Numerical Differentiation

by Loïc Quertenmont, PhD

LINFO1113 - 2019-2020

Programme

Cours	1	Librairies mathématiques, représentation des nombres en Python et erreurs liées
Cours	2,3	Résolution des systèmes linéaires
Cours	4,5	Interpolation et Régression Linéaires
Cours	6,7	Racines d'équations
Cours	**8**	**Différentiation numérique**
Cours	9	Intégration numérique
Cours	10,11,12	Introduction à l'optimisation
Cours	13	Rappel / Répétition

Outline

- **Introduction**
- **Finite Difference Approximation**
- **Richardson extrapolation**
- **Derivatives by interpolation**

Introduction

Introduction

- Goal:
  - Given the function $f(x)$, compute $\frac{d^n f}{dx^n}$ at given x
  - Given means
    - We know the algorithm $\mathrm{y} = f(x)$
    - OR we have a discrete set of points (x_i, y_i) resulting from the evaluation of the function/algorithm

Introduction

- Numerical differentiation is related to interpolation
  - One solution is to interpolate by a polynomial and derivate the polynomial itself.

- Taylor series expansion of $f(x)$ about the point x_k is another valid solution
  - This has the advantage to also provide us with the error involved in the approximation

- Numerical differentiation is not an accurate process.
  - Roundoff errors (caused by limited machine precision)
  - + Errors inherent in interpolation.
  - → Function derivative as lower precision that the function itself.

Finite Difference Approximation

Taylor Series Expansion

- Taylor Series Expansion about x

- Forward / Backward:

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

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

- Forward / Backward (2 steps):

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

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

Sum and Difference

- Sum and Difference:

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

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

- Sum and Difference (2 steps):

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

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

First Central Difference for f'

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

- The solution of equation *f)* for f' is:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(x) - \cdots$$

- Or

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

- This is the first central difference approximation for $f'(x)$
- The term $O(h^2)$ reminds us that the truncation error behaves as h^2

First Central Difference for f''

- *e):* $f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{h^4}{12} f^{(4)}(x) + \cdots$

- The solution of equation *e)* for f'' is:

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

- Or

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

- This is the first central difference approximation for $f''(x)$

First Central Difference

- The First Central difference approximation for other derivative can be obtain in the same manner from *a)* to *h)*

- $f'''(x)$ is obtained by eliminating $f'(x)$ from *f)* and *h)*:

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

- $f^{(4)}(x)$ is obtained by eliminating $f''(x)$ from *e)* and *g)*:

$$f^{(4)}(x) = \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4} + O(h^2)$$

First Central Difference

- Summary of the coefficients of the First Central difference approximation 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

First Non-Central Difference approximation

- If the function $\boldsymbol{f}$ is evaluated at points $\boldsymbol{x_0}$ till $\boldsymbol{x_n}$

- First derivate of $\boldsymbol{f}$ at $\boldsymbol{x_1}$ is computed using $\boldsymbol{x_0}$ and $\boldsymbol{x_2}$

- We are not able to compute the derivative of $\boldsymbol{f}$ at point $\boldsymbol{x_0}$
- Similarly for $\boldsymbol{x_n}$

- Central finite difference approximation is not always usable
- There is a need for finite difference expressions that require evaluations of the function only on one side of $\boldsymbol{x}$
  - **Forward and Backward finite difference approximation**

First Non-Central Difference approximation

- Non-Central difference can also be obtained from equations a) to h)

- Solving a) for $f'(x)$ leads to the **first forward difference**:

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

- Solving b) for $f'(x)$ leads to the **first backward difference**:

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

- **Note that the truncation error is now of order $\boldsymbol{h}$**
  This is not as good as the h^2 of central difference approximation

First Non-Central Difference approximation

- We can derive the higher derivative in the same manner

- For instance *a)* and *c)* yield:

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

- For instance *b)* and *d)* yield:

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

First Non-Central Difference approximation

- Coefficients of the First non-Central difference approximation of $O(h)$
- Forward

	$f(x)$	$f(x+h)$	$f(x+2h)$	$f(x+3h)$	$f(x+4h)$
$hf'(x)$	-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

- Back

	$f(x-4h)$	$f(x-3h)$	$f(x-2h)$	$f(x-h)$	$f(x)$
$hf'(x)$				-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

Second Non-Central Difference

- First non-central difference is not so popular because of the larger truncation error ($O(h)$) compared to central difference ($O(h^2)$)
- We can do better, by retaining more terms in the Taylor expansion

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

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

- We can eliminate f'' by taking c) – 4x a)
  - $f(x+2h) - 4f(x+h) = -3f(x) - 2hf'(x) + \frac{2h^3}{3} f'''(x) + \cdots$
  - We obtain the second non-central difference by solving it
    - $f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + O(h^2)$

Second Non-Central Difference

- Coeff. for Second non-central difference approximation of $O(h^2)$
- Forward

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

- Backward

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

- **Remark: The sum of the coefficient is always 0**

Example

- Given the data points

x	0	0.1	0.2	0.3	0.4
$f(x)$	0.0000	0.0819	0.1341	0.1646	0.1797

- Compute $f'(x)$ and $f''(x)$ at $x = 0$ and 0.2 using finite diff @ $O(h^2)$
- **Forward finite difference for x=0**

	$f(x)$	$f(x+h)$	$f(x+2h)$	$f(x+3h)$	$f(x+4h)$	$f(x+5h)$
$2hf'(x)$	-3	4	-1			
$h^2 f''(x)$	2	-5	4	-1		

$$f'(0) = \frac{-3f(0) + 4f(0.1) - f(0.2)}{2(0.1)} = \frac{-3(0) + 4(0.0819) - 0.1341}{0.2} = 0.967$$

$$f''(0) = \frac{2f(0) - 5f(0.1) + 4f(0.2) - f(0.3)}{(0.1)^2}$$

$$= \frac{2(0) - 5(0.0819) + 4(0.1341) - 0.1646}{(0.1)^2} = -3.77$$

Example

- Given the data points

x	0	0.1	0.2	0.3	0.4
$f(x)$	0.0000	0.0819	0.1341	0.1646	0.1797

- Compute $f'(x)$ and $f''(x)$ at $x = 0$ and 0.2 using finite diff @ $O(h^2)$
- **Central finite difference for x=0.2**

	$f(x-2h)$	$f(x-h)$	$f(x)$	$f(x+h)$	$f(x+2h)$
$2hf'(x)$		-1	0	1	
$h^2 f''(x)$		1	-2	1	

$$f'(0.2) = \frac{-f(0.1) + f(0.3)}{2(0.1)} = \frac{-0.0819 + 0.1646}{0.2} = 0.4135$$

$$f''(0.2) = \frac{f(0.1) - 2f(0.2) + f(0.3)}{(0.1)^2} = \frac{0.0819 - 2(0.1341) + 0.1646}{(0.1)^2} = -2.17$$

Errors in Finite Difference Approximations

- **If $\boldsymbol{h}$ is very small, the values of $\boldsymbol{f(x), f(x \pm h),\ f(x \pm 2h)}$, are approximately equal.**
  - They are multiplied by the coefficients and added
  - As the sum of the coefficient is 0, they cancel out each others.
  - Several significant figures can be lost (due to roundoff errors).
  - The effect on the roundoff error can be profound.

- **But $\boldsymbol{h}$ can not be too large, because then the truncation error would become excessive.**

- **There is no solutions, but:**
  - **Use double precisions**
  - **Use formulate that are accurate to at least $\boldsymbol{O(h^2)}$**

Errors in Finite Difference Approximations

- As an illustration, let's compute $f''(1)$ for $f(x) = e^{-x}$
  - Using first central finite difference approximation
  - Different values for h and for the precision

h	Six-Digit Precision	Eight-Digit Precision
0.64	0.380 610	0.380 609 11
0.32	0.371 035	0.371 029 39
0.16	0.368 711	0.368 664 84
0.08	0.368 281	0.368 076 56
0.04	0.368 75	0.367 831 25
0.02	0.37	0.3679
0.01	0.38	0.3679
0.005	0.40	0.3676
0.0025	0.48	0.3680
0.00125	1.28	0.3712

Truncation error dominates

3 correct figures

4 correct figures

Roundoff error dominates

- Real solution: $f''(1) = e^{-1} = 0.367\,879\,44.$

Richardson Extrapolation

Richardson Extrapolation

- Richardson Extrapolation is a simple procedure to boost the accuracy of certain numerical procedures,
  - including finite difference approximations,
  - but we will also use it later in this course.

Richardson Extrapolation

- **General Context:**
  - Suppose that we have an approximate means of computing some quantity G.
  - Assume that the result depends on a parameter h.
    Denoting the approximation by g(h),
  - we have G = g(h) + E(h)
    - where E(h) represents the error

- **Richardson extrapolation can remove the error, provided that it has the form $\boldsymbol{E(h) = ch^p}$, $\boldsymbol{c}$ and $\boldsymbol{p}$ being constants**

Richardson Extrapolation

- We start by computing $g(h)$ with some value of h, let's say $h = h_1$

$$G = g(h_1) + c{h_1}^p$$

- Then we repeat the calculation with $h = h_2$

$$G = g(h_2) + c{h_2}^p$$

- Combining these two equations to eliminate c and solving for G, We get the **Richardson extrapolation formula**

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

- In the particular case where $h_2 = h_1/2$, we get:

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

Example

- Illustration for $(e^{-x})''$ at $x = 1$
- Using the results of table on slide 21 with 6 digits precision

- As the extrapolation only work on truncation error,
  we must confine h to values that produces negligible roundoff errors.

$$g(0.64) = 0.380\,610 \qquad g(0.32) = 0.371\,035$$

- The truncation error in central difference approximation is

$$E(h) = O(h^2) = c_1h^2 + c_2h^4 + c_3h^6 + \ \ldots$$

- Therefore, we can eliminate the first (dominant) error term if we substitute $p = 2$ and $h_1 = 0.64$ in Richardson extrapolation formula

$$G = \frac{2^p g(h_1/2) - g(h_1)}{2^p - 1} = \frac{4\,g(0.32) - g(0.64)}{4 - 1} = \frac{4(0.37103\ \ ) - 0.38061}{3} = 0.36784$$

- This is an approximation of $(e^{-x})''$ with an error of h^4
  This is as accurate as the best results obtained with 8 digits

Example

- **Given the data points**

x	0	0.1	0.2	0.3	0.4
$f(x)$	0.0000	0.0819	0.1341	0.1646	0.1797

- **Compute $f'(x)$ at $x = 0$ as accurately as you can**
- **Use Richardson extrapolation and the forward finite difference**
- **Forward finite difference for x=0**

	$f(x)$	$f(x+h)$	$f(x+2h)$	$f(x+3h)$	$f(x+4h)$	$f(x+5h)$
$2hf'(x)$	-3	4	-1			

- **Richardson → once with h=0.2 and a second time with h=0.1**

$$g(0.2) = \frac{-3f(0) + 4f(0.2) - f(0.4)}{2(0.2)} = \frac{3(0) + 4(0.1341) - 0.1797}{0.4} = 0.8918$$

$$g(0.1) = \frac{-3f(0) + 4f(0.1) - f(0.2)}{2(0.1)} = \frac{-3(0) + 4(0.0819) - 0.1341}{0.2} = 0.9675$$

- **Recall that the error is in $O(h^2)$, so p=2**

$$f'(0) \approx G = \frac{2^2 g(0.1) - g(0.2)}{2^2 - 1} = \frac{4(0.9675) - 0.8918}{3} = 0.9927$$

- **This is an estimate in $O(h^4)$,**

Derivatives by interpolation

Introduction

- An effective solution for computing the derivative $f'(x)$ is to approximate $f(x)$ by an interpolation and derivate the interpolation.

- Very useful when the finite difference approximation can't be used

  – Uneven interval between points

    - They are technique to use finite difference but they have a larger error.

Polynomial Interpolant (1)

- Principle:
  - Simple
  - Fit the polynomial of degree n that pass through n+1 data points
  - $P_n\,(x) \;=\; a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$
  - Derivate the polynomial interpolation
  - Limit the degree of the polynomial to ~6 in order to avoid large oscillations (Runge Phenomena)
    - Oscillations are magnified at each differentiations
  - Interpolation is generally local
    (using only a few nearest-neighbor points)

Polynomial Interpolant (2)

- Remarks:
  - For evenly spaced data points:
    - Polynomial interpolation and finite difference approximations produce identical results.
    - In fact, the finite difference formulas are equivalent to polynomial interpolation.

Polynomial Interpolant (3)

- Polynomial interpolations
  - We have seen many techniques for polynomial interpolations: Lagrange, Newton, Neville.
  - Lagrange: $\mathrm{P_n}(\mathrm{x}) = \sum_i \mathrm{y_i}\ \mathrm{l_i}(\mathrm{x})$ with $\mathrm{l_i}(\mathrm{x}) = \prod_{j\neq i} \frac{(\mathrm{x}-\mathrm{x_j})}{(\mathrm{x_i}-\mathrm{x_j})}$
  - Newton: $\mathrm{P_k}(\mathrm{x}) = a_{n-k} + (x - x_{n-k})\ \mathrm{P_{k-1}}(\mathrm{x})$
  - Neville: $$\mathrm{P_k}[x_i \ldots x_{i+k}] = \frac{(x-x_{i+k})}{(x_i-x_{i+k})}\mathrm{P_{k-1}}[x_i \ldots x_{i+k-1}] + \frac{(x_i-x)}{(x_i-x_{i+k})}\mathrm{P_{k-1}}[x_{i+1} \ldots x_{i+k}]$$
  - None of these, is very much appropriate to find a polynomial of the form: $P_n\ (x)\ = \sum_{i=0}^{n} a_i x^i$

Polynomial Interpolant (4)

- Least Square fit with polynomials form:
- For a polynomial of degree m: $f(x) = \sum_{j=0}^{m} a_j x^j$
- The *normal equation* of the least square fit implies

$$\boldsymbol{A}\,\boldsymbol{a} = \boldsymbol{b}$$

  - **With:**

$$A_{kj} = \sum_{i=0}^{n} x_i^{j+k} \qquad b_k = \sum_{i=0}^{n} x_i^k y_i$$

  - **Or**

$$\mathbf{A} = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \ldots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \ldots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_i^{m-1} & \sum x_i^m & \sum x_i^{m+1} & \ldots & \sum x_i^{2m} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^m y_i \end{bmatrix}$$

  - **Can be solved with Choleski decomposition**

Polynomial Interpolant (5)

- Remarks:
  - **We use least square fit with m=n**
    - m is the degree of the polynomial
    - n+1 is the number of data points
  - In that particular case, the fit passes through all the data points.
    - When we discussed least square we were in the situation m<n
  - Once we have found the coefficients **a**, it's easy to differentiate the polynomial
  - **We could also use least square fit with (m<n) under fitting, if we believe the data are noisy**

Example

- **Given the data points**

x	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
$f(x)$	1.9934	2.1465	2.2129	2.1790	2.0683	1.9448	1.7655	1.5891

- **Compute $f'(0)$ and $f'(1)$ assuming the data are noisy**

- **Solution:**
- We use the least square fit with m<n
- We do it for m=2, 3, 4 and check the standard deviation (σ) to find which fit bests the data
- **Reminder:**

$$S(a_0, a_1, \ldots, a_m) = \sum_{i=0}^{n} [y_i - f(x_i)]^2$$

$$\approx \sum_i {r_i}^2$$

$$\sigma = \sqrt{\frac{S}{n-m}}$$

Example

- **Solution:**
  - **m=2**

```
Degree of polynomial ==> 2
Coefficients are:
[ 2.0261875   0.64703869 -0.70239583]
Std. deviation = 0.0360968935809
```

  - **m=3**

```
Degree of polynomial ==> 3
Coefficients are:
[ 1.99215     1.09276786 -1.55333333  0.40520833]
Std. deviation = 0.0082604082973
```

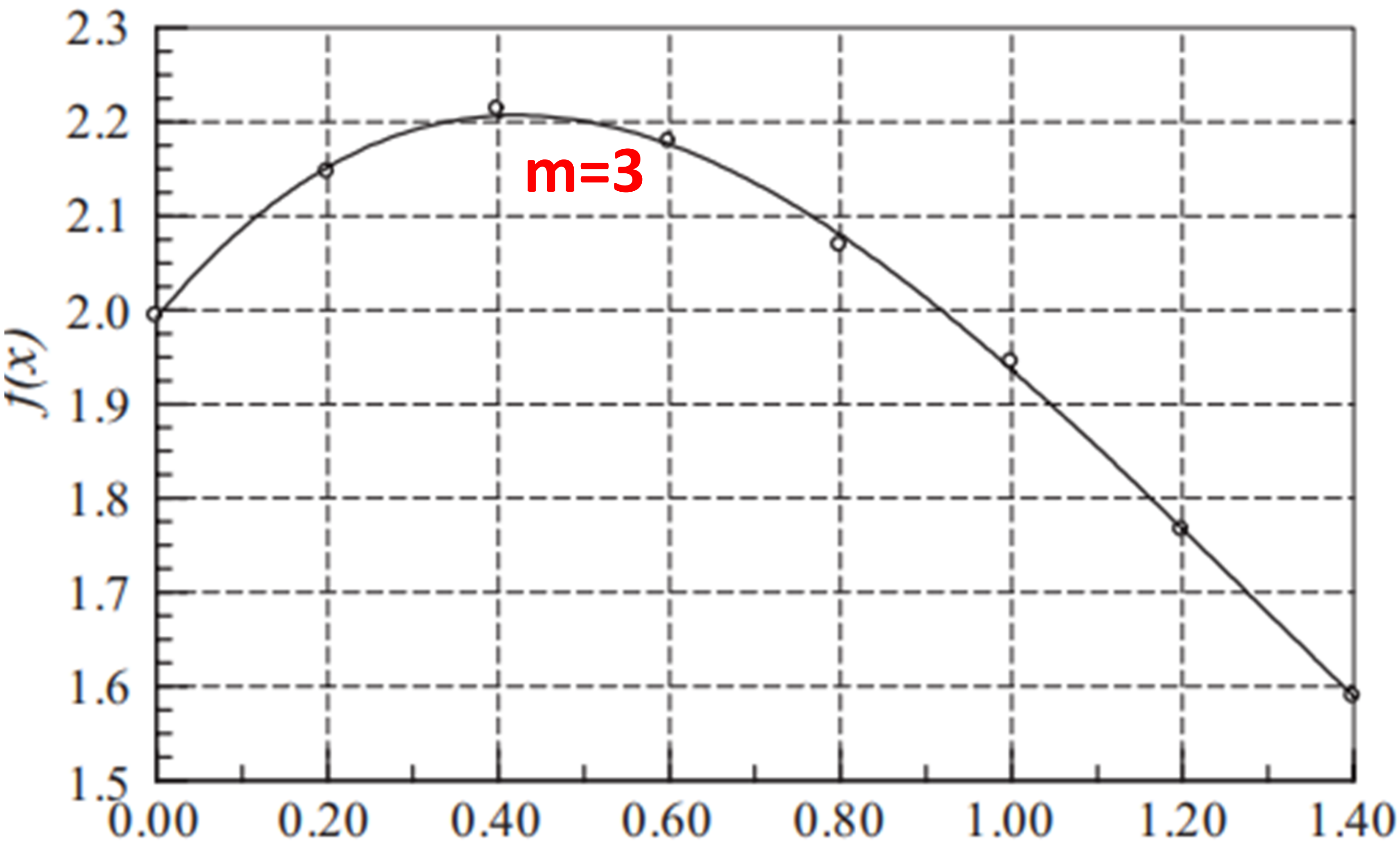
best

  - **m=4**

```
Degree of polynomial ==> 4
Coefficients are:
[ 1.99185568  1.10282373 -1.59056108  0.44812973 -0.01532907]
Std. deviation = 0.00951925073521
```

Example

- **Solution:**



Example

- **Solution:**
- **Function is :** $f(x) \approx a_0 + a_1 x + a_2 x^2 + a_3 x^3$
- **Derivative:** $f'(x) \approx a_1 + 2a_2 x + 3a_3 x^2$
- **Coeficients:**
  - a_0 = 1.99215
  - a_1 = 1.09276786
  - a_2 =-1.55333333
  - a_3 = 0.40520833

- **Results:**

$$f'(0) \approx a_1 = 1.093$$

$$f'(1) = a_1 + 2a_2 + 3a_3 = 1.093 + 2(-1.553) + 3(0.405) = -0.798$$

Cubic Spline Interpolant (1)

- Another option is to use a **cubic spline interpolation**

- Reminder: cubic spline is defined by :
- Curvature equation:
- $k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) = 6\left(\frac{y_{i-1}-y_i}{x_{i-1}-x_i} - \frac{y_i-y_{i+1}}{x_i-x_{i+1}}\right)$

- Cubic Interpolant on the segment:
- $f_{i,i+1}(x) = \frac{k_i}{6}\left[\frac{(x-x_{i+1})^3}{x_i-x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right]$
  $- \frac{k_{i+1}}{6}\left[\frac{(x-x_i)^3}{x_i-x_{i+1}} - (x - x_i)(x_i - x_{i+1})\right] + \frac{y_i(x-x_{i+1})-y_{i+1}(x-x_i)}{x_i-x_{i+1}}$

Cubic Spline Interpolant (2)

- First Derivative of the interpolant is:
- $$f'_{i,i+1}(x) = \frac{k_i}{6}\left[\frac{3(x-x_{i+1})^2}{x_i-x_{i+1}} - (x_i - x_{i+1})\right] - \frac{k_{i+1}}{6}\left[\frac{3(x-x_i)^2}{x_i-x_{i+1}} - (x_i - x_{i+1})\right] + \frac{y_i-y_{i+1}}{x_i-x_{i+1}}$$

- Second Derivative of the interpolant is:
- $$f''_{i,i+1}(x) = k_i\,\frac{x-x_{i+1}}{x_i-x_{i+1}} - k_{i+1}\,\frac{x-x_i}{x_i-x_{i+1}}$$

Example

- **Given the data points**

x	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
$f(x)$	1.9934	2.1465	2.2129	2.1790	2.0683	1.9448	1.7655	1.5891

- **Compute $f'(0)$ and $f'(1)$ assuming the data are NOT noisy**

- **Solution:**
- We use the cubic spline function to calculate the curvatures

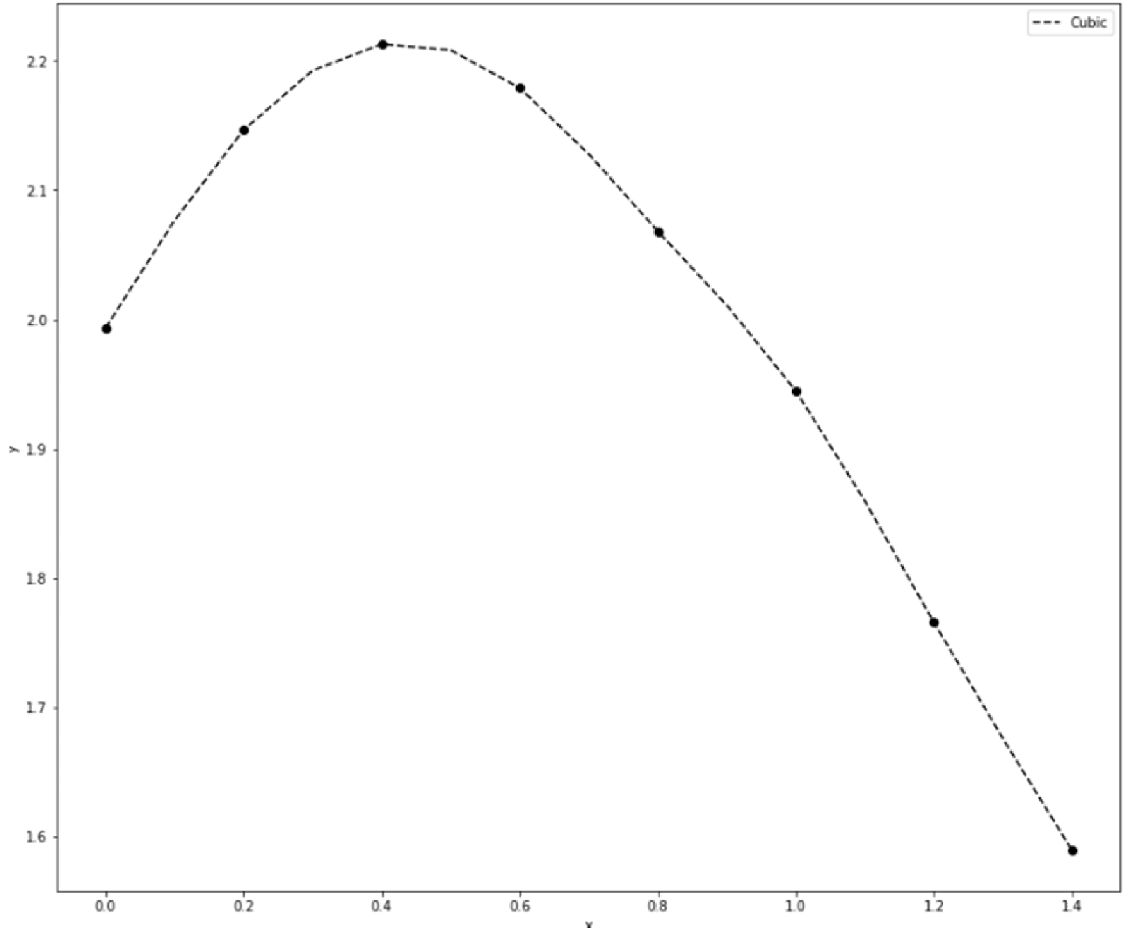
```
xData = np.array([0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4], dtype=np.float32)
yData = np.array([1.9934, 2.1465, 2.2129, 2.1790, 2.0683, 1.9448, 1.7655, 1.5891], dtype=np.float32)

k = curvatures(xData, yData)
print(k)
```

- k_0 = 0.0
- k_1 = -2.6278841
- k_2 = -2.49351027
- k_3 = -2.44304896
- k_4 = 0.74572371
- k_5 = -2.45986062
- k_6 = 0.72371581
- k_7 = 0.0

Example

- **Solution:**
- We can verify that cubic spline is a reasonable fit to the data



Example

- **Solution:**
- Derivative is given by:

$$f'_{i,i+1}(x) = \frac{k_i}{6}\left[\frac{3(x - x_{i+1})^2}{x_i - x_{i+1}} - (x_i - x_{i+1})\right] - \frac{k_{i+1}}{6}\left[\frac{3(x - x_i)^2}{x_i - x_{i+1}} - (x_i - x_{i+1})\right] + \frac{y_i - y_{i+1}}{x_i - x_{i+1}}$$

- $\boldsymbol{f'_{i,i+1}(0)}$ **→ Segment: 0,1 →** $\boldsymbol{x_i = 0}$ **and** $\boldsymbol{x_{i+1} = 0.2}$

$$f'_{i,i+1}(x) = \frac{0}{6}\left[\frac{3(0.0 - 0.2)^2}{0.0 - 0.2} - (0.0 - 0.2)\right] - \frac{-2.6279}{6}\left[\frac{3(0.0 - 0.0)^2}{0.0 - 0.2} - (0.0 - 0.2)\right] + \frac{1.9934 - 2.1465}{0.0 - 0.2}$$

$$f'_{i,i+1}(x) = 0 - \frac{-2.6279 * 0.2}{6} + \frac{1.9934 - 2.1465}{-0.2} = 0.6679$$

- $\boldsymbol{f'_{i,i+1}(1)}$ **→ Segment: 5,6 →** $\boldsymbol{x_i = 1.0}$ **and** $\boldsymbol{x_{i+1} = 1.2}$

$$f'_{i,i+1}(x) = \frac{-2.4598}{6}\left[\frac{3(1.0 - 1.2)^2}{1.0 - 1.2} - (1.0 - 1.2)\right] - \frac{0.7237}{6}\left[\frac{3(1.0 - 1.0)^2}{1.0 - 1.2} - (1.0 - 1.2)\right] + \frac{1.9448 - 1.7655}{1.0 - 1.2}$$

$$f'_{i,i+1}(x) = \frac{-2.4598 * 0.4}{6} - \frac{0.7237 * -0.2}{6} + \frac{1.9448 - 1.7655}{-0.2} = -1.0364$$