1. Introduction

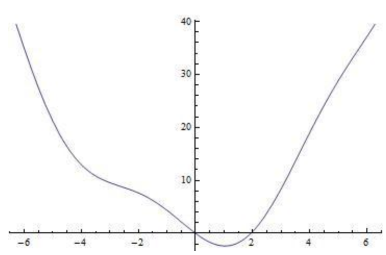
Numerical analysis deals with developing methods, called **numerical methods**, to approximate a solution of a given Mathematical problem (whenever a solution exists). The approximate solution obtained by this method will involve an error which is precisely the difference between the exact solution and the approximate solution. Thus, we have:

Exact Solution = Approximate Solution + Error.

We call this error the mathematical error. **Numerical methods** are mathematical techniques used for solving mathematical problems that cannot be solved or are difficult to solve (example: eq.1). The numerical solution is an approximate numerical value for the solution. Although numerical solutions are an approximation, they can be very accurate.

Example: Find the roots of the following equation

$$f(x) = x^2 - 4\sin(x) = 0 \quad (1)$$



- In many numerical methods, the calculations are executed in an iterative manner until a desired accuracy is achieved.

- **Example:** start at one value of x then change its value in small increment. A change in the sign of $f(x)$ indicates that there is a root within the last increment.

- Today, numerical methods are used in fast electronic digital computers that make it possible to execute many tedious and repetitive calculations that produce accurate (even though not exact) solutions in a very short time.
- For every type of mathematical problem there are several numerical techniques that can be used. The techniques differ in accuracy, length of calculations, and difficulty in programming.

2. Errors in numerical solutions

Since numerical solutions are an approximation, and since the computer program that executes the numerical method might have errors, a numerical solution needs to be examined closely. There are three major sources of error in computation: **human errors**, **truncation errors**, and **round-off errors**.

2.1 Human errors

Typical human errors are arithmetic errors, and/or programming errors: These errors can be very hard to detect unless they give obviously incorrect solution. In discussing errors, we shall assume that human errors are not present.

- Example of arithmetic errors: When parentheses or the rules about orders of operation are misunderstood or ignored:
- You can remember the correct order of operations rules which says to compute anything: inside Parentheses first, then compute Exponential expressions (powers) next, then compute Multiplications and Divisions from left to right, and finally compute Additions and Subtractions from left to right. The highest priority for parentheses means that you should follow the remaining rules for anything inside the parentheses to arrive at a result

for that part of the calculation.

2.2 Truncation Errors

Definition: Error in computation is the difference between the exact answer X_{ex} and the computed answer X_{cp}. This is also known as true error

Error = True Value - Approximate Value

- Since we are usually interested in the magnitude or absolute value of the error we define

Absolute Error =| Exact Solution - Approximate Solution |

- Note that the errors defined above cannot be determined in problems that require numerical methods for their solution. This is because the exact solution X_{ex} is not known. These error quantities are useful for evaluating the accuracy of different numerical methods when the exact solution is known (problem solved analytically).
- Since the true errors cannot, in most cases, be calculated, other means are used for estimating the accuracy of a numerical solution. For example if the numerical solution is 4.675383986896 but we do want only four digits so the answer will be: 4.6753

Where do we stop the calculation? How many terms do we include? Theoretically the calculation will never stop. If we do stop after a finite number of terms, we will not get the exact answer.

The difference between the value of the true derivative and the value that is calculated with this equation is called a truncation error. The truncation error is dependent on the specific numerical method or algorithm used to solve a problem. The truncation error is independent of round-off error.

2.3 Round-off error

Numbers can be represented in various forms. The familiar decimal system (base 10) uses ten digits 0, 1 , ..., 9. A number is written by a sequence of digits that correspond to multiples of powers of 10 can be written as, for example X_{ex} =3.262538342, if we want to use round off error with three digits thus: X_{cp}= 3.263

2.4 Relative Error Relative error (RE)—when used as a measure of precision—is the ratio of the absolute error of a measurement to the measurement being taken. In other words, this type of error is relative to the size of the item being measured. **RE** is expressed as:

As a formula, that's:

$$\textbf{Relative Error} = \frac{\textbf{Error}}{\textbf{True Value}}.$$

Example:

Find the absolute and relative errors of the approximation 125.67 to the value 119.66.

Solution:

Absolute error = |125.67-119.66|=6.01

Relative error = |125.67-119.66|/119.66 = 0.05022

3. Percentage of Errors

The percentage of RE is:

$$\textbf{Percentage Error} = 100 \times |\textbf{Relative Error}|.$$

As an example, the previous answer will be multiplied by 100 to get the percentage of the error which is:

0.05022 *100 = 5.022 %

Solution of nonlinear equation (f(x) =0)

One of the most frequently occurring problems in scientific work is to find the roots of an equation of the form

$$f(x) = 0. \qquad (1)$$

The function f(x) may be given explicitly as, for example, a polynomial or a transcendental function. Frequently, however, f(x) may be known only implicitly in that only a rule for evaluating it on any argument is known. In rare cases it may be possible to obtain the exact roots such as in the case of a factorizable polynomial. In general, however, we can hope to obtain only approximate values of the roots, relying on some computational techniques to produce the approximation. In this lecture, we will introduce some elementary iterative methods for finding a root of equation (1), in other words, a zero of f(x).

The methods are:

1- Bisection Method
2- False position Method
3- Newton-Raphson Method
4- Fixed Point Iterative Method

Bisection Technique

The first technique, based on the Intermediate Value Theorem, is called the **Bisection**, or **Binary-search, method**.

Suppose f is a continuous function defined on the interval $[a, b]$, with $f(a)$ and $f(b)$ of opposite sign. The Intermediate Value Theorem implies that a number p exists in (a, b) with $f(p) = 0$. Although the procedure will work when there is more than one root in the interval (a, b), we assume for simplicity that the root in this interval is unique. The method calls for a repeated halving (or bisecting) of subintervals of $[a, b]$ and, at each step, locating the half containing p.

To begin, set $a_1 = a$ and $b_1 = b$, and let p_1 be the midpoint of $[a, b]$; that is,

$$p_1 = a_1 + \frac{b_1 - a_1}{2} = \frac{a_1 + b_1}{2}.$$

- If $f(p_1) = 0$, then $p = p_1$, and we are done.
- If $f(p_1) \neq 0$, then $f(p_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$.
  - If $f(p_1)$ and $f(a_1)$ have the same sign, $p \in (p_1, b_1)$. Set $a_2 = p_1$ and $b_2 = b_1$.
  - If $f(p_1)$ and $f(a_1)$ have opposite signs, $p \in (a_1, p_1)$. Set $a_2 = a_1$ and $b_2 = p_1$.

ALGORITHM 2.1

Bisection

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs:

INPUT endpoints a, b; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$;
$FA = f(a)$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = a + (b - a)/2$; (*Compute* p_i.)
$FP = f(p)$.

Step 4 If $FP = 0$ or $(b - a)/2 < TOL$ then
OUTPUT (p); (*Procedure completed successfully.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 If $FA \cdot FP > 0$ then set $a = p$; (*Compute* a_i, b_i.)
$FA = FP$
else set $b = p$. (*FA is unchanged.*)

Step 7 OUTPUT ('Method failed after N_0 iterations, N_0 =', N_0);
(*The procedure was unsuccessful.*)
STOP.

Example 1 Show that $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in $[1, 2]$, and use the Bisection method to determine an approximation to the root that is accurate to at least within 10^{-4}.

Solution Because $f(1) = -5$ and $f(2) = 14$ the Intermediate Value Theorem 1.11 ensures that this continuous function has a root in $[1, 2]$.

For the first iteration of the Bisection method we use the fact that at the midpoint of $[1, 2]$ we have $f(1.5) = 2.375 > 0$. This indicates that we should select the interval $[1, 1.5]$ for our second iteration. Then we find that $f(1.25) = -1.796875$ so our new interval becomes $[1.25, 1.5]$, whose midpoint is 1.375. Continuing in this manner gives the values in Table 2.1. After 13 iterations, $p_{13} = 1.365112305$ approximates the root p with an error

$$|p - p_{13}| < |b_{14} - a_{14}| = |1.365234375 - 1.365112305| = 0.000122070.$$

Since $|a_{14}| < |p|$, we have

$$\frac{|p - p_{13}|}{|p|} < \frac{|b_{14} - a_{14}|}{|a_{14}|} \le 9.0 \times 10^{-5},$$

Table 2.1

n	a_n	b_n	p_n	$f(p_n)$
1	1.0	2.0	1.5	2.375
2	1.0	1.5	1.25	−1.79687
3	1.25	1.5	1.375	0.16211
4	1.25	1.375	1.3125	−0.84839
5	1.3125	1.375	1.34375	−0.35098
6	1.34375	1.375	1.359375	−0.09641
7	1.359375	1.375	1.3671875	0.03236
8	1.359375	1.3671875	1.36328125	−0.03215
9	1.36328125	1.3671875	1.365234375	0.000072
10	1.36328125	1.365234375	1.364257813	−0.01605
11	1.364257813	1.365234375	1.364746094	−0.00799
12	1.364746094	1.365234375	1.364990235	−0.00396
13	1.364990235	1.365234375	1.365112305	−0.00194

$$Bisection\ (f, x = [1, 2], tolerance = 0.005, output = sequence)$$

uses the Bisection method to produce the information

n	a_n	b_n	p_n	$f(p_n)$	relative error
1	1.0	2.0	1.500000000	2.37500000	0.3333333333
2	1.0	1.500000000	1.250000000	−1.796875000	0.2000000000
3	1.250000000	1.500000000	1.375000000	0.16210938	0.09090909091
4	1.250000000	1.375000000	1.312500000	−0.848388672	0.04761904762
5	1.312500000	1.375000000	1.343750000	−0.350982668	0.02325581395
6	1.343750000	1.375000000	1.359375000	−0.096408842	0.01149425287
7	1.359375000	1.375000000	1.367187500	0.03235578	0.005714285714

Bisection Method

Example : $f(x) = x^2 - 2x - 3 = 0$

initial estimeates $[x_l, x_u] = [2.0,\ 3.2]$

iter	x_l	x_u	x_r	$f(x_r)$	Δx
1	2.0	3.2	2.6	−1.44	1.2
2	2.6	3.2	2.9	−0.39	0.6
3	2.9	3.2	3.05	0.2025	0.3
4	2.9	3.05	2.975	−0.0994	0.15
5	2.975	3.05	3.0125	0.0502	0.075
6	2.975	3.0125	2.99375	−0.02496	0.0375

$f(2) = -3,\ f(3.2) = 0.84$

The following flowchart represents the method outlines

The Method of False Position

The method of (regular falsi) uses the idea that it often makes sense to assume that the function is linear locally. Instead of using the midpoint of the bracketing interval to select a new root estimate, use a weighted average:

$$w = \frac{f(b_i)a_i - f(a_i)b_i}{f(b_i) - f(a_i)}.$$

The **method of False Position** (also called *Regula Falsi*) generates approximations in the same manner as the Secant method, but it includes a test to ensure that the root is always bracketed between successive iterations. Although it is not a method we generally recommend, it illustrates how bracketing can be incorporated.

First choose initial approximations p_0 and p_1 with $f(p_0) \cdot f(p_1) < 0$. The approximation p_2 is chosen in the same manner as in the Secant method, as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_1, f(p_1))$. To decide which secant line to use to compute p_3, consider $f(p_2) \cdot f(p_1)$, or more correctly $\operatorname{sgn} f(p_2) \cdot \operatorname{sgn} f(p_1)$.

- If $\operatorname{sgn} f(p_2) \cdot \operatorname{sgn} f(p_1) < 0$, then p_1 and p_2 bracket a root. Choose p_3 as the x-intercept of the line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$.
- If not, choose p_3 as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_2, f(p_2))$, and then interchange the indices on p_0 and p_1.

In a similar manner, once p_3 is found, the sign of $f(p_3) \cdot f(p_2)$ determines whether we use p_2 and p_3 or p_3 and p_1 to compute p_4. In the latter case a relabeling of p_2 and p_1 is performed. The relabeling ensures that the root is bracketed between successive iterations. The process is described in Algorithm 2.5, and Figure 2.11 shows how the iterations can differ from those of the Secant method. In this illustration, the first three approximations are the same, but the fourth approximations differ.

ALGORITHM 2.5

False Position

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[p_0, p_1]$ where $f(p_0)$ and $f(p_1)$ have opposite signs:

INPUT initial approximations p_0, p_1; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 2$;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$.

Step 2 While $i \le N_0$ do Steps 3–7.

Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$. (*Compute* p_i.)

Step 4 If $|p - p_1| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$;
$q = f(p)$.

Step 6 If $q \cdot q_1 < 0$ then set $p_0 = p_1$;
$q_0 = q_1$.

Step 7 Set $p_1 = p$;
$q_1 = q$.

Step 8 OUTPUT ('Method failed after N_0 iterations, N_0 =', N_0);
(*The procedure unsuccessful.*)
STOP. ■

Example: Consider finding the root of $f(x) = x^2 - 3$, start with the interval [1, 2] with tolerance 0.0044.

a	b	f(a)	f(b)	c	f(c)	Update	Step Size
1.0	2.0	-2.00	1.00	1.6667	-0.2221	a = c	0.6667
1.6667	2.0	-0.2221	1.0	1.7273	-0.0164	a = c	0.0606
1.7273	2.0	-0.0164	1.0	1.7317	0.0012	a = c	0.0044

Homework

1. Use the Bisection method to find p_3 for $f(x) = \sqrt{x} - \cos x$ on $[0, 1]$.

2. Let $f(x) = 3(x+1)(x-\frac{1}{2})(x-1)$. Use the Bisection method on the following intervals to find p_3.

a. $[-2, 1.5]$ **b.** $[-1.25, 2.5]$

Chapter 5

Numerical Differentiation & Numerical integration

There are two reasons for approximating derivatives and integrals of a function f(x). One is when the function is very difficult to differentiate or integrate, or only the tabular values are available for the function. Another reason is to obtain solution of a differential or integral equation.

In section 1, we present numerical methods to find the approximated derivatives of a function. Rest of the chapter introduces various methods for numerical integration.

1- Numerical Differentiation

Numerical differentiation methods are obtained using one of the following techniques:

I. Methods based on Finite Difference Operators

II. Methods based on Interpolation (Lagrange and divided difference operator).

Through the first method, the numerical differentiation can be obtained by differentiating the Newton Gregory formula (forward or backward) then divide it by h for first derivative, h^2 for second derivative, etc.

Forward-difference: $f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$ when $h > 0$.

Backward-difference: $f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$ when $h < 0$.

We can simplify this considerably if we take k = 0, giving a derivative corresponding to $x = x_0$

$$f'(x_0) \approx \tfrac{1}{h}\left\{\Delta f_0 - \tfrac{1}{2}\Delta^2 f_0 + \tfrac{1}{3}\Delta^3 f_0 - \tfrac{1}{4}\Delta^4 f_0 + \ldots - (-1)^n \tfrac{1}{n}\Delta^n f_0\right\} \qquad (1)$$

(Same rule will be obtained for backward formula)

Examples

1. Using Newton's forward/backward differentiation method to find solution at x=0

Newton's forward differentiation table is as follows.

X	Y(X)	ΔY	$\Delta^2 Y$	$\Delta^3 Y$	$\Delta^4 Y$
0	**1**				
		-0.0025			
0.1	0.9975		**-0.005**		
		-0.0075		**0.0001**	
0.2	0.99		-0.0049		**-0.1**
		-0.0124		-0.0999	
0.3	0.9776		-0.1048		
		-0.1172			
0.4	0.8604				

The value of x at you want to find $f(x): x_0 = 0$

$h = x_1 - x_0 = 0.1 - 0 = 0.1$

$$\left[\frac{dy}{dx}\right]_{x=x_0} = \frac{1}{h} \cdot \left(\Delta Y_0 - \frac{1}{2} \cdot \Delta^2 Y_0 + \frac{1}{3} \cdot \Delta^3 Y_0 - \frac{1}{4} \cdot \Delta^4 Y_0\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=0} = \frac{1}{0.1} \cdot \left(-0.0025 - \frac{1}{2} \times -0.005 + \frac{1}{3} \times 0.0001 - \frac{1}{4} \times -0.1\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=0} = 0.25033$$

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_0} = \frac{1}{h^2} \cdot \left(\Delta^2 Y_0 - \Delta^3 Y_0 + \frac{11}{12} \cdot \Delta^4 Y_0\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=0} = \frac{1}{0.01} \cdot \left(-0.005 - 0.0001 + \frac{11}{12} \times -0.1\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=0} = -9.67667$$

Solution for $Pn'(0) = 0.25033$

Solution for $Pn''(0) = -9.67667$

Example

Use the data in the table below to estimate y'(1.7).
Use h = 0.2 and find the result using 1, 2, 3 and 4 terms of the formula.

x	y=e^x	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1.3	3.669				
		0.813			
1.5	4.482		0.179		
		0.992		0.041	
1.7	5.474		0.220		0.007
		1.212		0.048	
1.9	6.686		0.268		0.012
		1.480		0.060	
2.1	8.166		0.328		0.012
		1.808		0.072	
2.3	9.974		0.400		
		2.208			
2.5	12.182				

With one term : $y'(1.7) = \frac{1}{0.2}(1.212) = 6.060$

With two terms : $y'(1.7) = \frac{1}{0.2}(1.212 - \frac{1}{2}0.268) = 5.390$

With three terms : $y'(1.7) = \frac{1}{0.2}(1.212 - \frac{1}{2}0.268 + \frac{1}{3}0.060) = 5.490$

With four terms : $y'(1.7) = \frac{1}{0.2}(1.212 - \frac{1}{2}0.268 + \frac{1}{3}0.060 - \frac{1}{4}0.012) = 5.475$

H.W.
Use y = 1 + log x to determine y' at x = 0.15, 0.19 and 0.23 using
(a) one term, (b) two terms, (c) three terms.

Newton Backward differentiation formula

Formula

1. For $x = x_n$

$$\left[\frac{dy}{dx}\right]_{x=x_n} = \frac{1}{h}\cdot\left(\nabla Y_n + \frac{1}{2}\cdot\nabla^2 Y_n + \frac{1}{3}\cdot\nabla^3 Y_n + \frac{1}{4}\cdot\nabla^4 Y_n + ...\right)$$

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_n} = \frac{1}{h^2}\cdot\left(\nabla^2 Y_n + \nabla^3 Y_n + \frac{11}{12}\cdot\nabla^4 Y_n + ...\right)$$

2. For any value of x

$$\left[\frac{dy}{dx}\right] = \frac{1}{h}\cdot\left(\nabla Y_n + \frac{2t+1}{2}\cdot\nabla^2 Y_n + \frac{3t^2+6t+2}{6}\cdot\nabla^3 Y_n + \frac{4t^3+18t^2+22t+6}{24}\cdot\nabla^4 Y_n + ...\right)$$

$$\left[\frac{d^2y}{dx^2}\right] = \frac{1}{h^2}\cdot\left(\nabla^2 Y_n + (t+1)\cdot\nabla^3 Y_n + \frac{12t^2+36t+22}{24}\cdot\nabla^4 Y_n + ...\right)$$

Examples

1. Using Newton's Backward Difference formula to find solution at x=2.2

Newton's backward differentiation table is

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1.4	4.0552				
		0.8978			
1.6	4.953		0.1988		
		1.0966		0.0441	
1.8	6.0496		0.2429		**0.0094**
		1.3395		**0.0535**	
2	7.3891		**0.2964**		
		1.6359			
2.2	**9.025**				

$h = x_1 - x_0 = 1.6 - 1.4 = 0.2$

$$\left[\frac{dy}{dx}\right]_{x=x_n} = \frac{1}{h} \cdot \left(\nabla y_n + \frac{1}{2} \cdot \nabla^2 y_n + \frac{1}{3} \cdot \nabla^3 y_n + \frac{1}{4} \cdot \nabla^4 y_n\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=2.2} = \frac{1}{0.2} \times \left(1.6359 + \frac{1}{2} \times 0.2964 + \frac{1}{3} \times 0.0535 + \frac{1}{4} \times 0.0094\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=2.2} = 9.02142$$

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_n} = \frac{1}{h^2} \cdot \left(\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12} \cdot \nabla^4 y_n\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=2.2} = \frac{1}{0.04} \cdot \left(0.2964 + 0.0535 + \frac{11}{12} \times 0.0094\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=2.2} = 8.96292$$

$\therefore$ $Pn'(2.2) = 9.02142$ and $Pn''(2.2) = 8.96292$

First derivative by Lagrange interpolation formula

Formula

Langrange's formula

1. Find equation using Langrange's formula

$$f(x) = \frac{(x - x_1)(x - x_2)...(x - x_n)}{(x_0 - x_1)(x_0 - x_2)...(x_0 - x_n)} \times y_0 + \frac{(x - x_0)(x - x_2)...(x - x_n)}{(x_1 - x_0)(x_1 - x_2)...(x_1 - x_n)} \times y_1$$

$$+ \frac{(x - x_0)(x - x_1)(x - x_3)...(x - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)...(x_2 - x_n)} \times y_2 + ... + \frac{(x - x_0)(x - x_1)...(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)...(x_n - x_{n-1})} \times y_n$$

2. Now, differentiate f(x) with respect to x to get f'(x) and f''(x)

3. Now, substitute value of x in f'(x) and f''(x)

1. Example: Using Langrange's formula to find solution at x=5

Solution:
The value of table for x and y

x	2	4	9	10
y	4	56	711	980

Langrange's Interpolating Polynomial
Langrange's formula is

$$f(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} \times y_0 + \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} \times y_1 + \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} \times y_2 + \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \times y_3$$

$$f(x) = \frac{(x - 4)(x - 9)(x - 10)}{(2 - 4)(2 - 9)(2 - 10)} \times 4 + \frac{(x - 2)(x - 9)(x - 10)}{(4 - 2)(4 - 9)(4 - 10)} \times 56 + \frac{(x - 2)(x - 4)(x - 10)}{(9 - 2)(9 - 4)(9 - 10)} \times 711 + \frac{(x - 2)(x - 4)(x - 9)}{(10 - 2)(10 - 4)(10 - 9)} \times 980$$

$$f(x) = \frac{(x - 4)(x - 9)(x - 10)}{(-2)(-7)(-8)} \times 4 + \frac{(x - 2)(x - 9)(x - 10)}{(2)(-5)(-6)} \times 56 + \frac{(x - 2)(x - 4)(x - 10)}{(7)(5)(-1)} \times 711 + \frac{(x - 2)(x - 4)(x - 9)}{(8)(6)(1)} \times 980$$

$$f(x) = \frac{x^3 - 23x^2 + 166x - 360}{-112} \times 4 + \frac{x^3 - 21x^2 + 128x - 180}{60} \times 56 + \frac{x^3 - 16x^2 + 68x - 80}{-35} \times 711 + \frac{x^3 - 15x^2 + 62x - 72}{48} \times 980$$

$$f(x) = (x^3 - 23x^2 + 166x - 360) \times -0.0357 + (x^3 - 21x^2 + 128x - 180) \times 0.9333 + (x^3 - 16x^2 + 68x - 80) \times -20.3143 + (x^3 - 15x^2 + 62x - 72) \times 20.4167$$

$$f(x) = (- + 0.82x^2 - 5.93x + 12.86) + (0.93x^3 - 19.6x^2 + 119.47x - 168) + (-20.31x^3 + 325.03x^2 - 1381.37x + 1625.14) + (20.42x^3 - 306.25x^2 + 1265.83x - 1470)$$

$f(x) = x^3 - 2x$

$f(x) = x^3 - 2x$

Now, differentiate with x

$f'(x) = 3x^2 - 2$

$f''(x) = 6x$

Now, substitute $x = 5$

$f'(5) = 3 \times 5^2 - 2 = 73$

$f''(5) = 6 \times 5 = 30$

Remark: To compute the derivative using divided difference formula, same procedure will be followed as in Lagrange case, which means that you have to compute the function first then differentiate it.

Dr. Auras Khalid

Lecture 4

System of Equations

The most general form of a linear system is

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\cdots \\
&\cdots \\
&\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned} \tag{3.1}
$$

In the matrix notation, we can write this as

$$Ax = \mathbf{b}$$

where A is an $n \times n$ matrix with entries a_{ij}, $\mathbf{b} = (b_1, \cdots, b_n)^T$ and $\mathbf{x} = (x_1, \cdots, x_n)^T$ are n-dimensional vectors.

Theorem 3.1. *Let n be a positive integer, and let A be given as in (3.1). Then the following statements are equivalent*

I. $\det(A) \neq 0$

II. For each right hand side $\mathbf{b}$, the system (3.1) has unique solution $\mathbf{x}$.

III. For $\mathbf{b} = 0$, the only solution for the system (3.1) is the zero solution.

3.1 Gaussian Elimination

Let us introduce the **Gaussian Elimination** method for $n = 3$. The method for a general $n \times n$ system is similar.

Consider the 3×3 system

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \qquad (E1) \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \qquad (E2) \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \qquad (E3)
\end{aligned} \tag{3.2}
$$

Step 1: Assume that $a_{11} \neq 0$ (otherwise interchange the row for which the coefficient of x_1 is non-zero). Let us eliminate x_1 from (E2) and (E3). For this define

$$m_{21} = \frac{a_{21}}{a_{11}}, \quad m_{31} = \frac{a_{31}}{a_{11}}.$$

Multiply (E1) with m_{21} and subtract with (E2), and multiply (E1) with m_{31} and subtract with (E3) to give

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \qquad (E1) \\
a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 &= b_2^{(2)} \qquad (E2) \\
a_{32}^{(2)}x_2 + a_{33}^{(2)}x_3 &= b_3^{(2)} \qquad (E3)
\end{aligned}
$$

The coefficients $a_{ij}^{(2)}$ are defined by

$$
\begin{aligned}
a_{ij}^{(2)} &= a_{ij} - m_{i1}a_{1j}, \qquad i, j = 2, 3 \\
b_i^{(2)} &= b_i - m_{i1}b_1, \qquad i = 2, 3
\end{aligned}
$$

Step 2: Assume that $a_{22}^{(2)} \neq 0$ and eliminate x_2 from (E3). Define

$$m_{32} = \frac{a_{32}^{(2)}}{a_{22}^{(2)}}.$$

Subtract m_{32} times (E2) from (E3) to get

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \qquad (E1)\\ a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 &= b_2^{(2)} \qquad (E2)\\ a_{33}^{(3)}x_3 &= b_3^{(3)} \qquad (E3) \end{aligned}$$

Example 3.2. When we solve the linear system

$$\begin{aligned} 6x_1 + 2x_2 + 2x_n &= -2\\ 2x_1 + \frac{2}{3}x_2 + \frac{1}{3}x_n &= 1\\ x_1 + 2x_2 - x_n &= 0 \end{aligned}$$

Let us solve this system using Gaussian elimination method on a computer using a floating-point representation with four digits in the mantissa and all operations will be rounded.

The given system is

$$\begin{aligned} 6.000x_1 + 2.000x_2 + 2.000x_n &= -2.000\\ 2.000x_1 + 0.6667x_2 + 0.3333x_n &= 1.000\\ 1.000x_1 + 2.000x_2 - 1.000x_n &= 0.0000 \end{aligned}$$

After eliminating x_1 from the second and third equations, we get (with $m_{21} = 0.3333$, $m_{31} = 0.1667$)

$$\begin{aligned} 6.000x_1 + 2.000x_2 + 2.000x_n &= -2.000\\ 0.000x_1 + 0.0001x_2 - 0.3333x_n &= 1.667\\ 0.000x_1 + 1.667x_2 - 1.333x_n &= 0.3334 \end{aligned} \tag{3.4}$$

After eliminating x_2 from the third equation, we get (with $m_{32} = 16670$)

$$\begin{aligned} 6.000x_1 + 2.000x_2 + 2.000x_n &= -2.000\\ 0.000x_1 + 0.0001x_2 - 0.3333x_n &= 1.667\\ 0.000x_1 + 0.0000x_2 + 5555x_n &= -27790 \end{aligned}$$

Using back substitution, we get $x_1 = 1.335$, $x_2 = 0$ and $x_3 = -5.003$, whereas the actual solution is $x_1 = 2.6$, $x_2 = -3.8$ and $x_3 = -5$. The difficulty with this elimination process is that in (4.4), the element in row 2, column 2 should have been zero, but rounding error prevented it and makes the relative error very large. To avoid this, interchange row 2 and 3 in (4.4) and then continue the elimination. The final system is (with $m_{32} = 0.00005999$)

$$\begin{aligned} 6.000x_1 + 2.000x_2 + 2.000x_n &= -2.000\\ 0.000x_1 + 1.667x_2 - 1.333x_n &= 0.3334\\ 0.000x_1 + 0.0000x_2 - 0.3332x_n &= 1.667 \end{aligned}$$

with back substitution, we obtain the approximate solution as $x_1 = 2.602$, $x_2 = -3.801$ and D$x_3 = -5.003$.
□

Gauss Jordan Method

The Gauss Jordan method results in a diagonal form; for example, for a 3 × 3 system:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \rightarrow \begin{bmatrix} a'_{11} & 0 & 0 & b'_1 \\ 0 & a'_{22} & 0 & b'_2 \\ 0 & 0 & a'_{33} & b'_3 \end{bmatrix}$$

The Gauss-Jordan elimination method starts the same way that the Gauss elimination method does, but then instead of back-substitution, the elimination continues. The Gauss-Jordan method consists of:

- Creating the augmented matrix [A b]
- Forward elimination by applying EROs to get an upper triangular form
- Back elimination to a diagonal form which yields the solution

For a 2 × 2 system, this would yield

$$\begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{bmatrix} \rightarrow \begin{bmatrix} a'_{11} & 0 & b'_1 \\ 0 & a'_{22} & b'_2 \end{bmatrix}$$

and for a 3 × 3 system,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \rightarrow \begin{bmatrix} a'_{11} & 0 & 0 & b'_1 \\ 0 & a'_{22} & 0 & b'_2 \\ 0 & 0 & a'_{33} & b'_3 \end{bmatrix}$$

Notice that the resulting diagonal form does not include the right-most column.

For example, for the 2 × 2 system, forward elimination yielded the matrix:

$$\begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & 2 \end{bmatrix}$$

Now, to continue with back elimination, we need a 0 in the a_{12} position.

$$\begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & 2 \end{bmatrix} r_1 + r_2 \rightarrow r_1 \begin{bmatrix} 1 & 0 & 4 \\ 0 & -2 & 2 \end{bmatrix}$$

So, the solution is $x_1 = 4$; $-2x_2 = 2$ or $x_2 = -1$.

Here is an example of a 3×3 system:

$$\begin{aligned} x_1 + 3x_2 \quad\quad &= 1 \\ 2x_1 + x_2 + 3x_3 &= 6 \\ 4x_1 + 2x_2 + 3x_3 &= 3 \end{aligned}$$

In matrix form, the augmented matrix [A|b] is

$$\begin{bmatrix} 1 & 3 & 0 & 1 \\ 2 & 1 & 3 & 6 \\ 4 & 2 & 3 & 3 \end{bmatrix}$$

Forward substitution (done systematically by first getting a 0 in the a_{21} position, then a_{31}, and finally a_{32}):

$$\begin{bmatrix} 1 & 3 & 0 & 1 \\ 2 & 1 & 3 & 6 \\ 4 & 2 & 3 & 3 \end{bmatrix} r_2 - 2r_1 \to r_2 \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & -5 & 3 & 4 \\ 4 & 2 & 3 & 3 \end{bmatrix} r_3 - 4r_1 \to r_3 \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & -5 & 3 & 4 \\ 0 & -10 & 3 & -1 \end{bmatrix}$$

So

$$r_1 + \tfrac{3}{5} r_2 \to r_1 \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & -5 & 0 & -5 \\ 0 & 0 & -3 & -9 \end{bmatrix} \qquad \begin{aligned} x_1 &= -2 \\ -5x_2 &= -5 \\ x_2 &= 1 \\ -3x_3 &= -9 \\ x_3 &= 3 \end{aligned}$$

3.2 LU Factorization Method

Let $Ax = b$ denote the system to be solved with A the $n \times n$ coefficient matrix. In the Gaussian elimination, the linear system was reduced to the upper triangular system $Ux = g$ with

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ . & & \cdots & . \\ . & & \cdots & . \\ . & & \cdots & . \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

and $u_{ij} = a_{ij}^{(i)}$. Introduce an auxiliary lower triangular matrix L based on the multipliers m_{ij} as

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ . & & \cdots & . \\ . & & \cdots & . \\ . & & \cdots & . \\ m_{n1} & \cdots & m_{nn-1} & 1 \end{bmatrix}$$

The relationship of the matrices L and U to the original matrix A is given by the following theorem.

Theorem 3.3. *Let A be a non-singular matrix, and let L and U be defined as above. Then if U is produced without pivoting as in the Gaussian elimimation, then*

$$LU = A$$

and this is called the LU factorization of A.

LU factorization leads to another perspective on Gaussian elimination. Since $LU = A$, the linear system $A\boldsymbol{x} = \boldsymbol{b}$ can be re-written as

$$LU\boldsymbol{x} = \boldsymbol{b}.$$

And this is equivalent to solving the two systems

$$L\boldsymbol{g} = \boldsymbol{b}, \quad U\boldsymbol{x} = \boldsymbol{g} \tag{3.6}$$

The first system is the lower tirangular system

$$\begin{aligned} g_1 &= b_1 \\ m_{21}g_1 + g_2 &= b_2 \\ &. \\ &. \\ &. \\ m_{n1}g_1 + m_{n2}g_2 + \cdots + m_{nn-1}g_{n-1} + g_n &= b_n \end{aligned}$$

Once $\boldsymbol{g}$ is obtained by forward substitution from this system the upper triangular system $U\boldsymbol{x} = \boldsymbol{g}$ can be solved using back substitution. Thus once the factorization $A = LU$ is done, the solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$ is reduced to solving two triangular systems where the computational cost is reduced drastically in the situation when the system is to be solved for a fixed A but for various $\boldsymbol{b}$.

Rather than constructing L and U by using the elimination steps, it is possible to solve directly for these matrices. Let us illustrate the direct computation of L and U in the case of $n = 3$. Write $A = LU$ as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \tag{3.7}$$

The right hand matrix multiplication implies

$$\begin{aligned} a_{11} = u_{11}, a_{12} = u_{12}, a_{13} = u_{13}, \\ a_{21} = m_{21}u_{11}, a_{31} = m_{31}u_{11}. \end{aligned} \tag{3.8}$$

These gives first column of L and the first row of U. Next multiply row 2 of L times columns 2 and 3 of U, to obtain

$$a_{22} = m_{21}u_{12} + u_{22}, \quad a_{23} = m_{21}u_{13} + u_{23} \tag{3.9}$$

These can be solved for u_{22} and u_{23}. Next multiply row 3 of L to obtain

$$m_{31}u_{12} + m_{32}u_{22} = a_{32}, \quad m_{31}u_{13} + m_{32}u_{23} + u_{33} = a_{33} \tag{3.10}$$

These equations yield values for m_{32} and u_{33}, completing the construction of L and U. In this process, we must have $u_{11} \neq 0$, $u_{22} \neq 0$ in order to solve for L.

Note that in general the diagonal elements of L need not be 1. The above procedure of LU decomposition is called **Doolittle's method.**

Example 3.4. Let

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{bmatrix}$$

Using (3.8), we get

$$u_{11} = 1, \; u_{12} = 1, \;\; u_{13} = -1, \;\; m_{21} = \frac{a_{21}}{u_{11}} = 1, m_{31} = \frac{a_{31}}{u_{11}} = -2$$

Using (3.9) and (3.10),

$$\begin{aligned} u_{22} &= a_{22} - m_{21}u_{12} = 2 - 1 \times 1 = 1 \\ u_{23} &= a_{23} - m_{21}u_{13} = -2 - 1 \times (-1) = -1 \\ m_{32} &= (a_{32} - m_{31}u_{12})/u_{22} = (1 - (-2) \times 1)/1 = 3 \\ u_{33} &= a_{33} - m_{31}u_{13} - m_{32}u_{23} = 1 - (-2) \times (-1) - 3 \times (-1) = 2 \end{aligned}$$

Thus,

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

Taking $\boldsymbol{b} = (1, 1, 1)$, we now solve the system $A\boldsymbol{x} = \boldsymbol{b}$ using LU factorization, with the matrix A given above. As discussed above, first we have to solve the lower triangular system

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 3 & 1 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Forward substitution yields $g_1 = 1, g_2 = 0, g_3 = 3$. Keeping the vector $\boldsymbol{g} = (1, 0, 3)$ as the right hand side, we now solve the upper triangular system

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}.$$

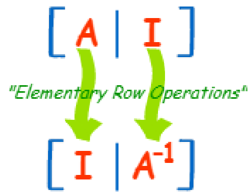
Backward substitution yields $x_1 = 1, x_2 = 3/2, x_3 = 3/2$.. □

Inverse of a Matrix using Elementary Row Operations

Also called the Gauss-Jordan method.

This is a fun way to find the Inverse of a Matrix:

Play around with the rows (adding, multiplying or swapping) until we make Matrix **A** into the Identity Matrix **I**



And by ALSO doing the changes to an Identity Matrix it magically turns into the Inverse!

The **"Elementary Row Operations"** are simple things like adding rows, multiplying and swapping ... but let's see with an example:

Example: find the Inverse of "A":

$$A = \begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix}$$

We start with the matrix A, and write it down with an Identity Matrix I next to it:

$$\left[\begin{array}{ccc|ccc} 3 & 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

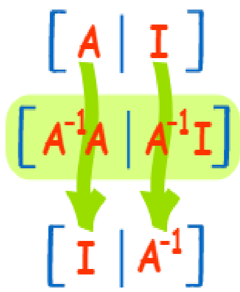
(This is called the "Augmented Matrix")

Or, more technically:

The **total effect of all the row operations** is the same as **multiplying by** $\mathbf{A}^{-1}$

So **A** becomes I (because $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$)

And I becomes $\mathbf{A}^{-1}$ (because $\mathbf{A}^{-1}\mathbf{I} = \mathbf{A}^{-1}$)



And we must do it to the **whole row**, like this:

Start with **A** next to **I**

$$\left[\begin{array}{ccc|ccc} 3 & 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right] \quad \swarrow A \quad \swarrow I$$

Add row 2 to row 1, (Add)

$$\left[\begin{array}{ccc|ccc} 5 & 0 & 0 & 1 & 1 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

then divide row 1 by 5, (Divide by 5)

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.2 & 0.2 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Then take 2 times the first row, and subtract it from the second row, (Subtract x 2)

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.2 & 0.2 & 0 \\ 0 & 0 & -2 & -0.4 & 0.6 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Multiply second row by -1/2, (Multiply by $-\frac{1}{2}$)

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.2 & 0.2 & 0 \\ 0 & 0 & 1 & 0.2 & -0.3 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Now swap the second and third row, (Swap)

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.2 & 0.2 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0.2 & -0.3 & 0 \end{array}\right]$$

Last, subtract the third row from the second row, (Subtract)

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.2 & 0.2 & 0 \\ 0 & 1 & 0 & -0.2 & 0.3 & 1 \\ 0 & 0 & 1 & 0.2 & -0.3 & 0 \end{array}\right] \quad I \nearrow \quad A^{-1} \nearrow$$

And we are done!

And matrix **A** has been made into an Identity Matrix …

… and at the same time an Identity Matrix got made into $\mathbf{A}^{-1}$

$$A^{-1} = \begin{bmatrix} 0.2 & 0.2 & 0 \\ -0.2 & 0.3 & 1 \\ 0.2 & -0.3 & 0 \end{bmatrix}$$

$$A \mid I = \left(\begin{array}{ccc|ccc} 4.0 & 5.0 & -2.0 & 1.0 & 0.0 & 0.0 \\ 7.0 & -1.0 & 2.0 & 0.0 & 1.0 & 0.0 \\ 3.0 & 1.0 & 4.0 & 0.0 & 0.0 & 1.0 \end{array}\right)$$

↓

GAUSS- JORDAN

↓

$$I \mid A^{-1} = \left(\begin{array}{ccc|ccc} 1.0 & 0.0 & 0.0 & 0.03896 & 0.14285 & -0.05194 \\ 0.0 & 1.0 & 0.0 & 0.14285 & -0.14285 & 0.14286 \\ 0.0 & 0.0 & 1.0 & -0.0649 & -0.07143 & 0.25324 \end{array}\right)$$

ITERATIVE METHODS

1- Jacobi Iterative method

2- Gauss-Seidel Iterative Method

3.5 Iterative Methods

The $n \times n$ linear system can also be solved using iterative procedures. The most fundamental iterative method is the Jacobi iterative method, which we will explain in the case of 3×3 system of linear equations.

Consider the 3×3 system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

When the diagonal elements of this system are non-zero, we can rewrite the above equation as

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3) \\ x_2 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3) \\ x_3 &= \frac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2) \end{aligned}$$

Let $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})$ be an initial guess to the true solution x, then define an iteration sequence:

$$\begin{aligned} x_1^{(m+1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(m)} - a_{13}x_3^{(m)}) \\ x_2^{(m+1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(m)} - a_{23}x_3^{(m)}) \\ x_3^{(m+1)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(m)} - a_{32}x_2^{(m)}) \end{aligned}$$

for $m = 0, 1, 2, \cdots$. This is called the **Jacobi Iteration method.**

A modified version of Jacobi method is the **Gauss-Seidel method** and is given by

$$\begin{aligned} x_1^{(m+1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(m)} - a_{13}x_3^{(m)}) \\ x_2^{(m+1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(m+1)} - a_{23}x_3^{(m)}) \\ x_3^{(m+1)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(m+1)} - a_{32}x_2^{(m+1)}) \end{aligned}$$

In the case of Jacobi method, we have

$$x_i^{(m+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(m)}\right), \quad i = 1, \cdots, n \quad m \geq 0$$

The Gauss-Seidal method reads

$$x_i^{(m+1)} = \frac{1}{a_{ii}}\left\{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(m)}\right\}, \quad i = 1, 2, \cdots, n.$$

Example 3.1. Consider the system

$$\begin{bmatrix} 3 & 2 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

The solution is $\vec{x} = (x_1, x_2)^T = (1, 1)^T$.
Jacobi's Iteration: Let the initial guess be $x_1^{(0)} = x_2^{(0)} = 0$.

$$\begin{aligned}
k &= 1, \quad 3x_1 + 2x_2 = 5 \\
&\quad x_1^{(1)} = (5 - 2x_2^{(0)})/3 = (5 - 2\cdot 0)/3 = \tfrac{5}{3} \\
&\quad x_1 + 5x_2 = 6 \\
&\quad x_2^{(1)} = (6 - x_1^{(0)})/5 = (6 - 0)/5 = \tfrac{6}{5} \\
k &= 2, \quad x_1^{(2)} = (5 - 2x_2^{(1)})/3 = (5 - 2\cdot \tfrac{6}{5})/3 = \tfrac{13}{15} \\
&\quad x_2^{(2)} = (6 - x_1^{(1)})/5 = (6 - \tfrac{5}{3})/5 = \tfrac{13}{15} \\
k &= 3, \quad x_1^{(3)} = (5 - 2x_2^{(2)})/3 = (5 - 2\cdot \tfrac{13}{15})/3 = \tfrac{49}{45} \\
&\quad x_2^{(3)} = (6 - x_1^{(2)})/5 = (6 - \tfrac{13}{15})/5 = \tfrac{77}{75}
\end{aligned}$$

Table 3.1. Jacobi's Iteration

k	0	1	2	3	$\cdots$	∞
$x_1^{(k)}$	0	$\frac{5}{3}$	$\frac{13}{15}$	$\frac{49}{45}$	$\cdots$	1
$x_2^{(k)}$	0	$\frac{6}{5}$	$\frac{13}{15}$	$\frac{77}{75}$	$\cdots$	1

Example 1: Solving a system of equations by the Gauss-Seidel method

Use the Gauss-Seidel method to solve the system

$$\begin{cases} 4x_1 + x_2 - x_3 = 3 \\ 2x_1 + 7x_2 + x_3 = 19 \\ x_1 - 3x_2 + 12x_3 = 31 \end{cases} \iff \begin{cases} x_1 = -1/4\,x_2 + 1/4\,x_3 + 3/4 \\ x_2 = -2/7\,x_1 - 1/7\,x_3 + 19/7 \\ x_3 = -1/12\,x_1 + 1/4\,x_2 + 31/12 \end{cases}$$

The difference between the Gauss-Seidel method and the Jacobi method is that here we use the coordinates $x_1^{(k)},\ldots,x_{i-1}^{(k)}$ of $x^{(k)}$ already known to compute its ith coordinate $x_i^{(k)}$.

If we start from $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$ and apply the iteration formulas, we obtain

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0	0	0
1	0,75	2,50	3,15
2	0,91	**2,00**	3,01
3	1,00	2,00	3,00
4	1,00	2,00	3,00

The exact solution is: $x_1 = 1, x_2 = 2, x_3 = 3$.

Homework

Use Gaussian elimination method (both with and without pivoting) to find the solution of the following systems:
(i) $6x_1 + 2x_2 + 2x_3 = -2,\ 2x_1 + 0.6667x_2 + 0.3333x_3 = 1,\ x_1 + 2x_2 - x_3 = 0$
(ii) $0.729x_1 + 0.81x_2 + 0.9x_3 = 0.6867,\ x_1 + x_2 + x_3 = 0.8338,\ 1.331x_1 + 1.21x_2 + 1.1x_3 = 1$

Study the convergence of the Jacobi and the Gauss-Seidel method for the following systems by starting with $x_0 = (0, 0, 0)^T$ and performing three iterations:
(i)$5x_1 + 2x_2 + x_3 = 0.12,\ 1.75x_1 + 7x_2 + 0.5x_3 = 0.1,\ x_1 + 0.2x_2 + 4.5x_3 = 0.5.$
(ii)$x_1 - 2x_2 + 2x_3 = 1,\ -x_1 + x_2 - x_3 = 1,\ -2x_1 - 2x_2 + x_3 = 1.$

Numerical Integration

In analysis, **numerical integration** comprises a family of algorithms for calculating the numerical value of a definite integral, and by extension, the term is also sometimes used to describe the numerical solution of differential equations.

In mathematics, and more specifically in numerical analysis, the **trapezoidal rule** (also known as the **trapezoid rule** or **trapezium rule** is a technique for approximating the definite integral.

Trapezoidal Rule Formula

Let f(x) be a continuous function on the interval [a, b]. Now divide the intervals [a, b] into n equal subintervals with each of width,

Δx = (b-a)/n, Such that $a = x_0 < x_1 < x_2 < x_3 < \ldots < x_n = b$

Then the Trapezoidal Rule formula for area approximating the definite integral $\int_a^b f(x)dx$ is given by:

$$\int_a^b f(x)dx \approx T_n = \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \ldots 2f(x_{n-1}) + f(x_n)]$$

Where, $x_i = a + i\Delta x$

If $n \to \infty$, R.H.S of the expression approaches the definite integral $\int_a^b f(x)dx$

Solved Examples

Go through the below given Trapezoidal Rule example.

Example 1:

Approximate the area under the curve y = f(x) between x =0 and x=8 using Trapezoidal Rule with n = 4 subintervals. A function f(x) is given in the table of values.

x	0	2	4	6	8
f(x)	3	7	11	9	3

Solution:

The Trapezoidal Rule formula for n= 4 subintervals is given as:

$T_4 = (\Delta x/2)[f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + f(x_4)]$

Here the subinterval width Δx = 2.

Now, substitute the values from the table, to find the approximate value of the area under the curve.

$A \approx T_4 = (2/2)[3 + 2(7) + 2(11) + 2(9) + 3]$

$A \approx T_4 = 3 + 14 + 22 + 18 + 3 = 60$

Therefore, the approximate value of area under the curve using Trapezoidal Rule is 60.

Example 2:

Approximate the area under the curve y = f(x) between x =-4 and x= 2 using Trapezoidal Rule with n = 6 subintervals. A function f(x) is given in the table of values.

x	-4	-3	-2	-1	0	1	2
f(x)	0	4	5	3	10	11	2

Solution:

The Trapezoidal Rule formula for n= 6 subintervals is given as:

$T_6 = (\Delta x/2)[f(x_0)+ 2f(x_1)+ 2f(x_2)+2f(x_3) + 2f(x_4)+2f(x_5)+ f(x_6)]$

Here the subinterval width $\Delta x = 1$.

Now, substitute the values from the table, to find the approximate value of the area under the curve.

$A \approx T_6 = (1/2)[0+ 2(4)+ 2(5)+2(3) + 2(10)+2(11) +2]$

$A \approx T_6 = (½) [8 + 10 + 6+ 20 +22 +2] = 68/2 = 34$

Therefore, the approximate value of area under the curve using Trapezoidal Rule is 34.

In numerical integration, **Simpson's rules** are several approximations for definite integrals, named after Thomas Simpson (1710–1761).

The most basic of these rules, called **Simpson's 1/3 rule**, or just **Simpson's rule**, reads

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right].$$

In German and some other languages, it is named after Johannes Kepler who derived it in 1615 after seeing it used for wine barrels (barrel rule, *Keplersche Fassregel*). The approximate equality in the rule becomes exact if *f* is a polynomial up to quadratic degree.

If the 1/3 rule is applied to *n* equal subdivisions of the integration range [*a*,*b*], one obtains the **composite Simpson's rule**. Points inside the integration range are given alternating weights 4/3 and 2/3.

Simpson's 3/8 rule, also called **Simpson's second rule** requests one more function evaluation inside the integration range, and is exact if *f* is a polynomial up to cubic degree.

$$I = \int_{x_0}^{x_3} f_n(x)\,dx$$

$$I = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]$$

where ξ is some number between a and b. Thus, the 3/8 rule is about twice as accurate as the standard method, but it uses one more function value. A composite 3/8 rule also exists, similarly as above.[4]

A further generalization of this concept for interpolation with arbitrary-degree polynomials are the Newton–Cotes formulas.

Composite Simpson's 3/8 rule [edit]

Dividing the interval $[a, b]$ into n subintervals of length $h = (b-a)/n$ and introducing the nodes $x_i = a + ih$ we have

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + 3f(x_5) + 2f(x_6) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)\right].$$

$$= \frac{3h}{8}\left[f(x_0) + 3\sum_{i \neq 3k}^{n-1} f(x_i) + 2\sum_{j=1}^{n/3-1} f(x_{3j}) + f(x_n)\right] \qquad \text{For: } k \in \mathbb{N}_0$$

While the remainder for the rule is shown as:

$$-\frac{h^4}{80}(b-a)f^{(4)}(\xi),$$ [4]

We can only use this if n is a multiple of three.

Example using Simpson's Rule

Approximate $\int_2^3 \frac{dx}{x+1}$ using Simpson's Rule with $n = 4$.

We haven't seen how to integrate this using algebraic processes yet, but we can use Simpson's Rule to get a good approximation for the value.

$$\Delta x = \frac{b-a}{n} = \frac{3-2}{4} = 0.25$$

$$y_0 = f(a)$$
$$= f(2)$$
$$= \frac{1}{2+1} = 0.3333333$$

$$y_1 = f(a+\Delta x) = f(2.25) = \frac{1}{2.25+1} = 0.3076923$$

$$y_2 = f(a+2\Delta x) = f(2.5) = \frac{1}{2.5+1} = 0.2857142$$

$$y_3 = f(a+3\Delta x) = f(2.75) = \frac{1}{2.75+1} = 0.2666667$$

$$y_4 = f(b) = f(3) = \frac{1}{3+1} = 0.25$$

So

$$\text{Area} = \int_a^b f(x)\,dx$$
$$\approx \frac{0.25}{3}(0.333333 + 4(0.3076923) + 2(0.2857142) + 4(0.2666667) + 0.25)$$
$$= 0.2876831$$

Example 1.

Use Simpson's Rule with $n = 4$ to approximate the integral $\int_0^8 \sqrt{x}dx$.

Solution.

It is easy to see that the width of each subinterval is

$$\Delta x = \frac{b-a}{n} = \frac{8-0}{4} = 2,$$

and the endpoints x_i have coordinates

$$x_i = \{0, 2, 4, 6, 8\}.$$

Calculate the function values at the points x_i :

$$\int_0^8 \sqrt{x}dx = \int_0^8 x^{\frac{1}{2}}dx = \left[\frac{x^{\frac{3}{2}}}{\frac{3}{2}}\right]_0^8 = \frac{2}{3}\left[\sqrt{x^3}\right]_0^8 = \frac{2}{3}\sqrt{8^3} = \frac{2}{3}\sqrt{2^9} = \frac{2}{3}\cdot 16\sqrt{2}$$
$$= \frac{32\sqrt{2}}{3} \approx 15.08$$

Hence, the error in approximating the integral is

$$|\varepsilon| = \left|\frac{15.08 - 14.86}{15.08}\right| \approx 0.015 = 1.5\%$$

$$f(x_0) = f(0) = \sqrt{0} = 0;$$

$$f(x_1) = f(2) = \sqrt{2};$$

$$f(x_2) = f(4) = \sqrt{4} = 2;$$

$$f(x_3) = f(6) = \sqrt{6};$$

$$f(x_4) = f(8) = \sqrt{8} = 2\sqrt{2}.$$

Substitute all these values into the Simpson's Rule formula:

$$\int_0^8 \sqrt{x}dx \approx \frac{\Delta x}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + f(x_4)]$$
$$= \frac{2}{3}[0 + 4\cdot\sqrt{2} + 2\cdot 2 + 4\cdot\sqrt{6} + 2\sqrt{2}] = \frac{2}{3}[6\sqrt{2} + 4 + 4\sqrt{6}] \approx 14.86$$

The true solution for the integral is

Simpson's 3/8 rule

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + 3f(x_5) + 2f(x_6) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)\right].$$

$$= \frac{3h}{8}\left[f(x_0) + 3\sum_{i\neq 3k}^{n-1} f(x_i) + 2\sum_{j=1}^{n/3-1} f(x_{3j}) + f(x_n)\right] \qquad \text{For: } k \in \mathbb{N}_0$$

Example

The vertical distance covered by a rocket from $x = 8$ to $x = 30$ seconds is given by

$$s = \int_8^{30}\left(2000\ln\left[\frac{140000}{140000 - 2100t}\right] - 9.8x\right)dx$$

Use Simpson 3/8 rule to find the approximate value of the integral.

Solution

$$h = \frac{b-a}{n}$$
$$= \frac{b-a}{3}$$
$$= \frac{30-8}{3}$$
$$= 7.3333$$

$$I \approx \frac{3h}{8} \times \{ f(x_0) + 3\,f(x_1) + 3\,f(x_2) + f(x_3) \}$$

$$x_0 = 8$$

$$f(x_0) = 2000\ln\left(\frac{140000}{140000 - 2100\times 8}\right) - 9.8\times 8$$
$$= 177.2667$$

$$\begin{cases} x_1 = x_0 + h \\ \quad = 8 + 7.3333 \\ \quad = 15.3333 \\ f(x_1) = 2000\ln\left(\frac{140000}{140000 - 2100\times 15.3333}\right) - 9.8\times 15.3333 \\ \quad = 372.4629 \end{cases}$$

$$\begin{cases} x_2 = x_0 + 2h \\ \quad = 8 + 2(7.3333) \\ \quad = 22.6666 \\ f(x_2) = 2000\ln\left(\frac{140000}{140000 - 2100\times 22.6666}\right) - 9.8\times 22.6666 \\ \quad = 608.8976 \end{cases}$$

$$\begin{cases} x_3 = x_0 + 3h \\ \quad = 8 + 3(7.3333) \\ \quad = 30 \\ f(x_3) = 2000\ln\left(\frac{140000}{140000 - 2100\times 30}\right) - 9.8\times 30 \\ \quad = 901.6740 \end{cases}$$

$$I = \frac{3}{8} \times 7.3333 \times \{177.2667 + 3\times 372.4629 + 3\times 608.8976 + 901.6740\}$$
$$= 11063.3104$$

The exact answer can be computed as

$$I_{exact} = 11061.34$$

Example

4	5	6	7
1.4	1.5	1.6	1.7
2.151	2.352	2.577	2.828

Use Simpson's-3/8 rule on interval $[1.4, 1.7]$. $h = 0.1$

$$\int_{1.4}^{1.7} f(x)\,dx \approx \frac{3h}{8}\left[f_4 + 3f_5 + 3f_6 + f_7\right]$$

$$= \frac{3(0.1)}{8}\left[2.151 + 3(2.352) + 3(2.577) + 2.828\right]$$

$$= 0.741225\,.$$

Chapter 6

Ordinary differential equations

NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS

Methods used to find numerical approximations to the solutions of ordinary differential equations (ODEs). $\frac{dy}{dx} = f(x,y)$ $\quad y(x_0) = y_0\,,\ \ h\ is\ increment$

1- Taylor Series Expansion Method

If f(x) is an initially differentiable function then Taylor series expansion of f(x) at x=c

$$f(x) = f(c) + f'(c)(x-c) + \frac{f''(c)(x-c)^2}{2!} + \cdots + \frac{f^{(n)}(c)(x-c)^n}{n!}.$$

Examples

1. Find y(0.2) for $y' = x^2y - 1$, y(0) = 1, with step length 0.1 using Taylor Series method

Solution:

Given $y' = x^2y - 1, y(0) = 1, h = 0.1, y(0.2) = ?$

Here, $x_0 = 0, y_0 = 1, h = 0.1$

Differentiating successively, we get

$y' = x^2y - 1$

$y'' = 2xy + x^2y'$

$y''' = 2y + 4xy' + x^2y''$

$y'''' = 6y' + 6xy'' + x^2y'''$

Now substituting, we get

$y_0' = x_0^2y_0 - 1 = -1$

$y_0'' = 2x_0y_0 + x_0^2y_0' = 0$

$y_0''' = 2y_0 + 4x_0y_0' + x_0^2y_0'' = 2$

$y_0'''' = 6y_0' + 6x_0y_0'' + x_0^2y_0''' = -6$

Putting these values in Taylor's Series, we have

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0'''' + \ldots$$

$$= 1 + 0.1 \cdot (-1) + \frac{(0.1)^2}{2!} \cdot (0) + \frac{(0.1)^3}{3!} \cdot (2) + \frac{(0.1)^4}{4!} \cdot (-6) + \ldots$$

$$= 1 + 0.1 \cdot (-1) + \frac{(0.1)^2}{2!} \cdot (0) + \frac{(0.1)^3}{3!} \cdot (2) + \frac{(0.1)^4}{4!} \cdot (-6) + \ldots$$

$$= 1 - 0.1 + 0 + 0.00033 + 0 + \ldots$$

$$= 0.90031$$

$$\therefore y(0.1) = 0.90031$$

Again taking (x_1, y_1) in place of (x_0, y_0) and repeat the process

Now substituting, we get

$$y_1' = x_1^2 y_1 - 1 = -0.991$$

$$y_1'' = 2x_1 y_1 + x_1^2 y_1' = 0.17015$$

$$y_1''' = 2y_1 + 4x_1 y_1' + x_1^2 y_1'' = 1.40592$$

$$y_1'''' = 6y_1' + 6x_1 y_1'' + x_1^2 y_1''' = -5.82983$$

Putting these values in Taylor's Series, we have

$$y_2 = y_1 + hy_1' + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1'''' + \ldots$$

$$= 0.90031 + 0.1 \cdot (-0.991) + \frac{(0.1)^2}{2!} \cdot (0.17015) + \frac{(0.1)^3}{3!} \cdot (1.40592) + \frac{(0.1)^4}{4!} \cdot (-5.82983) + \ldots$$

$$= 0.90031 - 0.0991 + 0.00085 + 0.00023 + 0 + \ldots$$

$$= 0.80227$$

$\therefore y(0.2) = 0.80227$

---

2. Find y(0.5) for $y' = -2x - y$, y(0) = -1, with step length 0.1 using Taylor Series method

Solution:

Given $y' = -2x - y, y(0) = -1, h = 0.1, y(0.5) = ?$

Here, $x_0 = 0, y_0 = -1, h = 0.1$

Differentiating successively, we get

$y' = -2x - y$

$y'' = -2 - y'$

$y''' = -y''$

$y'''' = -y'''$

Now substituting, we get

$y_0' = -2x_0 - y_0 = 1$

$y_0'' = -2 - y_0' = -3$

$y_0''' = -y_0'' = 3$

$y_0'''' = -y_0''' = -3$

Putting these values in Taylor's Series, we have

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0'''' + \ldots$$

$$= -1 + 0.1 \cdot (1) + \frac{(0.1)^2}{2!} \cdot (-3) + \frac{(0.1)^3}{3!} \cdot (3) + \frac{(0.1)^4}{4!} \cdot (-3) + \ldots$$

$= -1 + 0.1 - 0.015 + 0.0005 + 0 + \ldots$

$= -0.91451$

Again taking (x_1, y_1) in place of (x_0, y_0) and repeat the process

Now substituting, we get

$y_1' = -2x_1 - y_1 = 0.71451$

$y_1'' = -2 - y_1' = -2.71451$

$y_1''' = -y_1'' = 2.71451$

$y_1'''' = -y_1''' = -2.71451$

Putting these values in Taylor's Series, we have

$$y_2 = y_1 + hy_1' + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1'''' + \ldots$$

$$= -0.91451 + 0.1 \cdot (0.71451) + \frac{(0.1)^2}{2!} \cdot (-2.71451) + \frac{(0.1)^3}{3!} \cdot (2.71451) + \frac{(0.1)^4}{4!} \cdot (-2.71451) + \ldots$$

$= -0.91451 + 0.07145 - 0.01357 + 0.00045 + 0 + \ldots$

$= -0.85619$

Again taking (x_2, y_2) in place of (x_1, y_1) and repeat the process

Now substituting, we get

$y_2' = -2x_2 - y_2 = 0.45619$

$y_2'' = -2 - y_2' = -2.45619$

$y_2''' = -y_2'' = 2.45619$

$y_2'''' = -y_2''' = -2.45619$

Putting these values in Taylor's Series, we have

$$y_3 = y_2 + hy_2' + \frac{h^2}{2!}y_2'' + \frac{h^3}{3!}y_2''' + \frac{h^4}{4!}y_2'''' + \ldots$$

$$= -0.85619 + 0.1 \cdot (0.45619) + \frac{(0.1)^2}{2!} \cdot (-2.45619) + \frac{(0.1)^3}{3!} \cdot (2.45619) + \frac{(0.1)^4}{4!} \cdot (-2.45619) + \ldots$$

$= -0.85619 + 0.04562 - 0.01228 + 0.00041 + 0 + \ldots$

$= -0.82246$

Again taking (x_3, y_3) in place of (x_2, y_2) and repeat the process

Now substituting, we get
$y_3' = -2x_3 - y_3 = 0.22246$

$y_3'' = -2 - y_3' = -2.22246$

$y_3''' = -y_3'' = 2.22246$

$y_3'''' = -y_3''' = -2.22246$

Putting these values in Taylor's Series, we have
$$y_4 = y_3 + hy_3' + \frac{h^2}{2!}y_3'' + \frac{h^3}{3!}y_3''' + \frac{h^4}{4!}y_3'''' + ...$$

$$= -0.82246 + 0.1 \cdot (0.22246) + \frac{(0.1)^2}{2!} \cdot (-2.22246) + \frac{(0.1)^3}{3!} \cdot (2.22246) + \frac{(0.1)^4}{4!} \cdot (-2.22246) + ...$$

$$= -0.82246 + 0.02225 - 0.01111 + 0.00037 + 0 + ...$$

$$= -0.81096$$

Again taking (x_4, y_4) in place of (x_3, y_3) and repeat the process

Now substituting, we get
$y_4' = -2x_4 - y_4 = 0.01096$

$y_4'' = -2 - y_4' = -2.01096$

$y_4''' = -y_4'' = 2.01096$

$y_4'''' = -y_4''' = -2.01096$

Putting these values in Taylor's Series, we have
$$y_5 = y_4 + hy_4' + \frac{h^2}{2!}y_4'' + \frac{h^3}{3!}y_4''' + \frac{h^4}{4!}y_4'''' + ...$$

$$= -0.81096 + 0.1 \cdot (0.01096) + \frac{(0.1)^2}{2!} \cdot (-2.01096) + \frac{(0.1)^3}{3!} \cdot (2.01096) + \frac{(0.1)^4}{4!} \cdot (-2.01096) + ...$$

$$= -0.81096 + 0.0011 - 0.01005 + 0.00034 + 0 + ...$$

$$= -0.81959$$

$$\therefore y(0.5) = -0.81959$$

2- Euler method

In mathematics and computational science, the **Euler method** (also called **forward Euler method**) is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value $y(x_0)=y_0$.

$$\textit{Euler Method}$$
$$y_{i+1} = y_i + h\ f(x_i, y_i)$$

Examples:

1. Find y(0.2) for $y' = \frac{x-y}{2}$, y(0) = 1, with step length 0.1 using Euler method

Solution:

Given $y' = \frac{x-y}{2}, y(0) = 1, h = 0.1, y(0.2) = ?$

Euler method

$y_1 = y_0 + hf(x_0, y_0) = 1 + (0.1)f(0, 1) = 1 + (0.1) \cdot (-0.5) = 1 + (-0.05) = 0.95$

$y_2 = y_1 + hf(x_1, y_1) = 0.95 + (0.1)f(0.1, 0.95) = 0.95 + (0.1) \cdot (-0.425) = 0.95 + (-0.0425) = 0.9075$

$\therefore y(0.2) = 0.9075$

---

2. Find y(0.5) for $y' = -2x - y$, y(0) = -1, with step length 0.1 using Euler method

Solution:

Given $y' = -2x - y, y(0) = -1, h = 0.1, y(0.5) = ?$

Euler method

$y_1 = y_0 + hf(x_0, y_0) = -1 + (0.1)f(0, -1) = -1 + (0.1) \cdot (1) = -1 + (0.1) = -0.9$

$y_2 = y_1 + hf(x_1, y_1) = -0.9 + (0.1)f(0.1, -0.9) = -0.9 + (0.1) \cdot (0.7) = -0.9 + (0.07) = -0.83$

$y_3 = y_2 + hf(x_2, y_2) = -0.83 + (0.1)f(0.2, -0.83) = -0.83 + (0.1) \cdot (0.43) = -0.83 + (0.043) = -0.787$

$y_4 = y_3 + hf(x_3, y_3) = -0.787 + (0.1)f(0.3, -0.787) = -0.787 + (0.1) \cdot (0.187) = -0.787 + (0.0187) = -0.7683$

$y_5 = y_4 + hf(x_4, y_4) = -0.7683 + (0.1)f(0.4, -0.7683) = -0.7683 + (0.1) \cdot (-0.0317) = -0.7683 + (-0.00317) = -0.77147$

$\therefore y(0.5) = -0.77147$

3- Runge-Kutta Second Order (Heun Method)

$$k_1 = f(x_0, y_0)$$

$$k_2 = f(x_0 + h, y_0 + k_1 h)$$

$$Yi + 1 = yi + \frac{h}{2}(k_1 + k_2)$$

Example :

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

$Use\ RK2\ to\ find\ y(1.01), y(1.02)$

Step 1:

$$K_1 = f(x_0, y_0) = (1 + y_0{}^2 + x_0{}^3) = 18.0$$

$$K_2 = f(x_0 + h, y_0 + K_1 h) = (1 + (y_0 + 0.18)^2 + (x_0 + .01)^3) = 16.6227$$

$$y_1 = y_0 + \frac{h}{2}(K_1 + K_2) = -4 + \frac{0.01}{2}(18 + 16.6227) = -3.8268$$

$$h = 0.01$$

$$f(x, y) = 1 + y^2 + x^3$$

$$x_1 = 1.01, \quad y_1 = -3.8254$$

Step 2:

$$K_1 = f(x_1, y_1) = (1 + y_1{}^2 + x_1{}^3) = 16.6746$$

$$K_2 = f(x_1 + h, y_1 + K_1 h) = (1 + (y_1 + 0.1666)^2 + (x_1 + .01)^3) = 15.4576$$

$$y_2 = y_1 + \frac{h}{2}(K_1 + K_2) = -3.8268 + \frac{0.01}{2}(16.6746 + 15.4576) = -3.6661$$

i	x_i	y_i
0	1.00	-4.0000
1	1.01	-3.8254
2	1.02	-3.6661

4-Runge-Kutta fourth order

Fourth Order Runge – Kutta Method:

$$k_1 = h\, f(x_n, y_n)$$
$$k_2 = h\, f(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2})$$
$$k_3 = h\, f(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2})$$
$$k_4 = h\, f(x_n + h, y_n + k_3)$$
$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

Example:

Consider

$$\frac{dy}{dx} = y - x^2$$

The initial condition is: $y(0) = 1$

The step size is: $h = 0.1$

The exact Solution : $y = 2 + 2x + x^2 - e^x$

The example of a single step:

$$k_1 = h\,[f(x, y)] = 0.1\, f(0, 1) = 0.1\,(1 - 0^2) = 0.1$$

$$k_2 = h\left[f\left(x + \tfrac{1}{2}\,h, y + \tfrac{1}{2}\,k_1\right)\right] = 0.1\, f(0.05, 1.05) = 0.10475$$

$$k_3 = h\left[f\left(x + \frac{1}{2}\,h, y + \frac{1}{2}\,k_2\right)\right] = 0.1\, f(0.05, 1. + k_2/2) = 0.104988$$

$$k_4 = h\,[f(x + h, y + k_3)] = 0.1\, f(0.1, 1.104988) = 0.109499$$

$$y_{n+1} = y_n + \frac{1}{6}\,[k_1 + 2k_2 + 2k_3 + k_4] = 1.104829$$

Homework: Continue to solve for y(0.5)

lecture 7

٢١/١/٢٠٢١

Finite Difference Operators

Dr. Auras Khalid

Finite Difference Operators

- **Newton’s Forward Difference Interpolation Formula**
- **Newton’s Backward Difference Interpolation Formula**

- **Lagrange’s Interpolation Formula**

- **Divided Differences**
- **Newton’s divided difference formula**

Polynomial Interpolation Using Simple Operators

Shift Operator $Ef(x) = f(x + h)$

Forward Difference Op.

$\Delta f(x) = f(x + h) - f(x)$

Backward Difference Op.

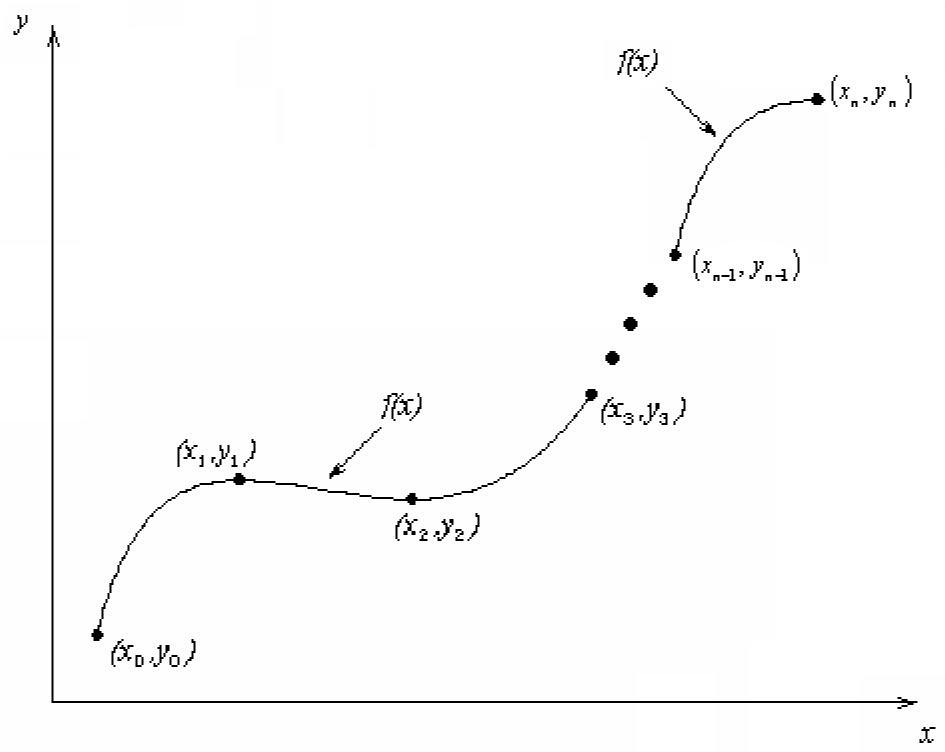
$\nabla f(x) = f(x) - f(x - h)$

Central Difference Op.

$\delta f(x) = f(x + h/2) - f(x - h/2)$

WHAT IS INTERPOLATION?

Given (x0,y0), (x1,y1), …, (xn,yn), finding the value of ‘y’ at a value of ‘x’ in (x0, xn) is called **interpolation**



NEWTON GREGORY FORWARD INTERPOLATION

For convenience we put $p = \frac{x - x_0}{h}$ and $f_0 = y_0$. Then we have

$$P(x_0 + ph) = y_o + pDy_0 + \frac{p(p-1)}{2!} D^2 y_0 + \frac{p(p-1)(p-2)}{3!} D^3 y_0 + \cdots +$$

$$\frac{p(p-1)(p-2) L (p-n+1)}{n!} D^n y_0$$

NEWTON GREGORY BACKWARD INTERPOLATION FORMULA

Taking $p = \frac{x - x_n}{h}$,

we get the interpolation formula as:

$$P(x_n + ph) = y_0 + p\nabla y_n + \frac{p(p+1)}{2!}\nabla^2 y_n + \frac{p(p+1)(p+2)}{3!}\nabla^3 y_n + \ldots\ldots + \frac{p(p+1)(p+2)\ldots(p+n-1)}{n!}\nabla^n y_n$$

Example

Estimate f (3.17)from the data using Newton Forward Interpolation.

x: 3.1 3.2 3.3 3.4 3.5

f(x): 0 0.6 1.0 1.2 1.3

Solution

First let us form the difference table

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
3.1	**0**				
		0.6			
3.2	0.6		**- 0.2**		
		0.4		**0**	
3.3	1.0		- 0.2		**0.1**
		0.2		0.1	
3.4	1.2		-0.1		
		0.1			
3.5	1.3				

Here $x_0 = 3.1$, $x = 3.17$, $h = 0.1$.

Example

Estimate f(42) from the following data using newtonbackward interpolation.

x: 20 25 30 35 40 45

f(x):354 332 291 260 231 204

Solution

The difference table is:

x	f	∇f	$\nabla^2 f$	$\nabla^3 f$	$\nabla^4 f$	$\nabla^5 f$
20	354					
		- 22				
25	332		- 19			
		- 41		29		
30	291		10		-37	
		- 31		- 8		45
35	260		2		8	
		- 29		0		
40	231		2			
		- 27				
45	204					

Here $x_n = 45$, $h = 5$, $x = 42$ and $p = -0.6$

Solution

Newton backward formula is:

$$P(x) = y_n + p\nabla y_n + \frac{p(p+1)}{2!}\nabla^2 y_n + \frac{p(p+1)(p+2)}{3!}\nabla^3 y_n + \frac{p(p+1)(p+2)(p+3)}{4!}\nabla^4 y_n + \frac{p(p+1)(p+2)(p+3)(p+4)}{5!}\nabla^5 y_n$$

$$P(42) = 204 + (-0.6)(-27) + \frac{(-0.6)(0.4)}{2} \times 2 + \frac{(-0.6)(0.40(1.4)}{6} \times 0 + \frac{(-0.6)(0.4)(1.4)(2.4)}{24} \times 8 + \frac{(-0.6)(0.4)(1.4)(2.4)(3.4)}{120} \times 45 = 219.1430$$

Thus, $f(42) = 219.143$

Chapter 7

Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points. The first degree polynomial equation is a line with slope *a*. A line will connect any two points, so a first degree polynomial equation is an exact fit through any two points with distinct x coordinates.

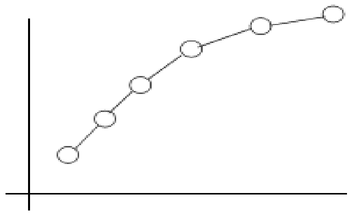
1) **Interpolation** (connect the data-dots)

If data is reliable, we can plot it and connect the dots

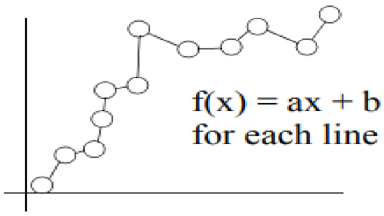
This is piece-wise, linear interpolation

This has limited use as a general function $f(x)$

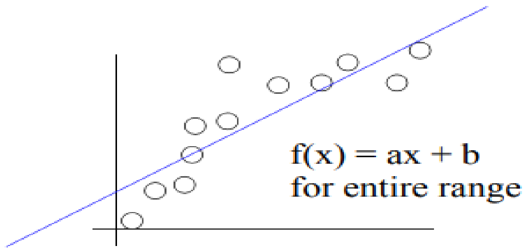
Since its really a group of small $f(x)$ s, connecting one point to the next it doesn't work very well for data that has built in random error (scatter)



2) **Curve fitting** - capturing the trend in the data by assigning a single function across the entire range. The example below uses a straight line function



Interpolation



Curve Fitting

A straight line is described generically by $f(x) = ax + b$

The goal is to identify the coefficients 'a' and 'b' such that f(x) 'fits' the data well

Linear curve fitting (linear regression)

Given the general form of a straight line

$$f(x) = ax + b$$

Solve for the a and b so that the previous two equations both = 0

re-write these two equations

$$a\sum x_i^2 + b\sum x_i = \sum (x_i y_i)$$
$$a\sum x_i + b*n = \sum y_i$$

put these into **matrix form**

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

what's unknown?

we have the data points (x_i, y_i) for $i = 1, \ldots, n$, so we have all the summation terms in the matrix

so unknows are a and b

Good news, we already know how to solve this problem

remember Gaussian elimination ??

$$A = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}, \quad X = \begin{bmatrix} b \\ a \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

so

$AX = B$

using built in Mathcad matrix inversion, the coefficients a and b are solved

```
>> X = A^-1*B
```

Note: A, B, **and** X **are not the same as** a, b, **and** x

Let's test this with an example:

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	1.5	3.0	4.5	6.0	7.5

First we find values for all the summation terms

$n = 6$

$\sum x_i = 7.5$, $\sum y_i = 22.5$, $\sum x_i^2 = 13.75$, $\sum x_i y_i = 41.25$

Now plugging into the matrix form gives us:

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix}$$

Note: we are using $\sum x_i^2$, NOT $(\sum x_i)^2$

$$\begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} * \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix}$$

or use Gaussian elimination...

The solution is $\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$ ===> $f(x) = 3x + 0$

This fits the data exactly. That is, the error is zero. Usually this is not the outcome. Usually we have data that does not exactly fit a straight line.

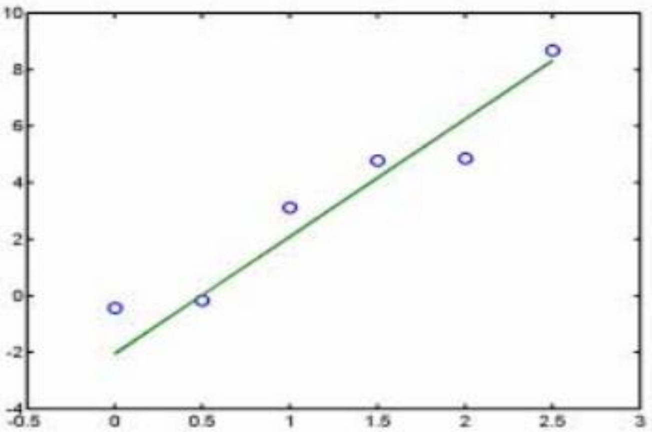
Here's an example with some 'noisy' data

x = [0 .5 1 1.5 2 2.5], y = [-0.4326 -0.1656 3.1253 4.7877 4.8535 8.6909]

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}, \quad \begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} * \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}, \quad \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.975 \\ 3.561 \end{bmatrix}$$

so our fit is $f(x) = 3.561\,x - 0.975$

Here's a plot of the data and the curve fit:



So...what do we do when a straight line is not suitable for the data set?

Polynomial Curve Fitting

Consider the general form for a polynomial of order j

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k \qquad (1)$$

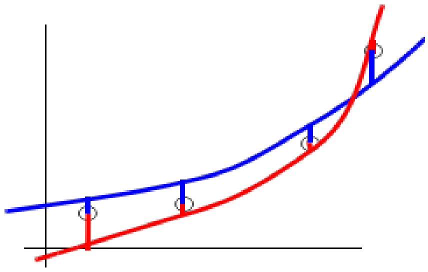
Just as was the case for linear regression, we ask:

How can we pick the coefficients that best fits the curve to the data? We can use the same idea:

The curve that gives minimum error between data y and the fit $f(x)$ is 'best'

Quantify the error for these two second order curves...

- Add up the length of all the red and blue verticle lines
- pick curve with minimum total error



re-write these $j+1$ equations, and put into matrix form

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & \ldots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \ldots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \ldots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \ldots & \sum x_i^{j+j} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum (x_i^2 y_i) \\ \vdots \\ \sum (x_i^j y_i) \end{bmatrix}$$

where all summations above are over $i = 1, \ldots, n$

we have the data points (x_i, y_i) for $i = 1, \ldots, n$

we want $a_0, a_k \quad k = 1, \ldots, j$

We already know how to solve this problem. Remember Gaussian elimination ??

$$A = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \ldots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \ldots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \ldots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \ldots & \sum x_i^{j+j} \end{bmatrix}, \quad X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum (x_i^2 y_i) \\ \vdots \\ \sum (x_i^j y_i) \end{bmatrix}$$

where all summations above are over $i = 1, \ldots, n$ data points

Note: No matter what the order j, we always get equations **LINEAR** with respect to the coefficients. This means we can use the following solution method

$$AX = B$$

using built in Mathcad matrix inversion, the coefficients a and b are solved

```
>> X = A^-1*B
```

Example #1:

Fit a second order polynomial to the following data

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	0.25	1.0	2.25	4.0	6.25

Since the order is 2 ($j = 2$), the matrix form to solve is

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

Now plug in the given data.

Before we go on...***what answers do you expect for the coefficients after looking at the data?***

$n = 6$

$\sum x_i = 7.5$, $\quad \sum y_i = 13.75$

$\sum x_i^2 = 13.75$, $\quad \sum x_i y_i = 28.125$

$\sum x_i^3 = 28.125$ $\quad \sum x_i^2 y_i = 61.1875$

$\sum x_i^4 = 61.1875$

$$\begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1875 \end{bmatrix}$$

Note: we are using $\sum x_i^2$, NOT $(\sum x_i)^2$. There's a big difference

using the inversion method $\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} * \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1857 \end{bmatrix}$

or use Gaussian elimination gives us the solution to the coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad ===> \quad f(x) = 0 + 0*x + 1*x^2$$

This fits the data exactly. That is, f(x) = y since y = x^2

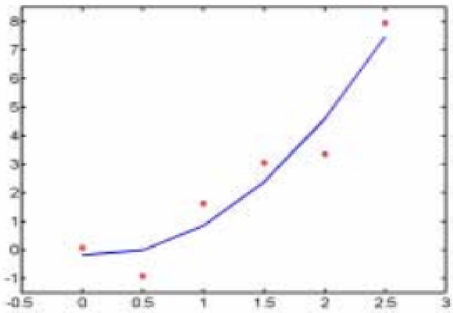
Example #2: uncertain data

Now we'll try some 'noisy' data

x = [0 .0 1 1.5 2 2.5]
y = [0.0674 -0.9156 1.6253 3.0377 3.3535 7.9409]

The resulting system to solve is:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} * \begin{bmatrix} 15.1093 \\ 32.2834 \\ 71.276 \end{bmatrix}$$



giving:
$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -0.1812 \\ -0.3221 \\ 1.3537 \end{bmatrix}$$

So our fitted second order function is:

$$f(x) = -0.1812 - 0.3221x* + 1.3537*x^2$$

Cramer's method

Find the system of Linear Equations using Cramers Rule:

2x + y + z = 3

x – y – z = 0

x + 2y + z = 0

it clear the Cramer's rule is to define the matrices A, X, Ax, Ay, and Az:

```
clc
% Cramer's method
A = [2 1 1; 1 -1 -1; 1 2 1];
X = [3; 0; 0];
Ax = [3 1 1 ; 0 -1 -1;0 2 1 ]
Ay = [2 3 1; 1 0 -1; 1 0 1]
Az = [2 1 3; 1 -1 0; 1 2 0]
x = det(Ax)/det(A)
```

```
y = det(Ay)/det(A)
z = det(Az)/det(A)

thus the answer will be:

Ax =

     3     1     1
     0    -1    -1
     0     2     1

Ay =

     2     3     1
     1     0    -1
     1     0     1

Az =

     2     1     3
     1    -1     0
     1     2     0
x =
     1
y =
    -2
z =
     3
```

NEWTONS DIVIDED DIFFERENCE

- What is divided difference?

$$f[x_0,x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

$$f[x_0,x_1,x_2] = \frac{f[x_1,x_2] - f[x_0,x_1]}{x_2 - x_1}$$

$$f[x_0, x_1, \ldots, x_{k-1},x_k] = \frac{f[x_1,x_2 - x_k] - f[x_0,\ldots,x_{k-1}]}{x_k - x_0}$$

for $k = 3, 4, \ldots n$.

These I^{st}, II^{nd}... and k^{th} order differences are denoted by Δf, $\Delta^2 f$, …, $\Delta^k f$.

- The divided difference interpolation polynomial is:

$P(x) = f(x_0) + (x - x_0)\, f[x_0, x_1] + (x - x_0)(x - x_1)\, f[x_0, x_1, x_2] + \ldots + (x - x_0)\ldots(x - x_{n-1})\, f[x_0, x_1, \ldots, x_n]$

$P(x) = f(x_0) + (x - x_0)\, f[x_0, x_1] + (x - x_0)(x - x_1)\, f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2)\, f[x_0, x_1, x_2, x_3]$

Example

- For the data

x:	−1	0	2	5
f(x) :7	10	22	235	

- Find the divided difference polynomial and estimate f(1).

Solution

X	f	Δf	$\Delta^2 f$	$\Delta^3 f$
-1	7			
		3		
0	10		1	
		6		2
2	22		13	
		71		
5	235			

$P(x) = f(x_0) + (x - x_0)\, f[x_0, x_1] + (x - x_0)(x - x_1)\, f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2)\, f[x_0, x_1, x_2, x_3]$

$= 7 + (x + 1) \times 3 + (x + 1)(x - 0) \times 1 + (x + 1)(x - 0)(x - 2) \times 2$

$= 2x^3 - x^2 + 10$

$P(1) = 11$

Use Newton's divided-difference method to compute f(2) from the experimental data shown in the following table:

x	-1.0	0.0	0.5	1.0	2.5	3.0
$y = f(x)$	3.0	-2.0	-0.375	3.0	16.125	19.0

x	$f(x)$	1st Divided Difference $f(x_0,x_1)$	2nd Divided Difference $f(x_0,x_1,x_2)$	3rd Divided Difference $f(x_0,x_1,x_2,x_3)$
−1.0	3.000			
		−5.000		
0.0	−2.000		5.500	
		3.250		−1.000
0.5	−0.375		3.500	
		6.750		−1.000
1.0	3.000		1.000	
		8.750		−1.000
2.5	16.125		−1.500	
		5.750		
3.0	19.000			

$$f(x) = f(x_0) + (x-x_0)f(x_0,x_1) + (x-x_0)(x-x_1)f(x_0,x_1,x_2) + (x-x_0)(x-x_1)(x-x_2)f(x_0,x_1,x_2,x_3)$$

$$\begin{aligned} f(2) &= -2.0 + (2-0)(3.250) + (2-0)(2-0.5)(3.500) + (2-0)(2-0.5)(2-1)(-1.000) \\ &= -2.0 + 6.5 + 10.5 - 3 \\ &= 12 \end{aligned}$$

Lagrange interpolation method

Theorem 5.1 (Lagrange Interpolation Formula).

Let $x_0, x_1, \cdots, x_n \in I = [a,b]$ be $n+1$ distinct nodes and let $f(x)$ be a continuous real-valued function defined on I. Then, there exists a unique polynomial p_n of degree $\le n$ (called **Lagrange Formula for Interpolating Polynomial***), given by*

$$p_n(x) = \sum_{k=0}^{n} f(x_k) l_k(x), \quad l_k(x) = \prod_{i=0, i\neq k}^{n} \frac{x-x_i}{x_k-x_i}, k = 0, \cdots, n \tag{5.1}$$

such that

$$p_n(x_i) = f(x_i), \quad i = 0, 1, \cdots, n. \tag{5.2}$$

The function $l_k(x)$ is called the **Lagrange multiplier**.

Lagrange Interpolation

Lagrange Interpolating takes the following general formula:

$$f_N(x) = y_0 \frac{(x-x_1)(x-x_2)\cdots(x-x_N)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_N)} + y_1 \frac{(x-x_0)(x-x_2)\cdots(x-x_N)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_N)} + \cdots + y_N \frac{(x-x_0)(x-x_1)\cdots(x-x_{N-1})}{(x_N-x_0)(x_N-x_1)\cdots(x_N-x_{N-1})}$$

Since Lagrange's interpolation is also an N^{th} degree polynomial approximation to **f(x)** and the N^{th} degree polynomial passing through **(N+1)** points is unique hence the Lagrange's and Newton's divided difference approximations are one and the same. However, Lagrange's formula is more convinent to use in computer programming and Newton's divided difference formula is more suited for hand calculations.

Example : Compute **f(0.3)** for the data

x	0	1	3	4	7
f	1	3	49	129	813

using Lagrange's interpolation formula (Analytic value is **1.831**)

$$f(x) = \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)} f_0 + \ldots + \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_0)(x_4-x_1)(x_4-x_2)(x_4-x_3)} f_4$$

$$= \frac{(0.3-1)(0.3-3)(0.3-4)(0.3-7)}{(-1)(-3)(-4)(-7)} 1 + \frac{(0.3-0)(0.3-3)(0.3-4)(0.3-7)}{1 \times (-2)(-3)(-6)} 3 +$$

$$\frac{(0.3-0)(0.3-1)(0.3-4)(0.3-7)}{3 \times 2 \times (-1)(-4)} 49 + \frac{(0.3-0)(0.3-1)(0.3-3)(0.3-7)}{4 \times 3 \times 1 (-3)} 129 +$$

$$\frac{(0.3-0)(0.3-1)(0.3-3)(0.3-4)}{7 \times 6 \times 4 \times 3} 813$$

= **1.831**

1. Find **f(2)** for the data **f(0) = 1**, **f(1) = 3** and **f(3) = 55**.

x	0	1	3
f	1	3	55

Solution :

By Newton's divided difference formula :

Divided difference table

x_i	f_i		
0	1		
		2	
1	3		8
		26	
3	55		

Now Newton's divided difference formula is

$$f(x) = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2]$$

$$f(2) = 1 + (2 - 0)\, 2 + (2 - 0)(2 - 1)\, 8$$

$$= 21$$

By Lagrange's formula :

$$f(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f_0 + \dots + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f_2$$

$$f(2) = \frac{(2 - 1)(2 - 3)}{(0 - 1)(0 - 3)} 1 + \frac{(2 - 0)(2 - 3)}{(1 - 0)(1 - 3)} 3 + \frac{(2 - 0)(2 - 1)}{(3 - 0)(3 - 1)} 55$$

$$f(2) = 21$$

2. Find f(3) for

x	0	1	2	4	5	6
f	1	14	15	5	6	19

Solution :

By Newton's divided difference formula :

Divided difference table

x_i	f_i					
0	1					
		13				
1	14		-6			
		1		1		
2	15		-2		0	
		-5		1		0
4	5		2		0	
		1		1		
5	6		6			
		13				
6	19					

Now Newton's divided difference formula is

$$f(x) = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) f[x_0, x_1, x_2, x_3]$$

$$+ (x - x_0)(x - x_1)(x - x_2)(x - x_3) f[x_0, x_1, x_2, x_3, x_4]$$

$$+ (x - x_0)(x - x_1)(x - x_2)(x - x_3)(x - x_4) f[x_0, x_1, x_2, x_3, x_4, x_5]$$

$$f(3) = 1 + (3 - 0)\,13 + (3 - 0)(3 - 1)\,{-6} + (3 - 0)(3 - 1)(3 - 2)\,1$$

$$= 10$$

By Lagrange's formula :

$$f(3) = \frac{(3-1)(3-2)(3-4)(3-5)(3-6)}{(0-1)(0-2)(0-4)(0-5)(0-6)}1 + \frac{(3-0)(3-2)(3-4)(3-5)(3-6)}{(1-0)(1-2)(1-4)(1-5)(1-6)}14 +$$

$$\frac{(3-0)(3-1)(3-4)(3-5)(3-6)}{(2-0)(2-1)(2-4)(2-5)(2-6)}15 + \frac{(3-0)(3-1)(3-2)(3-5)(3-6)}{(4-0)(4-1)(4-2)(4-5)(4-6)}5 +$$

$$\frac{(3-0)(3-1)(3-2)(3-4)(3-6)}{(5-0)(5-1)(5-2)(5-4)(5-6)}6 + \frac{(3-0)(3-1)(3-2)(3-4)(3-5)}{(6-0)(6-1)(6-2)(6-4)(6-5)}19$$

$f(2) = 10$

3. Find **f(0.25)** for

x	0.1	0.2	0.3	0.4	0.5
f	9.9833	4.9667	3.2836	2.4339	1.9177

Solution :

By Newton's divided difference formula :

Divided difference table

x_i	f_i				
0.1	9.9833				
		-50.166			
0.2	4.9667		166.675		
		-16.83		-416.68	
0.3	3.2836		41.67		833.42
		-8.497		-83.32	
0.4	2.4339		16.675		
		-5.162			
0.5	1.9177				

Now Newton's divided difference formula is

$$f(x) = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) f[x_0, x_1, x_2, x_3]$$

$$+ (x - x_0)(x - x_1)(x - x_2)(x - x_3) f[x_0, x_1, x_2, x_3, x_4]$$

f(3) = 9.9833 + (0.25 - 0.1) -50.166 + (0.25 - 0.2)(0.25 - 0.3) 166.675 +

(0.25 - 0.1) (0.25 - 0.2) (0.25 - 0.3) -416.68 + (0.25 - 0.1) (0.25 - 0.2) (0.25 - 0.3)(0.25 - 0.4) 833.42

= 3.912

By Lagrange's formula :

f(0.25) =

$$\frac{(.25-.2)(.25-.3)(.25-.4)(.25-.5)}{(.1-.2)(.1-.3)(.1-.4)(.1-.5)}9.9833+\frac{(.25-.1)(.25-.3)(.25-.4)(.25-.5)}{(.2-.1)(.2-.3)(.2-.4)(.2-.5)}4.9667+$$

$$\frac{(.25-.1)(.25-.2)(.25-.4)(.25-.5)}{(.3-.1)(.3-.2)(.3-.4)(.3-.5)}3.2836+\frac{(.25-.1)(.25-.2)(.25-.3)(.25-.5)}{(.4-.1)(.4-.2)(.4-.3)(.4-.5)}2.4339+$$

$$\frac{(.25-.1)(.25-.2)(.25-.3)(.25-.4)}{(.5-.1)(.5-.2)(.5-.3)(.5-.4)}1.9177$$

f(0.25) = 3.912

H. W.

Use a Lagrange interpolating polynomial of the first and second order to evaluate f(2) on the basis of the data:

$x_0 = 1$ $\quad f(x_0) = 0$

$x_1 = 4$ $\quad f(x_1) = 1.386294$

$x_2 = 6$ $\quad f(x_2) = 1.791760$

Chapter 5

Numerical Differentiation & Numerical integration

There are two reasons for approximating derivatives and integrals of a function f(x). One is when the function is very difficult to differentiate or integrate, or only the tabular values are available for the function. Another reason is to obtain solution of a differential or integral equation.

In section 1, we obtain numerical methods to find derivatives of a function. Rest of the chapter introduces various methods for numerical integration.

1- Numerical Differentiation

Numerical differentiation methods are obtained using one of the following techniques:

I. Methods based on Finite Difference Operators

II. Methods based on Interpolation (Lagrange and divided difference operator).

Through the first method, the numerical differentiation can be obtained by differentiating the Newton Gregory formula (forward or backward) then divide it by h for first derivative, h^2 for second derivative, etc.

Forward-difference: $f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$ when $h > 0$.

Backward-difference: $f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$ when $h < 0$.

We can simplify this considerably if we take k = 0, giving a derivative corresponding to $x = x_0$

$$f'(x_0) \approx \tfrac{1}{h}\left\{\Delta f_0 - \tfrac{1}{2}\Delta^2 f_0 + \tfrac{1}{3}\Delta^3 f_0 - \tfrac{1}{4}\Delta^4 f_0 + \ldots - (-1)^n \tfrac{1}{n}\Delta^n f_0\right\} \qquad (1)$$

(Same rule will be obtained for backward formula)

Examples

1. Using Newton's forward/backward differentiation method to find solution at x=0

Newton's forward differentiation table is as follows.

X	Y(X)	ΔY	$\Delta^2 Y$	$\Delta^3 Y$	$\Delta^4 Y$
0	**1**				
		-0.0025			
0.1	0.9975		**-0.005**		
		-0.0075		**0.0001**	
0.2	0.99		-0.0049		**-0.1**
		-0.0124		-0.0999	
0.3	0.9776		-0.1048		
		-0.1172			
0.4	0.8604				

The value of x at you want to find $f(x): x_0 = 0$

$h = x_1 - x_0 = 0.1 - 0 = 0.1$

$$\left[\frac{dy}{dx}\right]_{x=x_0} = \frac{1}{h} \cdot \left(\Delta Y_0 - \frac{1}{2} \cdot \Delta^2 Y_0 + \frac{1}{3} \cdot \Delta^3 Y_0 - \frac{1}{4} \cdot \Delta^4 Y_0\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=0} = \frac{1}{0.1} \cdot \left(-0.0025 - \frac{1}{2} \times -0.005 + \frac{1}{3} \times 0.0001 - \frac{1}{4} \times -0.1\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=0} = 0.25033$$

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_0} = \frac{1}{h^2} \cdot \left(\Delta^2 Y_0 - \Delta^3 Y_0 + \frac{11}{12} \cdot \Delta^4 Y_0\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=0} = \frac{1}{0.01} \cdot \left(-0.005 - 0.0001 + \frac{11}{12} \times -0.1\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=0} = -9.67667$$

Solution for $Pn'(0) = 0.25033$

Solution for $Pn''(0) = -9.67667$

Example

Use the data in the table below to estimate y'(1.7).
Use h = 0.2 and find the result using 1, 2, 3 and 4 terms of the formula.

x	y=e^x	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1.3	3.669				
		0.813			
1.5	4.482		0.179		
		0.992		0.041	
1.7	5.474		0.220		0.007
		1.212		0.048	
1.9	6.686		0.268		0.012
		1.480		0.060	
2.1	8.166		0.328		0.012
		1.808		0.072	
2.3	9.974		0.400		
		2.208			
2.5	12.182				

With one term : $y'(1.7) = \frac{1}{0.2}(1.212) = 6.060$

With two terms : $y'(1.7) = \frac{1}{0.2}(1.212 - \frac{1}{2}0.268) = 5.390$

With three terms : $y'(1.7) = \frac{1}{0.2}(1.212 - \frac{1}{2}0.268 + \frac{1}{3}0.060) = 5.490$

With four terms : $y'(1.7) = \frac{1}{0.2}(1.212 - \frac{1}{2}0.268 + \frac{1}{3}0.060 - \frac{1}{4}0.012) = 5.475$

H.W.
Use y = 1 + log x to determine y' at x = 0.15, 0.19 and 0.23 using
(a) one term, (b) two terms, (c) three terms.

Newton Backward differentiation formula

Formula

1. For $x = x_n$

$$\left[\frac{dy}{dx}\right]_{x=x_n} = \frac{1}{h}\cdot\left(\nabla Y_n + \frac{1}{2}\cdot\nabla^2 Y_n + \frac{1}{3}\cdot\nabla^3 Y_n + \frac{1}{4}\cdot\nabla^4 Y_n + ...\right)$$

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_n} = \frac{1}{h^2}\cdot\left(\nabla^2 Y_n + \nabla^3 Y_n + \frac{11}{12}\cdot\nabla^4 Y_n + ...\right)$$

2. For any value of x

$$\left[\frac{dy}{dx}\right] = \frac{1}{h}\cdot\left(\nabla Y_n + \frac{2t+1}{2}\cdot\nabla^2 Y_n + \frac{3t^2+6t+2}{6}\cdot\nabla^3 Y_n + \frac{4t^3+18t^2+22t+6}{24}\cdot\nabla^4 Y_n + ...\right)$$

$$\left[\frac{d^2y}{dx^2}\right] = \frac{1}{h^2}\cdot\left(\nabla^2 Y_n + (t+1)\cdot\nabla^3 Y_n + \frac{12t^2+36t+22}{24}\cdot\nabla^4 Y_n + ...\right)$$

Examples

1. Using Newton's Backward Difference formula to find solution at x=2.2

Newton's backward differentiation table is

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1.4	4.0552				
		0.8978			
1.6	4.953		0.1988		
		1.0966		0.0441	
1.8	6.0496		0.2429		**0.0094**
		1.3395		**0.0535**	
2	7.3891		**0.2964**		
		1.6359			
2.2	**9.025**				

$h = x_1 - x_0 = 1.6 - 1.4 = 0.2$

$$\left[\frac{dy}{dx}\right]_{x=x_n} = \frac{1}{h} \cdot \left(\nabla y_n + \frac{1}{2} \cdot \nabla^2 y_n + \frac{1}{3} \cdot \nabla^3 y_n + \frac{1}{4} \cdot \nabla^4 y_n\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=2.2} = \frac{1}{0.2} \times \left(1.6359 + \frac{1}{2} \times 0.2964 + \frac{1}{3} \times 0.0535 + \frac{1}{4} \times 0.0094\right)$$

$$\therefore \left[\frac{dy}{dx}\right]_{x=2.2} = 9.02142$$

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_n} = \frac{1}{h^2} \cdot \left(\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12} \cdot \nabla^4 y_n\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=2.2} = \frac{1}{0.04} \cdot \left(0.2964 + 0.0535 + \frac{11}{12} \times 0.0094\right)$$

$$\therefore \left[\frac{d^2y}{dx^2}\right]_{x=2.2} = 8.96292$$

$\therefore Pn'(2.2) = 9.02142$ and $Pn''(2.2) = 8.96292$

First derivative by Lagrange interpolation formula

Formula

Langrange's formula

1. Find equation using Langrange's formula

$$f(x) = \frac{(x-x_1)(x-x_2)...(x-x_n)}{(x_0-x_1)(x_0-x_2)...(x_0-x_n)} \times y_0 + \frac{(x-x_0)(x-x_2)...(x-x_n)}{(x_1-x_0)(x_1-x_2)...(x_1-x_n)} \times y_1$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_3)...(x-x_n)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)...(x_2-x_n)} \times y_2 + ... + \frac{(x-x_0)(x-x_1)...(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)...(x_n-x_{n-1})} \times y_n$$

2. Now, differentiate f(x) with respect to x to get f'(x) and f''(x)

3. Now, substitute value of x in f'(x) and f''(x)

1. Example: Using Langrange's formula to find solution at x=5

Solution:

The value of table for x and y

x	2	4	9	10
y	4	56	711	980

Langrange's Interpolating Polynomial

Langrange's formula is

$$f(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} \times y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \times y_1 + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} \times y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \times y_3$$

$$f(x) = \frac{(x-4)(x-9)(x-10)}{(2-4)(2-9)(2-10)} \times 4 + \frac{(x-2)(x-9)(x-10)}{(4-2)(4-9)(4-10)} \times 56 + \frac{(x-2)(x-4)(x-10)}{(9-2)(9-4)(9-10)} \times 711 + \frac{(x-2)(x-4)(x-9)}{(10-2)(10-4)(10-9)} \times 980$$

$$f(x) = \frac{(x-4)(x-9)(x-10)}{(-2)(-7)(-8)} \times 4 + \frac{(x-2)(x-9)(x-10)}{(2)(-5)(-6)} \times 56 + \frac{(x-2)(x-4)(x-10)}{(7)(5)(-1)} \times 711 + \frac{(x-2)(x-4)(x-9)}{(8)(6)(1)} \times 980$$

$$f(x) = \frac{x^3-23x^2+166x-360}{-112} \times 4 + \frac{x^3-21x^2+128x-180}{60} \times 56 + \frac{x^3-16x^2+68x-80}{-35} \times 711 + \frac{x^3-15x^2+62x-72}{48} \times 980$$

$$f(x) = (x^3-23x^2+166x-360) \times -0.0357 + (x^3-21x^2+128x-180) \times 0.9333 + (x^3-16x^2+68x-80) \times -20.3143 + (x^3-15x^2+62x-72) \times 20.4167$$

$$f(x) = (- \ +0.82x^2-5.93x+12.86) + (0.93x^3-19.6x^2+119.47x-168) + (-20.31x^3+325.03x^2-1381.37x+1625.14) + (20.42x^3-306.25x^2+1265.83x-1470)$$

$f(x) = x^3 - 2x$

$f(x) = x^3 - 2x$

Now, differentiate with x

$f'(x) = 3x^2 - 2$

$f''(x) = 6x$

Now, substitute $x = 5$

$f'(5) = 3 \times 5^2 - 2 = 73$

$f''(5) = 6 \times 5 = 30$

Remark: to compute the derivative using divided difference formula, same procedure will be followed as in Lagrange case

Lecture 10

Numerical Integration

In analysis, **numerical integration** comprises a family of algorithms for calculating the numerical value of a definite integral, and by extension, the term is also sometimes used to describe the numerical solution of differential equations.

In mathematics, and more specifically in numerical analysis, the **trapezoidal rule** (also known as the **trapezoid rule** or **trapezium rule** is a technique for approximating the definite integral.

Trapezoidal Rule Formula

Let f(x) be a continuous function on the interval [a, b]. Now divide the intervals [a, b] into n equal subintervals with each of width,

Δx = (b-a)/n, Such that $a = x_0 < x_1 < x_2 < x_3 < \ldots < x_n = b$

Then the Trapezoidal Rule formula for area approximating the definite integral $\int_a^b f(x)dx$ is given by:

$$\int_a^b f(x)dx \approx T_n = \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \ldots 2f(x_{n-1}) + f(x_n)]$$

Where, $x_i = a + i\Delta x$

If $n \to \infty$, R.H.S of the expression approaches the definite integral $\int_a^b f(x)dx$

Solved Examples

Go through the below given Trapezoidal Rule example.

Example 1:

Approximate the area under the curve y = f(x) between x =0 and x=8 using Trapezoidal Rule with n = 4 subintervals. A function f(x) is given in the table of values.

x	0	2	4	6	8
f(x)	3	7	11	9	3

Solution:

The Trapezoidal Rule formula for n= 4 subintervals is given as:

$T_4 = (\Delta x/2)[f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + f(x_4)]$

Here the subinterval width Δx = 2.

Now, substitute the values from the table, to find the approximate value of the area under the curve.

$A \approx T_4 = (2/2)[3 + 2(7) + 2(11) + 2(9) + 3]$

$A \approx T_4 = 3 + 14 + 22 + 18 + 3 = 60$

Therefore, the approximate value of area under the curve using Trapezoidal Rule is 60.

Example 2:

Approximate the area under the curve y = f(x) between x =-4 and x= 2 using Trapezoidal Rule with n = 6 subintervals. A function f(x) is given in the table of values.

x	-4	-3	-2	-1	0	1	2
f(x)	0	4	5	3	10	11	2

Solution:

The Trapezoidal Rule formula for n= 6 subintervals is given as:

$T_6 = (\Delta x/2)[f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + 2f(x_5) + f(x_6)]$

Here the subinterval width $\Delta x = 1$.

Now, substitute the values from the table, to find the approximate value of the area under the curve.

$A \approx T_6 = (1/2)[0 + 2(4) + 2(5) + 2(3) + 2(10) + 2(11) + 2]$

$A \approx T_6 = (½) [8 + 10 + 6 + 20 + 22 + 2] = 68/2 = 34$

Therefore, the approximate value of area under the curve using Trapezoidal Rule is 34.

In numerical integration, **Simpson's rules** are several approximations for definite integrals, named after Thomas Simpson (1710–1761).

The most basic of these rules, called **Simpson's 1/3 rule**, or just **Simpson's rule**, reads

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right].$$

In German and some other languages, it is named after Johannes Kepler who derived it in 1615 after seeing it used for wine barrels (barrel rule, *Keplersche Fassregel*). The approximate equality in the rule becomes exact if *f* is a polynomial up to quadratic degree.

If the 1/3 rule is applied to *n* equal subdivisions of the integration range [*a*,*b*], one obtains the **composite Simpson's rule**. Points inside the integration range are given alternating weights 4/3 and 2/3.

Simpson's 3/8 rule, also called **Simpson's second rule** requests one more function evaluation inside the integration range, and is exact if *f* is a polynomial up to cubic degree.

$$I = \int_{x_0}^{x_3} f_n(x)\,dx$$

$$I = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]$$

where ξ is some number between a and b. Thus, the 3/8 rule is about twice as accurate as the standard method, but it uses one more function value. A composite 3/8 rule also exists, similarly as above.[4]

A further generalization of this concept for interpolation with arbitrary-degree polynomials are the Newton–Cotes formulas.

Composite Simpson's 3/8 rule [edit]

Dividing the interval $[a, b]$ into n subintervals of length $h = (b-a)/n$ and introducing the nodes $x_i = a + ih$ we have

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + 3f(x_5) + 2f(x_6) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)\right].$$

$$= \frac{3h}{8}\left[f(x_0) + 3\sum_{i \neq 3k}^{n-1} f(x_i) + 2\sum_{j=1}^{n/3-1} f(x_{3j}) + f(x_n)\right] \qquad \text{For: } k \in \mathbb{N}_0$$

While the remainder for the rule is shown as:

$$-\frac{h^4}{80}(b-a)f^{(4)}(\xi),$$ [4]

We can only use this if n is a multiple of three.

Example using Simpson's Rule

Approximate $\int_2^3 \frac{dx}{x+1}$ using Simpson's Rule with $n = 4$.

We haven't seen how to integrate this using algebraic processes yet, but we can use Simpson's Rule to get a good approximation for the value.

$$\Delta x = \frac{b-a}{n} = \frac{3-2}{4} = 0.25$$

$$y_0 = f(a)$$

$$= f(2)$$

$$= \frac{1}{2+1} = 0.3333333$$

$$y_1 = f(a + \Delta x) = f(2.25) = \frac{1}{2.25+1} = 0.3076923$$

$$y_2 = f(a + 2\Delta x) = f(2.5) = \frac{1}{2.5+1} = 0.2857142$$

$$y_3 = f(a + 3\Delta x) = f(2.75) = \frac{1}{2.75+1} = 0.2666667$$

$$y_4 = f(b) = f(3) = \frac{1}{3+1} = 0.25$$

So

$$\text{Area} = \int_a^b f(x)\mathrm{d}x$$

$$\approx \frac{0.25}{3}(0.333333 + 4(0.3076923) \ +2(0.2857142) + 4(0.2666667) \ +0.25)$$

$$= 0.2876831$$

Example 1.

Use Simpson's Rule with $n = 4$ to approximate the integral $\int_0^8 \sqrt{x}dx$.

Solution.

It is easy to see that the width of each subinterval is

$$\Delta x = \frac{b-a}{n} = \frac{8-0}{4} = 2,$$

and the endpoints x_i have coordinates

$$x_i = \{0, 2, 4, 6, 8\}.$$

Calculate the function values at the points x_i :

$$\int_0^8 \sqrt{x}dx = \int_0^8 x^{\frac{1}{2}}dx = \left[\frac{x^{\frac{3}{2}}}{\frac{3}{2}}\right]_0^8 = \frac{2}{3}\left[\sqrt{x^3}\right]_0^8 = \frac{2}{3}\sqrt{8^3} = \frac{2}{3}\sqrt{2^9} = \frac{2}{3}\cdot 16\sqrt{2}$$

$$= \frac{32\sqrt{2}}{3} \approx 15.08$$

Hence, the error in approximating the integral is

$$|\varepsilon| = \left|\frac{15.08 - 14.86}{15.08}\right| \approx 0.015 = 1.5\%$$

$$f(x_0) = f(0) = \sqrt{0} = 0;$$

$$f(x_1) = f(2) = \sqrt{2};$$

$$f(x_2) = f(4) = \sqrt{4} = 2;$$

$$f(x_3) = f(6) = \sqrt{6};$$

$$f(x_4) = f(8) = \sqrt{8} = 2\sqrt{2}.$$

Substitute all these values into the Simpson's Rule formula:

$$\int_0^8 \sqrt{x}dx \approx \frac{\Delta x}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + f(x_4)]$$

$$= \frac{2}{3}[0 + 4 \cdot \sqrt{2} + 2 \cdot 2 + 4 \cdot \sqrt{6} + 2\sqrt{2}] = \frac{2}{3}[6\sqrt{2} + 4 + 4\sqrt{6}] \approx 14.86$$

The true solution for the integral is

Simpson's 3/8 rule

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + 3f(x_5) + 2f(x_6) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)].$$

$$= \frac{3h}{8}\left[f(x_0) + 3\sum_{i \neq 3k}^{n-1} f(x_i) + 2\sum_{j=1}^{n/3-1} f(x_{3j}) + f(x_n)\right] \qquad \text{For: } k \in \mathbb{N}_0$$

Example

The vertical distance covered by a rocket from $x = 8$ to $x = 30$ seconds is given by

$$s = \int_8^{30}\left(2000\ln\left[\frac{140000}{140000 - 2100t}\right] - 9.8x\right)dx$$

Use Simpson 3/8 rule to find the approximate value of the integral.

Solution

$$h=\frac{b-a}{n}$$
$$=\frac{b-a}{3}$$
$$=\frac{30-8}{3}$$
$$=7.3333$$

$$I\approx\frac{3h}{8}\times\{f(x_0)+3f(x_1)+3f(x_2)+f(x_3)\}$$

$$x_0=8$$

$$f(x_0)=2000\ln\left(\frac{140000}{140000-2100\times 8}\right)-9.8\times 8$$
$$=177.2667$$

$$\begin{cases} x_1=x_0+h \\ \quad =8+7.3333 \\ \quad =15.3333 \\ f(x_1)=2000\ln\left(\frac{140000}{140000-2100\times 15.3333}\right)-9.8\times 15.3333 \\ \quad =372.4629 \end{cases}$$

$$\begin{cases} x_2=x_0+2h \\ \quad =8+2(7.3333) \\ \quad =22.6666 \\ f(x_2)=2000\ln\left(\frac{140000}{140000-2100\times 22.6666}\right)-9.8\times 22.6666 \\ \quad =608.8976 \end{cases}$$

$$\begin{cases} x_3=x_0+3h \\ \quad =8+3(7.3333) \\ \quad =30 \\ f(x_3)=2000\ln\left(\frac{140000}{140000-2100\times 30}\right)-9.8\times 30 \\ \quad =901.6740 \end{cases}$$

$$I=\frac{3}{8}\times 7.3333\times\{177.2667+3\times 372.4629+3\times 608.8976+901.6740\}$$
$$=11063.3104$$

The exact answer can be computed as

$$I_{exact}=11061.34$$

Lecture 11

Ordinary differential equations

Numerical methods for ordinary differential equations

Methods used to find numerical approximations to the solutions of ordinary differential equations (ODEs). $\frac{dy}{dx} = f(x,y)$ $y(x_0) = y_0$, h *is increment*

1- Taylor Series Expansion Method

If f(x) is an initially differentiable function then Taylor series expansion of f(x) at x=c

$$f(x) = f(c) + f'(c)(x-c) + \frac{f''(c)(x-c)^2}{2!} + \cdots + \frac{f^{(n)}(c)(x-c)^n}{n!}.$$

Examples

1. Find y(0.2) for $y' = x^2y - 1$, y(0) = 1, with step length 0.1 using Taylor Series method

Solution:

Given $y' = x^2y - 1, y(0) = 1, h = 0.1, y(0.2) = ?$

Here, $x_0 = 0, y_0 = 1, h = 0.1$

Differentiating successively, we get

$y' = x^2y - 1$

$y'' = 2xy + x^2y'$

$y''' = 2y + 4xy' + x^2y''$

$y'''' = 6y' + 6xy'' + x^2y'''$

Now substituting, we get

$y_0' = x_0^2y_0 - 1 = -1$

$y_0'' = 2x_0y_0 + x_0^2y_0' = 0$

$y_0''' = 2y_0 + 4x_0y_0' + x_0^2y_0'' = 2$

$y_0'''' = 6y_0' + 6x_0y_0'' + x_0^2y_0''' = -6$

Putting these values in Taylor's Series, we have

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0'''' + ...$$

$$= 1 + 0.1 \cdot (-1) + \frac{(0.1)^2}{2!} \cdot (0) + \frac{(0.1)^3}{3!} \cdot (2) + \frac{(0.1)^4}{4!} \cdot (-6) + ...$$

$$= 1 + 0.1 \cdot (-1) + \frac{(0.1)^2}{2!} \cdot (0) + \frac{(0.1)^3}{3!} \cdot (2) + \frac{(0.1)^4}{4!} \cdot (-6) + ...$$

$$= 1 - 0.1 + 0 + 0.00033 + 0 + ...$$

$$= 0.90031$$

$$\therefore y(0.1) = 0.90031$$

Again taking (x_1, y_1) in place of (x_0, y_0) and repeat the process

Now substituting, we get

$$y_1' = x_1^2 y_1 - 1 = -0.991$$

$$y_1'' = 2x_1 y_1 + x_1^2 y_1' = 0.17015$$

$$y_1''' = 2y_1 + 4x_1 y_1' + x_1^2 y_1'' = 1.40592$$

$$y_1'''' = 6y_1' + 6x_1 y_1'' + x_1^2 y_1''' = -5.82983$$

Putting these values in Taylor's Series, we have

$$y_2 = y_1 + hy_1' + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1'''' + ...$$

$$= 0.90031 + 0.1 \cdot (-0.991) + \frac{(0.1)^2}{2!} \cdot (0.17015) + \frac{(0.1)^3}{3!} \cdot (1.40592) + \frac{(0.1)^4}{4!} \cdot (-5.82983) + ...$$

$$= 0.90031 - 0.0991 + 0.00085 + 0.00023 + 0 + ...$$

$$= 0.80227$$

$\therefore y(0.2) = 0.80227$

---

2. Find y(0.5) for $y' = -2x - y$, y(0) = -1, with step length 0.1 using Taylor Series method

Solution:

Given $y' = -2x - y, y(0) = -1, h = 0.1, y(0.5) = ?$

Here, $x_0 = 0, y_0 = -1, h = 0.1$

Differentiating successively, we get

$y' = -2x - y$

$y'' = -2 - y'$

$y''' = -y''$

$y'''' = -y'''$

Now substituting, we get

$y_0' = -2x_0 - y_0 = 1$

$y_0'' = -2 - y_0' = -3$

$y_0''' = -y_0'' = 3$

$y_0'''' = -y_0''' = -3$

Putting these values in Taylor's Series, we have

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0'''' + \ldots$$

$$= -1 + 0.1 \cdot (1) + \frac{(0.1)^2}{2!} \cdot (-3) + \frac{(0.1)^3}{3!} \cdot (3) + \frac{(0.1)^4}{4!} \cdot (-3) + \ldots$$

$= -1 + 0.1 - 0.015 + 0.0005 + 0 + \ldots$

$= -0.91451$

Again taking (x_1, y_1) in place of (x_0, y_0) and repeat the process

Now substituting, we get

$y_1' = -2x_1 - y_1 = 0.71451$

$y_1'' = -2 - y_1' = -2.71451$

$y_1''' = -y_1'' = 2.71451$

$y_1'''' = -y_1''' = -2.71451$

Putting these values in Taylor's Series, we have

$$y_2 = y_1 + hy_1' + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1'''' + \ldots$$

$$= -0.91451 + 0.1 \cdot (0.71451) + \frac{(0.1)^2}{2!} \cdot (-2.71451) + \frac{(0.1)^3}{3!} \cdot (2.71451) + \frac{(0.1)^4}{4!} \cdot (-2.71451) + \ldots$$

$= -0.91451 + 0.07145 - 0.01357 + 0.00045 + 0 + \ldots$

$= -0.85619$

Again taking (x_2, y_2) in place of (x_1, y_1) and repeat the process

Now substituting, we get

$y_2' = -2x_2 - y_2 = 0.45619$

$y_2'' = -2 - y_2' = -2.45619$

$y_2''' = -y_2'' = 2.45619$

$y_2'''' = -y_2''' = -2.45619$

Putting these values in Taylor's Series, we have

$$y_3 = y_2 + hy_2' + \frac{h^2}{2!}y_2'' + \frac{h^3}{3!}y_2''' + \frac{h^4}{4!}y_2'''' + \ldots$$

$$= -0.85619 + 0.1 \cdot (0.45619) + \frac{(0.1)^2}{2!} \cdot (-2.45619) + \frac{(0.1)^3}{3!} \cdot (2.45619) + \frac{(0.1)^4}{4!} \cdot (-2.45619) + \ldots$$

$= -0.85619 + 0.04562 - 0.01228 + 0.00041 + 0 + \ldots$

$= -0.82246$

Again taking (x_3, y_3) in place of (x_2, y_2) and repeat the process

Now substituting, we get

$y_3' = -2x_3 - y_3 = 0.22246$

$y_3'' = -2 - y_3' = -2.22246$

$y_3''' = -y_3'' = 2.22246$

$y_3'''' = -y_3''' = -2.22246$

Putting these values in Taylor's Series, we have

$$y_4 = y_3 + hy_3' + \frac{h^2}{2!}y_3'' + \frac{h^3}{3!}y_3''' + \frac{h^4}{4!}y_3'''' + ...$$

$$= -0.82246 + 0.1 \cdot (0.22246) + \frac{(0.1)^2}{2!} \cdot (-2.22246) + \frac{(0.1)^3}{3!} \cdot (2.22246) + \frac{(0.1)^4}{4!} \cdot (-2.22246) + ...$$

$$= -0.82246 + 0.02225 - 0.01111 + 0.00037 + 0 + ...$$

$$= -0.81096$$

Again taking (x_4, y_4) in place of (x_3, y_3) and repeat the process

Now substituting, we get

$y_4' = -2x_4 - y_4 = 0.01096$

$y_4'' = -2 - y_4' = -2.01096$

$y_4''' = -y_4'' = 2.01096$

$y_4'''' = -y_4''' = -2.01096$

Putting these values in Taylor's Series, we have

$$y_5 = y_4 + hy_4' + \frac{h^2}{2!}y_4'' + \frac{h^3}{3!}y_4''' + \frac{h^4}{4!}y_4'''' + ...$$

$$= -0.81096 + 0.1 \cdot (0.01096) + \frac{(0.1)^2}{2!} \cdot (-2.01096) + \frac{(0.1)^3}{3!} \cdot (2.01096) + \frac{(0.1)^4}{4!} \cdot (-2.01096) + ...$$

$$= -0.81096 + 0.0011 - 0.01005 + 0.00034 + 0 + ...$$

$$= -0.81959$$

$$\therefore y(0.5) = -0.81959$$

2- Euler method

In mathematics and computational science, the **Euler method** (also called **forward Euler method**) is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value $y(x_0)=y_0$.

Euler Method

$$y_{i+1} = y_i + h\ f(x_i, y_i)$$

Examples:

1. Find y(0.2) for $y' = \frac{x-y}{2}$, y(0) = 1, with step length 0.1 using Euler method

Solution:

Given $y' = \frac{x-y}{2}, y(0) = 1, h = 0.1, y(0.2) = ?$

Euler method

$y_1 = y_0 + hf(x_0, y_0) = 1 + (0.1)f(0, 1) = 1 + (0.1)\cdot(-0.5) = 1 + (-0.05) = 0.95$

$y_2 = y_1 + hf(x_1, y_1) = 0.95 + (0.1)f(0.1, 0.95) = 0.95 + (0.1)\cdot(-0.425) = 0.95 + (-0.0425) = 0.9075$

$\therefore y(0.2) = 0.9075$

---

2. Find y(0.5) for $y' = -2x - y$, y(0) = -1, with step length 0.1 using Euler method

Solution:

Given $y' = -2x - y, y(0) = -1, h = 0.1, y(0.5) = ?$

Euler method

$y_1 = y_0 + hf(x_0, y_0) = -1 + (0.1)f(0, -1) = -1 + (0.1)\cdot(1) = -1 + (0.1) = -0.9$

$y_2 = y_1 + hf(x_1, y_1) = -0.9 + (0.1)f(0.1, -0.9) = -0.9 + (0.1)\cdot(0.7) = -0.9 + (0.07) = -0.83$

$y_3 = y_2 + hf(x_2, y_2) = -0.83 + (0.1)f(0.2, -0.83) = -0.83 + (0.1)\cdot(0.43) = -0.83 + (0.043) = -0.787$

$y_4 = y_3 + hf(x_3, y_3) = -0.787 + (0.1)f(0.3, -0.787) = -0.787 + (0.1)\cdot(0.187) = -0.787 + (0.0187) = -0.7683$

$y_5 = y_4 + hf(x_4, y_4) = -0.7683 + (0.1)f(0.4, -0.7683) = -0.7683 + (0.1)\cdot(-0.0317) = -0.7683 + (-0.00317) = -0.77147$

$\therefore y(0.5) = -0.77147$

3- Runge-Kutta Second Order (Heun Method)

$$k_1 = f(x_0, y_0)$$

$$k_2 = f(x_0 + h, y_0 + k_1 h)$$

$$Yi + 1 = yi + \frac{h}{2}(k_1 + k_2)$$

Example :

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

Use RK2 to find $y(1.01), y(1.02)$

Step 1:

$$K_1 = f(x_0, y_0) = (1 + y_0{}^2 + x_0{}^3) = 18.0$$

$$K_2 = f(x_0 + h, y_0 + K_1 h) = (1 + (y_0 + 0.18)^2 + (x_0 + .01)^3) = 16.6227$$

$$y_1 = y_0 + \frac{h}{2}(K_1 + K_2) = -4 + \frac{0.01}{2}(18 + 16.6227) = -3.8268$$

$$h = 0.01$$

$$f(x, y) = 1 + y^2 + x^3$$

$$x_1 = 1.01, \quad y_1 = -3.8254$$

Step 2:

$$K_1 = f(x_1, y_1) = (1 + y_1{}^2 + x_1{}^3) = 16.6746$$

$$K_2 = f(x_1 + h, y_1 + K_1 h) = (1 + (y_1 + 0.1666)^2 + (x_1 + .01)^3) = 15.4576$$

$$y_2 = y_1 + \frac{h}{2}(K_1 + K_2) = -3.8268 + \frac{0.01}{2}(16.6746 + 15.4576) = -3.6661$$

i	x_i	y_i
0	1.00	-4.0000
1	1.01	-3.8254
2	1.02	-3.6661

4- Runge-Kutta fourth order

Fourth Order Runge – Kutta Method:

$$k_1 = h\, f(x_n, y_n)$$
$$k_2 = h\, f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$
$$k_3 = h\, f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$
$$k_4 = h\, f(x_n + h, y_n + k_3)$$
$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

Example:

Consider

$$\frac{dy}{dx} = y - x^2$$

The initial condition is: $y(0) = 1$

The step size is: $h = 0.1$

The exact Solution : $y = 2 + 2x + x^2 - e^x$

The example of a single step:

$$k_1 = h\,[\,f(x,y)] = 0.1\,f(0,1) = 0.1\,(1-0^2) = 0.1$$

$$k_2 = h\left[f\left(x+\tfrac{1}{2}\,h, y+\tfrac{1}{2}\,k_1\right)\right] = 0.1\,f(0.05, 1.05) = 0.10475$$

$$k_3 = h\left[f\left(x+\frac{1}{2}\,h, y+\frac{1}{2}\,k_2\right)\right] = 0.1\,f(0.05, 1.+k_2/2) = 0.104988$$

$$k_4 = h\,[\,f(x+h, y+k_3)] = 0.1\,f(0.1, 1.104988) = 0.109499$$

$$y_{n+1} = y_n + \frac{1}{6}\,[k_1 + 2k_2 + 2k_3 + k_4] = 1.104829$$

5- Homework: Continue to solve for y(0.5)

Lecture 12

Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points. The first degree polynomial equation is a line with slope *a*. A line will connect any two points, so a first degree polynomial equation is an exact fit through any two points with distinct x coordinates.

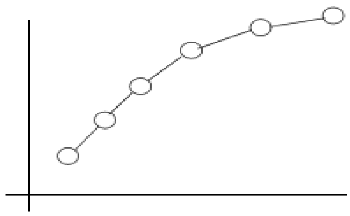
1) Interpolation (connect the data-dots)

If data is reliable, we can plot it and connect the dots

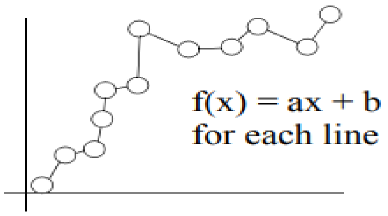
This is piece-wise, linear interpolation

This has limited use as a general function $f(x)$

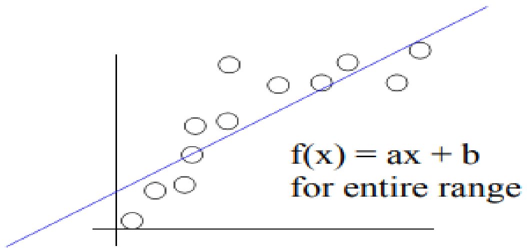
Since its really a group of small $f(x)$ s, connecting one point to the next it doesn't work very well for data that has built in random error (scatter)



2) Curve fitting - capturing the trend in the data by assigning a single function across the entire range. The example below uses a straight line function



Interpolation



Curve Fitting

A straight line is described generically by $f(x) = ax + b$

The goal is to identify the coefficients 'a' and 'b' such that f(x) 'fits' the data well

Linear curve fitting (linear regression)

Given the general form of a straight line

$$f(x) = ax + b$$

Solve for the a and b so that the previous two equations both = 0

re-write these two equations

$$a\sum x_i^2 + b\sum x_i = \sum (x_i y_i)$$
$$a\sum x_i + b*n = \sum y_i$$

put these into **matrix form**

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

what's unknown?

we have the data points (x_i, y_i) for $i = 1, \ldots, n$, so we have all the summation terms in the matrix

so unknows are a and b

Good news, we already know how to solve this problem

remember Gaussian elimination ??

$$A = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}, \quad X = \begin{bmatrix} b \\ a \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

so

$AX = B$

using built in Mathcad matrix inversion, the coefficients a and b are solved

```
>> X = A^-1*B
```

Note: A, B, and X are not the same as a, b, and x

Let's test this with an example:

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	1.5	3.0	4.5	6.0	7.5

First we find values for all the summation terms

$n = 6$

$\sum x_i = 7.5, \quad \sum y_i = 22.5, \quad \sum x_i^2 = 13.75, \quad \sum x_i y_i = 41.25$

Now plugging into the matrix form gives us:

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix}$$

Note: we are using $\sum x_i^2$, NOT $(\sum x_i)^2$

$$\begin{bmatrix} b \\ a \end{bmatrix} = inv\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} * \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix}$$

or use Gaussian elimination...

The solution is $\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$ ===> $f(x) = 3x + 0$

This fits the data exactly. That is, the error is zero. Usually this is not the outcome. Usually we have data that does not exactly fit a straight line.

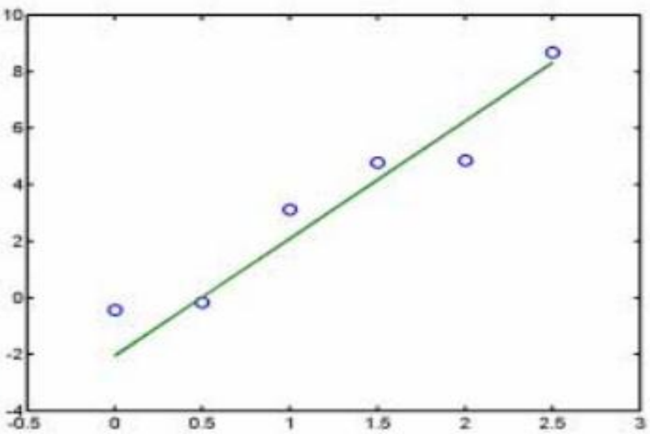
Here's an example with some 'noisy' data

x = [0 .5 1 1.5 2 2.5], y = [-0.4326 -0.1656 3.1253 4.7877 4.8535 8.6909]

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}, \quad \begin{bmatrix} b \\ a \end{bmatrix} = inv\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} * \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}, \quad \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.975 \\ 3.561 \end{bmatrix}$$

so our fit is $f(x) = 3.561\,x - 0.975$

Here's a plot of the data and the curve fit:



So...what do we do when a straight line is not suitable for the data set?

Polynomial Curve Fitting

Consider the general form for a polynomial of order j

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k \qquad (1)$$

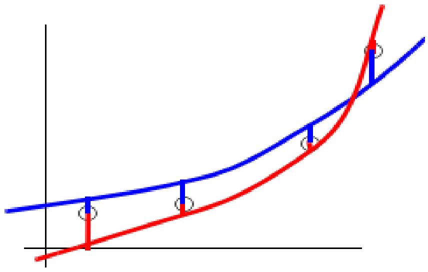
Just as was the case for linear regression, we ask:

How can we pick the coefficients that best fits the curve to the data? We can use the same idea:

The curve that gives minimum error between data y and the fit $f(x)$ is 'best'

Quantify the error for these two second order curves...

- Add up the length of all the red and blue verticle lines
- pick curve with minimum total error



re-write these $j+1$ equations, and put into matrix form

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & \dots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \dots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \dots & \sum x_i^{j+j} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum \left(x_i^2 y_i\right) \\ \vdots \\ \sum \left(x_i^j y_i\right) \end{bmatrix}$$

where all summations above are over $i = 1, \dots, n$

we have the data points (x_i, y_i) for $i = 1, \dots, n$

we want $a_0, a_k \quad k = 1, \dots, j$

We already know how to solve this problem. Remember Gaussian elimination ??

$$A = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \dots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \dots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \dots & \sum x_i^{j+j} \end{bmatrix}, \quad X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum \left(x_i^2 y_i\right) \\ \vdots \\ \sum \left(x_i^j y_i\right) \end{bmatrix}$$

where all summations above are over $i = 1, \dots, n$ data points

Note: No matter what the order j, we always get equations **LINEAR** with respect to the coefficients. This means we can use the following solution method

$$AX = B$$

using built in Mathcad matrix inversion, the coefficients a and b are solved

```
>> X = A^-1*B
```

Example #1:

Fit a second order polynomial to the following data

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	0.25	1.0	2.25	4.0	6.25

Since the order is 2 ($j = 2$), the matrix form to solve is

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

Now plug in the given data.

Before we go on...***what answers do you expect for the coefficients after looking at the data?***

$n = 6$

$\sum x_i = 7.5$, $\quad \sum y_i = 13.75$

$\sum x_i^2 = 13.75$, $\quad \sum x_i y_i = 28.125$

$\sum x_i^3 = 28.125$ $\quad \sum x_i^2 y_i = 61.1875$

$\sum x_i^4 = 61.1875$

$$\begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1875 \end{bmatrix}$$

Note: we are using $\sum x_i^2$, NOT $(\sum x_i)^2$. There's a big difference

using the inversion method $\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} * \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1857 \end{bmatrix}$

or use Gaussian elimination gives us the solution to the coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad ===> \quad f(x) = 0 + 0*x + 1*x^2$$

This fits the data exactly. That is, f(x) = y since y = x^2

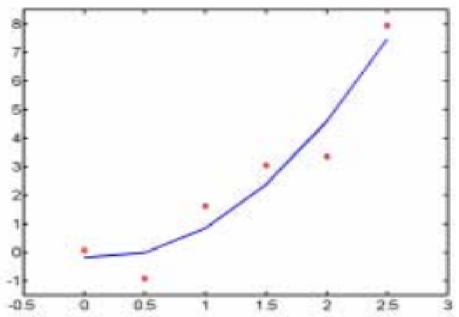
__Example #2: uncertain data__

Now we'll try some 'noisy' data

x = [0 .0 1 1.5 2 2.5]
y = [0.0674 -0.9156 1.6253 3.0377 3.3535 7.9409]

The resulting system to solve is:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv\begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} * \begin{bmatrix} 15.1093 \\ 32.2834 \\ 71.276 \end{bmatrix}$$



giving:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -0.1812 \\ -0.3221 \\ 1.3537 \end{bmatrix}$$

So our fitted second order function is:

$$f(x) = -0.1812 - 0.3221x* + 1.3537*x^2$$

Cramer's method

Find the system of Linear Equations using Cramers Rule:

2x + y + z = 3

x – y – z = 0

x + 2y + z = 0

it clear the Cramer's rule is to define the matrices A, X, Ax, Ay, and Az:

```
 clc
% Cramer's method
A = [2 1 1; 1 -1 -1; 1 2 1];
X = [3; 0; 0];
Ax = [3 1 1 ; 0 -1 -1;0 2 1 ]
Ay = [2 3 1; 1 0 -1; 1 0 1]
Az = [2 1 3; 1 -1 0; 1 2 0]
x = det(Ax)/det(A)
```

```
y = det(Ay)/det(A)
z = det(Az)/det(A)

thus the answer will be:

Ax =

     3     1     1
     0    -1    -1
     0     2     1

Ay =

     2     3     1
     1     0    -1
     1     0     1

Az =

     2     1     3
     1    -1     0
     1     2     0
x =
     1
y =
    -2
z =
     3
```

Numerical Methods Lecture 5 - Curve Fitting Techniques

Topics
- motivation
- interpolation
- linear regression
- higher order polynomial form
- exponential form

Curve fitting - motivation

For root finding, we used a given function to identify where it crossed zero

where does $f(x) = 0$??

Q: Where does this given function $f(x)$ come from in the first place?

- Analytical models of phenomena (e.g. equations from physics)
- Create an equation from observed data

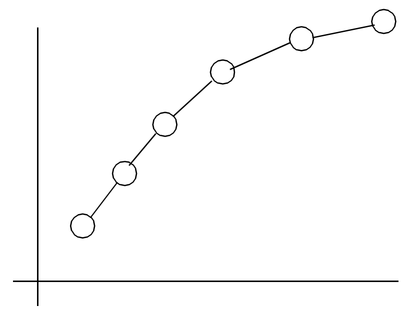
1) <u>Interpolation</u> (connect the data-dots)

If data is reliable, we can plot it and connect the dots

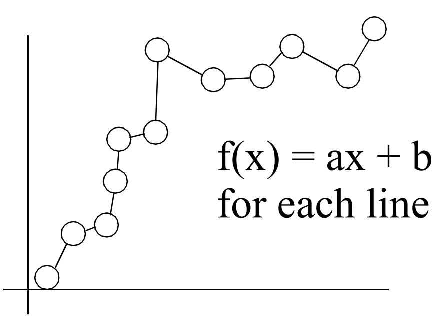
This is piece-wise, linear interpolation

This has limited use as a general function $f(x)$

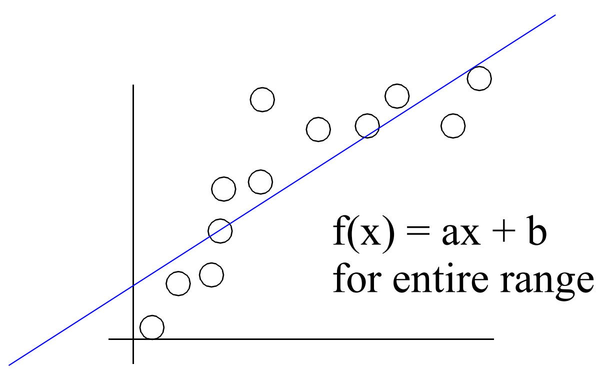
Since its really a group of small $f(x)$ s, connecting one point to the next it doesn't work very well for data that has built in random error (scatter)



2) <u>Curve fitting</u> - capturing the trend in the data by assigning a single function across the entire range. The example below uses a straight line function



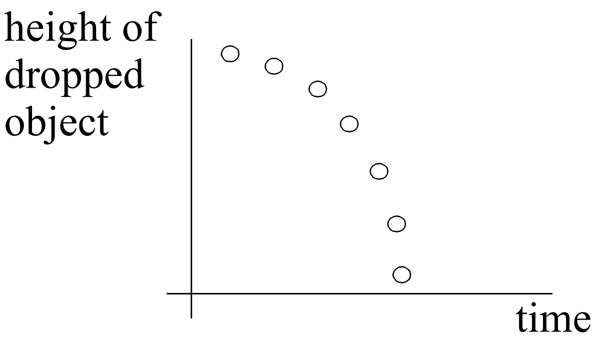
Interpolation

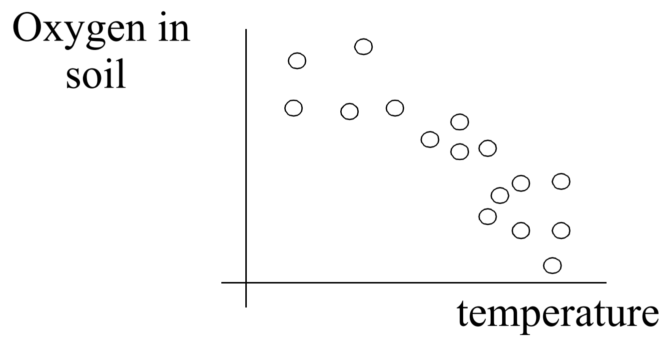


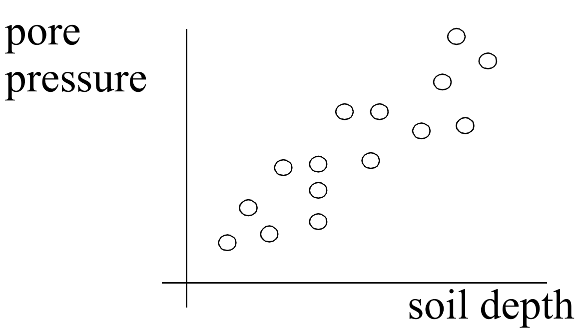
Curve Fitting

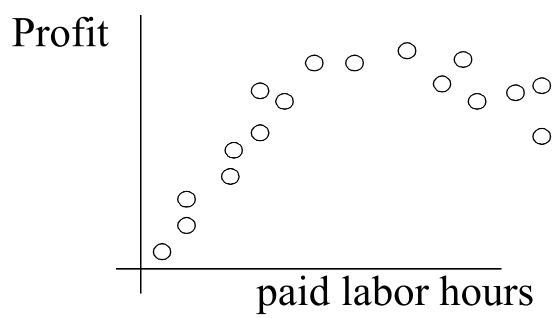
A straight line is described generically by $f(x) = ax + b$

The goal is to identify the coefficients 'a' and 'b' such that f(x) 'fits' the data well other examples of data sets that we can fit a function to.







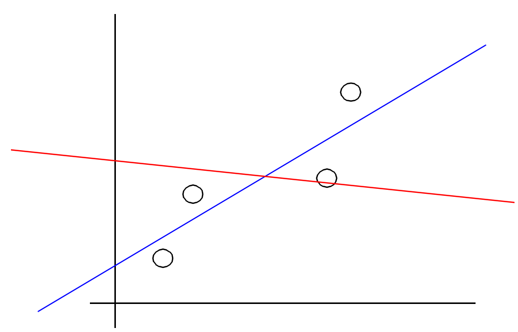


Is a straight line suitable for each of these cases ?
No. But we’re not stuck with just straight line fits. We’ll start with straight lines, then expand the concept.

Linear curve fitting (linear regression)

Given the general form of a straight line

$$f(x) = ax + b$$

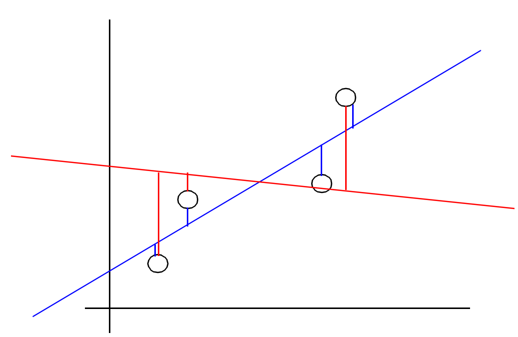


How can we pick the coefficients that best fits the line to the data?

First question: What makes a particular straight line a ‘good’ fit?

Why does the blue line appear to us to fit the trend better?

- Consider the distance between the data and points on the line
- Add up the length of all the red and blue verticle lines
- This is an expression of the ‘error’ between data and fitted line
- The one line that provides a <u>minimum</u> error is then the ‘best’ straight line

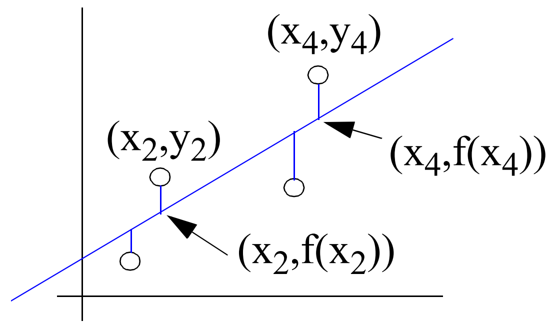


Quantifying error in a curve fit
assumptions:

1) positive or negative error have the same value
(data point is above or below the line)
2) Weight greater errors more heavily

we can do both of these things by squaring the distance

denote data values as (x, y) ==============>>
denote points on the fitted line as (x, f(x))
sum the error at the four data points



$$err = \sum (d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - f(x_4))^2$$

Our fit is a straight line, so now substitute $f(x) = ax + b$

$$err = \sum_{i=1}^{\#\text{ data points}} (y_i - f(x_i))^2 = \sum_{i=1}^{\#\text{ data points}} (y_i - (ax_i + b))^2$$

The 'best' line has **minimum error** between line and data points

This is called the **least squares approach**, since we minimize the square of the error.

$$\text{minimize} \quad err = \sum_{i=1}^{\#\text{ data points} = n} (y_i - (ax_i + b))^2$$

time to pull out the **calculus**... finding the minimum of a function
1) derivative describes the slope
2) slope = zero is a minimum
==> take the derivative of the error with respect to a and b, set each to zero

$$\frac{\partial err}{\partial a} = -2 \sum_{i=1}^{n} x_i(y_i - ax_i - b) = 0$$

$$\frac{\partial err}{\partial b} = -2 \sum_{i=1}^{n} (y_i - ax_i - b) = 0$$

Solve for the a and b so that the previous two equations both = 0
re-write these two equations

$$a\sum x_i^2 + b\sum x_i = \sum(x_i y_i)$$
$$a\sum x_i + b*n = \sum y_i$$

put these into **matrix form**

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum(x_i y_i) \end{bmatrix}$$

what's unknown?
we have the data points (x_i, y_i) for $i = 1, ..., n$, so we have all the summation terms in the matrix

so unknows are a and b
Good news, we already know how to solve this problem
remember Gaussian elimination ??

$$A = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}, \quad X = \begin{bmatrix} b \\ a \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum(x_i y_i) \end{bmatrix}$$

so
$AX = B$

using built in Mathcad matrix inversion, the coefficients a and b are solved

```
>> X = A^-1*B
```

Note: A, B, and X are not the same as a, b, and x

Let's test this with an example:

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	1.5	3.0	4.5	6.0	7.5

First we find values for all the summation terms
$n = 6$

$$\sum x_i = 7.5, \quad \sum y_i = 22.5, \quad \sum x_i^2 = 13.75, \quad \sum x_i y_i = 41.25$$

Now plugging into the matrix form gives us:

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix}$$

Note: we are using $\sum x_i^2$, NOT $(\sum x_i)^2$

$$\begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} * \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix}$$ or use Gaussian elimination...

The solution is $\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$ ===> $f(x) = 3x + 0$

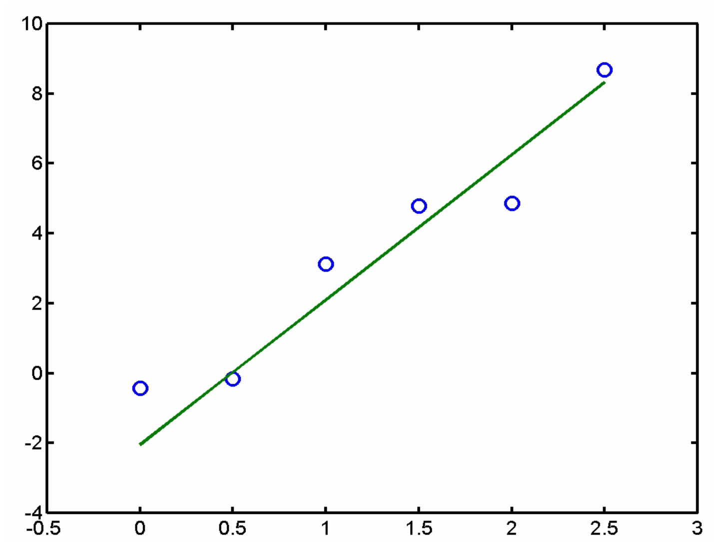
This fits the data exactly. That is, the error is zero. Usually this is not the outcome. Usually we have data that does not exactly fit a straight line.
Here's an example with some 'noisy' data

x = [0 .5 1 1.5 2 2.5], y = [-0.4326 -0.1656 3.1253 4.7877 4.8535 8.6909]

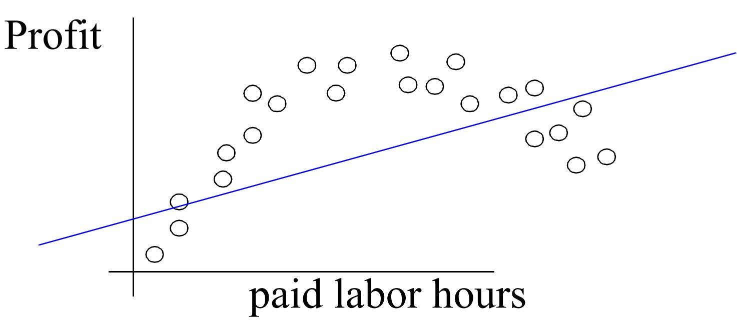
$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}, \quad \begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} * \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}, \quad \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.975 \\ 3.561 \end{bmatrix}$$

so our fit is $f(x) = 3.561\ x - 0.975$

Here's a plot of the data and the curve fit:



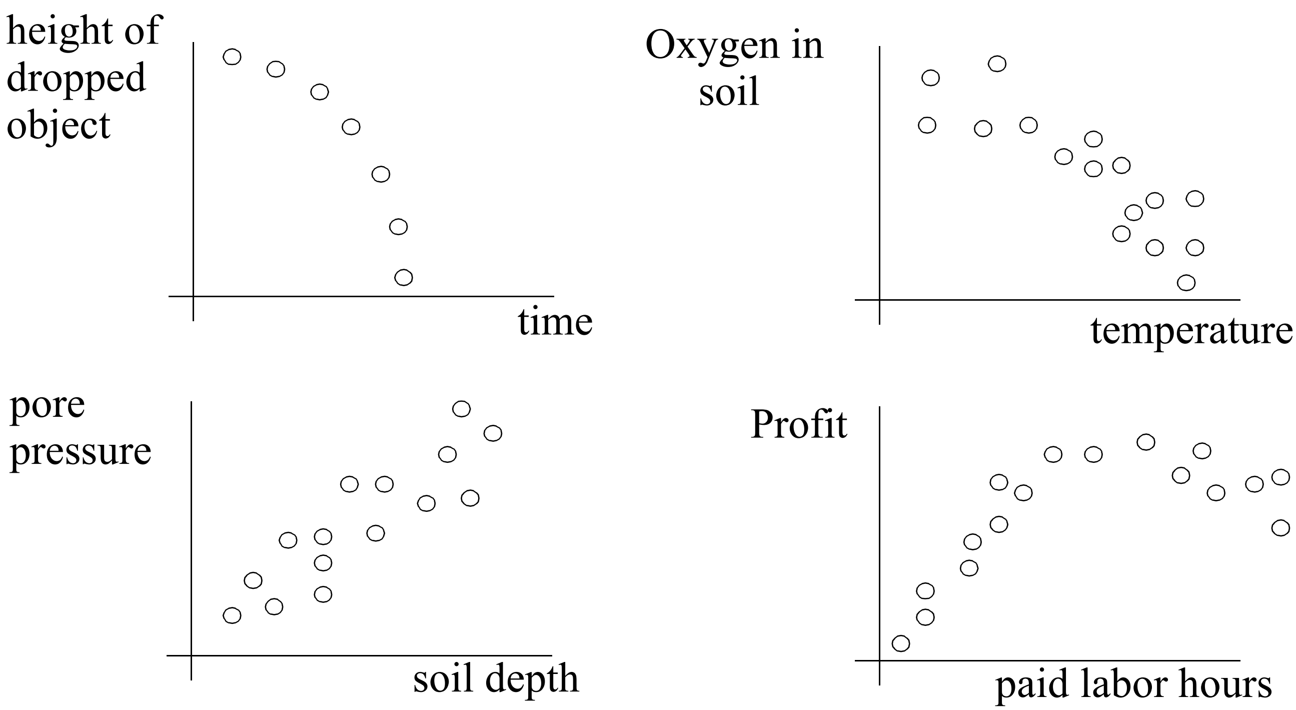
So...what do we do when a straight line is not suitable for the data set?



Straight line will not predict diminishing returns that data shows

Curve fitting - <u>higher order polynomials</u>

We started the linear curve fit by choosing a generic form of the straight line $f(x) = ax + b$
This is just one kind of function. There are an infinite number of generic forms we could choose from for almost any shape we want. Let's start with a simple extension to the linear regression concept
recall the examples of sampled data



Is a straight line suitable for each of these cases ? Top left and bottom right don't look linear in trend, so why fit a straight line? No reason to, let's consider other options. There are lots of functions with lots of different shapes that depend on coefficients. We can choose a form based on experience and trial/error. Let's develop a few options for non-linear curve fitting. We'll start with a simple extension to linear regression...higher order polynomials

Polynomial Curve Fitting

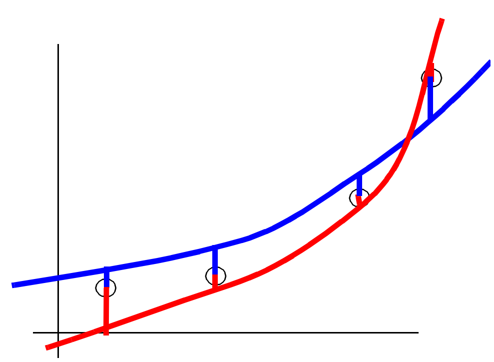
Consider the general form for a polynomial of order j

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k \qquad (1)$$

Just as was the case for linear regression, we ask:

How can we pick the coefficients that best fits the curve to the data? We can use the same idea:

The curve that gives minimum error between data y and the fit $f(x)$ is 'best'



Quantify the error for these two second order curves...

- Add up the length of all the red and blue verticle lines
- pick curve with minimum total error

Error - Least squares approach

The general expression for any error using the least squares approach is

$$err = \sum(d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - f(x_4))^2 \qquad (2)$$

where we want to minimize this error. Now substitute the form of our eq. (1)

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k$$

into the general least squares error eq. (2)

$$err = \sum_{i=1}^{n} \left(y_i - \left(a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + \ldots + a_j x_i^j \right) \right)^2 \qquad (3)$$

where: n - # of data points given, i - the current data point being summed, j - the polynomial order
re-writing eq. (3)

$$err = \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right)^2 \qquad (4)$$

find the best line = minimize the error (squared distance) between line and data points
Find the set of coefficients a_k, a_0 so we can minimize eq. (4)

CALCULUS TIME

To minimize eq. (4), take the derivative with respect to each coefficient $a_0, a_k \quad k = 1, \ldots, j$ set each to zero

$$\frac{\partial err}{\partial a_0} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) = 0$$

$$\frac{\partial err}{\partial a_1} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x = 0$$

$$\frac{\partial err}{\partial a_2} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x^2 = 0$$

$$\vdots$$

$$\frac{\partial err}{\partial a_j} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x^j = 0$$

re-write these $j + 1$ equations, and put into matrix form

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & \dots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \dots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \dots & \sum x_i^{j+j} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum \left(x_i^2 y_i\right) \\ \vdots \\ \sum \left(x_i^j y_i\right) \end{bmatrix}$$

where all summations above are over $i = 1, ..., n$

what's unknown?

we have the data points (x_i, y_i) for $i = 1, ..., n$

we want $a_0, a_k \quad k = 1, ..., j$

We already know how to solve this problem. Remember Gaussian elimination ??

$$A = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \dots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \dots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \dots & \sum x_i^{j+j} \end{bmatrix}, \quad X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum \left(x_i^2 y_i\right) \\ \vdots \\ \sum \left(x_i^j y_i\right) \end{bmatrix}$$

where all summations above are over $i = 1, ..., n$ data points

Note: No matter what the order j, we always get equations **LINEAR** with respect to the coefficients. This means we can use the following solution method

$AX = B$

using built in Mathcad matrix inversion, the coefficients a and b are solved

```
>> X = A^-1*B
```

Example #1:

Fit a second order polynomial to the following data

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	0.25	1.0	2.25	4.0	6.25

Since the order is 2 ($j = 2$), the matrix form to solve is

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

Now plug in the given data.

Before we go on...***what answers do you expect for the coefficients after looking at the data?***

$n = 6$

$\sum x_i = 7.5$, $\sum y_i = 13.75$

$\sum x_i^2 = 13.75$, $\sum x_i y_i = 28.125$

$\sum x_i^3 = 28.125$ $\sum x_i^2 y_i = 61.1875$

$\sum x_i^4 = 61.1875$

$$\begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1875 \end{bmatrix}$$

Note: we are using $\sum x_i^2$, NOT $(\sum x_i)^2$. There's a big difference

using the inversion method $\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} * \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1857 \end{bmatrix}$

or use Gaussian elimination gives us the solution to the coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad ===> \quad f(x) = 0 + 0*x + 1*x^2$$

This fits the data exactly. That is, f(x) = y since y = x^2

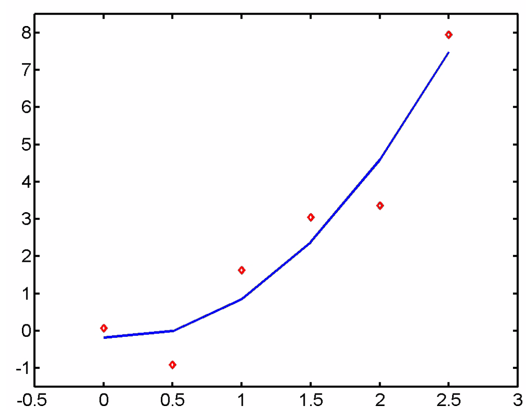
Example #2: uncertain data

Now we'll try some 'noisy' data

x = [0 .0 1 1.5 2 2.5]
y = [0.0674 -0.9156 1.6253 3.0377 3.3535 7.9409]

The resulting system to solve is:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} * \begin{bmatrix} 15.1093 \\ 32.2834 \\ 71.276 \end{bmatrix}$$



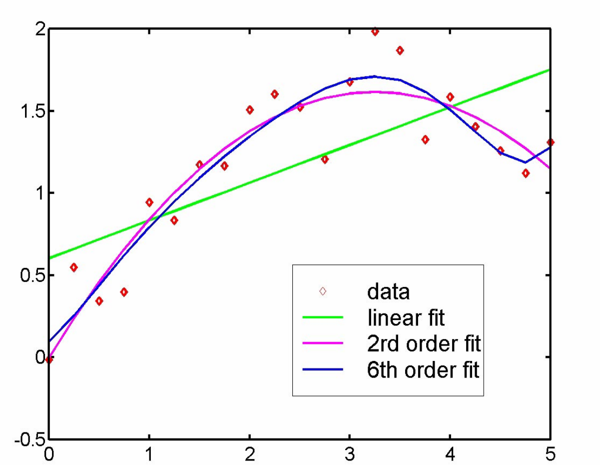
giving:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -0.1812 \\ -0.3221 \\ 1.3537 \end{bmatrix}$$

So our fitted second order function is:

$$f(x) = -0.1812 - 0.3221x* + 1.3537*x^2$$

Example #3 : data with three different fits



In this example, we're not sure which order will fit well, so we try three different polynomial orders
Note: Linear regression, or first order curve fitting is just the general polynomial form we just saw, where we use j=1,

- 2nd and 6th order look similar, but 6th has a 'squiggle to it. We may not want that...

Overfit / Underfit - picking an inappropriate order

Overfit - over-doing the requirement for the fit to 'match' the data trend (order too high)

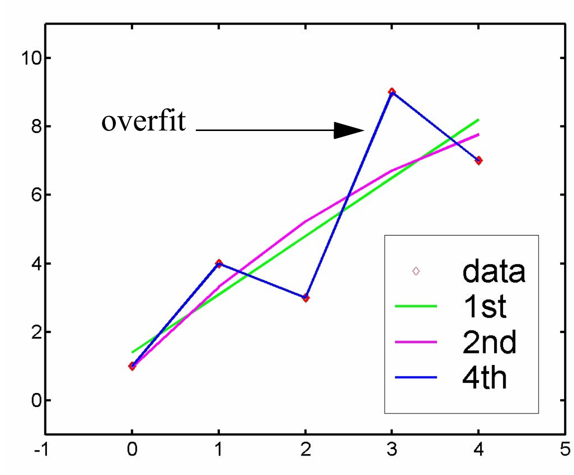
Polynomials become more 'squiggly' as their order increases. A 'squiggly' appearance comes from inflections in function

Consideration #1:

3rd order - 1 inflection point
4th order - 2 inflection points
nth order - n-2 inflection points

Consideration #2:

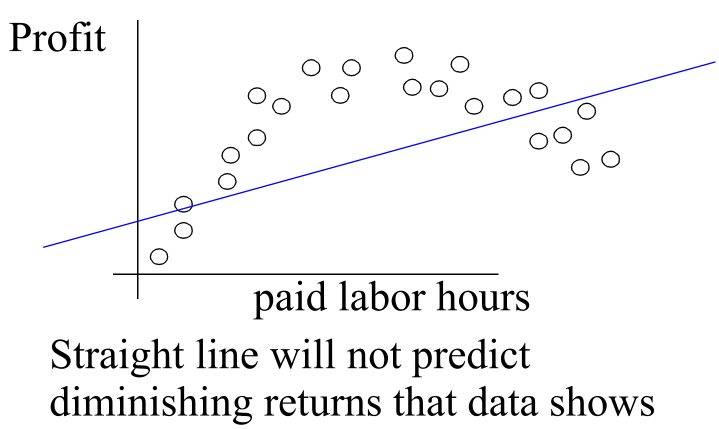
2 data points - linear touches each point
3 data points - second order touches each point
n data points - n-1 order polynomial will touch each point



SO: Picking an order too high will **overfit** data

General rule: pick a polynomial form at least several orders lower than the number of data points.
Start with linear and add order until trends are matched.

Underfit - If the order is too low to capture obvious trends in the data



General rule: View data first, then select an order that reflects inflections, etc.

For the example above:
1) Obviously nonlinear, so order > 1
2) No inflcetion points observed as obvious, so order < 3 is recommended
=====> I'd use 2nd order for this data

Curve fitting - Other nonlinear fits (exponential)

Q: Will a polynomial of any order necessarily fit any set of data?
A: Nope, lots of phenomena don't follow a polynomial form. They may be, for example, exponential

Example : Data (x,y) follows exponential form

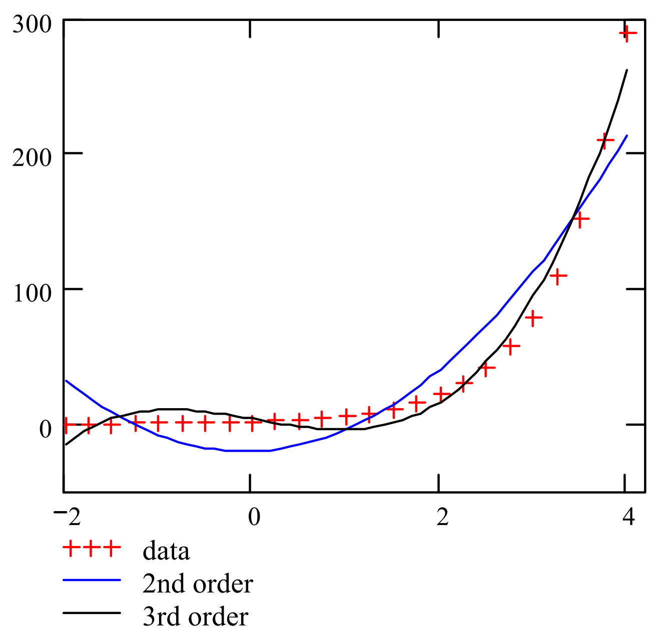
The next line references a separate worksheet with a function inside called Create_Vector. I can use the function here as long as I reference the worksheet first

Reference:C:\Mine\Mathcad\Tutorials\MyFunctions.mcd

X := Create_Vector(−2, 4, .25)   Y := 1.6·exp(1.3·X)

f2 := regress(X, Y, 2)   f3 := regress(X, Y, 3)

fit2(x) := interp(f2, X, Y, x)   fit3(x) := interp(f3, X, Y, x)   i := −2, −1.9.. 4



Note that neither 2nd nor 3rd order fit really describes the data well, but higher order will only get more 'squiggly'

We created this sample of data using an exponential function. Why not create a general form of the exponential function, and use the error minimization concept to identify its coefficients. That is, let's replace

the polynomial equation $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k$

With a general exponential equation $f(x) = Ce^{Ax} = C\exp(Ax)$
where we will seek C and A such that this equation fits the data as best it can.

Again with the error: solve for the coefficients C, A such that the error is minimized:

minimize $$err = \sum_{i=1}^{n} (y_i - (C\exp(Ax)))^2$$

Problem: When we take partial derivatives with respect to err and set to zero, we get two **NONLINEAR** equations with respect to C, A

So what? We can't use Gaussian Elimination or the inverse function anymore.
Those methods are for **LINEAR** equations only...

Now what?

Solution #1: Nonlinear equation solving methods

Remember we used Newton Raphson to solve a single nonlinear equation? (root finding)
We can use Newton Raphson to solve a system of nonlinear equations.
Is there another way? For the exponential form, yes there is

Solution #2: Linearization:

Let's see if we can do some algebra and change of variables to re-cast this as a linear problem...
Given: pair of data (x,y)

Find: a function to fit data of the general exponential form $y = Ce^{Ax}$

1) Take logarithm of both sides to get rid of the exponential $\ln(y) = \ln(Ce^{Ax}) = Ax + \ln(C)$

2) Introduce the following change of variables: $Y = \ln(y)$, $X = x$, $B = \ln(C)$

Now we have: $Y = AX + B$ which is a LINEAR equation

The original data points in the $x - y$ plane get mapped into the $X - Y$ plane.

This is called data linearization. The data is transformed as: $(x, y) \Rightarrow (X, Y) = (x, \ln(y))$

Now we use the method for solving a first order linear curve fit $$\begin{bmatrix} n & \sum X \\ \sum X & \sum X^2 \end{bmatrix} \begin{bmatrix} B \\ A \end{bmatrix} = \begin{bmatrix} \sum Y \\ \sum XY \end{bmatrix}$$

for A and B, where above $Y = \ln(y)$, and $X = x$

Finally, we operate on $B = \ln(C)$ to solve $C = e^B$

And we now have the coefficients for $y = Ce^{Ax}$

<u>Example</u>: repeat previous example, add exponential fit

$X := \text{Create\_Vector}(-2, 4, .25) \qquad Y := 1.6 \cdot \exp(1.3 \cdot X)$

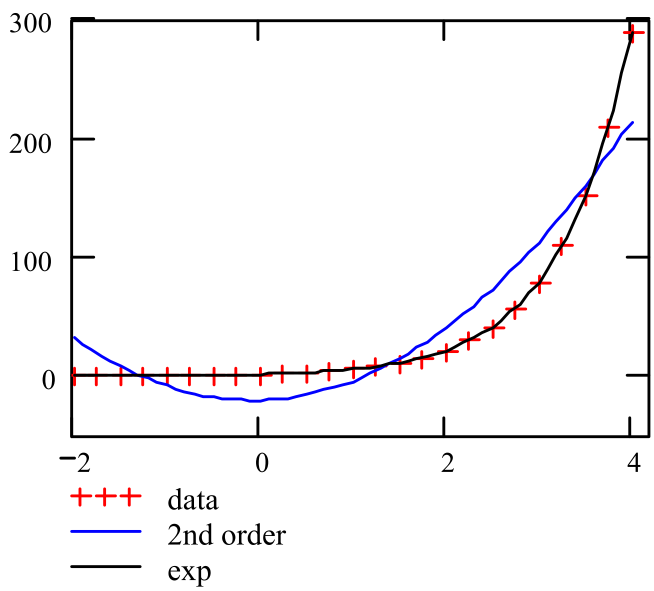
$f2 := \text{regress}(X, Y, 2) \qquad f3 := \text{regress}(X, Y, 3)$

$\text{fit2}(x) := \text{interp}(f2, X, Y, x) \qquad \text{fit3}(x) := \text{interp}(f3, X, Y, x)$

ADDING NEW STUFF FOR EXP FIT

$Y2 := \ln(Y) \qquad \text{fexp} := \text{regress}(X, Y2, 1) \qquad \text{coeff} := \text{submatrix}(\text{fexp}, 4, 5, 1, 1)$

$C := \exp\left(\text{coeff}_1\right) \qquad A := \text{coeff}_2 \qquad \text{fitexp}(x) := C \cdot \exp(A \cdot x) \qquad i := -2, -1.9 .. 4$



$A = 1.3$

$C = 1.6$