Notes for Week 3

Relation between Secant & Newton Rapson Method.

• Secant Method:

$$p_2 = p_1 - \frac{f(p_1)(p_1 - p_2)}{f(p_1) - f(p_0)}$$

If we make:

$$p_1 = p_0 + \varepsilon$$

$$p_2 = p_0 + \varepsilon - \frac{f(p_0+\varepsilon)(p_0+\varepsilon-p_0)}{f(p_0+\varepsilon) - f(p_0)}$$

$$\varepsilon \Rightarrow 0$$

$$p_2 = p_0 - \frac{f(p_0)\cdot\varepsilon}{f(p_1+\varepsilon) - f(p_0)}$$

$$p_2 = p_0 - \frac{f(p_0)}{f'(p_0)}$$

• Newton Method

The secant Method ~~coment~~ convert to Newton Method, $p_1 = p_0$

The ~~second m~~ solution of Linear system.

$$\mathbb{A}\,\mathbb{X} = \mathbb{B}$$

This is a system of equations.

eg.

$$6x + 3y + 2z = 29$$
$$3x + y + 2z = 17$$
$$2x + 3y + 2z = 21$$

$$\begin{bmatrix} 6 & 3 & 2 \\ 3 & 1 & 2 \\ 2 & 3 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 29 \\ 17 \\ 21 \end{bmatrix}$$

Matrix. vector

*Definitions.

N- Dimensional vector.

$$\mathbb{X} = (x_1, x_2, x_3 \ldots)$$

$x_1, x_2 \ldots x_n$ are called components of $\mathbb{X}$

When a vector is used to determine a position It

The ~~nonor~~ Euclidean Norm of a vector:

$|x| = (x_1^2 + x_2^2 + x_3^2 \cdots x_n^2)^{1/2}$

Multiplication Cx is strethed vector when $|c|>1$ & shrinks vector when $|c|<1$

$|Cx| = (C_1^2 x_1^2 + C_2^2 x_2^2 + \cdots C_n^2 x_n^2)^{1/2}$

$= |c|(x_1^2 + x_2^2 + \cdots + x_n^2)$

$= |c| \cdot |x|$.

Relationship between product & norm of a vector

$|x| = x_1^2 + x_2^2 + \cdots + x_n^2$

$= x \cdot x$

Displacement vector.

If x & y represent 2 points in the space, the disp vector from x to y is given by the difference $|y-x|$.

$|y-x|$

The distance between 2 points in Space:

$||y|-|x|| = ((x_1-y_1)^2 + (x_2-y_2)^2 \cdots \cdots (x_n-y_n)^2)^{1/2}$

eg:

Let $x = (-1, 3, 2)$ $y = (3, 5, 2)$

Sum $x+y = (-1+3, 3+5, 2+2) = (2, 8, 4)$

Difference: $x-y = (-1-3, 3-5, 2-2) = (-4, -2, 0)$

Matrices.

If $\mathbb{A}$ is a Matrices then the letter a_{ij} denotes the number in location i,j. (ith row & jth column)

If $\mathbb{A}$ is a $M \times N$ matrix, then it has M row & N colu

eg: $$\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$\mathbb{A}$ is a 3×4 matrixs

You can say that an $M \times N$ matrix has M rows of N dimensional vectors.

$$\mathbb{A} = \begin{bmatrix} \mathbb{V}_1 \\ \mathbb{V}_2 \\ \mathbb{V}_3 \\ \mathbb{V}_4 \end{bmatrix} = [\mathbb{V}_1, \mathbb{V}_2, \mathbb{V}_3, \mathbb{V}_4]$$

or $\mathbb{A}$ can be seen as N coloums with N dimensio

$\mathbb{A} = [C_1, C_2, C_3 \dots C_4]$

Operations:

- Equality $\mathbb{A} = \mathbb{B}$ if $a_{ij} = b_{ij}$.
- Sum. $\mathbb{A} + \mathbb{B} = [a_{ij} + b_{ij}]_{M \times N}$
- Negation $-\mathbb{A} = [-a_{ij}]_{M \times N}$.
- Difference $\mathbb{A} - \mathbb{B} = [a_{ij} - b_{ij}]_{M \times N}$.
- Scalar Mutiplication: $C\mathbb{A} = [Ca_{ij}]_{M \times N}$
- Weight sum. $p\mathbb{A} + q\mathbb{B} = [pa_{ij} + qb_{ij}]_{M \times N}$.

eg: $\mathbb{A} = \begin{bmatrix} 2 & 1 \\ 4 & 6 \\ 3 & 2 \end{bmatrix}$ $\mathbb{B} = \begin{bmatrix} 1 & 7 \\ 6 & 2 \\ 2 & 1 \end{bmatrix}$

$$3\mathbb{A} + 2\mathbb{B} = 3\begin{bmatrix} 2 & 1 \\ 4 & 6 \\ 3 & 2 \end{bmatrix} + 2\begin{bmatrix} 1 & 7 \\ 6 & 2 \\ 2 & 1 \end{bmatrix}$$

$$3\mathbb{A} + 2\mathbb{B} = \begin{bmatrix} 8 & 17 \\ 24 & 22 \\ 13 & 8 \end{bmatrix}$$

properties: Suppose A, B, C are $M \times N$ Matrices.

$$\begin{cases} \mathbb{B} + \mathbb{A} = \mathbb{A} + \mathbb{B} \\ \phi + \mathbb{A} = \mathbb{A} + \phi \\ \mathbb{A} - \mathbb{A} = \mathbb{A} + (-\mathbb{A}) = \phi \\ (\mathbb{A} + \mathbb{B}) + \mathbb{C} = \mathbb{A} + (\mathbb{B} + \mathbb{C}) \end{cases}$$

$$\begin{cases} (p+q)\mathbb{A} = p\mathbb{A} + q\mathbb{A} \\ p(\mathbb{A} + \mathbb{B}) = p\mathbb{A} + p\mathbb{B} \\ pq\mathbb{A} = (pq)\mathbb{A} \\ \mathbb{A} \cdot \mathbb{B} = \mathbb{C} \end{cases}$$

$$\begin{cases} \mathbb{A} = [a_{ij}]_{m \times n} \\ \mathbb{B} = [b_{kj}]_{n \times p} \end{cases}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

eg:

$$A = \begin{bmatrix} 3 & 1 & 2 \\ 2 & 6 & 5 \end{bmatrix} \quad b = \begin{bmatrix} 6 & 4 & 6 & 4 \\ 2 & 1 & 3 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix}$$

$$2 \times 3 \longleftrightarrow 3 \times 4$$

— Special Matrix.

$\phi = (\phi)_{M \times N}$

Identity Matrix.

$I_N = [\delta_{ij}]$, $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

eg

$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$\begin{cases} \mathbb{A}\mathbb{I} = \mathbb{A} \\ \mathbb{A}\mathbb{B}\mathbb{C} = \mathbb{A} \cdot (\mathbb{B} \cdot \mathbb{C}) \end{cases}$$

$$\begin{cases} \mathbb{I}\mathbb{A} = \mathbb{A}\mathbb{I} = \mathbb{A} \\ \mathbb{A}(\mathbb{B} + \mathbb{C}) = \mathbb{A} \cdot \mathbb{B} + \mathbb{B} \cdot \mathbb{C} \\ (\mathbb{A} + \mathbb{B}) \cdot \mathbb{C} = \mathbb{A} \cdot \mathbb{C} + \mathbb{B}\mathbb{C} \end{cases}$$

An $M \times N$ invertible.

$\mathbb{A} \cdot \mathbb{B} = \mathbb{B} \cdot \mathbb{A} = \mathbb{I}$.

— $\mathbb{B}$ is called an inverse of $\mathbb{A}$ or $\mathbb{A}^{-1}$

$\mathbb{A}^{-1} = \mathbb{A}^{-1} \cdot \mathbb{A} = \mathbb{I}$

Upper triangular system of Linear Equation.

· Matrix $A = [a_{ij}]$ is called upper trangular if $a_{ij} = \phi$ whe $i > j$

eg: $\begin{bmatrix} 5 & 3 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 4 \end{bmatrix}$

A matrix $A = [a_{ij}]$ is called a lower trangular if $a_{ij} = 0$ when $i < j$

eg: $\begin{bmatrix} 2 & 0 & 0 \\ 7 & 4 & 0 \\ 3 & 1 & 6 \end{bmatrix}$.

Upper trangular system of Linear Equation.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \cdots & \; a_{1N}x_N = b_1 \\ a_{22}x_2 + a_{23}x_3 \cdots & \; a_{2N}x_N = b_2 \\ a_{33}x_3 \cdots & \; a_{3N}x_N = b_3 \\ \vdots \quad & \quad \vdots \\ & \; a_{NN}x_N = b_N \end{aligned}$$

$A X = B$.

$\hat{}$ upper triangular.

To solve the system, use back substitution.

$$x_N = \frac{b_N}{a_{NN}}$$

$$x_{N-1} = \frac{b_{N-1} - a_{N-1N} x_N}{a_{N-1,N}}$$

$$\vdots$$

$$x_1 = \frac{b_1 - a_{1N}x_N - a_{N-1}x_{N-1} \cdots a_{12}x_{12}}{a_{12}}.$$

* Gauss Elimination

- It is a essential Method used to solve systems of linear equations
- It uses some transformation/equations on the equations that do not affect the system of linear equations
- Using these transformations it converts the ~~x~~ system to an upper ~~tra~~ triangular matrix that can be solved with back substitution

Transformations:

① Interchange.

The order of 2 equations can be changed

example

$x + 2y + 4z = \cancel{18}\ 19$
$8x + 9y + 3z = 33$
$x + y + z = 6$

② scaling:

We can multiply an equation by a constant & that will not affect the solution.

Example: * A by 2

$2x + 5y + 8z = 38$
$x + 9y + 3z = 33$
$x + y + z = 6$

③ Replacement.

An equation can be replaced by the sum of itself & one non zero multiple of another equation

example:

A

B+C

C

use these method to transfer to convert to a upper triangular ...

$1x + 3y + 4z = 19$
$8x + 9y + 3z = 35$
$x + y + z = 6$

$\mathbb{A}x = \mathbb{B}$

Augment Matrics

$$\left[\begin{array}{cccc} 1 & 3 & 4 & 19 \\ 8 & 9 & 3 & 35 \\ 1 & 1 & 1 & 6 \end{array}\right] \begin{array}{c} A \\ B \\ C \end{array}$$

We want A to be upper triangular to solve it usin back substitution.

Implementing Gauss Elimination in Matlab

```
function x = gauss(A, B)
    % input. A is a NxN nonsingular matrix
    %       B is a Nx1 matrix
    % output x is a Nx1 matrix with the solution Ax=B
    % get dimensio of A
    [N, N] = size(A)
    % initialize x with zeros
    x = zero(N,1).
    % obtain augmented matrix.
    Aug = [A B]
    % Gauss. limination
    % for all rows
    for p = 1:N
    % choose pivot, we ignore case when pivot is
      0 for simplicty
      piv = Aug(p,p)
      % when of the element in pivotal column
      for k = p+1 : N    % for all reminding rows
        m = Aug(k, p)
        for i = p:N+1
            Aug(K, i) = Aug(K, i) - m Aug(p, i) / piv
        end
      end        => O(N3)
    end
```

end end $\Rightarrow O(N^3)$

% back substitution

for k=N:-1

% accumulate all elements in upper trangolar

% upper trangolar matrix.

```
sum = 0
for j = k+1 : N
    sum = sum + Aug(k,j) * x(j,1)
end
x(k,1) = (Aug(k,k+1) - sum)/Aug(n,n)
end
```

$O(N^2)$

Notes.

→ If pivot is φ the switch rows to prevent divisions

→ to reduce the computational error you may use as the next pivot the largest number in the rows that are still left to trans

---

Triangular Factorization

A matrix has a triangular factorization if it can be expressed as the product of a lower triangular matrix L & upper triangular matrix U

$$A = L\,U$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ & & \cdots & \vdots \\ m_{N1} & m_{N2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ 0 & & & \\ \vdots & & & \\ 0 & & & u_{NN} \end{bmatrix}$$

Assume we have a linear system

$$AX = B$$

such that A has a triangular factorization

$A = L$ then

$$\mathbb{A}\mathbb{X} = \mathbb{B}$$

$$\mathbb{L}\,\mathbb{U}\cdot\mathbb{X} = \mathbb{B}$$

Then. We can define $\mathbb{Y} = \mathbb{U}_{n \times n}\mathbb{X}$ ①

then we have

$$\mathbb{L}\mathbb{Y} = \mathbb{B} \text{ ②} \qquad \mathbb{U}\mathbb{X} = \mathbb{Y}.$$

Then we can solve $\mathbb{X}$ by first solving y.. using ②

then. solving $\mathbb{X}$, using ①

This will help us solving multiple systems. of linear equations, of the form

$$\mathbb{A}X_1 = \mathbb{B}_1$$
$$\mathbb{A}X_2 = \mathbb{B}_2$$
$$\mathbb{A}X_3 = \mathbb{B}_3$$
$$\mathbb{A}X_m = \mathbb{B}_m$$

The A is the same in all of the systems.

only $\mathbb{B}$ changes

To solve them, we first find the $\mathbb{L}\,\mathbb{U}$ factorization of $\mathbb{A}$.

$O(n^3)$ $\mathbb{A} = \mathbb{L}\mathbb{U}$ $\qquad \mathbb{A}\mathbb{X}_1 = \mathbb{B}_1$

$$\mathbb{L}\underbrace{\mathbb{U}\mathbb{X}_1}_{\mathbb{Y}} = \mathbb{B}_1$$

Then for every $B_1 \dots B_m$

solve

m times $O(n^2)$ $\mathbb{L}\mathbb{Y}_i = \mathbb{B}_i$.

and then solve $\mathbb{X}_m$

m times $O(n^2)$ $\mathbb{U}\mathbb{X}_i = \mathbb{Y}_i$

If we use only Gauss elimination for the m system of equation.

It will cost $m\,O(N^3) = O(mN^3)$

If we use triangular factorization it will take

$$O(N^3) + mO(N^2)$$
$$= O(N^3) + O(mN^2)$$
$$= O(N^3 + mN^2)$$

- How do we do ~~GE~~ LU factorization.
- similar to Gauss elimination

Assume. we want to solve the equation

$$6x_1 + x_2 - 4x_3 = 3$$
$$5x_1 + 3x_2 + 2x_3 = 21$$
$$1x_1 \pm 4x_2 + 3x_3 = 10$$

&

$$6x_1 + x_2 - 4x_3 = 3.$$
$$5x_1 + 3x_2 + 2x_3 = 10$$
$$1x_1 - 4x_2 + 3x_3 = 0$$

Both systems use same A,
we do the factorization IA as follows

$$\underset{I}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \underset{A}{\begin{pmatrix} 6 & 1 & -4 \\ 5 & 3 & 2 \\ 1 & -4 & 3 \end{pmatrix}} \begin{array}{l} A \\ B \\ C \end{array}$$

Do the same in gauss elimination to make a upper triangular. and pass pivot division m. lest matrixs

$$\begin{array}{l} \\ m_2 \rightarrow \\ m_3 \rightarrow \end{array} \begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/6 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 1 & -4 \\ 0 & \frac{13}{6} & \frac{32}{6} \\ 0 & -\frac{25}{6} & \frac{22}{6} \end{bmatrix} \begin{array}{l} A \\ B - A\left(\frac{5}{6}\right) \; m. \\ C - A\left(\frac{1}{6}\right) \; m. \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/6 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 1 & -4 \\ 0 & \frac{13}{6} & \frac{32}{6} \\ 0 & -\frac{25}{6} & \frac{22}{6} \end{bmatrix} \begin{array}{l} A \\ B \\ C \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/6 & -\frac{25}{13} & 1 \end{bmatrix} \begin{bmatrix} 6 & 1 & -4 \\ 0 & \frac{13}{6} & \frac{32}{6} \\ 0 & \cancel{-\frac{25}{6}}\ 0 & \frac{1080}{78} \end{bmatrix} \begin{array}{l} A \\ B \\ C - B\left(-\frac{25}{13}\right) \\ \cancel{C - B} \quad C - BC \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/6 & -\frac{25}{13} & 1 \end{bmatrix} \begin{bmatrix} 6 & 1 & -4 \\ 0 & \frac{13}{6} & \frac{32}{6} \\ 0 & 0 & \frac{1080}{78} \end{bmatrix}$$

solving

$6x_1 + x_2 - 4x_3 = 3$

$5x_1 + 3x_2 + 2x_3 = 21$

$1x_1 - 4x_2 + 3x_3 = 10$

solve $Ly = b$

$$\begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/6 & -\frac{25}{13} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 21 \\ 10 \end{bmatrix}$$

$y_1 = 3$

$y_2 = \left[21 - \frac{5}{6}(3)\right]\frac{1}{1} = \frac{111}{6} = 18.5$

$y_3 = 10 - \frac{1}{6}(3) + \frac{25}{13}(18.5) = 10 - \frac{1}{2} + \frac{25 \cdot 18.5}{13} = 45.0769$

solve $Ux = y$

$$\begin{bmatrix} 6 & 1 & -4 \\ 0 & 13/6 & 32/6 \\ 0 & 0 & \frac{1096}{78} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 18.5 \\ 45.0964 \end{bmatrix}$$

$x_3 = 45.0964 \left(\frac{78}{1096}\right) = 3.2376$

$x_2 = 18.5 - \frac{32}{6} \cdot (3.2376)\frac{6}{13} = 0.5695$

$x_1 = \left[3 - (-4)(3.23767) - (1)(0.5695)\right]\frac{1}{6}$

$= 2.5635$

Do the same for the other system

$6x_1 + x_2 - 4x_3 = 3$

$5x_1 + 3x_2 + 2x_3 = 10$

$1x_1 - 4x_2 + 3x_3 = 0$

Iterative Methods for Linear Equations

Consider the system of Equations:

$3x + y = 5$

$x + 3y = 7$

We can create the following iterations

$3x + y = 5 \Rightarrow x_{k+1} = \frac{5 - y_k}{3}$

$x + 3y = 7 \Rightarrow y_{k+1} = \frac{7 - x_k}{3}$

$x_0 = 0, \quad y_0 = 0$

$x_1 = \frac{5-0}{3} = 1.667 \quad y_1 = \frac{7-0}{3} = 2.33$

$x_2 = \frac{5-2.33}{3} = .889, \quad y_2 = \frac{7-1.667}{3} = 1.77$

$x_3 = \frac{5-1.77}{3} = 1.074 \quad y_3 = \frac{7-.889}{3} = 2.037$

$x_4 = \frac{5-2.037}{3} = .0987 \quad y_4 = \frac{7-1.074}{3} = 1.979$

$x_n = 1 \qquad y_n = 2$

Jacobi Iteration.

Gauss Seidel iteration

We can speed up the convergence by using the values of the current iteration as they are obtained. This is called Gauss Seidel

Jacobi

$x_{k+1} = \frac{5 - y_k}{3}$

$y_{k+1} = \frac{5 - x_k}{3}$

Gauss Seidel

$x_{k+1} = \frac{5 - y_k}{3}$

$y_{k+1} = \frac{7 - x_{k+1}}{3}$

If we do M iterations, the cost will be $O(MN^2)$
since Gauss Elimination takes $O(N^3)$
then guess seidel or Jacobi are better
When $M < N$.

This Happens in spare matrices that have many zeros in their entries: the # of entries with $\emptyset$ is $O(N)$

- Example:

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & a_{nn} \end{bmatrix}$$

- These matrices are common in problems where there are many equations and each equation represents one component and its neighbors.
- If M is very large $\approx 10'00,000$
- ~~It~~ Gauss: $O(N^3) = ((10^6)^3)$ 10^6.

Jacobi:

in spare matrix the # of elemens that are none zero is $O(N)$

$\Rightarrow$ 1 iteration $O(N)$

M iterations: $O(MN)$

In this case use Jacobi could be faster that Gauss elimination

Lagrange Approximation

polinomial approximation

- Assume that we have 2 points (x_0, y_0) & (x_1, y_1). The polynomial that passes through these 2 points is a straight line

① $m = \frac{y-y_0}{x-x_0}$ & ② $m = \frac{y_1-y_0}{x_1-x_0}$

from ① & ②

$$\frac{y-y_0}{x-x_0} = \frac{y_1-y_0}{x_1-x_0}(x-x_0)$$

$$y-y_0 = \frac{y_1-y_0}{x_1-x_0}(x-x_0)$$

$$y = y_0 + \frac{y_1-y_0}{x_1-x_0}(x-x_0)$$

* lagrange Approximation uses a different method to buil[d] a polynomial

$$p_1(x) = y = y_0 \underbrace{\frac{x-x_1}{x_0-x_1}}_{\substack{0 \text{ when } x=x_1 \\ 1 \text{ when } x=x_0}} + y_1 \underbrace{\frac{x-x_0}{x_1-x_0}}_{\substack{0 \text{ when } x=x_0 \\ 1 \text{ when } x=x_1}}$$

$$p_1(x_0) = y_0 \cdot \left(\frac{x_0-x_1}{x_0-x_1}\right) + y_1\left(\frac{x_0-x_0}{x_1-x_0}\right).$$

$$p_1(x_0) = y_0$$

For $P_2(x) = (x_0, y_0), (x_1, y_1), (x_2, y_2)$.

$$p_2(x) = y_0 \cdot \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \cdot \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \; y_2 \cdot \frac{(x-x_1)(x-x_0)}{(x_2-x_1)(x_2-x_0)}$$

In General:

$$P_n(x) = \sum_{k=0}^{n} y_k L_{n,k}(x).$$

Newton Polynomials.

- In Lagrange polynomials
  $p_{N-1}(x)$ & $p_N(x)$ are not related
- Newton Polynomials can be built incrementally so the work can be done to build $p_{N-1}(x)$ can be used to build $p_N($

. Assume

$$p_1(x) = a_0 + a_1(x-x_0)$$
$$p_2(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1)$$
$$p_3(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-$$
$$\vdots$$
$$p_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) \cdots a_N(x-x_0)\cdots($$

$p_N(x)$ is the Newton polynomial of degree N.

How to compute $a_0, a_1 \cdots a_N$?
Assume. We want to build $p_1(x)$ with the point $(x_0, F(x_0$
$(x_1, F(x_1))$

$$p_0(x_0) = f(x_0) \qquad p_0(x_1) = f(x_1)$$
$$f(x_0) = p_1(x_0) = a_0 + a_1(x_0 - x_0) = a_0$$
$$f(x_0) = a_0. \quad ①$$
$$p(x_1) = f(x_1) = a_0 + a_1(x_1 - x_0)$$
$$\text{from①} \quad f(x_0) = f(x_0) + a_1(x_1 - x_0)$$
$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad ②$$

So we have that.

$$p_1(x_1) = a_0 + a_1(x - x_0) \quad a_0 = f(x_0), \ a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

With the same method, we build ~~x~~ $a_2, a_3 \cdots$

The divided difference of a function $f(x)$ are defined as

- $f[x_k] = f(x_k)$
- $f[x_{k+1}, x_k] = \dfrac{f[x_k] - f[x_{k-1}]}{x_k - x_{k+1}}.$
- $f[x_{k-2}, x_{k-1} x_k] = \dfrac{f[x_{k-1}, x_k] - f[x_{k-2}, x_{k-1}]}{x_k - x_{k-2}}.$

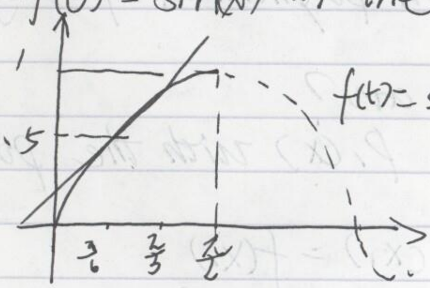
$\vdots$

$f[x_{k-j}, x_{k-j+1} \cdots x_k] = \dfrac{f[x_{k-j+1} - x_k] - f[k-j - x_{k-j-1}] \cdots}{x_k - x_{k-j}}.$

Where $a_k = f[x_0, x_1 \cdots x_k]$

*Example:

Build polynomials of degree 1, 2, 3 to approximate $f(x) = \sin(x)$ in the interval $(0, \frac{\pi}{2})$



Divided Difference Table.

x_k	$f[x_k]$	$f[x_{k-1}, x_k]$	$f[x_{k-2}, x_{k-1}, x_k]$
0	0		
$\frac{\pi}{6} = 0.5836$	0.5	$\dfrac{0.5 - 0}{0.5836 - 0} = 0.9549$	$\dfrac{.699 - .9549}{1.047 - 0} = -.02?$
$\frac{\pi}{3} = 1.047$	$.8659$	$\dfrac{.8659 - .5}{1.047 - .8659} = .699$	$\dfrac{.256 - .699}{1.5708 - .5236} = -.4230$
$\frac{\pi}{2} = 1.5718$	1	$\dfrac{1 - .8659}{1.5408 - 1.043} = 0.956$	

$p_2(x) = \dfrac{a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)}{p_1(x) + a_2(x - x_0)(x - x_1)}$

$= 1.082x - 0.2444x^2.$

$p_3 = a_0 + a_1(x - x_0) \ldots a_3(x - x_2) \;/\; x, x, (x - x_0$