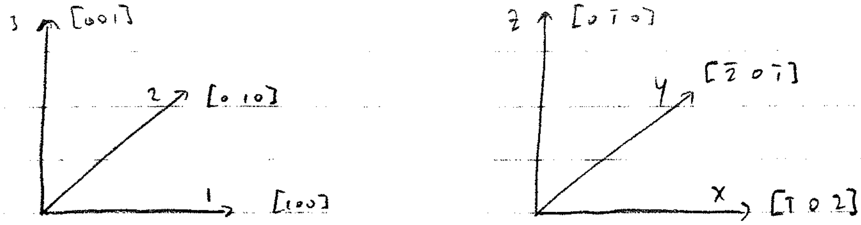
(iii)



$$z = x \times y$$

$$= \begin{bmatrix} i & j & k \\ -1 & 0 & 2 \\ -2 & 0 & -1 \end{bmatrix} = 0i - 5j + 0k$$

$$R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{matrix} x & \quad y & \quad z \end{matrix} \\ \begin{bmatrix} -1/\sqrt{5} & -2/\sqrt{5} & 0 \\ 0 & 0 & -1 \\ 2/\sqrt{5} & -1/\sqrt{5} & 0 \end{bmatrix} \end{array}$$

$x : [-1/\sqrt{5} \;\; 0 \;\; 2/\sqrt{5}]$

$y : [-2/\sqrt{5} \;\; 0 \;\; -1/\sqrt{5}]$

$z : [0 \;\; -1 \;\; 0]$

Shear: $[\sigma] = \begin{bmatrix} 0 & \sigma_{xy} & 0 \\ \sigma_{xy} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$$\sigma^{123} = R\, \sigma^{xyz} R^T$$

$$= \begin{bmatrix} -1/\sqrt{5} & -2/\sqrt{5} & 0 \\ 0 & 0 & -1 \\ 2/\sqrt{5} & -1/\sqrt{5} & 0 \end{bmatrix} \begin{bmatrix} 0 & \sigma_{xy} & 0 \\ \sigma_{xy} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1/\sqrt{5} & 0 & 2/\sqrt{5} \\ -2/\sqrt{5} & 0 & -1/\sqrt{5} \\ 0 & -1 & 0 \end{bmatrix}$$

$$= \sigma_{xy} \begin{bmatrix} -2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 0 & 0 & 0 \\ -1/\sqrt{5} & 2/\sqrt{5} & 0 \end{bmatrix} \begin{bmatrix} -1/\sqrt{5} & 0 & 2/\sqrt{5} \\ -2/\sqrt{5} & 0 & -1/\sqrt{5} \\ 0 & -1 & 0 \end{bmatrix}$$

$$= \sigma_{xy} \begin{bmatrix} 4/5 & 0 & -3/5 \\ 0 & 0 & 0 \\ -3/5 & 0 & -4/5 \end{bmatrix}$$

$$\underset{\sim}{\varepsilon} = \underset{\approx}{S}\,\underset{\sim}{\sigma}$$

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & S_{15} & 0 \\ S_{12} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ S_{13} & S_{23} & S_{33} & 0 & S_{35} & 0 \\ 0 & 0 & 0 & S_{44} & 0 & S_{46} \\ S_{15} & S_{25} & S_{35} & 0 & S_{55} & 0 \\ 0 & 0 & 0 & S_{46} & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} 4/5\,\sigma_{xy} \\ 0 \\ -4/5\,\sigma_{xy} \\ 0 \\ -3/5\,\sigma_{xy} \\ 0 \end{Bmatrix}$$

$$\varepsilon_{11} = \left[\frac{4S_{11}}{5} - 4/5\,S_{13} - 3/5\,S_{15}\right]\sigma_{xy} \checkmark$$

$$\varepsilon_{22} = \left[4/5\,S_{12} - 4/5\,S_{23} - 3/5\,S_{25}\right]\sigma_{xy}$$

$$\varepsilon_{33} = \left[4/5\,S_{13} - 4/5\,S_{33} - 3/5\,S_{35}\right]\sigma_{xy} \checkmark$$

$$\gamma_{23} = 0$$

$$\varepsilon_{13} = \frac{1}{2}\gamma_{13} = \left[4/5\,S_{15} - 4/5\,S_{35} - 3/5\,S_{55}\right]\sigma_{xy} \checkmark$$

$$\gamma_{12} = 0$$

$$\varepsilon^{xyz} = R^{T}\,\varepsilon^{123}\,R$$

MATLAB $\rightarrow$

$\leftarrow$ should include your code in HW for me to check! Few steps are left out here.

$$\varepsilon_{xy} = 0.0258\,\sigma_{xy}$$

$$\frac{\gamma_{xy}}{2} = 0.0258\,\sigma_{xy}$$

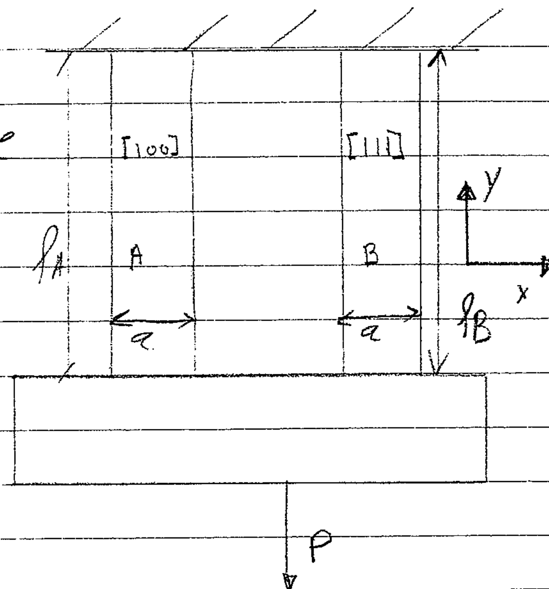
$$\gamma_{xy} = 0.0516\,\sigma_{xy}$$

$$\boxed{\sigma_{xy}/\gamma_{xy} = 19.38\ \text{GPa}}$$

$\approx$

1

Question 1



⇒ hypothetical composite material composed of 2 Fcc grains of iron

⇒ Isostrain stretching in the Y direction.

⇒ $E_{[100]} \neq E_{[111]}$

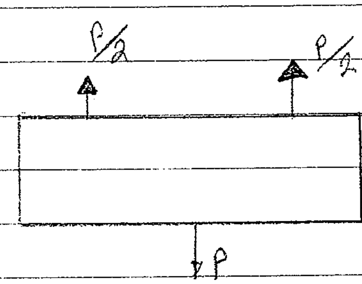
⇒ critical resolved shear stress of Fcc iron 40 MPa

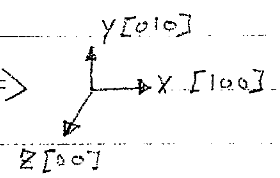
⇒ Which bar yields first?

⇒ Find stresses in A & B when yielding is first detected.

⇒ Isostrain conditions ⇒ $\Delta l_A = \Delta l_B$ + since $l_A = l_B$

⇒ $\boxed{\varepsilon y^A = \varepsilon y^B = \varepsilon_y}$



⇒ 

⇒ For bar A ⇒ Tension in the Y direction ⇒ Tension in [100] direction for bar A

$E_{[100]_A} = \frac{\sigma_A}{\varepsilon y^A} \Rightarrow \sigma_A = \varepsilon y^A E_{[100]A} = \varepsilon_y E_{[100]}$

⇒ For bar B ⇒ Tension in Y-direction → Tension in [111] direction

For bar b ⇒ $E_{[111]_B} = \frac{\sigma_B}{\varepsilon y^B} \Rightarrow \sigma_B = E_{[111]} \varepsilon y^B = \varepsilon_y E_{[111]}$.

⇒ Find Schmid Factors For Bar A

⇒ Loading direction = [100]

⇒ $M = \underbrace{\cos \phi}_{\text{angle between slip plane normal + tensile axis}} \underbrace{\cos \lambda}_{\text{angle between slip direction + tensile axis}}$

⇒ FCC Iron ⇒ 12 slip systems.

system 1: $M_{(111)[1\bar{1}0]} = \underbrace{\frac{(1*1)+(1*0)+(1*0)}{(1^2+1^2+1^2)^{1/2}(1^2)^{1/2}}}_{\cos\phi} * \underbrace{\frac{(1*1)+(-1*0)+(0*0)}{(1^2+-1^2+0^2)^{1/2}(1^2)^{1/2}}}_{\cos\lambda}$

$= \frac{1}{\sqrt{3}} \times \frac{1}{\sqrt{2}} = \frac{1}{\sqrt{6}}$

⇒ Similarly using MatLab (see page 5)

2) $M(111)[10\bar{1}] = 1/\sqrt{6}$
3) $M(111)[01\bar{1}] = 0$
4) $M(\bar{1}11)[101] = -1/\sqrt{6}$
5) $M(\bar{1}11)[01\bar{1}] = 0$
6) $M(\bar{1}11)[110] = -1/\sqrt{6}$
7) $M(1\bar{1}1)[101] = -1/\sqrt{6}$
8) $M(1\bar{1}1)[011] = 0$
9) $M(1\bar{1}1)[110] = 1/\sqrt{6}$
10) $M(11\bar{1})[1\bar{1}0] = 1/\sqrt{6}$
11) $M(11\bar{1})[101] = 1/\sqrt{6}$
12) $M(11\bar{1})[011] = 0$

max $= \frac{1}{\sqrt{6}}$ on systems

1, 2, 4, 6, 7, 9, 10, 11

⇒ $\tau_c = \sigma_{y_A} M_{max_A}$

$40\,MPa = \sigma_{y_A} \left(\frac{1}{\sqrt{6}}\right)$

σ_{y_A} 98 MPa

⇒ For Bar B.

* Loading direction = [1 1 1]

$M_1 = 0$
$M_2 = 0$
$M_3 = 0$
$M_4 = 2/3\sqrt{6}$
$M_5 = 0$
$M_6 = 2/3\sqrt{6}$
$M_7 = 0$
$M_8 = 2/3\sqrt{6}$
$M_9 = 2/3\sqrt{6}$
$M_{10} = 0$
$M_{11} = 2/3\sqrt{6}$
$M_{12} = 2/3\sqrt{6}$

$max = \frac{2}{3\sqrt{6}}$ on systems

4, 6, 8, 9, 11 + 12

$\tau_c = \sigma_{y_B} M_{max\,B}$

$40 = \sigma_{y_B} \left(\frac{2}{3\sqrt{6}} \right)$

$\boxed{\sigma_{y_B} = 147 \text{ MPa}}$

$\Rightarrow \sigma_A = E_{[100]} \varepsilon_y \qquad \sigma_B = E_{[111]} \varepsilon_y$

$\frac{\sigma_B}{\sigma_A} = \frac{E_{[111]} \varepsilon_y}{E_{[100]} \varepsilon_y} = \frac{E_{[111]}}{E_{[100]}} = 2.14$

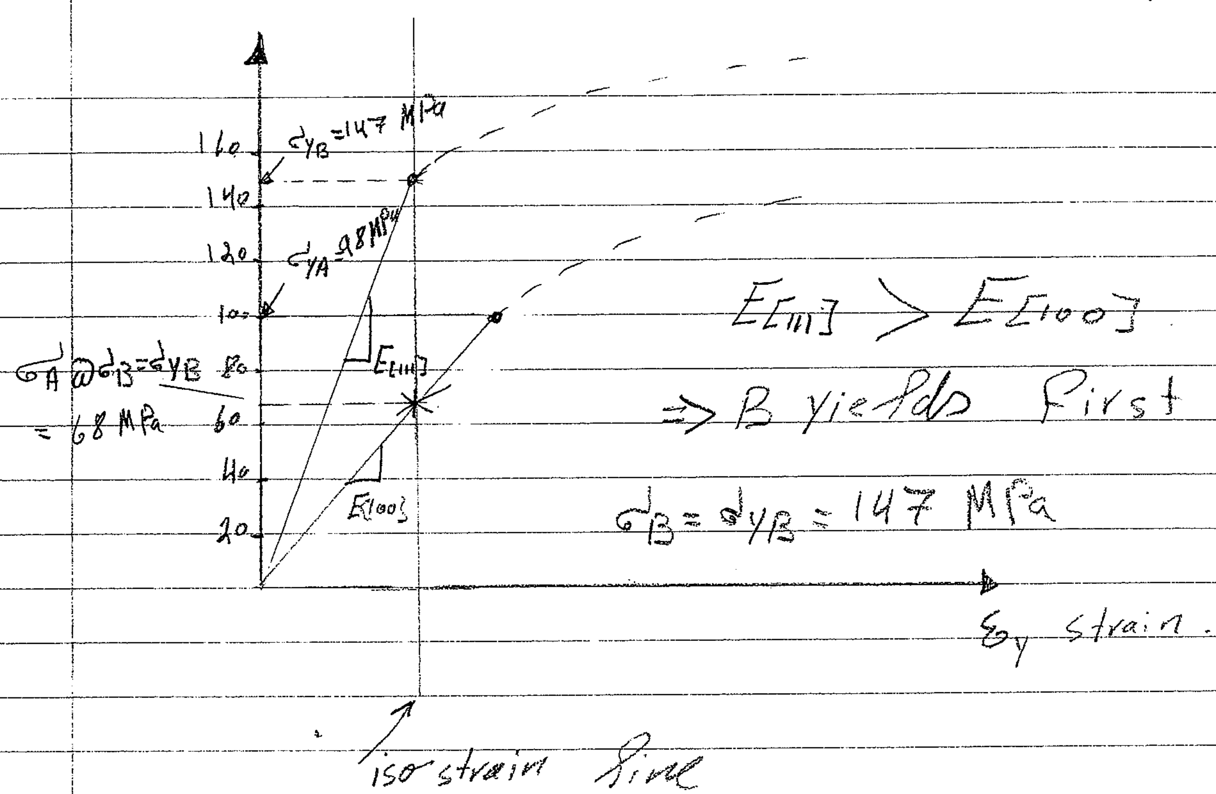
$\Rightarrow \frac{\sigma_{y_B}}{\sigma_{y_A}} = \frac{147}{98} = 1.5 < 2.14 \Rightarrow$ ~~so~~ bar B will yield first.

$\Rightarrow \sigma_B = \sigma_{y_B} = 147 \text{ MPa} \Rightarrow \frac{\sigma_B}{\sigma_A} = 2.14 \Rightarrow \sigma_A = \frac{\sigma_B}{2.14} = \frac{147}{2.14}$

$\sigma_A \doteq 68.7 < \sigma_{yA}$

⇒ stresses in Y direction when yielding is detected ⇒ $\sigma_B = \sigma_{y_B} = 147$

$\sigma_A = 68.7$ MPa



$E_{[111]} > E_{[100]}$

$\Rightarrow$ B yields first

$\sigma_B = \sigma_{YB} = 147$ MPa

when $\sigma_B = \sigma_{BY} = 147$ MPa.

$\sigma_A = 68.7$ MPa ✓

```
m = zeros(12,1);
LD = input('Loading Direction (Miller Indices Notation with [100] RD):   ');
LDC_UV=normr(LD); % Convert to Unit Vector

   n1=normr([1 1 1]); % 1st normal vector
   n2=normr([-1 1 1]); % 2nd normal vector
   n3=normr([1 -1 1]);% 3rd normal vector
   n4=normr([1 1 -1]);% 4th normal vector
   normal=[n1;n2;n3;n4];

   % Define The Slip Directions for each Slip Plane (FCC Crystal)
   b11=normr([1 -1 0]); % 1st burger vector accoiated with n1
   b12=normr([1 0 -1]);% 2nd burger vector accoiated with n1
   b13=normr([0 1 -1]);% 3rd burger vector accoiated with n1

   b21=normr([1 0 1]); % 1st burger vector accoiated with n2
   b22=normr([0 1 -1]);% 2nd burger vector accoiated with n2
   b23=normr([1 1 0]);% 3rd burger vector accoiated with n2

   b31=normr([-1 0 1]); % 1st burger vector accoiated with n3
   b32=normr([0 1 1]);% 2nd burger vector accoiated with n3
   b33=normr([1 1 0]);% 3rd burger vector accoiated with n3

   b41=normr([1 -1 0]); % 1st burger vector accoiated with n4
   b42=normr([1 0 1]);% 2nd burger vector accoiated with n4
   b43=normr([0 1 1]);% 3rd burger vector accoiated with n4
   burger=[b11;b12;b13;b21;b22;b23;b31;b32;b33;b41;b42;b43];

          w=1;
      for j=1:4;  % Loop for the 4 Slip Planes
          Cos_theta=dot(LDC_UV, normal(j,:));
          for k=1:3; % Loop for the three slip systems
              if j==1;
                   Cos_lamda=dot(LDC_UV, burger(j+k-1,:));
                else if j==2;
                Cos_lamda=dot(LDC_UV, burger(j+k+1,:));
                    else if j==3;
                    Cos_lamda=dot(LDC_UV, burger(j+k+3,:));
                        else if j==4;
                        Cos_lamda=dot(LDC_UV, burger(j+k+5,:));
                            end
                        end
                    end
              end
%               m(w,1) =abs(Cos_theta*Cos_lamda);
                m(w,1) =(Cos_theta*Cos_lamda);
              w=w+1;
          end
      end
```

Good to have a program!

$$C'_{ijkl} = C_{pqrs} Q_{ip} Q_{jq} Q_{kr} Q_{ls}$$

in voigt notation

$$C' = \begin{bmatrix} 306 & 155 & 104 & 0 & 38 & 0 \\ 155 & 255 & 155 & 0 & 0 & 0 \\ 104 & 155 & 306 & 0 & -38 & 0 \\ 0 & 0 & 0 & 129 & 0 & 0 \\ 38 & 0 & -38 & 0 & 78 & 0 \\ 0 & 0 & 0 & 0 & 0 & 129 \end{bmatrix}$$

```
Q = [-2/sqrt(5) 0 1/sqrt(5)
     0 1 0
     -1/sqrt(5) 0 -2/sqrt(5)];

C_nom = zeros(6,6);

C_nom(1,1)=223;

C_nom(1,2)=129;

C_nom(1,3)=99;

C_nom(1,5)=27;

C_nom(2,2)=241;

C_nom(2,3)=125;

C_nom(2,5)=-9;

C_nom(3,3)=200;

C_nom(3,5)=4;

C_nom(4,4)=76;

C_nom(4,6)=-4;

C_nom(5,5)=21;

C_nom(6,6)=77;


C=zeros(3,3,3,3);

C11=C_nom(1,1);

C12=C_nom(1,2);

C13=C_nom(1,3);

C15=C_nom(1,5);

C22=C_nom(2,2);

C23=C_nom(2,3);

C25=C_nom(2,5);

C33=C_nom(3,3);
```

```
C35=C_nom(3,5);

C44=C_nom(4,4);

C46=C_nom(4,6);

C55=C_nom(5,5);

C66=C_nom(6,6);

C(1,1,1,1)=C11;

C(1,1,2,2)=C12;

C(2,2,1,1)=C12;

C(1,1,3,3)=C13;

C(3,3,1,1)=C13;

C(1,1,1,3)=C15;

C(1,1,3,1)=C15;

C(1,3,1,1)=C15;

C(3,1,1,1)=C15;

C(2,2,2,2)=C22;

C(2,2,3,3)=C23;

C(3,3,2,2)=C23;

C(2,2,1,3)=C25;

C(2,2,3,1)=C25;

C(1,3,2,2)=C25;

C(3,1,2,2)=C25;

C(3,3,3,3)=C33;

C(3,3,1,3)=C35;

C(3,3,3,1)=C35;

C(1,3,3,3)=C35;
```

```
C(3,1,3,3)=C35;

C(2,3,2,3)=C44;

C(2,3,3,2)=C44;

C(3,2,3,2)=C44;

C(3,2,2,3)=C44;

C(2,3,1,2)=C46;

C(2,3,2,1)=C46;

C(3,2,2,1)=C46;

C(3,2,1,2)=C46;

C(1,2,2,3)=C46;

C(1,2,3,2)=C46;

C(2,1,2,3)=C46;

C(2,1,3,2)=C46;

C(1,3,1,3)=C55;

C(1,3,3,1)=C55;

C(3,1,1,3)=C55;

C(3,1,3,1)=C55;

C(1,2,1,2)=C66;

C(1,2,2,1)=C66;

C(2,1,2,1)=C66;

C(2,1,1,2)=C66;

%Barnett-Lothe formulation to compute interface dislocation fields

%Cvoigt=[C11 C12 C13 0 C15 0;C12 C22 C23 0 C25 0;C13 C23 C33 0 C35 0;0
0 0 C44 0 C46;C15 C25 C35 0 C55 0;0 0 0 C46 0 C66];

%Cvoigt-transpose(Cvoigt) %Checking symmetry
```

```
Ctransf=zeros(3,3,3,3);

   for a=1:1:3

       for b=1:1:3

           for c=1:1:3

               for d=1:1:3

                   for i=1:1:3

                       for j=1:1:3

                           for k=1:1:3

                               for l=1:1:3

                                   Ctransf(a,b,c,d)=Ctransf(a,b,c,d)
+C(i,j,k,l)*Q(i,a)*Q(j,b)*Q(k,c)*Q(l,d);

                                   %The columns of Q are the new
vectors

                               end

                           end

                       end

                   end

               end

           end

       end

   end

%Check with Voigt notation

Ctrvoigt=zeros(6,6);

Ctrvoigt(1,1)=Ctransf(1,1,1,1);

Ctrvoigt(1,2)=Ctransf(1,1,2,2);

Ctrvoigt(1,3)=Ctransf(1,1,3,3);
```

```
Ctrvoigt(1,4)=Ctransf(1,1,2,3);
Ctrvoigt(1,5)=Ctransf(1,1,1,3);
Ctrvoigt(1,6)=Ctransf(1,1,1,2);
Ctrvoigt(2,1)=Ctransf(2,2,1,1);
Ctrvoigt(2,2)=Ctransf(2,2,2,2);
Ctrvoigt(2,3)=Ctransf(2,2,3,3);
Ctrvoigt(2,4)=Ctransf(2,2,2,3);
Ctrvoigt(2,5)=Ctransf(2,2,1,3);
Ctrvoigt(2,6)=Ctransf(2,2,1,2);
Ctrvoigt(3,1)=Ctransf(3,3,1,1);
Ctrvoigt(3,2)=Ctransf(3,3,2,2);
Ctrvoigt(3,3)=Ctransf(3,3,3,3);
Ctrvoigt(3,4)=Ctransf(3,3,2,3);
Ctrvoigt(3,5)=Ctransf(3,3,1,3);
Ctrvoigt(3,6)=Ctransf(3,3,1,2);
Ctrvoigt(4,1)=Ctransf(2,3,1,1);
Ctrvoigt(4,2)=Ctransf(2,3,2,2);
Ctrvoigt(4,3)=Ctransf(2,3,3,3);
Ctrvoigt(4,4)=Ctransf(2,3,2,3);
Ctrvoigt(4,5)=Ctransf(2,3,1,3);
Ctrvoigt(4,6)=Ctransf(2,3,1,2);
Ctrvoigt(5,1)=Ctransf(1,3,1,1);
Ctrvoigt(5,2)=Ctransf(1,3,2,2);
Ctrvoigt(5,3)=Ctransf(1,3,3,3);
Ctrvoigt(5,4)=Ctransf(1,3,2,3);
```

```
Ctrvoigt(5,5)=Ctransf(1,3,1,3);

Ctrvoigt(5,6)=Ctransf(1,3,1,2);

Ctrvoigt(6,1)=Ctransf(1,2,1,1);

Ctrvoigt(6,2)=Ctransf(1,2,2,2);

Ctrvoigt(6,3)=Ctransf(1,2,3,3);

Ctrvoigt(6,4)=Ctransf(1,2,2,3);

Ctrvoigt(6,5)=Ctransf(1,2,1,3);

Ctrvoigt(6,6)=Ctransf(1,2,1,2);

%Ctrvoigt-transpose(Ctrvoigt)

%Transform the elas tic constants
```