Stress and strain:

Euler angle과 좌표변환법

강의명: 소성가공 (MSA0026)

정영웅

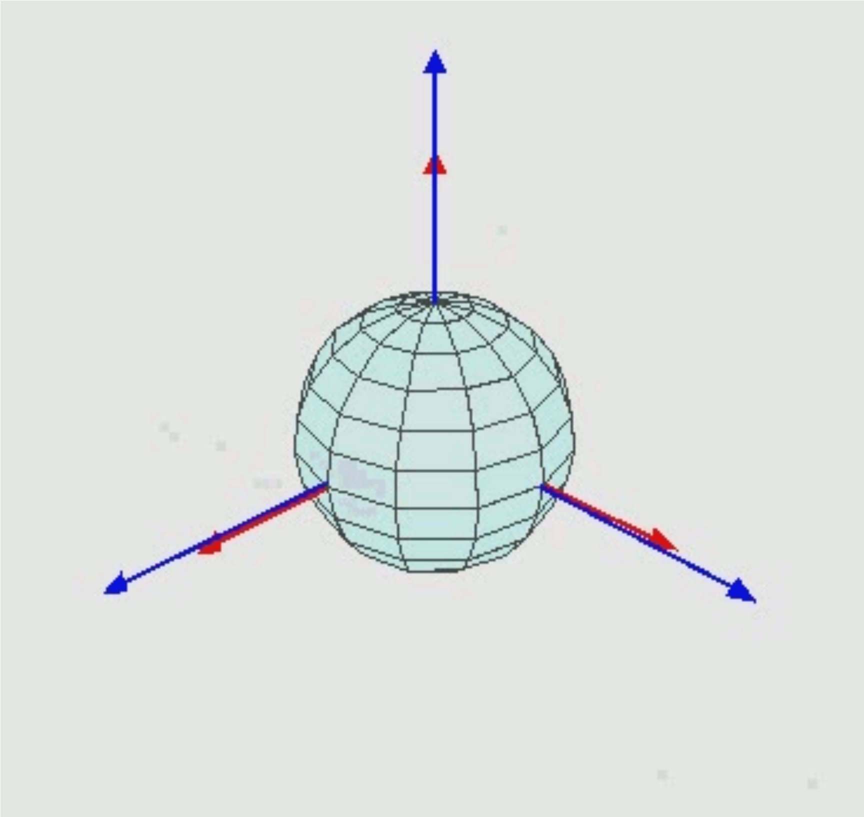
창원대학교 신소재공학부

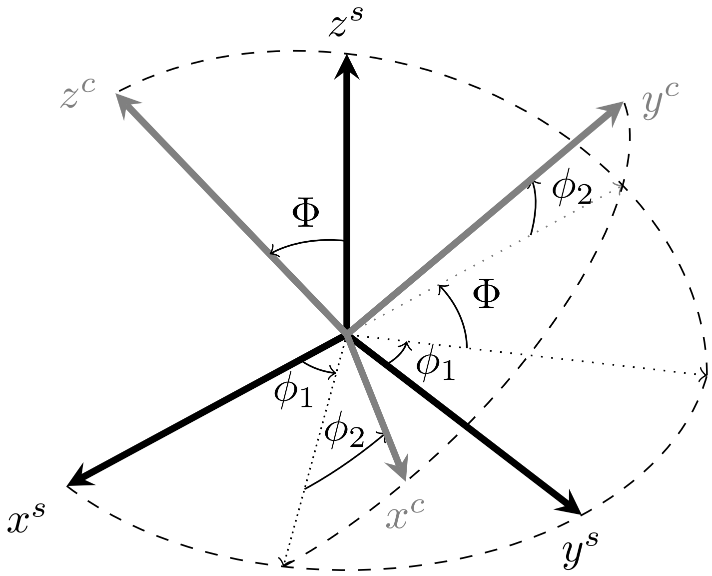
YJEONG@CHANGWON.AC.KR

연구실: #52-212 전화: 055-213-3694

HOMEPAGE: HTTP://YOUNGUNG.GITHUB.IO

Euler angles





References:
https://en.wikipedia.org/wiki/Euler_angles
https://youngung.github.io/euler/

Euler angle을 이용한 3차원 좌표 변환

- 두 삼차원 좌표계간의 관계를 표현하는 방법
- 여러 방법 중 Euler angle를 사용하는 방법이 MSE에서 자주 쓰인다.
  1. 한 3차원 좌표에 $\mathbf{e}_3$축 (z-axis)을 바라보며 시계 반대방향으로 ϕ_1 만큼 회전
  2. 다음으로 1.로 인해 회전된 좌표계의 $\mathbf{e}_1$축을 바라보며 시계 반대방향으로 Φ 만큼 회전
  3. 다음으로 1-2.로 인해 회전된 좌표계를 다시 $\mathbf{e}_3$ 축을 바라보며 시계 반대방향으로 ϕ_2 만큼 회전

$$\mathbf{R}^{\phi_1} = \begin{bmatrix} \cos\phi_1 & \sin\phi_1 & 0 \\ -\sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}^{\Phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix}$$

$$\mathbf{R}^{\phi_2} = \begin{bmatrix} \cos\phi_2 & \sin\phi_2 & 0 \\ -\sin\phi_2 & \cos\phi_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

위의 일련의 세 회전을 설명하는 '하나의' 좌표 변환 matrix를 다음을 통해 만들 수 있다.

$$\mathrm{R}_{ij} = \mathrm{R}^{c}_{ik}\mathrm{R}^{b}_{kl}\mathrm{R}^{a}_{lj} \qquad \mathbf{R} = \mathbf{R}^{c}\cdot\mathbf{R}^{b}\cdot\mathbf{R}^{a} \qquad \mathbf{R}\cdot\mathbf{v} = [\mathbf{R}^{c}\cdot\{\mathbf{R}^{b}\cdot(\mathbf{R}^{a}\cdot\mathbf{v})\}]$$

Recap: Einstein summation convention

Let’s practice #1

- Follow this link
- http://youngung.github.io/euler2ndtensor/
- You’ll find two links – one to open Google sheet another to download the sheet.

input	output											
		This excell sheet proves a means of coordinate system transformation										
			angle	radian								
		phi1	45	0.785								
	Three Euler angles	Phi	0	0.000								
		phi2	0	0.000								
						transformation matrix R				(transformation matrix)^t= R^t=R^-1		
	삼각 함수 값들											
	cos(phi1)	0.707	sin(phi1)	0.707		0.707	0.707	0.000		0.707	-0.707	0.000
	cos(Phi)	1.000	sin(Phi)	0.000		-0.707	0.707	0.000		0.707	0.707	0.000
	cos(phi2)	1.000	sin(phi2)	0.000		0.000	0.000	1.000		0.000	0.000	1.000
	2nd rank tensor in matrix form					R.T				R^t.R.T	**2nd rank tensor after coordinate transformation**	
	1	0	0			0.707	0.000	0.000		0.500	-0.500	0.000
	0	0	0			-0.707	0.000	0.000		-0.500	0.500	0.000
	0	0	0			0.000	0.000	0.000		0.000	0.000	0.000
	1st rank tensor (i.e., vector) in array form					R.v 1st rank tensor (vector) after coordinate transformation						
	1					0.70710678	-0.7071068	0				
	0											
	0											

Let’s practice #2

- At the bottom of the spread sheet you’ll find three separate matrices, which denote the three sequential rotation matrices.

$$\mathbf{R}^a = \begin{bmatrix} \cos\phi_1 & \sin\phi_1 & 0 \\ -\sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \quad \mathbf{R}^c = \begin{bmatrix} \cos\phi_2 & \sin\phi_2 & 0 \\ -\sin\phi_2 & \cos\phi_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Below is to obtain the transformation matrix by multipliying the three sequential simpler rotation matrices.

g1				g2				g3		
0.707	0.707	0.000		1.000	0.000	0.000		1.000	0.000	0.000
-0.707	0.707	0.000		0.000	1.000	0.000		0.000	1.000	0.000
0.000	0.000	1.000		0.000	0.000	1.000		0.000	0.000	1.000
g3 g2				g3g2g1						
1.000	0.000	0.000		0.707	0.707	0.000				
0.000	1.000	0.000		-0.707	0.707	0.000				
0.000	0.000	1.000		0.000	0.000	1.000				

- Of course, these are functions of phi1, Phi, phi2 values available at the top.

input	output					
		This excell sheet proves a means of coordinate system transformation				
			angle	radian		
		phi1	45	0.785		
	Three Euler angles	Phi	0	0.000		
		phi2	0	0.000		
					transformation matı	

Let’s practice #3

- Follow this link:
- http://youngung.github.io/tensors/

$$a'_i = R_{ij} a_j$$
$$\sigma'_{ij} = R_{ik} \sigma_{kl} R_{jl}$$
$$\mathbb{M}'_{ijkl} = R_{im} R_{jn} \mathbb{M}_{mnop} R_{ko} R_{lp}$$

- Tensor transformation rule is implemented into a Fortran code

Let’s practice #3 (Fortran)

```
program transform_vector
implicit none

dimension r(3,3), velocity_old(3), stress(3,3), velocity_new(3)
real*8 r, velocity_old, stress, th, velocity_new
integer i,j,k

!! the transformation matrix:
write(*,*)'Type: Rotation angle [in degree]:'
read(*,*) th
th = th* 3.141592 / 180. !! convert the degree to radian

r(:,:)=0.
r(1,1)=cos(th)
r(1,2)=sin(th)
r(2,1)=-sin(th)
r(2,2)=cos(th)
r(3,3)=1.

!! velocity
velocity_old(1)=30.
velocity_old(2)=0.
velocity_old(3)=0.

!! let's transform the velocity v`_i = r_ij v_j
do i=1,3
        velocity_new(i)=0.
do j=1,3
        velocity_new(i)=velocity_new(i)+r(i,j)*velocity_old(j)
enddo
enddo

!! print out the new velocity
write(*,*) 'old velocity'
write(*,'(3f5.1)') (velocity_old(i),i=1,3)
write(*,*) 'new velocity'
write(*,'(3f5.1)') (velocity_new(i),i=1,3)

end program transform_vector
```

변수 선언.

- E.g., R(3,3) is ‘real’ 실수, 그리고 (3,3) shape – 3x3 array

입력

- ’th’ 라는 변수에 user가 각도를 입력하면 radian 값으로 변환한다.

Transformation matrix

’th’ 라는 변수를 사용하여 3축을 잡고 ccw 회전시키는 transformation matrix를 만들어 변수 r에 저장

Velocity_old 변수 설정

Old coordinate system에 참조된 알고 있는 1차 텐서 velocity_old 변수를 설정 [30,0,0] array로 저장; ***1차 텐서는 벡터다**.

위 tensor를 변환하여 새로운 array에 저장

아래의 formula를 실행하여 1차 랭크 텐서 변환

$$v'_i = R_{ij} v_j$$

v 와 v`을 화면에 출력

Let's take a close look at the loop

```
do i=1,3
        velocity_new(i)=0.
do j=1,3
        velocity_new(i)=velocity_new(i)+r(i,j)*velocity_old(j)
enddo
enddo
```

1. In the above, each do-enddo pair
DO
ENDDO
allows you to form a loop: where integer i increases from 1 to 3, for each of which j increases from 1 to 3.

2. For instance, while i=1, you repeat
DO j=1,3
ENDDO

That means you perform

$$v_1^{new} = \sum_j^3 R_{1j} v_j^{old}$$

3. If you repeat Step 2 for i=2 and i=3 as well, you actually perform:

$$v_i^{new} = \sum_i^3 \sum_j^3 R_{ij} v_j^{old}$$

Remember that the above summation can be written short:

$$v_i^{new} = R_{ij} v_j^{old}$$

If you extend that idea for 2nd order tensor?

- Let's take an inverse approach for the 2nd order tensor transformation.
- We learned that the 2nd rank tensor transformation is done following the below rule:

$$\sigma'_{ij} = R_{ik}\sigma_{kl}R_{jl}$$

- The above can be implemented to a FORTRAN code such that

```
do i=1,3
do j=1,3
        s_new(i,j)=0.
do k=1,3
do l=1,3
        s_new(i,j)=s_new(i,j) + r(i,k)*s_old(k,l)*r(j,l)
enddo
enddo
enddo
enddo
```

- You might have been able to find certain rules that is applicable when you implement the tensor transformation. Also, you might have found the Einstein convention is very useful particularly when the formula is translated into FORTRAN code (how intuitive!).
- FORTRAN actually means 'FORMULA TRANSLATION'

Q. Extend that idea for 4th order tensor

- Within elastic region, metal follows Hooke's law which writes as below:
- $\sigma_{ij} = \mathbb{E}_{ijkl}\varepsilon_{kl}$
- (For advanced students) Can you write a short FORTRAN DO-ENDDO loop for the above operations?
- (For very advanced students; 신소재실험 학생들 필히 수행하세요…) Modify the source code available in the website and compile the code and run the code. You'll be able to find about the elastic modulus in other textbooks. Hint: you can reduce the above equation following Voigt's notation.

Let's practice #3 (Python)

```
import numpy as np
velocity_old=np.zeros(3)
velocity_new=np.zeros(3)
r=np.zeros((3,3))
velocity_old[0]=30.

th=raw_input('Type angle [in degree]: ')
th=np.pi*float(th)/180.

r[0,0]=np.cos(th)
r[0,1]=np.sin(th)
r[1,0]=-np.sin(th)
r[1,1]=np.cos(th)
r[2,2]=1.


## Apply v`_i = r_ij v_j
for i in xrange(3):
        for j in xrange(3):
                velocity_new[i]=velocity_new[i]+ \
        r[i,j]*velocity_old[j]

print 'old velocity'
print velocity_old
print 'new velocity'
print velocity_new
```

변수 선언.

- E.g., velocity_old와 velocity_new는 사이즈 3x1의 array
- R: 3x3 array;
- velocity_old 변수의 첫번째(0) element에 30 입력

입력

- 'th' 각도 입력한후 Radian값으로 변환

Transformation matrix

- 'th' 라는 변수를 사용하여 3축을 잡고 ccw 회전시키는 transformation matrix를 만들어 변수 r에 저장

위 tensor를 변환하여 새로운 array에 저장

- 아래의 formula를 실행하여 1차 랭크 텐서 변환
- $v_i' = R_{ij} v_j$

v 와 v`을 화면에 출력

Tensor and coordinate transformation

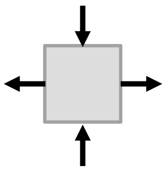
- Tensor is a method to represent physical quantities (and also some material properties).
- The physical quantity should remain the same even if you apply different coordinate system; The physical quantity should not be affected by the coordinate system of your own choice.
- But when you change the coordinate system, the values pertaining to individual components of the tensor change; That does not mean the associated property changes.
- The values of components that are changing w.r.t. coordinate system are used when you need quantification of associated physical quantity (or material property). That’s one of the reasons you should learn how to apply the coordinate transformation to tensors.

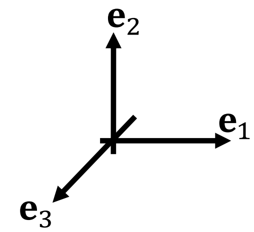
Example: pure shear

- Pure shear is a term referring to a stress (or strain) state where only shear components are non-zero.

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

I found the left is simple shear. Anything wrong with me?

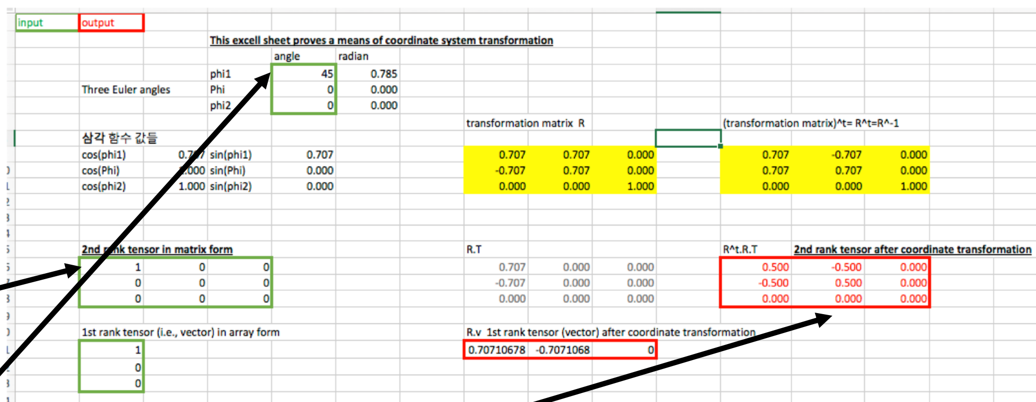




Let's check by using the spread sheet.

1. Put this value

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



input | output

This excell sheet proves a means of coordinate system transformation

		angle	radian
	phi1	45	0.785
Three Euler angles	Phi	0	0.000
	phi2	0	0.000

삼각 함수 값들

cos(phi1)	0.707	sin(phi1)	0.707
cos(Phi)	1.000	sin(Phi)	0.000
cos(phi2)	1.000	sin(phi2)	0.000

transformation matrix R

0.707	0.707	0.000
-0.707	0.707	0.000
0.000	0.000	1.000

(transformation matrix)^t= R^t=R^-1

0.707	-0.707	0.000
0.707	0.707	0.000
0.000	0.000	1.000

2nd rank tensor in matrix form

1	0	0
0	0	0
0	0	0

R.T

0.707	0.000	0.000
-0.707	0.000	0.000
0.000	0.000	0.000

R^t.R.T **2nd rank tensor after coordinate transformation**

0.500	-0.500	0.000
-0.500	0.500	0.000
0.000	0.000	0.000

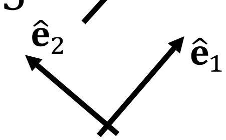
1st rank tensor (i.e., vector) in array form

1
0
0

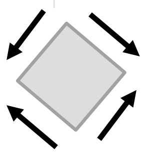
R.v 1st rank tensor (vector) after coordinate transformation

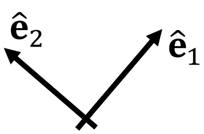
0.70710678	-0.7071068	0

2. Put $\phi_1 = 45\,°$
To obtain



3. Check the new tensor component values referred to the new coordinate system





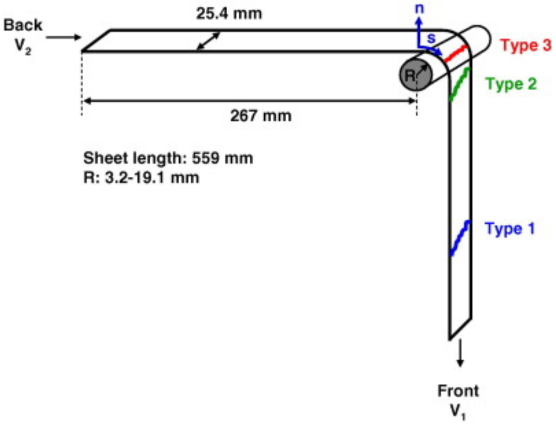
4. I wasn't wrong. With the new coordinate, the material is indeed under the pure shear condition!

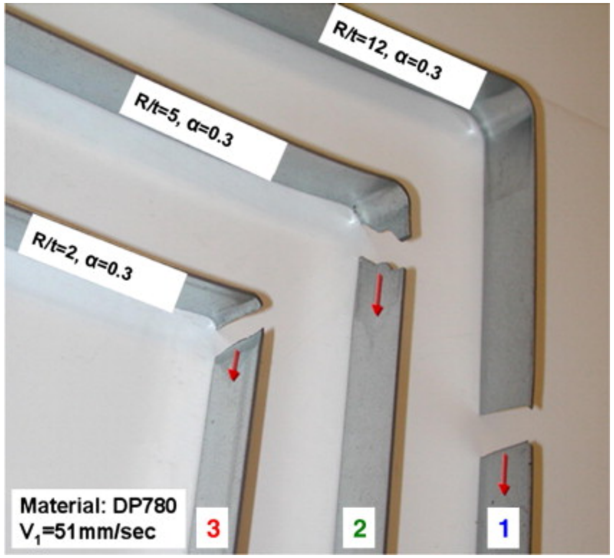
Example

- Elastic modulus ($\mathbb{E}$) is a 4th rank tensor and correlates the stress ($\boldsymbol{\sigma}$) and strain ($\boldsymbol{\varepsilon}$) in the elastic regime through
- $\boldsymbol{\sigma} = \mathbb{E}:\boldsymbol{\varepsilon}$
- Note that the colon symbol in the above denotes the double inner dot operation such that
- $\sigma_{ij} = \mathbb{E}_{ijkl}\varepsilon_{kl}$
- Q1. Express σ_{23} in the function of $\mathbb{E}$ and $\boldsymbol{\varepsilon}$ by explicitly denoting the indices of the associated tensors; Do not contract the expression by using Einstein's summation convention; Do not use the summation symbol.
- Q2. How many separate equations are hidden?

Where coordinate system transformation is required?

- Stretch bending test
- The failure criterion is usually written in terms of strain (or stress) state referred to the coordinate that is attached to the plane of the sheet metal.
- Here, as you can see, the region of specimen that eventually fractures, flows over the roller, during which it bends and ’rotates’.
- Therefore, you would want to ‘transform’ the stress state that was once referred to the global coordinate to the local coordinate system that ‘rotates’ together with the material.

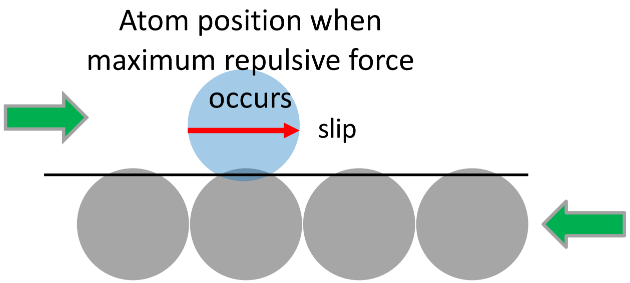


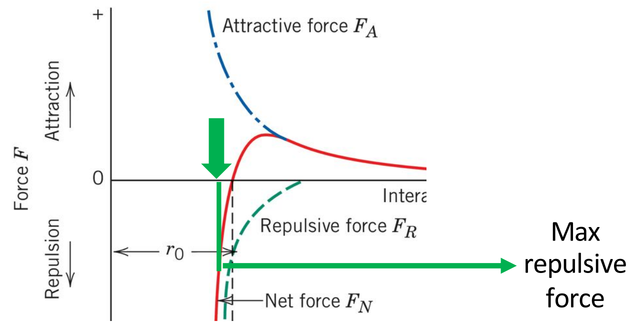


JH Kim et al, IJP 27, 2011

Where coordinate system transformation is required?

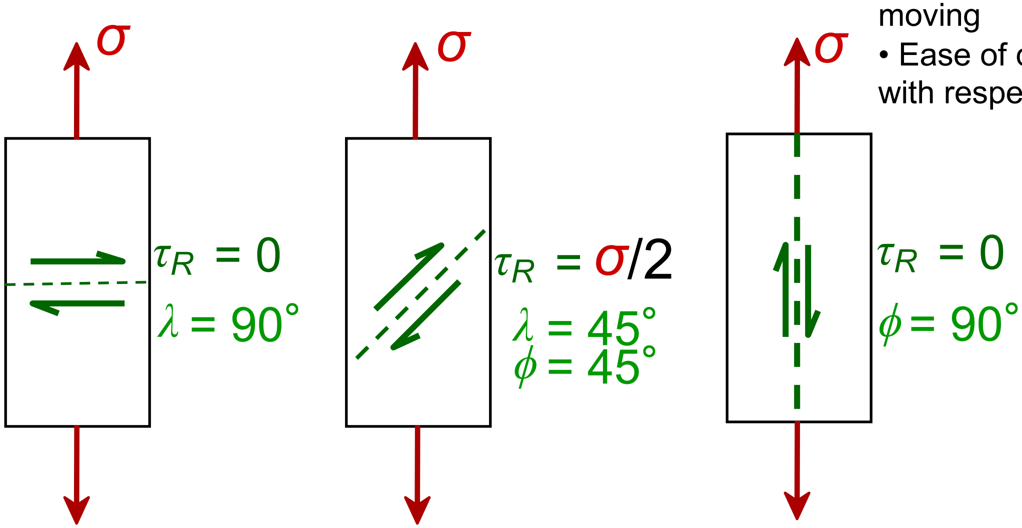
Critical Resolved Shear Stress





For dislocation to slip, this max. force should be overcome

Max repulsive force is closely related with the CRSS



- Condition for dislocation motion (= condition for plastic yielding): If RSS reaches a certain (critical) value, the dislocation will start moving
- Ease of dislocation motion depends on crystallographic orientation with respect to the external loading direction

$$\tau_{RSS} = \sigma \cos\lambda \cos\phi$$

$\cos\lambda \cos\phi$: Schmid's (orientation) factor

Dislocation slip condition (≈ atomic yield condition)

$$\tau_{RSS} \geq \tau_{CRSS}$$

Example: yield of single crystal

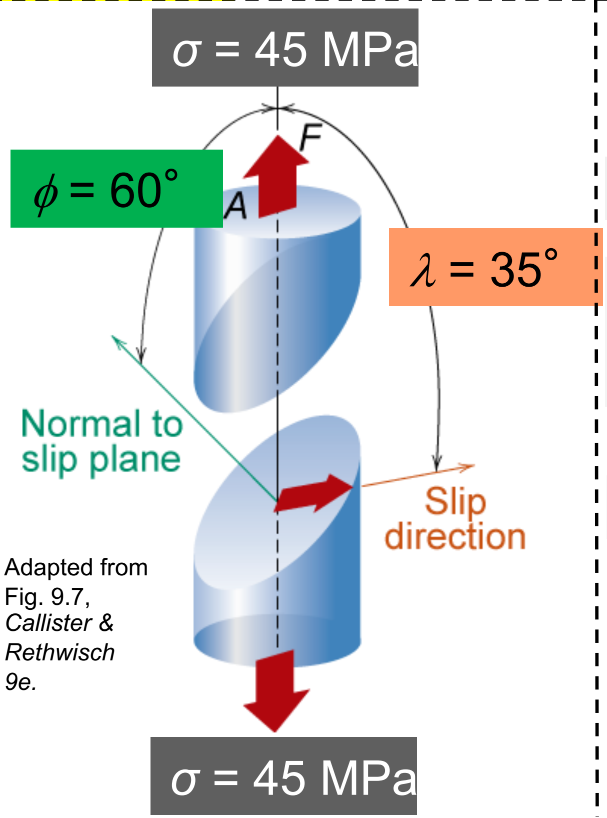
a) Will the single crystal yield?
b) If not, what stress is needed?

$\tau_{RSS} = \sigma \cos\lambda \cos\phi$

We learned this equation that correlates the external loading (σ) and the orientation of slip system (λ, ϕ).

Condition 1. External load of 45 MPa

Condition 2. Slip system characterized by $\lambda = 35°, \phi = 60°$



Adapted from Fig. 9.7, *Callister & Rethwisch 9e.*

Condition for dislocation to slip? → $\tau_{RSS} \geq \tau_{CRSS}$

Condition 1. $\tau_{CRSS} = \boxed{20.7}$ MPa

Condition 2. $\tau_{RSS} = \sigma \cos\lambda \cos\phi$

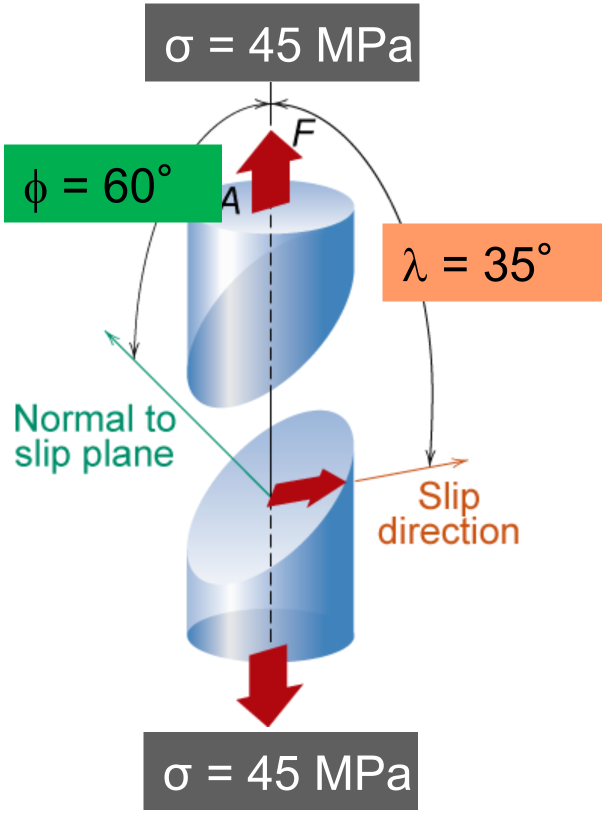
$= 45 \cos 35° \cos 60°$ [MPa]

$\approx 45 \times 0.819 \times 0.5 \approx \boxed{18.4}$ [MPa]

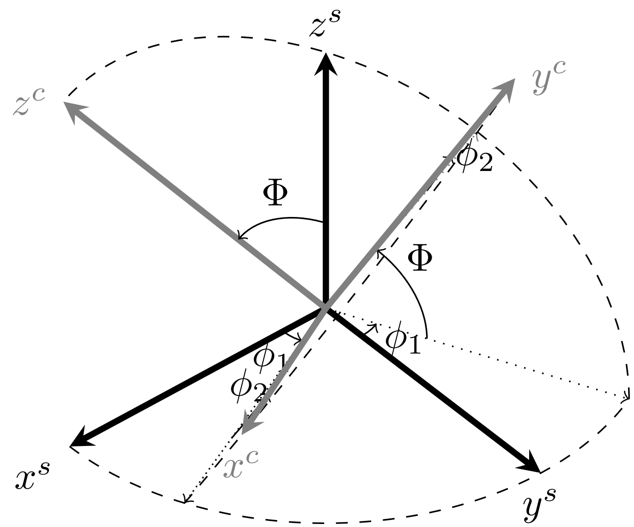
Check $\boxed{\tau_{RSS}} \geq \boxed{\tau_{CRSS}}$

45 MPa is not sufficient enough to cause this slip system ($\lambda = 35°, \phi = 60°$) to slip (yield)

Transformation for CRSS



$$\phi_1 = 25°, \Phi = 60°, \phi_2 = 19°$$



This gives the transformation matrix like:

0.788	0.547	0.282
-0.495	0.291	0.819
0.366	-0.785	0.500

If you transform

0	0	0
0	0	0
0	0	45

You'll get

3.577	10.389	6.344
10.389	30.173	18.424
6.344	18.424	11.250