1 A finite-element formulation of the linear elasticity

1.1 Initial definitions

In the following, several conventions are used to facilitate the layout:

- We distinguish between an equation (using = symbol) and definition (using ≡ symbol)
- Summation over repeated indices is assumed in any term in an equation or on the right side in a definition. Let's take an equation:

$$A_{ij}x_j = b_i \quad \text{where} \quad i = 1 \dots N, \quad j = 1 \dots M \tag{1}$$

  denotes a system of N linear equations where summation over j on the left is assumed:

$$\sum_{j=1}^{M} A_{ij}x_j = b_i \tag{2}$$

  Now let's take a definition:

$$C_i \equiv D_{ij}f_j \quad \text{where} \quad i = 1 \dots N, \quad j = 1 \dots M \tag{3}$$

  denotes a definition of a vector C of size N. Each element of this vector is defined as:

$$C_i \equiv \sum_{j=1}^{M} D_{ij}f_j. \tag{4}$$

- Derivative is denoted by a *comma notation.* So

$$f_{i,j} \equiv \frac{\partial f_i}{\partial x_j} \quad \text{where} \quad i = 1 \dots N, \quad j = 1 \dots M \tag{5}$$

  where $\mathbf{f}$ is a vectorial function (of dimension N) of M variables $x_1, \dots x_N$ and here we take a partial derivative of each component f_i w.r.t. each variable x_j. Therefore, if $\mathbf{f}$ is linear in each component, the term $f_{i,j}$ denotes a matrix of size $N \times M$.

- In all what follows in this document, we suppose that indices i, j, k are dimension-related indices. Since we focus on 3D elasticity, each of them is running over a set $\{1, 2, 3\}$. For example x_i is an element of a vector $\{x_1, x_2, x_3\}$ (an alternative for $\{x, y, z\}$).
- A combination of all that has been said can be shown on the following example:

$$div\boldsymbol{\sigma} \equiv \nabla \cdot \boldsymbol{\sigma} \equiv \sigma_{ij,j} \equiv \sum_j \frac{\partial \sigma_{ij}}{\partial x_j} \tag{6}$$

  where σ_{ij} is a 3×3 matrix, while $\sigma_{ij,j}$ is a column vector of 3 elements (since each element of σ_{ij} is derived w.r.t. x_j and summation over j is applied.

1.2 Linear Elasticity in 3D

We recall that the linear elasticity is given by following kinetic equation:

$$\nabla \cdot \boldsymbol{\sigma} + b = 0 \tag{7}$$

where $\nabla \cdot$ is an alternative notation for a divergence of vector field, $\boldsymbol{\sigma}$ is the stress tensor (3×3 matrix) defined as

$$\boldsymbol{\sigma} \equiv \lambda tr(\mathbf{e})\mathbf{I} + 2\mu\mathbf{e} \tag{8}$$

where λ and μ are the Lamé coefficients and $\mathbf{e}$ is the linearized Green strain tensor (3×3 matrix) defined as:

$$\mathbf{e} \equiv \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^\top). \tag{9}$$

If written in tensor notation, the definition of the 3D linear elasticity problem is:

$$\sigma_{ij,j} + b_i = 0 \tag{10}$$

where

$$\sigma_{ij} \equiv \lambda e_{kk}\delta_{ij} + 2\mu e_{ij} \tag{11}$$

where

$$e_{ij} \equiv \frac{1}{2}(u_{i,j} + u_{j,i}). \tag{12}$$

Notice that only the first one is a "real equation", the other two are just definitions of terms (stress and strain tensors). Also notice that we are still in a "continuous realm": e.g. the unknown variable u are (presumably continuous) functions of x_1, x_2, x_3. The same is true for the applied loads (b_i) and so $\sigma_{ij,j}$ can be regarded as a *functional operator* (in this case a differential operator) applied to functions.

1.3 Application of the finite element (FE) method

The finite element method reformulates a problem given by a differential operator (such as the one in Eq. 10) as a set of algebraic equations which can be solved numerically. Basically, instead of looking for an unknown function (e.g. a solution for an infinite number of degrees of freedom [DoFs]), we reformulate the problem so that we solve for an unknown vector (having a finite number of DoFs) which approximates the continuous solution given relevant interpolation functions. The (non-)linearity of the algebraic equations depends on the (non-)linearity of the initial differential operator.

The FE method has following basic steps:

- Weak reformulation of the problem (mainly the continuous differential operator).
- Discretization of the domain and unknown variables. As a necessary input, the type of elements together with the associated interpolation functions must be chosen here, resulting in the local system of algebraic equations assembled for each element.
- Assembling of the global system of equations from the local element equations.
- Application of the boundary conditions.
- Numerical solution of the system.

In what follows, we apply the FE method on the 3D linear-elasticity problem defined by Eq. 10, focusing on the first two steps.

1.3.1 Weak formulation

Introduction from Wikipedia:

Weak formulations are an important tool for the analysis of mathematical equations that permit the transfer of concepts of linear algebra to solve problems in other fields such as partial differential equations. In a weak formulation, an equation is no longer required to hold absolutely (and this is not even well defined) and has instead weak solutions only with respect to certain "test vectors" or "test functions".

In 3D linear elasticity, the weak (re-)formulation of the problem allows for redistribution of the derivatives and incorporation of the boundary conditions.

The weak form is obtained in two steps: the problem is integrated over the domain Ω where the problem is solved (e.g. the volume of the object undergoing the deformations) and multiplication of the test function (a sort of a "weight" function). Intuitively, the original problem where the equality given by Eq. 10 is required to hold at each point of the domain is replaced by a weaker condition, where only a weighted sum of integrated terms is to be required to hold in the equality. The validity (together with corresponding limitations) of such a weakening of the problem is shown by Lax-Milgram theorem (see corresponding literature or web).

In our case, the weak form of the original problem 10 is simply:

$$\int_\Omega (\sigma_{ij,j} + b_i) w_i d\Omega = 0 \tag{13}$$

where Ω is the problem domain (in 3D so we are assuming a volume integration) and w_i are the test functions. The weak form allows us to redistribute the derivatives using a corollary of divergence theorem (see the first corollary at `http://en.wikipedia.org/wiki/Divergence_theorem`), resulting in:

$$\int_\Omega \sigma_{ij} w_{i,j} d\Omega = \int_\Omega b_i w_i d\Omega + \int_{\partial\Omega} t_i w_i d\Gamma \tag{14}$$

where a new term appears on the right: the integration is taken over the boundary of the volume (denoted as $\partial\Omega$, sometimes replaced by Γ) and t_i are *tractions* applied to the surface of the object.

1.3.2 Discretization

This is the key step in the FE method: the continuous problem is approximated by discretized one. First, the continuous domain is discretized by finite elements, denoted by index e:

$$\Omega \equiv \biguplus \Omega_e \tag{15}$$

where Ω_e is the domain of an element e.

It is supposed that the element is a geometric primitive such as triangle, quad (in 2D), tetrahedron, hexahedron, pyramid, prism (in 3D): i.e. the element is given by a small number of points called nodes. Besides the geometrical description of the element, *interpolation* (alternatively called shape, basis or blending) functions are associated to each node of the element. The interpolation functions can be used to approximate (interpolate) any quantity (displacement, load, stress, strain, temperature, pressure etc.) in any position inside of the element provided the values in the nodes are given.

In the following, an interpolation function associated to an element and a node is denoted φ^{en}. Since we are in 3D, φ^{ne} is a scalar function of x_1, x_2, x_3. Two basic properties are required:

- φ^{ne} has a local support on the element e, i.e., it is non-zero only inside the associated element e and zero anywhere else:

$$\text{loc supp}_e \quad \varphi^{ne} \tag{16}$$

- φ^{ne} is bounded to the node n of the element e:

$$\varphi^{ne}(\mathbf{x}^m) = \delta_{nm} \tag{17}$$

where $\mathbf{x}^m$ is the location of a node m and δ_{nm} is a Kronecker symbol.

Now, the discretization can be applied to the weak form 13: first, instead of having one equation over the volume Ω, we have a sum of terms corresponding to each element Ω_e and the physical quantities (unknown displacements, stresses, strains etc.) are now related to local elements instead of the global domain. Second, the continuous unknown functions u^e are approximated with interpolation functions:

$$u_i^e \approx \varphi^{ne} U_i^{ne} \tag{18}$$

where U^{ne} is an (unknown) displacement of the node n in element e. Since the number of elements is finite and each element has a small number of nodes, U^{ne} is a vector of finite size where each number in the vector corresponds to one DoF of the discretized domain. Notice, that in Eq. 18, summation over n is assumed, whereas no summation is done over e and i (which are taken as the parameters of the definition, although symbol $\approx$ is used instead of $\equiv$). Also notice that now the derivative of the function u can be approximated as follows:

$$u_{i,j}^e \approx \varphi_{,j}^{en} U_i^{en} \equiv \frac{\partial \varphi^{en}}{\partial x_j} U_i^{en} \tag{19}$$

Finally, the *Galerkin method* is used (for the sake of simplicity, the method is not described in details, but it's only used as a "black-box"). This means that the same interpolation functions φ^{ne} (used to approximate the unknown displacements u) are employed as the test functions w_i:

$$w_i^{en} \equiv \varphi^{ne} W_i^{ne} \tag{20}$$

where W_i^{ne} are *virtual displacements* (note that no summation is done in this case as all the indices appear on left).

Putting it all together, the weak form is now discretized as:

$$\sum_e \int_{\Omega_e} \left(\sigma_{ij} \varphi_{,j}^{en} d\Omega\right) W_i^{en} = \sum_e \left(\int_{\Omega_e} b_i \varphi^{en} d\Omega + \int_{\partial\Omega_e} t_i \varphi^{en} d\Gamma \right) W_i^{en} \tag{21}$$

Since the equality must hold for any virtual displacements W_i^{ne}, the above sum can be rewritten as a system of equalities where for each element e we have:

$$\int_{\Omega_e} \sigma_{ij}^e \varphi_{,j}^{en} d\Omega = \int_{\Omega_e} b_i^e \varphi^{en} d\Omega + \int_{\partial\Omega_e} t_i^e \varphi^{en} d\Gamma \tag{22}$$

where the discretized stress tensor σ_{ij}^e is defined as

$$\sigma_{ij}^e = \lambda \varphi_{,k}^{ne} U_k^{ne} \delta_{ij} + \mu(\varphi_{,j}^{en} U_i^{en} + \varphi_{,i}^{en} U_j^{en}) \tag{23}$$

where we already used the definitions Eq. 11,12 and the discretization (Eq. 18, 19).

For the sake of simplicity, we assume the tractions to be zero everywhere (i.e. $t_i^e = 0$) and we assume that loads b_i^e do not depend on the position so it can be moved outside of the integral. This results in simplifying the right-hand side of 22. Thus, after substituting 23 into 22, the discretized formulation over an element e becomes:

$$\int_{\Omega_e} \left[\lambda \varphi_{,k}^{ne} U_k^{ne} \delta_{ij} + \mu(\varphi_{,j}^{en} U_i^{en} + \varphi_{,i}^{en} U_j^{en})\right] \varphi_{,j}^{en} d\Omega = b_i \int_{\Omega_e} \varphi^{ne} d\Omega \tag{24}$$

Notice that

- the term on the left-hand side is linear in U and only derivatives of the interpolation functions φ^{ne} appear there,
- the right-hand side represent applied body loads such as the gravity. In this case, the vector $\mathbf{b}$ is computed as
$$\mathbf{b} = \rho \left[0 \quad -g \quad 0\right]^{\top} \tag{25}$$
where ρ is the density and g is the gravitational acceleration, here applied along $-y$ axis.

1.3.3 Particular case: P1 elements (linear tetrahedra)

The P1 elements are the simplest elements in 3D: each element has four nodes (therefore in all what follows, $n = \in \{1, \ldots 4\}$) and the interpolation functions are linear. The simplest linear function in 3D is defined as:

$$\varphi(x_1, x_2, x_3) = a + bx_1 + cx_2 + dx_3 \tag{26}$$

In what follows, we focus on a single element e (therefore, we can drop the index e in rest of this section). This element has four nodes with positions $\mathbf{x}^n \equiv \{x_1^n, x_2^n, x_3^n\}$, $n \in \{1...4\}$. Now, in order to rewrite the Eq. 24 (which is defined generally for any element of any order) for the given P1 element, we need to obtain four linear interpolation functions φ^n, i.e. we need to determine the coefficients a^n, b^n, c^n, d^n (from Eq. 26). This can be done using the property of the interpolation functions defined by Eq. 17. In practical implementations, this is achieved by constructing a *nodal matrix* V_e:

$$\begin{bmatrix} 1 & x_1^1 & x_2^1 & x_3^1 \\ 1 & x_1^2 & x_2^2 & x_3^2 \\ 1 & x_1^3 & x_2^3 & x_3^3 \\ 1 & x_1^4 & x_2^4 & x_3^4 \end{bmatrix} \tag{27}$$

and solving for linear systems (one for each node, corresponding to one vector of the right-hand side):

$$\begin{pmatrix} 1 & x_1^1 & x_2^1 & x_3^1 \\ 1 & x_1^2 & x_2^2 & x_3^2 \\ 1 & x_1^3 & x_2^3 & x_3^3 \\ 1 & x_1^4 & x_2^4 & x_3^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \left(\begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \tag{28}$$

While the tensor notation was extremely useful in the derivation of weak forms and general equations (such as Eq. 24), the standard matrix (or vectorial) notation is more useful for the implementation purposes. Moreover, *Voigt* notation is usually employed in order to avoid higher-order tensors: for example, a 2nd order tensor in 3D (such as the stress tensor σ_{ij}) is a symmetric 3×3 matrix. In order to reduce its dimensionality, it can be stored as a vector of size 6, using the trick depicted in Fig. 1. Similarly, the unknown vector U_i^n, which is in fact a 4×3 matrix (for each of four nodes, we have three displacements along x_1, x_2, x_3), can be stored as 12×1 vector. This process can be described by introducing a mapping on the indices where the nodal index n and the dimension-related index i (or j) are joined together; thus the new "joined" index p would be computed as $p = 3n + i$ (we are not going to use the joined index, it's only to explain better what is behind these manipulations on the implementation level).

Putting this all together, the general 3D FE-discretized linear elasticity equation (Eq. 24) for an element e can be written as
$$\mathbf{K}^e \mathbf{u}^e = \mathbf{f}^e \tag{29}$$

$$\begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{12} & T_{22} & T_{23} \\ T_{13} & T_{23} & T_{33} \end{bmatrix} = \begin{bmatrix} T_{1} & T_{6} & T_{5} \\ . & T_{2} & T_{4} \\ . & . & T_{3} \end{bmatrix} = \begin{bmatrix} T_{1} \\ T_{2} \\ T_{3} \\ T_{4} \\ T_{5} \\ T_{6} \end{bmatrix}$$

Figure 1: Example of the Voigt notation

where $\mathbf{u}^e$ is a 12×1 vector gathering all the unknowns U_i^n (displacements in each degree of freedom), similarly $\mathbf{b}$ is a 12×1 vector gathering the integrated loads and $\mathbf{K}^e$ is a 12×12 stiffness matrix. This is possible thanks to the fact that the formulation given by Eq. 24 is linear in U, and so all the terms on the left-hand side can be reorganized (using the Voigt notation and the index mapping introduced above) in the matrix $\mathbf{K}^e$ which is of form:

$$\mathbf{K}_e \equiv \int_{\Omega_e} \mathbf{B}_e^\top \mathbf{D}_e \mathbf{B}_e d\Omega \tag{30}$$

where $\mathbf{B}_e$ is a *geometrical* matrix containing only the coefficients obtained by solving the system 28:

$$\mathbf{B}_e = \begin{pmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 & 0 \\ 0 & 0 & d_1 & 0 & 0 & d_2 & 0 & 0 & d_3 & 0 & 0 & d_4 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 & 0 \\ d_1 & 0 & b_1 & d_2 & 0 & b_2 & d_3 & 0 & b_3 & d_4 & 0 & b_4 \\ 0 & d_1 & c_1 & 0 & d_2 & c_2 & 0 & d_3 & c_3 & 0 & d_4 & c_4 \end{pmatrix} \tag{31}$$

and $\mathbf{D}_e$ is a *material* matrix containing the material parameters:

$$\mathbf{D}_e = \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \tag{32}$$

The final remark concerns the integration: since only derivatives of the interpolation function appear on the left hand side of the Eq. 24, only the coefficients b, c and d of the interpolation functions are present in the final matrix form (spatial derivatives of the linear shape functions). Therefore, there is no term inside $\mathbf{K}$ which depends on the spatial coordinates and therefore, the integration over the volume of the P1 element can be replaced by the multiplication by the element volume. For linear tetrahedron, the volume can be computed using the determinant of the vertex matrix $\mathbf{V}_e$ (see Eq. 27). Therefore:

$$\mathbf{K}_e \equiv \int_{\Omega_e} \mathbf{B}_e^\top \mathbf{D}_e \mathbf{B}_e d\Omega \equiv \frac{det(\mathbf{V}_e)}{6} \mathbf{B}_e^\top \mathbf{D}_e \mathbf{B}_e \tag{33}$$

However, on the right hand side, the shape functions φ^{ne} must be integrated over the element volume. Since the shape functions depends on positional variables, it is not possible to replace the integration by the multiplication by the element volume. Instead, *Gauss integration rule* is applied to perform the numerical integration. I.e., the integrand is evaluated in g integration points x_p and the integrated value is obtained as a weighted sum:

$$\int_{\Omega_e} \varphi^{ne}(x) d\Omega \approx \sum_{p=1}^{g} \omega_p \varphi^{ne}(x_p) \tag{34}$$

where ω_p are weights associated to each integration point. The values of points and weights can be found for given volume element.

The same approach must be applied to the left hand side if the integrand depends on the spatial variables. This is the case when a higher-order shape functions are used (i.e., quadratic functions).

1.4 Final remarks

The previous section gives a complete set of instructions how to implement a linear elasticity FEM. However, before the system 29 can be solved (by direct or iterative solver), two steps are needed:

- global assembly of the system: the Eq. 33 gives a prescription for one element. In reality, such matrix must be computed for each element and put together to form the global system. Basically, during the global assembly is done as follows: for a node m where m denotes a global numbering in the FE mesh of the object domain, the contributions of all the elements attached to the node m are added. This is usually done directly during the computation of the local matrix $\mathbf{K}_e$ which is not stored directly, but its elements are added to the global matrix according to the local-to-global index mapping: while locally, the node is represented by the pair (e, n) where e is the element index and n is the local node index in the element (e.g. for P1 $n \in \{1, 2, 3, 4\}$), it has also the global index $\hat{n} = 1 \dots N$ where N is the number of nodes. The mapping is then given as $(n, e) \rightarrow \hat{n}$.

  In 3D case employing the P1 elements, the local matrix $\mathbf{K}^e$ is a 12×12 matrix composed of 3×3 blocks representing the stiffness relations between the nodes of the element e. During the assembly, the block placed at the position (n, m) in the local matrix is added to the global matrix on position $(\hat{n}, \hat{m})$ where $(n, e) \rightarrow \hat{n}$ and $(m, e) \rightarrow \hat{m}$.

- The essential boundary conditions such as the homogeneous Dirichlet condition must be applied using one of the methods introduced in the lecture (penalisation, projection or Lagrange multipliers). This step is performed on the global matrix.

It must be emphasized that linear formulation can be used only for small strains (up to 10%), otherwise, artifacts can be observed such as increase in volume shown in Fig. 2.

Recall that the initial equations given by Eq. 10,12, 11 contain two types of linearities:

- material linearity: the material relation between the stress and strain is linear (also called StVenant-Kirchhoff formulation),

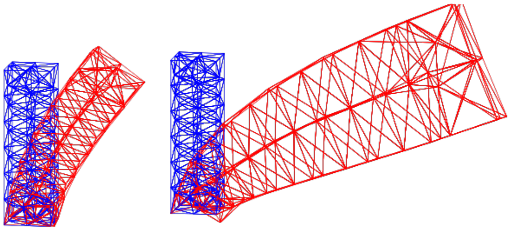

Figure 2: Linear elasticity showing important artifacts in large deformations.

- geometrical linearity: the full Green strain tensor is linearized, i.e. in the strain tensor definition

$$e_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i} + u_{i,j}u_{j,i}) \tag{35}$$

  the last quadratic term is neglected. Basically, this linearlization results in the large-deformation artifacts.

The employment of the full Green tensor with e.g. StVenant-Kirchhoff (or another such as Mooney-Rivlin) material would result in non-linear hyperelastic formulation. However, the resulting equations would be much more complicated, as the resulting formulation (based on the weak form and the linearization) would be non-linear and some iterative solution method would be necessary (e.g. Newton-Raphson). Nevertheless, in this case, Jacobian (a derivative w.r.t. spatial coordinates) must be computed using so called Fréchet derivatives.

An alternative approach is represented by a corotational FEM (see for instance in Nesme, M., Payan, Y., Faure, F.: Efficient, physically plausible finite elements. In Dingliana, J., Ganovelli, F., eds.: Eurographics 2005, Short papers, August, 2005, Trinity College, Dublin, Irlande (2005)) or C.Felippa: A systematic approach to the element-independent corotational dynamics of finite elements, 2000.