Stress and strain: Stress tensor

강의명: 소성가공 (MSA0026)

정영웅

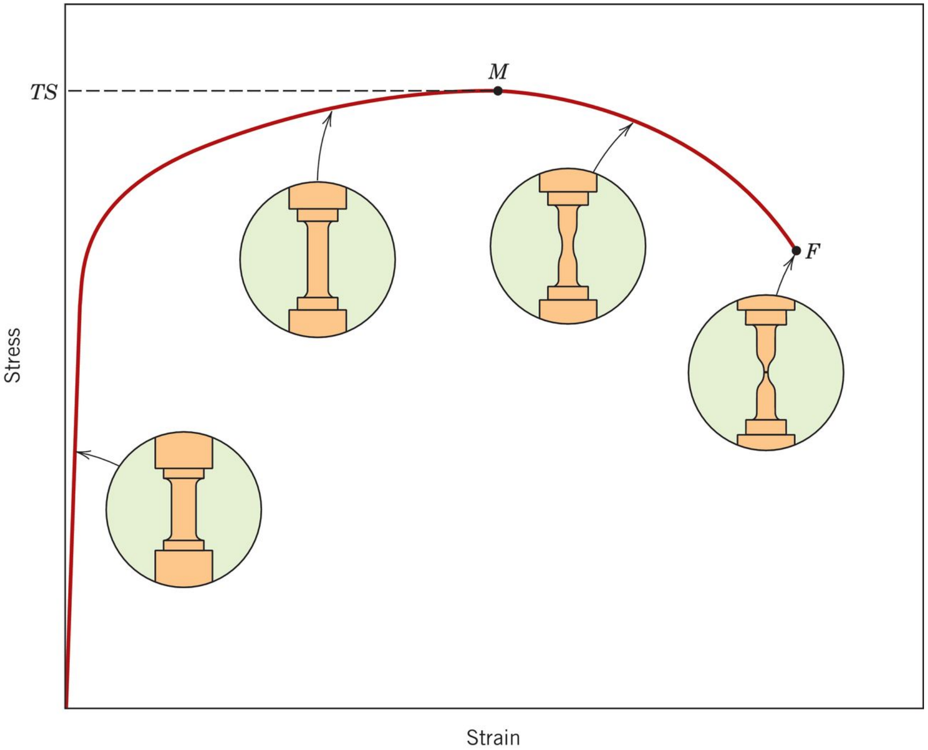
창원대학교 신소재공학부

YJEONG@CHANGWON.AC.KR

연구실: #52-212 전화: 055-213-3694

HOMEPAGE: HTTP://YOUNGUNG.GITHUB.IO

Necking



공칭 응력 / 공칭 변형률 곡선에서 보면 재료의 강도가 낮아지는 것처럼 보인다.

하지만 실제로는 변형이 집중되는 부분의 단면적이 감소함으로써 시편에 걸린 하중(load)이 전체적으로 감소하는 것이다.

실제로 재료의 '강도'는 소성변형으로 인해 증가한다 (세기, 크기 물리량 비교...).

위와 같이 공칭 응력에서는 시편의 단면적 감소를 고려하지 못하는 단점이 있다.

그리고 extensometer로는 시편의 국부 지역의 변형을 제대로 고려할 수 없다.

따라서, 시편의 단면적 감소를 고려한 응력의 정의가 필요하다.

진응력과 공칭 응력의 관계

- 앞서 아래의 정의를 배웠다.
- $\sigma^{engi} = \frac{F}{A_0}$ (1)
- $\sigma^{true} = \frac{F}{A}$ (2)

- 금속 재료의 경우, 소성 변형에 의한 부피가 매우 적다. 이는
- $A_0 l_0 = A_1 l_1$ (3)
- 의 관계로 이어진다. 이를 다시

정리하면

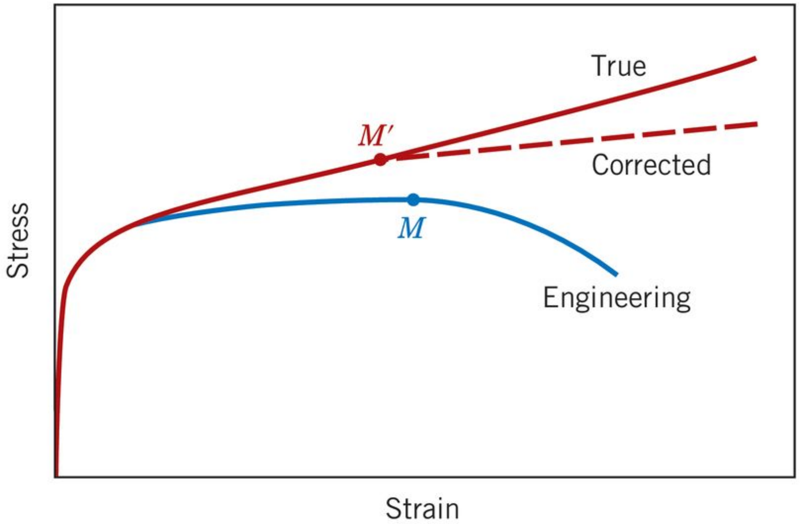
- $\frac{A_0}{A_1} = \frac{l_1}{l_0} = \frac{l_0 + \Delta l}{l_0}$ (4)

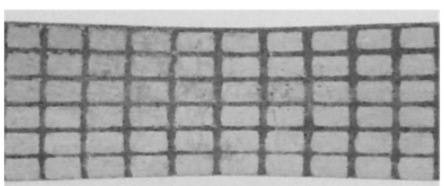
- (1)과 (2)에서
- $\frac{\sigma^{true}}{\sigma^{engi}} = \frac{A_0}{A_1} = 1 + \epsilon^{engi}$
- 따라서
- $\boldsymbol{\sigma^{true} = \sigma^{engi}(1 + \epsilon^{engi})}$

주의!

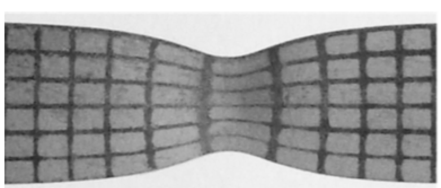
위의 관계식은 시편의 변형이 '균일' 할때만 가능 -> extensometer의 strain 측정법에 대해 잘 생각해보자. 변형이 '균일' 할 때만, 해당 시그널을 대표성이 있는 변형률로 변형이 가능하다.

변형률 곡선의 비교



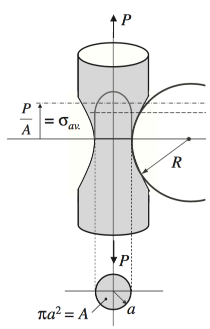


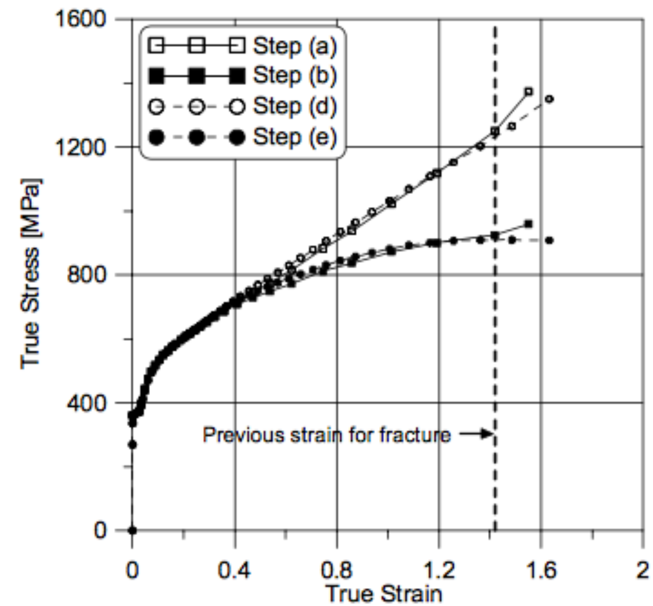
(a) $\varepsilon^p = 0.41$



(b) $\varepsilon^p = 1.42$ (close to fracture)

Bridgmann method





Choung and Cho, JMST 22, 2008 p 1039-1051

진응력/진변형률 관계 Model

- 일축 인장 실험을 통해 얻어진 진응력과 진변형률 곡선에서 쉽게 얻을 수 있는 물성중에 하나가 가공경화(work-hardening / strain-hardening)이다.
- 그러한 금속의 진응력 σ 과 진변형률 ε 곡선을 단순한 수학적 방정식으로 나타내면 편리하다.

$$\boxed{\sigma = K\varepsilon^{n} \quad \text{(Hollomon equation)}}$$

소성 변형 중의 탄성 회복

- 일축 인장 실험 도중에 하중을 제거하면, 전체 변형량에 포함이 된 '탄성' 변형은 '회복'이 된다. 따라서

$$\varepsilon^{\text{total}} = \varepsilon^{\text{elastic}} + \varepsilon^{\text{plastic}}$$

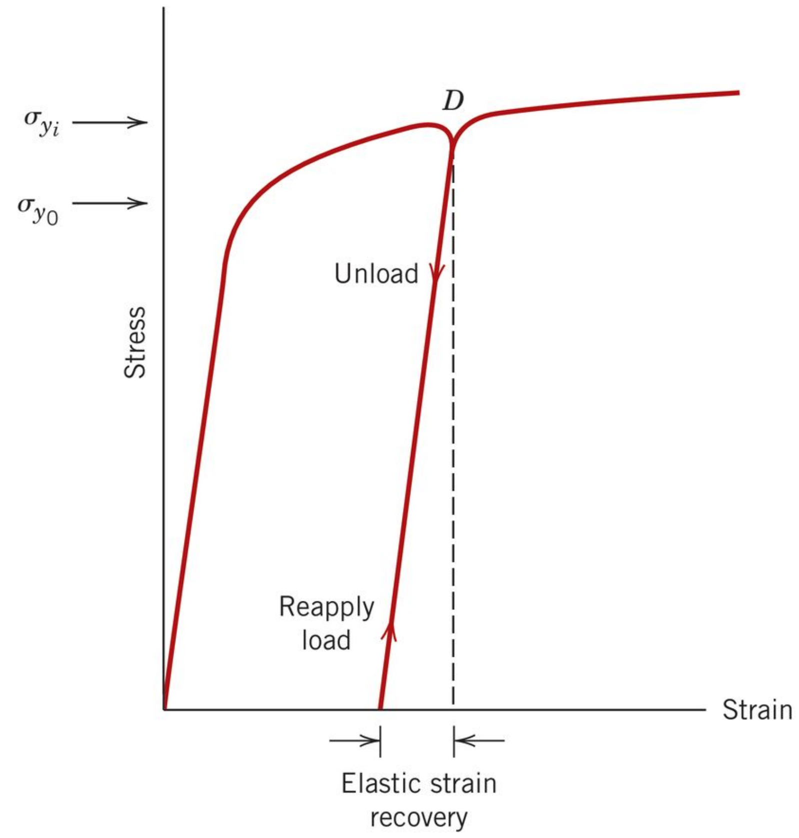
Elastic portion 은 하중이 제거되면 회복(즉 zero)된다.

$$\varepsilon^{\text{total}} = \varepsilon^{\text{elastic}} + \varepsilon^{\text{plastic}} = \mathbb{E}^{-1}\sigma + \varepsilon^{\text{plastic}}$$

Plastic strain은 elastic strain에 비해 더 복잡한 방식으로 응력과 관계가 있다.

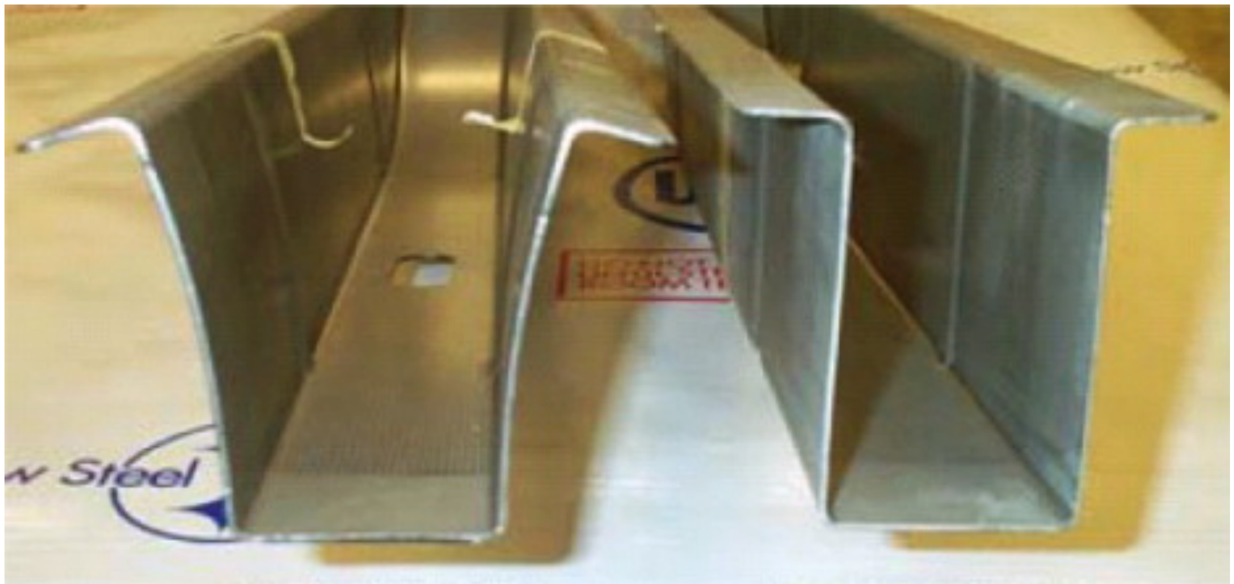
하중이 제거된 상태에서 다시 reload하면 앞서보다 '높은' 항복 강도를 보인다.

$$\sigma_{y0} \rightarrow \sigma_{yi}$$



소성 변형 중의 탄성 회복 (Springback)

친환경 자동차 개발 요구 -> 연비 개선 요구 -> 차체 경량화 요구
-> 고강도 금속, 혹은 낮은 탄성 계수 금속의 사용 -> springback 증가

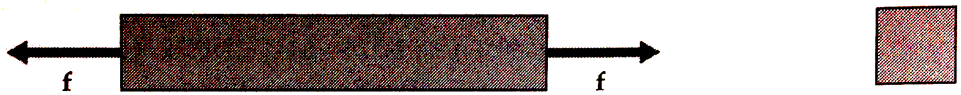


DP 350/600 **HSLA 350/450**

RH Wagoner, H Lim, M-G Lee, IJP 45, 2013 p 3-20

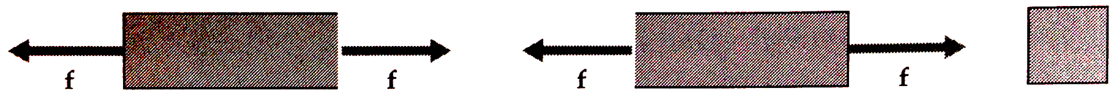
Internal stress of materials in force equilibrium

(a)



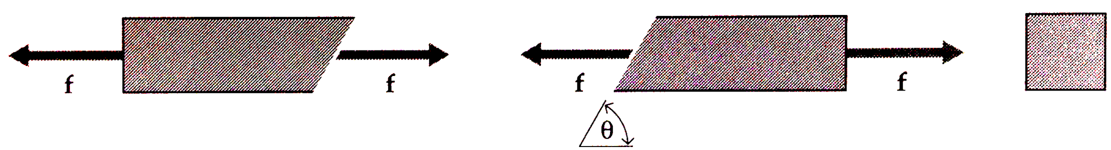
- 양끝에 F 라는 힘이 작용하는 막대기가 힘의 평형 상태에 놓여있다 (force equilibrium)

(b)



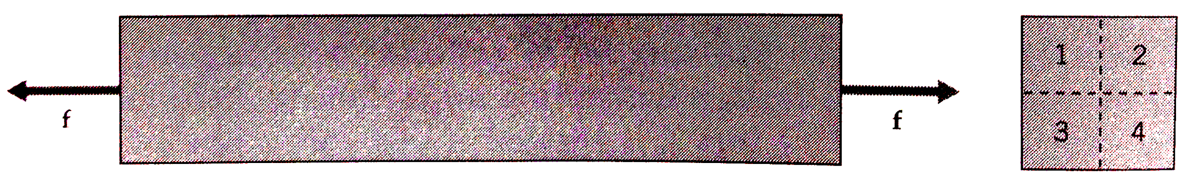
- 막대 내부에 작용하는 힘 (internal forces)을 살펴보기 위해 ‘가상으로 자른’ 면 (힘의 방향과 수직 방향)에도 힘이 작용할까?

(c)



- (b)와 마찬가지, 하지만 비스듬하게 자른 ’가상’의 면에는?

(d)



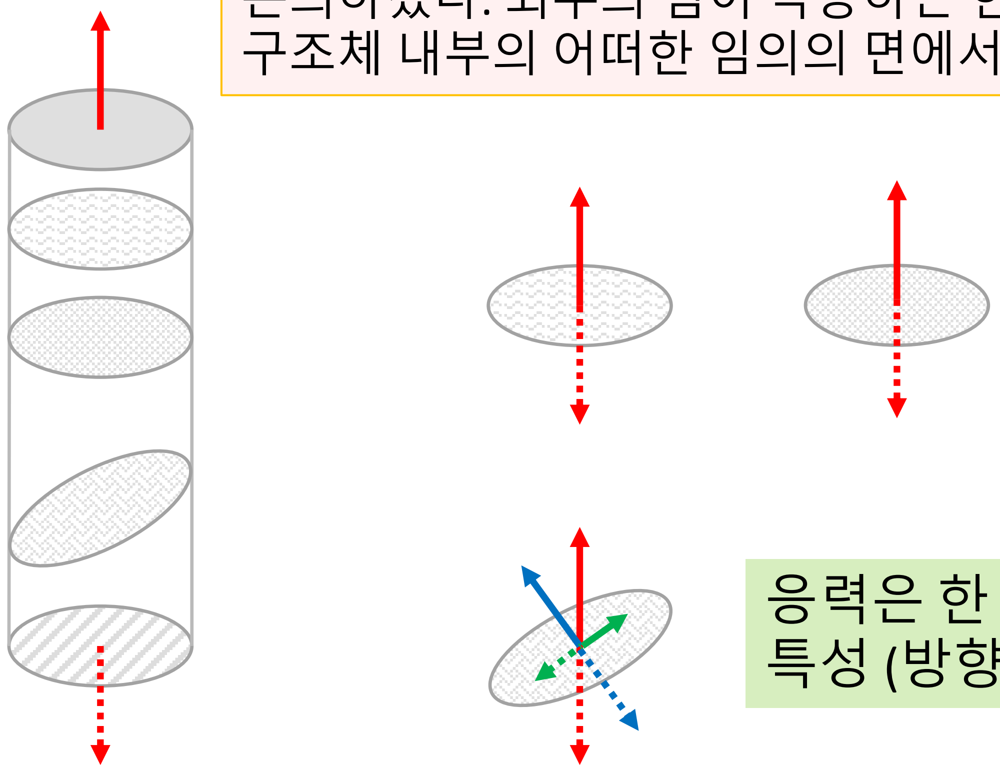
- 힘과 같은 방향으로 자를 경우?

* 가상의 면에서의 작용하는 힘을 이해하기 어렵다면, 결정 구조를 가진 금속 물체를 생각해보자.

Fundamentals of metal forming, R. H. Wagoner, Jean-Loup Chenot

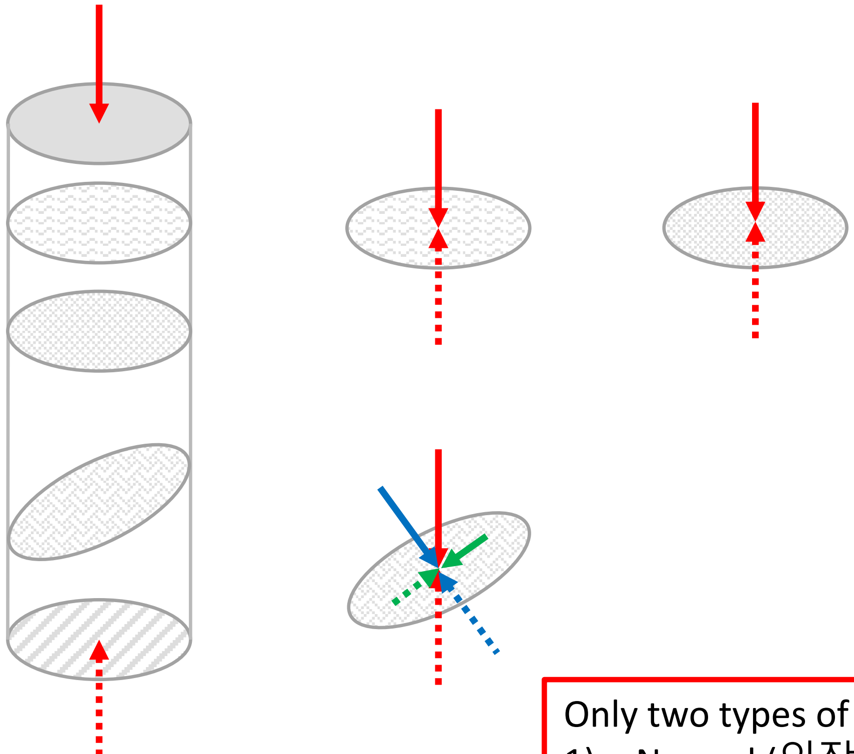
Recap: Tensile stress acting on a plane

앞으로 우리는 힘 평형 상태를 이루고 있는 물체만을 대상으로 논의하겠다. 외부의 힘이 작용하는 한 구조체가 힘 평형 상태라면, 그 구조체 내부의 어떠한 임의의 면에서도 힘 평형상태를 이루어야 한다.



응력은 한 면에 작용하는 힘과 그 면의 특성 (방향, 면적)으로 정의됨을 기억하자.

Recap: Compressive stress acting on a plane



Louis Cauchy

Only two types of stress components:
1) Normal (인장+ 혹은 압축-)
2) Shear (+, -)

3차원 공간에서는 오직 3면에서의 수직, 전단 응력 성분을 알기만 하면 되겠다.

속도의 예 (벡터와 비교)

- 한 물체의 속도를 정확히 알기위해서는 서로 독립적인 세 방향에서의 속도 성분을 알면 된다.
- 그러한 세 방향을 체계적으로 나타내기 위해 직각 좌표계(x,y,z축)를 활용할 수 있다.
  - 속도 벡터 v의 x,y,z 성분을 알려주고, 해당 좌표계를 명기하여 정확한 속도를 알려줄 수 있다.
- 응력의 경우도 비슷하다.
  - 응력을 정확하게 알기 위해서는 서로 독립적인 세방향이 필요하다.
  - 다만, 응력과 관계된 힘의 세방향, 그리고 그 힘이 작용할 수 있는 서로 독립적인 3면이 필요.

stress state (응력상태)

- 따라서, 응력상태는 총 9개의 구성 성분으로 표현될 수 있다.
  - 그 9개의 구성성분 중 3가지는 수직성분(normal components)이고, 나머지 6가지는 전단 성분(shear components)이다.
- A vector (1st rank tensor) consists of three components in the 3D space.
  - Example in 2D
    - 2차원 평면 동서남북 좌표계에서 영희가 동쪽으로 10m/s , 북쪽으로 5m/s 로 이동중이라면?
  - Example in 3D
    - 우주에서 유영하고 있는 우주인의 속도는 어떻게 나타내나? (기준 좌표계가 필요) 해당 좌표계의 basis axes (basis vectors)들과 평행한 성분들로 '분해'하여 표현.
- 만약, 구조체가 힘 평형 상태라면 언급된 9개의 응력 구성성분 중 오직 6개만 독립성분으로 남는다. (응력 텐서의 대칭성)
  - Static condition
  - Quasi-static condition

응력과 응력상태를 이루는 구성 요소
Stress state and stress (tensor) components

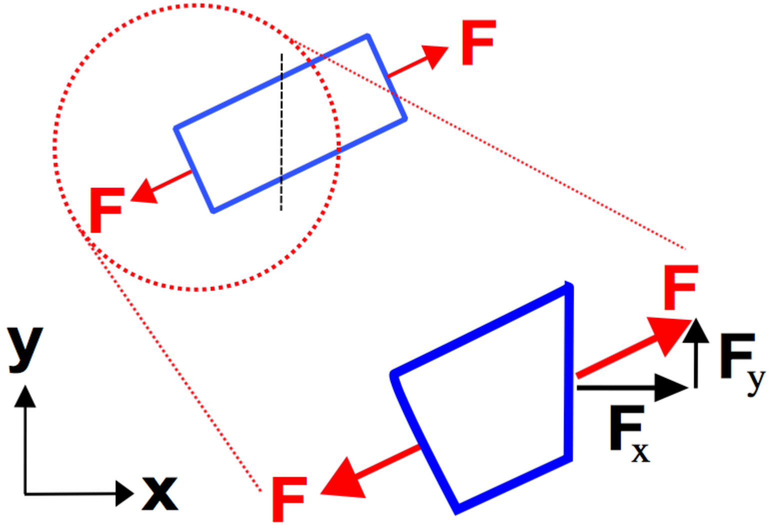
- Stress is defined as the **force** (load) applied to a certain **area**
- 전단과 수직 성분은 일반적인 응력상태에서 매우 *극단적인* 케이스라고 이해 할 수 있다.
- 앞서 살펴보았듯, 우리는 응력 상태를 표현할때 두가지 이상적인 조건에 만족하는 요소들로만 표현하기로 약속한다.
  A. 힘이 작용한 면적과 평행한 경우 (전단)
  B. 힘이 작용한 면적과 수직인 경우 (압축,인장)
- 대부분의 경우, 응력 상태는 A.와 B.의 경우가 혼재해 있다. 이를 표현하기 위해서는 위의 '성분'들의 각 물리량을 값 (숫자, value, quantity)으로 나타내면 되겠다.

Directions in stress

Two types of directions are involved for stress state:

1. The direction of force (힘의 방향)
2. The direction of the area (plane); which is usually represented by the normal of the plane (힘이 작용하는 면의 방향)

- These directions can be also represented by ‘values’ if you refer them to a certain coordinate system;
- The magnitude of ‘force’ can be also ‘quantified’ – that means it can be represented by ‘values’)



2차원 좌표계(x-y coordinates),
관심면(plane of interest),
그리고 힘 (Force)

Recap.

- 본 강의는 **힘평형(force equilibrium)** 조건에 해당하는 문제들로 국한하여 다룬다.
- 응력상태를 구성하는 성분 두가지는:
  - Normal: tensile(+) or compression(-)
  - Shear: forward shear (+) and backward shear (-) (전단의 방향 구분을 ‘sense’로 일컫는다)
- 응력상태를 표현하기 위해서는 두가지 방향이 필요하다. 그리고 각 방향은 벡터의 성분처럼 projection을 통해 ’값(value)’으로 표현이 가능하다. 2nd rank 텐서의 경우, 이 값들은 ‘두’ 방향이 좌표계에 projection되어 얻어진 ’세기’.
- 벡터는 최소한 세개의 독립적인 값(component value)으로 이루어져있다. 2nd rank 텐서의 경우에도 최소한의 독립적인 값의 개수가 정해져있다:
  - 벡터로 표현되는 힘과 다르게 응력 텐서는 두 종류 (normal, shear)가 있었고 이는 두 독립적인 방향사이의 관계 (수직이냐, 평형이냐)에 따라 나뉘었다.
  - 그렇다면, 응력텐서를 구성하는 최소한의 독립적인 값의 개수는 무엇일까?

Invention of Cauchy stress tensor 코시응력텐서

- **Cauchy** found that only **nine independent values** are required to fully describe a stress state; Any arbitrary stress state can represented by Cauchy stress **tensor**.

- Before learning more about stress tensor, let's review the stress; What is stress?
  - You might have some fundamental concept about stress: force/area

Stress at a material point

$$\sigma = \frac{F}{A}$$

특정한 크기를 가진 면에 작용하는 힘을 이용하여 적당한 크기를 가진 면에 '균일'하게 작용하는 응력을 표현; Within the area (denoted as A), the force is homogeneous.

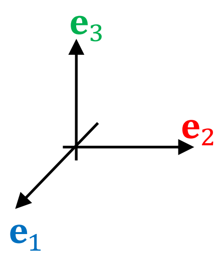
비균질한 재료의 경우, 응력이 재료의 각 점마다 다른 값을 가지며 '분포' 되어 있다. 이를 표현하기 위해서, 각 '점' 마다의 응력 상태를 표현할 수 있어야 한다. 따라서 매우 작은 점의 응력은 '균일'하다 할 수 있다. 이를 수학적으로 표현하자면

$$\boxed{\sigma = \lim_{A \to 0} \frac{F}{A}}$$

앞으로는 가상의 '면'이 한 물질점(material point)에 귀속되어 있다고 가정한다. 그리고 무한소 큐브를 이용한 응력 상태를 표현하는 방법 (응력 텐서)를 알아보자.

Coordinate system and basis vectors

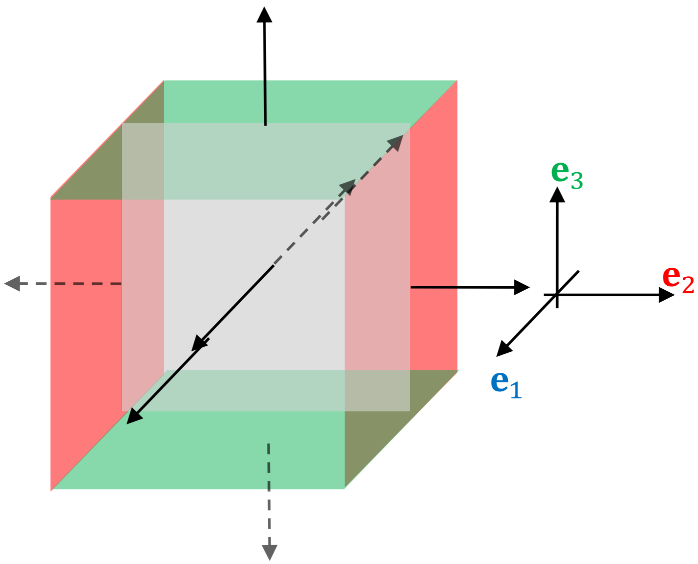
- 앞으로 좌표계를 설명할때 좌표계의 근간이 되는 방향들을 normal vector (즉 크기가 1인 벡터)로 표현.
- Cartesian coordinate system은 orthonormal coordinate system
- 서로 수직한 세 normal vector로 표현이 가능하다.



그 세 normal vector들을 basis vector로 칭하겠다. 그리고 각각 e_1, e_2, e_3 로 나타내겠다.

앞으로 다루게 될 응력과 변형률 텐서의 성분(component)을 표현할때 쓰이는 subscript 인덱스 (1,2,3)는 각각 e_1, e_2, e_3 벡터들을 뜻한다.

Cauchy stress tensor



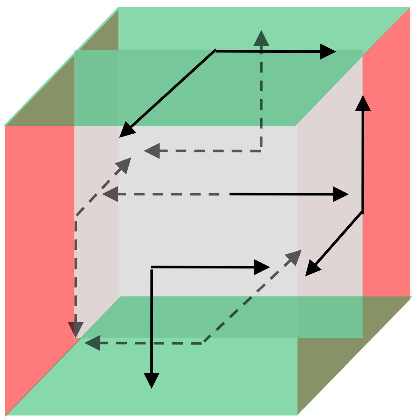
응력 텐서는 무한소 큐브의 3면에 작용하는 성분들은 총 '두'가지 방향에 의존한다:
1) 작용하는 면의 수직방향 (normal)
2) 작용하는 힘의 방향 (normal or shear)

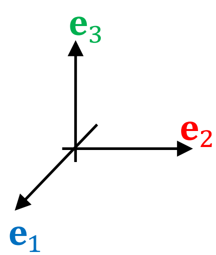
3차원 직교좌표계 (Cartesian coordinate)에서 **총 9개의 조합이 가능하다.** 즉 9개의 성분으로 표현가능. 하지만 힘의 평형 상태에서는 몇몇의 shear 성분들이 서로 같은 값을 가져야 한다(추후에 다루자). 그에 따라 총 6개의 '독립'적인 성분(component)들만 남게 된다.

위의 무한소 큐브는 무한히 작은 물질점을 대표한다.

즉, 주어진 좌표계에서 6개의 독립적인 성분(component)들만 안다면 해당 물질 무한소(material infinitesimal point)의 응력 상태(stress state)를 완벽히 나타낼 수 있다.

Cauchy stress tensor





응력 텐서는 무한소 큐브의 3면에 작용하는 성분들은 총 ‘두’가지 방향에 의존한다:

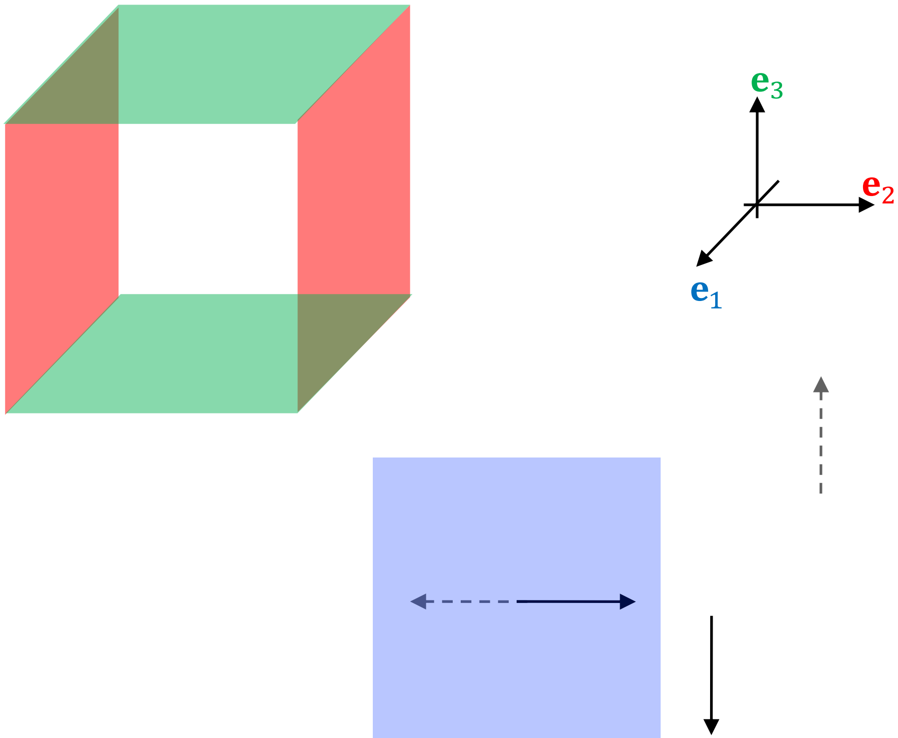
1) 작용하는 면의 수직방향 (normal)
2) 작용하는 힘의 방향 (normal or shear)

3차원 직교좌표계 (Cartesian coordinate)에서 **총 9개의 조합이 가능하다.** 즉 9개의 성분으로 표현가능. 하지만 힘의 평형 상태에서는 몇몇의 shear 성분들이 서로 같은 값을 가져야 한다(추후에 다루자). 그에 따라 총 6개의 ’독립’적인 성분(component)들만 남게 된다.

위의 무한소 큐브는
무한히 작은
물질점을 대표한다.

즉, 주어진 좌표계에서 6개의 독립적인 성분(component)들만 안다면 해당 물질 무한소(material infinitesimal point)의 응력 상태(stress state)를 완벽히 나타낼 수 있다.

Shear stress (and its sign)



2D stress tensor representation using matrix

- 주어진 (x-y) 좌표계가, 혹은 $(\mathbf{e}_1 - \mathbf{e}_2)$ 좌표계가 명확히 주어졌다면 '행렬'의 형태를 빌려 표현할 수 있다.

$$\sigma_{xx} = \frac{F_x}{A_x}$$

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}$$

$$\boxed{\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}}$$

$$\sigma = \begin{bmatrix} \sigma_{11} & \tau_{12} \\ \tau_{21} & \sigma_{22} \end{bmatrix}$$

이 표기법을
앞으로 사용하자

τ (tau) 기호가 전단력을 normal stress component 와 구분하기 위해 종종 사용된다.

각 응력 성분에 붙인 첨자 형태의 수(1,2,3)은 해당 성분과 관련있는 basis vector(e_1, e_2, 혹은 e_3)를 의미한다.

Warning: a tensor is not a matrix. We just borrow the form of matrix to write down the tensor components.

There are cases where you cannot write down the tensor components to matrix;
Tensors can be in multi-dimension – like 6, or 9.

3D stress tensor represented by matrix

- 2D 응력 텐서의 확장

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

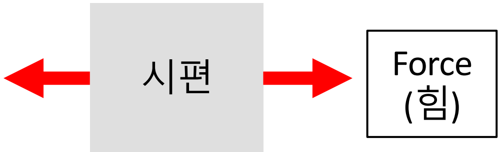
파란색: normal of the plane
빨간색: direction of the force

The subscript number refers to the basis axis, with which the associated direction is parallel.

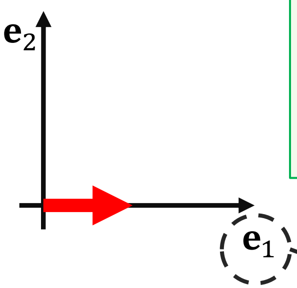
같은 가로줄에 놓인 성분들(열, 수평축)은
같은 면에 작용한다.

같은 세로축(기둥)에 놓인 성분들은,
관계있는 힘이 같은 방향(+/-)에
작용한다.

좌표변환: 2차원 공간내의 벡터



좌표계 (coordinate system) 변환은 좌표축 (coordinate axes) 변환으로도 불린다.



1. Force vector, **F** 또한 주어진 좌표계(coordinate system)이 있다면 행렬의 형태로 표현 가능

$$\boxed{\vec{\boldsymbol{F}} = [F_1 \quad F_2]}$$

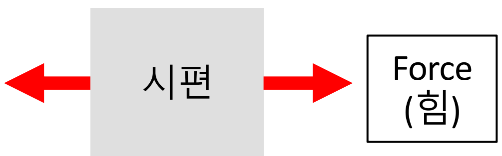
2. 혹은 주어진 좌표계의 'basis vector'를 사용하여서 표현 가능

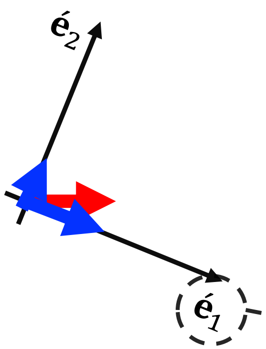
$$\vec{\boldsymbol{F}} = F_1\vec{\boldsymbol{e}}_1 + F_2\vec{\boldsymbol{e}}_2$$

F_1, F_2 는 주어진 좌표계 ($\mathbf{e}_1, \mathbf{e}_2$ basis vector로 이루어진)에서 표현된 Force vector **F**의 성분 (component)

앞으로 벡터위의 화살표는 생략하고, 대신 **굵은 글씨체** (**bold face**) 사용.

좌표변환: 2차원 공간내의 벡터



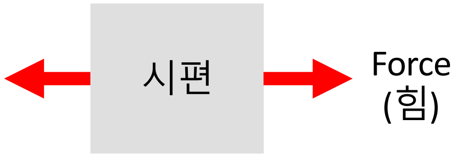


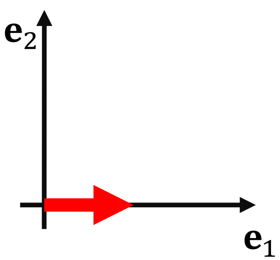
$$\boldsymbol{F} = [\acute{F}_1 \quad \acute{F}_2]$$

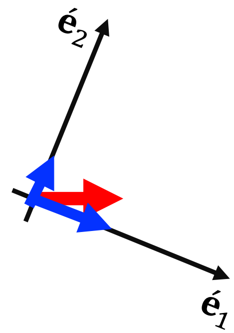
혹은 주어진 좌표계의
'basis vector'를
사용하여서 표현 가능

$$\boldsymbol{F} = \acute{F}_1 \acute{\mathbf{e}}_1 + \acute{F}_2 \acute{\mathbf{e}}_2$$

좌표변환: 2차원 공간내의 벡터



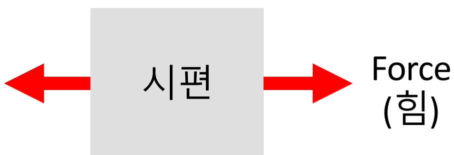


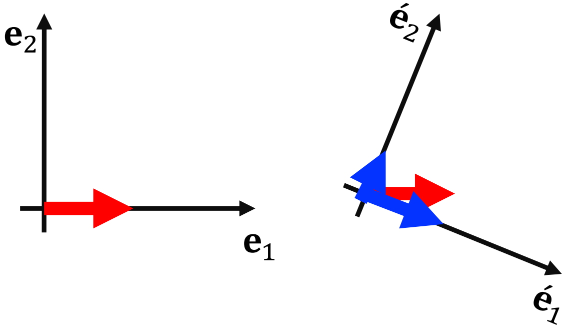


Vector를 나타내는 좌표계의 변환일 뿐, Force vector 자체의 물리량은 변환없다. 즉 시편에 작용하는 힘은 변화 없다.

$$\boldsymbol{F} = F_1\mathbf{e}_1 + F_2\mathbf{e}_2 = \acute{F}_1\acute{\mathbf{e}}_1 + \acute{F}_2\acute{\mathbf{e}}_2$$

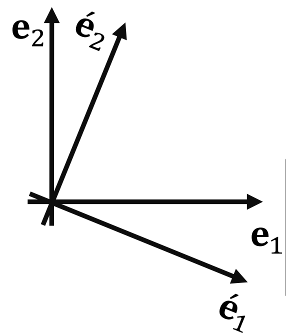
좌표변환: 2차원 공간내의 벡터





$$\boldsymbol{F} = F_1\mathbf{e}_1 + F_2\mathbf{e}_2 = \acute{F}_1\acute{\mathbf{e}}_1 + \acute{F}_2\acute{\mathbf{e}}_2$$

Vector를 나타내는 좌표계의 변환일 뿐,
Force vector 자체의 물리량은 변환없다.
즉 시편에 작용하는 힘은 변화 없다.



두 좌표계간의 '관계'를 안다면 F_1, F_2를 바탕으로 $\acute{F}_1, \acute{F}_2$로 '변환' 할 수 있다.

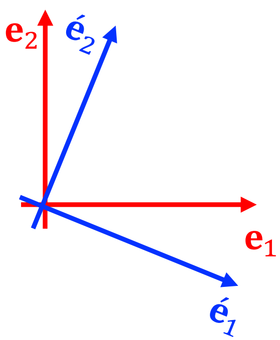
이때의 '변환'을 transformation,
즉 'form' (형태) 의 trans (바뀜)

Transformation은 각 basis vector간의 관계로 설명이 가능

$\mathbf{e}_1$과 $\acute{\mathbf{e}}_1$그리고 $\mathbf{e}_1$ 과$\acute{\mathbf{e}}_2$
$\mathbf{e}_2$과 $\acute{\mathbf{e}}_1$그리고 $\mathbf{e}_2$ 과$\acute{\mathbf{e}}_2$

관계를 각 basis vector (좌표계의 축) 간 angle로 표현한다면?

2차원 좌표계 간의 관계



2차원 좌표계의 각 축간의 각도를 표현하려면?

한 좌표계에서 다른 좌표계로 '변환'을 시켜주는 **direction cosine**들의 모임을 '행렬'로 표현

Old co. sys.

New co. sys.

$$\begin{array}{cc} & \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} \\ \begin{bmatrix} \acute{\mathbf{e}}_1 \\ \acute{\mathbf{e}}_2 \end{bmatrix} & \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \end{array}$$

a_{ij}

Old Co. Sys. 의 j 번째 basis vector와 New Co. Sys의 i 번째 basis vector사이의 direction cosine

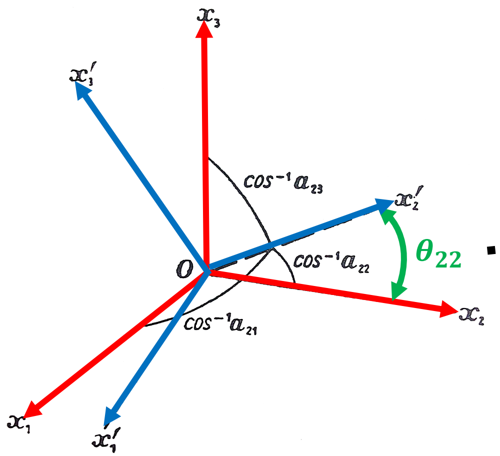
direction cosine? 다음장에…

방위의 수학적 표현 방법

- Quaternion (convenient mathematical notation for representing orientations)
- Rodriguez (a vector and an angle; any arbitrary orientation; misorientation)
- Euler angles (Bunge notation (ϕ_1, Φ, ϕ_2) is widely used – ZXZ convention)
- Transformation matrix

A LaTeX script to generate below illustration:
https://youngung.github.io/euler/

두 축 (axes) 간의 방위 관계



$a_{22} = cos(\theta_{22})$

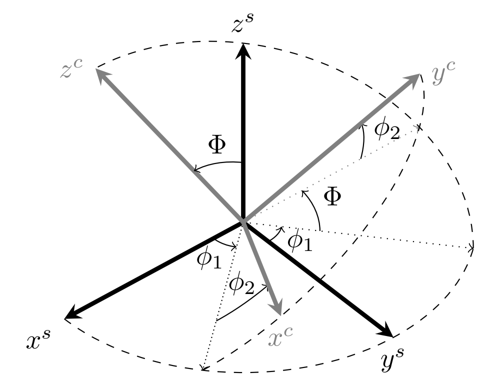
위 그림에서는 x_1, x_2, x_3 로 이루어진 좌표계1과 $x`_1, x`_2, x`_3$로 이루어진 또 다른 좌표계2가 나타나있다. 좌표계 2의 $x`_2$ basis vector와 좌표계 1의 각 basis vector들과의 관계를 a_{21}, a_{22}, a_{23} 의 direction cosine으로 표현했다.

Old co. sys.

New co. sys.

$$\begin{array}{cc} & [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] \\ \begin{bmatrix} \acute{\mathbf{e}}_1 \\ \acute{\mathbf{e}}_2 \\ \acute{\mathbf{e}}_3 \end{bmatrix} & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \end{array}$$

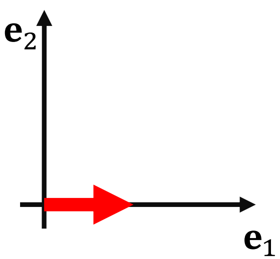
Transformation matrix (not tensor)



서로 다른 두 좌표계(axes, coordinate)의 방위 관계가 하나의 transformation matrix로 표현됨.

2차원 벡터의 좌표 변환 표기#1

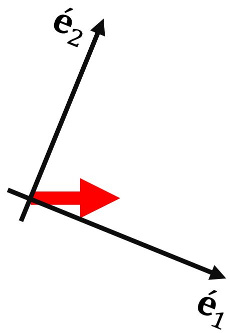
Old coordinate system



$$\boldsymbol{F} = [10\text{N} \quad 0\text{N}]$$

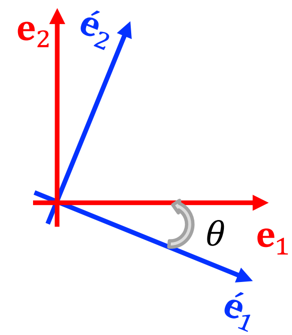
10N의 힘이 작용

New coordinate system



$$\boldsymbol{F} = [?\,N \quad ?\,\text{N}]$$

2차원 coordinate system간에 관계는 한 방향으로(시계반대반향ccw 기준)의 '각'회전 angular rotation으로 나타낼 수 있다.



따라서 $-\theta$ 로 Old Coordinate 를 CCW 회전하면 New Coordinate가 된다.

Old co. sys.

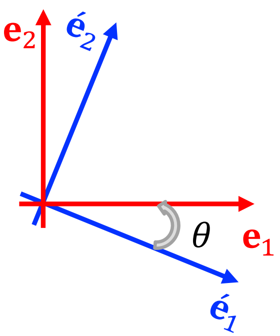
New co. sys.

$$\begin{array}{cc} & [\mathbf{e}_1 \quad \mathbf{e}_2] \\ \begin{bmatrix} \acute{\mathbf{e}}_1 \\ \acute{\mathbf{e}}_2 \end{bmatrix} & \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \end{array}$$

$$a_{11} = \cos(-\theta) = \cos\theta$$
$$a_{21} = \cos(90° - \theta) = \sin\theta$$
$$a_{12} = \cos(90° + \theta) = -\sin\theta$$
$$a_{22} = \cos(\theta)$$

2차원 벡터의 좌표 변환 표기#2

2차원 coordinate system간에 관계는 한 방향으로(시계반대반향ccw 기준)의 '각'회전 angular rotation으로 나타낼 수 있다.



따라서 $-\theta$ 로 Old Coordinate 를 CCW 회전하면 New Coordinate가 된다.

Old co. sys.

$$\begin{array}{c|c} & [\mathbf{e}_1 \quad \mathbf{e}_2] \\ \hline \text{New co. sys.} \begin{bmatrix} \acute{\mathbf{e}}_1 \\ \acute{\mathbf{e}}_2 \end{bmatrix} & \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \end{array}$$

$$a_{11} = \cos\theta$$
$$a_{21} = \cos(90^\circ - \theta) = \sin\theta$$
$$a_{12} = \cos(90^\circ + \theta) = -\sin\theta$$
$$a_{22} = \cos\theta$$

$$\acute{F}_1 = \cos\theta_{11} F_1 + \cos\theta_{12} F_2 = a_{11}F_1 + a_{12}F_2$$
$$\acute{F}_2 = \cos\theta_{21} F_1 + \cos\theta_{22} F_2 = a_{21}F_1 + a_{22}F_2$$

$$\acute{F}_1 = a_{11}F_1 + a_{12}F_2 = \sum_{\mathrm{j}}^{\mathrm{j}=1,2} a_{1\mathrm{j}}F_{\mathrm{j}}$$

$$\acute{F}_2 = a_{21}F_1 + a_{22}F_2 = \sum_{\mathrm{j}}^{\mathrm{j}=1,2} a_{2\mathrm{j}}F_{\mathrm{j}}$$

2차원 벡터의 좌표 변환 표기#3

$$\acute{F}_1 = \cos\theta_{11} F_1 + \cos\theta_{12} F_2 = a_{11}F_1 + a_{12}F_2$$
$$\acute{F}_2 = \cos\theta_{21} F_1 + \cos\theta_{22} F_2 = a_{21}F_1 + a_{22}F_2$$

$$\acute{F}_1 = a_{11}F_1 + a_{12}F_2 = \sum_{j}^{j=1,2} a_{1j}F_j$$

$$\acute{F}_2 = a_{21}F_1 + a_{22}F_2 = \sum_{j}^{j=1,2} a_{2j}F_j$$

3차원으로
확장

$$\acute{F}_1 = a_{11}F_1 + a_{12}F_2 \boxed{+ a_{13}F_3} = \sum_{j}^{j=1,\boxed{3}} a_{1j}F_j$$

$$\acute{F}_2 = a_{21}F_1 + a_{22}F_2 \boxed{+ a_{23}F_3} = \sum_{j}^{j=1,\boxed{3}} a_{2j}F_j$$

$$\boxed{\acute{F}_3 = a_{31}F_1 + a_{32}F_2 + a_{33}F_3 = \sum_{j}^{j=1,3} a_{3j}F_j}$$

쉬워보이지만, 직접 연습해보면 생각보다 어려울 수 있습니다.
따라서, 혼자서 연습해보는 게 필요해요.

2차원 벡터의 좌표 변환 표기#4

$$\acute{F}_1 = a_{11}F_1 + a_{12}F_2 + a_{13}F_3 = \sum_{j}^{3} a_{1j}F_j$$

$$\acute{F}_2 = a_{21}F_1 + a_{22}F_2 + a_{23}F_3 = \sum_{j}^{3} a_{2j}F_j$$

$$\acute{F}_3 = a_{31}F_1 + a_{32}F_2 + a_{33}F_3 = \sum_{j}^{3} a_{3j}F_j$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Transformation ‘matrix’.
엄밀하게 얘기하면
tensor가 아니다.

앞을 더욱 축약하자면… 한 coordinate system ($\mathbf{e}_i$, i=1,2,3) 에서 표기된 벡터 vector **F**를 또 다른 coordinate system ($\acute{\mathbf{e}}_i$, i=1,2,3) 으로 변환 (transformation) 하는 작업을 두 coordinate system을 ‘이어’주는 transformation matrix $[a_{ij}]$ 를 사용하여 다음과 같이 축약하여 이용할 수 있다.

$$\acute{F}_i = \sum_{j}^{3} a_{ij}F_j \qquad (i = 1,2,3)$$

위를 더욱 더 축약하자면 …

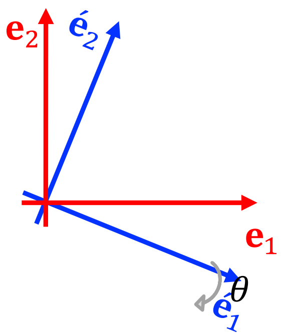
$$\acute{F}_i = a_{ij}F_j \qquad (i, j = 1,2,3)$$

각 좌우변에서 **반복되는** index에 대한 ‘summation’이 생략되었다.

반대로, 축약된 방식으로 표현된 tensor ‘operation’을 보고 생략된 summation 기호를 파악해내어야 한다.

2차원 벡터의 좌표 변환 표기#5

2차원 coordinate system간에 관계는 한 방향으로(시계반대반향ccw 기준)의 '각'회전 angular rotation으로 나타낼 수 있다.

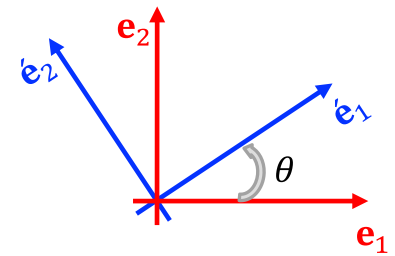


따라서 $-\theta$ 로 Old Coordinate 를 CCW 회전하면 New Coordinate가 된다.

$$\begin{array}{ccc} & & \text{Old co. sys.} \\ & & [\mathbf{e}_1 \quad \mathbf{e}_2] \\ \text{New co. sys.} & \begin{bmatrix} \acute{\mathbf{e}}_1 \\ \acute{\mathbf{e}}_2 \end{bmatrix} & \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \end{array}$$

$$a_{11} = \cos\theta$$
$$a_{21} = \cos(90^\circ - \theta) = \sin\theta$$
$$a_{12} = \cos(90^\circ + \theta) = -\sin\theta$$
$$a_{22} = \cos\theta$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



따라서 θ 로 Old Coordinate 를 CCW 회전하면 New Coordinate가 된다.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

좌표전환: 2nd order 텐서

Vector의 경우 각 component에 basis vector가 ‘하나’씩 관련됨

2nd order tensor의 경우 2개의 basis vector가 관련됨 (힘의 방향과 면의 방향)

Einstein summation convention

아인슈타인 축약 표기법

$$\acute{\sigma}_{ij} = \sum_{k}^{3}\left\{\sum_{l}^{3} a_{ik}a_{jl}\sigma_{kl}\right\} = a_{ik}a_{jl}\sigma_{kl}$$

$$\mathbf{R} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

좌표전환 matrix의 transpose?

행렬의 **R**의 transpose operation은 …

R^{T}: transformation 행렬 R의 transpose; $\left(\mathrm{R_{ij}}\right)^{\mathrm{T}} = \mathrm{R_{ji}}$

R^{T}: transformation 행렬 R의 transpose; $\left(\mathrm{R_{ij}}\right)^{\mathrm{T}} = \mathrm{R}^{-1}$

Transpose operation 예제

- 다음과 같은 transformation matrix가 주어져 있다.

$$\mathbf{R} = \begin{bmatrix} 0.848 & -0.528 & 0.044 \\ 0.530 & 0.845 & -0.075 \\ 0.003 & 0.087 & 0.996 \end{bmatrix}$$

- 위 transformation matrix를 활용하여, old coordinate system에서 [1,0,0]으로 표기되는 벡터가 좌표 변환 후 어떻게 바뀌는지 각 구성요소 [x,y,z]의 x,y,z 값을 구하시오.

- 위 transformation matrix의 transpose 행렬을 구하고, 이를 활용하여 앞서 구한 x,y,z 벡터를 변환한 다음 [a,b,c]로 표기되는 벡터 값을 구하시오.

좌표계 변환의 의미

- 물리학에서 좌표계는 단순히 관찰자의 편의에 의해 설정 된다. 하지만 어떠한 관찰자가 보더라도 물리적 의미와 법칙은 영향을 받지 않는다. 즉 물리량은 임의로 설정된 좌표계와 상관없이 일정하고, 물리 법칙은 관찰자의 좌표계에 무관하다.
- 텐서의 형태로 표현되는 물리량 (혹은 물성) 들도 좌표계 전환에 의해서 바뀌는 것은 아니다.
- 다만, 물리량을 표현하는 텐서의 표기법(약속)에 의해 텐서의 성분값들이 바뀌는 것일 뿐이다. 앞서 우리는 텐서라는 표기법(약속)에 따라 주어진 좌표계에서 또 다른 좌표계로 바뀌어 참조될때 텐서의 성분값들이 어떻게 변하는지 살펴보았다.
- 텐서의 ‘rank’에 따라서 좌표계 전환법이 어떻게 바뀌는지 알아보았다.

Inner dot product

- Tensor 표기를 index에 함께 표기하지 않고 **bold-face** symbol로 표기하기도 함
  - 예) σ_{ij} 대신 $\boldsymbol{\sigma}$로, F_i 대신 **F** 로 표기.

- Inner dot product는 텐서와 텐서간의 여러 operation중에 하나로, 참여하는 텐서간의 '안쪽' index가 되풀이되어 더해지는(summed) 작업.

- 앞서 보았던 벡터의 좌표변환 (coordinate transformation)을 **bold-face**로 바꿔 center dot을 사용하여 표기
  - $\mathbf{\acute{F}} = \mathbf{R} \cdot \mathbf{F}$ 로 표기
  - $\acute{F}_i = R_{ij} F_j$ (matrix 형태를 빌려 index를 표기 하는 법)

- 2nd order tensor의 좌표변환은…
  - $\boldsymbol{\acute{\sigma}} = \mathbf{R} \cdot \boldsymbol{\sigma} \cdot \mathbf{R}^{T}$
  - $\acute{\sigma}_{ij} = R_{ik} \sigma_{kl} \left(R^{T}\right)_{lj} = R_{ik} \sigma_{kl} R_{jl} = R_{ik} R_{jl} \sigma_{kl} = R_{jl} R_{ik} \sigma_{kl}$

Double Inner dot product (= tensor contraction)

- Work done to an infinitesimal material point:

$$\text{Work done} = \sigma_{ij}\varepsilon_{ij} = \sum_{i}^{3}\sum_{j}^{3}\sigma_{ij}\varepsilon_{ij}$$

Summation over a pair of index

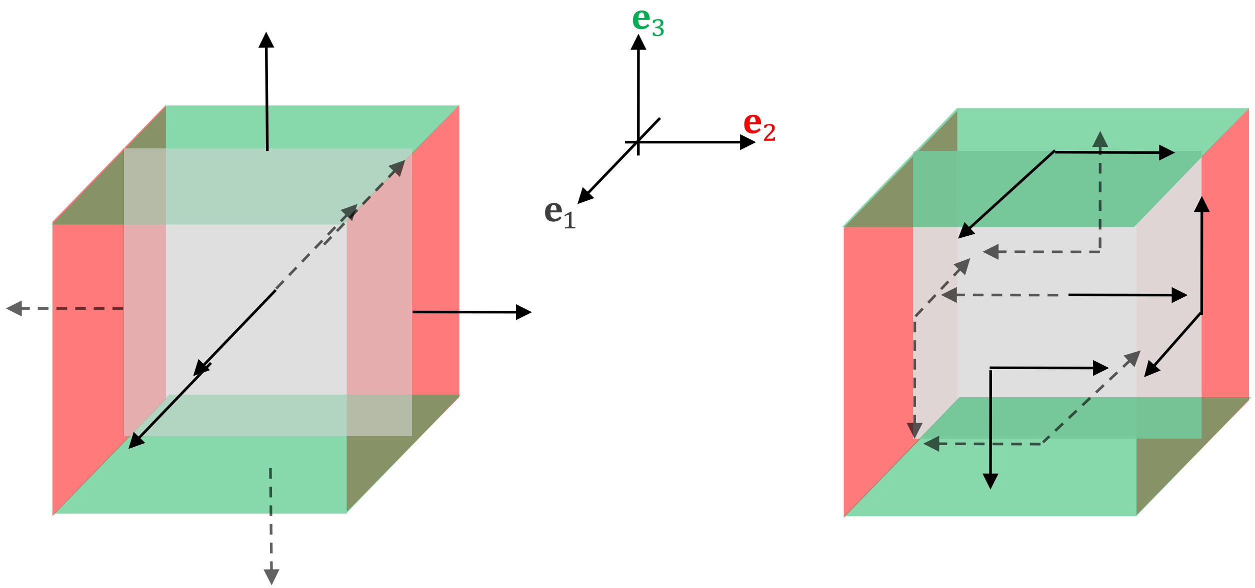
$$= \sigma_{11}\varepsilon_{11} + \sigma_{12}\varepsilon_{12} + \sigma_{13}\varepsilon_{13} + \sigma_{21}\varepsilon_{21} + \sigma_{22}\varepsilon_{22} + \sigma_{23}\varepsilon_{23} + \sigma_{31}\varepsilon_{31} + \sigma_{32}\varepsilon_{32} + \sigma_{33}\varepsilon_{33}$$

Colon 기호(:)로 double inner dot operation (=tensor contraction)을 나타낸다.

$$\mathrm{W} = \boldsymbol{\sigma} : \boldsymbol{\varepsilon}$$

2^{nd} order tensor의 symmetry

- Centro symmetry (짝힘, paired force – due to force equilibrium)
  - 원점에 대한 대칭



- Force (혹은 momentum) equilibrium -> Diagonal symmetry
  - $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{T}}$ 즉 $\sigma_{ij} = \sigma_{ji}$

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix}$$

따라서, 총 6개의 독립적인 component가 존재...

물리량 표기법

- 물리량은 대게 'Greek' alphabet을 기호로 사용하여, 기준이되는 좌표계에의 성분(component)값을 표현하는 index를 대개 subscript (간혹 superscript)로 덧붙여 나타낸다.
- Scalar를 표현하는 기호에는 인덱스가 없다 – 좌표계에 무관
  - 예: 질량 (m) 밀도 (ρ) , 온도(T)
- vector에는 인덱스를 하나 붙인다
  - 예: 속도 (v_i), 힘 (f_i). 이때 각 index는 1,2,3 즉 세개의 구성성분 (component)가 존재.
- 2nd rank tensor에는 두개의 인덱스를 붙인다
  - 예: 응력 (σ_{ij}) 각각의 index i와 j는 1,2,3 세개의 구성성분을 가진다. 따라서 총 9개의 구성성분 (3x3)이 존재한다.
- 3rd rank tensor는 세개의 인덱스 총 3x3x3 27개의 구성성분 필요
- 4th rank tensor는 네개의 인덱스, 총 3x3x3x3, 81개의 구성성분 필요:
  - 탄성계수 (elastic modulus) 텐서
- 벡터는 주로 [1x3] 혹은 [3x1]의 행렬 형태를 빌려 쓴다
- Scalar는 0th rank tensor, vector는 1st rank tensor

type	No. of Indices (=No. of transformation needed)	예시
Scalar	0	mass, density
Vector	1	velocity, force
2nd rank tensor	2	stress, strain
3rd rank tensor	3	Piezoelectric moduli
4th rank tensor	4	Elastic moduli

참고: Moduli는 modulus의 복수형

Transformation rule for tensor

- Follow this link:

- https://youngung.github.io/tensors/