NUMERICAL ALGORITHMS

GEORGE SALMAN

E-mail address: George.sa@campus.technion.ac.il.

Remark. These notes are based on Numerical Algorithms course given at the Technion. Please note this text may contain error. If something not clear don't hesitate to mail the provided Email.

Introduction.

Contents

Introduction. 1
Numerical Linear Algebra (NLA) 2
Fields that require NLA 3
Is the course relevant? 6
Example of Application. 7

LU & Cholesky Decomposition 16
Remainder: Gauss elimination. 16
LU-Decomposition. 18
Comparsion. 21
Non regular matrices. 21
LDV decomposition. 26
Reminder (inituition for Gram matrix). 28
Gram matrices. 34
Cholesky decomposition. 35

Least Squares. 36
How to solve the LS? 38
Deffrentiating vector matrices. 40
Application to least squares: Matching to curve to a data 42
Weighted Least Sqauares. 47
regularization to LS problems. 48
Example. (Noise reduction from voice signal). 50

Orthogonal matrices and QR decomposition 52
Orthonormal basis. 53
Gram-Schmidt 56
Remainder from Algebra. 57
GS Psuedo-Code. 59
Gram-Schmidt Algorithm 59
Modified Gram-Schmidt 64

Stable Gram-Schmidt (SGS) 65
Complexity of GS. 66
Numerically sensitivty of GS 66
QR Decomposition. 69

Abstract. What is numerical analysis? umerical analysis is the study of **algorithms** that use numerical approximation (as opposed to symbolic manipulations) for the problems of **mathematical analysis** (as distinguished from discrete mathematics). Numerical analysis finds application in all fields of engineering and the physical sciences, and in the 21st century also the life and social sciences, medicine, business and even the arts. Current growth in computing power has enabled the use of more complex numerical analysis, providing detailed and realistic mathematical models in science and engineering. Examples of numerical analysis include: ordinary differential equations as found in celestial mechanics (predicting the motions of planets, stars and galaxies), numerical linear algebra in data analysis, and stochastic differential equations and Markov chains for simulating living cells in medicine and biology.

Topics.

Solution of equation system, optimization e.g., we have a multi-dimension with $10,000$ variables function and if we plug-in the $10,000$ variables we obtain scalar which is minimim height.

Numerical Linear Algebra (NLA). Numerical Linear Algebra deal with algorithms which execute instruction taken from linear algebra. (e.g., matrices, vectors).

What this means?

(1) NLA is not a subsection in numerical analysis, it touch everything, where the main focus on tools - matrices, vectors.
(2) The field has many intriguiting applications.

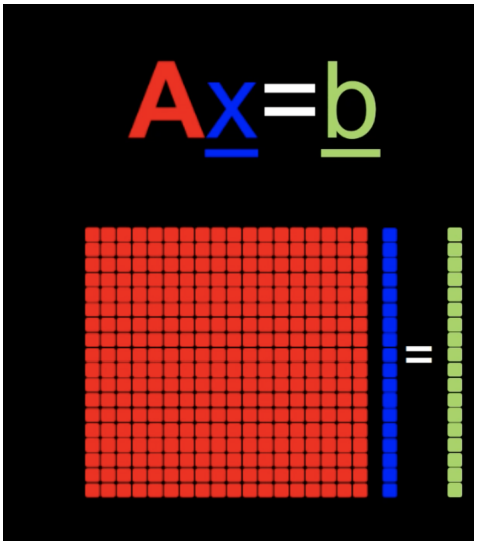
Plan. We will start with system of linear equation and its solution, optimization, eigenvalues, and eigenvectors. We will talk about intrepolation methods.

Fields that require NLA.

- Networking
- Robotics
- Signal and Image proccessing
- Weather Forecast
- Encryption

In all this fields and many othere, NLA is a saginificant tool. In many cases engineers and researchers are asked to solve such a problem e.g., identify whether if a ZIP file testifies happiness or sadness. This engineer is lucky because he knows NLA.

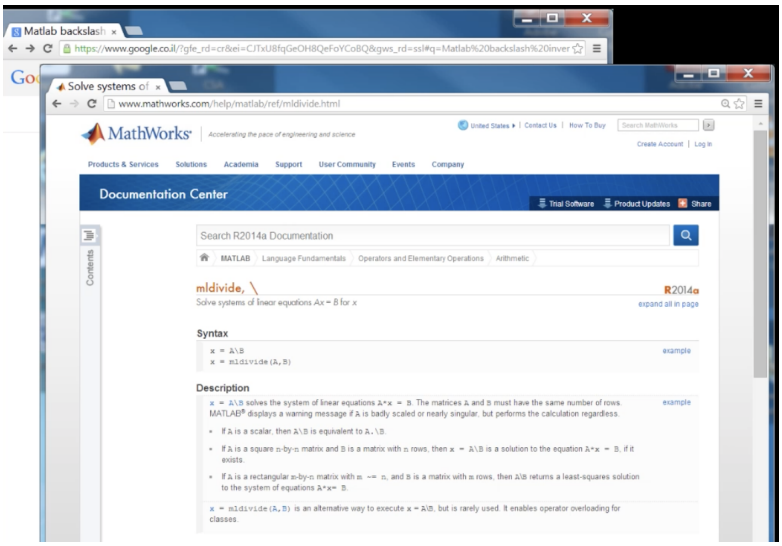
Highlights in the course.



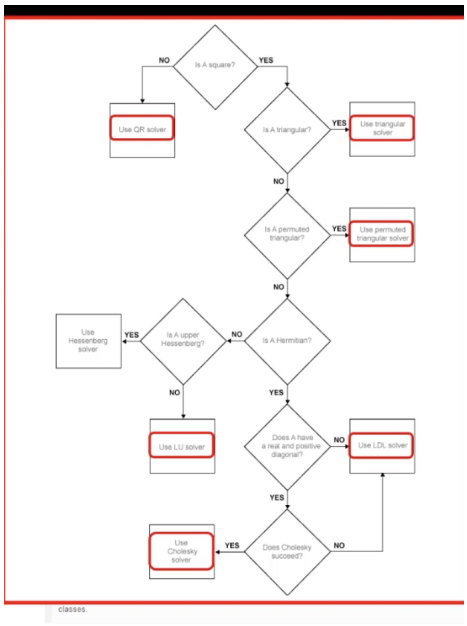
Solving system of linear equation. Assuming we want to find x which is solution. As we did in Linear Algebra one, we can find the solurion using the inverse matrix but it will cost us too much, therefore we are going to see different numerical approach.

(1) System of n variables with n variables is the power of engineers on all its aspects.
(2) We are interested in this question - How to solve the problem using computer.
(3) Its obvious that the solution to the question depend on the properties of A and in the dimentsion of the problem hence many options are offered.

Matlab command that find $\vec{x}$ (solution for $Ax = b$).



We can see that using the slash command we can find x. How Matlab tackle the problem.



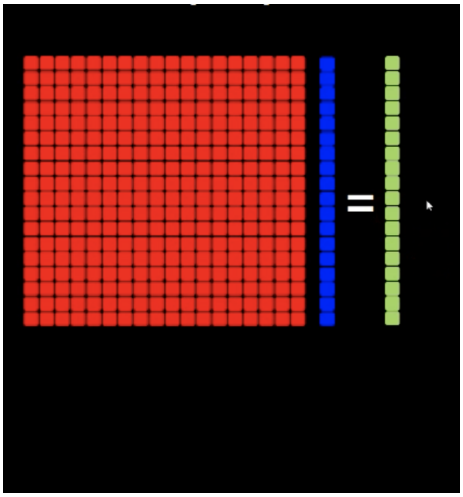
So what Matlab do is the procedure above, first checking if A is square matrix? If not then do QR decomposotion. If yes, then check if A is triangle matrix, if not then it check many conditions, all the rectangles above are algorithms which we are going to see throughout the course.

Main Question in Computer science. Given a matrix A and vector b and we are interested in investigate numerical errors and the influence of finite word length in the solution of linear equations system i.e.,

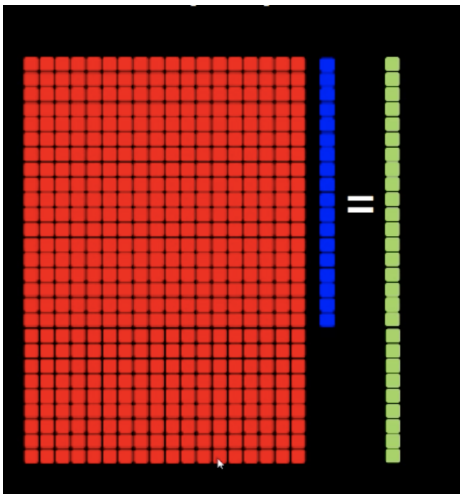
$$Ax = b \Rightarrow (A + E)x = b + e$$

Note that we are not finding a real solution but rather a approxiomate solution since we are solving a system in which A, b are deviated matrix, so we will get deviated solution but we will try to find optimal upper bound for x.

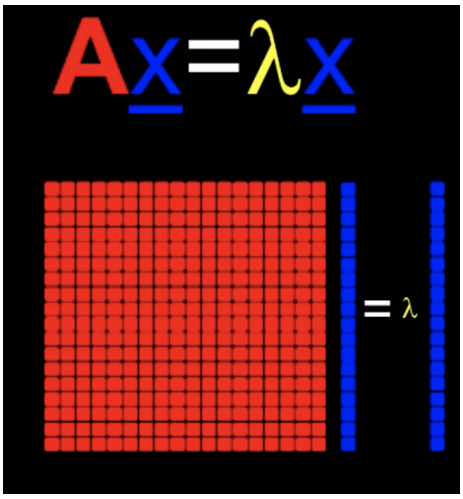
Main things.



Assuming we got a system of equations as above, and the person give us the system is so generous and give us more equations than required on the same set of variables as follows,



We can solve it for minimzied system i.e., pick 100 equation from the 200 given but in this case there is no such x which is agreed upon all the system for any set of equations we pick, so the strategy is to find the most optimal approximated solution i.e., in other words we ask what x should be in which lead the both sides are the most close this is optimization problem $Ax \approx b$ (least squares). Other probelm we are going to see is as follows,



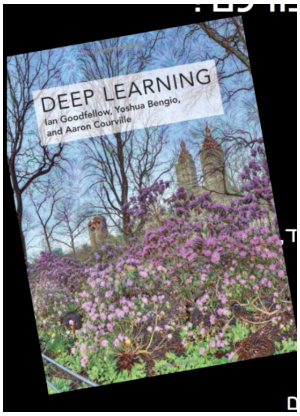
This problem is finding a eigenvalue\vector for a matrix A so we are intersted in finding such a vector x in which multiplting it with the matrix A will affect him merely byscalar mutliplication, again the size of the system will have a implication in terminology of which stratedy to use . A private case where A has a rotation structure so there is connection to well known operation which is called convulotion which will lead us to Fourier and Laplace transformations.

Is the course relevant? What is the most common field in the computer science recently?

Hints.

- Facebook use it intensively.
- Google use it obsessively
- Likewise Intel, Apple, Twitter, NVIDIA..

Answer: Maching learning and Deep learning. In fact the fields spread and did a great revolutionary. We hear about autonomous cars, or new medicine based on data which obtain diagnose those are example of Deep learning.



The picture above is not real i.e., not captured by a camera rather illusion of a computer (done using Deep learnign) by letting Network generating it.

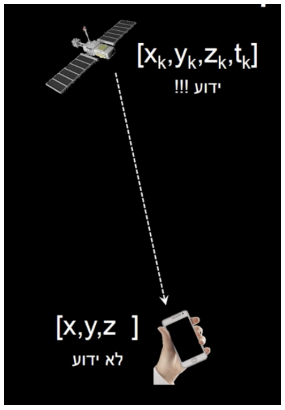
Example of Application.

Example 1: (GPD) is a popular global navigation tool using satalites used today and available almost in every cellphone and camera, how it works?

USA goverment launched to space 31 satalites hovering in space relative to specific known position relative to the cooadination of the earth.

- GPD device get a signal from a satalite (there are 31) with a packet of data includes:
  - The identity of the satalite (serial number k)
  - Cooardination of the satalite in space (x_k, y_k, z_k) relative to the origin of the earth $(0, 0, 0)$
  - Time point t_k in which the information was broadcasted relative to a accepted global clock

Remark. Each satalite know at any time its x_k, y_k, z_k and t_k which is critical and taken from a ground stations which broadcast to the satalites a synchonpus time which all the satalites share, and also the broadcast time is considered i.e., the time took (Sending t_k n from a ground stations to the satalites). In total all the satalites are known to their timing.



We people on earth are interested to know our position so what we do is turning the GPD option on our cellphones and recive signals from multiple Satalites and using those satalites we want to know our $[x, y, z, t]$ relative to the earth origin. Likewise, we will assume that relative to the same uniersal clock, our time point which we recieve all the satalites is t (other variable) so we obtain system of 4 equation with 4 variables. In mathmatic formulation we can present the problem as follows, first the distnace between the satalite and our device cooardination (x, y, z) presented us,

$$(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2$$

The same distance we can write in other way since the time is know for the device and for the satalie miltiplied by the broadcasting speed c (light speed) hence, the distance is equal to $c^2(t_k - t)^2$ where the term present the distance from the satalite to the device by the difference of times from broadcasting to recieving the signal on earth, so what we can do to obtain 4 equations is to use 4 satalites and get 4 equation with 4 variables however its very risky approach since those equations are not convenient to work with. Hence, other approach is first the first satalite we recieve signal from will be chosen to be the satalite in which we are going to substract other satalites k so we obtain a linear equation system with 4 variables,

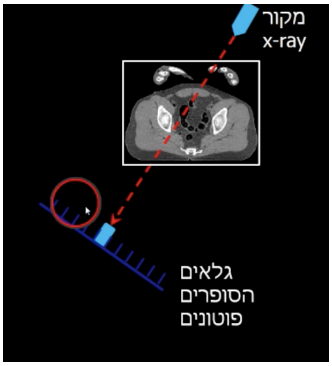
$$\begin{gathered}(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2 = c^2(t - t_k)^2 \\ - \quad (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 = c^2(t - t_1)^2 \\ \hline = (x_1 - x_k)\, x + (y_1 - y_k)\, y + (z_1 - z_k)\, z + c^2(t_k - t_1)t = F_{1,k}\end{gathered}$$

So we get a linear system with variables x, y, z, t with known coeffcient, so we can take 5 satalites in which one we exploit as reference and the other are used to generate four equation. However, for more accuracy we can see that more than 5

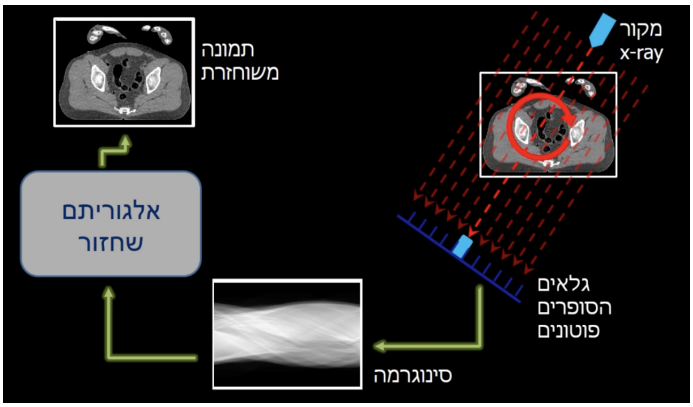
satalites used sometimes $6, 7, 10, 12$ and we can use all of them to take a solution which given as $Ax \approx b$ as much good as possible (i.e., least squares case).

Example 2: CT (computer Tomography) is a method to capture a human bodt with no harm. Exists in every hospital which could diagnose diseases at early stages, how it works?

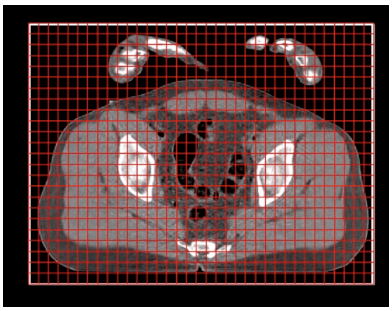
A branch of medicine that uses imaging technology to diagnose using this pictures and treat disease (radiology). We can see in pictures above a cut of head, and a body in which we can use to diagnose. How the camera works? given a person chill on a bed a X-ray radiation is sent with enormous amount of photons, some of them abosrbed to the body and the others reach the other side to the photons detector which till us how many of them survived the path as in the following picture,



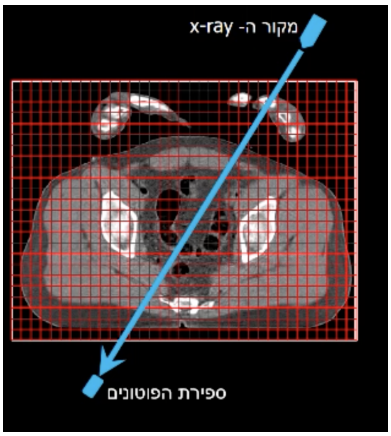
Assuming we start with $1m$ photons and the detector detect $10k$ it tell us many stories about the photon pass which will help us to extract the image. For example, if we spread $10k$ photons in parallel and when we finish we get x-ray which is projection of x-ray and get the picture, CT doesn't stop here, it rotate in one degree and capture again, we can see the circle in the machine which contain a projector that spread photons which rotate around the person which generate the infrastructure in which we will use to get a senograma that has all the information about the picutem and using retriveing algorithm we can get the image.



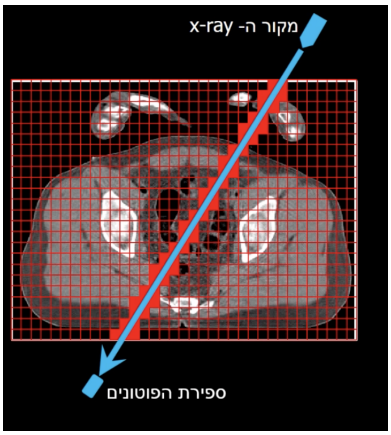
How the retriving algorithm works? At the beginning we take the image in which we want to retrieve and divide it into small rectangles NxM where $N = M = 1000$ i.e., $1m$ rectangles



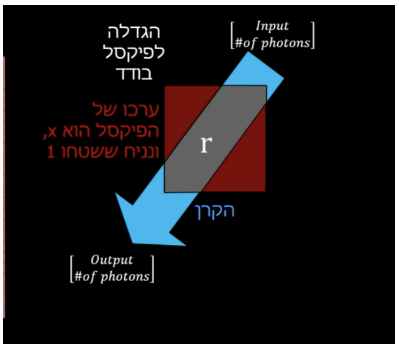
Each rectangles (pixel) will get a serial number which tell us how much tissue is dense in it. E.g., if it in region of bone then 1, in a fat region 0.2, in space region 0. The variables in this case is those numbers i.e., $1m$ variables since for each pixel we should tell its RGB band from WHITE (which is 1) and BLACK (which is 0). Our problem is to find NM variables which construct the image. We will see the tomographic retriving i.e., find those values is not more than finding the solution to the system of equation.



Assuming that certian photon cross number of pixels and other pixels don't so they have no contribution as in the following picture,



So we can see the the photon cross number of pixels marked in red, now each pixel absorb a certain amount of photons the more he absorb the more dense the tissue is (e.g., black color absorb the most photons when White almost doesn't absorb) hence, for each pixel we can intriduce the following relation,



$$[\text{output of number of photons}] = [\text{input of number of photons}] \cdot e^{-xr}$$

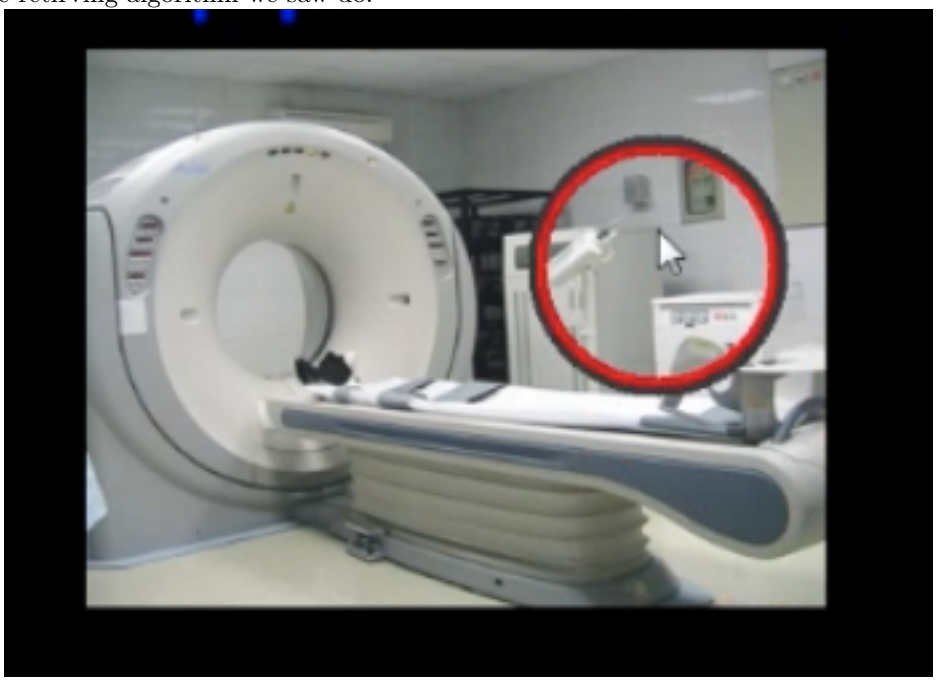
The bigger x is the more absorbtion for certian pixel. So we can obtain the following equation,

$$P_{out} = P_{in} \cdot e^{-x_1 r_1} \cdot e^{-x_2 r_2} \cdots e^{-x_n r_n}.$$

$$= P_{in} \cdot e^{-\sum_{k=1}^{n} x_k r_k}$$

$$\Rightarrow -log\left(\frac{P_{out}}{P_{in}}\right) = \sum_{k=1}^{n} x_k r_k$$

Since the left side of the equation is known, (We know number of photons got in and got out) so we can spread as much photons since each photon construct one equation and since we have $1m$ variables we can do more than $1m$ photons to get more accurate result and we are back to the least squares case $Ax \approx b$ this is what the retirving algorithm we saw do.



Interesting question. When a human enter the CT room he may assume that the main part of the machine is the circle, however, this circle is not more that a machine which spread photons on different rotations and abosrb. The main path of the machine is the two boxes (marked in red circle in figure above) which the first one contain GPU and other box run the software generating the image by running the heavy computational part based on the mentioned retriving algorithm.

.

Example 3: (Classification) assuming we interested in taching the computer identifying between picture taken for male or female i.e., constructing an algorithm that can classify a picture of face into two catagories, how this works?

Remark. This mission a person do it automatically in life using what we see. The mentioned probem and the method presented to solving it are related to the field of machine learning where NLA tools are used profusely.

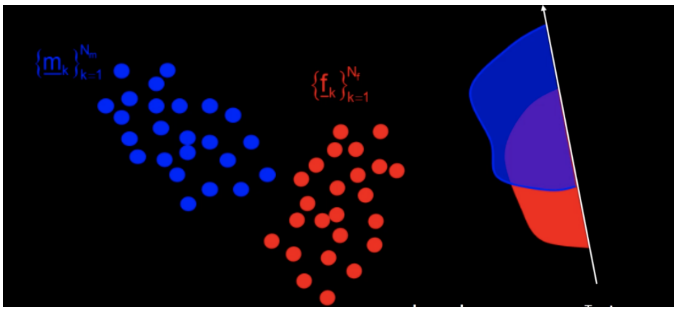
Core Assumptions. Assuming that the algorithm require to treat images in constant size of $100x100$ *pixels representing a person face. Hence each example is* $10,000$ *numbers entered to PC for identifying purpose.*

Training examples. In our hands there is set of hundred thousand images in which for each of them known its gender representing (i.e., male or female).

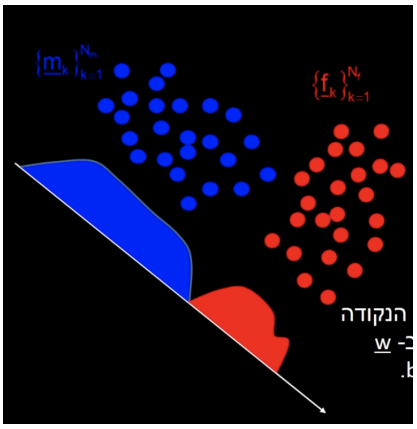
Notation. $\{f_k\}_{k=1}^{N_m}$ is trainging images for female, and $\{m_k\}_{k=1}^{N_f}$.

Linear classification. Given a new picture x (again vector in length $10k$) the wat to determine the gender of it will be found ($C < 0$ negative-men, $C > 0$ positive women) of $sgn\left(w^T x - b\right) = C$ (scalar) so the main work here is to manage b, w based on the samples we got in our hand.

Linear Projection. The term $x^T x - b$ mean theat the projectopn of a vectpr x to real number by multiplying it with w, and check its values whether bigger or less then b, note that the vector w determine the direction of the projection, our target is to split as much a possible the values we get from $w^T x$ where x is a picture, note that each picture in $mat_{nxm}(\mathbb{R}) \cong \mathbb{R}^{n+m}(vector)$ i.e., we can represent picture as vector, hence we can do this for all trainnign pictures and getting values in $\mathbb{R}$, our target is to obtain values for men which are close to each and for women, but both set are not overlap, as in this figure,



Our target is to obtain w which give the less overlapping between to sets e.g., w as follows,



We want direction of projection w that minimze all the red elements in a small interval on the projection axes, where red elements and blue are most far from other as above, so how we are going to find that w? since we require w in which the distance between red element set and blue element set is the biggest this is equivalent to,

$$\min_{\vec{w}} \left[\frac{\sum_{k=1}^{N_f}\sum_{j=1}^{N_f} \left(w^T f_k - w^T f_j\right)^2 + \sum_{k=1}^{N_m}\sum_{j=1}^{N_m} \left(w^T m_k - w^T m_j\right)^2}{\sum_{k=1}^{N_f}\sum_{j=1}^{N_m} \left(w^T f_k - w^T m_j\right)^2} \right]$$

To understant the intiuition behind this mathamatical formulation we will try to undersstand the idea discussed. First, the term $\sum_{k=1}^{N_f}\sum_{j=1}^{N_f} \left(w^T f_k - w^T f_j\right)^2$ tell us that take minimal distance between points in the female set, the same $\sum_{k=1}^{N_m}\sum_{j=1}^{N_m} \left(w^T m_k - w^T m_j\right)^2$ tell us if we minimze it then taking minimal distance between points in the male set ensure us to get as much closed point as possible in the same set, and dividing $\sum_{k=1}^{N_f}\sum_{j=1}^{N_m} \left(w^T f_k - w^T m_j\right)^2$ which tell us take two points of each set and stretch them to get the most distnace between two set since higher quontient yields less values, now using algebric trick we obtain,

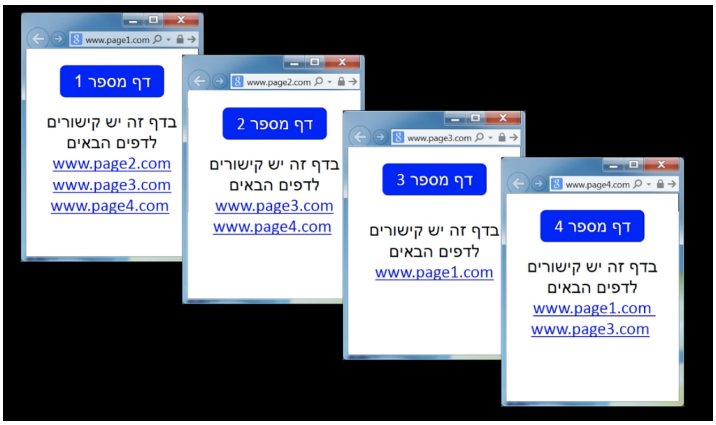
$$\min_{\vec{w}} \left[\frac{\sum_{k=1}^{N_f}\sum_{j=1}^{N_f} \left(w^T f_k - w^T f_j\right)^2 + \sum_{k=1}^{N_m}\sum_{j=1}^{N_m} \left(w^T m_k - w^T m_j\right)^2}{\sum_{k=1}^{N_f}\sum_{j=1}^{N_m} \left(w^T f_k - w^T m_j\right)^2} \right]$$

$$= \min_{\vec{w}} \left[\frac{w^T \left[\sum_{k=1}^{N_f}\sum_{j=1}^{N_f} \left(f_k - f_j\right)\left(f_k - f_j\right)^T + \sum_{k=1}^{N_m}\sum_{j=1}^{N_m} \left(m_k - m_j\right)\left(m_k - m_j\right)^T \right]}{w^T \left[\sum_{k=1}^{N_f}\sum_{j=1}^{N_m} \left(f_k - m_j\right)\left(f_k - m_j\right)^T \right]} \right]$$

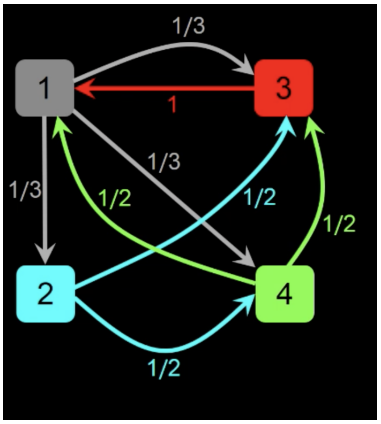
$$= \min_{\vec{w}} \left[\frac{w^T R w}{w^T Q w} \right]$$

This is eigen values problem $Ax = \lambda x$ if we know to solve it then we get the optimal w this problem known as (FLD) abbreviation to Fisher's Linear Discriminant.

Example 4: (page rank) in the 90's a search in internet was done by the matching to the seaching words and word in sites possbile, this leads to many wrong results since usally the results were non important sites. The solution to the problem was proposed in one of Msc works at Stanford by Larry Page (the lecturer friend). How it works?



Assuming we have only 4 sites with content and hyperlinks to one site to other, which site is the most important? let try to rank them by priority, we cab solve the problem by the graph, a site priotiry determined using how much sites are link to it, so the garph we obtain will look as follows,



We can see that the equation are,

$$\begin{aligned} x_1 &= \tfrac{1}{1}x_3 + \tfrac{1}{2}x_4 \\ x_2 &= \tfrac{1}{3}x_1 \\ x_3 &= \tfrac{1}{3}x_1 + \tfrac{1}{2}x_2 + \tfrac{1}{2}x_4 \\ x_4 &= \tfrac{1}{3}x_1 + \tfrac{1}{2}x_2 \end{aligned}$$

Hence, we get a matric

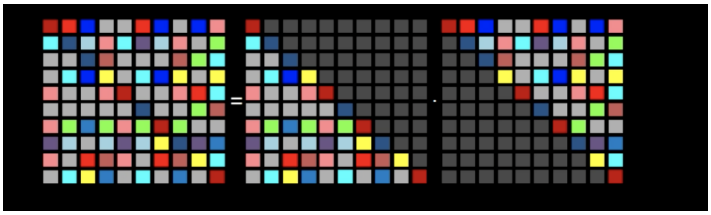
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0.5 \\ 0.33 & 0 & 0 & 0 \\ 0.33 & 0.5 & 0 & 0.5 \\ 0.33 & 0.5 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Note that we always get that 1 is eigenvalue because we observe that the sum of all columns is 1. The problem is equivalent to find eigenvector x for $Ax = \lambda x$ correspond to it, note that this eigen vector is solution and λx is solution, however, in Google the values are normalized to 10 and the matrix is way bigger, and the solution is $x_1 = 0.387, x_2 = 0.129, x_3 = 0.290, x_4 = 0.194$ so what Google do is multiplying the vector $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$ by 10 and get $\begin{pmatrix} 3.87 \\ 1.29 \\ 2.9 \\ 1.94 \end{pmatrix}$ we can see that sum is $\approx$ 10 and x_1 is the most valuable i.e., site 1 is the most important.

LU & Cholesky Decomposition

We want to show that every matric could be written as multiplication of up triagnle matrix and down triangle matrix, this will help us in:

- Solve system of equation
- Caculate determinant of matric.
- Determine a rank of matrix



Remainder: Gauss elimination. We are using simple operation on the extened matrix $[A, b]$ of the problem to solve system of linear equations,

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 7 \\ 1 \end{pmatrix}$$

$$\Rightarrow [A, b] = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 2 & 6 & 1 & 7 \\ 1 & 1 & 4 & 3 \end{array}\right)$$

Now, doing $R_2 \to R_2 - R_1, R_3 \to R_3 - R_1$ yields,

$$[A, b] = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & -1 & 3 & 1 \end{array}\right)$$

Now, $R_3 \to R_3 + 0.5R_2$

$$[A,b] = \left(\left|\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & 0 & 2.5 & 2.5 \end{array}\right.\right)$$

Hence, $z = 1, y = 2, x = -3$. Now, this operation could be done by mutltiplying in elementary matrix for example the first operation $R_2 \to R_2 - R_1$ on the extended matrix is equialvent to multiply the extended matrix with the unit matrix in which we will apply the gauusain operaion $R_2 \to R_2 - R_1$ i.e.,

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

So, multiplying it with the extended matrix yields,

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \left(\left|\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 2 & 6 & 1 & 7 \\ 1 & 1 & 4 & 3 \end{array}\right.\right) = \left(\left|\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -1 & 3 \\ 1 & 1 & 4 & 3 \end{array}\right.\right)$$

So in order to make 0 in the third raw first column we will multiply by elementary matrix again E_2,

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}}_{E_2} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{E_1} \left(\left|\begin{array}{ccc|c} \underbrace{1}_{Pivot} & 2 & 1 & 2 \\ 2 & 6 & 1 & 7 \\ 1 & 1 & 4 & 3 \end{array}\right.\right) = \left(\left|\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & -1 & 3 & 1 \end{array}\right.\right)$$

Remark. In this two operation we relied on a pivot element which is 1 and marked above with the word pivot in which using it we obtain 0 in the first cooardination in other raw.

Now, again we want to obtain 0 in the third raw second column by multiplying with the elementary matrix,

$$E_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{pmatrix}$$

where now that pivot element we used to obtain that is $a_{22} = 2$ in the result matrix $\left(\left|\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & -1 & 3 & 1 \end{array}\right.\right)$ so those matrices E_1, E_2, E_3 has a special structure,

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_2 = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}}$$

$$E_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{pmatrix}$$

Observations.

(1) All the diagonal elements are 1 - because for each raw the operations preserve the raw and add to it other raw multiplied by scalar.
(2) Those are lower triangle matrices - because we always add to a such a raw other raws above it.

We will call those matrices by **basic lower triangle matrices**

What about the inverse matrices of those elementary matrices, notice that in order to cancel the operation for example E_1 which is $R_2 \to R_2 - 2R_1$ we will do $R_2 \to R_2 + 2R_1$ i.e.,

$$E_1^{-1} = L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

.

$$E_2^{-1} = L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$E_3^{-1} = L_3 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & -0.5 & 1 \end{pmatrix}$$

Notice that the inverse matrices are also **basic lower triangle matrices.**

LU-Decomposition. We started by taking the system $Ax = b$ and convert it to a equivalent system,

$$E_3E_2E_1x = E_3E_2E_1b$$

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 2.5 \end{pmatrix}$$

Note that we get a upper triangle matrix $\begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2.5 \end{pmatrix}$ and recall it by U (the same U appear in the name LU.

Remark. The two systems are equivalent, why? since the matrices we multiplied in are squares matrices and invertable.

So we obtain

$$Ux = E_3E_2E_1x = E_3E_2E_1b$$

$$\Rightarrow b = (E_3E_2E_1)^{-1}Ux = E_1^{-1}E_2^{-1}E_3^{-1}Ux = L_1L_2L_3Ux$$

*Observation. Note that $L = L_1L_2L_3$ we get without any calculations, moreover, its a **basic lower triangle matrices**.*

$$L_1L_2L_3 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -0.5 & 1 \end{pmatrix}$$

$$b = (E_3E_2E_1)^{-1}Ux = E_1^{-1}E_2^{-1}E_3^{-1}Ux = L_1L_2L_3Ux = LUx$$

Corollary. *We got $A = LU$ for regular matrices i.e., the gaussian elimination procedure lead us to the conclusion $b = Ax = LUx$ where in our example above,* $A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -0.5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2.5 \end{pmatrix}$.

Remark. The decomposition above is not dependent with b and all of it stem from A.

Unfortunately we may state the following Theorem which is not true for every matric.

Theorem. *(1.1). Every square matrix could be written in form $A = LU$ where,*

- *L is **basic lower triangle matrices.***
- *U upper triangle matrix*

Theorem (1.1) is not true! not for each matix the procedure expected to work, hence, we will give a definition.

Definition. A matrix A will called "regular" if its square matrix and could be written as $A = LU$ where

- L is **basic lower triangle matrices.**
- U upper triangle matrix **with diagonal elements** $\neq 0$.

Property. The decomposition is unique.

Remarks.

- If that the case, for now our treatment is related to regular matrix and for matrices in which the procedure "Guass elimination" successful.
- Why the procedure could terminate for non regular matrices?

Application in LU for solving system of equation. The main use of LU decomposition is for solving the system since,

$$b = Ax = LUx = L(Ux) = Ly$$

As first step, we need to solve the following system using **Front subsitution** - its a sequential procedure in which finding y_1 give y_2 and so forth till y_n, becasue we may observe that,

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & 0 & \cdots & 0 \\ l_{41} & l_{42} & l_{43} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n4} & l_{n4} & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_n \end{pmatrix}$$

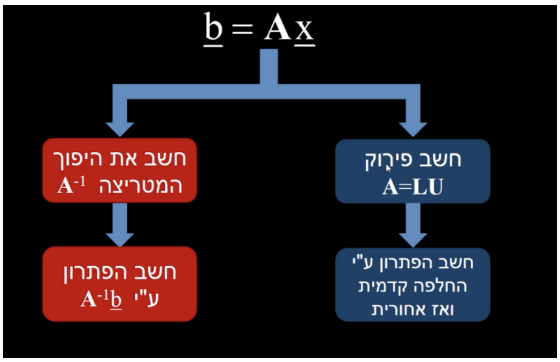
We can see that $y_1 = b_1$ then $y_2 + l_{21}y_1 = b_2$ and we know y_1 and we can continue, this happend becuase the structure of the matrix which make the system $Ly = b$ easy to solve.

Question. Is there a case in which the system will not have a solution? Hell no - determinat of the matrix is clearly not 0. Now, we don't stop because we are interested in finding x, and we know that the solution y we found is equal to Ux where U is upper triangle matrix with diagonal non zeros (otherwise the determinant is 0 and there is no unique solution), hence, we obtain,

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & u_{24} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & u_{34} & \cdots & u_{3n} \\ 0 & 0 & 0 & u_{44} & \cdots & u_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{pmatrix}$$

Now, this is **Back Subsitution** we again can see that $u_{nn}x_n = y_n$ where y_n is known, its called back becasue we start from bottom to up then we cam find x_{n-1} and so forth till we reach x_1.

What is the cost? Which approach better? Inversing the matrix then finding $A^{-1}b$ or LU decomposotion?



Cost of LU.

*Simple observation. Given a a matrix nxn number of operations requred for multiplying it in vector nx*1 *is* n^2 *operations.* How much operation required for back/front subsitution?

Solution. Note that for the first raw for example in front subsitution cost 1 operation is the next raw is 2 and forth so in total $1+2+3+...+n = \frac{1}{2}n^2$ so back and front together is $2 \cdot \frac{1}{2}n^2 = n^2$.

Exercise. Show that number of operations to execute LU decomposition is $\frac{n^3}{3}$.

Exercise. Given a matrix A how much operation needed to find A^{-1}?

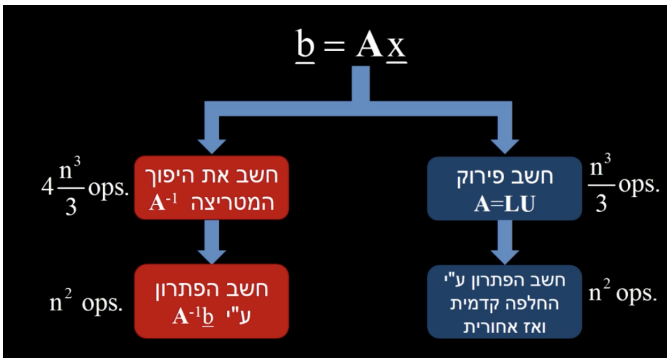
Solution. First, we will see algorithm which find A^{-1} note that this is equivalent to find $AX = I$ where $X = A^{-1}$, so we get a sytem of

$$A\begin{pmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \\ \vdots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, A\begin{pmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \\ \vdots \\ x_{n2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, ..., A\begin{pmatrix} x_{1n} \\ x_{2n} \\ x_{3n} \\ x_{4n} \\ \vdots \\ x_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

(1) Take A and aply on it LU decomposition - $\frac{n^3}{3}$
(2) For each of the n system above apply front and back subsitution so for each yo get a one column in the inverse matrix X - $n \cdot n^2 = n^3$

In total, $n^3 + \frac{n^3}{3} = \frac{4}{3}n^3$. (Remand to prove that this algorithm is optimal).

Comparsion.



Keep in mind that those approaches are way different, the right approach to find a solution is way better since its work in 4 times less than the right approach. In terminology of numerical stability the right approach also wins. Running big simulation may demand one day in the left approach where in the right approach 6 hours which could save a lot.

Remark. All what we discussed is valid for regular matrix, but what about other matrices which are not regular. Now, we are going to treat those type of matrices using permutation.

Non regular matrices. Given for example, the following matrix:

$$A = \begin{pmatrix} 0 & 3 & 1 \\ 2 & 6 & 1 \\ 1 & 0 & 4 \end{pmatrix}$$

Note that the reqiurment for regular matrix are

- L is **basic lower triangle matrices.**
- U upper triangle matrix **with diagonal elements** $\neq 0$.

One of that requirment is satisfied, and the other not, the main thing here is that we can change the order of rows using matrix P,

$$PA = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & 3 & 1 \\ 2 & 6 & 1 \\ 1 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 6 & 1 \\ 0 & 3 & 1 \\ 1 & 0 & 4 \end{pmatrix}$$

So now again as we saw in the case of regular matrix we see that,

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.5 & 0 & 1 \end{pmatrix}$$

hence,

$$E_1 PA = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.5 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 6 & 1 \\ 0 & 3 & 1 \\ 1 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 6 & 1 \\ 0 & 3 & 1 \\ 0 & -3 & 3.5 \end{pmatrix}$$

$$E_2 E_1 PA = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 6 & 1 \\ 0 & 3 & 1 \\ 0 & -3 & 3.5 \end{pmatrix} = \begin{pmatrix} 2 & 6 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 4.5 \end{pmatrix} = U$$

So we got

$$PA = (E_2 E_1)^{-1} U = LU$$

Where,

$$PA = LU = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.5 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 6 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 4.5 \end{pmatrix}$$

Theorem. *Every matrix square A can be written in from $PA = LU$ where,*

- *P is permutation matrix which change raws.*
- *L matrix base lower tringular*
- *U some upper tringular matrix*

Given the system $Ax = b$ and the decomposition $PA = LU$, how to find x?

Solution. We can see that

$$Ax = b \Rightarrow PAx = Pb \Rightarrow LUx = Pb$$

- First, we will do front subsitution for $Ly = Pb$
- Given the solution we found, we will construct a solution x from the system $Ux = y$ using back subsitution.

Example. applying LU decomposition on

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

Solution. First step is changing raw $1, 2$ so the 0 will be in next raw (can't use him as pivot), then we get

$$P_1 A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 2 & 3 & 4 \\ 2 & 1 & 1 & 1 \\ 4 & 4 & 5 & 6 \\ 2 & 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 4 & 4 & 5 & 6 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$

Now we will apply the gaussian operations,

$$E_2E_1P_1A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Note that now the pivot is raw 2 in which we will use to get 0 in raw $3, 4$ however, we will not get the tringular structure because raw 3 will be 0 so we won't get a tringular matrix, this means that we should change raw 3 and 4 in order to make the procedure work, wo we will back to the beginning, and we will apply two permutation on the matrix A.

$$P_2P_1A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 2 & 3 & 4 \\ 2 & 1 & 1 & 1 \\ 4 & 4 & 5 & 6 \\ 2 & 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 2 & 2 & 2 & 2 \\ 4 & 4 & 5 & 6 \end{pmatrix}$$

Again we will apply the gauusian operations,

$$E_2E_1P_2P_1A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \mathbf{-1} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \mathbf{-2} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 2 & 2 & 2 & 2 \\ 4 & 4 & 5 & 6 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \end{pmatrix}$$

$$E_4E_3E_2E_1P_2P_1A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \mathbf{-1} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \mathbf{-0.5} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 0 & 0 & -0.5 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

We can see that marked number in matrices E_1, E_2, E_3, E_4 and those are the operation we do on the matrix, so we can find the inverse easily by changing sign i.e., canceling the opertaion and get L matrix,

$$PA = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0.5 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \\ 0 & 0 & -0.5 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix} = LU$$

Calculate determinant using LU decomposition. Assuming we have in our hands $PA = LU$. Applying the determinant on both sides,

$$det(P)det(A) = det(U)det(L)$$

So in the LU method the determinant is known because,

$$det(P) = \pm 1, det(L) = 1, det(U) = \prod_{k=1}^{n} u_{kk}$$

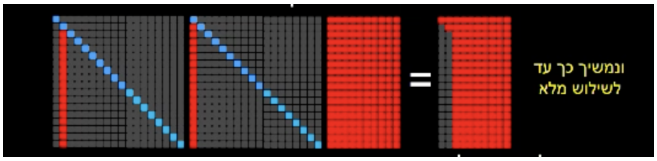
$$\Rightarrow det(A) = \pm u_{11} \cdot u_{22} \cdots u_{nn}$$

So the complexity is $\frac{n^3}{3}$ which is pretty low compared to $n! \sim n^n$ ($det(A) = \sum_{\sigma \in S_n} sign(\sigma) a_{1\sigma(1)} \cdots a_{n\sigma(n)}$.) (which is all possible permutations - brutal force approach).

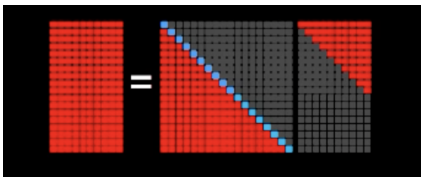
Theorem. *Given a square matrix with decomposition $PA = LU$ if all the diagnal elements of U are $\neq 0$ then A is not singular matrix.*

Generalization to a non quadratic forms.

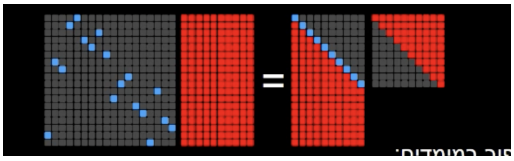
- Each matrix could be written as LU (with permuatation) though its not square matrix.
- Assuming that we have matrix with no need of permutation, what we can do? first we use the element a_{11} as pivot till we get 0 at tfirsts cooadrdination on all raw which are not raw 1 then we continue with raw 2 and now the pivot is a_{22} as described in the figure below,



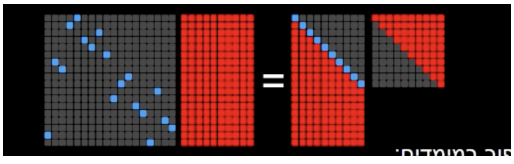
Inverting the down tringular base matrix operation yields,



For example a matrix in size nxm where $n > m$ could be written as LU and will look as follows,



And in case $m > n$ we will get a LU as follows,



Finding perfect pivot. Remember that we use permutation when pivot is 0 and many othercases which will simplify the procedure of LU, for getting intiuition consider the following example,

$$\begin{pmatrix} 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Lets try to solve it using gaussain elimination ($R_2 \to R_2 - 10^5 R_1$)where the pivot element in a_{11} we obtain,

$$\begin{pmatrix} 1 & 0 \\ -10^5 & 1 \end{pmatrix} \begin{pmatrix} 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -10^5 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

So we get that,

$$\begin{pmatrix} 10^{-5} & 1 \\ 0 & 1 - 10^5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ -10^5 \end{pmatrix}$$

Therefore,

$$y = \frac{-10^5}{1 - 10^5} \cong 1$$

The solution for the system is,

$$x = \frac{1}{10^{-5}} \left(1 + \frac{10^5}{1 - 10^5} \right) = \frac{10^5}{1 - 10^5} \cong -1$$

Assuming that want to do that on computer, the problem is that for computer he round down the number, i.e., will consider $1 - 10^5$ as -10^5 then,

$$\begin{pmatrix} 1 & 0 \\ -10^5 & 1 \end{pmatrix} \begin{pmatrix} 10^5 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -10^5 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

So we get that,

$$\begin{pmatrix} 10^{-5} & 1 \\ 0 & 1 - 10^5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ -10^5 \end{pmatrix}$$

But now

$$y = \frac{-10^5}{-10^5} = 1$$

and

$$x = \frac{1}{10^{-5}} \left(1 + \frac{10^5}{-10^5} \right) = 0$$

but $x \cong 1$ and now we got 0 which is big deviation this stem becuase me mutliplied the first raw in a number which is bigger than 1, so we must always that the pivot element is the biggest elemnt in the first cooadination of all raws (which is 1 in this case) so how can we solve this problem? the answer is permutation, first, we will change first raw and second raw i.e. we make the biggest pivot element possible in this operations, so we get

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 1 & 1 \\ 10^{-5} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

now the pivot element is 1 (still rounding up to 4 digits) by applying $R_2 \to R_2 - 10^{-5}R_1$

$$\begin{pmatrix} 1 & 0 \\ -10^{-5} & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 10^{-5} & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -10^{-5} & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

So we obtain,

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Where we considered $1 \cdot -10^{-5} + 1 \cong 1$ (because we round down\up), so the result we get now is more accurate numerically $y = 1, x = -1$. (this method in which we take the pivot always the biggest in its column is called partial pivotting).

Corollary. *It will be convenient to choose the pivot element be the higest element values in its column.*

Full pivotting. In this method, we apply permutation on raws and permutation on column - where the meaning is changing the variables order. Algebric it could be written as follows, PAQ where Q is permutation on columns and P on raws so we get $PAQ = LU$

LDV decomposition.

- We saw that each matrix A could be written in from $PA = LU$
- When we are talking about square matrices, if all the diagonal elements of U are $\neq 0$, then we know that the matrix is not singular (invertible).
- From the two fact mentioned above we stem the following decomposition,

  Theorem. *Each square matrix A which is not singular could be written in form $PA = LDV$ where:*
  - *P is permutation that change raws order*
  - *L is a lower basic trungular matrix*
  - *V is a upper basic trungular matrix*
  - *D diagonal matrix in which all the digonal elements are $\neq 0$*

Example. We saw earlier the following example,

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -0.5 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2.5 \end{pmatrix}$$

LDV decompistion will obtain the follows,

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -0.5 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2.5 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -0.5 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2.5 \end{pmatrix}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & -0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

How we obtained the result? First we take U diagonal and put it into the matrix D the same diagonal and all other elements will be 0, and the matrix V will be in which each element will be the divide result in U element with the diagonal element in the specific raw for example

$$v_{11} = \frac{u_{11}}{u_{11}} = 1, v_{12} = \frac{u_{12}}{u_{11}} = 2, v_{13} = \frac{u_{13}}{u_{11}} = 1$$

$$v_{22} = \frac{u_{21}}{u_{22}} = 0, v_{22} = \frac{u_{22}}{u_{22}} = 1, v_{23} = \frac{u_{23}}{u_{22}} = -0.5$$

$$v_{33} = \frac{u_{33}}{u_{33}} = 1, v_{32} = \frac{u_{32}}{u_{33}} = 0, v_{31} = \frac{u_{31}}{u_{33}} = 0$$

So

$$V = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & -0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

Remark. We can see why we are avoiding the case in which a diagonal element is 0 (we can't divide by 0).

Symmetric matrices and their LDV decomposition.

Definition. A matrix symmetric is a square matrix which imply $A = A^T$, and matrix is anti symmetric if $A = -A^T$.

Claim. Every square matrix A could be written as a sum of symmertic matrix and anit symmetric matrix,

$$A = \frac{A + A^T}{2} + \frac{A - A^T}{2}$$

For example

$$\begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix} = * + **$$

$$* = \underbrace{\frac{\begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix}^T}{2}}_{\frac{A+A^T}{2}} = \frac{\begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 2 & 1 & 3 \\ 4 & 5 & 5 \\ 4 & 3 & 6 \end{pmatrix}}{2} = \begin{pmatrix} 2 & 2.5 & 3.5 \\ 2.5 & 5 & 4 \\ 3.5 & 4 & 6 \end{pmatrix}$$

$$** = \underbrace{\frac{\begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix} - \begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix}^T}{2}}_{\frac{A+A^T}{2}} = \frac{\begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix} - \begin{pmatrix} 2 & 1 & 3 \\ 4 & 5 & 5 \\ 4 & 3 & 6 \end{pmatrix}}{2} = \begin{pmatrix} 0 & 1.5 & 0.5 \\ -1.5 & 0 & -1 \\ -0.5 & 1 & 0 \end{pmatrix}$$

So notice that $*$ is symmetric and $**$ is anti-symmetric also,

$$* + ** = \begin{pmatrix} 2 & 2.5 & 3.5 \\ 2.5 & 5 & 4 \\ 3.5 & 4 & 6 \end{pmatrix} + \begin{pmatrix} 0 & 1.5 & 0.5 \\ -1.5 & 0 & -1 \\ -0.5 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 4 \\ 1 & 5 & 3 \\ 3 & 5 & 6 \end{pmatrix}$$

Observations.

- We want to define in convient way a LDV decomposition for symmetric matrices
- A possible approach is that given a symmertric matrix non singular, exists for it decomposition LDV in form $PA = LDV$

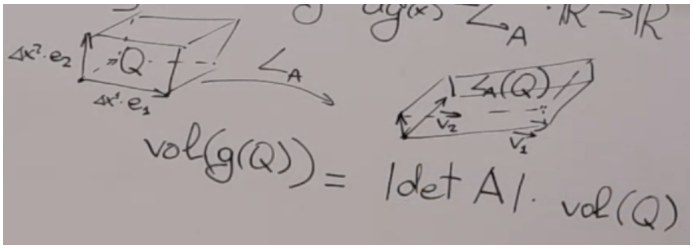
- its obvious that the claim above is true, however in the propoPDed decomposition we lose the symmetric property at all. So is there a way in which we can do similar decomposition with preserving the symmmetric property? Note that A symmetric and PA is not neccessiraly symmetric
- The approach: applying similar permutation on raws and on colmuns will preserve symmetry because,
$$(PAP^T) = (P^T)^T A^T P^T \underset{\text{A is symmetric - } A = A^T}{=} PAP^T$$
- Unfortunately, this leads us to state a wrong theorem (1.2)

  Theorem. *(1.2). Every symmetric matrix which is not singular could be written as form $PAP^T = LDL^T$ where,*

  - P is a permutation matrix that changes raws order
  - L is lower basic tringular matrix
  - D diagonal matrix in which all diagonal elements are $\neq 0$
- The theoem above is wrong, in fact we can provide counter example,
$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = P^T AP = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
  Note that what happended is that P^T ruin the raws change in which we want to make the pivot element 1 not 0, so we are back to the same matrix. Hence, not every symmertic matrix non signular gives the LDL^T decomposition.
- When we get a square matrix non singular, our interest is in decomposition of from $PAP^T = LDL^T$ where preserve the symmetry of the matrix, but we should emphasize that such a decomposition not neccessarily exists.
- Example for LDL^T is provided in the Tutorial chapter, with the algorithm obtain it.

Remark. This decomposition is lower than LU in factor of 2 becuase of the symmetric prtoberty, and its complexity is $\frac{n^3}{6}$operations. However, in order to obtain it we need to find different algorithm which will also be provided in the Tutorial chapter.

Reminder (inituition for Gram matrix). If we have $g = dg(x) = L_A : \mathbb{R}^n \to \mathbb{R}^n$ and $Q \subset \mathbb{R}^n$ a box. If we apply L_A on Q then we get from box a other box as we could in the following photo,



We want to define a volume of $m < n$ dimentional of a domain Ω in $\mathbb{R}^n$. E.g length of a curve, i.e we want to define a surface in dimention lower than n and for example for a curve there is a volume in $\mathbb{R}$, in the same way we want to define a length of a 1-dimentional curve in $\mathbb{R}^n$ or for example a area of surface (sphere -

2 dimentional) in $\mathbb{R}^3$. What is the area of the sphere? the are of sphere in $\mathbb{R}^2$ is $4\pi r^2$ and the volume in $\mathbb{R}^3$ $\frac{4}{3}\pi r^3$.

Remark. A volume of spehere in $\mathbb{R}^2$ is 0 but area is different than volume and it's not 0.

A m dim volume of a m dimentional box in $\mathbb{R}^n$, $m \leq n$.

Example. $m = 1, n = 2$ what is a parallelepiped P in $\mathbb{R}$? a line, and a 1 dim volume is length of a interval. So how we find it? we can think about it as a vector and the length of it is $||\vec{v} = (v_1, v_2)|| = \sqrt{v_1^2 + v_2^2} = vol_{1-dim}P$. if $m = 2, n = 3$ then the volume is $||v_1 x v_2|| = ||v_3|| = vol_{2-dim}P = ||v_1|| \cdot ||v_2|| \cdot sin(\alpha)$ which is the area. How can we generalize it?

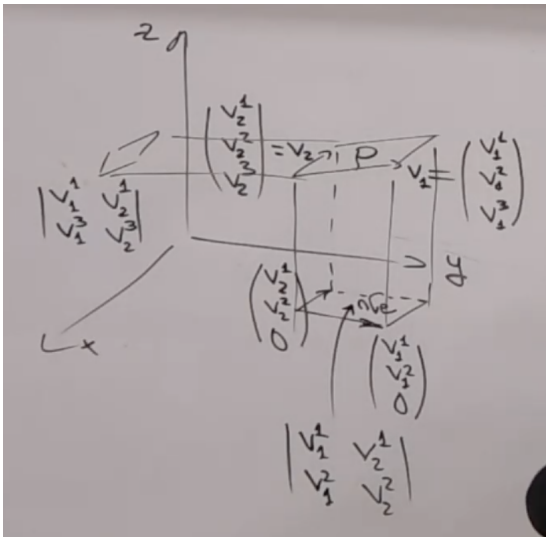
$$(vol_{2-dim}P)^2 = ||v_1 x v_2||^2 = ||v_1||^2 \cdot ||v_2||^2 \cdot sin^2\alpha =$$

$$= ||v_1||^2 \cdot ||v_2||^2 \cdot (1 - cos^2\alpha) =$$

$$= ||v_1||^2 \cdot ||v_2||^2 - < v_1, v_2 >^2$$

$$=< v_1, v_1 >< v_2, v_2 > - < v_1, v_2 >^2$$

$$= det \begin{pmatrix} < v_1, v_1 > & < v_1, v_2 > \\ < v_1, v_2 > & < v_2, v_2 > \end{pmatrix} = \text{Gram-matrix}$$



Remember that the three minors we get from the cross product. This remind us in pythagoras theorem. We can see that $||v + w|| = ||v|| + ||w|| \iff v \perp w$. Now, for three vectors, we can see that we have as above assuming we are in $\mathbb{R}^3$ also we can do that for $\mathbb{R}^n$ but lets focus on $\mathbb{R}^3$ we can see for example that we must have each time cooardination 0. So $||v + w + z||^2 = M_1^2 + M_2^2 + M_3^2$ where M_1, M_2, M_3 are the minors.

$$||v + w + z||^2 = ||v + w + z|| \cdot ||v + w + z||$$

$=_{pythagoras-in-n=2} \sqrt{||v||^2+||w||^2+||z||^2}\cdot\sqrt{||v||^2+||w||^2+||z||^2} = (||v||^2+||w||^2+||z||^2)$

We can see above that area between each two vectors on different planes is a det of the vectoss and we can omit the 0's in the corresponding cooardination this will give us also the vector v_3 which is normal to them each coordination of v_3 vectos is a just a determinant of matrix or more specific is the minor so M_1, M_2, M_3 are the cooardiations of the vectors, and since the volume is the norm of corss product between v_1, v_2 now we can think about it as

$$vol^2(P) = ||v_3||^2 = \sqrt{(M_1^2+M_2^2+M_3^2)} = M_1^2+M_2^2+M_3^2$$

$$= det\begin{pmatrix} < v_1, v_1 > & < v_1, v_2 > \\ < v_1, v_2 > & < v_2, v_2 > \end{pmatrix} = \text{Gram-matrix}$$

since each of the minors are the cooradintion of v_3 .

Remark. This is true only when $m = n-1$.

Remark. In order to understand the whole picture we can give example in life, in 3 dimention we want to give a meaning of object in their dimention for example, in out world a line got a length which is vol_{1-dim} but a line can't have a vol_{2-dim} since it will get volume 0 (already showed), however we can give it a area in higher dimentions which we can observe i.e when $n > m$. So in the example above we took a private case in which we are looking at box, a box in 1 dim which is a line has a vol_{1-dim} in $\mathbb{R}^2$ but not a vol_{2-dim} in $\mathbb{R}^2$. So now we are going to do that using matrix gra m which could give a m dim object a vol_{m-dim} in $\mathbb{R}^n$.

Theorem. *Let $m \le n$ and let $P(v_1,..,v_m)$ parallelepiped which is span of the vectors $v_1,..,v_m \in \mathbb{R}^n$ then $Vol_m(P) = \sqrt{det(G)}$,*

$$G = \begin{pmatrix} < v_1, v_1 > & \cdots & < v_1, v_m > \\ \vdots & \ddots & \vdots \\ < v_m, v_1 > & \cdots & < v_m, v_m > \end{pmatrix}$$

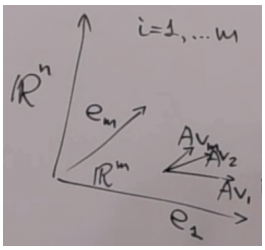
Proof. if $m = n$ then

$$(Vol_{m=n}(P))^2 = \left(det\begin{pmatrix} \vdots & \vdots & \vdots \\ v_1 & \cdots & v_n \\ \vdots & \vdots & \vdots \end{pmatrix}\right)^2 = det(V^TV)$$

$$det\left(\begin{pmatrix} \cdots & v_1^t & \cdots \\ \ldots & \vdots & \ldots \\ \cdots & v_n & \cdots \end{pmatrix}\begin{pmatrix} \vdots & \vdots & \vdots \\ v_1 & \cdots & v_n \\ \vdots & \vdots & \vdots \end{pmatrix}\right)$$

$$= det\begin{pmatrix} < v_1, v_1 > & \cdots & < v_1, v_n > \\ \vdots & \ddots & \vdots \\ < v_n, v_1 > & \cdots & < v_n, v_n > \end{pmatrix}$$

if $m < n$ then we can rotate all the vectors by otthogonal matrix $A \in O(n)$ we rotate all the v_i to a case in which all of them will be in $\mathbb{R}^n$



So now $Av_i \in \mathbb{R}^n$ and

$$Av_i = \begin{pmatrix} (*)_{mx1} \\ (0)_{(n-m)x1} \end{pmatrix}$$

. Therefore,

$$Vol\left(P(v_1,...,v_m)\right)^2 = Vol\left(P(Av_1,...,Av_m)\right)^2 = det(G_{mxm})$$

Stemming,

$$g_{ij} =< Av_i, Av_j >=< v_i, A^tAv_j >=< v_i, v_j >$$

□

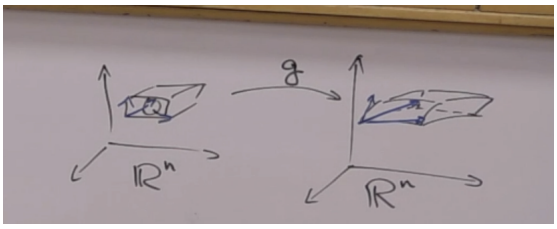
Example. $m = 2, n = 3$ in this case we mentioned that a norm cross product of two vectors in $\mathbb{R}^m$ yields the $volume_m$ of the parellelpiped in $\mathbb{R}^n$ and also is the sum of all minors of the cross product matrix (the vector $||v_1xv_2||$ cooardination, and is also the same result we get from the gram matrix. Let see this, $v_1 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}, v_2 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$ then $G = \begin{pmatrix} 11 & 5 \\ 5 & 5 \end{pmatrix} \Rightarrow det(G) = 30$. In the other hand,

$$v_1xv_2 = (1 \cdot 0 - 1 \cdot 2)i + (1 \cdot 1 - 3 \cdot 0)j + (3 \cdot 2 - 1 \cdot 1)k$$

$$= -2i + 1j + 5k$$

$$||v_1xv_2||^2 = \sqrt{(-2)^2 + 1^2 + 5^2} = \sqrt{30}$$

Now assuming that we don't have a parellelpiped instead we have a map g which is not linear, we can do the same as the substitutopn forumla intriduced earler but now instead of using the jacobian matrix we use the Gram matrix.



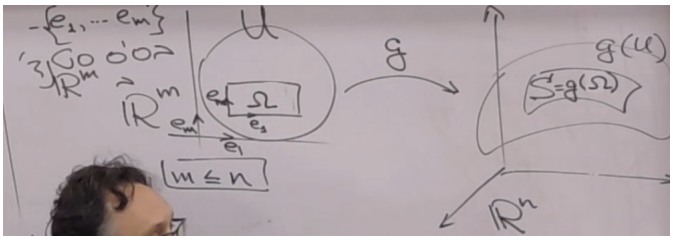
$$Vol_n g(Q) = \int_Q 1 \cdot |J_g| dV$$

$$J_g = det \begin{pmatrix} \underbrace{\begin{matrix} \frac{\partial g_1}{\partial x_1} \\ \vdots \\ \frac{\partial g_n}{\partial x_1} \end{matrix}}_{v_1} & \cdots & \underbrace{\begin{matrix} \frac{\partial g_1}{\partial x_1} \\ \vdots \\ \frac{\partial g_n}{\partial x_1} \end{matrix}}_{v_n} \end{pmatrix}$$

$$det \begin{pmatrix} \vdots & \vdots & \vdots \\ v_1 & \cdots & v_n \\ \vdots & \vdots & \vdots \end{pmatrix} = Vol_n P(v_1, .., v_n) = \sqrt{detG}$$

$$Vol_n g(Q) = \int\limits_{g(Q)} 1 dV = \int\limits_{Q} \sqrt{detG} dV$$

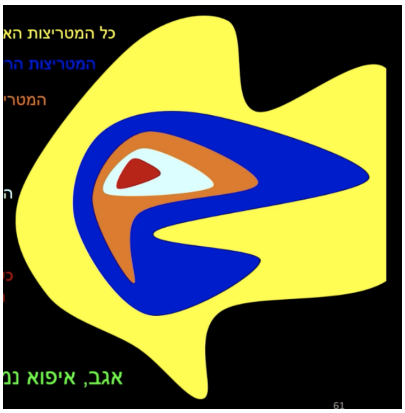
Now we have a problem why can we ensure that on different parameterazation on Ω we get the same result.



We are going to define a volume of surface or length of a curve in a heigher dimentions spaces. We have the same thing as before but now we have $m \le n$. We need to define it proberly, let $g \in C^1(U, \mathbb{R}^n)$ injective map wnd $rand(dg(x)) = m$. A Ω (Closed and bounded) with volume, $S = g(\Omega)$ is called a parametric surface.

Definition. Define $Vol_m(S) = \int\limits_{\Omega} \sqrt{detG}_{\text{mxm matrix}}\, dV$ when $g_{ij}(x) =< De_i, De_j >$ where $D = \begin{pmatrix} \underbrace{\begin{matrix} \frac{\partial g_1}{\partial x_1} \\ \vdots \\ \frac{\partial g_n}{\partial x_1} \end{matrix}}_{v_1} & \cdots & \underbrace{\begin{matrix} \frac{\partial g_1}{\partial x_1} \\ \vdots \\ \frac{\partial g_n}{\partial x_1} \end{matrix}}_{v_n} \end{pmatrix} |_x$ is a represnetation matrix of $dg(x)$.

Symmetric matrices and LDV decomposition.



- Every possible matrix has a $PA = LU$ decomposition - Yellow
- A square matrices in which U is a upper tringular matrix - Blue
- Square matrices which are not singular and in which U diagonal elements are $\neq 0$ - Orange
- Symmetric matrices which are not sinuglar - in which the decompostion preserve symmetry $PAP^T = LDL^T$ - White
- Special family of positive definite matraices which are symmetric and $PAP^T = LDL^T$ decomposition is implied.

Definition. Given a square matrix and symmetric K is positive definite if it satisfy $\forall x \neq 0, x^T K x > 0$.

- We will denote this property by $K \succ 0$. The definition for matrices is note more than the generalzation of scalar number - $a > 0 \iff ax^{2>0}$ for all $x \neq 0$.

Example. (1.1) Given the following matrices $K = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ is it definite positive? i.e., $\forall x = [x_1, x_2, x_3] \neq 0$ satsified $xKx^T \underset{?}{>} 0$. Let check, note that we have,

$$[x_1 + x_2 + x_3, x_1 + x_2 + x_3, x_1 + x_2 + x_3] [x_1, x_2, x_3]^T$$

$$= x_1^2 + x_2 x_1 + x_3 x_1 + x_2^2 + x_1 x_2 + x_3 x_2 + x_3^2 + x_1 x_3 + x_2 x_3$$

$$= \left(x_1^2 + x_2^2 + x_3^2\right) + 2x_1 x_2 + 2x_1 x_3 + 2x_2 x_3$$

$$= (x_1 + x_2 + x_3)^2$$

For example, if we take $x_1 = -100, x_2 = 100, x_3 = 0$ we get 0 , so its not definite positive for every $x \neq 0$. (However, its PDD which we will define)

Theorem. *Given a square matrix, symmetric, and positive definite (PD), then K is not singular.*

Proof. Assume that K is singular i.e., 0 is eigen value and exists v eigenvector in which $(K - 0I)v = 0$ ($v \in Ker(K)$) so implies that $vKv = v \cdot 0 = 0$ but this is contradiction to that fact that K is PD. □

Definition. A squaare matrix and symmetrix is positive semi defenite (PDD) if it satisfy $\forall x \neq 0, x^T K x \geq 0$.

Example. The K in example (1.1) is PDD.

Exercise. For which a where $K = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}$ will be PD or PDD?

Solution. We will open the term and get that

$$(x_1, x_2) \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix} (x_1, x_2)^T = (x_1, x_2) \begin{pmatrix} x_1 + a x_2 \\ a x_1 + x_2 \end{pmatrix}$$

$$= x_1^2 + 2a x_1 x_2 + x_2^2 = (x_1 + a x_2)^2 + (1 - a^2) x_2^2$$

If $a = \pm 1$ then K is PDD, if $-1 < a < 1$ ($|a| < 1$)then K is PD. If $|a| > 1$ then the matrix is not PD or PDD.

Example. For which condition we have that a diagonal matrix $K = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_n \end{pmatrix}$ is PD? for example consider $n = 3$, so for all $x = [x_1, x_2, x_3]$, $x K x^T \underset{?}{>} 0$. Note that $x^T K x = d_1 x_1^2 + d_2 x_2^2 + d_3 x_3^2$ then if d_1, d_2, d_3 are positive then we get that K is PD (i.e., in termenilogy of eigenvalue d_1, d_2, d_3 we require them all to be positive). If for example, $d_1 = 0$ then we can take any vector $x = [1, 0, 0] \neq 0$ where $x K x^T = 0$ so PDD. So in general,

$$x^T D x = x \begin{pmatrix} d_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_n \end{pmatrix} x^T = \sum_{k=1}^{n} d_{kk} x_k^2$$

Corollary. *If all the diagnoal element are positive, then K is PD. If the are non-negtaive, then K is PDS.*

Question. Is symmetry needed inorder to get the PD property (i.e., $x^T D x > 0$) ?
No. For counter example consider the non symmetric matrix $K = \begin{pmatrix} 1 & 0 \\ a & 2 \end{pmatrix}$ by definition,

$$x^T K x = [x, y] \begin{pmatrix} 1 & 0 \\ a & 2 \end{pmatrix} [x, y]^T$$

.

$$= [x, y] \begin{bmatrix} x \\ ax + 2y \end{bmatrix} = x^2 + axy + 2y^2$$

$$= (x + \frac{a}{2} y)^2 + (2 - \frac{a^2}{4}) y^2$$

Note that for all $|a| < \sqrt{8} \Rightarrow K \succ 0$. But K is not symmetric.

Remark. Remeber the in the definition of PD we also required the K is symmetric

Gram matrices. Is a matrix is form $K = A^T A$ is is PDD?

Observation. Note that K is symmetric becasue $K^T = \left(A^T A\right)^T = A^T A^{T^T} = A^T A$.

Theorem. *A Gram matrix is neccessarily PDD. If it columns of A is independent, then the matrix is PD.*

Proof. We will start with observation with out the identity of A, the matrix K is PDD

$$K = A^T A \Rightarrow x^T A^T A x = ||Ax||_2^2 \geq 0$$

If the columns of A is linearly independent, there there is not vector x which gives $Ax = 0$ except $x = 0$. Hence, the term will not vanish for every vector x which is not trivial, hence the matrix is PD. □

Theorem. *If C is a PD matrix and A has linearly independent columns then $A^T C A$ is PD.*

Remark. Its a gerneralization of the gram matrix where $C = I$.

Proof. The core of the proof is that $Ax = z$ is not 0 when $x \neq 0$ because A has linearly independent columns so for $K = A^T C A$ we obtain

$$x^T A^T C \underbrace{Ax}_{z} = z^T C z >_{C-PS} 0$$

□

Cholesky decomposition.

- We saw that for symmetric matrices which are not singular, there is interest in finding decomposition $PAP^T = LDL^T$(which is not always guranteed).
- Given a matrix K which is PD, obvious that its symmetric and nonsingularthem such a LDL^T decomposition exists.

Theorem. *Given a symmertic matrix K also PD then its neccessarily "regular" so exists the following decomposition with no need of permutation $K = LDL^T$ then all the diagonal elements of D are > 0.*

Proof. Why the digonal elements are > 0, we will consider more general decomposition with permutation,

$$PKP^T = LDL^T \Rightarrow K = P^T LDL^T P$$

$$\Rightarrow K = P^T LDL^T P \Rightarrow \forall x \neq 0, x^T P^T LDL^T Px > 0$$

If we denote $L^T Px = z$ then we get, ($\forall x \neq 0$ we have $z \neq 0$ since L, P is L.I and so injective)

$$\forall z \neq 0, z^T Dz > 0 \Rightarrow D \succ 0$$

□

Observation. Given a matrix $K = LDL^T$ where D is diagonal matrix where L is lower tringular matrix then

$$K = LDL^T = \underbrace{LD^{\frac{1}{2}}}_{M}\underbrace{D^{\frac{1}{2}}L^T}_{M^T} = MM^T$$

so we get $M = LD^{\frac{1}{2}}$ which is the root of K, this is Cholesky decomposition to the matrix K in cost of $\frac{n^3}{6}$ operations.

Remark. Is M the only root for K? no, by looking at family of orthognal matrices which satisfy $Q^TQ = I$ we can do $K = \underbrace{LD^{\frac{1}{2}}Q}_{M}\underbrace{Q^TD^{\frac{1}{2}}L^T}_{M^T}$ and get many other roots but they are not lower/upper tringular.

Application in Matlab. A special command in matlab which could geneare LU composition for matrix A is done by the co mmand LU.

Least Squares.

System of equation and least squares. Assuming that we have m equations in n variables in which we are looking for their solution, $f_1(x) = 0, f_2(x) = 0, ..., f_m(x) = 0$. We can define function whivh is combination of all those functions,

$$p(x) = [f_1(x)^2] + ... + [f_m(x)^2] = ||f(x)||_2^2$$

And find a vector x^* which will obtain the minimum for $p(x)$.

- Its clear that such a solution to the source system, the function value in minimun will be 0. Hence, the two problems are equivalent.
- However, if there is no solution to the system, there could still minimzation for $p(x)$ with a certian value, i.e., its meaning is finding optimal solution with minimum square error to the source problem.

Summary. We move from the representation of the problem:

$$f_1(x) = 0, f_2(x) = 0, ..., f_m(x) = 0$$

to minimization problem i.e.,

$$\min_s p(x) = \min_s \sum_{i=1}^{k} [f_k(x)]^2$$

.

$$f(x) = 0 \Rightarrow \min_s ||f(x)||_2^2$$

System of equations and least squres.

- Assuming we are dealing in collection of m linear equation and n variables in which is shared solution we want to find,

$$\left\{ \begin{array}{c} a_1x - b_1 = 0 \\ a_2x - b_2 = 0 \\ \vdots \\ a_mx - b_m = 0 \end{array} \right\}$$

- In other words we can write it as

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} x - \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = Ax - b = 0$$

- The cost function will be LS problem,

$$p(x) = \sum_{k=1}^{m} (a_k x - b_k)^2 = (Ax-b)^T \underbrace{(Ax-b)}_{r}$$

$$= ||Ax-b||_2^2 = ||r||_2^2$$

- We want so much that our $r = 0$ however, this is not the case in minimization.

Summary. We move from the representation of the problem,

$$\begin{Bmatrix} a_1 x - b_1 = 0 \\ a_2 x - b_2 = 0 \\ \vdots \\ a_m x - b_m = 0 \end{Bmatrix}$$

.

$$\Rightarrow \min_x \sum_{k=1}^{m} (a_k x - b_k)^2$$

So,

$$Ax = b \Rightarrow \min_x ||Ax-b||_2^2$$

- If the system of equation has solution, then the two representation are equivalent.
- If there is no solution, the more new solution is more presious.

Other view to the problem.

- Assuming in our hands there is sequence of $m > n$ vectors,

$$\{v_1, v_2, .., v_n\} \in \mathbb{R}^m$$

- This vectors span a subspace with dim $< n$. Namely, every vector in the subspace could be written as,

$$\sum_{k=1}^{n} x_k v_k = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Recall the relation above in other notation by a matrix A in which its columns are the vector v_k this will be a matrix with m raws and n columns, $\sum_{k=1}^{n} x_k v_k = Ax$

- Assuming we have a arbitrary vector, b, in length m, and we want to find its projection on V i.e., we want to find vector which is the most close to b in the euclidean distance in this subspace i.e.,

$$\min_x \; || \sum_{k=1}^{n} x_k v_k - b||_2^2 = \min_x ||Ax - b||_2^2$$

- Every least squares problems may be considered as a projection problem to thw columns of matrix A, where the projection result is a vector Ax^*

How to solve the LS?.

$$p(x) = ||Ax - b||_2^2 = (Ax - b)^T (Ax - b)$$

$$= \left(x^T A^T - b^T\right)(Ax - b) = x^T A^T Ax - x^T A^T b - b^T Ax + b^T b$$

$$= x^T A^T Ax - 2x^T A^T b + b^T b = *$$

Note that $A^T A$ is symmetric and PS, denote $A^T A = K, A^T b = f$, we obtain,

$$* = x^T A^T Ax - 2x^T A^T b + b^T b$$

$$= x^T Kx - 2x^T f + c$$

- Note that K is Gram matrix hence symmetric and PS.
- The function obtained get input x with dimentions n, andd reutnr a scalar. That is "multi-dimensions parabola" since it contain element x in power of $0, 1, 2$ hence its square problem.

Minimization of square problem.

Theorem. *Given the following problem, $p(x) = x^T Kx - 2x^T f + c$ if K is PD then for $p(x)$ there is unique minimum global point defined by $x^* = K^{-1} f$ and the function value in this point will be $p(x^*) = c - (x^*)^T K(x^*)$.*

Remark. In the solution forumla we require to inverse K, its legel since we assume that the matrix is PD.

Proof. We will use the relation $x^* = K^{-1} f = Kx^* = f$, we will write $p(x)$ as follow,

$$p(x) = x^T Kx - 2x^T f + c = x^T Kx - 2x^T Kx^* + c$$

$$= (x - x^*)^T K(x - x^*) + c - (x^*)^T K(x^*)$$

Observations.

(1) We note that in the last transition we exploited that fact that K is symmetric.
(2) In the last pattern we got, since K is PD its obvious that,

$$(x - x^*)^T K(x - x^*) \geq 0$$

In fact this its brought to minimal value, 0, only for $x = x^*$. Therefore, this is minimal unique global point, and the function value obtained directly. □

- If we back to the source problem,

$$p(x) = ||Ax - b||_2^2 = (Ax - b)^T(Ax - b)$$

The solution will be,

$$x^* = K^{-1}f = (A^TA)^{-1}A^Tb$$

where multiplying in A^TA yields,

$$\left(A^TA\right)x^* = A^Tb \iff A^T(Ax^* - b) = 0$$

Where

$$Ax^* - b = r$$

- and its true only if A columns are linearly independent, so we ensure A^TA invertible (PD).
- This equation is called the normal equation" and the reason behind it is that in the optimal solution obtained since that remainder perpendicular to the columns of A

$$(A^TA)x^* = A^Tb$$
$$\Rightarrow A^T(Ax^* - b) = A^Tr^* = 0$$

- What happen when A is square and invertible?
  - The system of equation $Ax = b$ has unique solution which is given by $x^* = A^{-1}b$, plugging in the solution in the LS $\underset{x}{min}\ ||Ax - b||_2^2$ where its solution is

$$x^* = \left(A^TA\right)^{-1}A^Tb = A^{-1}A^{-T}A^Tb = A^{-1}b$$

    problem will give $p(x^*) = 0$ hence its indeed optimal solution
- The LS problem give a generelaztion of matrix inverse which concide with the regular inverse of the square matrix and invertible. This generelaztion is known as Psuedo inverse

$$\text{Psuedo-Inverse}(A) = A^\dagger = (A^TA)^{-1}A^T$$

so

$$x^* = \left(A^TA\right)^{-1}A^Tb = A^\dagger b$$

- Note that if we have $AB = I$ and A is invertible then $B = A^{-1}$ also $BA = I, AB = I$ however, in case A not invertible, we know that

$$A^\dagger A = (A^TA)^{-1}A^TA = I$$

However,

$$AA^\dagger = A(A^TA)^{-1}A^T$$

and we are stuck! i.e., A has a left inverse but not right inverse, so in one direction it give the right behavious however in the other direction its not.
- How all this related to the decompositions we saw? LU and Cholesky? The answer is simple:
  - We start from LS problem in form

$$p(x) = ||Ax - b||_2^2 = (Ax - b)^T(Ax - b)$$

  - The solution required will be the solution of system of equations,

$$(A^TA)x^* = A^Tb$$

  - This solution will be obtained using LU decomposition, or more accurate the Cholesky decomposition, since we assume that the matrix A^TA is PD.
- Till now we assumed that A^TA is PD. What about otherwise?
- We will state the source theorem with little modification

Theorem. *Given the problem, $p(x) = x^TKx - 2x^Tf + c$. If K is PSD then every vector which satisfy $Kx^* = f$ will be optimal solution to the problem with minimal value, $P(x^*) = c - (x^*)^TK(x^*)$.*

Proof. In fact, the proof rely on the same transition in Theorem before. We will assume that the vector x^* in our hand satisfy $Kx^* = f$ hence,

$$p(x) = x^TKx - 2x^Tf + c = x^TKx - 2x^TKx^* + c$$

$$= (x - x^*)^TK(x - x^*) + c - x^{*T}Kx^*$$

Since, K is PSD then, $(x - x^*)^TK(x - x^*) \geq 0$. Thus term will get 0 when $K(x - x^*) = 0$ i.e., $Kx = Kx^* = f$ hence, every solution will constitute global minima, and the value of function in it will be 0. □

Remark. What are all solutions? If $Kd = 0$ then if x^* is solution, also $x^* + d$ because $K(x^* + d) = Kx^* = f$ for all $d \in Ker(K)$ i.e, $Kd = 0$ so given a minimal solution x^* and $Ker(K) \neq \emptyset$ then we have infinite solution.

Remainder. Back to the interpetation of LS problem as projection problem, given a vector b and we want to find a projection on A collection of vectors which are columns of matrix A, we saw that for finding the projection we need to look at,

$$\min_x || \sum_{k=1}^{n} x_k v_k - b||_2^2 = \min_x ||Ax - b||_2^2$$

- If A columns are Linearly dependent, we understant that there are infinite solution to the LS problem, this is in terminology of vector x.
- However, all the solution will lead s to one projection vector which is Ax^*, why?
- Because exists d in which $Ad = 0$ then if x^* is solution also $x^* + d$ note that we get $A(x^* + d) = Ax^*$. Hence, the projection is unique though we can get from many x but Ax is unique.

Deffrentiating vector matrices. Consider the example to the LS problem,

$$\min_x ||Ax - b||_2^2 = \min_x || \begin{pmatrix} 7 & 10 \\ 5 & -4 \\ 0 & 2 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{pmatrix} 10 \\ -12 \\ 3 \end{pmatrix} ||_2^2$$

The solution we got from formula given was

$$x^* = (A^TA)^{-1}A^Tb = \begin{pmatrix} 74 & 50 \\ 50 & 120 \end{pmatrix}^{-1} \begin{pmatrix} 10 \\ 154 \end{pmatrix} = \begin{pmatrix} 1.0188 \\ 1.7078 \end{pmatrix}$$

Now, let doit with more simple language, given the function

$$p(x_1, x_2) = (7x_1 + 10x_2 - 10)^2 + (5x_1 - 4x_2 + 12)^2 + (2x_2 - 3)^2$$

$$= 74x_1^2 + 120x_2^2 + 100x_1x_2 - 20x_1 - 308x_2 + 253$$

We can obtain minima to the function by deffrentiatin and set to 0 the derivatives,

$$\frac{\partial p(x_1, x_2)}{\partial x_1} = 2 \cdot 74x_1 + 100x_2 - 20 = 0$$

$$\frac{\partial p(x_1, x_2)}{\partial x_2} = 100x_1 + 2 \cdot 120x_2 - 308 = 0$$

So, we got,

$$x^* = \begin{pmatrix} 74 & 50 \\ 50 & 120 \end{pmatrix}^{-1} \begin{pmatrix} 10 \\ 154 \end{pmatrix} = \begin{pmatrix} 1.0188 \\ 1.7078 \end{pmatrix}$$

General Formula. The function we are dealing with is $p(x) = \mathbb{R}^n \to \mathbb{R}^+$ which is given by $p(x) = ||Ax - b||_2^2 = (Ax - b)^T(Ax - b)$.

- What is the gradent vector pf the function,

$$\nabla p(x) = \begin{pmatrix} \frac{\partial p(x)}{\partial x_1} \\ \frac{\partial p(x)}{\partial x_2} \\ \vdots \\ \frac{\partial p(x)}{\partial x_n} \end{pmatrix}$$

  Its a collection of deravitives we want to vanish.
- Our target: Get a formula for the gradeint vector
- Since its looking hard to obtain let start with simple case

Assuming that the function is given is nuilt from 1 equation i.e.,

$$p(x) = (ax - b)^2 = \left(\sum_{k=1}^{n} a_k x_k - b \right)^2$$

in this case the gradient will be,

$$\nabla p(x) = \begin{pmatrix} \frac{\partial p(x)}{\partial x_1} \\ \frac{\partial p(x)}{\partial x_2} \\ \vdots \\ \frac{\partial p(x)}{\partial x_n} \end{pmatrix} = 2 \left(\sum_{k=1}^{n} a_k x_k - b \right) \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

Now, since our problem is one equation in form,

Where a is vector of coeffcient, and now we can add more vector like that and get more intricate $p(x)$ which is built from to vectors of coeffcient a_1, a_2 i.e.,

$$p(x) = (a_1 x - b_1)^2 - (a_2 x - b_2)^2$$

$$\nabla p(x) = 2a_1^T(a_1 x - b_1) + 2a_2^T(a_2 x - b_2)$$

$$= 2 \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} \begin{bmatrix} a_1x - b_1 \\ a_2x - b_2 \end{bmatrix} = 2A^T(Ax - b)$$

This formula could be generialized for matrix of coeffcient with m vector $a_1, ..., a_m$ i.e., mxn matrix and a vector of n variables $x = [x_1, ..., x_n]^T$ where b is vector $mx1$ and since the gradient must be vector of number of variables i.e., $nx1$ we can see that indeed $2A^T(Ax - b)$ is a vector with dimensions $nx1$. Now, to get minima we want $\nabla p(x) = 0$ i.e.,

$$\nabla p(x) = 2A^T(Ax - b)$$

$$\Rightarrow A^TAx = A^Tb$$

Which we already saw as the normal equation, and if A columns are Linearly independent then we get only one solution unique, otherwise, each vector imply it will make the gradient vanish.

Remark. What is the second derivative? Note that first derivative we deffrentiate by each of the variables $x_1, .., x_n$ and now each term we get we can deffrentiate by $x_1, ..., x_n$ so we get a well know matrix which is called the Hessian matrix, and in order to check if the point is minima we need to see that the hessian matrix at this point is PD matrix.

What about $p(x) = x^TKx - 2x^Tf + c$, in this case,

$$p(x) = [x_1, x_2] \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} [x_1, x_2]^T - 2[x_1, x_2]f + c$$

$$= k_{11}x_1^2 + k_{12}x_1x_2 + k_{21}x_1x_2 + k_{22}x_2^2 - f_1x_1 - 2f_2x_2 + c$$

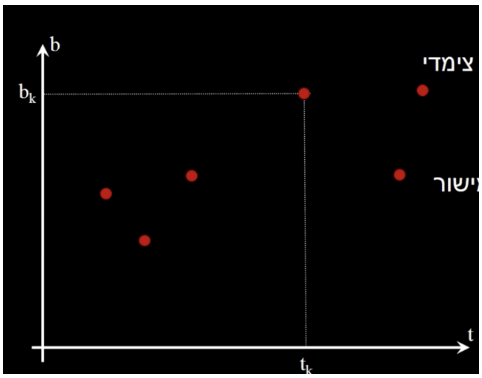
I.e we want to deffrentiate the term respect to $x_1, x_2$and obtain,

$$\nabla p(x) = \begin{pmatrix} 2k_{11}x_1 + k_{12}x_2 + k_{21}x_2 - 2f_1 \\ 2k_{22}x_2 + k_{12}x_1 + k_{21}x_1 - 2f_2 \end{pmatrix}$$

$$= 2 \begin{pmatrix} k_{11} & \frac{k_{12}+k_{21}}{2} \\ \frac{k_{12}+k_{21}}{2} & k_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - 2 \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = 2(Kx - f)$$

So this is the expected result since K is symmertic so we got that minima is in x where $Kx = f$. The formula holds for higher dimentions.

Application to least squares: Matching to curve to a data. Assuming we have acollection of data pairs $\{t_i, b_i\}_{i=1}^m$ which descrive a point in the plane as follows,



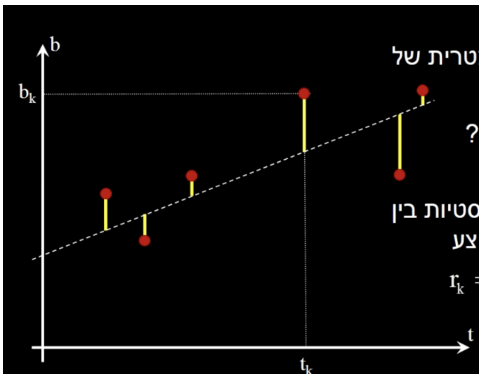
Assume we know that this points must fall on a line, and as a result of deviation in measurm ents they are scattered, we want to find a line $at + \beta$ which will match the data in the best way. Assuming that we found such a line that we claim he is the best for our points, so when a line pass in point t_k it values must be $\alpha t_k + \beta$ and he supposed to be b_k so what we can do? We must get b_k the most close to $\alpha t_k + \beta$ so this is classical least squares problem where the task is,

$$\min_{\alpha,\beta} \sum_{k=1}^{m} (\alpha t_k + \beta - b_k)^2$$

We will define the matrix A and vector b as follows,

$$\begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ 1 & t_3 \\ 1 & \vdots \\ 1 & t_m \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix} = Ax - b$$

What is the Geometric meaning?



Each solution will generate a deviation from the red points to the proposed line where

$$r_k = \alpha t_k + \beta - b_k$$

Sum of the deviation we want to minimize in solution of the LS.

Solution to the problem. we want to find

$$\min_x \left\| \begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ 1 & t_3 \\ 1 & \vdots \\ 1 & t_m \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix} \right\|_2^2 = \min_x \|Ax - b\|_2^2$$

We know that

$$x^* = \left(A^T A\right)^{-1} A^T b = \begin{bmatrix} \beta^* \\ \alpha^* \end{bmatrix}$$

Where,

$$A^T A = \begin{pmatrix} m & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & \sum_{i=1}^m t_i^2 \end{pmatrix}$$

$$A^T b = \begin{pmatrix} \sum_{i=1}^m t_i \\ \sum_{i=1}^m b_i t_i \end{pmatrix}$$

Hence,

$$x^* = \begin{pmatrix} m & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & \sum_{i=1}^m t_i^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^m t_i \\ \sum_{i=1}^m b_i t_i \end{pmatrix}$$

Is the matrix invertible? Is there unique solution? its suffcient that we have two point where t_k are not equal, why? becuase if all t_k are equals, then we can take the average of b_k and every line which cross the average point in $t_1 = t_2 = ... = t_m$ will be a good solution.

Other interpetation to the problem. Assuming we choose two functio $\phi_1(t) = 1, \phi_2(t) = t$ and we want to construct a function which is combination of ϕ_1, ϕ_2 i.e.,

$$f(t) = \beta\phi_1(t) + \alpha\phi_2(t) = \alpha t + \beta$$

Where we want to minimize,

$$p(\alpha, \beta) = \sum_{i=1}^m \left(\beta\phi_1(t_k) + \alpha\phi_2(t_k) - b_k\right)^2 \underset{\alpha,\beta}{\to} min$$

This is exactly what we did but nore general, we can choose ϕ_1, ϕ_2 to be any function we want, and even we can add more functio $\phi_3, ..., \phi_n$ which will determin number of column in the matrix A in the least squares problem,

$$\left\| \begin{pmatrix} \phi_1(t_1) & t_1 \\ \phi_1(t_2) & t_2 \\ \phi_1(t_3) & t_3 \\ \vdots & \vdots \\ \phi_1(t_k) & t_m \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix} \right\|_2^2 = \min_x \|Ax - b\|_2^2 \to min$$

Let make it more interesting, assuming we are doing a physics simulation, where we got a sample of point in plane $\{t_i, b_i\}_{i=1}^m$ and we want to match this data to a curve, assuming that we want to approach the values by a function

$$f(t) = x_1 sin(3t) + x_2 ln(|t + 40| + 3) + x_3 t^{3.5}$$

We want to find x_1, x_2, x_3 in which the curve is the most suitable, so we can do that by formulating our problem as follows,

$$\underset{x_1,x_2,x_3}{Arg\ min} \sum_{i=1}^{m} (f(t_i) - b_i)^2$$

$$= \underset{x_1,x_2,x_3}{Arg\ min} \sum_{i=1}^{m} \left(x_1 sin(3t_i) + x_2 ln(|t_i + 40| + 3) + x_3 t_i^{3.5} - b_i\right)^2$$

So we obtain a system of equation as follow,

$$\left\| \begin{pmatrix} sin(3t_1) & ln\left(|t_1 + 40| + 3\right) & t_1^{3.5} \\ sin(3t_2) & ln\left(|t_2 + 40| + 3\right) & t_2^{3.5} \\ sin(3t_3) & ln\left(|t_3 + 40| + 3\right) & t_3^{3.5} \\ \vdots & \vdots & \vdots \\ sin(3t_m) & ln\left(|t_m + 40| + 3\right) & t_m^{3.5} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix} \right\|_2^2 = \min_x \|Ax - b\|_2^2 \to min$$

So we can do that more general,

- We have a set of point $\{t_i, b_i\}_{i=1}^m$
- The approxiomation is done by $\{\phi_k(t)\}_{k=1}^p$
- $f(t) = \sum_{i=1}^{p} x_k \phi_k(t)$
- The variables are $x_1, ..., x_P$

We start with the term,

$$\{\hat{x_i}\}_{i=1}^P = \underset{\{x_k(t)\}_{k=1}^p}{Arg\ min} \sum_{i=1}^{m} \left(\sum_{i=1}^{p} x_k \phi_k(t_i) - b_i\right)^2$$

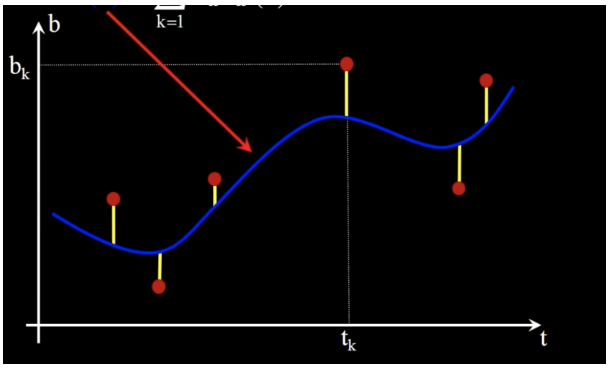
Where now the matrix in the least squares problem,

$$\left\| \begin{pmatrix} \phi_1(t_1) & \cdots & \phi_P(t_1) \\ \phi_1(t_2) & \cdots & \phi_P(t_2) \\ \phi_1(t_3) & \cdots & \phi_p(t_3) \\ \vdots & \cdots & \vdots \\ \phi_1(t_m) & \cdots & \phi_P(t_m) \end{pmatrix}_{mxP} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix} \right\|_2^2 = \min_x \|Ax - b\|_2^2 \to min$$

Where the approxiomation will propose the following $f(t)$ function,

$$\hat{f}(t) = \sum_{i=1}^{p} \hat{x_k} \phi_k(t)$$

Where our error is again sum of the sqaures yellow intervals in the following figure,



What can we do with the error we got?

- Noise reduction
- Invrease resolution for PNG images (Interpolation)
- prediction (Extrapolation)
- Compressing

Question. What are the functions $\{\phi_k(t)\}_{k=1}^{p}$? We have a freedom to choose any depend on the task understanding.

Example. We can choose polynomials then we get a well known matrix "Vandermonde" which is given by,

$$\begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^p \\ 1 & t_2 & t_2^2 & \cdots & t_2^p \\ 1 & t_3 & t_3^2 & \cdots & t_3^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^P \end{pmatrix}$$

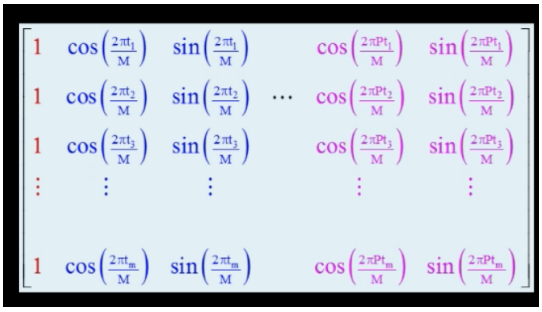
Where many interesting properties are related to this matrix

- If $P+1=m$ then the matrix is invertible if t_k are not equal
- Thos fact is known to us from Algebra which states that given m different point, there is only one and unique polynomial which cross them and with degree of $m-1$.
- If there is more points from our polynomial degree, then the matrix has a full rank if the $t_k$are different from each other, hence, for the LS problem there is unique solution.
- However, this type of matrices are sensitive and tend to cause numerical deviations

Other example. Harmonic function which is related to Fourier Transformation

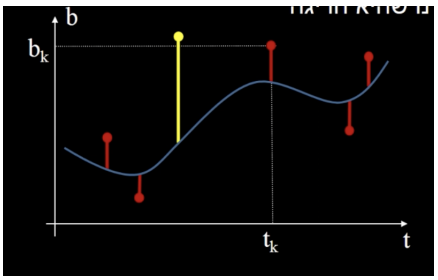
$$\{\phi_k(t)\}_k = \left\{ cos\left(\frac{2\pi kt}{M}\right), sin\left(\frac{2\pi kt}{M}\right) \right\}_{k=0}^{p}$$

Where the matrix look as follows,



Weighted Least Sqauares.

- Back to the Least Squares problem of matching curves we will add a additional data - collection of data $\{t_i, b_i, w_i\}_{i=1}^m$ where w_i is a importance of a point - higher value indicate more imortant.
- For example, assume for a yellpw point we want to give weight we want to give low weight in order to reflect yhe new data that it exceeded



- We want to match the data to a approxiomation in form $f(t) = \sum_{k=1}^{P} x_k \phi_k(t)$ where w_i taken into account.
- The term we want to minimzie is,

$$\{\hat{x_i}\}_{i=1}^P = \underset{\{x_k(t)\}_{k=1}^p}{Arg\ min} \underbrace{\sum_{i=1}^m w_i \left(\sum_{i=1}^p x_k \phi_k(t_i) - b_i \right)^2}_{\sum_{i=1}^N w_i r_i^2(x)}$$

- Note that w_i multiply the sum, so if a data t_i, b_i, w_i has importance then the error will be higher becasue w_i is higher.
- Now the LS problem is defined by a diagonal matrix W where in the digonal there are the weights

$$(Ax - b)^T W (Ax - b) = ||Ax - b||_w^2 \underset{x}{\rightarrow} min$$

- Openning the term yields,

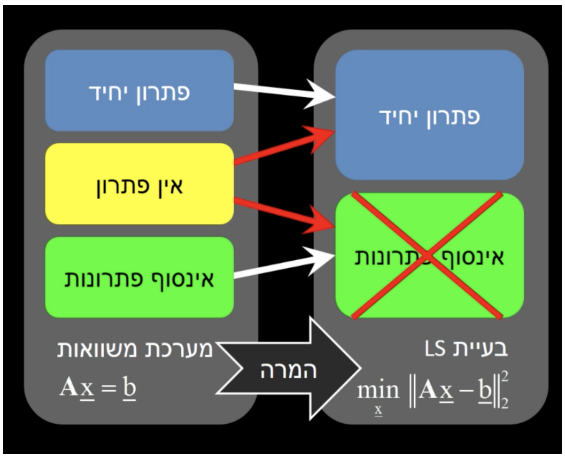
$$(Ax - b)^T W (Ax - b) = x^T A^T W A x - 2x^T A^T W b + b^T W b$$

- From here we obtain that the solution to the LS problem is x^* which imply,

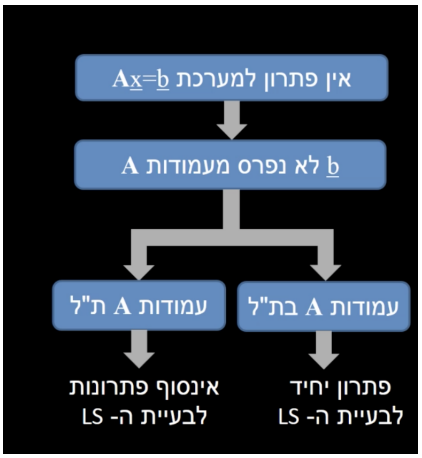
$$A^T W A x^* = A^T W b$$

- The solution will be unique of the colmumns of A are linearly independent and in assumtion that all the weights in the diagonal of W are positive. Why?

Regularization to the LS problems.



Assuming we want don't have solution then we can't do combination of A in which we obtain b now if A has L.I column then there is unique solution to the LS problem, if A has Linearly dependent column then there is inifnite solution to the LS problem,



regularization to LS problems.

- The main idea: A solution to the LS problem is given by
$$(A^T A)x^* = A^T b$$
- Inifnite solution happen is when $A^T A$ is not invertible. i.e., sinuglar

Observation. The matrix $A^T A + \lambda I$ for $\lambda > 0$ is PD.

Proof. Note that $\forall x \neq 0$,
$$x^T \left(A^T A + \lambda I\right) x = ||Ax||_2^2 + \lambda ||x||_2^2 \geq \lambda ||x||_2^2 > 0$$

□

- So we can do small modification in which the normal equation will be

$$\left(A^TA + \lambda I\right) x^* = A^Tb$$

Theorem. *To the following problem with $\lambda > 0$,*

$$\min_x \left(||Ax - b||_2^2 + \lambda||x||_2^2\right)$$

Then the minimun is given by, $x^* = \left(A^TA + \lambda I\right)^{-1} A^Tb$

- The term we added is $\lambda||x||_2^2$ call regulzation, and its called like that because the problem now is regular.

Proof. Again deffrentiating the term, first denote by

$$p(x) = ||Ax - b||_2^2 + \lambda||x||_2^2$$

So,

$$\nabla p(x) = 2A^T(Ax - b) + 2\lambda x = 0$$

$$\Rightarrow \left(A^TA + \lambda I\right) x = A^Tb$$

Since, $\left(A^TA + \lambda I\right)$ is PD we can inverse it then we are finished. □

Motivaition. There is battle between the two term we want want to minimze $||Ax - b||_2^2, \lambda||x||_2^2$ e.g., for $\lambda \to \infty$ we want $x = 0$ and the term $||Ax - b||_2^2$ is kind of neglected, and when $\lambda \to 0$ then we will got sometinh which is very close to the source LS problem. The problem is will known and many researched were devoted to investigate which λ is optimal.

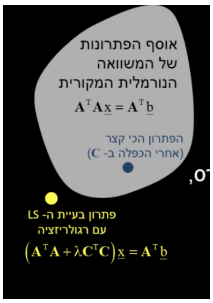
- We can propose better regulization in form, $\min_x \left(||Ax - b||_2^2 + \lambda||Cx||_2^2\right)$ where C^TC is PD, the solution to the new problem will be unique, again the solution obtained by deffrentiating, denote by,

$$p(x) = ||Ax - b||_2^2 + \lambda||Cx||_2^2$$

$$\nabla p(x) = 2A^T(Ax - b) + 2\lambda C^TCx = 0$$

$$\Rightarrow (A^TA + \lambda C^TC)^{-1}A^Tb = x^*$$

- If the LS problem has ∞ solution then the regulization will transform the problem to a new one that will have unique solution...but the solution obtained is not one of the ∞ solution but others so basically we didn't get out target, because the main motivation of defining the new problem with the regulzation was to get a solution to the problem but this is not the case.



We observe that of the problem $\min_x \left(||Ax-b||_2^2\right)$ given there is infinite solution then those are the solution of the normal equation $A^TAx = A^Tb$ so what was naive to do is that from all the solution we got that satisfy the normal solution we can pick the most shortest one which give the smallest $\lambda||Cx||_2^2$ but that not what we did! Instead we solve the equation,

$$(A^TA + \lambda C^TC)x = A^Tb$$

Which contradict the LS because the only x which imply,

$$A^TAx = A^T$$

$$(A^TA + \lambda C^TC)x = A^Tb$$

Is $x = 0$ but $x \neq 0$, so we did regulization becuase ∞ solution is not a something we want the most shortest of those solution, but it give us solution which is short i.e., smallest $\lambda||Cx||_2^2$ however, it contradict the LS so not fine with the LS. So what can we do? i.e., how to find a problemn in which we obtain unique short solution and also behaive well with the LS.

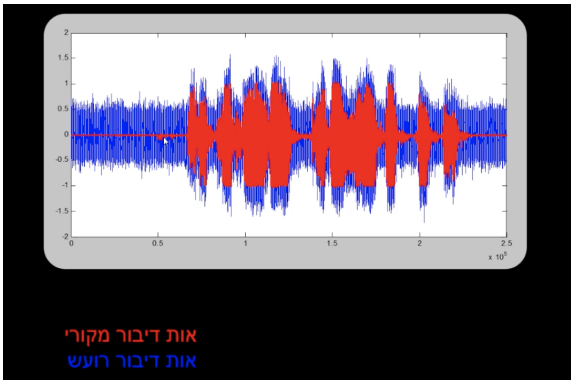
- We can define problem with constain i.e., we want $\min_x ||Cx||_2^2$ s,t, $A^T(Ax-b) = 0$ (Lagrange multipliers).

Example. (Noise reduction from voice signal). Some Basic facts:

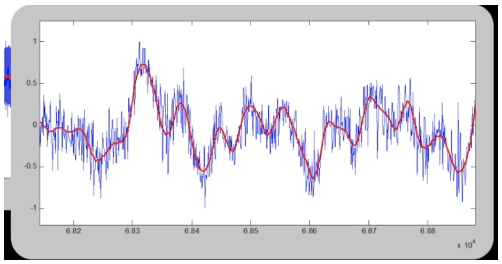
- A voice signal is a vector of numbers which describe the volume in time, we will focus on signal with $250k$ elements.
- When the voice is recorded, collected 44100 numbers every second - i.e., our signal refer to a record in length of 6 seconds.
- A noise in a record means a addition of random number (independent statistically) to the source numbers.
- Noise in a record heard as noise! We want to clean it in order to understand better whatever said in the record.

The things we are going to do is presenting simple LS. However, its worth to mention that,

- Exists more sophisticated way for noise reduction
- Things will be described in a superficial form with focusing on the relation to LS



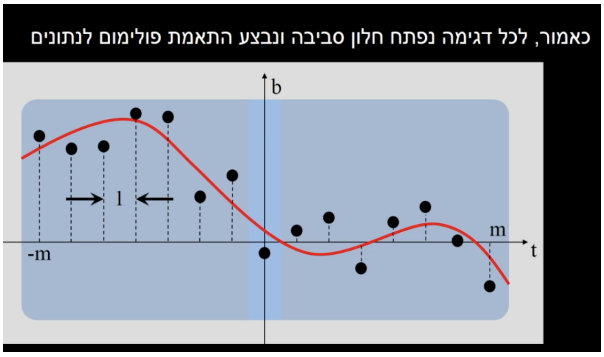
By zooming-in we can see that the graph will look as follow at each ms



Our hope is to try getting the graph above in red that we don't have now and the reason it interest us is because it give us the real voice we want to hear and not noise.

Toward solution.

- Assumption that our data is smooth and behave well where the noise signal jumo down/up randomly near the right values. This is very bad! becuase assuming we want to take a vector in since $250k$ and match it to polynomial with degree $20k$ this we want to avoid because we get a huge Vandermonde matrix. So what we can do instead?
- The other approach - is that in each point we will look at the point near it and focus only on this sample and for it we will match polynomial, we will do that for each sample in the vector of $250k$ samples.
- So at the end we will solve $250k$ LS problems, it will not be trerrible , will see now...



Our sample point k is the point in the middle where $t = 0$ and we will choose m point after and before. For this segment we will match polynomial by solving LS, afterward, we will know which height the obtained polynomial give in the origin then it will be the right value of the k sample, then we move to the next sample which is other black point and then we will put it in origin and check again the value of the polynmial in the origin and it will be the right value, we keep doing that for all $250k$ points.

Here is the LS problem for sample k,

$$\left\| \begin{pmatrix} (-m)^0 & (-m)^1 & (-m)^2 & \cdots & (-m)^P \\ (-m+1)^0 & (-m+1)^1 & (-m+1)^2 & \cdots & (-m+1)^P \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ (-1)^0 & (-1)^1 & (-1)^2 & \cdots & (-1)^P \\ (0)^0 & (0)^1 & (0)^2 & \cdots & (0)^P \\ 1^0 & 1^1 & 1^2 & \cdots & 1^P \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ (m-1)^0 & (m-1)^1 & (m-1)^2 & \cdots & (m-1)^P \\ m^0 & m^1 & m^2 & \cdots & m^P \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_P \end{pmatrix} - \begin{pmatrix} f_{k-m} \\ f_{k-m+1} \\ \vdots \\ f_{k-1} \\ f_k \\ f_{k+1} \\ \vdots \\ f_{k+m-1} \\ f_{k+m} \end{pmatrix} \right\|_2^2 \underset{x}{\to} min$$

$$x^* = (A^T A)^{-1} A^T b$$

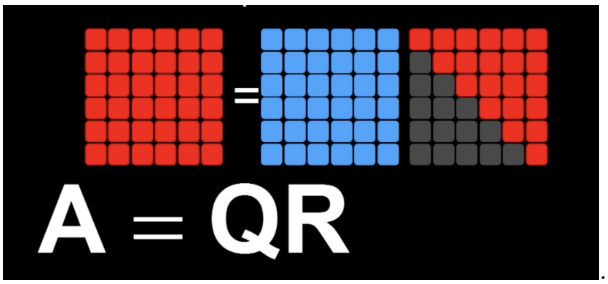
Note that the matrix A is constant becuase each time we choose a sample where it will be in the origin and we check m point after and before, however, the vector b will be changed, for example when we check samoke $k+1$ then in the middle will be f_{k+1} insteead of f_k.

Summary. Some words on the obtained LS problem we got:

- Its identical to what we saw earlier
- In all the $250k$ LS problem we will solve we have the same matrix A
- Conversely, the vector b will be different from problem to other
- We found polynomial for a problem that has sample k in the origin, what we can do with it? We will use it to find the clean value
- Since we are interested in the f_k sample which is in the middle where $t = 0$ the value if the polynomial obtained in this sample will be the clen value and evaluated by the equation, $\sum_{k=0}^{p} x_k t^k \mid_{t=0} = x_0$

Orthogonal matrices and QR decomposition

Motivation. In this chapter we are going to see a decomposition $A = QR$, we will deal with orthogonoemal matrices and understand their siginificant application. Also, we will see how to use QR for solving the LS problem.



. We can see that in the decompositio appear a upper tringular matrix, and the blue matrix which is going to be the orthonormal matrix which imply the property $Q^T Q = I \iff Q^{-1} = Q^T$.

Orthonormal basis.

- Assuming we have a matrix A in size of nxm where $n > m$ with a full rank, and a vector b.
- Our task: solving the least squares problem defined by the pair A, b and find a minimal height of the function $\min_x ||Ax - b||_2^2$.
- Remember that the solution where

$$x^* = \left(A^T A\right)^{-1} A^T b$$

$$||Ax^* - b||_2^2 = b^T b - (x^*)^T \left(A^T A\right)(x^*)$$

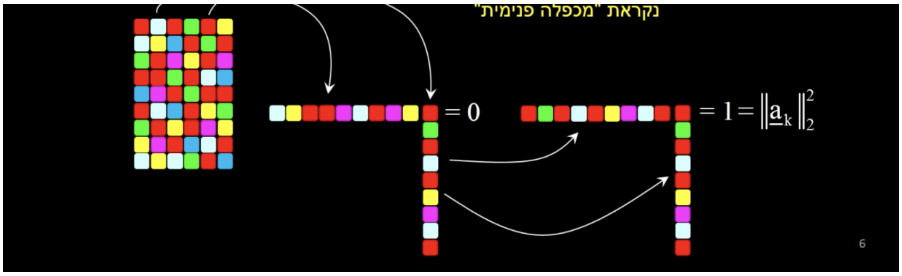
- Denote the columns of A by $a_1, ..., a_m$in the following way,

$$A = \begin{pmatrix} | & | & | & | \\ a_1 & a_2 & \cdots & a_m \\ | & | & | & | \end{pmatrix}$$

where

$$a_j^T a_k = \begin{Bmatrix} 1, & j = k \\ 0, & j \neq k \end{Bmatrix}$$

This operations called inner product.



Let solve the LS with knowing that A is orthonormal. By definition,

$$A^T A = \begin{pmatrix} -a_1^T- \\ -a_2^T- \\ \vdots \\ -a_m^T- \end{pmatrix} \begin{pmatrix} | & | & | & | \\ a_1 & a_2 & \cdots & a_m \\ | & | & | & | \end{pmatrix}$$

$$= \begin{pmatrix} a_1^T a_1 & a_1^T a_2 & \cdots & a_1^T a_m \\ a_2^T a_1 & a_2^T a_2 & \cdots & a_2^T a_m \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T a_1 & a_m^T a_2 & \cdots & a_m^T a_m \end{pmatrix}_{mxm} = I_{mxm}$$

We can see that it will be gift for us if we have the orthonormal property in our hands,

$$x^* = \left(A^T A\right)^{-1} A^T b =_{Orthonormal} A^T b$$

With

$$p(x^*) = b^T b - (x^*)^T (A^T A)(x^*) = ||b||_2^2 - \left(A^T b\right)^T \left(A^T A\right)\left(A^T b\right)$$

$$= ||b||_2^2 - b^T A A^T b = ||b||_2^2 - ||A^T b||_2^2$$

Corollary. *If we have A which is orthonormal, then the LS problem will be solved way easier.*

Theorem. *Given a set of m vector a_k which are orthonormal with dim n ($n > m$) then they are L.I.*

Proof. By definition, assume that they are dependent hence exists c_i where not all of them 0

$$\sum_{k=1}^{m} c_k a_k = 0$$

multuplting the saum in a_j for $j = 1, 2, ..., n$ will give,

$$0 = a_j^T \left(\sum_{k=1}^{m} c_k a_k \right) = \sum_{k=1}^{m} c_k a_j^T a_k = c_j \neq 0$$

since its true, we get that all the c_i are 0 and its contradition □

Definition. A matrix Q is orthonormal if its square matrix and its columns constitue a orthonormal base in $\mathbb{R}^n$. Some times we will use the term unitary matrix for complex numbers.

Remark. Since $Q^T Q = I \iff Q^{-1} = Q^T$ we can get some properties,

(1) For family of this matrices there is unique property - there inverse is obtained easily.
(2) Also the row will be orthonormal
(3) $det(I) = det(QQ^T) = det(Q)^2 \iff det(Q) = \pm 1$
(4) The multiplication of those matrices also orthonormal becuse

$$(Q_1 Q_2)^T (Q_1 Q_2) = Q_2^T Q_1^T Q_1 Q_2 = I$$

(5) Multiplying vector in orthonormal matrix will not change its length in L_2 since,

$$||Qx||_2^2 = x^T Q^T Q x = x^T x = ||x||_2^2$$

Last two proerties are obvious in geometry because orthonormal matrices are a rotation matrices relative to the origin hence, the distnace will be the same, and also composition of rotation is also rotation.

Examples.

1. The permutation matrix we saw in the LU decomposition is orthonormal becuase its changing raw order.

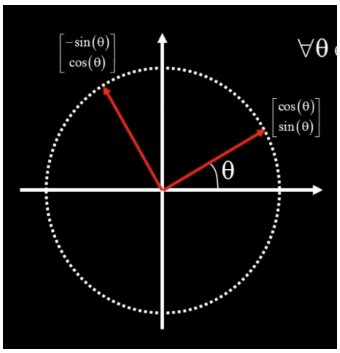
- The columns and raw are orthonormal for each other.

$$Q = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow Q^{-1} = Q^T = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

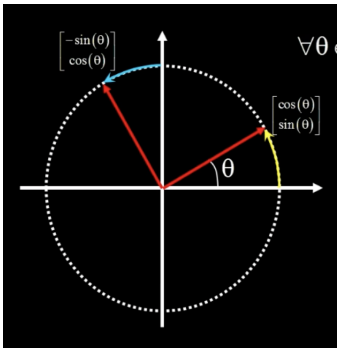
2. Rotation matrix in $\mathbb{R}^2$,

$$Q = \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix}, \forall \theta \in [0, 2\pi]$$

.



$$Q\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}cos(\theta)\\sin(\theta)\end{pmatrix}, Q\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}-sin(\theta)\\cos(\theta)\end{pmatrix}$$



- Reflection matrices,

$$Q=\begin{pmatrix}1&0\\0&-1\end{pmatrix}, Q=\begin{pmatrix}-1&0\\0&1\end{pmatrix}$$

3. Given matrices propose rotation between to axis in vector with high dimension,

$$G=\begin{pmatrix}1&&&&&&&&\\&\ddots&&&&&&&\\&&cos(\theta)&&&&-sin(\theta)&&\\&&&\ddots&&&&&\\&&&&1&&&&\\&&&&&\ddots&&&\\&&sin(\theta)&&&&cos(\theta)&&\\&&&&&&&\ddots&\\&&&&&&&&1\end{pmatrix}$$

. We can that there is rotation between to axis i, j only, where other axis are the same.

4. reflectios Householder matrices. Given that we have vector $v \in \mathbb{R}^n$ implies $v^T v = 1$ then the matrix

$$H = I - 2 \underbrace{vv^T}_{\text{matrix } mxm}$$

We will show that it orthonormal also symmetric,

$$H^T = \left(I - 2vv^T\right)^T = I^T - 2\left(v^T\right)^T v^T = \left(I - 2vv^T\right) = H$$

Otrhonormal becuase,

$$HH^T = \left(I - 2vv^T\right)\left(I - 2vv^T\right)^T =$$

$$\left(I - 2vv^T\right)\left(I - 2vv^T\right) = I - 2vv^T - 2vv^T + 4vv^T vv^T = I$$

Note that the motivation for the name of House-Holder is that for each $w \in S = \{w : wv = 0\} = v^{\perp}$ it keep its in the same direction because,

$$Hw = \left(1 - 2vv^T\right) w = w$$

However, for vector v the define the H matrix satisfied,

$$Hv = \left(1 - 2vv^T\right) v = v - 2vv^T v = -v$$

So v direction is changed unlike what happend in the orthogonal space of v.

Gram-Schmidt.

- We saw how much convient is to work with orthonormal basis
- The question we are going to deal with - given a base which is not orthonormal, or ayn set of vectors, we want to find a orthonormal base, how are we going to find one?
- Gram-Schmidt procedure help us to obtain that

Procedure.

- Given a set of input vectors $w_1, ..., w_L \in \mathbb{R}^n$.
- GS will construct from theese vectors a sequence of orthonormal vector with L or less vectors, $u_1, ..., u_L$.
- The procedure is sequential and satisfy the followinh property in every k iteration,

$$span\{w_1, ..., w_k\} = span\{u_1, ..., u_k\}$$

Remainder from Algebra.

Definition. V a vector space on $\mathbb{F}$, a biliniar form on V is a map $g : VxV \to \mathbb{F}$ where is linear in each variable, $\forall u, v, w \in V, \forall c \in \mathbb{F}$,

(1) $g(y, v + w) = g(u, v) + g(u, w)$
(2) $g(u, cv) = cg(u, v)$
(3) $g(u + v, w) = g(u, w) + g(v, w)$
(4) $g(cu, v) = cg(u, v)$

Examples.

1. A dot product in $\mathbb{R}^n$, $g(x, y) =< x, y >= \sum_{i=1}^{n} x_i y_i$. 2. Let $A \in Mat_{nxn}(\mathbb{F})$ we will define $g_A : \mathbb{F}^n x \mathbb{F}^n$ by

$$g_A(v, w) = v^T Aw = (v_1, ..., v_n) \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}$$

$$= \sum_{i,j=1}^{n} a_{ij} v_i w_j$$

Note that for $\mathbb{F} = \mathbb{R}, A = I_n$ we get $g_A =< \cdot, \cdot >$ - dot product.

3. Let $V = f \in \left(C^0[0, 1]\right)$ *- all continious functions* $[0, 1] \to \mathbb{R}$, *we can define*

$$g(f, h) = \int_0^1 f(t)h(t)dt$$

.

Remark. For defining inner product we can use the definition of biliniar form, however, in case $\mathbb{F} = \mathbb{C}$ this are very special biliniar forms and called sesquiliniar form and written as $1\frac{1}{2}$ linear form.

Definition. V a vector space on $\mathbb{C}$, a sesquiliniar form ($1\frac{1}{2}$ linear form) on V is a map $h : VxV \to \mathbb{C}$ where , $\forall u, v, w \in V, \forall c \in \mathbb{F}$,

(1) $h(\bar{u}, \bar{v} + \bar{w}) = h(\bar{u}, \bar{v}) + h(\bar{u}, \bar{w}))$
(2) $h(\bar{u} + \bar{v}, \bar{w}) = h(\bar{u}, \bar{w}) + h(\bar{v}, \bar{w})$
(3) $h(c\bar{u}, \bar{v}) = c \cdot h(\bar{u}, \bar{v})$
(4) $h(\bar{u}, c\bar{v}) = \bar{c} \cdot h(\bar{u}, \bar{v})$

Examples.

1. Given $\mathbb{F} = \mathbb{C}^n$ *where*

$$g(z, w) =< \bar{z}, \bar{w} >= \sum_{i=1}^{n} z_i \bar{w}_i$$

This is called standard Hermetian product.

2. Given $A \in Mat_{nxn}(\mathbb{C})$ where

$$g_A(\bar{z}, \bar{w}) = \bar{z}^T A \bar{\bar{w}} = (z_1, ..., z_n) A \begin{pmatrix} \bar{w_1} \\ \vdots \\ \bar{w_n} \end{pmatrix}$$

3. $V = \mathbb{C}_b[x], p, q \in V$ where,

$$g(p,q) = \int_0^1 p(t)\bar{q}(t)dt$$

.

$$= \int_0^1 Re\left(p(t)\bar{q}(t)\right) dt + i \cdot \int_0^1 Im\left(p(t)\bar{q}(t)\right) dt$$

Definition. V us a vector space over $\mathbb{F}$, $g : VxV \to \mathbb{F}$ is biliniar then,

(1) g is symmetric if $\forall u, w \in V$, satisfied $g(v, w) = g(w, v)$
(2) g is ani-symmetric if $\forall u, w \in V$, satisfied $g(v, w) = -g(w, v)$

Examples.

1. $<,>$ on $\mathbb{R}^n$ symmetric, becuase $< x, y >=< y, x >$.

2. $V = C[0,1]$ then $g(f,h) = \int_0^1 fhdt$ is symmetric.

3. $V = \mathbb{F}^2$ where $g\left(\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right) = det \begin{pmatrix} v_1 & w_1 \\ v_2 & w_2 \end{pmatrix}$ is antisymmetric.

Definition. V vector spave over $\mathbb{R}$. Inner product is a biliniar form symmetric and defined positive, $\forall v \in V, < v, v > \geq 0$ and equality when $v = 0$.

Definition. Given V vector space over $\mathbb{R}$ with $<,>$ inner product over V. Using inner product we can define:

(1) Length of vector defined by norm $||v|| = \sqrt{< v, v >}$. Where $<,>$ could be any of the example above, the most knows is when $<,>$ dot product and the norm is called euclidean norm defined by $< x, x >^{0.5} = \sqrt{|x_1|^2 + ... + |x_n|^2}$.

Exercise. Each norm in $\mathbb{R}^n$ obtained from a inner product g satisfy that,

$$||x + y||^2 + ||x + y||^2 = 2||x||^2 + 2||y||^2$$

Remark. Some norms are not obtained fomr a inner product for example $||x||_\infty = \max_i x_i$

(2) Unit vector is vector with length 1.
(3) Distance betweem $v, w \in V$ given by $d(v, w) = ||v - w||$
(4) A angle between two vectors define by $\alpha = arcos\left(\frac{<v,w>}{||v||\cdot||w||}\right)$

Definition. u, v vectors are called orthogonal respect to $<,>$ if $< u, w >= 0 \iff cos\alpha = 0 \iff \alpha = \frac{\pi}{2}$.

Notation. $v \perp w$.

Definition. V is a vector space with inner product, $< \cdot, \cdot >$, $B = \{v_1, ..., v_n\}$ is a base for V. Then,

- B is orthogonal if $v_i \perp v_j, \forall i \neq j$.
- B is orthonormal if
  - $v_i \perp v_j, \forall i \neq j$.
  - $||v_i|| = 1, \forall i$

Examples.

1. Standard base is orthonormal.

2. Given that $\{v_1, .., v_n\}$*is orthogonal base* $\Rightarrow \left\{ \frac{v_1}{||v_1||}, ..., \frac{v_n}{||v_n||} \right\}$ *is orthonormal.*

3. $dim(V) = 1 \Rightarrow \{v\}$*is a orthogonal base.*

4. Given the matrix

$$[g]_B = [g(v_i, v_j)]_{i,j=1}^n = I_n$$

Note that

$$g = \begin{pmatrix} < v_1, v_1 > & \cdots & < v_1, v_n > \\ \vdots & \ddots & \vdots \\ < v_n, v_1 > & \cdots & < v_n, v_n > \end{pmatrix} = I$$

.

Claim. Let B be orthonormal base $B = \{v_1, .., v_n\}$ over inner product space V, then $\forall w \in V$ then

$$w =< w, v_1 > v_1 + ... + < w, v_n > v_n$$

$$\iff [w]_B = \begin{pmatrix} < w, v_1 > \\ \vdots \\ < w, v_n > \end{pmatrix}$$

Proof. Let $w = \sum_{i=1}^{n} \lambda_i v_i$ applying inner product with v_j gives,

$$< w, v_j > = < \sum_{i=1}^{n} \lambda_i v_i, v_j > = \sum_{i=1}^{n} \lambda_i \underbrace{< v_i, v_j >}_{\substack{i=j, \quad 1 \\ i \neq j, \quad 0}} = \lambda_j$$

Since its true for all j then we are finished. □

GS Psuedo-Code.

Gram-Schmidt Algorithm.

Algorithm 1 Gram schmidt

Initalize:

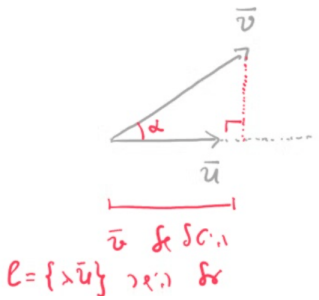
- Let $k = 1$
- Take $u_1 = w_1$

Iteration:

- $u_k = w_k - \sum_{j=1}^{k-1} (u_j^T w_k) u_j$
- Normalize: $u_k = \frac{1}{||u_k||_2} u_k$
- $k = k + 1$
- Take w_k

Geometric iniution for the formula of orthogonal projection. Remember that,

$$\alpha = arcos\left(\frac{< v, w >}{||v|| \cdot ||w||}\right) \iff ||v|| \cdot cos\alpha = \frac{< u, v >}{||u||}$$



Note that the projection vector of v must be in length of

$$||v||cos\alpha = \frac{< u, v >}{||u||}$$

and in direction of u so we can multiply it with,

$$\frac{u}{||u||}$$

and we get,

$$\frac{< u, v >}{||u||}$$

So the projection vector is,

$$\frac{< u, v >}{||u||} \frac{u}{||u||} = \frac{< u, v >}{< u, u >} u$$

where

$$||u||^2 =< u, u >$$

since,

$$< u, u >^{0.5} = ||u||$$

Then, we can write v as a sum of two components one is in direction of u and other is orthogonol to u direction,

$$v = \underbrace{\frac{< u, v >}{< u, u >} u}_{\text{In } u \text{ direction}} + \left(\underbrace{v - \frac{< u, v >}{< u, u >} u}_{\text{Orthogonal to } u \text{ direction}} \right)$$

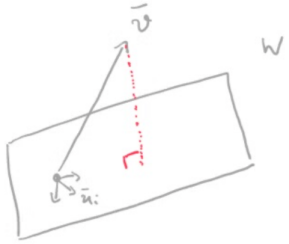
So we can generelize that for set of vectors, $v_1, ..., v_n$ which are not orthogonal, and assumume that we found $W = \{u_1, ..., u_{k-1}\}$ which are orthogonal, then the projection of v on W will be,

$$\sum_1^{k-1} \underbrace{\frac{< u_i, v >}{< u_i, u_i >} u_i}$$

And the component of v which is orthogonal to W will be,

$$v - \sum_1^{k-1} \underbrace{\frac{< u_i, v >}{< u_i, u_i >} u_i}$$

as descirbed in the figure below in red,



Algorithm Gram-Schmidt.

Step 1. We will construct an orthogonal base with the property $u_i \perp u_j, \forall i \neq j$. *Also,* $\forall k, span\{u_1, ..., u_k\} = span\{v_1, .., v_k\}$ ..

Step 2. We will define $C = \left\{ w_1 = \frac{u_1}{||u_1||}, ..., w_n = \frac{u_n}{||u_n||} \right\}$, *note that* C *is orthonormal because,*

$$< w_i, w_j > = < \frac{u_i}{||u_i||}, \frac{u_j}{||u_j||} > = \frac{< u_i, u_j >}{||u_i|| \cdot ||u_j||} =_{i \neq j} 0$$

and for $i = j$ *we get* 1. *Also,*

$$span\{w_1, ..., w_k\} = span\{u_1, .., u_k\} =_{step-1} span\{v_1, .., v_k\}$$

.

Construction of step 1. Induction on K,

Base for $k = 1$, we take $u_1 = v_1$ assuming we did $k-1$ and construct orthogonal base $\{u_1, .., u_{k-1}\}$ in which $u_i \perp u_j$ for all $1 \leq j \neq k \leq k-1$ also, span $\{u_i\}_{i=1}^{k-1} =$ span $\{v_i\}_{i=1}^{k-1}$ we want u_k s.t. $u_k \perp u_i$ for all $1 \leq i \leq k-1$. Let,

$$u_k = v_k - \sum_1^{k-1} \underbrace{\frac{< u_i, v_k >}{< u_i, u_i >} u_i}$$

. Then, note that, $\forall 1 \leq j \leq k$,

$$< u_k, u_j > = < v_k - \sum_1^{k-1} \underbrace{\frac{< u_i, v_k >}{< u_i, u_i >} u_i}, u_j >$$

$$=_{Linear} < v_k, u_j > - < \sum_1^{k-1} \underbrace{\frac{< u_i, v_k >}{< u_i, u_i >} u_i}, u_j >$$

Now, note that,

$$< \sum_1^{k-1} \underbrace{\frac{< u_i, v_k >}{< u_i, u_i >} u_i}, u_j > = \sum_1^{k-1} \underbrace{\frac{< u_i, v_k >}{< u_i, u_i >} < u_i, u_j >} = \frac{< u_j, v_k >}{< u_j, u_j >} < u_j, u_j >$$

So, in total,

$$< u_k, u_j > = < v_k, u_j > - \frac{< u_j, v_k >}{< u_j, u_j >} < u_j, u_j > = 0$$

I.e., u_k is orthogonal to u_j as expected. Now, let show that

$$span\{u_1, .., u_k\} = span\{v_1, .., v_k\}$$

Note that $\forall 1 \leq i \leq k-1$,

$$u_i \in span\{v_1, .., v_{k-1}\}$$

$$u_k \in span\{u_1, .., u_{k-1}, v_k\} \subseteq span\{v_1, ..v_k\}$$

Hence,

$$span\{u_1, .., u_k\} \subseteq span\{v_1, ..v_k\}$$

In other hand,

$$v_1, .., v_{k-1} \in span\{u_1, .., u_k\}$$

Also,

$$v_k = u_k + \sum_1^{k-1} \underbrace{\frac{< u_i, v_k >}{< u_i, u_i >} u_i} \in span\{u_1, .., u_k\}$$

Hence,

$$span\{v_1, .., v_k\} \subseteq span\{u_1, ..u_k\}$$

So we got equality.

GS variations.

variation 1.

- Step 1 - we got w_1 and we want to create u_1, we can do that by normalizing the vectors,
$$u_1 = \frac{1}{||w_1||_2^2} w_1$$

Remark. If $||w_1||_2^2 = 0$ this this is 0 vector so we throw it. Its obvious that $span\{w_1\} = span\{u_1\}$.

- Step 2 - we have w_2 and we want to create u_2, so our requirments that u_2 must satisfy are,
  - $||u_2||_2^2 = 1$
  - $u_2 \perp u_1$
  - $span\{w_1, w_2\} = span\{u_1, u_2\}$
- The method: peeling u_1 from w_2 to obtain orthogonality, then normalizing,
$$u_2 = w_2 - cu_1$$
$$u_k = w_k - \sum_1^{k-1} \underbrace{\frac{< u_i, w_2 >}{< u_i, u_i >} u_i}, k = 2$$
Where $c = \frac{<u_1, v_2>}{<u_1, u_1>} =$ Now, they second property satisfied becuase,
$$\left[0 = u_1^T u_2 = u_1^T w_2 - c\underbrace{u_1^T u_1}_{1} \Rightarrow c = u_1^T w_2 \right]$$
Normalizing the vector u_2 with,
$$u_2 = \frac{1}{||u_2||_2^2} u_2$$
The third property is satisfied, because u_1, u_2 are constructed from linear combinatio of w_1, w_2 hence $span\{u_1, u_2\} \subseteq span\{w_1, w_2\}$. Since the dimension of the two subspaces is 2 we have equality.
- Now, we want u_3 to satisfy the same three properties, the norm of it is 1, and also orthogonal to u_1, u_2 also satisfy the requirment $span\{w_1, w_2, w_3\} = span\{u_1, u_2, u_3\}$. Again, define,
$$u_3 = w_3 - c_1 u_1 - c_2 u_2$$
where we know that,
$$0 = u_1^T u_3 = u_1^T w_3 - c_1 \Rightarrow c_1 = u_1^T w_3$$
$$0 = u_2^T u_3 = u_2^T w_3 - c_2 \Rightarrow c_2 = u_2^T w_3$$
Moreover,
$$u_3 = \frac{1}{||u_3||_2} u_3$$
- In step k we define,
$$u_k = w_k - \sum_{i=1}^{k-1} c_i u_i$$

where

$$c_i = \underbrace{\frac{< u_i, w_k >}{< u_i, u_i >}}_{} = u_i^T w_k$$

and again we normalize,

$$u_k = \frac{1}{||u_k||_2} u_k$$

Theorem. *Given a set of vector L.I denote by* $\{w_k\}_{k=1}^L$ *which define subspace then exists a orthonormal base* $\{u_k\}_{k=1}^L$ *which will span the same space i.e.,* $span\{w_1, .., w_L\} = span\{u_1, .., u_L\}$.

Proof. GS procedure, we show it already. □

Is the orthonormal base unique?

No!, note that we used specific order from the possible $n!$, or we can change in the code in the iteration $u_k = \frac{-1}{||u_k||_2} u_k$ and still work. Using the fact thata the span will be the same always so there could be many ways in which we obtian different orthonormal base for the same set of vectors the algorithm get as input.

Modified Gram-Schmidt.

- We are going to disccuss a other variation of GS
- This variation is equivalent to the one discussed earlier
- Its main idea of this variation will be emphsized when we discuss QR
- The method: rely on the equation which connect between vectors in input and output.

$$w_1 = r_{11} u_1$$
$$w_2 = r_{21} u_1 + r_{22} u_2$$
$$w_3 = r_{31} u_1 + r_{32} u_2 + r_{33} u_3$$
$$\vdots$$
$$w_L = r_{L1} u_1 + r_{L2} u_2 + ... + r_{LL} u_L$$

- In the first equatiom we will choose r_{11} to get a normaliztion of the vector u_1 which is,

$$||w_1||_2^2 = ||r_{11} u_1||_2^2 = r_{11}^2 u_1^T u_1 = 1$$
$$r_{11}^2 = ||w_1||_2^2$$

  so the first equation is $w_1 = \pm ||w_1||_2 u_1$.
- Ragarding second equation,

$$w_2 = r_{21} u_1 + r_{22} u_2$$
$$\Rightarrow u_2 = \frac{1}{r_{22}} (w_2 - r_{21} u_1)$$

  since u_1, u_2 are orthogonal we get that r_{21} is known because,

$$u_1^T w_2 = r_{21} u_1^T u_1 + r_{22} u_1^T u_2 = r_{21}$$

  Also now that,

$$||w_2||_2^2 = (r_{21} u_1 + r_{22} u_2)^T (r_{21} u_1 + r_{22} u_2)$$
$$= r_{21}^2 + r_{22}^2$$

hence,

$$r_{22} = \pm\sqrt{||w_2||_2^2 - r_{21}^2}$$

- Now, for

$$w_k = r_{k1}u_1 + r_{k2}u_2 + ... + r_{kk-1}u_{k-1} + r_{kk}u_k$$

Because of the orthogonality property of u_k to u_j for $j = 1, 2, ..., k-1$ we get the $k-1$ coeffcient are known because,

$$u_j^T w_k = r_{k1}u_j^T u_1 + r_{k2}u_j^T u_1 + ... + r_{kj}u_j^T u_j + ... + r_{kk}u_j^T u_k = r_{kj}$$

Now, the coeffcient r_{kk} is determined in order to get nomrlization for u_k therefore,

$$||w_k||_2^2 = ||r_{k1}u_1 + r_{k2}u_2 + ... + r_{kk-1}u_{k-1} + r_{kk}u_k|| = \sum_{j=1}^{k} r_{kj}^2$$

$$\Rightarrow r_{kk} = \pm\sqrt{||w_k||_2^2 - \sum_{j=1}^{k-1} r_{kj}^2}$$

Stable Gram-Schmidt (SGS).

- This is the third variation of GS
- This method is equivalent to the variation described earlier
- The goal of this method as its name is to obtain stable GS numerically, and therefore, its way more stable than variations we saw.
- First step - we get w_1 and we want to create u_1 the method is to normalize the input vector $u_1 = \frac{1}{||w_1||_2} w_1$
- Till now the treament is the same as previous variation, but now come the big difference,
- Before going on w_2 we will peel all the given vectors by peeling u_1 from them, this mean that we other vectors will be ortogonal to u_1 this is done by,

$$\forall j = 2, 3, ..., L, w_j \leftarrow w_j - \left(u_1^T w_j\right) u_1$$

- Now, we got w_2 and we want to create u_2
- Before, we wanted to peel the vector the u_1 component and then normilze it, however, since we did step before, we need only to normalize it so

$$u_2 = \frac{1}{||w_2||_2} w_2$$

- Before going on w_3 we will peel all the given vectors by peeling u_1 from them, this mean that we other vectors will be ortogonal to u_2 this is done by,

$$\forall j = 3, 4, ..., L, w_j \leftarrow w_j - \left(u_2^T w_j\right) u_2$$

- This is going to be continue till step k - obtained w_k and wawnt to create u_k
- first we normilize

$$u_k = \frac{1}{||w_k||_2} w_k$$

- Now we peel all vectors in order to make them orthogonal to u_k so

$$\forall j = k+1, ... \; L, w_j \leftarrow w_j - \left(u_k^T w_j\right) u_k$$

Algorithm 2 Stable Gram schmidt

Initalize:

- Let $k = 1$
- Take w_1

Iteration:

- Normalize $u_k = \frac{w_k}{||w_k||_2}$
- Peel: $w_j \leftarrow w_j - \left(u_k^T w_j\right) u_k$ for $k < j \leq L$
- $k = k + 1$

Complexity of GS..

- Assuming we are working on $n > L$ vectors which are L.I in dim of n.
- Let analyse the SGS algorithm
  - Each normaliztion will cost $2n$ operations, $u_k = \frac{w_k}{||w_k||_2}$
  - Each peeling operation will cost $2n$ operations, $w_j \leftarrow w_j - \left(u_k^T w_j\right) u_k$
- Number of total opertaion is $\frac{L^2}{2}$ and we have to multiply it by $2n$ hence nL^2 operations.

Remark. When we are working on n vectors (i.e., matrix of size nxn) then we will have n^3 operaions.

צעד	מספר פעולות נירמול	מספר פעולות קילוף
1	1	L-1
2	1	L-2
	...	
L	1	0
סיכום:	L פעולות	L(L-1)/2 פעולות

Numerically sensitivty of GS.

- We already disccussed 3 varitions of the GS
  - GS
  - modified GS
  - Stable GS
- All of them are equivalent theoritically
- In a numerical treatment, the first two methods are so sensitive to rounding error, where SGS is more solid

Example. Assuming in our have we have the set of vectors,

$$W = \begin{bmatrix} 1 & 1 & 1 \\ e & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix}$$

If we assume that e is very small number - in which the computer gives $1 + e = e$.

What GS will do? First,

$$w_1 = \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} \Rightarrow u_1 = \frac{1}{\sqrt{1+e^2}} \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix}$$

Note that already we aggerated small error which going to acculmulate from one vector to the other vector, now we are going to peel so we take,

$$w_2 = \begin{bmatrix} 1 \\ 0 \\ e \\ 0 \end{bmatrix} \Rightarrow v_2 = w_2 - (u_1^T w_2)u_1 = \begin{bmatrix} 1 \\ 0 \\ e \\ 0 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -e \\ e \\ 0 \end{bmatrix}$$

Now,

$$||w_2|| = \sqrt{e^2 + e^2} = \sqrt{2}e$$

$$u_2 = \frac{w_2}{||w_2||_2} = \frac{1}{\sqrt{2}e} \begin{bmatrix} 0 \\ -e \\ e \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

Now,

$$w_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e \end{bmatrix} \Rightarrow u_3 = w_3 - \underbrace{(u_1^T w_3)}_{1} u_1 - \underbrace{(u_2^T w_3)}_{0} u_2$$

$$= \begin{bmatrix} 1 \\ 0 \\ 0 \\ e \end{bmatrix} - 1 \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} - 0 \cdot \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -e \\ 0 \\ e \end{bmatrix}$$

So the matrikx we obtain for now is

$$U = \begin{bmatrix} 1 & 0 & 0 \\ e & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ 0 & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{11}{\sqrt{2}} \end{bmatrix}$$

$$G = \begin{pmatrix} u_1^T u_1 & u_1^T u_2 & u_1^T u_3 \\ u_2^T u_1 & u_2^T u_2 & u_2^T u_3 \\ u_3^T u_1 & u_3^T u_2 & u_3^T u_3 \end{pmatrix} = \begin{pmatrix} 1 & -0.707e & -0.707e \\ -0.707e & 1 & 0.5 \\ -0.707e & 0.5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix}$$

Note that u_2 is not orthogonal to u_3 and the numerical diviation from 0 is $\frac{1}{2}$ which is very high, so its real problem!!!

What SGS will do? First,

$$w_1 = \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} \Rightarrow u_1 = \frac{1}{\sqrt{1+e^2}} \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix}$$

Now, we statrt peeling w_2, w_3 we have,

$$w_2 = w_2 - (u_1^T w_2)u_1 = \begin{bmatrix} 1 \\ 0 \\ e \\ 0 \end{bmatrix} - 1 \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -e \\ e \\ 0 \end{bmatrix}$$

$$w_3 = w_3 - (u_1^T w_3)u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e \end{bmatrix} - 1 \begin{bmatrix} 1 \\ e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -e \\ 0 \\ e \end{bmatrix}$$

Now, we take w_2 annd we normalize it,

$$w_2 = \begin{bmatrix} 0 \\ -e \\ e \\ 0 \end{bmatrix} \Rightarrow u_2 = \frac{1}{\sqrt{e^2+e^2}} \begin{bmatrix} 0 \\ -e \\ e \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

now, we peel w_3 by,

$$w_3 = w_3 - (u_2^T w_3)u_2 = \begin{bmatrix} 0 \\ -e \\ 0 \\ e \end{bmatrix} - \frac{e}{\sqrt{2}} \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{e}{2} \\ -\frac{e}{2} \\ e \end{bmatrix}$$

And now we normalize, w_3 and we get,

$$w_3 = \begin{bmatrix} 0 \\ -\frac{e}{2} \\ -\frac{e}{2} \\ e \end{bmatrix} \Rightarrow u_3 = \frac{1}{\sqrt{1.5e^2}} \begin{bmatrix} 0 \\ -\frac{e}{2} \\ -\frac{e}{2} \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{bmatrix}$$

Now,

$$U = \begin{bmatrix} 1 & 0 & 0 \\ e & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{6}} \end{bmatrix}$$

$$G = \begin{pmatrix} u_1^T u_1 & u_1^T u_2 & u_1^T u_3 \\ u_2^T u_1 & u_2^T u_2 & u_2^T u_3 \\ u_3^T u_1 & u_3^T u_2 & u_3^T u_3 \end{pmatrix} = \begin{pmatrix} 1 & -0.707e & \frac{-e}{\sqrt{6}} \\ -0.707e & 1 & 0 \\ \frac{-e}{\sqrt{6}} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

So not we get vectors which are nicely orthogonal to each other.

QR Decomposition.

- We will start from the GS algorithm in its second version the modified GS and we want to understand what was really proposed
- Set of equation the procedure constructs are

$$w_1 = r_{11}u_1$$

$$w_2 = r_{21}u_1 + r_{22}u_2$$

$$w_3 = r_{31}u_1 + r_{32}u_2 + r_{33}u_3$$

$$\vdots$$

$$w_L = r_{L1}u_1 + r_{L2}u_2 + ... + r_{LL}u_L$$

Observe that the Gram procedure may be written as matrix mutiplication, as follows,

$$A = \begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ w_1 & w_2 & w_3 & \cdots & w_L \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ u_1 & u_2 & u_3 & \cdots & u_L \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} r_{11} & \cdots & r_{L1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{LL} \end{bmatrix} = QR$$

- The relation we saaw is true because we see that

$$Ae_1 = w_1 = r_{11}u_1$$

Moreover,

$$Ae_2 = w_2 = r_{21}u_1 + r_{22}u_2$$

and so forth..

Theorem. *Every sqaure matrix and non-singular (invertible) A could be written as unique decomposition $A = QR$ where Q is orthonormal matrix and R is upper tringular matric in which all the diagonal elements are positive.*

Remark. We required that all the elements positive because in thee procedure we got freedom to divide in $\pm||w_i||_2$ but now we will choose only the $+$ so we get unique solution.

Proof. Since the matrix is invertible then its column are L.I. hence in the procdure of GS will obtain n output vectors and generate orthonormal matrix. □

- Note the QR exists also for singular matrix, then we will obtain 0 values in the main diagonal elements R.

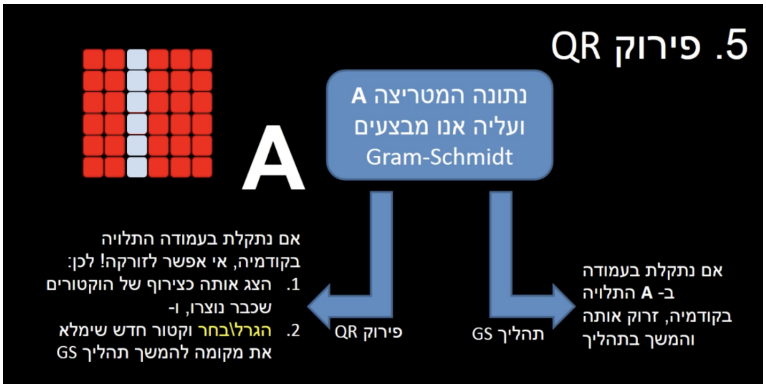
Example. We will look at the matrix and get Matlab QR. We will note that its L.I hence those matrix singular

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

we will get,

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 3 & 0 \\ 1 & 1 & 3 & 1 \end{pmatrix} = QR$$

$$= \begin{pmatrix} 0.5 & -0.5 & 0.707 & 0 \\ 0.5 & -0.5 & -0.707 & 0 \\ 0.5 & 0.5 & 0 & -0.707 \\ 0.5 & 0.5 & 0 & -0.707 \end{pmatrix} \begin{pmatrix} 2 & 1 & 5 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & \mathbf{0} & -0.707 \\ 0 & 0 & 0 & 0.707 \end{pmatrix}$$



Example. We can see that there is marked 0 in R matrix which tell us that the third column in the source marix is linerly dependent in column before. Moreover, we got 4 orthonormal vectors which is satisfying result, but what Matlab did? Unlike the GS algorithm which throw the linearly dependent vector and give only 3 vector output. However in Matlab this is not option in fact it consider the linearly dependent vector so the first thing it check whether it L.I we can see that w_3 which is $\begin{bmatrix} 2 \\ 2 \\ 3 \\ 3 \end{bmatrix}$ is $5u_1 + u_2$ where $u_1 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$ and $u_2 = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$ therefore, what matlab do is choosing randim vector gor exmple in this case $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and apply GS on it which gives $u_3 = \begin{bmatrix} 0.707 \\ -0.707 \\ 0 \\ 0 \end{bmatrix}$ now the forth vector continue as normal, i.e., check whether w_4 is $L.I$ in u_1, u_2, u_3 if no then again we again choose random vector and apply GS on it...