Coordinate Transformations

Introduction

Coordinate transformations are nonintuitive enough in 2-D, and positively painful in 3-D. This page tackles them in the following order: (i) vectors in 2-D, (ii) tensors in 2-D, (iii) vectors in 3-D, (iv) tensors in 3-D, and finally (v) 4th rank tensor transforms.

A major aspect of coordinate transforms is the evaluation of the transformation matrix, especially in 3-D. This is touched on here, and discussed at length on the next page.

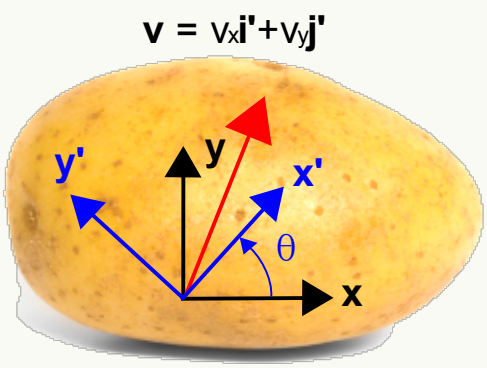
It is very important to recognize that all coordinate transforms on this page are rotations of the coordinate system while the object itself stays fixed. The "object" can be a vector such as force or velocity, or a tensor such as stress or strain in a component. Object rotations are discussed in later sections.

2-D Coordinate Transforms of Vectors

The academic potato provides an excellent example of how coordinate transformations apply to vectors, while at the same time stressing that it is the coordinate system that is rotating and not the vector... or potato.

The potato on the left has a vector on it. But without a coordinate system, there is no way to describe the vector. So a coordinate system has been added to the potato as shown on the right, allowing the vector to now be described as $\mathbf{v} = 2\mathbf{i} + 9\mathbf{j}$.

So now we introduce a rotated coordinate system shown in blue below, using x' and y'. The new system is rotated counter-clockwise by an angle, θ, from the initial coordinate system. Note that the vector itself does not change at all. It is still the very same vector as before. But it is described by different numerical values in the new coordinate system. In this case, the vector is more closely parallel to the new x' axis than the y' axis, so the $\mathbf{i}'$ component will be greater than the $\mathbf{j}'$ component. The transformation is given below the figure.



The 2-D vector transformation equations are

$$v'_x = \quad v_x \cos\theta + v_y \sin\theta$$

$$v'_y = -v_x \sin\theta + v_y \cos\theta$$

This can be seen by noting that

- the part of v_x that lies along the x' axis is $v_x \cos\theta$
- the part of v_y that lies along the x' axis is $v_y \sin\theta$
- the part of v_x that lies along the y' axis is $-v_x \sin\theta$
- the part of v_y that lies along the y' axis is $v_y \cos\theta$

These four factors make up the four terms in the transformation equations. They are easily checked by setting $\theta = 0°$ and $\theta = 90°$. When $\theta = 0°$, then $v'_x = v_x$ and $v'_y = v_y$. When $\theta = 90°$, then $v'_x = v_y$ and $v'_y = -v_x$.

Returning to the Potato...

So, if $\theta = 50°$, then

$$v'_x = \quad 2\cos 50° + 9\sin 50° = 8.18$$

$$v'_y = -2\sin 50° + 9\cos 50° = 4.25$$

The v'_x component is indeed greater than the v'_y component as expected.

Transformation Matrix

It is more convenient to write (and work with) transformation equations using matrices.

$$\left\{ \begin{array}{c} v'_x \\ v'_y \end{array} \right\} = \left[\begin{array}{cc} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{array} \right] \left\{ \begin{array}{c} v_x \\ v_y \end{array} \right\}$$

The $\cos\theta$ terms are on the matrix diagonal while the $\sin\theta$ terms are off-diagonal. The only potential gotcha is remembering which $\sin\theta$ term has the minus sign on it. It is always the lower-left term.

The above equation is written in matrix notation as

$$\mathbf{v}' = \mathbf{Q} \cdot \mathbf{v}$$

where $\mathbf{Q}$ is the usual letter chosen for the transformation matrix.

Transformation vs Rotation Matrices

If this topic weren't already difficult enough, many books and websites add to the confusion by not clarifying what is fixed and what is rotating. In this page and the next, it is the coordinate system that is rotating while the object remains fixed. So the term transformation matrix is used here to emphasize this.

However, we will later address situations in which the object rotates while the coordinate system remains fixed. In this scenario, the term rotation matrix will be used to emphasize that the object is rotating.

Much confusion arises from the amazing fact that each matrix (transform and rotation) is just the transpose of the other! So they look extremely similar. In 2-D problems, the only practical difference is whether the minus sign in front of $\sin\theta$ is on the q_{12} term, or the q_{21} term.

Transformation Matrix Properties

Transformation matrices have several special properties that, while easily seen in this discussion of 2-D vectors, are equally applicable to 3-D applications as well. This list is useful for checking the accuracy of a transformation matrix if questions arise. While a matrix still could be wrong even if it passes all these checks, it is definitely wrong if it fails even one!

- The determinant of $\mathbf{Q}$ equals one.
- The transpose of $\mathbf{Q}$ is its inverse.
- The dot product of any row or column with itself equals one.
  Ex: $(\cos\theta\,\mathbf{i} + \sin\theta\,\mathbf{j}) \cdot (\cos\theta\,\mathbf{i} + \sin\theta\,\mathbf{j}) = 1$
- The dot product of any row with any other row equals zero.
  Ex: $(\cos\theta\,\mathbf{i} + \sin\theta\,\mathbf{j}) \cdot (-\sin\theta\,\mathbf{i} + \cos\theta\,\mathbf{j}) = 0$
- The dot product of any column with any other column equals zero.
  Ex: $(\cos\theta\,\mathbf{i} - \sin\theta\,\mathbf{j}) \cdot (\sin\theta\,\mathbf{i} + \cos\theta\,\mathbf{j}) = 0$

$$\mathbf{Q} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

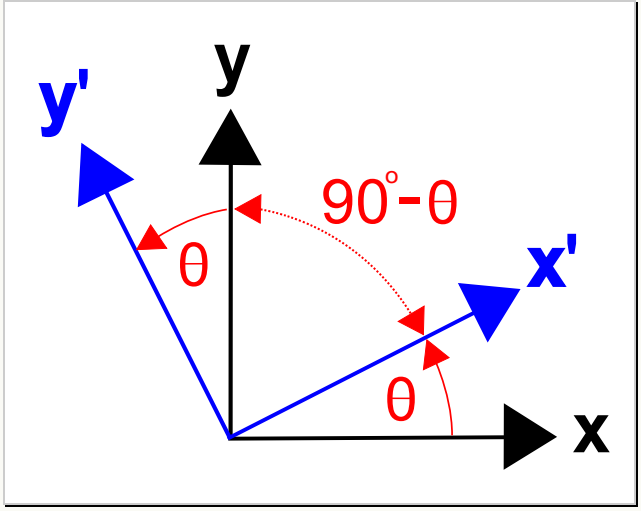
A general method exists for formulating transformation matrices based on the cosines of the angles between the axes of the two coordinate systems, i.e., direction cosines. (This also applies to 3-D transforms.) The transformation matrix can be written as

$$\mathbf{Q} = \begin{bmatrix} \cos(x',x) & \cos(x',y) \\ \cos(y',x) & \cos(y',y) \end{bmatrix}$$

where (x',x) represents the angle between the x' and x axes, (x',y) is the angle between the x' and y axes, etc.

The angle between x' and y is $(90^\circ - \theta)$, and $\cos(x', y) = \cos(90^\circ - \theta) = \sin\theta$.

Likewise, the angle between y' and x is $(90^\circ + \theta)$, and $\cos(y', x) = \cos(90^\circ + \theta) = -\sin\theta$.



Transformation Matrix Formulation

Another way of building up the transformation matrix (and my favorite) is as follows

$$\mathbf{Q} = \begin{bmatrix} \begin{pmatrix} \text{x-comp} \\ \text{of } \mathbf{i}' \end{pmatrix} & \begin{pmatrix} \text{y-comp} \\ \text{of } \mathbf{i}' \end{pmatrix} \\ \begin{pmatrix} \text{x-comp} \\ \text{of } \mathbf{j}' \end{pmatrix} & \begin{pmatrix} \text{y-comp} \\ \text{of } \mathbf{j}' \end{pmatrix} \end{bmatrix}$$

"x-comp of $\mathbf{i}'$" means the x-component of the $\mathbf{i}'$ unit vector. Pay close attention to which terms have primes on them and which don't. Don't confuse this with "the x'-component" because "the x'-comp of $\mathbf{i}'$" is simply 1. Yet another way to say this is, "the first component of the $\mathbf{i}'$ unit vector in the non-primed reference $x - y$ coordinate system."

In summary,

- the x-comp of $\mathbf{i}'$ is $\cos\theta$
- the y-comp of $\mathbf{i}'$ is $\sin\theta$
- the x-comp of $\mathbf{j}'$ is $-\sin\theta$
- the y-comp of $\mathbf{j}'$ is $\cos\theta$

Tensor Notation

The coordinate transform is written in tensor notation as

$$v'_i = \lambda_{ij} v_j$$

where λ_{ij} is the transformation matrix $\mathbf{Q}$. (I don't know why $\mathbf{Q}$ is used in matrix notation, but λ_{ij}, not q_{ij}, is used in tensor notation.) λ_{ij} is defined as

$$\lambda_{ij} = \cos(x'_i, x_j)$$

For example, if $i = 1$ and $j = 2$, then

$$\lambda_{12} = \cos(x'_1, x_2) = \cos(x', y)$$

λ_{ij} is the direction cosine of the angle between the x'_i axis and the x_j axis. Again, this is equally applicable to 3-D transformations as well.

Multiplication of Transformation Matrices

Recall from above that the dot product of any two different rows or columns of a transformation matrix is zero, while the dot product of any row or column with itself is one. This can be written in matrix and tensor notation as

$$\mathbf{Q} \cdot \mathbf{Q}^T = \mathbf{I} \qquad \text{and} \qquad \lambda_{ik}\lambda_{jk} = \delta_{ij}$$

This shows that the transpose of a transformation matrix is also it's inverse.

2-D Coordinate Transforms of Tensors

This section will present the what and how of tensor transforms. The why will have to wait until later.

Coordinate transformations of 2nd rank tensors involve the very same $\mathbf{Q}$ matrix as vector transforms. A transformation of the stress tensor, $\boldsymbol{\sigma}$, from the reference $x-y$ coordinate system to $\boldsymbol{\sigma}'$ in a new $x'-y'$ system is done as follows.

$$\boldsymbol{\sigma}' = \mathbf{Q} \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}^T$$

Writing the matrices out explicitly gives

$$\begin{bmatrix} \sigma'_{xx} & \sigma'_{xy} \\ \sigma'_{xy} & \sigma'_{yy} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

(Note that the stress tensor is always symmetric, even following transformations.)

Multiplying the matrices out gives

$$\begin{aligned} \sigma'_{xx} &= \sigma_{xx}\cos^2\theta + \sigma_{yy}\sin^2\theta + 2\sigma_{xy}\sin\theta\cos\theta \\ \sigma'_{yy} &= \sigma_{xx}\sin^2\theta + \sigma_{yy}\cos^2\theta - 2\sigma_{xy}\sin\theta\cos\theta \\ \sigma'_{xy} &= (\sigma_{yy} - \sigma_{xx})\sin\theta\cos\theta + \sigma_{xy}(\cos^2\theta - \sin^2\theta) \end{aligned}$$

These three equations are exactly the 2-D transform of a stress tensor resulting from summing forces on a differential element and imposing equilibrium. This is also represented by Mohr's circle.

Transformations of Vectors vs Tensors

Note that each stress component in the above equation is multiplied by exactly two trig functions. In contrast, only one trig function is multiplied by any vector component in a

vector transformation. Ex: $v'_x = v_x \cos\theta + v_y \sin\theta$. This is not a coincidence. Each trig function comes from the transform matrix, $\mathbf{Q}$. One matrix is used to transform 1st-rank tensors (i.e., vectors) and two matrices are used to transform 2nd-rank tensors like stress and strain.

$$\mathbf{v}' = \mathbf{Q} \cdot \mathbf{v} \qquad \text{and} \qquad \boldsymbol{\sigma}' = \mathbf{Q} \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}^T$$

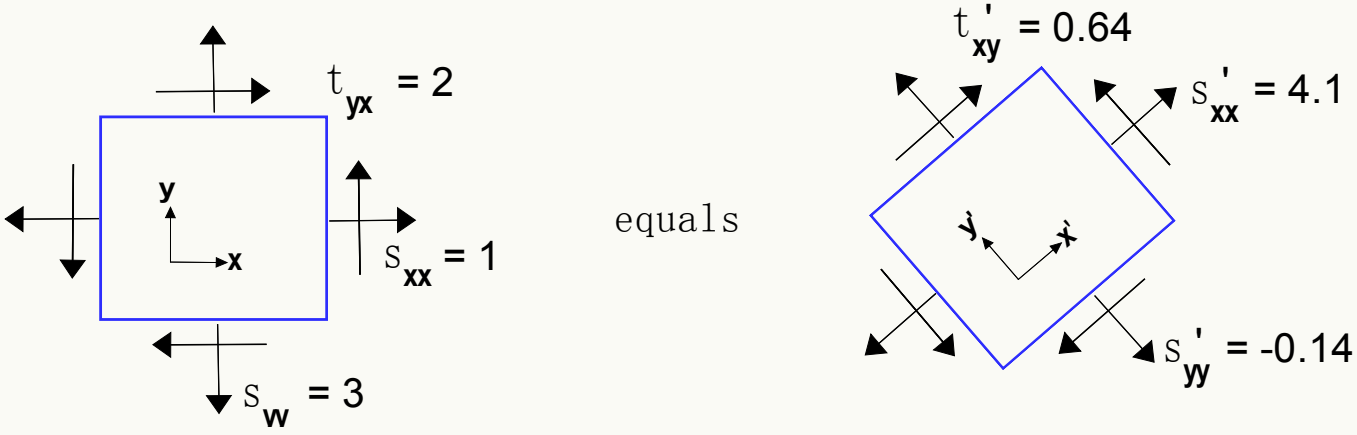
2-D Tensor Transform Example

If the stress tensor in a reference coordinate system is $\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$, then in a coordinate system rotated 50° , it would be written as

$$\begin{bmatrix} \sigma'_{xx} & \sigma'_{xy} \\ \sigma'_{xy} & \sigma'_{yy} \end{bmatrix} = \begin{bmatrix} \cos 50^\circ & \sin 50^\circ \\ -\sin 50^\circ & \cos 50^\circ \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} \cos 50^\circ & -\sin 50^\circ \\ \sin 50^\circ & \cos 50^\circ \end{bmatrix}$$

$$= \begin{bmatrix} 4.143 & 0.638 \\ 0.638 & -0.143 \end{bmatrix}$$

As with the vector example above, the stress state has not changed at all. Only the values in the matrices are different because the orientations of the coordinate systems are different.



Tensor Notation

The coordinate transform is written in tensor notation as

$$\sigma'_{mn} = \lambda_{mi} \lambda_{nj} \sigma_{ij}$$

As usual, tensor notation provides extra insight into the process. This time, the insight comes from the subscripts on the lambdas. Each lambda effectively pairs up a subscript on $\boldsymbol{\sigma}'$ with one on $\boldsymbol{\sigma}$. This is true regardless of the rank of the tensor.

3-D Coordinate Transforms of Vectors

Many of the general equations used in 2-D transformations are also applicable in 3-D. Examples include

$$\mathbf{v}' = \mathbf{Q} \cdot \mathbf{v} \qquad v'_i = \lambda_{ij} v_j \qquad \lambda_{ij} = \cos(x'_i, x_j)$$

Only now the details are different. The vectors have z-components and the transformation matrices are 3x3 instead of 2x2.

$$\mathbf{v} = \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} \qquad \text{and} \qquad \mathbf{Q} = \begin{bmatrix} \cos(x', x) & \cos(x', y) & \cos(x', z) \\ \cos(y', x) & \cos(y', y) & \cos(y', z) \\ \cos(z', x) & \cos(z', y) & \cos(z', z) \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} \begin{pmatrix} \text{x-comp} \\ \text{of } \mathbf{i}' \end{pmatrix} & \begin{pmatrix} \text{y-comp} \\ \text{of } \mathbf{i}' \end{pmatrix} & \begin{pmatrix} \text{z-comp} \\ \text{of } \mathbf{i}' \end{pmatrix} \\ \begin{pmatrix} \text{x-comp} \\ \text{of } \mathbf{j}' \end{pmatrix} & \begin{pmatrix} \text{y-comp} \\ \text{of } \mathbf{j}' \end{pmatrix} & \begin{pmatrix} \text{z-comp} \\ \text{of } \mathbf{j}' \end{pmatrix} \\ \begin{pmatrix} \text{x-comp} \\ \text{of } \mathbf{k}' \end{pmatrix} & \begin{pmatrix} \text{y-comp} \\ \text{of } \mathbf{k}' \end{pmatrix} & \begin{pmatrix} \text{z-comp} \\ \text{of } \mathbf{k}' \end{pmatrix} \end{bmatrix}$$

$$\begin{Bmatrix} v'_x \\ v'_y \\ v'_z \end{Bmatrix} = \begin{bmatrix} \cos(x', x) & \cos(x', y) & \cos(x', z) \\ \cos(y', x) & \cos(y', y) & \cos(y', z) \\ \cos(z', x) & \cos(z', y) & \cos(z', z) \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix}$$

3-D Coordinate Transforms of Tensors

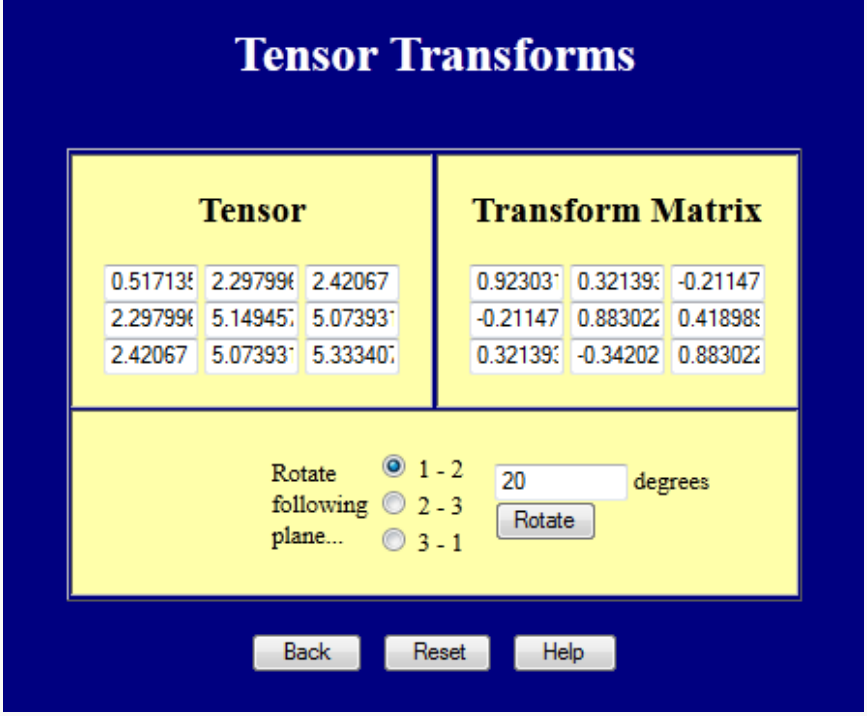
Once again, the rules don't change, only the particulars do.

$$\boldsymbol{\sigma}' = \mathbf{Q} \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}^T \qquad \sigma'_{mn} = \lambda_{mi} \lambda_{nj} \sigma_{ij} \qquad \lambda_{ij} = \cos(x'_i, x_j)$$

Writing the matrices out explicitly gives

$$\begin{bmatrix} \sigma'_{xx} & \sigma'_{xy} & \sigma'_{xz} \\ \sigma'_{xy} & \sigma'_{yy} & \sigma'_{yz} \\ \sigma'_{xz} & \sigma'_{yz} & \sigma'_{zz} \end{bmatrix} = \begin{bmatrix} \cos(x', x) & \cos(x', y) & \cos(x', z) \\ \cos(y', x) & \cos(y', y) & \cos(y', z) \\ \cos(z', x) & \cos(z', y) & \cos(z', z) \end{bmatrix} \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \begin{bmatrix} \cos(x', x) & \cos(y', x) & \cos(z', x) \\ \cos(x', y) & \cos(y', y) & \cos(z', y) \\ \cos(x', z) & \cos(y', z) & \cos(z', z) \end{bmatrix}$$

This webpage performs coordinate transforms on 3-D tensors. Try it out.



Coordinate Transforms of 4th Rank Tensors

We will see in the section on Hooke's Law that the stiffness tensor is 4th rank, i.e., 3x3x3x3 (not 4x4). It is written as C_{ijkl} because it relates any strain component, ϵ_{kl}, to any stress component, σ_{ij}, i.e., $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$. The coordinate transformation law for the 4th rank stiffness tensor is easily written in tensor notation as

$$C'_{ijkl} = \lambda_{im}\lambda_{jn}\lambda_{ko}\lambda_{lp}C_{mnop}$$

The tensor equation directs how to write the transformation in matrix notation.

$$\mathbf{C}' = \mathbf{Q}\cdot\mathbf{Q}\cdot\mathbf{C}\cdot\mathbf{Q}^T\cdot\mathbf{Q}^T$$

4th Rank Tensor Transform Example

The slides below show the stiffness calculation of a typical single ply steel belt. The angle α is the rotation of the cables off the longitudinal axis. Note how the equations contain trig functions to the 4th power, This is consistent with the use of four $\mathbf{Q}$ matrices in the transformation equations.

Lamina Description

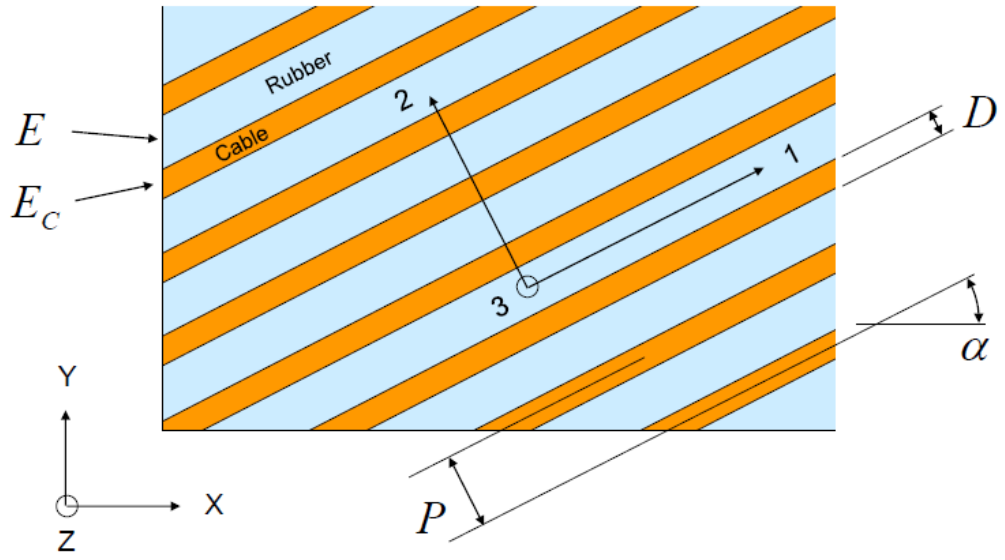
Material Coordinates 1,2,3

Body Coordinates X,Y,Z

Lamina Assumptions:

1. Small deformations – Plane Stress
2. Rubber is incompressible ($\nu = 0.5$)
3. Cable is incompressible (cable stress too low to generate significant strain)
4. Cable modulus $E_C >> E$, rubber modulus
5. Cables don't deform, the rubber does
6. Strain in the cable direction is a function of the cable properties only

   $\varepsilon_1 = \frac{\sigma_1}{V_C E_C}$ where V_C = volume fraction of cable



We can obtain the strain / stress relations for the lamina in the body coordinates: (Note – finite cable modulus) (contribution from cable and rubber and each stress in each strain) (contribution from rubber scaled by (P-D)/P)

$$\varepsilon_x = \frac{\cos^2\alpha}{V_c E_c}\left[\sigma_x \cos^2\alpha + \sigma_y \sin^2\alpha + 2\tau_{xy}\sin\alpha\cos\alpha\right] + \frac{1+\upsilon}{E}\left(\frac{P-D}{P}\right)\left\{\begin{matrix}\sigma_x \sin^2\alpha\left[2-(1+\upsilon)\sin^2\alpha\right] - \sigma_y(1+\upsilon)\cos^2\alpha\sin^2\alpha - \\ \tau_{xy}\sin 2\alpha\left[1-(1+\upsilon)\cos^2\alpha\right]\end{matrix}\right\}$$

$$\varepsilon_y = \frac{\sin^2\alpha}{V_c E_c}\left[\sigma_x \sin^2\alpha + \sigma_y \cos^2\alpha - 2\tau_{xy}\sin\alpha\cos\alpha\right] + \frac{1+\upsilon}{E}\left(\frac{P-D}{P}\right)\left\{\begin{matrix}\sigma_y \cos^2\alpha\left[2-(1+\upsilon)\cos^2\alpha\right] - \sigma_x(1+\upsilon)\cos^2\alpha\sin^2\alpha - \\ \tau_{xy}\sin 2\alpha\left[1-(1+\upsilon)\cos^2\alpha\right]\end{matrix}\right\}$$

$$\gamma_{xy} = \frac{\sin 2\alpha}{V_c E_c}\left[(\sigma_y - \sigma_x)\sin\alpha\cos\alpha + \tau_{xy}(\cos^2\alpha - \sin^2\alpha)\right] + \frac{1+\upsilon}{E}\left(\frac{P-D}{P}\right)\left\{\begin{matrix}-\sigma_y \sin 2\alpha\left[1-(1+\upsilon)\cos^2\alpha\right] - \sigma_x \sin 2\alpha\left[1-(1+\upsilon)\sin^2\alpha\right] - \\ \tau_{xy}\left[2-(1+\upsilon)\sin^2 2\alpha\right]\end{matrix}\right\}$$

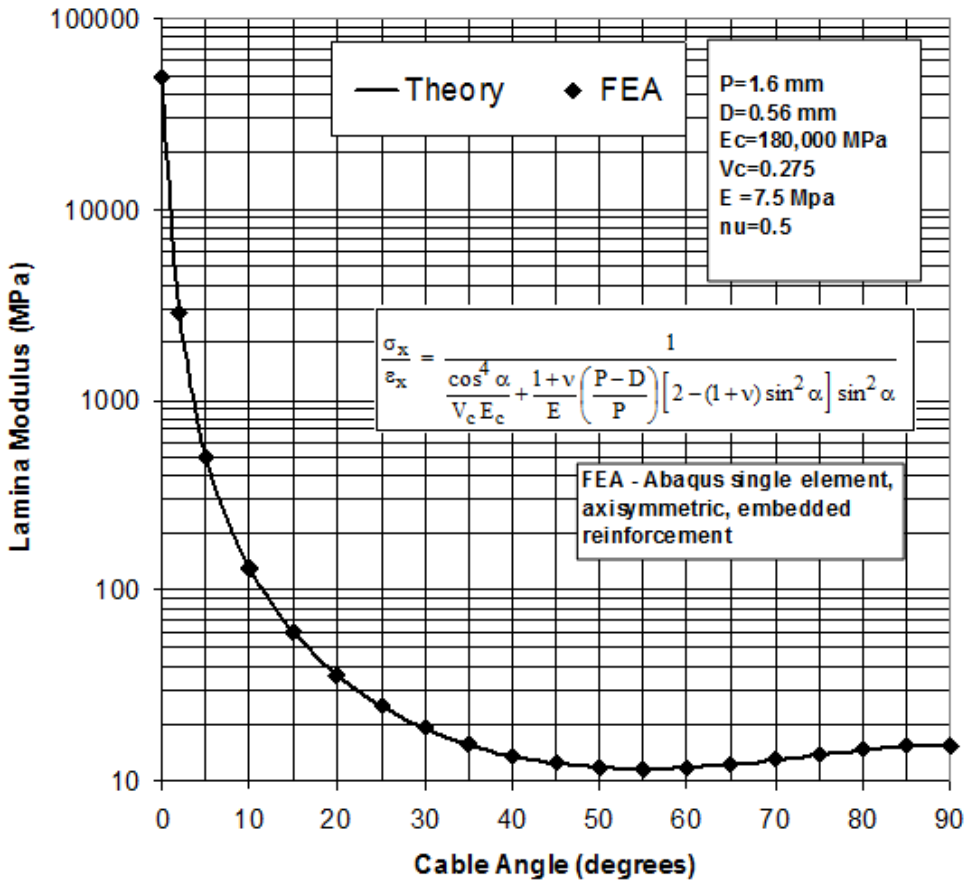


FIG. 2 – *Lamina modulus versus the belt cable angle. The analytical solution, EQ. 20, is compared with a finite element solution. The belt parameters are typical of those used in modern radial tire belts. The agreement is very good over the full range of possible belt cable angles.*

Reference: McGinty, R. D., Rhyne, T. B., and Cron, S. M., "Analytical Solution for the Stresses Arising in +/- Angle Ply Belts of Radial Tires," Tire Science and Technology, TSTCA, Vol. 36, No. 4, October - December 2008, pp. 244-274.

Summary

The coordinate transform of a vector in matrix and tensor notation is

$$\mathbf{v}' = \mathbf{Q} \cdot \mathbf{v} \qquad \text{and} \qquad v'_i = \lambda_{ij} v_j$$

The coordinate transform of a tensor in matrix and tensor notation is

$$\boldsymbol{\sigma}' = \mathbf{Q} \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}^T \qquad \text{and} \qquad \sigma'_{mn} = \lambda_{mi}\lambda_{nj}\sigma_{ij}$$

Note that $\mathbf{Q}$ and λ_{ij} are the same transformation matrix.

In 2-D, $\mathbf{Q}$ and λ_{ij} are defined as

$$\mathbf{Q} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

which is a special case of the more general 3-D form

$$\mathbf{Q} = \begin{bmatrix} \cos(x',x) & \cos(x',y) & \cos(x',z) \\ \cos(y',x) & \cos(y',y) & \cos(y',z) \\ \cos(z',x) & \cos(z',y) & \cos(z',z) \end{bmatrix}$$

where $\cos(x',x)$ is the direction cosine of the angle between the rotated x' axis and the reference x axis.

The coordinate transform of a 4th rank tensor is

$$\mathbf{C}' = \mathbf{Q} \cdot \mathbf{Q} \cdot \mathbf{C} \cdot \mathbf{Q}^T \cdot \mathbf{Q}^T$$

in matrix notation and

$$C'_{ijkl} = \lambda_{im}\lambda_{jn}\lambda_{ko}\lambda_{lp}C_{mnop}$$

in tensor notation.

Table of Contents

Divergence Theorem

Transformation Matrices