TensorsCalculator Documentation

Release 1.0

Olga Kononova, Valeri Barsegov

Sep 10, 2017

Contents

Theoretical background

Notations

- $\mathbf{X}$ - material coordinates of a particle in reference configuration;
- $\mathbf{x}$ - material coordinates of a particle in deformed configuration;
- $\chi(\mathbf{X}, t)$ - mapping from reference configuration to a deformed configuration;
- α, β, γ - vectors or tensors components (x, y, z), superscripts;
- i, j - particle number, subscript;
- $\mathbf{T}$ - force vector;
- A - area;
- $\mathbf{F} = \{F^{\alpha\beta}\}$ - deformation gradient tensor;
- $J = det\mathbf{F}$ - determinant of deformation gradient tensor, change in unit volume;
- $\mathbf{C} = \{C^{\alpha\beta}\}$ - Green's (Cauchy-Green) deformation tensor;
- $\mathbf{E} = \{E^{\alpha\beta}\}$ - strain tensor;
- $\mathbf{U} = \{U^{\alpha\beta}\}$ - right stretch tensor;
- $\mathbf{R} = \{R^{\alpha\beta}\}$ - rotation tensor;
- $\sigma = \{\sigma^{\alpha\beta}\}$ - stress tensor (Cauchy);
- $\mathbf{P} = \{P^{\alpha\beta}\}$ - first Piola-Kirchhoff stress tensor;
- $\mathbf{S} = \{S^{\alpha\beta}\}$ - second Piola-Kirchhoff stress tensor;
- $\mathbf{K} = \{K^{\alpha\beta}\}$ - Kirchhoff stress tensor;
- $\mathbf{B} = \{B^{\alpha\beta}\}$ - second Biot (Jaumann) stress tensor;

Deformation in continuum mechanics is the transformation of a body from a reference configuration to a current configuration. Consider a material body, which is a set of elements, called particles and in a chosen reference configuration each particle is identified by the position vector $\mathbf{X}$. The components of :math'{bf X}' in the chosen reference frame are called *material coordinates*. Mapping χ from one configuration to another is called *deformation of the body*. Let

x to be a position of the particle in the deformed configuration, such that $\mathbf{x} = \chi(\mathbf{X})$. At any different time, the body may occupy different configurations so that a material paricle occupies a series of point in space. A smooth sequence of configurations in time describes a motion of the body *[Dill2006]*.
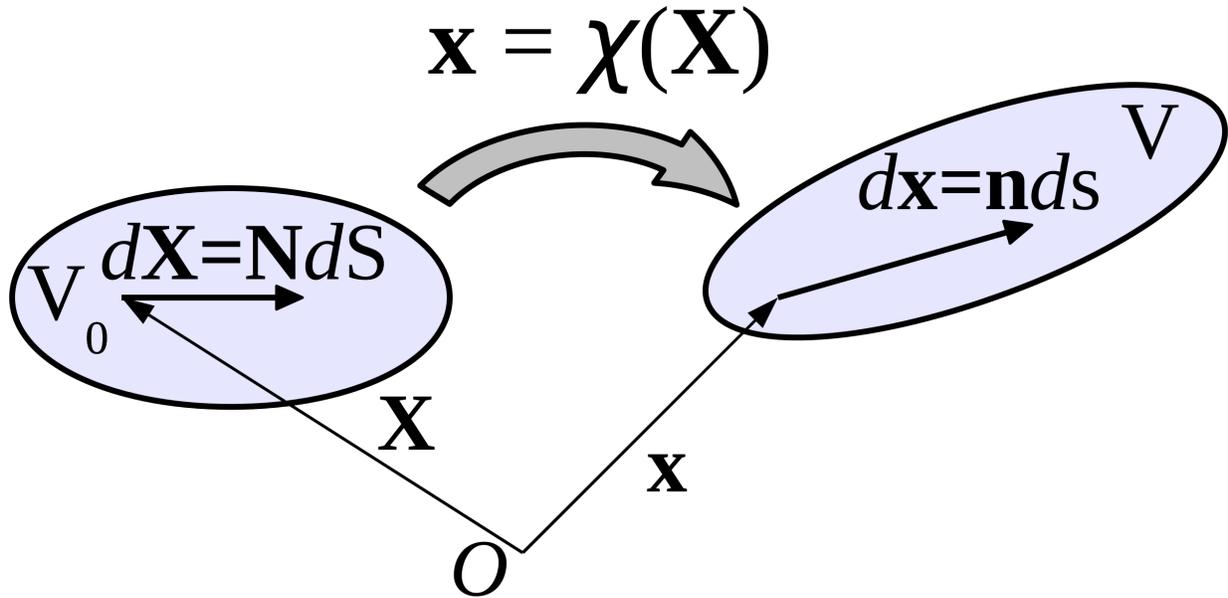

Fig. 1.1: **Figure 1:** Schematic of the body transformation from reference configuration (right) with position vector **X** to the deformed configuration (left) with position vector **x** occuring under smooth mapping χ.

Strain is a description of deformation in terms of relative displacement of particles in the body. Depending on the amount of strain, or local deformation, the analysis of deformation is subdivided into three deformation theories:

- Finite strain theory (large strain theory) deals with deformations in which both rotations and strains are arbitrarily large. In this case, the undeformed and deformed configurations of the continuum are significantly different and a clear distinction has to be made between them. This is commonly the case with elastomers, plastically-deforming materials and other fluids and biological soft tissue.

- Infinitesimal strain theory (small strain theory) - strains and rotations are both small. In this case, the undeformed and deformed configurations of the body can be assumed identical. The infinitesimal strain theory is used in the analysis of deformations of materials exhibiting elastic behavior, such as materials found in mechanical and civil engineering applications, e.g. concrete and steel.

- Large-displacement or large-rotation theory, which assumes small strains but large rotations and displacements.

In a body that is not deformed, the arrangement of the molecules corresponds to a state of thermal equilibrium, meaning that, if some portion of the body is considered, the resultant of the forces on that portion is zero. When a deformation occurs, the arrangement of the molecules is changed, and the body ceases to be in its original state of equiliubrium. Forces therefore arise which tend to return the body to equilibrium. These internal forces which occur when a body is deformed are called *internal stresses*. If no deformation occurs, there are no internal stresses *[Landau1970]*.

Quantitatively, stress at a point A in a body is defined as

$$\sigma = \lim_{\Delta A \to 0} \frac{\Delta \mathbf{T}}{\Delta A} \tag{1.1}$$

where ΔA is the area, surrounding the point A and $\Delta \mathbf{T}$ is acting force. The stress σ is a measure of the average force per unit area of a surface within the body on which internal forces act.

The stresses considered in continuum mechanics are only those produced during the application of external forces and the consequent deformation of the body; that is to say, *relative changes in deformation are considered rather*

than absolute values. A body is considered stress-free if the only forces present are those inter-atomic forces (ionic, metallic, and van der Waals forces) required to hold the body together and to keep its shape in the absence of all external influences, including gravitational attraction. This assertion is not valid in cases where the deformation of the body results in macroscopic electric fields in it, so we will not discuss such bodies in futher.

Strain

A strain is a normalized measure of deformation representing the *displacement between particles in the body relative to a reference length*. It is in general a tensor quantity. Physical insight into strains can be gained by observing that a given strain can be decomposed into *normal* and *shear* components. The amount of stretch or compression along a material line elements or *fibers* is the normal strain, and the amount of distortion associated with the sliding of plane layers over each other is the shear strain, within a deforming body. This could be applied by elongation, shortening, or volume changes, or angular distortion. Strains are dimensionless and are usually expressed as a decimal fraction, a percentage or in parts-per notation. Strains measure how much a given deformation differs locally from a rigid-body deformation.

Consider a particular point $\mathbf{X}$ and a neighboring point of the reference configuration are connected by the vector $d\mathbf{X}$. The direction of the fiber is given by unit vector $\mathbf{N}$ and its length is dS, such that $d\mathbf{X} = \mathbf{N}dS$. After deformation, the particle moves to the place $\mathbf{x}$ and the material fiber **rotates** and **elongates** to the deformed position $d\mathbf{x} = \mathbf{n}ds$. The deformed fiber is determined by the deformation:

$$d\mathbf{X} = \mathbf{F} \cdot d\mathbf{X} \tag{1.2}$$

where

$$\mathbf{F} = \frac{\partial \mathbf{x}(\mathbf{X})}{\partial \mathbf{X}} \tag{1.3}$$

Tensor $\mathbf{F}$ is called the *deformation gradient*. The deformation gradient at a point completely determines the rotation and the change in the length of any fiber of material emanating from this point.

Let us consider a pair of fibers $d_1\mathbf{X}$ and $d_2\mathbf{X}$, which are deformed to $d_1\mathbf{x}$ and $d_2\mathbf{x}$, then we have:

$$d_1\mathbf{x} \cdot d_2\mathbf{x} = d_1\mathbf{X} \cdot \mathbf{C} \cdot d_2\mathbf{X} \tag{1.4}$$

According to Eq. (1.3)

$$\mathbf{C} = \mathbf{F}^\tau \cdot \mathbf{F} \tag{1.5}$$

The tensor $\mathbf{C}$ is a function of $\mathbf{X}$ that is called the *deformation tensor* or also *Cauchy-Green tensor* or *Green's deformation tensor*.

Suppose that $d_1\mathbf{X} = d_2\mathbf{X} = d\mathbf{X}$. The *stretch* α is the ration of the deformed length to the initial length:

$$ds = \alpha dS \tag{1.6}$$

The stretch is therefore a positive number, equal to 1 for no change in the length, greater that 1 for an increase in length, and less than 1 for shortening of the fiber. The change in length divided by the reference length is the *extension* or *normal strain*, equal to $\alpha - 1$, then

$$\alpha^2 = \mathbf{N} \cdot \mathbf{C} \cdot \mathbf{N} \tag{1.7}$$

From Eqs. (1.6) and (1.2):

$$\alpha\mathbf{n} = \mathbf{F} \cdot \mathbf{N} \tag{1.8}$$

Next, consider two fibers that are initially perpendicular to each other. Let γ denotes the change in angle between the fibers upon deformation, positive for a decrease in angle. The angle between deformed fibers is $\pi/2 - \gamma$, so we call γ the *shear* or *shear strain* on the fibers. From Eq. (1.4):

$$\alpha_1 \alpha_2 sin\gamma = \mathbf{N_1} \cdot \mathbf{C} \cdot \mathbf{N_2} \tag{1.9}$$

The deformation tensor $\mathbf{C}$ at $\mathbf{X}$ determines the stretch of any fiber at $\mathbf{X}$ with given direction $\mathbf{N}$ (Eq. (1.7)), at the same time $\mathbf{C}$ also determines the shear of any pair of orthogonal fibers at $\mathbf{X}$ (Eq. (1.9)). Therefore, strain of the body at $\mathbf{X}$ is completely determined by $\mathbf{C}$. When $\mathbf{C} = \mathbf{1}$, no strain of fibers at $\mathbf{X}$ occurs. For material characterization, it is sometimes more convenient to have a measure of strain called *strain tensor*, which is zero when no strain occurs. One such strain is $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{1})$.

The choice of strain measure is arbitrary. In practice, it is decided by the simplicity of the mathematical formulation of the convenience in correlation with experiments on a particular material. *Any symmetric tensor that is one-to-one correspondence with the stretch :math:'{bf U}' (see definition below) and is zero for no deformation can be used as a strain measure*:

$$\mathbf{E} = f(\mathbf{U}), f(\mathbf{0}) =), \mathbf{U} = f^{-1}(\mathbf{E}) \tag{1.10}$$

For example, a general formula for *Lagrangian strain tensors* is:

$$\mathbf{E}_m = \frac{1}{2m}(\mathbf{U}^{2m} - \mathbf{1}) \tag{1.11}$$

for different values of m we can get:

$$
\begin{aligned}
\mathbf{E}_{(1)} &= \frac{1}{2}(\mathbf{U}^2 - \mathbf{1}) = \frac{1}{2}(\mathbf{C} - \mathbf{1}) \text{ - Green's strain tensor} \\
\mathbf{E}_{(1/2)} &= (\mathbf{U} - \mathbf{1}) \text{ - Biot (engeneering) strain tensor} \\
\mathbf{E}_{(0)} &= ln(\mathbf{U}) \text{ - Logarithmic (natural, true, Hencky) strain tensor}
\end{aligned}
\tag{1.12}
$$

Now, consider three fibers at $\mathbf{X}$ that are not collinear and make a right-handed system. The volume of the parallelepiped with these three adjacent edges in undeformed configuration is $dV_0 = [d_1\mathbf{X}, d_2\mathbf{X}, d_3\mathbf{X}]$. The bracket notation here denores the scalar triple product. In deformed configuration the there fibers determine the volume of deformed volume element $dV = [d_1\mathbf{x}, d_2\mathbf{x}, d_3\mathbf{x}]$. Using Eq. (1.2) for each fiber, we have:

$$dV = det\mathbf{F} \ dV_0 \tag{1.13}$$

If consider only deformations for which the volume is never decreased to zero ($J \equiv det\mathbf{F} > 0$) then the deformation gradient $\mathbf{F}$ is non-singular tensor and the inverse tensor $\mathbf{F}^{-1}$ exists. By the polar decomposition theorem, there exists a *rotational tensor* $\mathbf{R}$ and positive defined symmetric *right and left stretch tensors* $\mathbf{U}$ and $\mathbf{V}$, respectively, such that

$$
\begin{aligned}
\mathbf{F} &= \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R}, \\
\mathbf{U} &= \mathbf{U^T}, det\mathbf{U} > 0, \\
\mathbf{V} &= \mathbf{V^T}, det\mathbf{V} > 0, \\
\mathbf{R}^{-1} &= \mathbf{R^T}, det\mathbf{R} = +1
\end{aligned}
\tag{1.14}
$$

The tensor $\mathbf{U}$ is called *right stretch tensor* and $\mathbf{V}$ is called *left stretch tensor*. The representation in Eq. (1.14) is a decomposition of the deformation gradient into a stretch $\mathbf{U}$ followed by rotation $\mathbf{R}$, or a rotation $\mathbf{R}$ is followed by stretch $\mathbf{V}$. It follows from Eq. (1.5) that

$$\mathbf{C} = \mathbf{U^2}, \mathbf{U} = \mathbf{C^{1/2}} \tag{1.15}$$

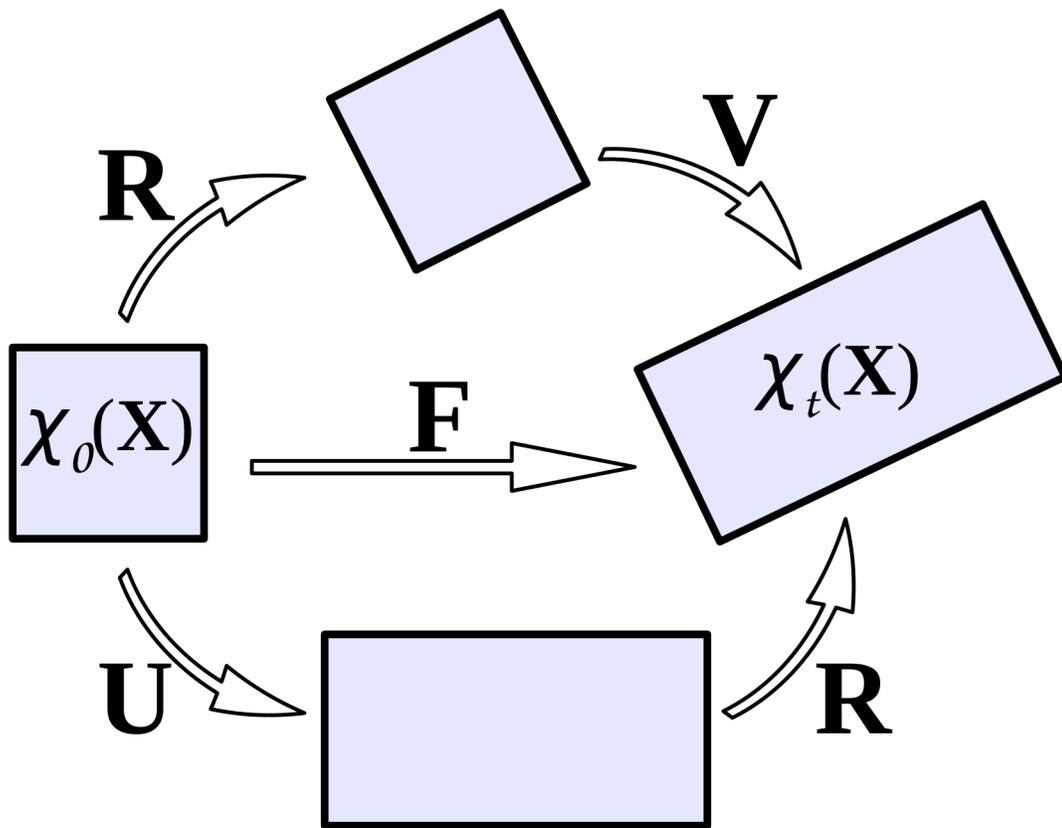For the further reading see *[Dill2006]*.

Fig. 1.2: **Figure 2:** Schematic representation of the polar decomposition of the deformation gradient. The reference (undeformed) configuration is $\chi_0(\mathbf{X})$, the deformed configuration in a moment t is $\chi_t(\mathbf{X})$.

Stress

Let us consider the total force acting on some portion of the body. Firstly, this total force is equal to the sum of all the forced on all the volume elements in that portion of the body, i.e. it can be written as the volume integral $\int \mathbf{T}dV$, where $\mathbf{T}$ is the force per unit volume and $\mathbf{T}dV$ is the force on the volume element dV. Secondly, the forces with which various parts of the portion considered act on one another are canceled by Newton's third law and, therefore, give zero in the total resultant force. Thus, the required total force can be regarded as the sum of the forces exerted on the given portion of the body by the portion surrounding it.

Hence, for any portion of the body, each of three components $\int \mathbf{T}^{\alpha}dV$ of the resultant of all the internal stresses can be transformed into an integral over the surface. According to divergence theorem, the integral of a scalar over an arbitrary volume can be transformes into an integral over the surface if the scalar is the divergence of a vector. In the present case we have the integral of a vector, then the vector T^{α} must be the divergence of a tensor of rank two, i.e. be of the form $T^{\alpha} = \partial\sigma^{\alpha\beta}/\partial x^{\beta}$. Then the force on any volume can be written as an integral over the closed surface bounding that volume:

$$\int T^{\alpha}dV = \int \frac{\partial\sigma^{\alpha\beta}}{\partial x^{\beta}}dV = \oint \sigma^{\alpha\beta}dn^{\beta} \tag{1.16}$$

where dn^{β} are the component of the surface element vector $d\mathbf{n}$, directed along the outward normal.

The tensor $\sigma^{\alpha\beta}$ is called the *stress tensor*. As we see from Eq. (1.16), $\sigma^{\alpha\beta}dn^{\beta}$ is the α-th component of the force $\mathbf{T}$ acting on the surface element $d\mathbf{n}$. By taking elements of area in the planes of xy, yz, xz, we find that the component $\sigma^{\alpha\beta}$ of the stress tensor is the α-th component of the force acting on a unit area perpendicular to the x^{β}-axis *[Landau1970]*.

Depending on the orientation of the plane under consideration, the stress vector may not necessarily be perpendicular to that plane, i.e. parallel to $\mathbf{n}$, and can be resolved into two components:

- one component normal to the plane, called *normal stress*:

$$\sigma_{\mathbf{n}} = \lim_{\Delta A \to 0} \frac{\Delta T_{\mathbf{n}}}{\Delta A} = \sigma \cdot \mathbf{n} = \sum_{\alpha,\beta} \sigma^{\alpha\beta}n^{\alpha}n^{\beta} \tag{1.17}$$

where ΔT_n is the normal component of the force $\Delta\mathbf{T}$ to the differential area ΔA (see Eqs. (1.1) and (1.16)).

- and the other component parallel to this plane, called the *shear stress*:

$$\tau_{\mathbf{n}} = \lim_{\Delta A \to 0} \frac{\Delta T_{\mathbf{s}}}{\Delta A} = \sqrt{\sigma^2 - \sigma_{\mathbf{n}}^2} = \sqrt{\sum_{\alpha,\beta,\gamma} \sigma^{\alpha\beta}\sigma^{\alpha\gamma}n^{\beta}n^{\gamma}} \tag{1.18}$$

where ΔT_s is the tangential component of the force $\Delta\mathbf{T}$ to the differential surface area ΔA (see Eqs. (1.1) and (1.16)). The shear stress can be further decomposed into two mutually perpendicular vectors.

Consider a body in reference state and deformed state. Let $dA\mathbf{N}$ denotes a unit area with normal vector $\mathbf{N}$ in reference state and $da\mathbf{n}$ denotes the same unit area in deformed state with the normal vector $\mathbf{n}$. Let $d\mathbf{T}$ be the differential force vector that represents the resultant of the total internal forces acting on da in deformed state. According to Eqs. (1.1) and (1.16), we can define force $\Delta\mathbf{T}$ and unit area ΔA in arbitrary way and evaluate several measures of stress:

- *Cauchy stress* (σ) is the most commonly used measure of stress (also is called true stress tensor), which is defined as force in deformed configuration per unit area in deformed configuration:

$$d\mathbf{T} = (\mathbf{n}da) \cdot \sigma \tag{1.19}$$

The Cauchy stress tensor is used for stress analysis of material bodies experiencing small deformations where the differences in stress distribution in most cases can be neglected. The Cauchy stress tensor is symmetric.

- *First Piola-Kirchhoff stress* ($\mathbf{P}$) is the measure of stress defined as force in deformed configuration per unit area in reference configuration:
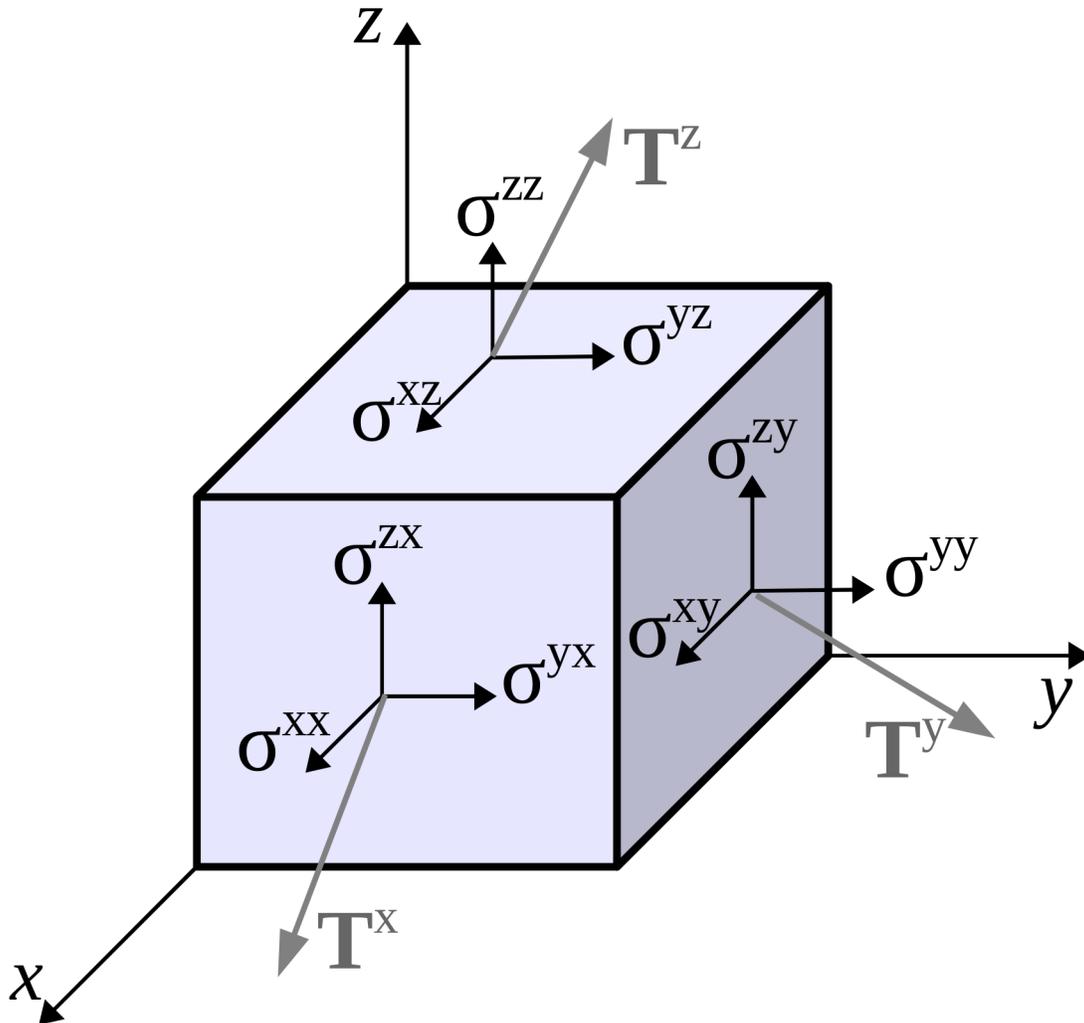
Fig. 1.3: **Figure 3:** The components of the stress vector on a coordinate surface are equal to three components of the stress tensor, such that the first index corresponds to the direction on the component and the second index identifies the direction of surface.

$$\mathbf{dT} = (dA\mathbf{N}) \cdot \mathbf{P},$$
$$\mathbf{P} = J\mathbf{F}^{-1} \cdot \sigma \tag{1.20}$$

The first Piola-Kirchhoff stress tensor is used for stress analysis of material bodies experiencing large deformations. Like deformation gradient $\mathbf{F}$, the first Piola-Kirchhoff tensor is unsymmetric two-point tensor.

- *Second Piola-Kirchhoff stress* ($\mathbf{S}$)} is the measure of stress defined as modified force in deformed configuration ($\mathbf{d\hat{T}}$) per unit area in reference configuration. The purpose of this is that the second Piola-Kirchhoff stress tensor will now be a symmetric tensor:

$$\mathbf{d\hat{T}} = \mathbf{F}^{-1} \cdot \mathbf{dT},$$
$$\mathbf{d\hat{T}} = (dA\mathbf{N}) \cdot \mathbf{S}, \tag{1.21}$$
$$\mathbf{S} = J\mathbf{F}^{-1} \cdot \sigma \cdot \mathbf{F}^{-T}$$

If the material rotates without a change in stress state (rigid rotation), the components of the first Piola-Kirchhoff stress tensor will vary with material orientation, but the components of the second Piola-Kirchhoff stress tensor remain constant, irrespective of material orientation. For infinitesimal deformations or rotations, the Cauchy and Piola-Kirchhoff tensors are identical.

- *Kirchhoff stress tensor* ($\mathbf{K}$) is a scalar multiple of the Cauchy stress tensor σ. The scalar factor is the determinant of deformation gradient tensor, $\mathbf{F}$:

$$\mathbf{K} = J\sigma \tag{1.22}$$

It is used widely in numerical algorithms in metal plasticity because relates to the Cauchy stress via scalar.

- *Biot (Jaumann) stress tensor* ($\mathbf{B}$) is the measure of stress defined as force due to stretch only applied in the undeformed body ($\mathbf{d\hat{T}}$) per unit undeformed area.

$$\mathbf{d\hat{T}} = \mathbf{R}^{-1} \cdot \mathbf{dT},$$
$$\mathbf{d\hat{T}} = (dA\mathbf{N}) \cdot \mathbf{B}, \tag{1.23}$$
$$\mathbf{B} = J\mathbf{F}^{-1} \cdot \sigma \cdot \mathbf{R} = J\mathbf{F}^{-1} \cdot \sigma \cdot \mathbf{F}^{-T} \cdot \mathbf{U}$$

where $\mathbf{R}$ and $\mathbf{U}$ is a rotational and rigth stretch components respectively of deformation gradient tensor due to polar decomposition (see Eq. (1.14)). Biot stress tensor is used in combination with Biot (engeneering) strain tensor *[Dill2006]*, *[Bonet2008]*.

Numerical approaches

The calculations of stress and strain tensors of a biological particle are based on the coarse-grained representation of the molecule, described by *Self-Organized Polymer (SOP) model*. That is, the molecule is described by one or two interacting centers (C_α's only or C_α and C_β) and total potential energy is follow

$$U_{SOP} = U_{FENE} + U_{NB}^{ATT} + U_{NB}^{REP} \tag{2.1}$$

The details of the SOP model, as well as complete explanation of how to use SOP-GPU package can be found in SOP-GPU documentation.

Strain

Because strain is fundamentally a continuum quality, its computation from atomic displacement data requires either interpolation of a continuous displacement field ($\chi(\mathbf{X})$) or discretization of the deformation gradient tensor ($\mathbf{F}$). Here we go with the second way.

To calculate an atomic deformation gradient tensor ($\mathbf{F_i}$) we utilized the approach purposed by Horstemeyer and Baskes *[Horstemeyer1999]* and improved by Gullete et al. *[Gullett2008]*. The deformation gradient tensor emerges from a weighted least squares optimization of local deformation data. In an arbitrary atomic system, the deformation in the neighborhood of atom i is characterized by the changes in the relative position of its neighbors. Let atom i is located at the position $\mathbf{X}_i$ in reference configuration and position $\mathbf{x}_i$ in deformed configuration. The relative position of neighboring atom j is given by the vector $\Delta\mathbf{X}_{ij} = \mathbf{X}_j - \mathbf{X}_i$ in reference configuration and by $\Delta\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ in deformed configuration. Here the subscripts denote atom numbers, and the vectors $\Delta\mathbf{X}_{ij}$ and $\Delta\mathbf{x}_{ij}$ connect atom i to its neighbors j. There exists a unique linear mapping that transforms the relative position vector $\Delta\mathbf{X}_{ij}$ to $\Delta\mathbf{x}_{ij}$:

$$\Delta\mathbf{x}_{ij} = \mathbf{F}_i \cdot \Delta\mathbf{X}_{ij} \tag{2.2}$$

where $\mathbf{F}_i$ is the value of the deformation gradient at atom i. Here, Eq. (2.2) defines the operation of the discrete deformation gradient at atom i in a manner analogous to the continuum deformation gradient of Eq. (1.2), where it maps a vector in the reference configuration to a vector in the deformed configuration. Writing Eq. (1.2) for atom i and each of its neighbors leads to a system of linear equations that, because of the freedom of the atomic motion, cannot generally be satisfied by a single mapping $\mathbf{F}_i$. For this reason, an optimal local deformation gradient ($\hat{\mathbf{F}}_i$) is

sought, by calculating the mapping error between atom i and a single neighbor j as l^2-norm of the difference between the $\Delta\mathbf{x}_{ij}$ and $\hat{\mathbf{F}}_i \cdot \Delta\mathbf{X}_{ij}$:

$$\phi_{ij} = (\Delta\mathbf{x}_{ij} - \hat{\mathbf{F}}_i \cdot \Delta\mathbf{X}_{ij})^\tau (\Delta\mathbf{x}_{ij} - \hat{\mathbf{F}}_i \cdot \Delta\mathbf{X}_{ij}) \tag{2.3}$$

Thus, the weighted least squares error among the neighbors of i is given by

$$\phi_i = \sum_{j=1}^{N} (\Delta\mathbf{x}_{ij} - \hat{\mathbf{F}}_i \cdot \Delta\mathbf{X}_{ij})^\tau (\Delta\mathbf{x}_{ij} - \hat{\mathbf{F}}_i \cdot \Delta \tag{2.4}$$

where N is the number of neighboring atoms, and w_j is a weight factor.

This formulation takes into account the fact that local deformation at individual atoms in system are subjected to inhomogeneous deformations; thus the local deformation gradient is computed without extension of the mapping error to the bulk. The optimal deformation gradient $\hat{\mathbf{F}}_i$, in least square sense, is obtained by minimizing ϕ_i with respect to components of $\hat{\mathbf{F}}_i$.

$$\frac{\partial\phi_i}{\partial\hat{F}^{\alpha\beta}} = \sum_{j=1}^{N} (-2\Delta x_{ij}^\alpha \Delta X_{ij}^\beta + 2\Delta X_{ij}^\beta \hat{F}^{\alpha\gamma} \Delta X_{ij}^\gamma) w_j \tag{2.5}$$

Here greek superscripts denote components of vectors $\Delta\mathbf{X}_{ij}$ and $\Delta\mathbf{x}_{ij}$ and tensor $\hat{\mathbf{F}}_i$. Setting Eq. (2.5) equal to zero and solving for the components of $\hat{\mathbf{F}}_i$ get the linear system of equations which in matrix notation can be written as

$$\hat{\mathbf{F}}\mathbf{D} = \mathbf{A} \tag{2.6}$$

where $\mathbf{D}$ and $\mathbf{A}$ are 3×3 matricies such as

$$\mathbf{D} = \sum_j \Delta\mathbf{X}_{ij}\Delta\mathbf{X}_{ij}^\tau w_j,$$
$$\mathbf{A} = \sum_j \Delta\mathbf{x}_{ij}\Delta\mathbf{X}_{ij}^\tau w_j \tag{2.7}$$

Provided that $\mathbf{D}$ was formed from a minimum of three non-coplanar nearest neighbors having nonzero weights, the optimal discrete deformation gradient is determined as

$$\hat{\mathbf{F}} = \mathbf{A}\mathbf{D}^{-1} \tag{2.8}$$

which is *the deformation gradient of the system in current configuration with respect to reference configuration [Gullett2008]*.

To obtain the value of atomic stretch tensor ($\mathbf{U_i}$) we computed the Green's strain tensor according to Eq. (1.5) and calculated square root (see Eq. (1.15)) using eigenvalue decomposition of tensor $\mathbf{C} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}$, where {bf R} is the matrix of eigenvectors and $\mathbf{\Lambda}$ is eigenvalue matrix. Since tensor $\mathbf{C}$ is real symmetric matix then eigenvectors and eigenvalues are defined in a real space, $\mathbf{\Lambda}$ is positive-defined and $\mathbf{R}^{-1} = \mathbf{R}^\tau$ is orthogonal matrix. Then

$$\mathbf{U} = \mathbf{C}^{1/2} = \mathbf{R}\mathbf{\Lambda}^{1/2}\mathbf{R}^\tau \tag{2.9}$$

The eigenvalues of matrix $\mathbf{C}$ (also called *principal* values) were obtained analyticaly, writing the characteristic polynomial for an arbitrary symmetric 3×3 matrix:

$$M = \begin{pmatrix} a & d & e \\ d & b & f \\ e & f & c \end{pmatrix}$$

$$\lambda^3 + A\lambda^2 + B\lambda + C = 0 \tag{2.10}$$

where coefficients are

$$A = -(a + b + c),$$
$$B = ab + bc + ac - d^2 - e^2 - f^2,$$
$$C = abc + 2dfe - e^2b - f^2a - d^2c = det(M)$$

are invariats of matrix (tensor), meaning that they do not change with rotation of the coordinate system. The solution of cubic equation was obtained using Viete's substitution (also Viete's triginometric formulas):

$$\lambda_1 = -2\sqrt{Q}\cos{(\phi)} - \frac{A}{3},$$
$$\lambda_2 = -2\sqrt{Q}\cos{(\phi + \frac{2}{3}\pi)} - \frac{A}{3}, \tag{2.11}$$
$$\lambda_3 = -2\sqrt{Q}\cos{(\phi - \frac{2}{3}\pi)} - \frac{A}{3}$$

where

$$\phi = \frac{1}{3}\arccos{\frac{R}{\sqrt{Q^3}}}, \ Q = \frac{A^2 - 3B}{9}, \ R = \frac{2A^3 - 9AB + 27C}{54}$$

Since we work in the space of symmetric real matricies all the possiblel solution of the characteristic polynomial (Eq. (2.10)) are real positive values (Eqs. (2.11)).

Solving problems Eqs. (2.10) - (2.11) for each Green's tensor $\mathbf{C}_i$ for each atom i, we sustitute solution into Eq. (2.9)) and can obtain atomic stretch tensor $\mathbf{U}_i$. We can calculate different atomic strain tensors $\mathbf{E}_i$ using Eqs. (1.11).

Stress

We calculated atomic stress using forlmulas, widely utilized to measure atomic stress in crystals (*[Basinski1971]*, *[Cormier2001]*, *[Delph2005]*) and proteins (*[Ishikura2012]*), according to which

$$\sigma_i = \frac{1}{2\Omega_i}(\sum_{j \neq i}\mathbf{T}_{ij} \otimes \mathbf{r}_{ij} + m_i\mathbf{v}_i \otimes \mathbf{v}_i) \tag{2.12}$$

where σ_i is the atomic stress tensor of atom i, $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, $\otimes$ represents the tensor product, $\mathbf{T}_{ij}$ is the pairwise force vector acting on atom i from atom j, and m_i, $\mathbf{v}_i$ and Ω_i are the mass, velocity and volume of atom i, respectively. In our calculations we neglect kinetic energy of an atom (second term in Eq. (2.12)) since we consider coarse-grained representation of the molecule when all the particles have same kinetic energy term.

If consider $T_{ij}^\alpha = -\frac{dW_{ij}(r_{ij})}{dr_{ij}}\frac{dr_{ij}^\alpha}{r_{ij}}$, where $W_{ij}(r_{ij})$ is pair potential between atom i and j, then the stress tensor representation in component becomes:

$$\sigma_i^{\alpha\beta} = -\frac{1}{2\Omega_i}\sum_{j \neq i}\frac{dW_{ij}(r_{ij})}{dr_{ij}}\frac{r_{ij}^\alpha r_{ij}^\beta}{r_{ij}} \tag{2.13}$$

The question arises how calculate the atomic volume Ω_i. This is the point which separates all the numerical formulations of stress measure provided previously (*[Hardy1982]*, *[Cormier2001]*) and it can be calculated in different ways. In our calculation, for each atom we considered it interacting in pairwise way with other particles within the certain cut-off radius. Then we found so-called "effective volume" originated from these pairs:

$$\Omega_i = \frac{4\pi}{3}a_i^3, \ \text{where} \ a_i = \frac{\sum_{j \neq i} r_{ij}^{-1}}{2\sum_{j \neq i} r_{ij}^{-2}} \tag{2.14}$$

here r_{ij} is the distance between two particles within a chosen cut-off. For our coarse-grained tensors the values for $a_i \approx 3.1 - 3.9$ Å, corresponding to a covalent $C_\alpha - C_\alpha$ bond length.

The Eq. (2.13) in the present form corresponds to the local atomic Cauchy stress. In order to obtain any other stress measures one can use the Eqs. (1.19) - (1.23). Another way, showed by Delph (*[Delph2005]*), to obtain second Piola-Kirchhoff stress tensor using formula:

$$P_i^{\alpha\beta} = -\frac{1}{2\Omega_i} \sum_{j \neq i} \frac{dW_{ij}(r_{ij})}{dr_{ij}} \frac{R_{ij}^\alpha R_{ij}^\beta}{R_{ij}} \tag{2.15}$$

where $\mathbf{R_{ij}} = \mathbf{X}_j - \mathbf{X}_i$ - the distance between particles in reference confuguration.

The Eq. (2.13) were modified for a **spherical geometry** by Zandi and Reguera (*[Zandi2005]*) to describe the stress measures of icosahedral shells. They derived formulas in terms of spherical basis $(\mathbf{e}_r, \mathbf{e}_\phi, \mathbf{e}_\theta)$, which correspond to *radial*, *azimutal* and *polar unit vectors*, respectively. According to this geometry, stress measures can be defined as follow:

$$(\sigma_T)_i = -\frac{1}{2\Omega^{2D}} \sum_{j \neq i} \frac{dW_{ij}(r_{ij})}{dr_{ij}} \frac{(r_{ij} \cdot \mathbf{e}_\theta)^2 + (r_{ij} \cdot \mathbf{e}_\phi)^2}{2r_{ij}} \text{ - lateral (tangential) component of stress,}$$

$$(\sigma_{\theta\phi})_i = -\frac{1}{2\Omega^{2D}} \sum_{j \neq i} \frac{dW_{ij}(r_{ij})}{dr_{ij}}(r_{ij} \cdot \mathbf{e}_\theta)(r_{ij} \cdot \mathbf{e}_\phi) \text{ - (45}^o\text{)-shear component of stress,} \tag{2.16}$$

$$(\sigma_R)_i = -\frac{1}{2\Omega^{3D}} \sum_{j \neq i} \frac{dW_{ij}(r_{ij})}{dr_{ij}}(r_{ij} \cdot \mathbf{e}_r)^2 \text{ - radial component of stress,}$$

First two stress components are defined in-plane as the projection on the spherical surface, therefore the volume element, Ω^{2D}, corresponding to particle is also in-plane, i.e. area. The third stress component, radial, is still three-dimentional (3D), therefore Ω^{3D} is atomic volume, as described above.

Using TensorCalculator

To build executable file change directory to the TensorCalculator/src/ type in command line:

```
make
```

The approaches described in a Section *Numerical approaches* are implemented in TensorCalculator (TC) program package, which allows to calculate and visualize the distribution of a stress/strain measure on a surface of considered biological particle. There are two regimes available in TC: **tensor computing regime** and **tensor visualization regime**. Running TC requires presence of the following files:

- Structure file: file of the coarse-grained structure in *.pdb* format.

- Topology file: file of the coarse-grained structure topology in Gromacs *.top* format (see SOP-GPU documentation for details).

- Trajectory file: file of coordinates of a system in coarse-grained representation, in *.dcd* format. Note: the trajectory as well as initial *.pdb* structure should contain only the atoms of a considered biomolecule or its part (no water, ions, mica surface or tip atoms).

- If running TC just in visualization regime, tensor files *.tnsr* are also required.

All these as well as other parameters should be specify in configuration file (see Section *Input parameters file*). To run TC, type in terminal:

```
tensors config_file.config
```

The program will generate output files with the atomic tensors values arranged in *.tnsr* format or/and *.pdb* format for visualization.

Format *.tnsr*

The calculated values of atomic tensors can be written out in output files, in order to use them later for visualization. Tensor values file *.tnsr* has one-to-one correspondance with *.dcd* trajectory file and each frame is separated by the word END:

```
-------(a)--(b)-(c)-(d)-(e)------(f)---
ATOM      1 ALA   1 A  A1   -2.00000 ... 2.50000
ATOM      2 ALA   2 A  A1   -3.00000 ... 3.50000
ATOM      3 ALA   3 A  A1   -4.00000 ... 4.50000
...
```

(a): atom ID; (b): residue name; (c): redidue number; (d): chain identificator; (e): segment identificator; (f): tensor components;

Note: depending on calculating tensor type and the output value, there may be from 1 to 9 components, written at position (f). One component if output is scalar (energy distribution), three components is output is vector (spherical tensors disribution) and nine components if output is the second-ranked tensor, i.e. matrix (stress or strain distribution).

Tensors calculation

Using TC one can compute the following measures:

- Potential energy distribution: total potential energy, calculated according to Eq. (2.1) for each atom in the system.

- Stress tensor distribution: Cauchy stress tensor (Eq. (1.19)) calculated per atom using Eq. (2.13).

- Spherical tensors distribution: stress tensor projected along the combination of spherical unit vectors, calculated per atom for an object possesing spherical geometry (Eq. (2.16)).

- Strain tensor distribution: calculation of the Green strain tensor (Eq. (1.12)) per atom, the deformation gradient tensor (Eq. (1.3)) is also calculted as intermediate using Eq. (2.8). If required, atomic stretch tensor (Eq. (1.14)) can be calculated within the strain tensor option using Eq. (2.9).

Tensors visualization

If tensor files already exist, they can be visualized separately from calculation. Most of the time it is required if appropriate representation of the values is sought. Visualization regime requires presence of tensor files (see Section *Format .tnsr*) as well as structure and trajectory file in order to produce output file in PDB format with multiple frames. This PDB file can be uploaded in VMD (*[Humphrey1996]*) using TCL script (see Section *Example of VMD TCL script* below).

Since a tensor value has nine components, in order to obtain scalar value for visualization, TC calculates projection of the tensor on a preferred direction (normal vector). Hence, separating tensor into two components: normal and shear (Eqs. (1.17) and (1.18), see Section *Stress* for details). The normal vector should be specified in configuration file (see Section *Input parameters file* below). The normal and shear components of the tensor will be written out in *occupancy* and *beta* columns of the output PDB file, respectively. In case of potential energy distribution, the scalar atomic value will be written out in the *occupancy* column only. In case of spherical tensor, all three values of the lateral, 45°-shear and radial stresses will be written out in *occupancy*, *beta* and *charge* columns of the PDB file, respectively.

Input parameters file

The TC program can work both with coarse-grained and all-atomic representations of the system and trajectory, transforming all-atom structure and trajectory into coarse-grained and saving coarse-grained trajectory file separately. However, it still requires as an input to have topology file in *.top* format, which can be obtained running `sop-top` utility of the SOP-GPU package.

General parameters

- **name** *<protein name>*

    Type: String.

    Status: Optional.

    Purpose: Name, assigned to the protein. Used mostly for files naming.

- **structure** *<filename>*

    Type: String.

    Status: Required.

    Purpose: Path to the structure coordinates file in PDB format either in all-atom or coarse-grained representation. This coordinates are also taken as *reference* configuration for calculation of stress and strain tensors.

- **topology** *<filename>*

    Type: String.

    Status: Required.

    Purpose: Path to the structure topology file in Gromacs-like format *.top*.

- **dcd** *<filename>*

Type: String.

Status: Required.

Purpose: Path to the trajectory coordinates file in *.dcd* format.

- **allAtom** *<yes/no>*

  Type: Boolean.

  Status: Optional.

  Default value: No.

  Purpose: Specifies if initial srtucture and trajectory files are in all-atom representation, rather than coarse-grained.

- **dcdCa** *<filename>*

  Type: String.

  Status: Required if **allAtom** is *on*.

  Purpose: Path to the trajectory coordinates file in *.dcd* format saved for coarse-grained representation.

- **initialFrame** *<frame number>*

  Type: Integer.

  Status: Optional.

  Default value: 0;

  Purpose: Number of an initial frame to start calculations and visualization. If one performes just visualization of tensors, the **initialFrame** number should be in consistence with tensor input files initial frame.

- **frameCount** *<number of frames>*

  Type: Integer.

  Status: Optional.

  Default value: 1;

  Purpose: Number of frames, started from the **initialFrame**, using for tensor calculation/visualization. If one performes just visualization of tensors, the **frameCount** number should be in consistence with the amount of frames in tensor input files.

- **compute** *<yes/no>*

  Type: Boolean.

  Status: Optional.

  Default value: Yes.

  Purpose: Run calculation of the tensor measures as despribed in Section *Using TensorCalculator* for a structure and trajectory specified in **structure** and **dcd** respectively.

Tensor visualization parameters

- **printPDB** *<yes/no>*

Type: Boolean.

Status: Optional.

Default value: Yes.

Purpose: Create PDB files for visualization of the tensor measures as despribed in Section *Using Tensor-Calculator* using existing tensor files *.tnsr*.

- **average** *<string>*

  Type: "no"/"cutoffAv"/"cutoffSum"/"segmentAv"/"segmentSum".

  Status: Optional.

  Default value: "no".

  Purpose: Specifies if averaging or summation for an atom is required for better representation: average over all atoms within cut-off distance (**cutoffAv**), summation over all atoms within cut-off distance (**cutoffSum**), averaging over all atoms in each segment (**segmentAv**) and summation over all atoms in each segment (**segmentSum**). Valid only if **printPDB** is *on*.

- **normalVector** *<string>*

  Type: "endtoend"/"vector".

  Status: Required if **printPDB** is *on* .

  Default value: "vector".

  Purpose: Specifies normal vector to calculate normal and shear components of the tensor (see Secton *Stress*). If "endtoend" the normal vector will be calculated as the end-to-end vector between two given particles.

- **normalDirection** *<x, y, z normalized coordinates>*

  Type: Vector.

  Status: Required if **normalVector** is *vector*.

  Purpose: Direction of the normal vector, using to obtain normal and shear components of the tensor.

- **end1/end2** *<residue ID>*

  Type: Integer.

  Status: Required if **normalVector** is *endtoend*.

  Purpose: Direction of the normal vector, using to obtain normal and shear components of the tensor.

Energy distribution parameters

- **energy** *<on/off>*

  Type: Boolean.

  Status: Optional.

  Default value: Off.

  Purpose: Switching on/off calculation/visualization of the potential energy distribution per atom (Eq. (2.1)).

- **outputEnergy** *<filename>*

Type: String.

Status: Required if **energy** is *on*.

Purpose: Path to the output tensor file in *.tnsr* format with the values of the atomic potential energy.

- **outputPDBEnergy** *<filename>*

  Type: String.

  Status: Required if **energy** is *on*.

  Purpose: Path to the PDB output file with energy values written out in *occupancy* column.

- **inputEnergy** *<filename>*

  Type: String.

  Status: Required if **energy** is *on* and **compute** is *off*.

  Purpose: Path to the input tensor file in *.tnsr* format.

- **energyScaleFactor** *<scale factor>*

  Type: Float.

  Status: Optional.

  Default value: 1.0.

  Purpose: When specified, the value written out in PDB file will be multiplied by this factor. Used for better representation big/small values, which do not fit PDB column.

Stress distribution parameters:

- **cauchyStress** *<on/off>*

  Type: Boolean.

  Status: Optional.

  Default value: Off.

  Purpose: Switching on/off calculation/visualization of the Cauchy stress distribution per atom (Eq. (2.13)).

- **outputCauchyStress** *<filename>*

  Type: String.

  Status: Required if **cauchyStress** is *on*.

  Purpose: Path to the output file in *.tnsr* format with the components of the atomic Cauchy stress tensor.

- **outputPDBCauchyStress** *<filename>*

  Type: String.

  Status: Required if **cauchyStress** is *on*.

  Purpose: Path to the PDB output file with Cauchy stress tensor normal and shear components written out in *occupancy* and *beta* columns, respectively.

- **inputCauchyStress** *<filename>*

  Type: String.

  Status: Required if **cauchyStress** is *on* and **compute** is *off*.

  Purpose: Path to the input tensor file in *.tnsr* format.

- **cauchyScaleFactor** <*scale factor*>

  Type: Float.

  Status: Optional.

  Default value: 1.0.

  Purpose: When specified, the value in PDB file will be multiplied by this factor. Used for better representation big/small values, which do not fit PDB column.

- **sphereStress** <*on/off*>

  Type: Boolean.

  Status: Optional.

  Default value: Off.

  Purpose: Switching on/off calculation/visualization of the Cauchy stress projected on spherical unit vectors (Eq. (2.16)).

- **outputSphereStress** <*filename*>

  Type: String.

  Status: Required if **sphereStress** is *on*.

  Purpose: Path to the output file in *.tnsr* format with the components of the atomic Cauchy stress tensor projected into spherical unit vectors.

- **outputPDBSphereStress** <*filename*>

  Type: String.

  Status: Required if **sphereStress** is *on*.

  Purpose: Path to the PDB output file with the lateral, (45°)-shear and radial spherical stress tensor components written out in *occupancy*, *beta* and *charge* columns, respectively.

- **inputSphereStress** <*filename*>

  Type: String.

  Status: Required if **sphereStress** is *on* and **compute** is *off*.

  Purpose: Path to the input tensor file in *.tnsr* format.

- **sphereScaleFactor** <*scale factor*>

  Type: Float.

  Status: Optional.

  Default value: 1.0.

  Purpose: When specified, the value in PDB file will be multiplied by this factor. Used for better representation big/small values, which do not fit PDB column.

Strain distribution parameters

- **strain** <*on/off*>

Type: Boolean.

Status: Optional.

Default value: Off.

Purpose: Switching on/off calculation/visualization of the Green's strain distribution per atom (Eq. (1.12)).

- **outputGreenStrain** *<filename>*

Type: String.

Status: Required if **strain** is *on*.

Purpose: Path to the output file in *.tnsr* format with the components of the atomic Green's strain tensor.

- **outputPDBGreenStrain** *<filename>*

Type: String.

Status: Required if **strain** is *on*.

Purpose: Path to the PDB output file with the normal and shear strain tensor components written out in *occupancy* and *beta* columns, respectively.

- **inputGreenStrain** *<filename>*

Type: String.

Status: Required if **strain** is *on* and **compute** is *off*.

Purpose: Path to the input tensor file in *.tnsr* format.

- **strainScaleFactor** *<scale factor>*

Type: Float.

Status: Optional.

Default value: 1.0;

Purpose: When specified, the value in PDB file will be multiplied by this factor. Used for better representation big/small values, which do not fit PDB column.

- **stretch** *<on/off>*

Type: Boolean.

Status: Optional.

Default value: Off.

Purpose: Switching on/off calculation/visualization of the stretch distribution per atom.

- **outputStretch** *<filename>*

Type: String.

Status: Required if **stretch** is *on*.

Purpose: Path to the output file in *.tnsr* format with the components of the atomic stretch tensor (Eq. (2.9)).

- **outputPDBStretch** *<filename>*

Type: String.

Status: Required if **stretch** is *on*.

Purpose: Path to the PDB output file with the normal and shear stretch tensor components written out in *occupancy* and *beta* columns, respectively.

- **inputStretch** *<filename>*

  Type: String.

  Status: Required if **stretch** is *on* and **compute** is *off*.

  Purpose: Path to the input tensor file in *.tnsr* format.

- **stretchScaleFactor** *<scale factor>*

  Type: Float.

  Status: Optional.

  Default value: 1.0.

  Purpose: When specified, the value in PDB file will be multiplied by this factor. Used for better representation big/small values, which do not fit PDB column.

- **outputDeformGradient** *<filename>*

  Type: String.

  Status: Optional.

  Purpose: Path to the output file in *.tnsr* format with the components of the atomic deformation gradient tensor (Eq. (2.8)).

- **outputPDBDeformGradient** *<filename>*

  Type: String.

  Status: Optional.

  Purpose: Path to the PDB output file with the normal and shear deformation gradient tensor components written out in *occupancy* and *beta* columns, respectively.

- **inputDeformGradient** *<filename>*

  Type: String.

  Status: Optional.

  Purpose: Path to the input tensor file in *.tnsr* format.

- **deformGradientWeightCovalent,   deformGradientWeightNative,   deformGradientWeightRepulsive** *<weight factors>*

  Type: Float.

  Status: Optional.

  Default value: 1.0, 1.0, 1.0.

  Purpose: Weight factor for covalent, native and repulsive interactions, used in Eq. (2.7).

Example of VMD TCL script

This TCL script can be utilize to visualize tensors using VMD software (*[Humphrey1996]*):

```
set fname <tensorPDBOutput>.pdb
mol new $fname waitfor all
set all [atomselect top all]
set frame 0
set in [open $fname r]
set occupancy {}
while { [gets $in line] != -1 } {
   switch -- [string range $line 0 3] {
      END {
            $all frame $frame
            $all set user $occupancy
            set occupancy {}
            incr frame
         }
      ATOM {
            #column numbers changed to reflect occupancy column
            lappend occupancy [string range $line 54 60];
         }
   }
}
```

To load the script run the following command in VMD terminal window:

```
play <script_name>.vmd
```

By changing the column numbers in the line `lappend occupancy [string range \$line 54 60];` the appropriate column of PDB file can be specified.

To download TensorCalculator use the following link .

Bibliography

[Dill2006] E. H. Dill, *Continuum Mechanics: Elasticity, Plasticity, Viscoelasticity.* Germany: CRC Press, 2006.

[Landau1970] L. D. Landau and E. M. Lifshitz, *Theory of elasticity.* Pergamon Press Ltd., 1970.

[Bonet2008] J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis.* Cambridge University Press, 2008.

[Horstemeyer1999] M. F. Horstemeyer and M. I. Baskes (1999) "Strain tensors at the atomic scale," *MRS Proceedings* **578**: 15–20.

[Gullett2008] P. M. Gullett, M. F. Horstemeyer, M. I. Baskes and H. Fang (2008) "A deformation gradient tensor and strain tensors for atomistic simulations," *Modeling Simul. Matter. Sci. Eng.* **16** (1): 015001.

[Basinski1971] Z. S. Basinski, M. S. Duesbery, and R. Taylor (1971) "Influence of shear stress on screw dislocations in a model sodium lattice," *Can. J. Phys.* **49** (16): 2160-2180.

[Cormier2001] J. Cormier, J. M. Rickman, and T. J. Delph (2001) "Stress calculation in atomistic simulations of perfect and imperfect solids," *J. Appl. Phys.* **89**: 99-104.

[Delph2005] T. J. Delph (2005) "Local stresses and elastic constants at the atomic scale," *Proc. R. Soc. A* **461**: 1869-1888.

[Ishikura2012] T. Ishikura, T. Hatano, and T. Yamato (2012) "Atomic stress tensor analysis of proteins," *Chem. Phys. Lett.* **539-540**: 144-150.

[Hardy1982] R. J. Hardy (1982) "Formulas for determining local properties in molecular dynamics simulations: Shock waves," *J. Chem. Phys.* **76** (1): 622-628.

[Zandi2005] R. Zandi and D. Reguera (2005) "Mechanical properties of viral capsids," *Phys. Rev. E* **72**: 021917.

[Humphrey1996] W. Humphrey, A. Dalke, and K. Schulten (1996) "VMD: visual molecular dynamics," *J. Mol. Graphics* **14** (1): 33-38.