_i(x)$ are

$$\begin{aligned} l_i(x) &= \frac{x - x_0}{x_i - x_0} \cdot \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_n}{x_i - x_n} \\ &= \prod_{j=0; j \neq i}^n \frac{x - x_j}{x_i - x_j} \end{aligned}$$

- The cardinal functions are n degree polynomials and have the property $l_i(x_j) = \delta_{ij}$.

Example: Cardinal Functions

- Cardinal functions for a three point interpolation ($n = 2$) with $x_0 = 0, x_1 = 2$ and $x_2 = 3$.



Newton's Method

- Lagrange's method is conceptually simple, but it can not be computed efficiently.
- A better computational method is the **Newton's method**, and the resulting polynomial is said to have the **Newton form**.

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \cdots \\ + (x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{n-1})a_n.$$

- The Newton and Lagrange forms are just two different derivations for precisely the **same polynomial**.
- The Newton form has the advantage of easy extensibility to accommodate additional data points.

Newton's Method

- Consider there are four data points ($n = 3$), and the interpolating polynomial is

$$\begin{aligned}P_n(x) &= a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 \\&\quad + (x - x_0)(x - x_1)(x - x_2)a_3 \\&= a_0 + (x - x_0)\{a_1 + (x - x_1)[a_2 + (x - x_2)a_3]\}\end{aligned}$$

- The polynomial can be evaluated backwards with the recurrence relation:

$$P_0(x) = a_3$$

$$P_1(x) = a_2 + (x - x_2)P_0(x)$$

$$P_2(x) = a_1 + (x - x_1)P_1(x)$$

$$P_3(x) = a_0 + (x - x_0)P_2(x)$$

Newton's Method

- For arbitrary n ,

$$P_0(x) = a_n \quad P_k(x) = a_{n-k} + (x - x_{n-k})P_{k-1}(x), \quad k = 1, 2, \dots, n$$

- The coefficients of P_n are determined by the condition $y_i = P_n(x_i), i = 0, 1, \dots, n$. This yields the coupled equations

$$y_0 = a_0$$

$$y_1 = a_0 + (x_1 - x_0)a_1$$

$$y_2 = a_0 + (x_2 - x_0)a_1 + (x_2 - x_0)(x_2 - x_1)a_2$$

$$\vdots$$

$$y_n = a_0 + (x_n - x_0)a_1 + \cdots + (x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})a_n$$

Divided Differences

- Define the divided-difference notation

$$\nabla y_i = \frac{y_i - y_0}{x_i - x_0}, \quad i = 1, 2, \dots, n$$

$$\nabla^2 y_i = \frac{\nabla y_i - \nabla y_1}{x_i - x_1}, \quad i = 2, 3, \dots, n$$

$$\nabla^3 y_i = \frac{\nabla^2 y_i - \nabla^2 y_2}{x_i - x_2}, \quad i = 3, 4, \dots, n$$

$$\vdots$$

$$\nabla^n y_n = \frac{\nabla^{n-1} y_i - \nabla^{n-1} y_{n-1}}{x_n - x_{n-1}}.$$

- The solution for the coefficients is

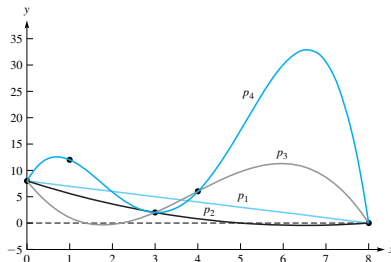
$$a_0 = y_0, \quad a_1 = \nabla y_1, \quad a_2 = \nabla^2 y_2, \quad \dots \quad a_n = \nabla^n y_n$$

Errors in Polynomial Interpolation

- When a function f is approximated on an interval $[a, b]$ by an interpolating polynomial p , the discrepancy between f and p will (theoretically) be zero at each node of interpolation.
- A natural expectation is that the function f will be well approximated at all intermediate points and that as the number of nodes increases, this agreement will become better and better.
- If the function f is well-behaved, it is **dangerous** to assume that the differences $|f(x) - p(x)|$ will be small when the number of interpolating nodes is large, even for functions that possess continuous derivatives of all orders on the interval.

Errors in Polynomial Interpolation

- Five data points: $(0, 8)$, $(1, 12)$, $(3, 2)$, $(4, 6)$, $(8, 0)$.



- With more points added, the situation became **worse** instead of better!
- The reason is that a polynomial of degree n has n zeros. If all of these zero points are real, then the curve crosses the x -axis n times, resulting in **wild oscillations**.

Errors in Polynomial Interpolation

- It can be shown that the error in polynomial interpolation is

$$f(x) - P_n(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_n)}{(n + 1)!} f^{(n+1)}(\xi),$$

where ξ lies in the interval (x_0, x_n) .

Warning

- An interpolating polynomial intersecting more than **six points** must be viewed with suspicion.
- The data points that are far from the point of interest do not contribute to the accuracy of the interpolating polynomial.
- If **extrapolation** using the interpolating polynomial is necessary, one should be careful.
- **Plot the data** and visually verify that the extrapolated value makes sense.
- Use a **low-order polynomial** based on nearest-neighbor data points.
- Work with a plot of $\log x$ vs. $\log y$, which is usually much smoother than the $x - y$ curve, and thus safer to extrapolate.

Numerical Differentiation: Introduction

- Numerical differentiation deals with the following problem: given the function $y = f(x)$, obtain its derivatives at the point $x = x_k$.
- The function is usually given as a set of discrete data points (x_i, y_i) , $i = 0, 1, \dots, n$, which can be the output of some computation or measurement.
- One method to obtain numerical differentiation is through **interpolation**. Approximate the function locally by a **polynomial** and then differentiate it.
- Another method is using **finite difference** method based on the Taylor series.

Warning

- Numerical differentiation is **not** a particularly accurate process.
- It suffers from a conflict between **roundoff errors** (due to limited machine precision) and **errors inherent in interpolation**.
- A derivative of a function can **never** be computed with the same precision as the function itself.

Finite Difference Approximation

- Finite difference approximation is based on Taylor series expansion. It has the advantage of providing us with information about the **error** involved in the approximation.
- The derivation of the finite difference approximations for the derivatives of $f(x)$ is based on **forward** and **backward** Taylor series expansions of $f(x)$ about x .

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + \frac{h^4}{4!} f^{(4)}(x) + \dots$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f'''(x) + \frac{h^4}{4!} f^{(4)}(x) - \dots$$

$$f(x+2h) = f(x) + 2hf'(x) + \frac{(2h)^2}{2!} f''(x) + \frac{(2h)^3}{3!} f'''(x) + \frac{(2h)^4}{4!} f^{(4)}(x) + \dots$$

$$f(x-2h) = f(x) - 2hf'(x) + \frac{(2h)^2}{2!} f''(x) - \frac{(2h)^3}{3!} f'''(x) + \frac{(2h)^4}{4!} f^{(4)}(x) - \dots$$

Taylor Series

- The sums and differences of the series are given by

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{h^4}{12} f^{(4)}(x) + \dots$$

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3} f'''(x) + \dots$$

$$f(x+2h) + f(x-2h) = 2f(x) + 4h^2 f''(x) + \frac{4h^4}{3} f^{(4)}(x) + \dots$$

$$f(x+2h) - f(x-2h) = 4hf'(x) + \frac{8h^3}{3} f'''(x) + \dots$$

- The sums contain only **even** derivatives, whereas the differences retain just the **odd** derivatives.
- These can be viewed as simultaneous equations that can be solved for various derivatives of $f(x)$.

First-Derivative Formulas

- The derivative for the function f at x_0 is

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

- The obvious method to approximate $f'(x_0)$ is

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

for small values of h .

Error

- To estimate the error in the finite difference approximation, we keep the order h^2 term in the Taylor expansion,

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{1}{2}f''(\xi)h^2,$$

where ξ is in the interval between x_0 and $x_0 + h$.

- Rearranging, we have

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{1}{2}f''(\xi)h$$

- The **truncation error** of our approximation is $-\frac{1}{2}f''(\xi)h \sim O(h)$. This error will be present even if the calculations are performed with infinite precision.
- Additional **round-off errors** will be present when performing the calculation on a finite-precision machine.

First Non-Central Difference Formulas

- For positive h , we have the **forward-difference formula**

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

and the **backward-difference formula**

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$$

- The truncation error is of order $O(h)$.

Higher Derivatives

- Approximations for higher derivatives can be derived in the same manner.
- The second derivative of $f(x)$ in the forward difference approximation is

$$f''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} + O(h).$$

- The truncation error is of order $O(h)$.
- Higher order derivatives can be obtained in the same way.

Higher Derivatives: $O(h)$

- Forward finite difference approximations of $O(h)$

	$f(x)$	$f(x+h)$	$f(x+2h)$	$f(x+3h)$	$f(x+4h)$
$hf'(x)$	-1	1			
$h^2 f''(x)$	1	-2	1		
$h^3 f'''(x)$	-1	3	-3	1	
$h^4 f^{(4)}(x)$	1	-4	6	-4	1

- Backward finite difference approximations of $O(h)$

	$f(x-4h)$	$f(x-3h)$	$f(x-2h)$	$f(x-h)$	$f(x)$
$hf'(x)$				-1	1
$h^2 f''(x)$			1	-2	1
$h^3 f'''(x)$		-1	3	-3	1
$h^4 f^{(4)}(x)$	1	-4	6	-4	1

First Central Difference Formulas

- To improve the accuracy, one could combine the following two Taylor series

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \dots$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \dots$$

- Subtracting them, we obtain the **first central difference approximation** for $f'(x)$:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

- The truncation error is of order $O(h^2)$.
- This formula for numerical differentiation is very useful in the numerical solution of certain differential equations.

Higher Derivatives: $O(h^2)$

- Second derivative is

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

- The truncation error is of order $O(h^2)$.
- Central finite difference approximations of $O(h^2)$:

	$f(x-2h)$	$f(x-h)$	$f(x)$	$f(x+h)$	$f(x+2h)$
$2hf'(x)$		-1	0	1	
$h^2 f''(x)$		1	-2	1	
$2h^3 f'''(x)$	-1	2	0	-2	1
$h^4 f^{(4)}(x)$	1	-4	6	-4	1

Second Non-Central Difference Formulas

- For the forward or backward difference approximations, it is also preferable to use formulas with error of order $O(h^2)$.
- For example, to obtain $f'(x)$, we use

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \cdots$$

$$f(x+2h) = f(x) + 2hf'(x) + 2h^2f''(x) + \frac{4h^3}{3}f'''(x) + \frac{2h^4}{3}f^{(4)}(x) + \cdots$$

- Eliminate $f''(x)$, we obtain the **second forward difference** formula,

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + O(h^2)$$

- The truncation error is of order $O(h^2)$.

Higher Derivatives: $O(h^2)$

- Forward finite difference approximations of $O(h^2)$

	$f(x)$	$f(x+h)$	$f(x+2h)$	$f(x+3h)$	$f(x+4h)$	$f(x+5h)$
$2hf'(x)$	-3	4	-1			
$h^2 f''(x)$	2	-5	4	-1		
$2h^3 f'''(x)$	-5	18	-24	14	-3	
$h^4 f^{(4)}(x)$	3	-14	26	-24	11	-2

- Backward finite difference approximations of $O(h^2)$

	$f(x-5h)$	$f(x-4h)$	$f(x-3h)$	$f(x-2h)$	$f(x-h)$	$f(x)$
$2hf'(x)$				1	-4	3
$h^2 f''(x)$			-1	4	-5	2
$2h^3 f'''(x)$		3	-14	24	-18	5
$h^4 f^{(4)}(x)$	-2	11	-24	26	-14	3

Errors in Finite Difference Approximations

- The effect on since in all finite difference expressions the sum of the coefficients is zero.
- If h is very small, the values of $f(x), f(x \pm h), f(x \pm 2h)$, etc. will be approximately equal. The **roundoff error** can be dominant and significant figures will be lost.
- If h is too big, the **truncation errors** can dominate.
- To minimize this problem, use double- or higher-precision arithmetics, and employ finite difference formulas of $O(h^2)$.

Derivatives by Interpolation

- If $f(x)$ is given as a set of discrete data points, interpolation can be a very effective means of computing its derivatives.
- The derivative of $f(x)$ is approximated by the derivative of the interpolant.
- This method is particularly useful if the data points are located at **uneven intervals** of x , when the finite difference approximations discussed previously are not applicable.

Polynomial Interpolant

- We want to fit the data to the polynomial of degree n

$$P_{n-1}(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_nx^n$$

through $n + 1$ data points and evaluate its derivative.

- The degree of the polynomial is limited to less than 6 in order to avoid spurious **oscillations** of the interpolant.
- These oscillations are **magnified** with each differentiation.
- The interpolation should be a local one, involving no more than a few nearest-neighbor data points.
- When the data is noisy, it is advisable to use the **least-squares fit** to find the best fitting polynomial.

First-Derivative Formulas via Interpolation Polynomial

- If p is the polynomial of degree ≤ 1 that interpolate f at two nodes x_1, x_2 ,

$$p_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) = f(x_1) + f[x_1, x_2](x - x_1)$$

- The first derivative

$$f'(x) \approx p'_1(x) = f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}.$$

- Interpolation through 3 nodes x_1, x_2, x_3 ,

$$p_2(x) = f(x_1) + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2)$$

where $f[x_1, x_2, x_3] = (f[x_2, x_3] - f[x_1, x_2]) / (x_3 - x_1)$.

- The first derivative

$$f'(x) \approx p'_2(x) = f[x_1, x_2] + f[x_1, x_2, x_3](2x - x_1 - x_2).$$

Richardson's Extrapolation

- Richardson's extrapolation is a method to generate **high-accuracy results** using **low-order formulas**.
- For a given quantity G , if we can approximate it by $g(h)$, which depends on h , with an error $E(h) = ch^p$, where c and p are constants, such that,

$$G = g(h) + E(h).$$

- For a given $h = h_1$, $G = g(h_1) + ch_1^p$, and for $h = h_2$, $G = g(h_2) + ch_2^p$.
- Eliminating c and solving for G , we obtain the **Richardson extrapolation formula**,

$$G = \frac{\left(\frac{h_1}{h_2}\right)^p g(h_2) - g(h_1)}{\left(\frac{h_1}{h_2}\right)^p - 1}.$$

Application to Differentiation

- It is common practice to choose $h_2 = h_1/2$, and

$$G = \frac{2^p g(h_1/2) - g(h_1)}{2^p - 1}.$$

- Consider the central difference formula

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x-h)}{2h} + a_2 h^2 + a_4 h^4 + a_6 h^6 + \dots \\ &= \phi(h) + a_2 h^2 + a_4 h^4 + a_6 h^6 + \dots \end{aligned}$$

- $\phi(h) = \frac{f(x+h) - f(x-h)}{2h}$ is an approximation to $f'(x)$ with an error of order $O(h^2)$.

Central Difference Approximation

- Evaluate ϕ at h and $h/2$,

$$\phi(h) = f'(x) - a_2 h^2 - a_4 h^4 - a_6 h^6 - \dots$$

$$\phi\left(\frac{h}{2}\right) = f'(x) - a_2 \left(\frac{h}{2}\right)^2 - a_4 \left(\frac{h}{2}\right)^4 - a_6 \left(\frac{h}{2}\right)^6 - \dots$$

- Eliminate the dominant error term $O(h^2)$,

$$\phi\left(\frac{h}{2}\right) + \frac{1}{3} \left[\phi\left(\frac{h}{2}\right) - \phi(h) \right] = f'(x) + \frac{1}{4} a_4 h^4 + \frac{5}{16} a_6 h^6 + \dots$$

- The precision is improved to $O(h^4)$ because the error series of the new combination begins with $\frac{1}{4} a_4 h^4$.

Improved Precision

- Define

$$\Phi(h) = \phi\left(\frac{h}{2}\right) + \frac{1}{3} \left[\phi\left(\frac{h}{2}\right) - \phi(h) \right]$$

- Repeat the process, we obtain

$$\Phi\left(\frac{h}{2}\right) + \frac{1}{15} \left[\Phi\left(\frac{h}{2}\right) - \Phi(h) \right] = f'(x) - \frac{1}{20} b_6 h^6 + \dots$$

- The precision is improved to $O(h^6)$.
- The same procedure can be **repeated over and over again** to kill higher and higher terms in the error.

General Form

- Let $N(h)$ be a function which approximates an unknown M , such that

$$M = N(h) + \sum_{k=1}^{\infty} a_k h^k.$$

- Assumed that $N(h)$ can be computed for any $h > 0$, so

$$M = N\left(\frac{h}{2}\right) + \sum_{k=1}^{\infty} a_k \left(\frac{h}{2}\right)^k.$$

- Eliminate the term involving a_1 ,

$$M = \left[2N_1\left(\frac{h}{2}\right) - N_1(h) \right] + a_2 \left(\frac{h^2}{2} - h^2 \right) + a_3 \left(\frac{h^3}{4} - h^3 \right) + \cdots,$$

where $N_1(h) \equiv N(h)$.

General Form

- Define

$$N_2(h) = \left[2N_1\left(\frac{h}{2}\right) - N_1(h) \right] = N_1\left(\frac{h}{2}\right) + \left[N_1\left(\frac{h}{2}\right) - N_1(h) \right]$$

and we obtain the $O(h^2)$ approximation formula ,

$$M = N_2(h) - \frac{a_2}{2}h^2 - \frac{3a_3}{4}h^3 - \dots .$$

- Repeat the process again, we obtain the $O(h^3)$ formula,

$$\begin{aligned} M &= \left[N_2\left(\frac{h}{2}\right) - \frac{N_2\left(\frac{h}{2}\right) - N_2(h)}{3} \right] + \frac{a_3}{8}h^3 + \dots \\ &= N_3(h) + \frac{a_3}{8}h^3 + \dots . \end{aligned}$$

General Form

- The process can be repeated to construct an $O(h^4)$ approximation,

$$N_4(h) = N_3\left(\frac{h}{2}\right) + \frac{N_3\left(\frac{h}{2}\right) - N_3(h)}{7},$$

and $O(h^5)$ approximation ,

$$N_5(h) = N_4\left(\frac{h}{2}\right) + \frac{N_4\left(\frac{h}{2}\right) - N_4(h)}{15}.$$

General Form

- In general, if M can be written as

$$M = N(h) + \sum_{j=1}^{m-1} a_j h^j + O(h^m)$$

then for each $j = 2, 3, \dots, m$, we have an $O(h^j)$ approximation of the form

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}\left(\frac{h}{2}\right) - N_{j-1}(h)}{2^{j-1} - 1}.$$

General Form

- The approximations are generated by rows in the order indicated in the following table ($N_1(h) \equiv N(h)$):

$O(h)$	$O(h^2)$	$O(h^3)$	$O(h^4)$
1: $N_1(h)$			
2: $N_1\left(\frac{h}{2}\right)$	3: $N_2(h)$		
4: $N_1\left(\frac{h}{4}\right)$	5: $N_2\left(\frac{h}{2}\right)$	6: $N_3(h)$	
7: $N_1\left(\frac{h}{8}\right)$	8: $N_2\left(\frac{h}{4}\right)$	9: $N_3\left(\frac{h}{2}\right)$	10: $N_4(h)$