Linear Algebra

# Principal Component Analysis (PCA) & Jacobian Matrix, Hessian Matrix

Automotive Intelligence Lab.

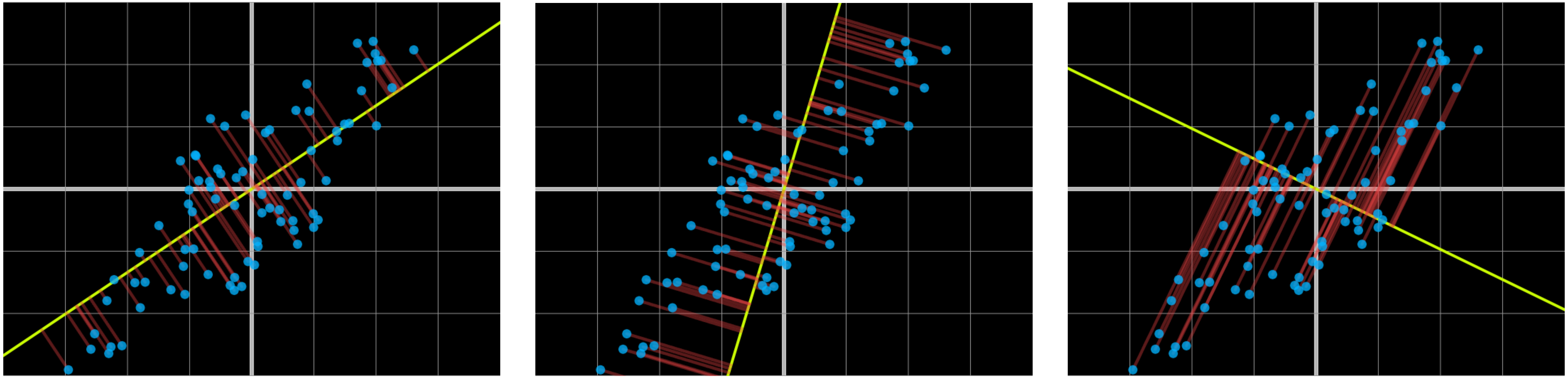
# Contents

- PCA (Principal component analysis)
- Jacobian matrix
- Hessian matrix

# Principal Component Analysis (PCA)

# What PCA Says

- **If data is orthogonally projected to reduce dimension…,**
  - Onto which vector should the data be projected to best maintain the original structure of the data?



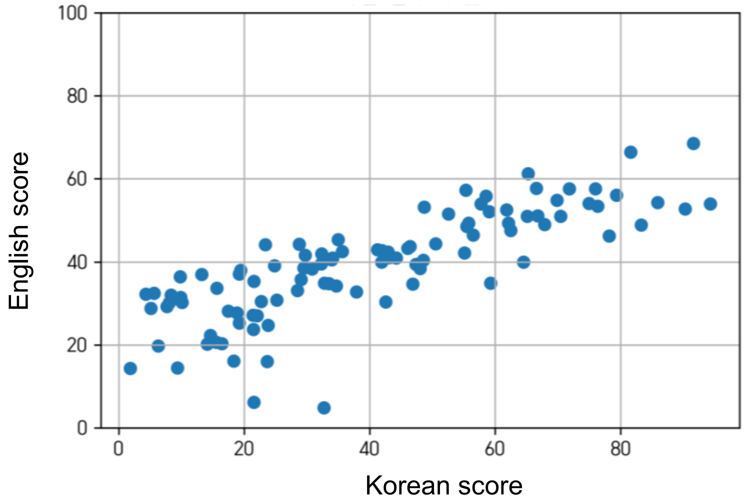
Orthogonally project data to reduce dimension

# PCA: Method for Calculating Composite Score Well

## Let's consider that 100 students took a Korean and an English test.

- Consider English test as a bit more difficult.
  - Some of the results were approximately as following table.

| Score | |
|---|---|
| Korean | English |
| 100 | 83 |
| 70 | 50 |
| 30 | 25 |
| 45 | 30 |
| ⋮ | ⋮ |
| 80 | 60 |



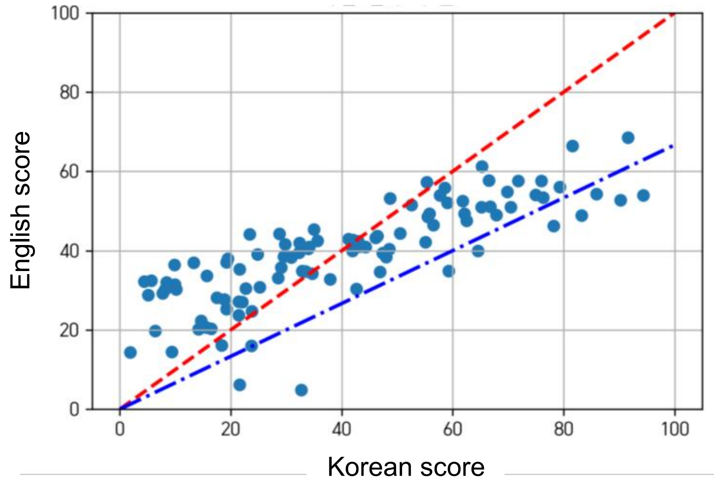

Distribution of test scores

- How to make **composite score** of Korean and English?
  - Just taking average of the two scores.
  - In other case, Adding the scores of Korean to English with weight of $6:4$.
    - Since English test was relatively more difficult.
  - How to express it mathematically?

# Express Composite Score Mathematically

- **For example, student A scored $100$ in Korean and $80$ in English.**
  - (1) Taking average with a ratio of $5:5$ means.
    - $100 \times 0.5 + 80 \times 0.5$
  - (2) Calculating a composite score with a ratio of $6:4$ means.
    - $100 \times 0.6 + 80 \times 0.4$
  - Equation (1) is dot product of the vector $[100 \quad 80]$ with the vector $[0.5 \quad 0.5]$.
  - Equation (2) is dot product of the vector $[100 \quad 80]$ with the vector $[0.6 \quad 0.4]$.

- **When obtaining a composite score**
  - Method of getting it with a ratio of 5:5 or 6:4 can be mathematically dealt with...,
    - To the problem of **dot product** of the score vector with a **vector representing the specific ratio**.
    - Dot product: geometrically, **orthogonal projection**.
  - So what main point needs to be considered?




Lines with a ratio $5:5$ (red) and $6:4$ (blue)

# Main Point of PCA

## Main Consideration

- Which vector does dot product (or orthogonal projection) of a data vector to give the optimal result?

## Secondary consideration

- Isn't it better to find a vector that moves with the center of the data distribution as **pivot axis**?
  - While finding a vector (or axis) for orthogonal projection

## Solution to these problems

- Can be found from the **covariance matrix**.

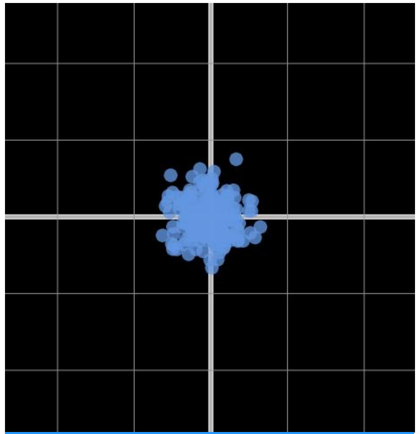
## Covariance matrix

- Mathematical method that describes the **structure (or shape)** of the data.
- Particularly represents how much the variations of feature pairs similar to each other.
  - In other words, to what extent they vary together.

# Geometric Meaning of Covariance

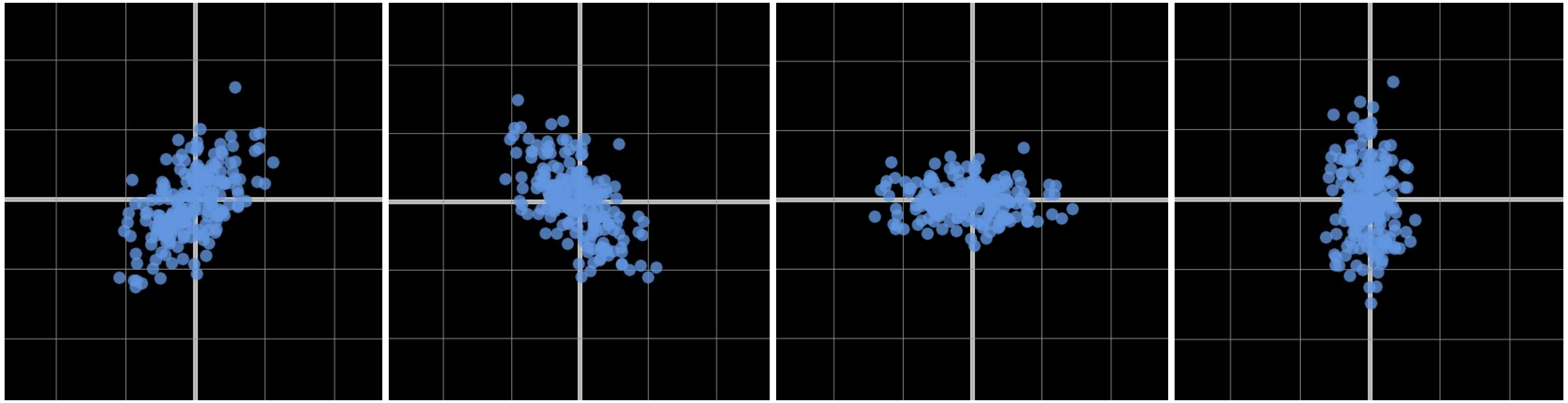
## What happens when a matrix is applied to data?

- Apply a matrix to perform a linear transformation.
- Can check covariance values of the results of the linear transformation.
- Let's talk about first example (covariance matrix $\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$) in next page.



Original data distribution

**Linear transformation**

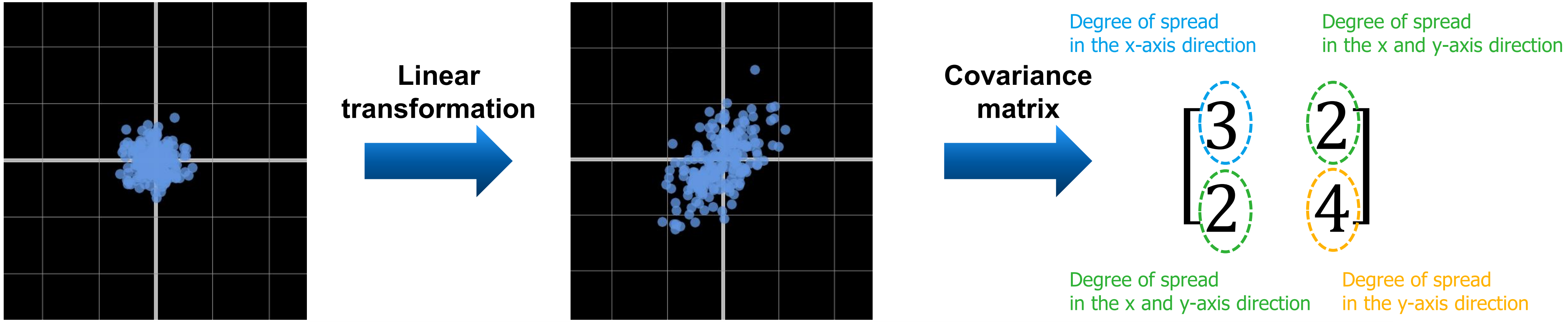


$\boxed{\text{Covariance Matrix}}$ : $\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$ $\begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$ $\begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$

Covariance matrix obtained through the results after linear transformation by each matrix

# Explanation about Covariance

- **Element in first row and first column of Matrix 1.**
  - Represents variance of the first feature.
  - Tells how much to spread in the x-axis direction
- **Elements in the first row, second column, and second row, first column.**
  - Each tell us how much to spread in the $x$ and $y$ axes **together**.
- **Element in second row and second column.**
  - Tells how much spread in the y-axis direction
- **Let's think about covariance and eigenvectors in relation to each other.**




Geometric meaning of covariance

# Covariance Matrix and Eigenvectors

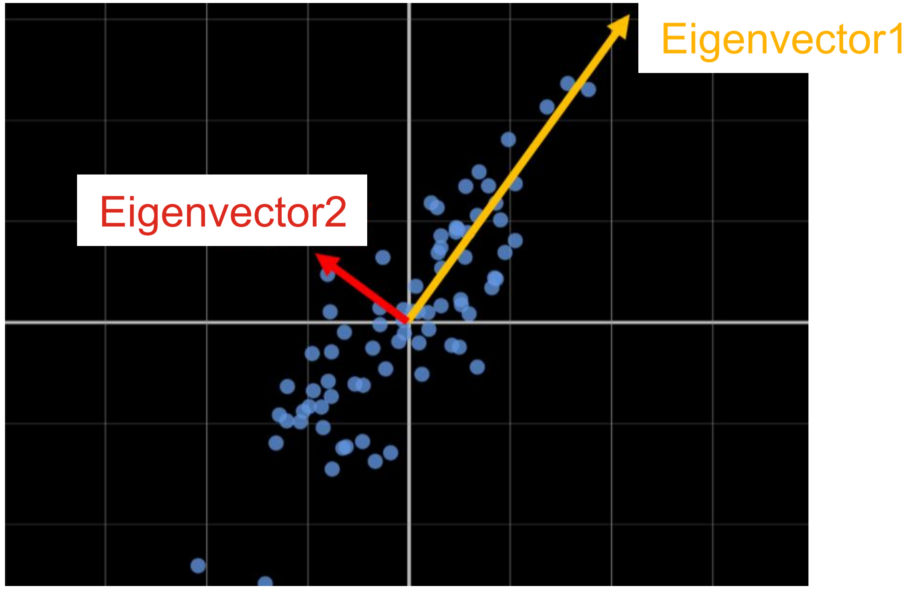
- **Eigenvector represents…,**
  - Direction of principal axes through which the matrix acts on vectors.
  - Eigenvectors of covariance matrix
    - Can be said to indicate **directions** in which the data is dispersed
- **Eigenvalue represents…,**
  - Extent to which the vector space is scaled in the direction of the eigenvectors.
  - Effectively determine **principal components** in order of **importance**.
    - By arranging eigenvectors in **descending** order of their eigenvalues.
- **Let's go to covariance matrix 1 again.**
  - On the next page.

# Find Eigenvectors and Eigenvalues of Covariance Matrix

- **Figure below, see two eigenvectors.**
  - magnitude of each vector signifies its corresponding **eigenvalue**.
- **What's the problem we were considering?**
  - Which vector gives the optimal result,
    - when taking the dot product(or orthogonal projection) with a data vector?
  - Solution to this problem
    - Possible by finding eigenvalues and eigenvectors of covariance matrix.

**Covariance matrix**

$$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$$




Eigenvectors of covariance matrix 1.

# Mathematical Meaning of Covariance Matrix

## Let's understand covariance matrix mathematically.

- Covariance matrix
  - Matrix expressing covariance between $\boxed{\text{two Variable}}$ values when there are more than two variables.

$$\operatorname{var}(\mathbf{X}) = \operatorname{cov}(\mathbf{X}, \mathbf{X}) = \mathrm{E}\left[(\mathbf{X} - \mathrm{E}[\mathbf{X}])(\mathbf{X} - \mathrm{E}[\mathbf{X}])^{\mathsf{T}}\right]$$

## For example,

- Let's say we extract $d$ feature from $n$ people.
- And express data as matrix $\boldsymbol{X}$ as Eq 1..
  - Assume that average value of each column (feature) of matrix as Eq 1. is $0$.
    - It will help you find vector that moves center of data distribution as pivot.

$$\boldsymbol{X} = \begin{pmatrix} | & | & | & | & | \\ \boldsymbol{X}_1 & \boldsymbol{X}_2 & \boldsymbol{X}_3 & \cdots & \boldsymbol{X}_d \\ | & | & | & | & | \end{pmatrix} \in \mathbb{R}^{n \times d}$$

Eq 1. Data matrix $\boldsymbol{X}$ obtained by stacking $d$ $n$-dimensional column vectors

# Consider Example of Creating Matrix

(640x480)

⋮

사진비교

- **Let's put height and weight into matrix $\boldsymbol{D}$ as Eq 1..**
  - From 5 people
- **Then, matrix $\boldsymbol{X}$ can be obtained as Eq 2..**
  - By subtracting average of each column from matrix $\boldsymbol{D}$.

$$\boldsymbol{D} = \begin{bmatrix} 170 & 70 \\ 150 & 45 \\ 160 & 55 \\ 180 & 60 \\ 170 & 80 \end{bmatrix}$$

Eq 1. Matrix $\boldsymbol{D}$

$$\boldsymbol{X} = \boldsymbol{D} - mean(\boldsymbol{D}) = \begin{bmatrix} 170 & 70 \\ 150 & 45 \\ 160 & 55 \\ 180 & 60 \\ 170 & 80 \end{bmatrix} - \begin{bmatrix} 166 & 62 \\ 166 & 62 \\ 166 & 62 \\ 166 & 62 \\ 166 & 62 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ -16 & -17 \\ -6 & -7 \\ 14 & -2 \\ 6 & 18 \end{bmatrix}$$

Eq 2. Calculate matrix $\boldsymbol{X}$

# Obtain Covariance Matrix Using Data Matrix $X$

- **Here, $dot(\bullet, \bullet)$ means** inner product **operation of two vectors.**
- **Meaning of $X^T X$**
  - Looking at last term in Eq 1. $(X^T X)_{ij}$ means…,
    - By obtaining values from all people and performing inner product operation to determine how similar $ith$ feature and $jth$ feature among d features.

$$X^T X = \begin{pmatrix} - & X_1 & - \\ - & X_2 & - \\ & \cdots & \\ - & X_d & - \end{pmatrix} \begin{pmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{pmatrix}$$

$$= \begin{pmatrix} dot(X_1, X_1) & dot(X_1, X_2) & \cdots & dot(X_1, X_1) \\ dot(X_2, X_1) & dot(X_2, X_2) & \cdots & dot(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ dot(X_d, X_1) & dot(X_d, X_2) & \cdots & dot(X_d, X_d) \end{pmatrix}$$

Eq 1. Process of calculating how similar variation of each data feature is to each other

# Calculate $X^T X$

## Calculate $X^T X$ matrix

- Result is as shown in Eq 1..

## Problem of $X^T X$ matrix

- As number $n$ increases, inner product value continues to $\boxed{\text{increase}}$
- In other words, more samples you collect, $\boxed{\text{larger}}$ result.

$$X^T X = \begin{bmatrix} 4 & -16 & -6 & 14 & 6 \\ 8 & -17 & -7 & -2 & 18 \end{bmatrix} \begin{bmatrix} 4 & 8 \\ -16 & -17 \\ -6 & -7 \\ 14 & -2 \\ 6 & 18 \end{bmatrix} = \begin{bmatrix} 540 & 426 \\ 426 & 730 \end{bmatrix}$$

Eq 1. Result of $X^T X$

# How to Prevent Problem of $X^T X$

- **Divide dot product by $\boxed{n}$ as Eq 1..**
- **Covariance matrix**
  - Matrix represented in Eq 1..
- **For data matrix $X$,**
  - Covariance matrix $\Sigma$ is Eq 2..
- **Calculating covariance matrix from example data as Eq 3..**

$$\frac{X^T X}{n} = \frac{1}{n}\begin{pmatrix} dot(X_1, X_1) & dot(X_1, X_2) & \cdots & dot(X_1, X_1) \\ dot(X_2, X_1) & dot(X_2, X_2) & \cdots & dot(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ dot(X_d, X_1) & dot(X_d, X_2) & \cdots & dot(X_d, X_d) \end{pmatrix}$$

Eq 1. Covariance matrix

$$\Sigma = \frac{1}{n} X^T X$$

Eq 2. Covariance matrix for data matrix $X$

$$\Sigma = \frac{1}{5} X^T X = \frac{1}{5}\begin{bmatrix} 540 & 426 \\ 426 & 730 \end{bmatrix} = \begin{bmatrix} 108 & 85.2 \\ 85.2 & 146 \end{bmatrix}$$

Eq 2. Covariance matrix for data matrix $X$

# Eigenvectors and Maximum Variance of Covariance Matrix

## In this part…,

- Explain why variance of data obtained by orthogonally projecting data to the eigenvector is maximized.

## For example,

- Let's say that $d$-dimensional data is reduced to $1$-dimension through orthogonal projection.
  - Consider **arbitrary unit vector** $\vec{\boldsymbol{e}}$ that is subject to orthogonal projection.
    - $\vec{\boldsymbol{e}}$ is $d \times 1$-dimensional vector.
- If data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times \boldsymbol{d}}$ is orthogonally projected onto unit vector $\vec{\boldsymbol{e}}$,
  - $X\vec{e}$, its size is as Eq 1..
- Therefore, variance of data orthogonally projected to $\vec{\boldsymbol{e}}$ is as Eq 2..

$$\boldsymbol{X\vec{e}} \in \mathbb{R}^{\boldsymbol{n \times 1}}$$

Eq 1. Size of $X\vec{e}$

$$\boldsymbol{Var(X\vec{e}) = \frac{1}{n}\sum_{i=1}^{n}(X\vec{e} - E(X\vec{e}))^2}$$

Eq 2. Variance of data orthogonally projected to $\vec{\boldsymbol{e}}$

# Variance of Orthographic Data

- **In Eq 1.,**
  - Assuming that average of each column of $\boldsymbol{X}$ is 0,
    - Can be expressed as Eq 2..
- **Therefore,**
  - You can obtain result as Eq 3..

$$Var(X\vec{e}) = \frac{1}{n}\sum_{i=1}^{n}(X\vec{e} - E(X\vec{e}))^2$$

Eq 1. Variance of data orthogonally projected to $\vec{e}$

$$Var(X\vec{e}) = \frac{1}{n}\sum_{i=1}^{n}(X\vec{e} - E(X\vec{e}))^2 = \frac{1}{n}\sum_{i=1}^{n}(X\vec{e} - E(\underbrace{\cancel{X}}_{0})\vec{e})^2 = \frac{1}{n}\sum_{i=1}^{n}(X\vec{e})^2$$

Eq 2. Variance of data orthogonally projected to $\vec{e}$ when average of each column of $\boldsymbol{X}$ is 0

$$Var(X\vec{e}) = \frac{1}{n}(X\vec{e})^T(X\vec{e})$$
$$= \frac{1}{n}\vec{e}^T X^T X\vec{e} = \frac{1}{n}\vec{e}^T(X^T X)\vec{e}$$
$$= \vec{e}^T\left(\frac{X^T X}{n}\right)\vec{e}$$
$$= \boxed{\vec{e}^T\Sigma\vec{e}}$$

Eq 3. Variance of data orthogonally projected to $\vec{e}$

# Let's Find Out How to Choose $\vec{e}$

- **Choose $\vec{e}$ to maximize displacement of orthogonally projected data.**
- **Use Lagrange multiplier method.**
  - Objective function: $\vec{e}^T \Sigma \vec{e}$
  - Constraints: $|\vec{e}|^2 = 1$
- **Therefore, auxiliary equation as Eq 1. can be created.**
- **Partial differentiation of Eq 1. with respect to $\vec{e}$ is equivalent to Eq 2..**
  - When choose $\vec{e}$ satisfying condition of $\Sigma\vec{e} = \lambda\vec{e}$,
    - Objective function $\vec{e}^T \Sigma \vec{e}$ can be maximum.

$$L = \vec{e}^T \Sigma \vec{e} - \lambda(|\vec{e}|^2 - 1)$$

Eq 1. Auxiliary equation $L$

$$\frac{\partial L}{\partial \vec{e}} = 2\Sigma\vec{e} - 2\lambda\vec{e} = 0$$

Eq 2. Partial differentiation of Auxiliary equation $L$ with respect to $\vec{e}$

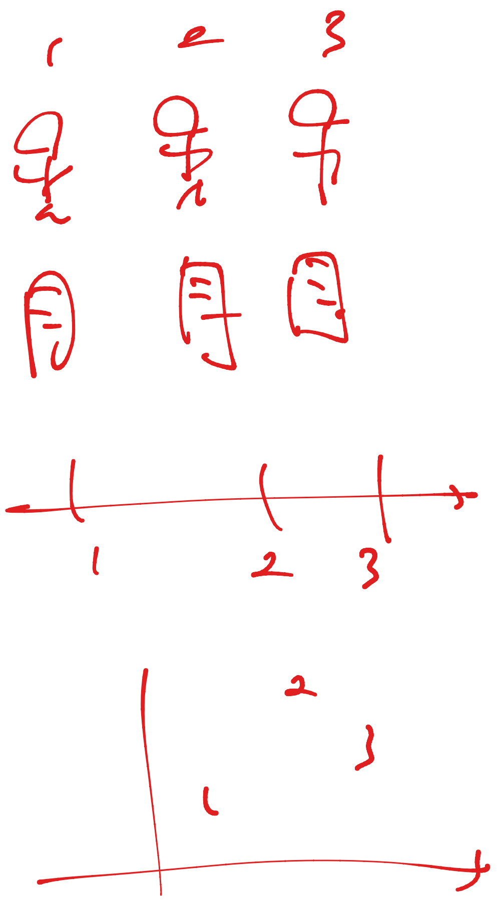
# Eigenvalue

- **Through result obtained by Lagrange multiplier method,**
  - You can get $\boldsymbol{\Sigma\vec{e}} = \boldsymbol{\lambda\vec{e}}$.
  - Maximize displacement $\boldsymbol{\vec{e}}$ is eigenvector.

- **Therefore, when orthogonally projected through eigenvector,**
  - **Variance** is **eigenvalue**.

$$\boldsymbol{Var(X\vec{e}) = \vec{e}^T\Sigma\vec{e} = \vec{e}^T\lambda\vec{e} = \lambda\vec{e}^T\vec{e} = \lambda}$$

Eq 1. Variance of data




매칭!

# How Many Dimensions?

- **Main purpose of PCA**
  - **Reduce dimensionality** in multidimensional data.
- **But to how extent is it appropriate to reduce dimensionality of high-dimensional data?**
- **For example,**
  - Let's reduce $d$-dimensional data to $m$-dimension.
    - Of course, $m < d$
  - Since it is $d$-dimensional data, total of $d$ eigenvalues can be calculated.
    - Of course, this is assuming that covariance matrix of data is full rank.
    - Let's express it as $\lambda_1, \lambda_2, \cdots, \lambda_d$.
      - $\lambda_1 \geq \lambda_2 \geq \cdots, \geq \lambda_d$
  - One logical method
    - Reduce variance of entire data to level that explains as much as $90\%$.
      - Find appropriate $m$ to reduce it to that dimension.

분포를 가장 넓게 유지해주는 값 찾기.

1~3차원

$$\frac{\sum_{j=1}^{m} \lambda_j}{\sum_{i=1}^{d} \lambda_i} = 0.9$$

1~11차원

Reducing variance of entire data to level that explains as much as 90%

# Another Logical Method

- **Using scree plot.**
  - Draw 2-dimensional plot as Figure 1..
    - $x$-axis: Dimensions
    - $y$-axis: Eigenvalue of that dimension

- **See Figure 1. to interpret Scree plot.**
  - You can see sudden bend starting from third eigenvalue.
  - Then, determine that dimensionality will be reduced to 3 dimensions.

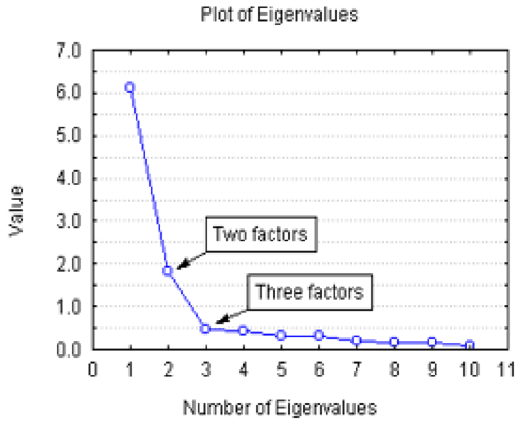



Figure 1. Scree plot

# Jacobian Matrix

# Definition of Jacobian Matrix

- **Let's assume there is vector function that produces as output $\boldsymbol{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$**
  - Input: Vector $\boldsymbol{x} \in \mathbb{R}^n$
  - Output: Vector $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^m$
- **If first partial derivative of this function exists in the real vector space of $\mathbb{R}^n$,**
  - Jacobian can be defined as an $\boldsymbol{m} \times \boldsymbol{n}$ matrix as Eq 1..
- **Things you can notice when looking at Eq 1.**
  - Elements of Jacobian matrix are all composed of first-order differential coefficients
  - Jacobian matrix is a linear transformation regarding small changes.
- **In fact, What Jacobian is trying to say**
  - 'Nonlinear transformation' is approximated to **linear transformation in microscopic domain**.

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{x}_n} \end{bmatrix}$$

Eq 1. Jacobian matrix

# Chain Rule for Multivariate Functions

- **Before understanding Jacobian matrix,**
  - Let's briefly discuss most essential part, **chain rule**.
- **In general,**
  - Multivariate function can be considered function that has two or more inputs.
- **In examples,**
  - We will limit ourselves to two-variable functions to learn about chain rule.
    - Because we will use functions that can be displayed on two-dimensional plane.
- **In multivariate function $\boldsymbol{z = f(x,\ y), x = g(t), y = h(t)}$,**
  - If $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{g}(t), \boldsymbol{h}(t)$ are all differential functions, Eq 1. is true.

$$\frac{dz}{dt} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial t}$$

$$dz = \frac{\partial z}{\partial x}\partial x + \frac{\partial z}{\partial y}\partial y$$

Eq 1. Chain rule for multivariate function

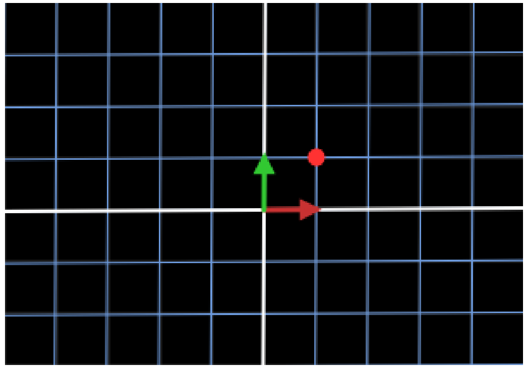
# Linear Transformation

## Geometric characteristics

- $\boxed{\qquad}$ of origin does not change even after conversion.
- Even after conversion, shape of grids maintains **straight line**.
- Spacing between grids must be even.

## Look at shape of grid before and after transformation of shear matrix as Figure 1..

- You can see that it satisfies geometric characteristics of linear transformation.



$$\xrightarrow{\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}$$

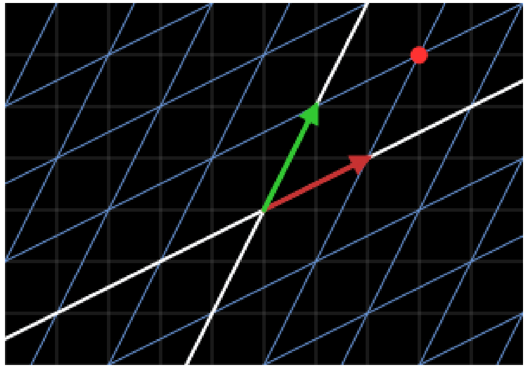


Figure 1. Linear transformation: Shear matrix

# Linear Transformation



## Code Exercise (15_01)

- Linear transform with shear matrix

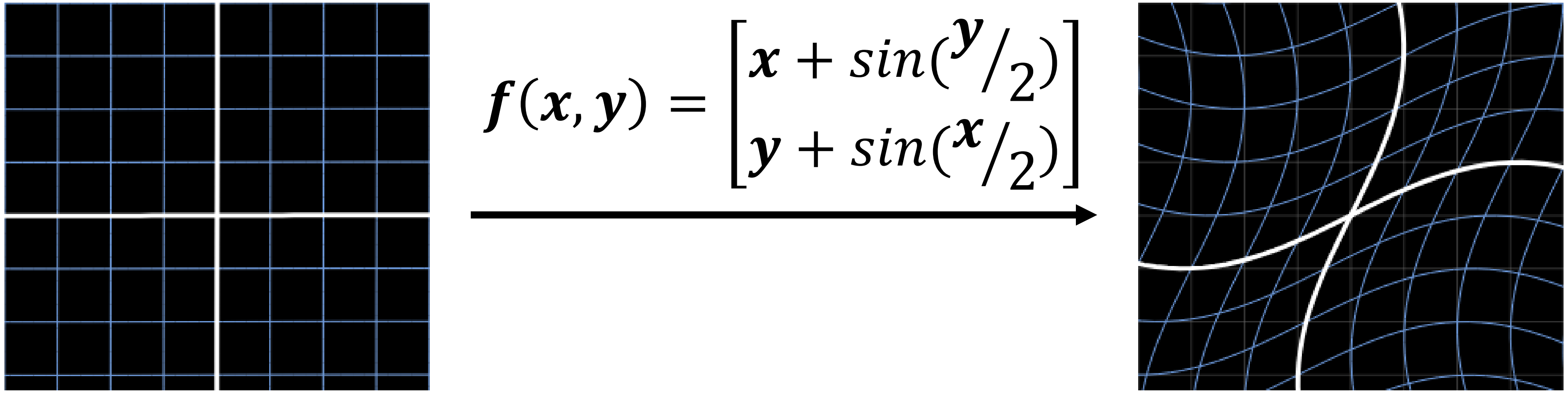
```
% Clear workspace, command window, and close all figures
clear; close all; clc;

[X,Y]=ndgrid(-6:1:6);

% Define shear matrix
A = [2,1;1,2];

n_steps = 20;
figure;
set(gcf,'color','w');
set(gca, 'nextplot','replacechildren');

% Simulate the lineaer transform of shear matrix
for i_steps = 0:n_steps
    step_mtx = (A-eye(2))/n_steps*i_steps;

    % Calculate the linear transformed X and Y
    new_xy = (eye(2)+step_mtx)*[X(:), Y(:)]';
    new_XY = reshape(new_xy,[2,size(X,1),size(X,1)]);

    for i = -3:3
        for j=-4:0
            line([i i],[-j j], 'color','k');
            hold on;

            line([-j j],[i i], 'color', 'k');
        end
    end
    plot(squeeze(new_XY(1,:,:)), squeeze(new_XY(2,:,:)), '-','color','r');
    plot(squeeze(new_XY(1,:,:))', squeeze(new_XY(2,:,:))', '-', 'color','r');

    axis equal
    xlim([-4,4])
    ylim([-4,4])
    axis off

    drawnow;
    if i_steps<n_steps
        cla
    end
end
```

MATLAB code of linear transform using shear matrix

# Nonlinear Transformation

- **Nonlinear transformation is transformations.**
  - ▶ Not linear transformation.

- **Look at example of nonlinear transformation as Figure 1..**
  - ▶ You can see that it does not satisfy geometric characteristics of linear transformation.



$$f(x,y) = \begin{bmatrix} x + sin(^{y}/_{2}) \\ y + sin(^{x}/_{2}) \end{bmatrix}$$

Figure 1. Nonlinear transformation

# Nonlinear Transformation

## Code Exercise (15_02)

- Check how the coordinate is changing.
- Why this change is nonlinear?
- Use the given function file ‘my_nonlin_func.m’.

```
% Clear workspace, command window, and close all figures
clear; close all; clc;

n_steps = 20;
function_num = 'basic'; % 'polar':극좌표계
range = 11;

for i_step = 0:n_steps
    % Original coordinate
    for i = -3:3
        for j=-4:0
            line([i i],[-j j], 'color','k');
            hold on;

            line([-j j],[i i], 'color', 'k');
        end
    end

    T = linspace(-range,range,100);
    for i_t = -range:range
        % Vertical axle
        X = i_t*ones(1,100);
        [tempx, tempy, VerticalX, VerticalY] = my_nonlin_func(X,T,function_num);
        newVerticalX = X+VerticalX*i_step/n_steps;
        newVerticalY = T+VerticalY*i_step/n_steps;

        % Horizontal axle
        Y = i_t*ones(1,100);
        [tempx2, tempy2, HorizontalX, HorizontalY] = my_nonlin_func(T,Y,function_num);
        newHorizontalX = T+HorizontalX*i_step/n_steps;
        newHorizontalY = Y+HorizontalY*i_step/n_steps;

        % Plot
        plot(newVerticalX,newVerticalY,'r');
        plot(newHorizontalX,newHorizontalY,'r');
        hold on;
    end

    grid on;
    xlim([-4 4])
    ylim([-4 4])
    hold off;
    drawnow;
    if i_step < n_steps
        cla
    end
end
```

MATLAB code of nonlinear transformation

# Nonlinear Transformation in Microscopic Domain

- **When looking at definition of Jacobian earlier,**
  - It was mentioned that what Jacobian is trying to say
    - 'Nonlinear transformation' is approximated to linear transformation in **microscopic domain**.
- **So, if nonlinear transformation is really viewed in microscopic domain,**
  - Can it be sufficiently approximated by linear transformation?
- **Check out visual example as Figure 1..**
  - Even after transformation,
    - Shape of grids is **close to straight line**.
    - Spacing between grids is **maintained evenly**.

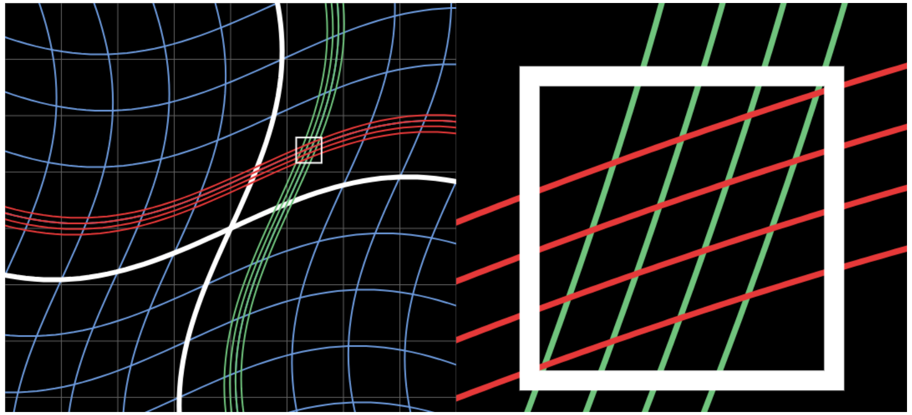


Figure 1. Nonlinear transformation in microscopic domain

# How to Resolve Fact That Position of Origin Does Not Change

- **In Figure 1.,**
  - Consider point at $(x_0, y_0)$ you want to transform as **origin**.
  - Obtain matrix you want to approximate.
  - Then, obtain Jacobian matrix.
    - Matrix that approximates [ ] transformation with [ ] transformation.

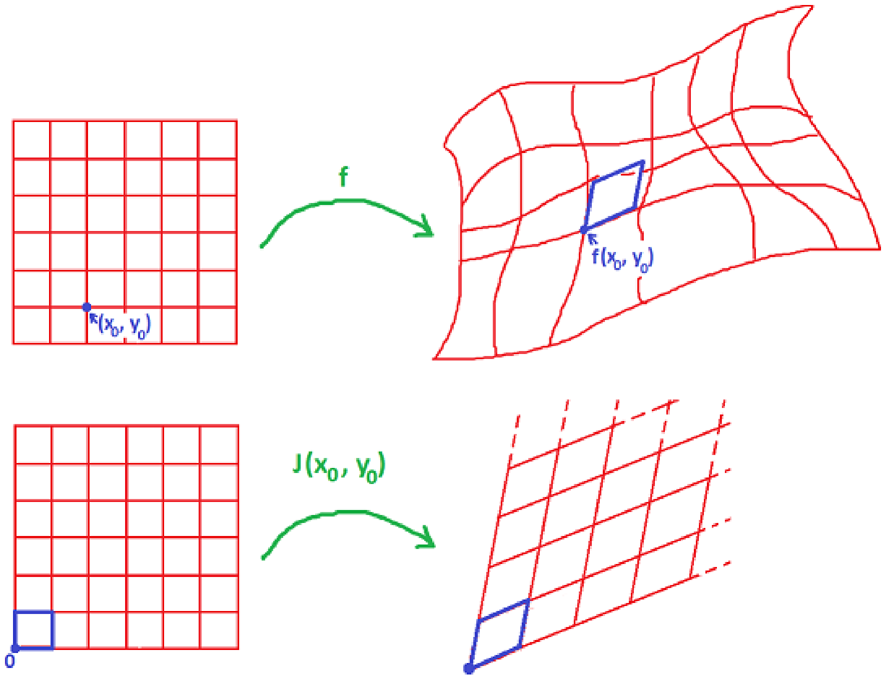



Figure 1. Jacobian matrix

# Derivation of Jacobian Matrix

- **As shown in Figure 1.,**
  - When result of nonlinear transformation is approximated to be similar to linear transformation,
    - Let's assume that it can be seen as case of changing from $(u, v)$ coordinate system to $(x, y)$ coordinate system.

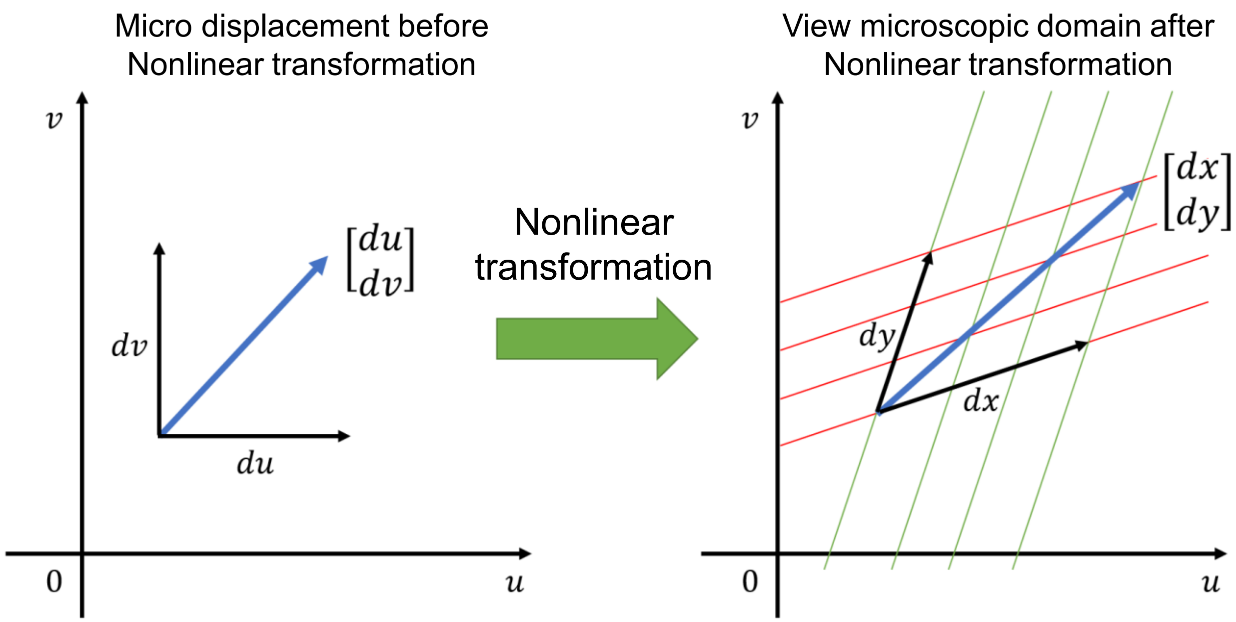



Figure 1. Jacobian matrix

# Chain Rule for Jacobian Matrix

- **Then, it can be seen that $du$ and $dv$ are converted to $dx$ and $dy$ by some linear transformation $J$.**
  - Expressed in Eq 1. and Eq 2..
- **Through Eq 2. and Eq 3.,**
  - Relations before and after **local** nonlinear transformation can be obtained.
  - Then, Jacobian matrix can be thought of as Eq 4..

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = J \begin{bmatrix} du \\ dv \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix}$$

Eq 1. Jacobian matrix

$$dz = \frac{\partial z}{\partial x} \partial x + \frac{\partial z}{\partial y} \partial y$$

Eq 3. Chain rule for multivariate function

$$dx = a \times du + b \times dv$$
$$dy = c \times du + d \times dv$$

Eq 2. Jacobian matrix in expanded form

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix}$$

Eq 4. Jacobian matrix from chain rule

# Geometric Meaning of Determinant

- **In linear transformation,**
  - Determinant indicates how much unit area increases.
- **As can be seen in Figure 1.,**
  - Before linear transformation,
    - Area of rectangle was 1,
  - After linear transformation,
    - Area of rectangle is transformed into $\boxed{\phantom{xxxxx}}$
      - Area equal to determinant value.

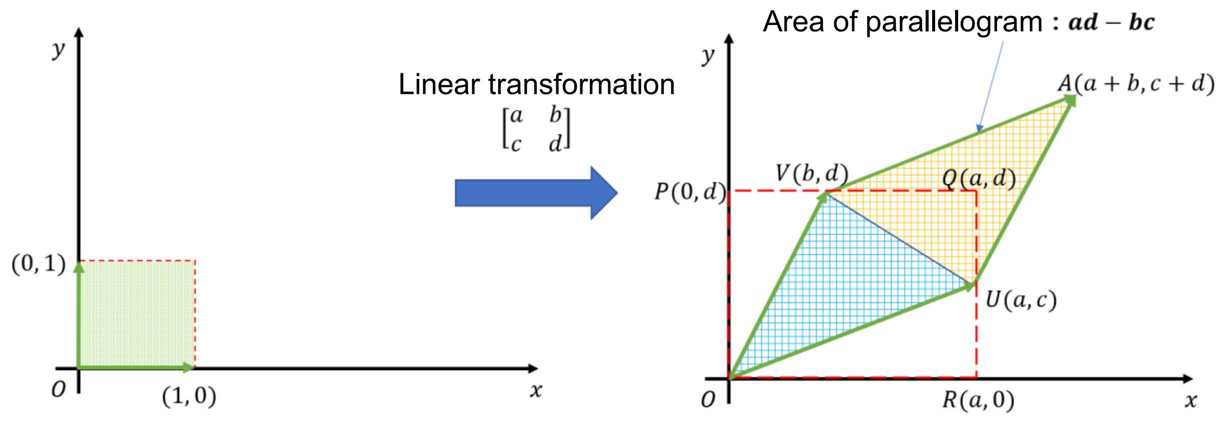



Figure 1. Geometric meaning of determinant

# Meaning of Determinant of Jacobian Matrix



- **Rate of change in area**
  - When transformed from original coordinate system to transformed coordinate system.
- **In Figure 1.,**
  - When transformed from $(u, v)$ coordinate system to $(x, y)$ coordinate system,
    - Relation between $dx \times dy$ and $du \times dv$ is as Eq 1..
      - $|\boldsymbol{J}|$: Determinant of Jacobian matrix
      - $\times$: Simple multiplication symbol, not cross product symbol

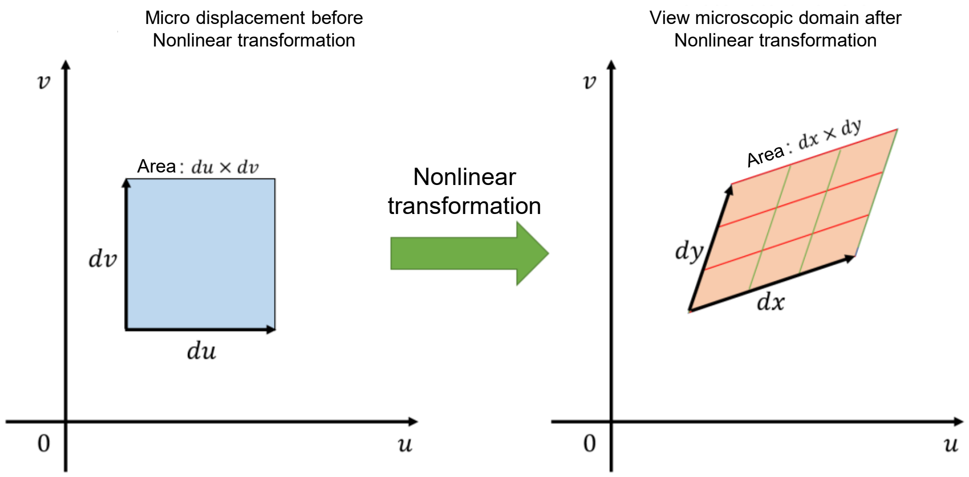



Figure 1. Geometric meaning of determinant

$$\boldsymbol{dx \times dy = |J|(du \times dv)}$$

Eq 1. Transformation from $(u, v)$ coordinate system to $(x, y)$ coordinate system

# Perform Example of Finding Area of Circle Using Jacobian

- **There are several way to find area of circle.**
  - One way is to use method to find $\boxed{\phantom{xxxxxxxx}}$ in $(r, \theta)$ coordinate system.
- **What does it mean to find area of circle in $(\boldsymbol{r}, \boldsymbol{\theta})$ coordinate system using Jacobian?**
  - Calculate area in coordinate system.
    - Horizontal axis is $r$ and vertical axis is $\theta$.
  - Then, transform to coordinate system.
    - Horizontal axis is $x$ and vertical axis is $y$.
  - Here, role of Jacobian determinant
    - **Correction value** for area required when transforming between coordinate system.

# Visual Example of Finding Area of Circle Using Jacobian

- **Let's observe Figure 1. and Figure 2..**
  - In first scene,
    - You can see points in $(r, \theta)$ coordinate system.
  - In last scene,
    - You can see that these points are moved to $(x, y)$ coordinate system.
    - In Figure 1.,
      - Area of circle appears sparse.
    - In Figure 2.,
      - Area of circle appears to be somewhat full.
- **At this time,**
  - Area correction value sufficient to properly fill space is **determinant value of Jacobian**.

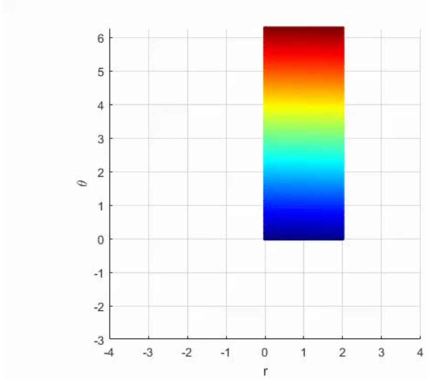



Figure 1. Polar to $(x, y)$ coordinate system without determinant value of Jacobian

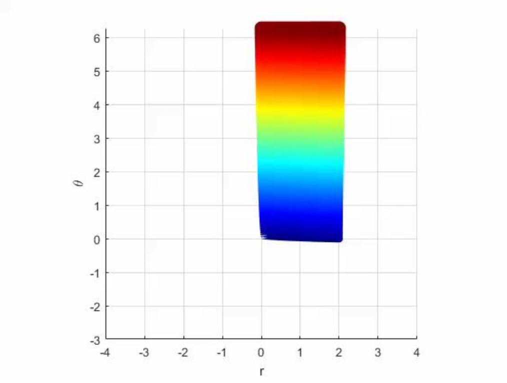



Figure 2. Polar to $(x, y)$ coordinate system with determinant value of Jacobian

# Visual Example of Finding Area of Circle Using Jacobian



## Code Exercise (15_03)

- Use the given function file ‘my_nonlin_func.m’.
- Watch the figure and find the difference between using jacobian or not.

```
% Clear workspace, command window, and close all figures
clear; close all; clc;

% Applying jacobian(1) or not(0)
applying_jacobian = 1;

% Density of points
n_points = 90;

n_steps = 100; size = 10;
my_colors = jet(n_points^2);
r = 2; % radius

% Polar coordinate
[R, THETA] = ndgrid(linspace(0, r, n_points), linspace(0, 2*pi, n_points));

% Nonlinear function using polar coordinate
[newX, newY, changeR, changeTHETA] = my_nonlin_func(R(:), THETA(:), 'polar');
figure;
set(gcf, 'Color', 'w');

for i_step = 0:n_steps
    % Calculate the changing points
    new_R = R(:) + changeR * i_step / n_steps;
    new_THETA = THETA(:) + changeTHETA * i_step / n_steps;

    if applying_jacobian == 0
        scatter(new_R, new_THETA, size, my_colors, 'filled');
    else
        dist = sqrt(abs(new_R.^2 + new_THETA.^2));
        scatter(new_R, new_THETA, dist * size * 2 + 0.01, my_colors, 'filled');
    end

    xlabel('r'); ylabel('\theta');
    if i_step == n_steps
        xlabel('x = r cos\theta'); ylabel('y = r sin\theta');
    end

    grid on;
    xlim([-4, 4]);
    ylim([-r * 1.5, 2*pi]);
    axis square;
    hold off;
    drawnow;
end
```

MATLAB code of finding area of circle using jacobian matrix

# Transformation Equation to $(x, y)$ Coordinate System

- Equation from $(r, \theta)$ coordinate system to $(x, y)$ coordinate system as Eq 1.
- Calculating Jacobian matrix from Eq 1. is Eq 2..
- Calculate determinant value of Jacobian as Eq 3..

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} rcos(\theta) \\ rsin(\theta) \end{bmatrix}$$

Eq 1. Transformation from $(r, \theta)$ coordinate system to $(x, y)$ coordinate system

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -r\sin(\theta) \\ \sin(\theta) & r\cos(\theta) \end{bmatrix}$$

Eq 2. Calculate Jacobian matrix from Eq 1.

$$|J| = rcos^2\theta + rsin^2\theta = r$$

Eq 3. Calculate determinant value of Jacobian

# Finding Area of Circle Using Determinant Value of Jacobian

- **Calculate area of circle with radius $3$ in $(r, \theta)$ coordinate system.**
  - ▶ Using determinant value of Jacobian.

$$\int_{r=0}^{r=3} \int_{\theta=0}^{\theta=2\pi} dxdy$$

$$\int_{r=0}^{r=3} \int_{\theta=0}^{\theta=2\pi} |J| dr d\theta$$

$$\int_{r=0}^{r=3} \int_{\theta=0}^{\theta=2\pi} r dr d\theta$$

$$\int_{r=0}^{r=3} r\,\theta \Big|_{\theta=0}^{\theta=2\pi} dr$$

$$= 2\pi \frac{1}{2} r^2 \Big|_{r=0}^{r=3}$$

$$= 2\pi \cdot \frac{1}{2} 3^2 = 3^2\pi$$

Calculate area of circle with radius 3 in $(r, \theta)$ coordinate system using determinant value of Jacobian

미분 2번 한거.

# Hessian Matrix

# Definition of Hessian Matrix



## Form of Hessian matrix

- Constructed using the **second-order partial derivatives** of a function.
  - All elements of Hessian matrix are ______________________.
- If second-order partial derivatives are continuous…,
  - Mixed partial derivatives are equal.
    - $\frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1}$
  - Hessian matrix is a symmetric matrix.

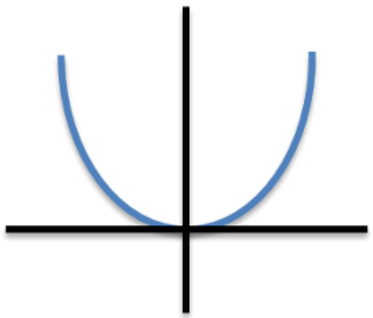
## Then what is the meaning of the second-order ?

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

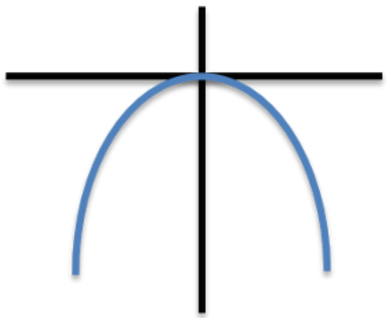
Form of Hessian matrix

# Meaning of Second-Order Derivative

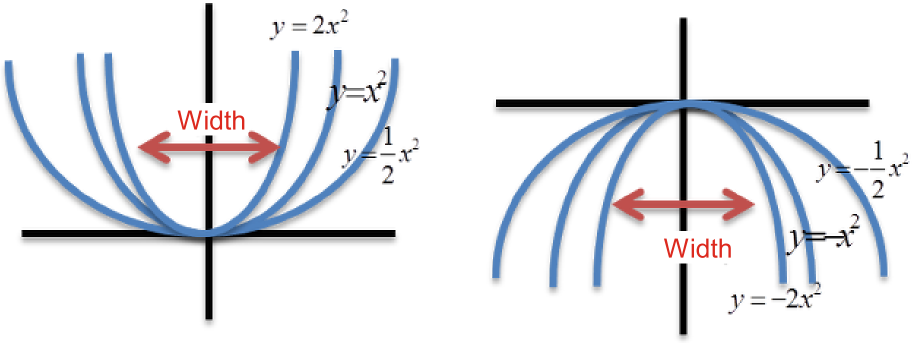
- **Consider $\boldsymbol{f(x) = \frac{1}{2}ax^2 + bx + c}$**
  - Second-order derivative of $f(x)$ : $\boldsymbol{a}$
  - Positive $a$: function has a **convex** shape.
  - Negative $a$: function has a **convex** shape.
  - If larger value of $|a|$ :
    - The shape becomes more convex.



$a > 0$



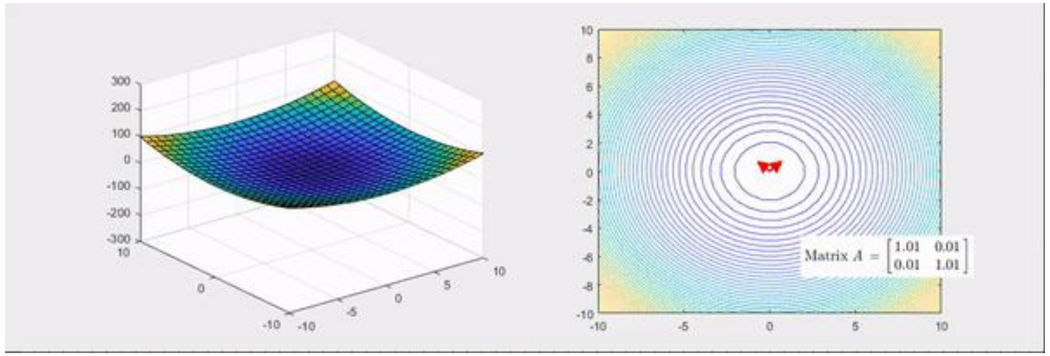
$a < 0$



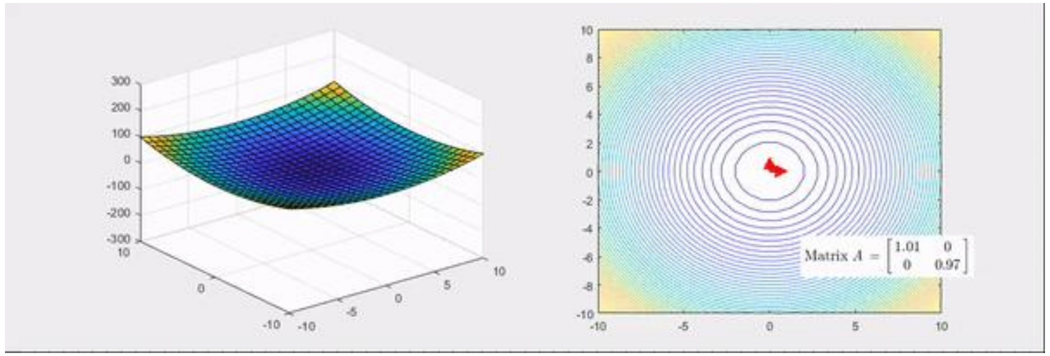

Difference in the graph's shape depending on magnitude of $a$

# Geometric Meaning of Hessian Matrix

- **Every matrix can be considered as a linear transformation.**
  - Linear transformation is a type of **spatial transformation**.
    - When thought of geometrically.
- **Geometrical meaning of linear transformation performed by Hessian matrix**
  - Makes a basic bowl-shaped function **more convex** or **concave**.
- **What analysis is needed to understand it ?**
  - Key geometric features of transformation shown by Hessian matrix.
  - Main axis of linear transformation must be identified and its size quantified.
    - Possible by identifying eigenvalues and eigenvectors of Hessian matrix.




Make more convex




Make bowl form to saddle form

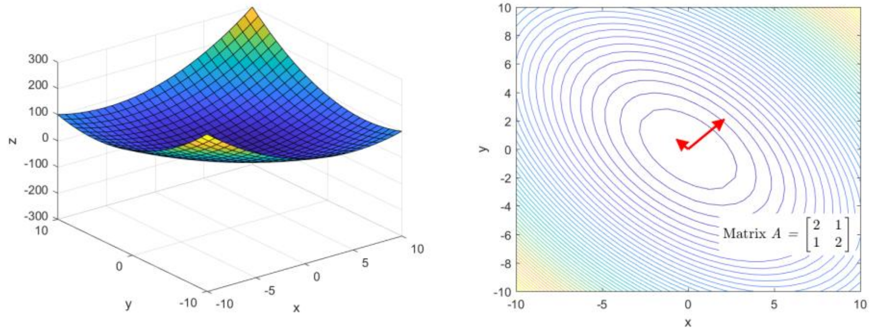
# Meaning of Eigenvalues and Eigenvectors of Hessian Matrix

## After linear transformation,

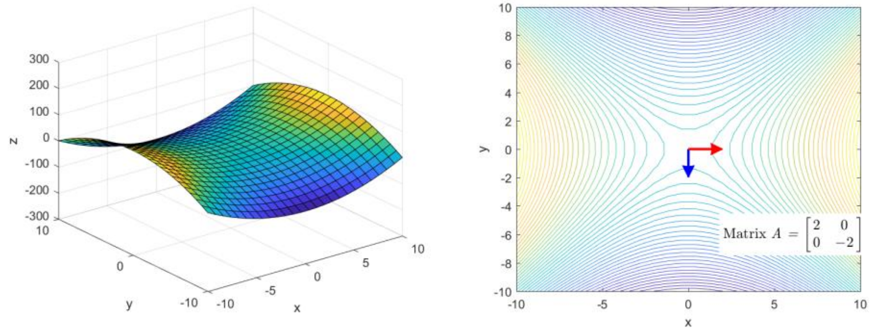
- Eigenvectors: do not change $\boxed{\quad\quad}$ but may change in $\boxed{\quad\quad}$.
- Eigenvalues: extent to which a vector has changed.
- Below figures are last scenes from previous page's figures.
  - Direction of arrow: eigenvector.
  - Length of arrow: eigenvalue.
  - Red arrow: positive eigenvalue.
  - Blue arrow: negative eigenvalue.

## By using Hessian matrix…,

- Possible to understand that **Determination of second-order derivative** can be conducted.
- More details on next page.




Make more convex




Make bowl form to saddle form

# Details of Meaning of Eigenvalues and Eigenvectors of Hessian Matrix

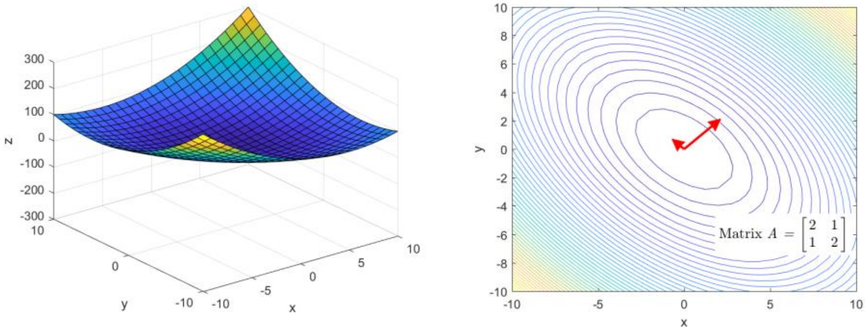
## Using the Hessian matrix.

- Possible to determine whether a specific point on a function is,
  - **convex up**, **convex down**, or **saddle point**.

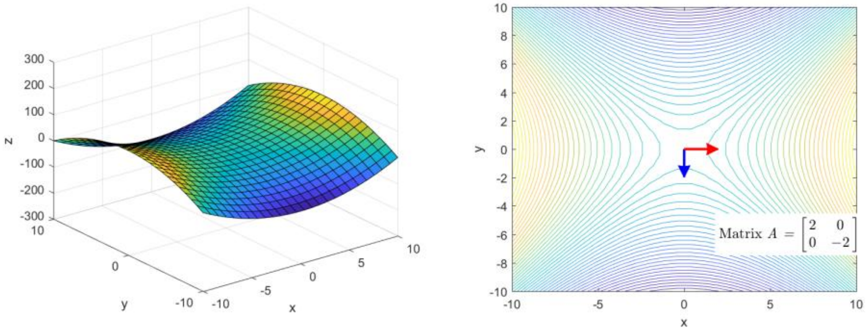
positive definite

| | Eigenvalues | | |
|---|---|---|---|
| | All possible | All negative | Mix of positive and negative |
| Function | **Convex down** | **Convex up** | **Shape of saddle** |
| At critical point | **Minimum point** | **Maximum point** | **Saddle point** |

Table of the characteristics of the Hessian matrix




Make more convex




Make bowl form to saddle form

# Eigenvalues and Eigenvectors of Hessian Matrix



## Code Exercise (15_04)

- Compare the result of code with the convex form of hessian matrix in previous slide.

```
% Clear workspace, command window, and close all figures
clear; close all; clc;

A = [2 1; 1 2]; % Hessian matrix form of convex
b = [0 0]';
c = 0;

figure('position', [230,100,1153,387]);
[X,Y] = meshgrid(-10:0.8:10);
fcn = @(x,y) (1/2 * A(1,1)*x.^2 + 1/2 * (A(1,2)+A(2,1))*x.*y + 1/2*A(2,2)*y.^2-b(1)*x - b(2)*y +c);

% Plot 3D
subplot(1,2,1);
surf(X, Y, fcn(X,Y))
xlim([-10,10])
ylim([-10,10])
zlim([-300,300])
xlabel('x')
ylabel('y')
zlabel('z')

subplot(1,2,2);
contour(X,Y,fcn(X,Y),50); hold on;

% Eigenvectors & Eigenvalues
[V,D] = eig(A);
quiver(0, 0, V(1,1)*D(1,1), V(2,1)*D(1,1), 'AutoScale', 'off', 'Color', 'r', 'LineWidth', 1, 'MaxHeadSize', 10);
quiver(0, 0, V(1,2)*D(2,2), V(2,2)*D(2,2), 'AutoScale', 'off', 'Color', 'r', 'LineWidth', 1, 'MaxHeadSize', 2);

str = ['Matrix {A} =', '$$ \left[ {\matrix{ ', num2str(A(1,1)),' & ', num2str(A(1,2)), ...
' \cr ', num2str(A(2,1)),' & ', num2str(A(2,2)),' } } \right] $$' ];
t = text(0.6, 0.2, str, 'unit' , 'normalized', 'Interpreter', 'latex', ...
'BackgroundColor', 'w', 'Fontsize', 12);
xlabel('x');
ylabel('y');
```

MATLAB code of eigenvalues and eigenvectors of Hessian matrix of convex form

# Eigenvalues and Eigenvectors of Hessian Matrix



## Code Exercise (15_05)

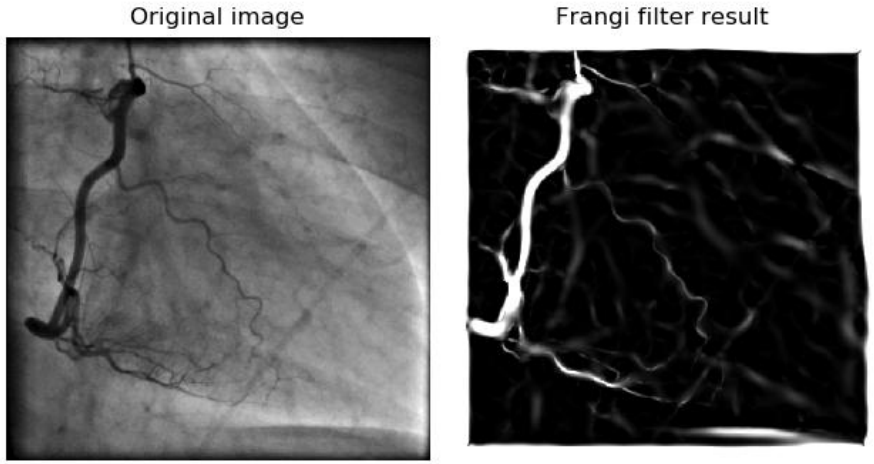
- Compare the result of code with the saddle form of hessian matrix in previous slide.

```
% Clear workspace, command window, and close all figures
clear; close all; clc;

A = [2 0; 0 -2]; % Hessian matrix of saddle form
b = [0 0]';
c = 0;

figure('position', [230,100,1153,387]);
[X,Y] = meshgrid(-10:0.8:10);
fcn = @(x,y) (1/2 * A(1,1)*x.^2 + 1/2 * (A(1,2)+A(2,1))*x.*y + 1/2*A(2,2)*y.^2-b(1)*x - b(2)*y +c);

% Plot 3D
subplot(1,2,1);
surf(X, Y, fcn(X,Y))
xlim([-10,10])
ylim([-10,10])
zlim([-300,300])
xlabel('x')
ylabel('y')
zlabel('z')

subplot(1,2,2);
contour(X,Y,fcn(X,Y),50); hold on;

% Eigenvectors & Eigenvalues
[V,D] = eig(A);
quiver(0, 0, V(1,1)*D(1,1), V(2,1)*D(1,1), 'AutoScale', 'off', 'Color', 'r', 'LineWidth', 1, 'MaxHeadSize', 10);
quiver(0, 0, V(1,2)*D(2,2), V(2,2)*D(2,2), 'AutoScale', 'off', 'Color', 'b', 'LineWidth', 1, 'MaxHeadSize', 2);

str = ['Matrix {A} =', '$$ \left[ {\matrix{ ', num2str(A(1,1)),' & ', num2str(A(1,2)), ...
' \cr ', num2str(A(2,1)),' & ', num2str(A(2,2)),' } } \right] $$' ];
t = text(0.6, 0.2, str, 'unit' , 'normalized', 'Interpreter', 'latex', ...
'BackgroundColor', 'w', 'Fontsize', 12);
xlabel('x');
ylabel('y');
```

MATLAB code of eigenvalues and eigenvectors of Hessian matrix of convex form

# Applications of Hessian Matrix

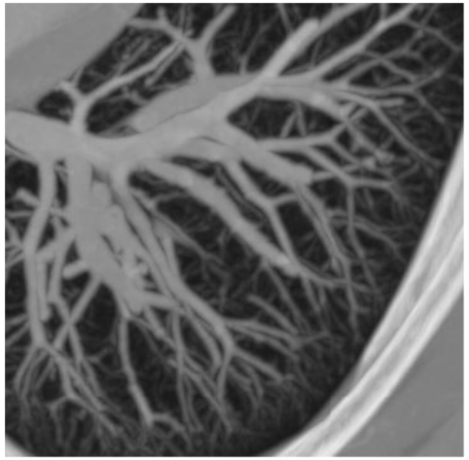
- **Usage of the Hessian matrix in various methods**
  - Convex optimization, second derivative test, newton method, **image processing**
- **Example of the Hessian matrix in image processing.**
  - In case of **vessel detection**
  - Figure below shows vessel detection.
    - Performed through image processing using a
    - is created using the Hessian matrix.



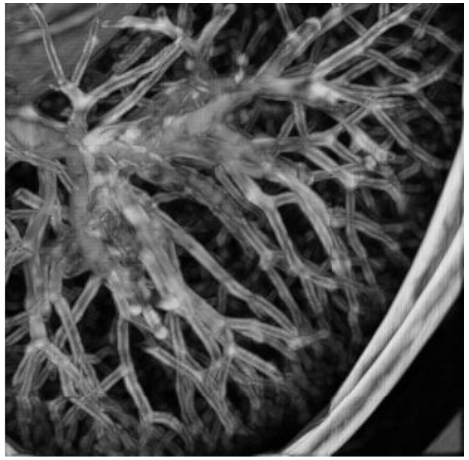

Result of vessel detection through Frangi filter

# Basic Idea of Vessel Detection Using Hessian Matrix

- **Hessian matrix**
  - Indicating **how much bowl shape** of a function has been deformed.
  - Eigenvectors of Hessian matrix.
    - Represent **principal axes of deformation**.
  - Eigenvalues of Hessian matrix.
    - Indicate **degree of deformation**.
- **Significant difference in the magnitude of eigenvalues of the Hessian at a certain point.**
  - It can be inferred that point will have an elongated shape.



Original image (shown in MIP mode)



Result after processing with Frangi filter (shown in MIP mode)

Result of vessel detection by 3D CT scan through Frangi filter

# Summary

# Summary

## PCA

- Objective
  - Finds set of weights such that linear weighted combination of data features has maximal variance.
  - Reflects assumption underlying PCA, which is that “variance equals relevance”.
- Implementation
  - Eigendecomposition of data covariance matrix
  - Eigenvector
    - Feature weightings
    - Eigenvalues can be scaled to encode percent variance accounted for by each component.

## Jacobian Matrix

- ‘Nonlinear transformation’ is approximated to linear transformation in microscopic domain.
- Determinant
  - Rate of change in area
    - When transformed from original coordinate system to transformed coordinate system.

## Hessian Matrix

- Constructed using the second-order partial derivative of a function.
- Makes basic bowl-shaped function more convex or concave or shape of saddle.
  - Hessian matrix’s function is determined by its eigenvectors and eigenvalues.

# Code Exercises

# Data Validation for PCA

- **This exercise needs Statistics and Machine Learning Toolbox**
- **Import and inspect the data.**
- **Made several plots of the data shown in Figure 1.**
- **Load stock data from download link**
  - stock dataset
    - dates(1 column)
    - market return(2~10 columns)
- **Plot data, correlation, covariance**

# Data Validation for PCA

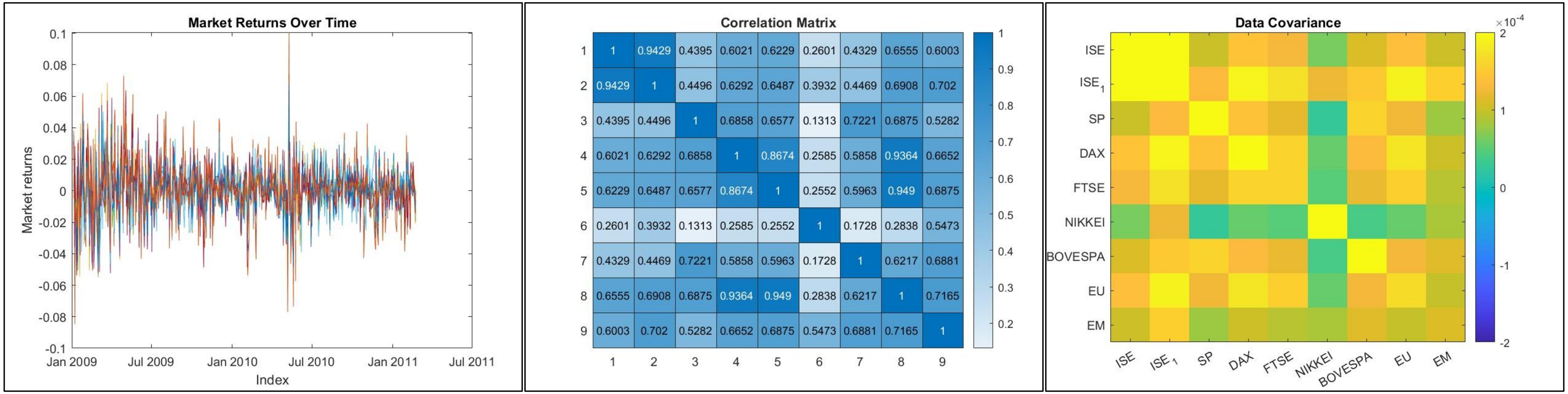


```
% Data citation: Akbilgic, Oguz. (2013). ISTANBUL STOCK EXCHANGE. UCI
Machine Learning Repository.
% data source website: https://archive-
beta.ics.uci.edu/ml/datasets/istanbul+stock+exchange

% hint
% corr()
% cov()
% diag()
% eig()
% evecs()
% sum()
% sort()
% mean()

% plot()
% heatmap()

% import the data
url = 'https://archive.ics.uci.edu/ml/machine-learning-
databases/00247/data_akbilgic.xlsx';
raw_data = readtable(url, 'Sheet', 1, 'Range', 'A2');

dates = ;
data = ;
location = raw_data.Properties.VariableNames(2:end);

% show the correlation matrix in an image
figure;
corrMatrix = ; % code here!
```

```
% calculate covariance matrix (zero mean)
X = data;
X =;
covmat = ;
% show some data in line plots
figure;
% code here!
% ;
xlabel('Index');
ylabel('Market returns');
title('Market Returns Over Time');
saveas(gcf, 'Figure_14_01a.png');

% code here!
% ;
title('Correlation Matrix');
saveas(gcf, 'Figure_14_01b.png');

% visualize it
figure;
imagesc(covmat);
colorbar;
title('Data Covariance');
xticks(1:size(X, 2));
xticklabels(location);
yticks(1:size(X, 2));
yticklabels(location);
caxis([-.0002 .0002]);
saveas(gcf, 'Figure_14_01c.png');
```

Sample code

# Expected Result




Result plot

# PCA Using Validated Stock Data

- Now for the PCA. Implement the PCA using outlined code.
- Visualize the results as next page. Use code to demonstrate several features of PCA

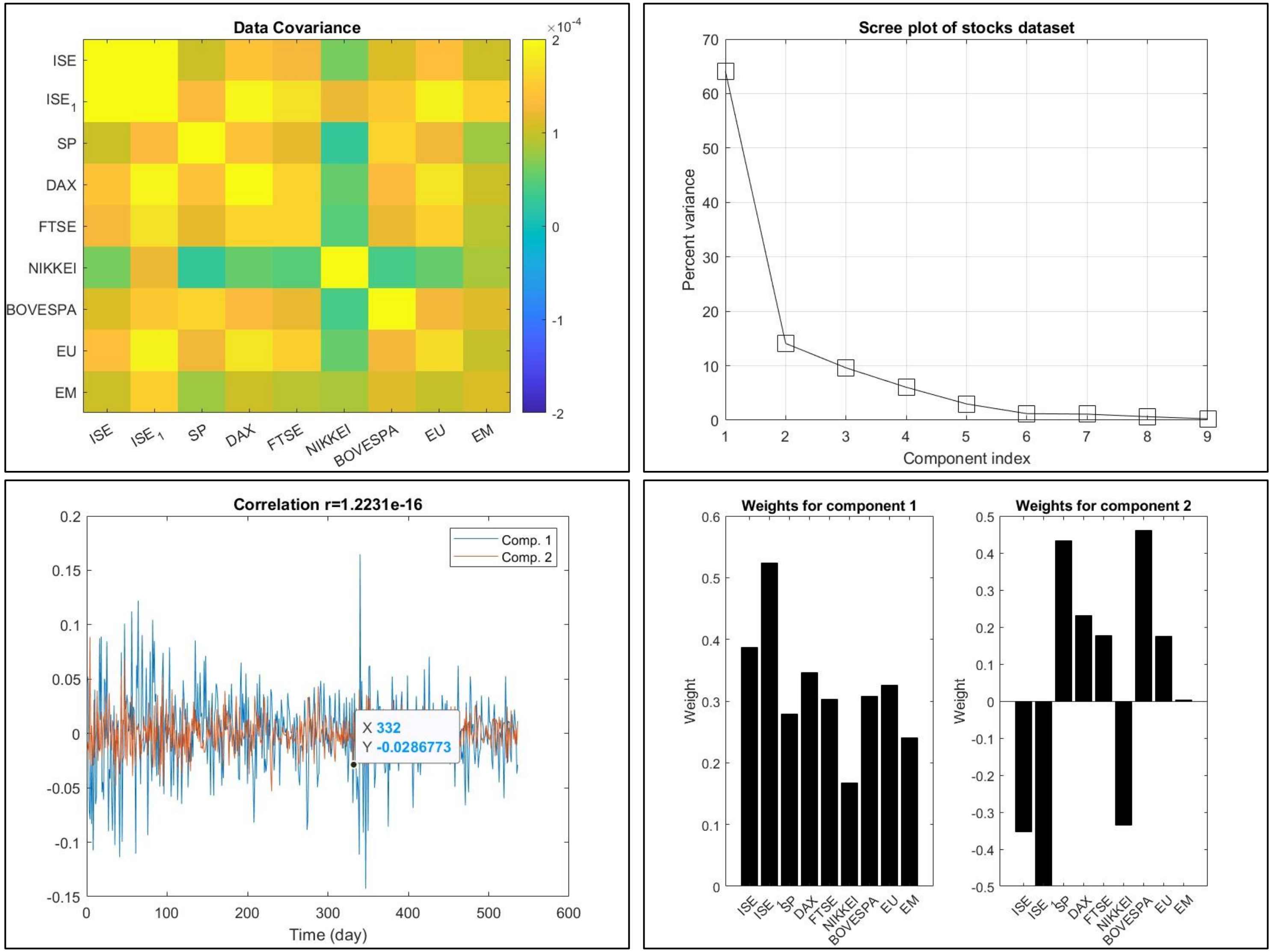
1. The variance of the component time series equals the eigenvalue associated with that component. You can see the results in disp()
2. The correlation between principal components (that is, the weighted combina- tions of the stock exchanges) 1 and 2 is zero, i.e., orthogonal.
3. Visualize the eigenvector weights for the first two components. The weights show how much each variable contributes to the component.

# PCA Using Validated Stock Data

```matlab
% Data citation: Akbilgic, Oguz. (2013). ISTANBUL STOCK EXCHANGE. UCI
Machine Learning Repository.
% data source website: https://archive-
beta.ics.uci.edu/ml/datasets/istanbul+stock+exchange

% import the data
url = 'https://archive.ics.uci.edu/ml/machine-learning-
databases/00247/data_akbilgic.xlsx';
raw_data = readtable(url, 'Sheet', 1, 'Range', 'A2');

dates = ;
data = ;
location = raw_data.Properties.VariableNames(2:end);

% show the correlation matrix in an image
figure;
corrMatrix = ; % code here!

% PCA Step 1: covariance matrix
% calculate covariance matrix (zero mean)
X = data;
X =;
covmat =;

% PCA Step 2: eigendecomposition

% PCA Step 3: sort results

% PCA Step 4: component scores using top 2 sort results

% PCA Step 5: eigenvalues to % variance

% Show that variance of the components equals the eigenvalue
disp('Variance of first two components:');
disp(var(components, 1));

disp('First two eigenvalues:');
disp(evals(1:2));

% visualize it
figure;
imagesc(covmat);
colorbar;
title('Data Covariance');
xticks(1:size(X, 2));
xticklabels(location);
yticks(1:size(X, 2));
yticklabels(location);
caxis([-.0002 .0002]);
saveas(gcf, 'Figure_14_01c.png');

% show scree plot
figure;
plot(factorScores, 'ks-', 'MarkerSize', 15);
xlabel('Component index');
ylabel('Percent variance');
title('Scree plot of stocks dataset');
grid on;
saveas(gcf, 'scree_plot.png');

% correlate first two components
figure;
plot(components);
xlabel('Time (day)');
legend('Comp. 1', 'Comp. 2');
title(['Correlation r=', num2str(corr(components(:, 1), components(:,
2)))]);

% bar plots of component loadings
figure;
subplot(1, 2, 1);
bar(evecs(:, 1), 'k');
xticks(1:size(X, 2));
xticklabels(location);
xtickangle(45);
ylabel('Weight');
title('Weights for component 1');

subplot(1, 2, 2);
bar(evecs(:, 2), 'k');
xticks(1:size(X, 2));
xticklabels(location);
xtickangle(45);
ylabel('Weight');
title('Weights for component 2');
```

Sample code

# Expected Result




Result plot

# THANK YOU
# FOR YOUR ATTENTION