Programming Clinic Four

MEG 324 SSG 321 Continuum Mechanics

Mechanical & Systems Engineering, University of Lagos

Star Problem

- We know that a tensor transforms a vector to another vector linearly. In addition to this, we are aware of a number of special tensors:
  - An Identity Tensor does nothing.
  - A Spherical Tensor simply elongates the vector or contracts it without changing direction
  - A Rotatation Tensor simply rotates the vector without changing its magnitude.

Arbitrary Transformation

- The star challenge for this week's programming clinic is to use these known attributes to create a tensor that can transform one vector to another.
- You will choose the inpur vector, choose the output vector and use that information to create the transforming tensor that will take the input abd produce the output.

My Code

```
a=List[1.,-5.,4.];aa=Normalize[a];
b=List[3.,15.,10.];bb=Normalize[b];
S1=(Norm[b]/Norm[a])IdentityMatrix[3];
S2=(Norm[a]/Norm[b])IdentityMatrix[3];
θ=ArcCos[Dot[aa,bb]];
W=-LeviCivitaTensor[3] . Normalize[Cross[aa,bb]];
Q[x_Real]:=IdentityMatrix[3]+W  Sin[x] +W . W (1-Cos[x]);
S1.Q[θ].a
S2.Transpose[Q[θ]].b
```

Your Preparation Step One

- My code is already working. Don’t waste your time asking for any help, Type it in and keep correcting your typos until it runs on your system
- You can see that I have included two vectors of my own. Use your own vectors. Confirm that the tensors generated do what we intend them to do.

Your Preparation Step Two

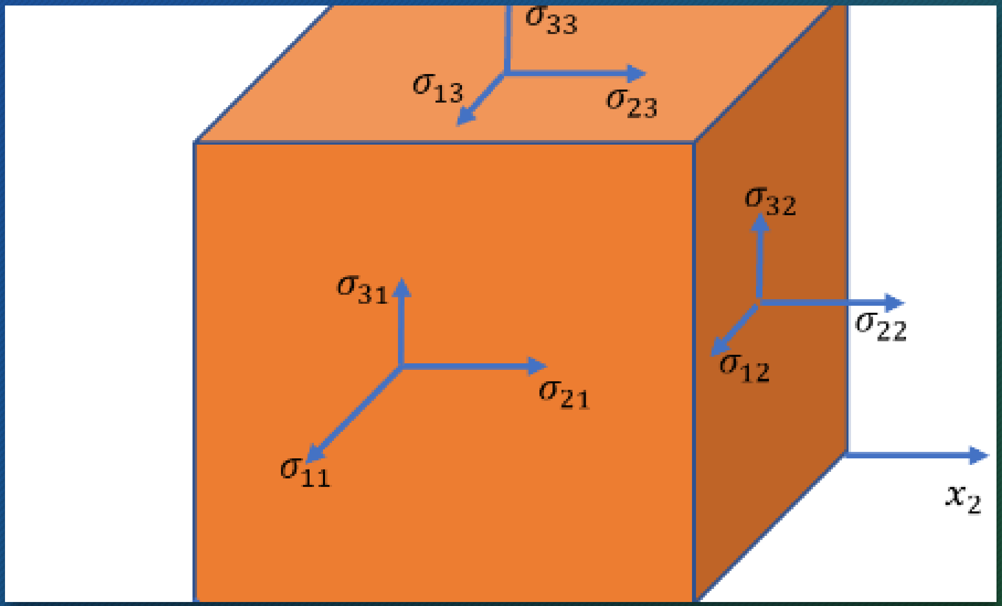
- Understand each step of the code.
- Be prepared to be asked to explain why this program is doing what we intend it to do.
- In particular, Test and see that for any value of θ the tensor $Q[\theta]$ is a rotation tensor.
- Display the tensors $\mathbf{S}_1\mathbf{Q}[\theta]$ as well as $\mathbf{S}_2\mathbf{Q}^{\mathrm{T}}[\theta]$. What are they really doing? How are they related.

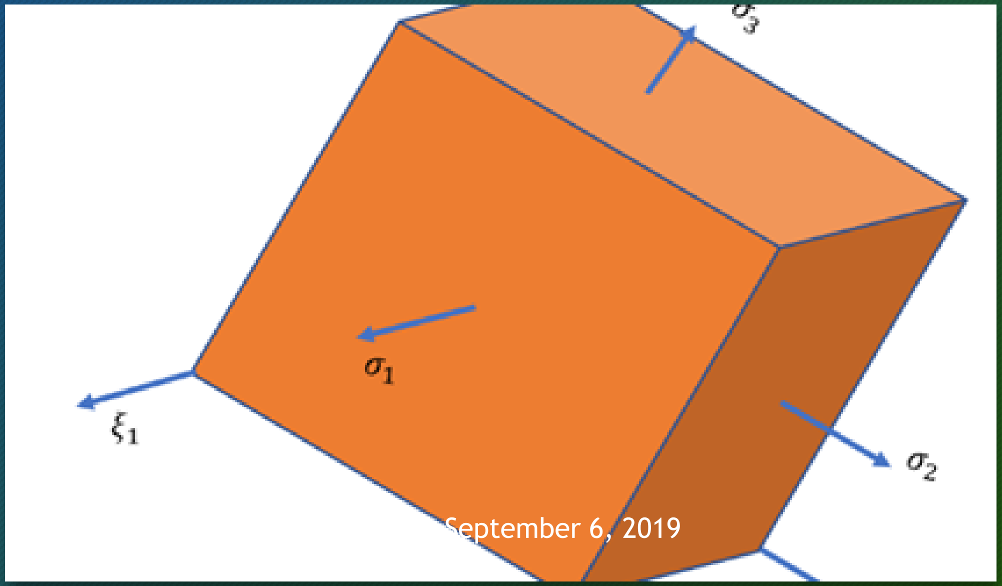
Your Preparation Step Three

- If we started with the tensor $\mathbf{H} = \mathbf{S}_1\mathbf{Q}[\theta]$, Could we actually create a multiplicative decomposition that will break the tensor into one part that can rotate as well as the other part that can stretch?
- Is this decomposition unique?
- Can you guess any practical use of this work?
- Do you think what we are doing here has any relationship to the eigenvectors of the tensor?

Stresses at Angular Planes

```
CauchyStr = {{σx, τxy, 0}, {τxy, σy, 0}, {0, 0, σz}};
e1 = {1, 0, 0};
e2 = {0, 1, 0};
e3 = {0, 0, 1};
er[α_] := Cos[α] e1 + Sin[α] e2;
eθ[α_] := -Sin[α] e1 + Cos[α] e2;
ez[α_] := e3;
Rot[α_] := TensorProduct[e1, er[α]] + TensorProduct[e2,
Rot[θ]
{{Cos[θ], Sin[θ], 0}, {-Sin[θ], Cos[θ], 0}, {0, 0, 1}}
```





- rr

```
RotStr[θ] = Rot[θ].CauchyStr.Transpose[Rot[θ]]

{{Sin[θ] (Sin[θ] σy + Cos[θ] τxy) + Cos[θ] (Cos[θ] σx + Sin[θ] τxy),
  Cos[θ] (Sin[θ] σy + Cos[θ] τxy) - Sin[θ] (Cos[θ] σx + Sin[θ] τxy), 0
 {Cos[θ] (-Sin[θ] σx + Cos[θ] τxy) + Sin[θ] (Cos[θ] σy - Sin[θ] τxy),
  -Sin[θ] (-Sin[θ] σx + Cos[θ] τxy) + Cos[θ] (Cos[θ] σy - Sin[θ] τxy)

MatrixForm[Simplify[%]]
```

xForm=

$$\begin{pmatrix} \cos^2(\theta)\,\sigma_x + \sin(2\,\theta)\,\tau_{xy} + \sin^2(\theta)\,\sigma_y & \frac{1}{2}\left(-\sin(2\,\theta)\,\sigma_x + 2\cos(2\,\theta)\,\tau_{xy} + \sin(2\,\theta)\,\sigma_y\right) & 0 \\ \frac{1}{2}\left(-\sin(2\,\theta)\,\sigma_x + 2\cos(2\,\theta)\,\tau_{xy} + \sin(2\,\theta)\,\sigma_y\right) & \sin^2(\theta)\,\sigma_x + \cos(\theta)\left(\cos(\theta)\,\sigma_y - 2\sin(\theta)\,\tau_{xy}\right) & 0 \\ 0 & 0 & \sigma_z \end{pmatrix}$$

$$\boldsymbol{\sigma} = (\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3)\begin{pmatrix} 3 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix} \otimes \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{pmatrix}$$

Find principal stresses, principal directions and find the traction vector on a plane whose unit normal is $\frac{1}{\sqrt{2}}(0,1,1)$. Also rotate the stress tensor back to its principal directions

```
CauchyStr = {{3, 1, 1}, {1, 0, 2}, {1, 2, 0}};
NorVec = 1 / Sqrt[2] {0, 1, 1};
TracVec = CauchyStr.NorVec
```

$\{\sqrt{2}, \sqrt{2}, \sqrt{2}\}$

```
Eigenvalues[CauchyStr]
```

{4, -2, 1}

```
vecs = Eigenvectors[CauchyStr]
```

{{2, 1, 1}, {0, -1, 1}, {-1, 1, 1}}

```
e1 = {1., 0, 0}; e2 = {0, 1., 0}; e3 = {0., 0, 1.};
Q = TensorProduct[e1, Normalize[vecs[[1]]]] + TensorProduct[e2, Normalize[vecs[[2]]]] +
  TensorProduct[e3, Normalize[vecs[[3]]]]
```

{{0.816497, 0.408248, 0.408248}, {0., -0.707107, 0.707107}, {-0.57735, 0.57735, 0.57735}}

```
NewCauchy = Q. CauchyStr .Transpose[Q] // MatrixForm
```

trixForm=

$$\begin{pmatrix} 4. & 0. & 0. \\ 0. & -2. & 0. \\ -1.66533\times 10^{-16} & 0. & 1. \end{pmatrix}$$