Quantitative approach to strain modelling

using Python, Numpy and Matplotlib

Mgr. Ondrej Lexa, Ph.D.

Institute of Petrology and Structural Geology
Faculty od Science
Charles University in Prague

lexa@natur.cuni.cz

2016

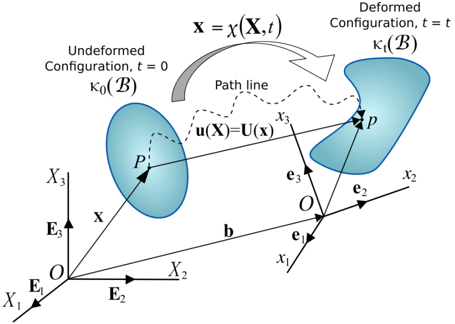
Outlines

- Concept of homogeneous deformation
- From finite to continuous deformation
- Superposition of deformations

Deformation in terms of displacement

A change in the configuration of a continuum body results in a displacement from an initial or undeformed configuration to a current or deformed configuration. The displacement of a body has two components:

- **Rigid-body displacement**
  - Translation
  - Rotation
- **Deformation or strain**
  - Distortion - isochoric change in shape
  - Dilation - change in volume



Homogeneous deformation

Often described as deformation during which lines remain as lines and parallel lines remain parallel. Homogeneous deformation could be described as **affine transformation** of initial coordinates:

$$x = aX + bY + t_X$$
$$y = cX + dY + t_Y$$

or in matrix form using homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_X \\ c & d & t_Y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Properties of homogeneous deformation are not spatially dependent.

Deformation gradient

Without translation the homogeneous deformation (rotation and strain) could be described as:

$$x = aX + bY$$
$$y = cX + dY$$
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

or

$$x = \mathbf{F}X$$

where **F** is so called **deformation gradient**.

Note, that as we excluded translation, the origin of coordinates do not change during deformation:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{F} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Displacement gradient

Displacement of particle is vector between initial and final postion, i.e:

$$u = x - X = aX + bY - X = (a-1)X + bY$$
$$v = y - Y = cX + dY - Y = cX + (d-1)Y$$

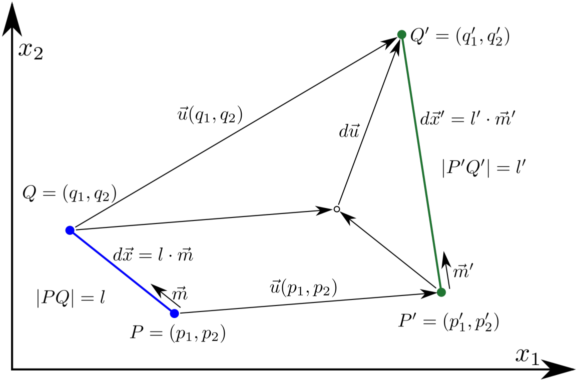
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a-1 & b \\ c & d-1 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

or

$$u = (\mathbf{F} - \mathbf{I})X = \boldsymbol{\nabla}\mathbf{u}X$$

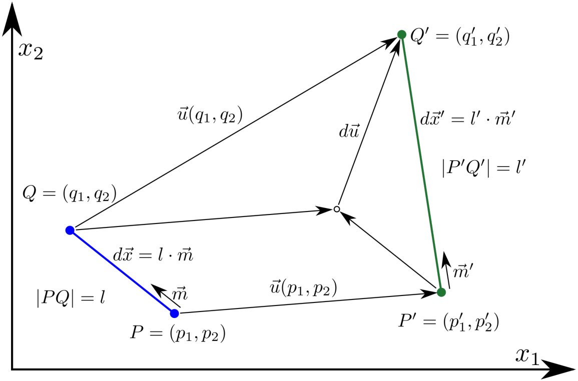
where $\boldsymbol{\nabla}\mathbf{u}$ is so called **displacement gradient**.

More details on displacement and deformation gradient I.



$$Q = P + d\vec{x}$$
$$Q' = P' + d\vec{x}'$$

More details on displacement and deformation gradient II.



$$P' = P + \vec{u}(p_1, p_2)$$
$$P' = P + \nabla\mathbf{u}P$$
$$P' = (\mathbf{I} + \nabla\mathbf{u})P$$

$$Q' = Q + \vec{u}(q_1, q_2)$$
$$Q' = Q + \nabla\mathbf{u}Q$$
$$Q' = (\mathbf{I} + \nabla\mathbf{u})Q$$

Above we demonstrated that $\mathbf{F} = \mathbf{I} + \nabla\mathbf{u}$, so $P' = \mathbf{F}P$ and $Q' = \mathbf{F}Q$

Transformation of vectors

Using above defined equations

$$Q = P + d\vec{x}$$
$$P' = P + \nabla \mathbf{u} P$$
$$Q' = Q + \nabla \mathbf{u} Q$$

we can define $Q' = P + d\vec{x} + \nabla \mathbf{u} P + \nabla \mathbf{u} d\vec{x}$. The vector connecting two points changes according to

$$d\vec{u} = d\vec{x}' - d\vec{x} = (Q' - P') - (Q - P)$$
$$d\vec{u} = P + d\vec{x} + \nabla \mathbf{u} P + \nabla \mathbf{u} d\vec{x} - P - \nabla \mathbf{u} P - P - d\vec{x} + P$$
$$d\vec{u} = \nabla \mathbf{u} d\vec{x}$$

or

$$d\vec{x}' = d\vec{x} + d\vec{u} = d\vec{x} + \nabla \mathbf{u} d\vec{x} = (\mathbf{I} + \nabla \mathbf{u}) d\vec{x} = \mathbf{F} d\vec{x}$$

Time to think...

$\mathbf{F}$ maps any undeformed vector into its deformed state. This vector can also be a position vector of a point. Therefore $\mathbf{F}$ also maps any point into its new position after deformation. In another words, deformation gradient $\mathbf{F}$ maps a undeformed vector into its deformed state. Considering two successive deformations $\mathbf{F_1}$ and $\mathbf{F_2}$ write transformation equation....

Python exercise

Lets try to visualize how unit circle deforms during homogeneous deformation:

```
from pylab import *

# parametric definition of unit circle
theta = linspace(0, 2*pi, 300)
Xc, Yc = cos(theta), sin(theta)
Xs, Ys = [-1, 1, 1, -1, -1], [1, 1, -1, -1, 1]
plot(Xc, Yc, 'g', Xs, Ys, 'b', lw=2)
# Apply deformation gradient and plot ellipse
F = array([[2, 0], [0, 0.5]])
xe, ye = dot(F, [Xc, Yc])
xq, yq = dot(F, [Xs, Ys])
plot(xe, ye, 'r', xq, yq, 'm', lw=2)
axis('equal')
```

Python exercise

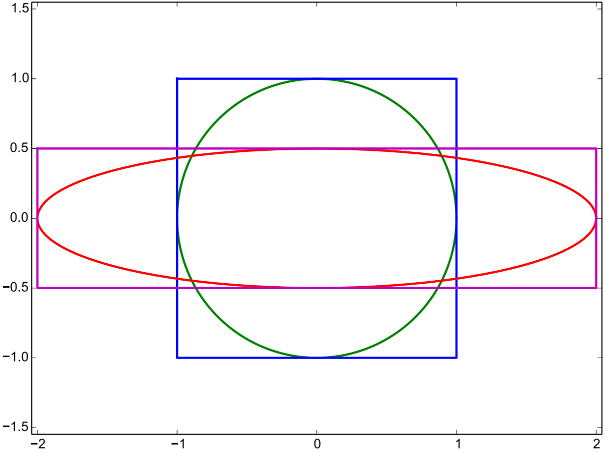


Figure : Transformation of circle to ellipse.

Python exercise

To visualize displacement field we have to calculate it for points on regular grid and plot it using command **quiver**.

```
# create rectangular grid
Xg, Yg = meshgrid(linspace(-2.2, 2.2, 15),
                  linspace(-1.9, 1.9, 12))
X, Y = Xg.flatten(), Yg.flatten()

# calculate displacements
J = F - eye(F.ndim)
u, v = dot(J, [X, Y])

# plot
quiver(X, Y, u, v, angles='xy', lw=0.5, headwidth=4)
plot(Xc, Yc, 'g', Xs, Ys, 'b',
     xe, ye, 'r', xq, yq, 'm', lw=2)
axis('equal')
```

Python exercise

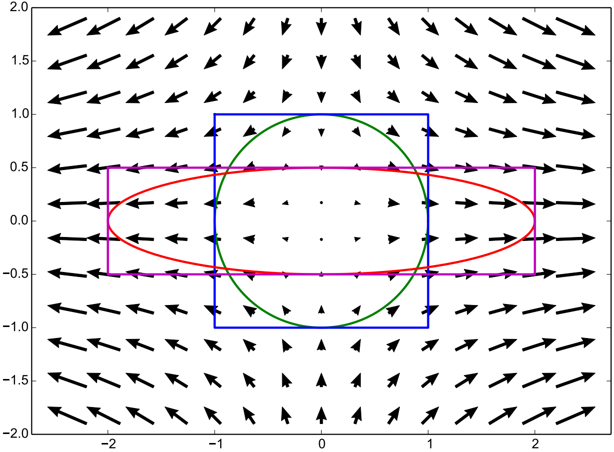


Figure : Pure shear displacement field.

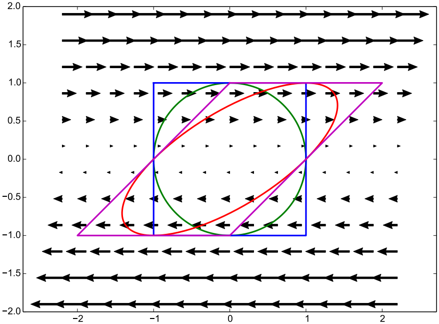
Examples of simple shear

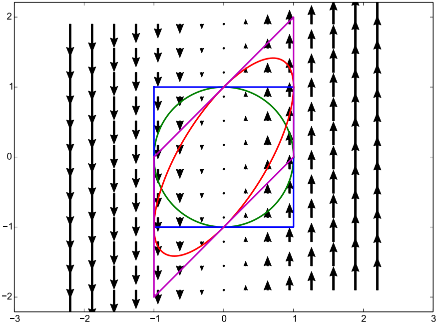
$$\mathbf{F} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\nabla \mathbf{u} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\nabla \mathbf{u} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$





Examples of general shear

$$\mathbf{F} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} 1.3 & -0.375 \\ 0.75 & 0.65 \end{bmatrix}$$

$$\nabla\mathbf{u} = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \qquad \nabla\mathbf{u} = \begin{bmatrix} 0.3 & -0.375 \\ 0.75 & -0.35 \end{bmatrix}$$