# 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, Prague

lexa@natur.cuni.cz

2019

# Outlines

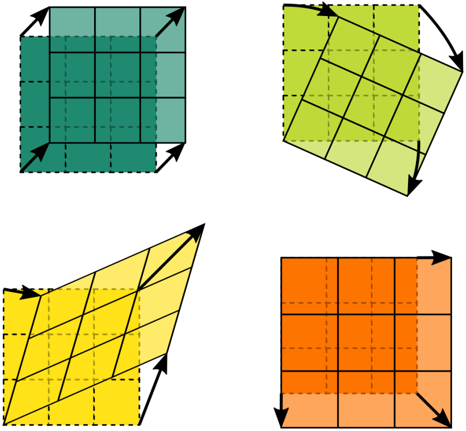
- Concept of homogeneous deformation
- From finite to continuous deformation
- Superposition of deformations

# Components of deformation

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

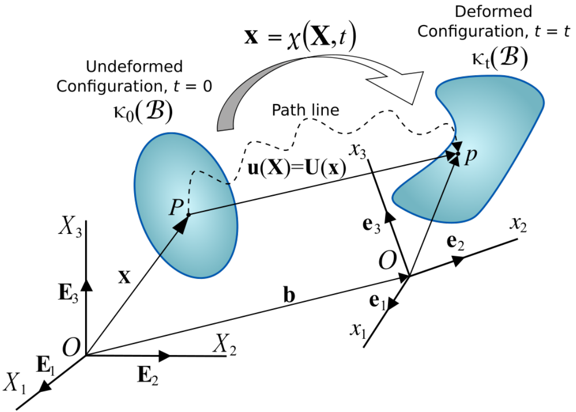
  implies the change in shape and/or size of the body from an initial or undeformed configuration



# Kinematics of continuum body

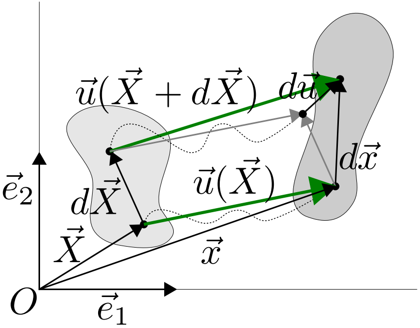
The motion of a continuum body is a **continuous** time sequence of displacements. Thus, the material body will occupy **different configurations** at different times so that a particle occupies a series of points in space which describe a **pathline**. There is **continuity** during deformation or motion of a continuum body in the sense that:

- The material points forming a closed curve at any instant will always form a closed curve at any subsequent time.
- The material points forming a closed surface at any instant will always form a closed surface at any subsequent time and the matter within the closed surface will always remain within.



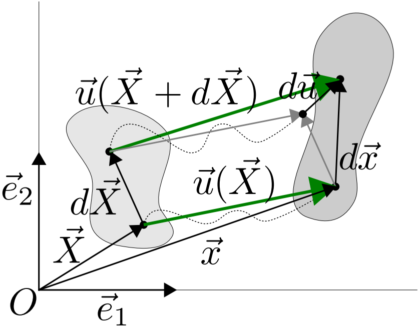
# Kinematics: deformation and motion I.

It is convenient to identify a **reference configuration or initial condition** which all subsequent **deformed configurations** are referenced from. Often, the configuration at $t = 0$ is considered the reference configuration.




The components $x_i$ of the position vector $\vec{x}$ of a particle, taken with respect to the reference configuration, are called the **material or reference coordinates**.

# Kinematics of infinitesimal deformation I.




The displacement of first point is decribed as:

$$\vec{x} = \vec{X} + \vec{u}(\vec{X})$$

while displacement of second surrounding point is described as:

$$\vec{x} + d\vec{x} = \vec{X} + d\vec{X} + \vec{u}(\vec{X} + d\vec{X})$$

Substituting first equation into second we got:

$$\vec{X} + \vec{u}(\vec{X}) + d\vec{x} = \vec{X} + d\vec{X} + \vec{u}(\vec{X} + d\vec{X})$$

which simplifies to:

$$d\vec{x} = d\vec{X} + \vec{u}(\vec{X} + d\vec{X}) - \vec{u}(\vec{X})$$

# Detour on Taylor's theorem

Taylor's theorem states that any function that is infinitely differentiable may be represented by a Taylor series expansion:

$$f(X+dX) = f(X) + \frac{f'(X)}{1!}dX + \frac{f''(X)}{2!}dX^2 + ... = \sum_{k=0}^{\infty} \frac{f^{(k)}(X)}{k!}dX^k$$

than

$$f(X+dX) - f(X) = \frac{f'(X)}{1!}dX + \frac{f''(X)}{2!}dX^2 + ... = \sum_{k=0}^{\infty} \frac{f^{(k)}(X)}{k!}dX^k$$

neglecting higher terms as $\left|d\vec{X}\right| \ll 1$ as $dX^k$ is very small, it is:

$$f(X+dX) - f(X) = \frac{f'(X)}{1!}dX = (\boldsymbol{\nabla f})dX$$

where $\boldsymbol{\nabla f}$ is gradient of vector field.

# Kinematics of infinitesimal deformation II.

Using that for infinitesimal deformation equation

$$d\vec{x} = d\vec{X} + \vec{u}(\vec{X} + d\vec{X}) - \vec{u}(\vec{X})$$

it could be written in terms of gradient as:

$$d\vec{x} = d\vec{X} + (\boldsymbol{\nabla u})d\vec{X}$$

where $\boldsymbol{\nabla u}$ is gradient of displacement field or **displacement gradient**.

# Displacement gradient

The **displacement gradient** is the derivative of each component of the linear element displacement $d\vec{u}$ with respect to each component of the reference element $d\vec{X}$:

$$\boldsymbol{\nabla u} = u_{i,j} = \frac{\partial u_i}{\partial X_j} = \begin{bmatrix} \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_2} & \frac{\partial u_1}{\partial X_3} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} & \frac{\partial u_2}{\partial X_3} \\ \frac{\partial u_3}{\partial X_1} & \frac{\partial u_3}{\partial X_2} & \frac{\partial u_3}{\partial X_3} \end{bmatrix}$$

and characterise the local change of the displacement field at a material point with position vector $\vec{X}$. Knowing that:

$$d\vec{u} = d\vec{x} - d\vec{X}$$

it could be also written as:

$$d\vec{u} = (\boldsymbol{\nabla u})d\vec{X}$$

# Deformation gradient

Recalling that $d\vec{u} = d\vec{x} - d\vec{X}$

$$(\boldsymbol{\nabla u}) = \frac{\partial u_i}{\partial X_j} = \frac{\partial (x_i - X_i)}{\partial X_j} = \frac{\partial x_i}{\partial X_j} - \frac{\partial X_i}{\partial X_j} = \mathbf{F} - \mathbf{I}$$

where $\mathbf{F}$ is so called **deformation gradient**, i.e the derivative of each component of the deformed linear element $d\vec{x}$ with respect to each component of the reference element $d\vec{X}$:

$$\mathbf{F} = x_{i,j} = \frac{\partial x_i}{\partial X_j} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix}$$

and characterizes the local deformation at a material point with position vector $\vec{X}$, assuming continuity. As

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

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

# Properties of deformation gradient

**Deformation gradient** $\mathbf{F}$ contains all the required local information about the changes in length, volumes and angles due to the deformation as follows:

- When vector $\vec{N}$ in the reference configuration is deformed into the vector $\vec{n}$, these vectors are related as: $\vec{n} = \mathbf{F}\vec{N}$
- The ratio between the local volume of the deformed configuration to the local volume in the reference configuration is equal to the determinant of the deformation gradient tensor: $J = \det \mathbf{F}$
- Two infinitesimal areas with $da$ and $dA$ being their magnitudes and $\vec{n}$ and $\vec{N}$ are unit vectors perpendicular to them, then the relationship is given by: $(da)\vec{n} = \det(\mathbf{F})(dA)\mathbf{F}^{-T}\vec{N}$
- An isochoric deformation is a deformation preserving local volume, i.e., $\det \mathbf{F} = 1$
- A deformation is called homogeneous if $\mathbf{F}$ is constant at every point. Otherwise, the deformation is called non-homogeneous
- The physical restriction of possible deformation: $\det \mathbf{F} > 0$

# Homogeneous deformation

A **homogeneous deformation** is one where the deformation gradient is uniform, i.e. independent of the coordinates, and the associated motion is termed **affine**. Every part of the material deforms as the whole does, and straight parallel lines in the reference configuration map to straight parallel lines in the deformed configuration.

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

or in matrix form:

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

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

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

where $\mathbf{F}$ 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

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

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

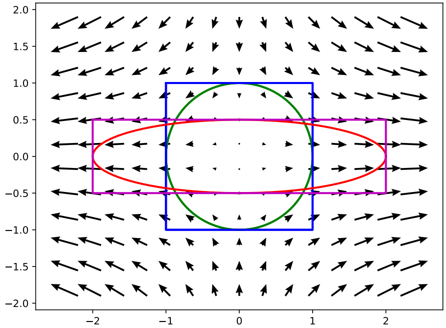
# Examples of pure shear

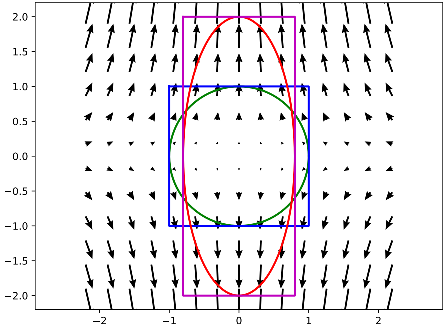
$$\mathbf{F} = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$\nabla \mathbf{u} = \begin{bmatrix} 1 & 0 \\ 0 & -0.5 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0.8 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\nabla \mathbf{u} = \begin{bmatrix} -0.2 & 0 \\ 0 & 1 \end{bmatrix}$$

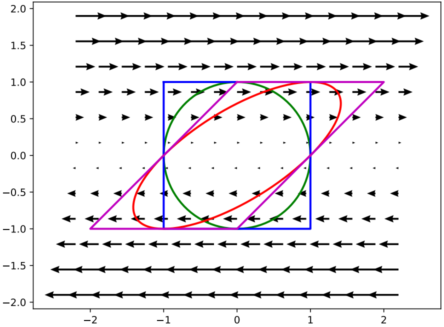


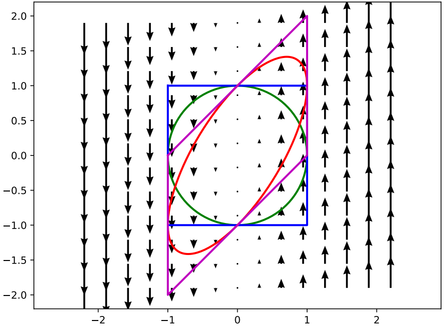



# Examples of simple shear

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

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



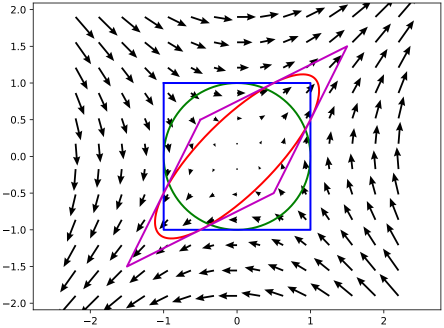
# Examples of general shear

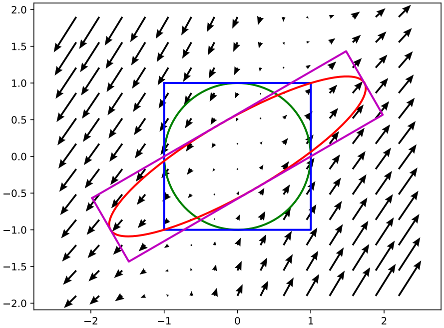
$$\mathbf{F} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$\nabla\mathbf{u} = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 1.732 & -0.25 \\ 1 & 0.433 \end{bmatrix}$$

$$\nabla\mathbf{u} = \begin{bmatrix} 0.732 & -0.25 \\ 1 & -0.567 \end{bmatrix}$$



# 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. Considering two successive deformations $\mathbf{F_1}$ and $\mathbf{F_2}$ write transformation equation....

# 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. Considering two successive deformations $\mathbf{F_1}$ and $\mathbf{F_2}$ write transformation equation....

$$\vec{x}_1 = \mathbf{F_1} \cdot \vec{X}$$

$$\vec{x}_2 = \mathbf{F_2} \cdot \vec{x}_1$$

Substitute first equation to second gives:

$$\vec{x}_2 = \mathbf{F_2} \cdot \mathbf{F_1} \cdot \vec{X}$$

so

$$\vec{x}_2 = \mathbf{F} \cdot \vec{X}$$

where

$$\mathbf{F} = \mathbf{F_2} \cdot \mathbf{F_1}$$

# Polar Decomposition I.

In last example the object has clearly been stretched and rotated. But by how much? the following two-step process of deformation followed by rigid body rotation gets you there...

$$\mathbf{F} = \begin{bmatrix} 1.732 & -0.25 \\ 1 & 0.433 \end{bmatrix} = \begin{bmatrix} \cos(30°) & -\sin(30°) \\ \sin(30°) & \cos(30°) \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$$

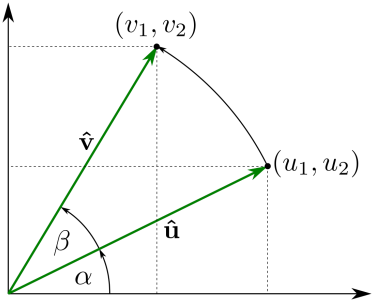
or

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U}$$

where $\mathbf{R}$ is the **rotation matrix**, and $\mathbf{U}$ is the **right stretch tensor** that is responsible for all the problems in life: stress, strain, fatigue, cracks, fracture, etc. Note that the process is read from right to left, not left to right. $\mathbf{U}$ is applied first, then $\mathbf{R}$.
This partitioning of the **deformation gradient** into the product of a **rotation matrix** and **stretch tensor** is known as a **polar decomposition**.

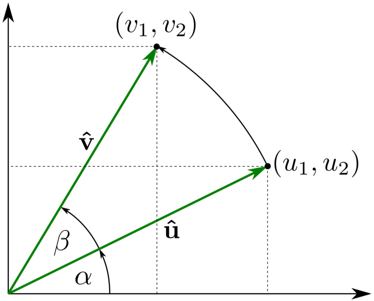
# Derivation of 2D rotation matrix

In order to rotate unit vector $\hat{\mathbf{u}} = (u_1, u_2)$ to vector $\hat{\mathbf{v}} = (v_1, v_2)$, we can write following equations for $\cos(\alpha)$, $\sin(\alpha)$, $\cos(\alpha + \beta)$ and $\sin(\alpha + \beta)$:



# Derivation of 2D rotation matrix

In order to rotate unit vector $\hat{\mathbf{u}} = (u_1, u_2)$ to vector $\hat{\mathbf{v}} = (v_1, v_2)$, we can write following equations for $\cos(\alpha)$, $\sin(\alpha)$, $\cos(\alpha + \beta)$ and $\sin(\alpha + \beta)$:




$$\cos(\alpha) = u_1, \ \sin(\alpha) = u_2$$

$$\cos(\alpha + \beta) = v_1, \ \sin(\alpha + \beta) = v_2$$

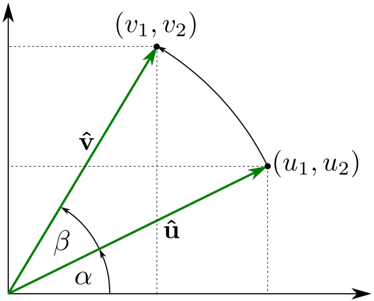
Substituting to the angle sum trigonometric identities:

$$\cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$$

$$\sin(\alpha + \beta) = \cos\alpha\sin\beta + \sin\alpha\cos\beta$$

# Derivation of 2D rotation matrix

In order to rotate unit vector $\hat{\mathbf{u}} = (u_1, u_2)$ to vector $\hat{\mathbf{v}} = (v_1, v_2)$, we can write following equations for $\cos(\alpha)$, $\sin(\alpha)$, $\cos(\alpha+\beta)$ and $\sin(\alpha+\beta)$:




$$\cos(\alpha) = u_1,\ \sin(\alpha) = u_2$$

$$\cos(\alpha+\beta) = v_1,\ \sin(\alpha+\beta) = v_2$$

Substituting to the angle sum trigonometric identities:

$$\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$$

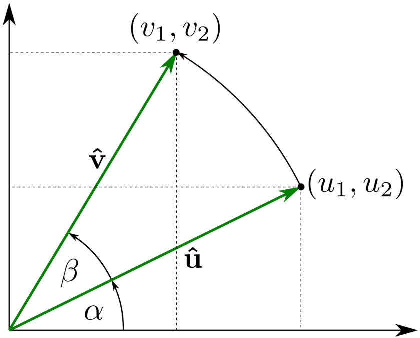
$$\sin(\alpha+\beta) = \cos\alpha\sin\beta + \sin\alpha\cos\beta$$

$$v_1 = u_1\cos\beta - u_2\sin\beta$$

$$v_2 = u_1\sin\beta + u_2\cos\beta$$

# Derivation of 2D rotation matrix

In order to rotate vector $\hat{\mathbf{u}}$ by angle $\beta$, we can use the **rotation matrix** $\mathbf{R}$:




$$v_1 = u_1 \cos\beta - u_2 \sin\beta$$

$$v_2 = u_1 \sin\beta + u_2 \cos\beta$$

or in matrix form

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

or

$$\hat{\mathbf{v}} = \mathbf{R} \cdot \hat{\mathbf{u}}$$

# Polar Decomposition II.

The deformed and rotated state could equally-well be arrived at by rotating it first, and then deforming it second. In this case, the reference configuration, is first rotated by the same $30^\circ$ angle to arrive at an intermediate configuration and then the intermediate configuration is deformed to arrive at the final, deformed state.
The deformation gradient can be written as:

$$\mathbf{F} = \begin{bmatrix} 1.732 & -0.25 \\ 1 & 0.433 \end{bmatrix} = \begin{bmatrix} 1.625 & 0.65 \\ 0.65 & 0.875 \end{bmatrix} \begin{bmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{bmatrix}$$

or

$$\mathbf{F} = \mathbf{V} \cdot \mathbf{R}$$

where $\mathbf{R}$ is the same **rotation matrix**, and $\mathbf{V}$ is the **left stretch tensor**.

# Polar Decomposition III.

It is relatively easy to develop a relationship between $\mathbf{V}$ and $\mathbf{U}$. Since $\mathbf{F} = \mathbf{V} \cdot \mathbf{R}$ and $\mathbf{F} = \mathbf{R} \cdot \mathbf{U}$, then

$$\mathbf{V} \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{U}$$

and post-multiplying through by $\mathbf{R}^T$ gives

$$\mathbf{V} \cdot \mathbf{R} \cdot \mathbf{R}^T = \mathbf{R} \cdot \mathbf{U} \cdot \mathbf{R}^T$$

But since $\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$, this leaves

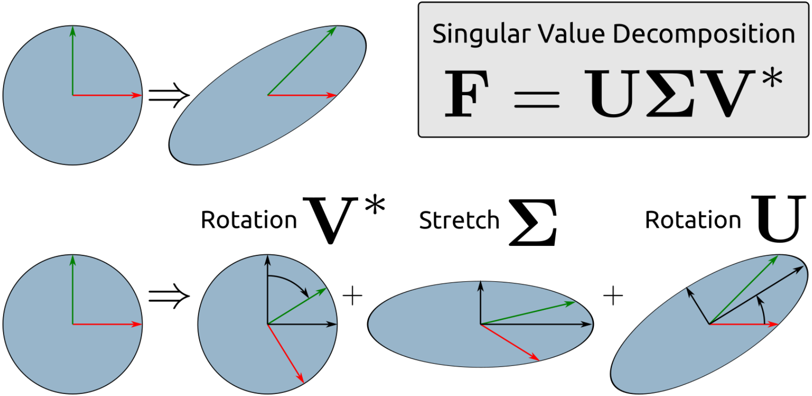
$$\mathbf{V} = \mathbf{R} \cdot \mathbf{U} \cdot \mathbf{R}^T$$

as the relationship between $\mathbf{V}$ and $\mathbf{U}$. Alternatively, solving for $\mathbf{U}$ gives

$$\mathbf{U} = \mathbf{R}^T \cdot \mathbf{V} \cdot \mathbf{R}$$

# Singular value decomposition

In linear algebra, the singular value decomposition (**SVD**) is a factorization of a real or complex matrix $\mathbf{F} = \mathbf{U\Sigma V}^*$. Thus the expression $\mathbf{U\Sigma V}^*$ can be intuitively interpreted as a composition of three geometrical transformations: a **rotation or reflection**, a **scaling**, and another **rotation or reflection**.



# Python exercise

Calculate orientation and axial ratio of strain ellipse for deformation gradient $\mathbf{F}$ using SVD.

```
>>> from pylab import *

>>> F = array([[1, 1], [0, 1]])  # deformation gradient

>>> # calculate singular value decomposition
>>> U, s, V = svd(F)
>>> # calculate axial ratio and orientation
>>> ar = s[0]/s[1]
>>> ori = degrees(arctan2(U[1, 0], U[0, 0]))
>>> print('Orientation:{:g} AR:{:g}'.format(ori, ar))
Orientation:31.7175 AR:2.61803
```

# Detour on conic sections

In mathematics, a **conic section** is a curve obtained as the intersection of the surface of a cone with a plane. The three types of conic section are the **hyperbola**, the **parabola**, and the **ellipse**. The circle is a special case of the ellipse sometimes called a fourth type of conic section. Conic sections are the sets of points whose coordinates satisfy a second-degree polynomial equation:

$$Q(x,y) = Ax^2 + 2Bxy + Cy^2 + Dx + Ey + F = 0$$

can be written in matrix notation as:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} D & E \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + F = 0$$

The the first three terms $Ax^2 + 2Bxy + Cy^2$ is the **quadratic form** associated with the equation and defined by **matrix of the quadratic form**.

$$Ax^2 + 2Bxy + Cy^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^T \mathbf{A_Q} x$$

Note that $Q$ is an ellipse if and only if $\det \mathbf{A_Q} > 0$.

# Strain ellipse or ellipsoid

According to definition, the **strain ellipse** results from transformation of unit circle, which in matrix form is given by equation:

$$\begin{bmatrix} X & Y \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = X^T X = 1$$

The deformation gradient equation could be written in terms of deformed coordinates as:

$$X = \mathbf{F}^{-1} x$$

Substituting into equation of unit circle we obtain:

$$(\mathbf{F}^{-1} x)^T \cdot \mathbf{F}^{-1} x = x^T (\mathbf{F}^{-1})^T \cdot \mathbf{F}^{-1} x = x^T (\mathbf{F} \cdot \mathbf{F}^T)^{-1} x = x^T \mathbf{B}^{-1} x = 1$$

where matrix $\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T$ is called **Finger** or **Left Cauchy-Green** deformation tensor. It's inverse $\mathbf{B}^{-1}$ represents ellipse or ellipsoid and is commonly called **ellipsoid tensor** or **Cauchy deformation tensor**.

# Reciprocal ellipse or ellipsoid

According to definition, the **reciprocal ellipse** is transformed to unit circle, which in matrix form is given by equation:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^T x = 1$$

Substituting equation for deformation gradient $x = \mathbf{F}X$ into equation of unit circle in deformed coordinates we obtain:

$$(\mathbf{F}X)^T \cdot \mathbf{F}X = X^T \mathbf{F}^T \cdot \mathbf{F}X = X^T \mathbf{C}X = 1$$

where matrix $\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}$ is called **Green's** or **Right Cauchy-Green** deformation tensor.

# Deformation tensors and polar decomposition

Plugging the polar decomposition into equations for above defined deformation tensors gives a rather surprising result. Using $\mathbf{F} = \mathbf{V} \cdot \mathbf{R}$:

$$\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T = (\mathbf{V} \cdot \mathbf{R}) \cdot (\mathbf{V} \cdot \mathbf{R})^T = \mathbf{V} \cdot \mathbf{R} \cdot \mathbf{R}^T \cdot \mathbf{V}^T = \mathbf{V} \cdot \mathbf{V}^T$$

Using $\mathbf{F} = \mathbf{R} \cdot \mathbf{U}$:

$$\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} = (\mathbf{R} \cdot \mathbf{U})^T \cdot (\mathbf{R} \cdot \mathbf{U}) = \mathbf{U}^T \cdot \mathbf{R}^T \cdot \mathbf{R} \cdot \mathbf{U} = \mathbf{U}^T \cdot \mathbf{U}$$

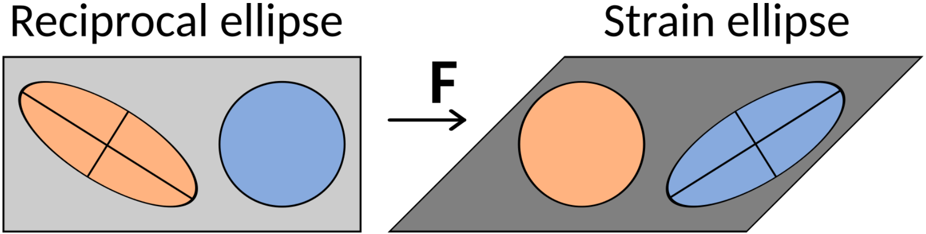
As $\mathbf{U}$ and $\mathbf{V}$ are both symmetric, so $\mathbf{U} = \mathbf{U}^T$ and $\mathbf{U}^T \cdot \mathbf{U} = \mathbf{U} \cdot \mathbf{U}$. Likewise $\mathbf{V} = \mathbf{V}^T$ and $\mathbf{V} \cdot \mathbf{V}^T = \mathbf{V} \cdot \mathbf{V}$. Finally, $\mathbf{U} \cdot \mathbf{U}$ is sometimes written as $\mathbf{U}^2$ and $\mathbf{V} \cdot \mathbf{V}$ is sometimes written as $\mathbf{V}^2$. Therefore:

$$\mathbf{B} = \mathbf{V}^2 \text{ and } \mathbf{C} = \mathbf{U}^2$$

The surprising result here is that the rotation matrix, $\mathbf{R}$, has been eliminated from the problem in both cases.

# Properties of strain and reciprocal deformation

The eigenvectors of $\mathbf{B}$ define orientation of principal axes of the **strain ellipse/ellipsoid** in deformed state. The eigenvalues are quadratic elongations along principal directions, i.e. the lengths of semi-axes of the strain ellipse or ellipsoid are the square roots of the corresponding eigenvalues.




The eigenvectors of $\mathbf{C}$ define orientation of principal axes of the **reciprocal ellipse/ellipsoid** in undeformed state. The eigenvalues are quadratic elongations along principal directions, i.e. the lengths of semi-axes of the reciprocal ellipse or ellipsoid are the square roots of the corresponding eigenvalues.

# Actions of strain tensors

Consider two vectors in the reference configuration $X_1$ and $X_2$ which are mapped into the vectors $x_1$ and $x_2$ in the current configuration. Than for dot product:

$$x_1 \cdot x_2 = x_1^T x_2 = (\mathbf{F} X_1)^T \cdot \mathbf{F} X_2 = X_1^T \mathbf{F}^T \cdot \mathbf{F} X_2 = X_1^T \mathbf{C} X_2$$

Similarly for current configuration:

$$X_1 \cdot X_2 = X_1^T X_2 = (\mathbf{F}^{-1} x_1)^T \cdot \mathbf{F}^{-1} x_2 = x_1^T (\mathbf{F} \cdot \mathbf{F}^T)^{-1} x_2 = x_1^T \mathbf{B}^{-1} x_2$$

For dot product of vector with itself we can write:

$$x \cdot x = x^T x = (\mathbf{F} X)^T \cdot \mathbf{F} X = X^T \mathbf{F}^T \cdot \mathbf{F} X = X^T \mathbf{C} X$$

Similarly for current configuration:

$$X \cdot X = X^T X = (\mathbf{F}^{-1} x)^T \cdot \mathbf{F}^{-1} x = x^T (\mathbf{F} \cdot \mathbf{F}^T)^{-1} x = x^T \mathbf{B}^{-1} x$$

# Quadratic elongation

**Quadratic elongation** $\lambda$ (square of the stretch) is defined as the square of ratio of the length of a deformed line element to the length of the corresponding undeformed line element:

$$\lambda = S^2 = \frac{|x|^2}{|X|^2}$$

As dot product of vector with itself is $x \cdot x = x^T x = |x|\ |x| \cos(0) = |x|^2$, the above equation could be written as:

$$\lambda_X = \frac{X^T \mathbf{C} X}{|X|^2} = \hat{X}^T \mathbf{C} \hat{X}$$

Similarly for $X^T X = |X|^2$:

$$\lambda_x^{-1} = \frac{x^T \mathbf{B}^{-1} x}{|x|^2} = \hat{x}^T \mathbf{B}^{-1} \hat{x}$$

where $\hat{X} = \frac{X}{|X|}$ and $\hat{x} = \frac{x}{|x|}$ are unit vector in the directions of $X$ and $x$.

# Change of angle I.

The change in angle between any two vectors may also be given in terms of stretch. Let $X_1$ and $X_2$ be arbitrary vectors which become $x_1$ and $x_2$, respectively, during a deformation. By the dot product,

$$x_1 \cdot x_2 = x_1^T x_2 = |x_1| \ |x_2| \cos(\theta)$$

we may compute the angle $\theta$ between $x_1$ and $x_2$ from its cosine

$$\cos(\theta) = \frac{x_1 \cdot x_2}{|x_1| \ |x_2|} = \frac{X_1^T \mathbf{C} X_2}{|x_1| \ |x_2|} = \frac{X_1^T \mathbf{C} X_2}{\sqrt{X_1^T \mathbf{C} X_1}\sqrt{X_2^T \mathbf{C} X_2}}$$

Dividing both the numerator and denominator by the $|X_1| \, |X_2|$ we got:

$$\cos(\theta) = \frac{\hat{X_1}^T \mathbf{C} \hat{X_2}}{S_{X_1} S_{X_2}}$$

where $\hat{X}_1 = \frac{X_1}{|X_1|}$, $\hat{X}_2 = \frac{X_2}{|X_2|}$ are unit vector in the directions of $X_1$, $X_2$.

# Change of angle II.

Similarly for current configuration

$$X_1 \cdot X_2 = X_1^T X_2 = |X_1| \; |X_2| \cos(\phi)$$

we may compute the angle $\phi$ between $X_1$ and $X_2$ from its cosine

$$\cos(\phi) = \frac{X_1 \cdot X_2}{|X_1| \; |X_2|} = \frac{x_1^T \mathbf{B}^{-1} x_2}{|X_1| \; |X_2|} = \frac{x_1^T \mathbf{B}^{-1} x_2}{\sqrt{x_1^T \mathbf{B}^{-1} x_1} \sqrt{x_2^T \mathbf{B}^{-1} x_2}}$$

Dividing both the numerator and denominator by the $|x_1| \, |x_2|$ we got:

$$\cos(\phi) = \frac{\hat{x_1}^T \mathbf{B}^{-1} \hat{x_2}}{S_{x_1}^{-1} S_{x_2}^{-1}} = S_{x_1} S_{x_2} \hat{x_1}^T \mathbf{B}^{-1} \hat{x_2}$$

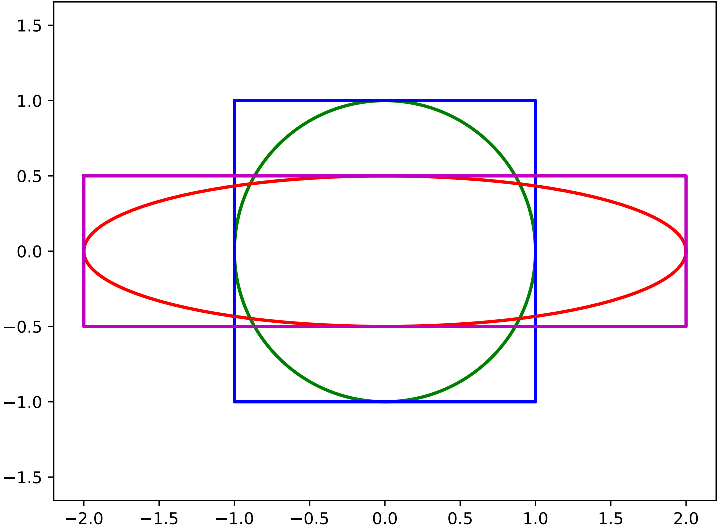
where $\hat{x_1} = \frac{x_1}{|x_1|}$, $\hat{x_2} = \frac{x_2}{|x_2|}$ are unit vector in the directions of $x_1$, $x_2$.

# Python exercise

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

```
# parametric definition of unit circle
theta = linspace(0, 2*pi, 300)
c = cos(theta), sin(theta)
s = [-1, 1, 1, -1, -1], [1, 1, -1, -1, 1]
plot(c[0], c[1], 'g', s[0], s[1], 'b', lw=2)
# Apply deformation gradient and plot ellipse
F = array([[2, 0], [0, 0.5]])
e = dot(F, c)
q = dot(F, s)
plot(e[0], e[1], 'r', q[0], q[1], 'm', lw=2)
axis('equal')
```

# Python exercise



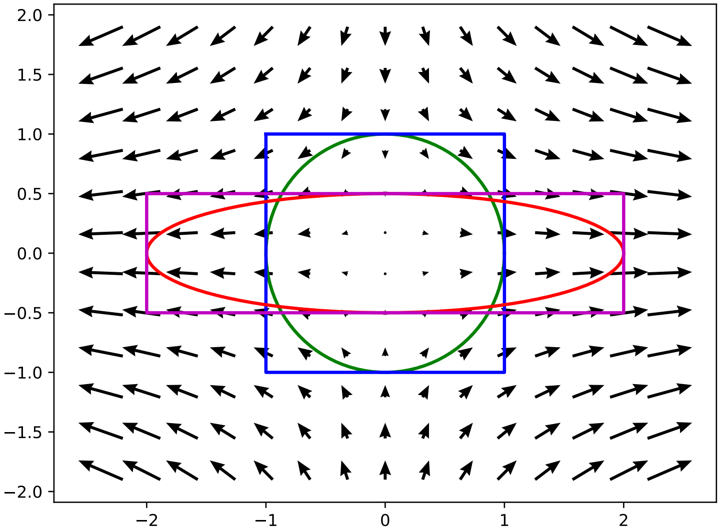
# Python exercise

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

```
# create rectangular grid
X = meshgrid(linspace(-2.2, 2.2, 15),
             linspace(-1.9, 1.9, 12))

# calculate displacements
J = F - eye(2)
u = tensordot(J, X, axes=1)
# plot
quiver(X[0], X[1], u[0], u[1],
       angles='xy', lw=0.5, headwidth=4)
plot(c[0], c[1], 'g', s[0], s[1], 'b',
     e[0], e[1], 'r', q[0], q[1], 'm', lw=2)
axis('equal')
```

# Python exercise



# Python exercise

Find orientation and axial ratio of strain ellipse for deformation gradient $\mathbf{F}$.

```
>>> from pylab import *
>>> # deformation gradient
>>> F = array([[1, 1], [0, 1]])
>>> # calculate Left Cauchy-Green deformation tensor
>>> B = dot(F, F.T)
>>> s, U = eig(B)
>>> # calculate axial ratio and orientation
>>> ar = sqrt(s[0]/s[1])
>>> ori = degrees(arctan2(U[1, 0], U[0, 0]))
>>> print('Orientation:{:g} AR:{:g}'.format(ori, ar))
Orientation:31.7175 AR:2.61803
```

# Superposed strain

Similarly, we can obtain ellipse or ellipsoid equation resulting from two superposed deformation. When first deformation $\mathbf{F_1}$ results in intermediate ellipse or ellipsoid $\mathbf{B_1}$, i.e.:

$$x^T\mathbf{B_1}^{-1}x = 1$$

then substituting equation for deformation $\mathbf{D_2}$ we obtain:

$$x^T\mathbf{F_2}^{-T}\mathbf{B_1}^{-1}\mathbf{F_2}^{-1}x = x^T\mathbf{B_{12}}^{-1}x1$$

where $\mathbf{B_{12}}^{-1}$ is:

$$\begin{aligned}\mathbf{B_{12}}^{-1} &= \mathbf{F_2}^{-T}\mathbf{B_1}^{-1}\mathbf{F_2}^{-1}\\ &= \mathbf{F_2}^{-T}\mathbf{F_1}^{-T}\mathbf{F_1}^{-1}\mathbf{F_2}^{-1}\\ &= (\mathbf{F_2}\mathbf{F_1})^{-T}(\mathbf{F_2}\mathbf{F_1})^{-1}\\ &= \mathbf{F_{12}}^{-T}\mathbf{F_{12}}^{-1}\end{aligned}$$

where $\mathbf{F_{12}} = \mathbf{F_2}\mathbf{F_1}$. Consequently $\mathbf{B_{12}}$ could be written as

$$\mathbf{B_{12}} = \mathbf{F_{12}}\mathbf{F_{12}}^T$$

# Python exercise - our first library

Function to plot unit circle and strain ellipse from deformation gradient:

```
import numpy as np
import matplotlib.pyplot as plt

def def_ellipse(F):
    # Draw ellipse from deformation gradient
    theta = np.linspace(0, 2*np.pi, 180)
    c = np.cos(theta), np.sin(theta)
    e = np.dot(F, c)
    plt.plot(c[0], c[1], 'r', e[0], e[1], 'g')
    plt.axis('equal')
```

# Python exercise

Function to visualize displacement field from displacement gradient:

```
def dis_field(J):
    # Visualize displacement gradient
    X = np.meshgrid(np.linspace(-3, 3, 21),
                    np.linspace(-2, 2, 17))
    u = u = tensordot(J, X, axes=1)
    plt.quiver(X[0], X[1], u[0], u[1], angles='xy')
```

# Python exercise

Function to plot unit circle and strain ellipse from displacement gradient:

```
def dis_ellipse(J):
    # Draw ellipse from displacement gradient
    F = np.asarray(J) + np.eye(J.ndim)
    def_ellipse(F)
```

Function to visualize displacement field from deformation gradient:

```
def def_field(F):
    # Visualize deformation gradient
    J = np.asarray(F) - np.eye(F.ndim)
    dis_field(J)
```

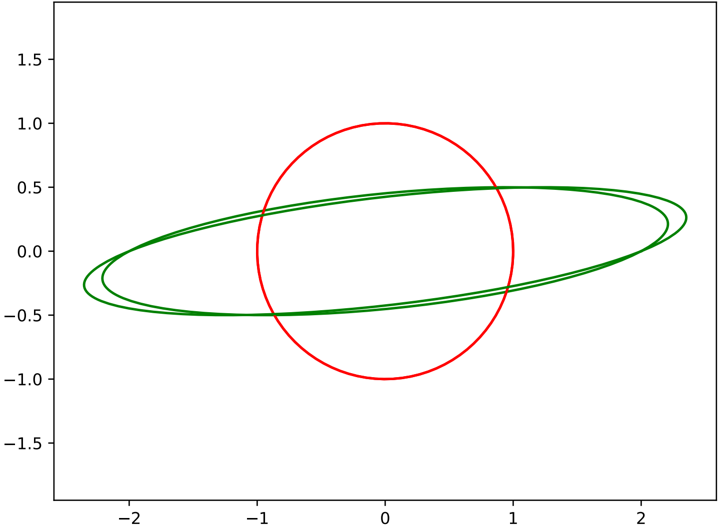
# Python exercise

Script to simulate simultaneous simple and pure shear by mutual combination of incremental deformations:

```
n = 5  # number of increments
SS = np.array([[1, 1./n], [0, 1]])  # simple shear increment
PS = np.array([[2**(1./n), 0], [0, 0.5**(1/n)]])  # pure shear increment
# initial stage
F1 = F2 = np.array([[1,0],[0,1]])
# both orders of superpositions of increments
for i in range(n):
    F1 = np.dot(SS, np.dot(PS, F1))
    F2 = np.dot(PS, np.dot(SS, F2))

def_ellipse(F1)
def_ellipse(F2)
```

# Python exercise



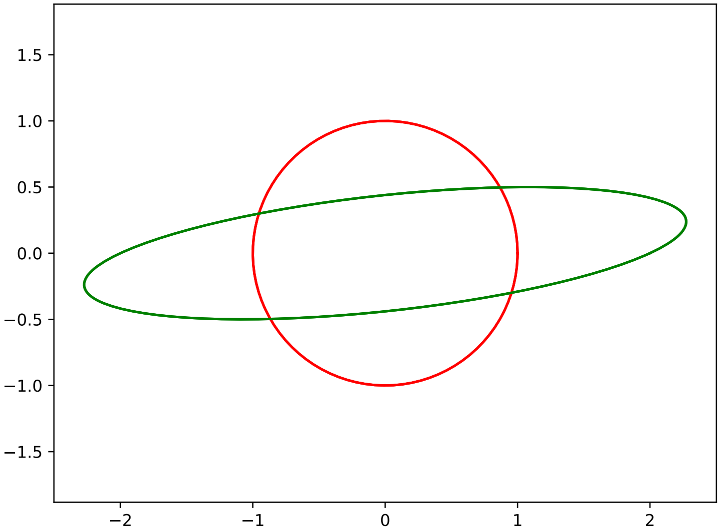
# Python exercise

Script to simulate simultaneous simple and pure shear by mutual combination of incremental deformations:

```
n = 500  # number of increments
SS = np.array([[1, 1./n], [0, 1]])  # simple shear increment
PS = np.array([[2**(1./n), 0], [0, 0.5**(1/n)]])  # pure shear increment
# initial stage
F1 = F2 = np.array([[1,0],[0,1]])
# both orders of superposition
for i in range(n):
    F1 = np.dot(SS, np.dot(PS, F1))
    F2 = np.dot(PS, np.dot(SS, F2))

def_ellipse(F1)
def_ellipse(F2)
```

# Python exercise



# Volume change and area change I.

Consider an infinitesimal volume element $(dX_1, dX_2, dX_3)$ in the reference configuration $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}$
so the vectors representing the edges of the element are $d\mathbf{X}_1 = dX_1\ \mathbf{E}_1$ ; $d\mathbf{X}_2 = dX_2\ \mathbf{E}_2$ ; $d\mathbf{X}_3 =$
$dX_3\ \mathbf{E}_3$. The volume of the element is given by triple product of it's edges:

$$dV = d\mathbf{X}_1 \cdot (d\mathbf{X}_2 \times d\mathbf{X}_3) = dX_1\ dX_2\ dX_3\ \mathbf{E}_1 \cdot (\mathbf{E}_2 \times \mathbf{E}_3) = dX_1\ dX_2\ dX_3$$

Upon deformation, these edges go to $(d\mathbf{X}_1, d\mathbf{X}_2, d\mathbf{X}_3)$ where:

$$d\mathbf{x}_1 = \mathbf{F} \cdot d\mathbf{X}_1 = \left[\frac{\partial x_i}{\partial X_j}\right] \cdot d\mathbf{X}_1 = dX_1 \left[\frac{\partial x_i}{\partial X_j}\right] \cdot \mathbf{E}_1 = dX_1 \left[\frac{\partial x_i}{\partial X_1}\right]$$

$$d\mathbf{x}_2 = \mathbf{F} \cdot d\mathbf{X}_2 = \left[\frac{\partial x_i}{\partial X_j}\right] \cdot d\mathbf{X}_2 = dX_2 \left[\frac{\partial x_i}{\partial X_j}\right] \cdot \mathbf{E}_2 = dX_2 \left[\frac{\partial x_i}{\partial X_2}\right]$$

$$d\mathbf{x}_3 = \mathbf{F} \cdot d\mathbf{X}_3 = \left[\frac{\partial x_i}{\partial X_j}\right] \cdot d\mathbf{X}_3 = dX_3 \left[\frac{\partial x_i}{\partial X_j}\right] \cdot \mathbf{E}_3 = dX_3 \left[\frac{\partial x_i}{\partial X_3}\right]$$

# Volume change and area change II.

The deformed volume is given by triple product of deformed edges:

$$dv = d\mathbf{x}_1 \cdot (d\mathbf{x}_2 \times d\mathbf{x}_3) = \left[\frac{\partial x_i}{\partial X_1}\right] \cdot \left(\left[\frac{\partial x_i}{\partial X_2}\right] \times \left[\frac{\partial x_i}{\partial X_3}\right]\right) \; dX_1 \; dX_2 \; dX_3$$

Knowing that scalar triple product can also be understood as the determinant of the 3×3 matrix also known as **Jacobian of the deformation gradient**:

$$\left[\frac{\partial x_i}{\partial X_1}\right] \cdot \left(\left[\frac{\partial x_i}{\partial X_2}\right] \times \left[\frac{\partial x_i}{\partial X_3}\right]\right) = \det\left(\left[\frac{\partial x_i}{\partial X_j}\right]\right) = \det\left(\mathbf{F}\right) = J$$

Therefore:

$$dv = \det\left(\mathbf{F}\right) \; dV = J \; dV$$

# Area change - Nanson's formula

**Nanson's formula** is an important relation that can be used to go from areas in the current configuration to areas in the reference configuration and vice versa. Using area elements ($d\mathbf{A} = dA\ \mathbf{N};\ d\mathbf{a} = da\ \mathbf{n}$) and above derived equation for volume change, we can write:

$$
\begin{aligned}
dv &= J\ dV \\
d\mathbf{l} \cdot d\mathbf{a} &= J\ d\mathbf{L} \cdot d\mathbf{A} \\
\mathbf{F} \cdot d\mathbf{L} \cdot d\mathbf{a} &= J\ d\mathbf{L} \cdot d\mathbf{A} \\
d\mathbf{L} \cdot (\mathbf{F}^T \cdot d\mathbf{a}) &= d\mathbf{L} \cdot J\ d\mathbf{A} \\
\mathbf{F}^T \cdot d\mathbf{a} &= J\ d\mathbf{A} \\
d\mathbf{a} &= J\ \mathbf{F}^{-T} \cdot d\mathbf{A} \\
da\ \mathbf{n} &= \det(\mathbf{F})\ dA\ \mathbf{F}^{-T} \cdot \mathbf{N}
\end{aligned}
$$

where $da$ is an area of a region in the current configuration, $dA$ is the same area in the reference configuration, and $\mathbf{n}$ is the outward normal to the area element in the current configuration while $\mathbf{N}$ is the outward normal in the reference configuration.

# Python exercise

Script to volume change and area changes in principal coordinate planes:

```
>>> F = array([[1, 0, 1],[0, 2, 0],[0, 0, 0.5]])
>>> print('Volume change: {:.0%}'.format(det(F) - 1))
Volume change: 0%
>>> dA1, dA2, dA3 = eye(3)  # unit length basis vectors
>>> da = lambda n: norm(dot(det(F)*inv(F).T, n))
>>> tmpl = 'Area change within plane {}: {:.0%}'
>>> print(tmpl.format(dA1, da(dA1) - 1))
Area change within plane [1. 0. 0.]: 124%
>>> print(tmpl.format(dA2, da(dA2) - 1))
Area change within plane [0. 1. 0.]: -50%
>>> print(tmpl.format(dA3, da(dA3) - 1))
Area change within plane [0. 0. 1.]: 100%
```

# Velocity gradient I.

Velocity gradients are absolutely essential to analyses involving path dependent materials and are useful to better understanding deformations. The **velocity gradient** is to velocities what the deformation gradient is to displacements.The **velocity gradient** is used as a measure of the rate at which a material is deforming. Consider two fixed neighbouring points, $x$ and $x + dx$. The velocities of the material particles at these points at any given time instant are $\mathbf{v}(x)$ and $\mathbf{v}(x + dx)$, and

$$\mathbf{v}(x + dx) = \mathbf{v}(x) + \frac{\partial \mathbf{v}}{\partial x} dx$$

The relative velocity between the points is

$$d\mathbf{v} = \frac{\partial \mathbf{v}}{\partial x} dx = \mathbf{L} dx$$

where $\mathbf{L}$ is defined to be the **spatial velocity gradient**.

# Velocity gradient II.

The **spatial velocity gradient** $\mathbf{L}$ is defined as:

$$\mathbf{L} = \frac{\partial v_i}{\partial x_j} = \begin{bmatrix} \frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial x_2} & \frac{\partial v_1}{\partial x_3} \\ \frac{\partial v_2}{\partial x_1} & \frac{\partial v_2}{\partial x_2} & \frac{\partial v_2}{\partial x_3} \\ \frac{\partial v_3}{\partial x_1} & \frac{\partial v_3}{\partial x_2} & \frac{\partial v_3}{\partial x_3} \end{bmatrix}$$

Note that the derivative is with respect to the current coordinates, $x$, not the reference coordinates $X$.

# Velocity gradient III.

Calculations that involve the time-dependent deformation of a body often require a time derivative of the deformation gradient to be calculated. The time derivative of $\mathbf{F}$ is:

$$\dot{\mathbf{F}} = \frac{\partial \mathbf{F}}{\partial t} = \frac{\partial}{\partial t}\frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial}{\partial \mathbf{X}}\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial}{\partial \mathbf{X}}\boldsymbol{v} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{X}}$$

where $\boldsymbol{v}$ is the velocity in reference frame and the derivative on the right hand side represents a *material velocity gradient*. It is common to convert that into a spatial gradient applying the chain rule to the above result, i.e.,

$$\dot{\mathbf{F}} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \mathbf{L} \cdot \mathbf{F}$$

where $\mathbf{L}$ is the *spatial velocity gradient*. Post multiplying both sides by $\mathbf{F}^{-1}$ gives the equation for $\mathbf{L}$ in terms of $\mathbf{F}$:

$$\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}$$

# Velocity gradient IV.

If the **spatial velocity gradient** is constant, the above equation can be solved exactly using *matrix exponential*:

$$\mathbf{F} = \mathrm{e}^{\mathbf{L}t}$$

where $t$ is time. Similarly, the spatial velocity gradient could be calculated from deformation gradient using *matrix logarithm*:

$$\mathbf{L} = ln(\mathbf{F})$$

We can use these equation to calculate deformation gradient for any intermediate deformation. When deformation in time $t = 1$ is defined by $\mathbf{F}$, than for any time $t = [0, 1]$:

$$\mathbf{F}(t) = \mathrm{e}^{ln(\mathbf{F})t}$$

# Decomposition of velocity gradient

The **spatial velocity gradient** can be decomposed into symmetric and antisymmetric parts as follows.

$$\mathbf{L} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right) + \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^T\right) = \mathbf{D} + \mathbf{W}$$

The first symmetric term is called **rate of deformation tensor** $(\mathbf{D})$, while second antisymmetric term is known as **spin tensor** $(\mathbf{W})$.

$$\mathbf{D} = \begin{bmatrix} \dot{\varepsilon}_{xx} & \dot{\gamma}_{xy} & \dot{\gamma}_{xz} \\ \dot{\gamma}_{xy} & \dot{\varepsilon}_{yy} & \dot{\gamma}_{yz} \\ \dot{\gamma}_{xz} & \dot{\gamma}_{xz} & \dot{\varepsilon}_{zz} \end{bmatrix}$$

where 6 independent components are referred as **stretching and shearing strain rates**.

$$\mathbf{W} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3) = \frac{d\phi}{dt}\boldsymbol{u}$ is the angular velocity vector defining rate of rotation $\frac{d\phi}{dt}$ about an axis $\boldsymbol{u}$.

# Python exercise

Lets visualize symmetric and antisymmetric parts of L for simple shear:

```
import scipy.linalg as la
X = meshgrid(linspace(-2, 2, 15), linspace(-2, 2, 15))
F = array([[1, 1], [0, 1]])
L = la.logm(F)
D, W = (L + L.T)/2, (L - L.T)/2
f, (ax1, ax2) = plt.subplots(1,2,figsize=(10, 6))
u = tensordot(D, X, axes=1)
ax1.quiver(X[0], X[1], u[0], u[1], angles='xy')
ax1.set_aspect('equal', adjustable='box')
ax1.set_title('rate of deformation tensor')
u = tensordot(W, X, axes=1)
ax2.quiver(X[0], X[1], u[0], u[1], angles='xy')
ax2.set_aspect('equal', adjustable='box')
ax2.set_title('spin tensor')
```

# Python exercise

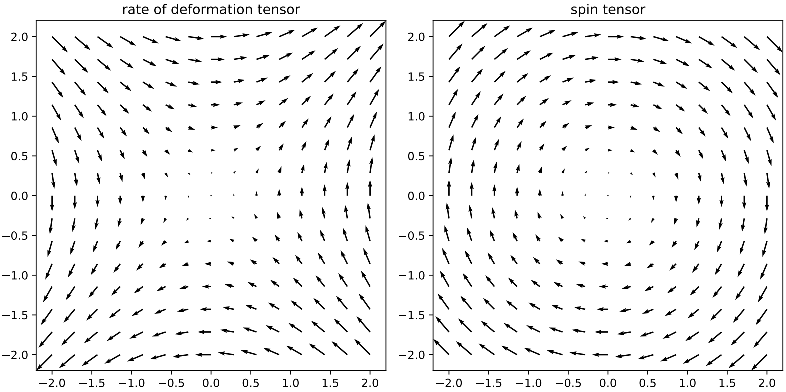



Figure: Velocity field for symmetric and antisymmetric parts of $\mathbf{L}$ for simple shear with $\gamma = 1$.

# Python exercise

Script to plot evolution of axial ratio of finite strain ellipse during progressive deformation:

```
gdot = 1e-14
yearsec = 365.25*24*3600
times = linspace(0, 5, 100)[1:]
ar = []
L = array([[0, gdot], [0, 0]])
for t in times:
    F = la.expm(L*t*1e6*yearsec)
    U, s, V = svd(F)
    ar.append(s.max()/s.min())

plot(times, ar)
xlabel('Time [Ma]')
ylabel('Axial ratio')
```

# Python exercise

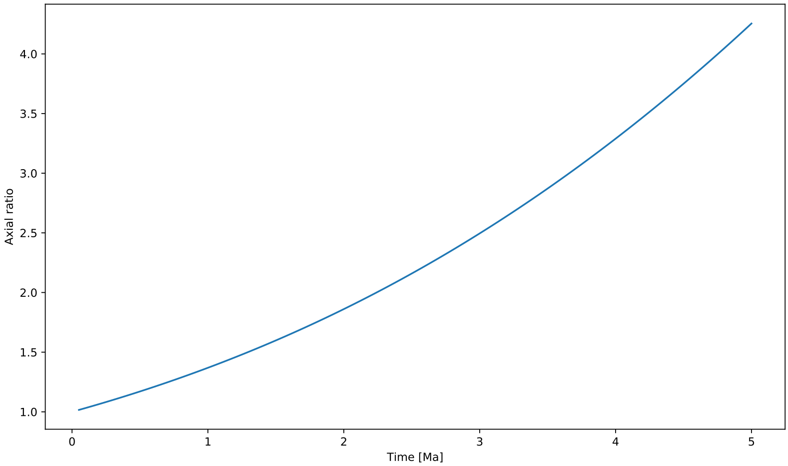



Figure: Axial ratio of finite strain ellipse during simple shear.

# Simultaneous deformation

As velocity gradient represents vector field, the velocity gradient of simultaneous deformation could be described as addition of individual velocity gradients, i.e:

$$\mathbf{L} = \mathbf{L_1} + \mathbf{L_2}$$

and finite deformation could be calculated as:

$$\mathbf{F} = \mathrm{e}^{\mathbf{L_1} + \mathbf{L_2}}$$

In case we know individual deformation gradients, the simultaneous deformation could be calculated as:

$$\mathbf{F} = \mathrm{e}^{ln(\mathbf{F_1}) + ln(\mathbf{F_2})}$$

# Python exercise

Calculate deformation gradient for simultaneous pure shear with $S_x = 2$ and $S_y = 0.5$ and simple shear with $\gamma = 1$:

```
>>> F1 = array([[2, 0], [0, 0.5]])
>>> F2 = array([[1, 1], [0, 1]])

>>> F = la.expm(la.logm(F1) + la.logm(F2))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'la' is not defined
>>> print(F)
[[1.  0.  1. ]
 [0.  2.  0. ]
 [0.  0.  0.5]]
```

# Python exercise

Script to plot evolution of axial ratio of finite strain ellipse during progressive simultaneous pure shear and simple shear:

```
edot = 1e-14
ar = []
L = array([[edot, gdot],[0, -edot]])
for t in times:
    F = la.expm(L*t*1e6*yearsec)
    U, s, V = svd(F)
    ar.append(s.max()/s.min())

plot(times, ar)
xlabel('Time [Ma]')
ylabel('Axial ratio')
```

# Python exercise

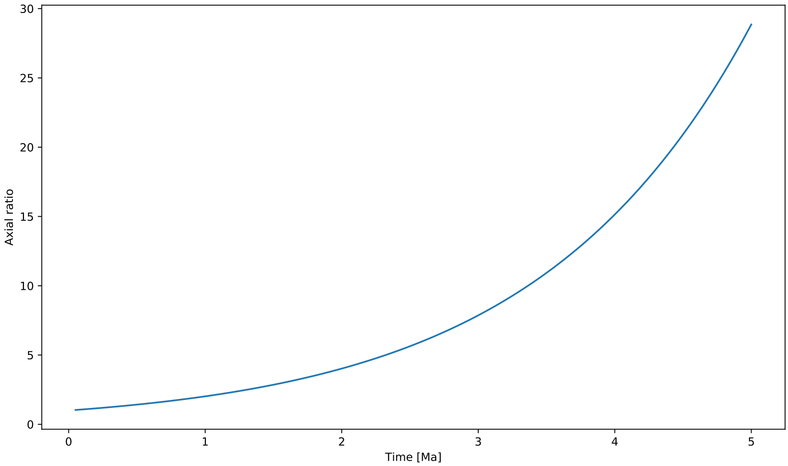



Figure: Axial ratio of progressive simultaneous deformation.

# Python exercise

Plot evolution of deformation intensity (D) and symmetry (K) within transpression zone defined by convergence angle $\alpha = 30°$, $\dot{\gamma}_{xy} = 10^{-14}s{-}1$, $\dot{e}_{yy} = -10^{-14}s{-}1$ and $\dot{e}_{zz} = 10^{-14}s{-}1$ during 5Ma years.

# Python exercise

```
K, D = [], []
L = array([[0, gdot, 0 ],
           [0,-edot, 0 ],
           [0,  0, edot]])

for t in times:
    F = la.expm(L * t * 1e6*yearsec)
    U, s, V = svd(F)
    xy, yz = s[:-1]/s[1:] - 1
    K.append(xy/yz)
    D.append(sqrt(xy**2 + yz**2))

plot(times, D, times, K)
xlabel('Time [Ma]')
ylabel('D and K')
```

# Python exercise

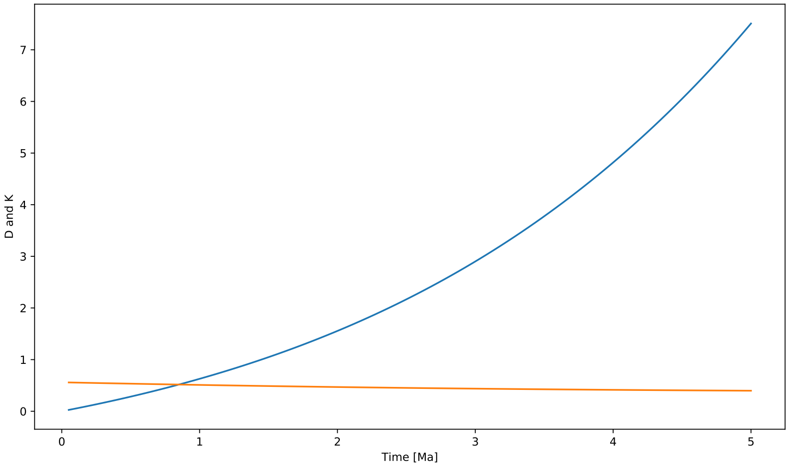



Figure: Intensity (D) and symmetry (K) during transpression.