ME201 Project 2: Deformation

Tristan Schwab

February 2022

1.1 Deformation Gradients

The deformation gradient, is defined by $\mathbf{F(t)} = \nabla x = \frac{dx}{dX}$ where:

$$x = u + X \tag{1}$$

. To define the deformation gradient in terms of $A(t)$, we can write x as $u + X$, and take the gradient w.r.t X yielding: $\mathbf{F(t)} = \nabla x = \nabla u + I$. We know that u w.r.t X is simply $A(t)$, so $\mathbf{F(t)} = A(t) + I$. This correlates to a 3 x 3 matrix.

For $A(t)$ to be physically permissible, we need the following to hold:

$$0 < det(\mathbf{F(t)}) < \infty \tag{2}$$

where $J = det(\mathbf{F}(t))$. This is the case, because for deformation to occur, the deformation gradient must fall within all real-positive values. Values less than one will correspond to a shrink while values that tend towards infinity will indicate expansion. If there is no deformation at t=0 (undeformed), then $A(t) = 0$ and $J = det(\mathbf{F}(t)) = det(A(t) + I) = 1$. This relation can be observed by the following relation of conservation of mass:

$$\frac{m}{dV} = \frac{mJ}{dV_0} \tag{3}$$

$$\rho_{final} = J\rho_0 = \rho_0 \tag{4}$$

The expression for the deformed state of a matrix $\mathbf{A}$ can be written in terms of ρ_0 ρ given the relation (2). Rearranging:

$$\rho_0 = \rho(det(\mathbf{A}(t) + I)) \tag{5}$$

where ρ_0 and ρ are the original and final densities, respectively. As aforementioned, the relation $\mathbf{A}$ must satisfy for volume preservation is that the Jacobian (the determinant of the deformation matrix $\mathbf{F}(t)$) must be equal to one.

Refers to Problems 1 and 2

1.2 Determining Components of Deformation

This section considers the laws relating the second Piola-Kirchhoff stress to the Green-Lagrange strain with bulk and shear moduli κ and μ (see appendix)

A second order Green-Lagrange Strain, E can be determined via using the provided $A(t)$ matrix:

$$E = \frac{1}{2} * F^T * F - I \tag{6}$$

$$F(t) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix} \tag{7}$$

$$E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0.5 & 0.5 \end{bmatrix} \tag{8}$$

The Second-Order Piola-Kirchoff Stress can be calculated using the relations in the appendix assuming isotropic material behavior.

$$S = E_L * ES = \begin{bmatrix} \frac{k}{2} - \frac{\mu}{3} \\ \frac{k}{2} - \frac{\mu}{3} \\ \frac{k}{2} + \frac{2*\mu}{3} \\ 0 \\ \mu \\ 0 \end{bmatrix} \tag{9}$$

The Cauchy-Stress is calculated using:

$$\sigma = \frac{1}{J} * F * S * F^T \tag{10}$$

where $J = det(F) = 8$

$$\sigma = \begin{bmatrix} \frac{k}{4} - \frac{\mu}{6} & 0 & 0 \\ 0 & \frac{5k}{16} + \frac{5\mu}{12} & \frac{k}{8} + \frac{2\mu}{3} \\ 0 & \frac{k}{8} + \frac{2\mu}{3} & \frac{k}{4} + \frac{\mu}{3} \end{bmatrix} \tag{11}$$

The volumetric dialation is the hydrostatic pressure constant determined from trace of the Cauchy-Stress. This will be used to evaluate the deviatoric stress, which is the difference between the Cauchy-Stress and the Hydrostatic Pressure calculated to determine the change of shape of a body.

$$P = \frac{trace(\sigma)}{3} = \frac{12k}{48} + \frac{7\mu}{36} \tag{12}$$

$$\sigma_{dev} = \begin{bmatrix} \frac{-k}{48} - \frac{13\mu}{36} & 0 & 0 \\ 0 & \frac{k}{24} + \frac{2\mu}{9} & \frac{k}{8} + \frac{2\mu}{3} \\ 0 & \frac{k}{8} + \frac{2\mu}{3} & \frac{-k}{48} + \frac{5\mu}{36} \end{bmatrix} \tag{13}$$

1.3 Arbitrary Uniform Displacement with Isotropic Material Properties

If the initial location of a material point is given by $X = [1\ 1\ 1]^T$ then the deformed location of **x** (at the same material point) is determined by:
$u = A(t) * X$ and using *eq.(1)*

$$x = \begin{bmatrix} 1.02 \\ 1.05 \\ 1.06 \end{bmatrix} \tag{14}$$

If, on the other hand we are given the deformed location of a material point instead, x, then the inverse is true. Obeying the linear algebra:
$X = (A(t) + I)^{-1} * x$

$$X = \begin{bmatrix} 0.9810 \\ 0.9523 \\ 0.9429 \end{bmatrix} \tag{15}$$

The Cauchy-Stress Tensor for the deformation discussed in (4) and using *eq.(10)* is the following:

$$\sigma = \frac{1}{1.05} \begin{bmatrix} 1 & 0.01 & 0.01 \\ 0.01 & 1.02 & 0.02 \\ 0.01 & 0.02 & 1.03 \end{bmatrix} * \begin{bmatrix} s11 & s12 & s13 \\ s21 & s22 & s23 \\ s31 & s32 & s33 \end{bmatrix} * \begin{bmatrix} 1 & 0.01 & 0.01 \\ 0.01 & 1.02 & 0.02 \\ 0.01 & 0.02 & 1.03 \end{bmatrix} \tag{16}$$

To determine a traction vector, we use the following:

$$t = \sigma * n \tag{17}$$

With knowledge of the surface, we can determine the gradient of the surface, which is *ALWAYS* normal to the surface. This will help in determining n. The gradient of the function provided is: $\nabla s = 2e^{(2x_1)} + 2x_2 + 2sin(2x_3)$ with magnitude $||\nabla s|| = \sqrt{4e^{(4x_1)} + 4sin(2x_3)^2 + 4x_2^2}$. $n = \frac{\nabla s}{||\nabla s||}$. The calculations for the traction vector can be found in the appendix

1.4 Reference Configuration of Deformation using Balance of Linear Momentum

$$\nabla_x E(t) * S(t) + f = \rho \ddot{x} = \nabla_x \sigma(t) + f = \rho \ddot{x}$$

$$\nabla_x \sigma(t) = 0$$

$$x = u + X \quad \dot{x} = \dot{u} + \dot{X} = \dot{u} \quad \ddot{x} = \ddot{u}$$

$$U = A(t) * X \implies \ddot{u} = \ddot{A}(t) * X = \ddot{x}$$

$$f = \rho \ddot{x} \implies \rho(\ddot{A}(t) * X)$$

Converting to Referential Coordinates

$$\rho = \rho_0 J = \rho_0(det(A(t) + I))$$

$$f = \frac{\ddot{A}(t) * X}{det(A(t) + I)}$$

1.5 Determining λ Principle Stresses and Directions

The eigen-value problem is first determined by *eq:(10)* and solving

$$\sigma v = \lambda v \implies (\sigma - \lambda I) = 0 \tag{18}$$

$$\sigma = \begin{bmatrix} d & 0 & d \\ 0 & \frac{1}{9} & 0 \\ d & 0 & -d \end{bmatrix}$$

To solve for Eigen-Values, we subtract a diagonal lambda matrix from σ and take the determinant. This can also be done in MatLab, using the eig(:) command. The Eigen-Values, which correspond to the **principal stresses** for the C-S Tensor are:

$$\lambda = \begin{bmatrix} \frac{1}{9} \\ d\sqrt{2} \\ -d\sqrt{2} \end{bmatrix}$$

The Eigen-Vectors, which correspond to the **principal stress directions** for the C-S Tensor are calculated by taking the maximum λ value and solving for v in *eq:(18)*. The Eigen-Vectors can also be determined in Matlab using the eig(:) command.

$$v = \begin{bmatrix} 0 & \sqrt{2}+1 & 1-\sqrt{2} \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Here, we know that the principal stress of largest magnitude is $\lambda = d\sqrt{2}$ which corresponds to the direction of $[x_1\ x_2\ x_3] = [\sqrt{2}+1\ \ 0\ \ 1]$.

1.6 Von Mises Failure Criterion

$$\sigma_{\text{v}}^2 = \frac{1}{2}\left[(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2 + 6\left(\sigma_{23}^2 + \sigma_{31}^2 + \sigma_{12}^2\right)\right]$$

$$\sigma_Y = \sqrt{\frac{d-\frac{1}{9}^2}{2} + \frac{d+\frac{1}{9}^2}{2} + 5d^2}$$

and rewriting into terms of known constant σ_Y we get

$$d = \sqrt{\frac{81\sigma_Y^2 - 1}{(9\sqrt{6}}}$$

which is the value of d at which yielding will occur using the Von Mises Stress Criterion

Prepared for: ME201: Adv. Manufacturing Simulations Spring 2022

Table of Contents

Prepared by: Tristan W. Schwab, B.Sc. Mechanical Engineering 2023. Intended to accompany pdf submission for Project 2.

Basics of Deformation Problem 2,3

Below, we compute deformation parameters assuming an arbitrary starting conditon

```
F = [1 0 0;0 1 1;0 0 1]+eye(3);
A_prime = [1 0 0;0 1 1;0 0 1];
E = 0.5*(A_prime'*A_prime-eye(3));
E6=[E(1,1);E(2,2);E(3,3);2*E(1,2);2*E(2,3);2*E(3,1)];% E-values
 provided
syms k
syms mu
c1=k+4/3*mu; % constant k
c2 = k-2/3*mu; % constant
EE=[c1,c2,c2;c2,c1,c2;c2,c2,c1] % Green Lagrange
imu = eye(3)*mu; %mu's
EE=[EE zeros(3,3);zeros(3,3) imu]; % Green lagrange padded with zeros
S =EE*E6 % second piola-kirchoff
S_3 = [S(1) 0 0; 0 S(2) mu ; 0 mu S(3)];
sigma = 1/det(F)*(F*S_3*F')%cauchy stress
P = diag([1/3 * trace(sigma) 1/3 * trace(sigma) 1/3 * trace(sigma)]) %
 Pressure Tensor (volumetric component of sigma)
dev_sigma = sigma - P %deviatoric stress
```

```
EE =

[ k + (4*mu)/3, k - (2*mu)/3, k - (2*mu)/3]
[ k - (2*mu)/3, k + (4*mu)/3, k - (2*mu)/3]
[ k - (2*mu)/3, k - (2*mu)/3, k + (4*mu)/3]


S =       k/2 - mu/3
          k/2 - mu/3
     k/2 + (2*mu)/3
                  0
                 mu
                  0
```

```
sigma =

[ k/4 - mu/6,                    0,                0]
[          0, (5*k)/16 + (5*mu)/12, k/8 + (2*mu)/3]
[          0,       k/8 + (2*mu)/3,     k/4 + mu/3]


P =

[ (13*k)/48 + (7*mu)/36,                     0,                     0]
[                     0, (13*k)/48 + (7*mu)/36,                     0]
[                     0,                     0, (13*k)/48 + (7*mu)/36]


dev_sigma =

[ - k/48 - (13*mu)/36,                 0,                 0]
[                   0, k/24 + (2*mu)/9,    k/8 + (2*mu)/3]
[                   0,  k/8 + (2*mu)/3, (5*mu)/36 - k/48]
```

Consideraton of a different displacement

```
A = 0.01 * [0 1 1; 1 2 2; 1 2 3];
%Part A
X_init = [1 1 1]';
g=A+eye(3);
u=A*X_init;
little_x = u+X_init
% Part B
x_final = [1 1 1]';
X_prev=inv(g)*x_final

% Part C
syms s11 s22 s33 s12 s13 s23 s31 s32 s21
J=det(F)
S=[s11 s21 s31; s12 s22 s13; s13 s23 s33];
sigma_cauchy = 1/J *( F * S * F') % write in terms of a product of
 matrices

% part D
syms x1 x2 x3
s = exp(2*x1)+ x2^2 - cos(2*x3);
g_s=gradient(s);
g_n=norm(g_s);
n=g_s/g_n
t= sigma_cauchy*n %traction vector
```

```
little_x =

    1.0200
    1.0500
    1.0600


X_prev =

    0.9810
    0.9523
    0.9429


J =

     8


sigma_cauchy =

[          s11/2,                   s21/2 + s31/4,          s31/2]
[ s12/2 + s13/4, s13/4 + s22/2 + s23/4 + s33/8, s13/2 + s33/4]
[          s13/2,                   s23/2 + s33/4,          s33/2]


n =

 (2*exp(2*x1))/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 +
 4*exp(4*real(x1)))^(1/2)
        (2*x2)/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 +
 4*exp(4*real(x1)))^(1/2)
 (2*sin(2*x3))/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 +
 4*exp(4*real(x1)))^(1/2)
```

NOTE: traction vector is poorly evaluated in MatLab

```
t =

                                             (2*x2*(s21/2 + s31/4))/
(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 + 4*exp(4*real(x1)))^(1/2)
 + (s11*exp(2*x1))/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 +
 4*exp(4*real(x1)))^(1/2) + (s31*sin(2*x3))/(4*abs(sin(2*x3))^2 +
 4*abs(x2)^2 + 4*exp(4*real(x1)))^(1/2)
 (2*x2*(s13/4 + s22/2 + s23/4 + s33/8))/(4*abs(sin(2*x3))^2 +
 4*abs(x2)^2 + 4*exp(4*real(x1)))^(1/2) + (2*exp(2*x1)*(s12/2 +
 s13/4))/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 + 4*exp(4*real(x1)))^(1/2)
 + (2*sin(2*x3)*(s13/2 + s33/4))/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 +
 4*exp(4*real(x1)))^(1/2)
                                             (2*x2*(s23/2 + s33/4))/
(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 + 4*exp(4*real(x1)))^(1/2)
 + (s13*exp(2*x1))/(4*abs(sin(2*x3))^2 + 4*abs(x2)^2 +
 4*exp(4*real(x1)))^(1/2) + (s33*sin(2*x3))/(4*abs(sin(2*x3))^2 +
 4*abs(x2)^2 + 4*exp(4*real(x1)))^(1/2)
```

~~mu~~
~~0~~

New Material Deformations, Principal Cauchy Stress

```
syms d
F= [3 0 0; 0 1 0; 0 0 3];
S = [d 0 d; 0 1 0; d 0 -d];
J=det(F)
sigma = 1/J *( F * S * F')
[vectors, values] = eig(sigma)
direction = vectors(:,2)

% determine the von mises criterion
sigma_y = (0.5*((sigma(1,1) - sigma(2,2))^2 + (sigma(2,2) -
 sigma(3,3))^2 + (sigma(3,3)-sigma(1,1))^2 + 6*(sigma(2,3)^2 +
 sigma(1,3)^2 + sigma(1,2)^2)))^(1/2)
```

```
J =

     9


sigma =

[ d,   0,  d]
[ 0, 1/9,  0]
[ d,   0, -d]


vectors =

[ 0, 2^(1/2) + 1, 1 - 2^(1/2)]
[ 1,           0,           0]
[ 0,           1,           1]


values =

[ 1/9,          0,           0]
[   0, 2^(1/2)*d,           0]
[   0,          0, -2^(1/2)*d]


direction =

 2^(1/2) + 1
           0
           1

sigma_y =

((d - 1/9)^2/2 + (d + 1/9)^2/2 + 5*d^2)^(1/2)
```