OUTLINE

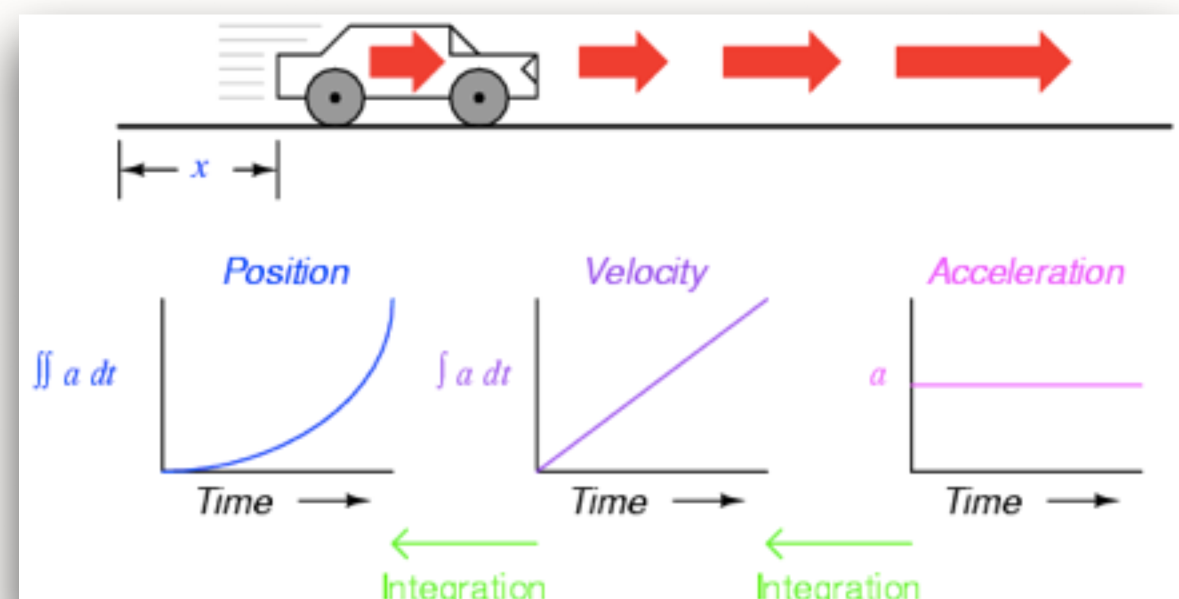
- Models: General Overview
- Mechanics and Continuum Mechanics
- Mechanics of Solid Objects and Elasticity
- Kinematics: displacements, deformations, strains
- Kinetics: forces, pressures, stresses, tractions
- Linear Elasticity: continuous formulation, FEM, solution
- Hyperelasticity: towards non-linear models
- Co-rotational approach: geometry-based compromise

MODELS

- A model is an **abstract structure** that uses **mathematical language** to describe the behaviour of a system.
- typical examples of models:
  - electrophysiological model: describes electrical properties of tissue (e.g. electrophysiological model of heart)
  - model of fluid dynamics: describes behaviour of liquid (e.g. cardiovascular fluid mechanics (blood circulation)
  - biomechanical model of an organ: describes elastic/plastic behaviour of tissues (e.g. hyperelastic model of liver)
- the mathematical language is usually based on differential equations
  - the behaviour is "a change of state" (derivative)

MECHANICS

- area of science dealing with physical bodies subject to force and/or displacements
- **classical** (Newtonian) vs. quantum mechanics :-)
  - **kinematics (geometry of motion):** moving points/bodies without considering the causes of motion
  - **(analytical) dynamics:** relationship between motion of bodies and its causes



$$\mathbf{F} = m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = m\mathbf{a}$$

$$\sum \mathbf{F} = 0 \Leftrightarrow \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = 0$$

CONTINUUM MECHANICS

- deals with the analysis of the **kinematics** and the **mechanical behavior** of materials modeled as a **continuous mass** rather than as discrete particles
- **continuum hypothesis**: well defined properties in infinitely small points (*reference element of volume*)
- **solid mechanics:** study of continuous materials with defined rest shape
- **fluid mechanics:** study of fluid materials (liquids, gases, plasmas)
  - e.g. CFD (computational fluid dynamics)
- obeying common laws**: conservation of mass, energy, [linear and angular] momentum**

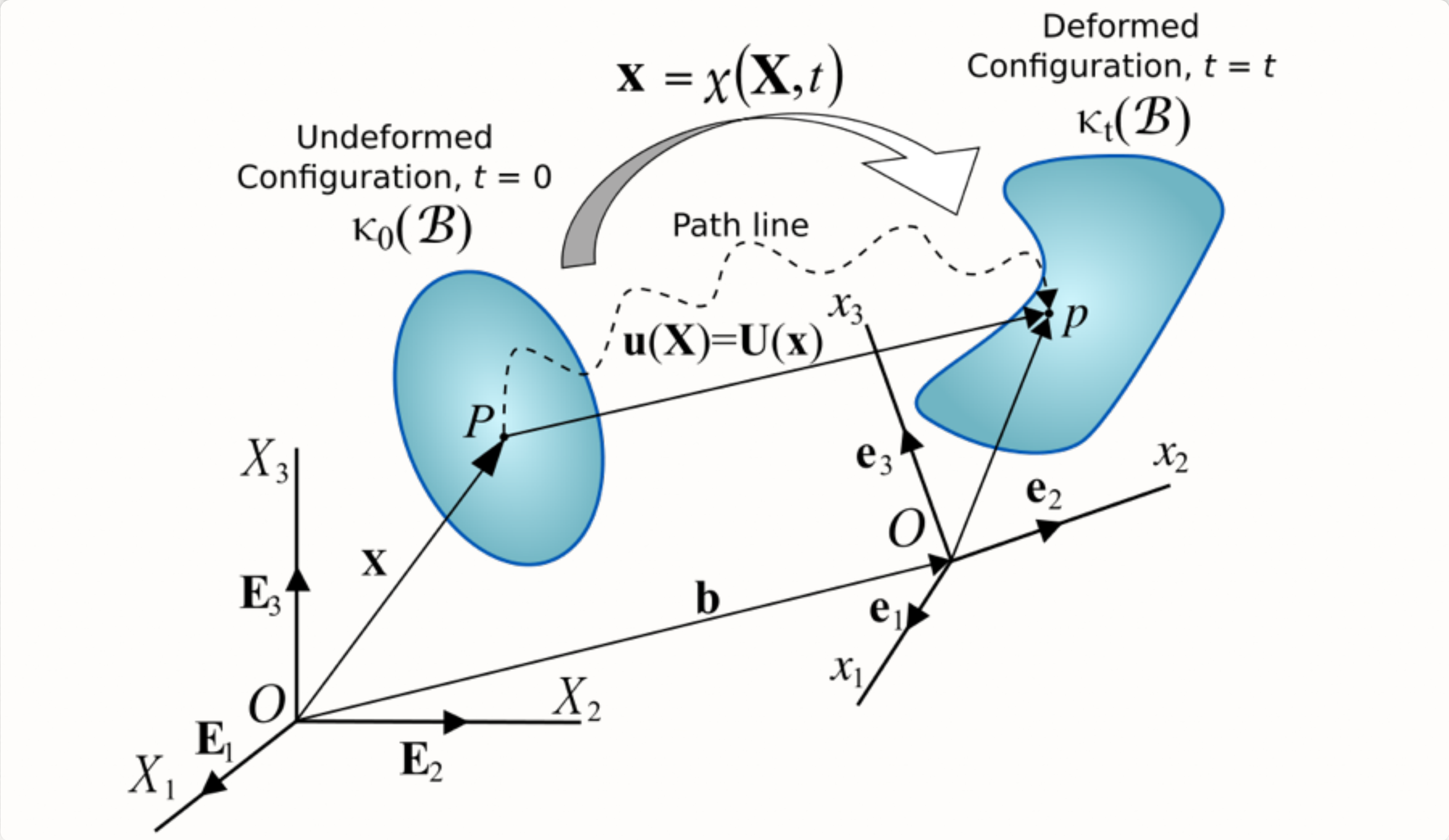
SOLID MECHANICS

- studies the behavior of solid materials, especially their **motion and deformation under the action of forces**, temperature changes, phase changes, and other external or internal agents.
- **elasticity:** describes materials that return to their rest shape after applied stresses are removed
- **viscoelasticity**: elastic material with damping (hysteresis loop)
- **plasticity:** describes materials that permanently deform after a sufficient applied stress
- **thermoplasticity**: coupling between mechanics and thermal properties.

ELASTICITY

- ability of a body to resist a distorting influence or stress and to return to its original size and shape when the stress is removed
- basically, it defines mathematic relation between displacements and applied forces
  - kinematics: relates displacement to strain (geometry)
  - kinetics: relates forces to stresses (e.g. equilibrium)
  - constitutive law: relation between the stress and strain (the material)
- linear elasticity: keeping all relations linear (non-conservative!)
- hypoelasticity: extension of linear elasticity
- hyperelasticity: a family of models (materials), typically used for tissues

TOWARDS THE LINEAR ELASTICITY



VECTOR AND TENSOR FIELDS I

- continuum mechanics: body as a continuum set of particles (3D points)
- initial configuration **X** (X,Y,Z) vs. deformed configuration **x** (x,y,z)
- displacement – vector function in 3D defined for in each particle (vector field)

$$\mathbf{u}(x, y, z) = (u_x(x, y, z), u_y(x, y, z), u_z(x, y, z))$$

$$\mathbf{x} = \mathbf{X} + \mathbf{u}$$

- elasticity theory formulated using tensors
  - similarly as vector field, tensor field is a "tensorial" function defined in each particle (i.e., over the continuous domain)
  - typical operators on fields: gradient, divergence, curl

VECTOR AND TENSOR FIELDS II

Vector-matrix notation:

–using bold symbols: **A, σ** (matrix), **v** (vector)

–derivatives written as operators: gradient: $\nabla f = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^\top f$

Tensor notation:

–summation over repeated indices

$$a_{ij}b_j \equiv \sum_j a_{ij}b_j$$

–derivative using ‘,’ notation

$$f_{i,j} \equiv \frac{\partial f_i}{\partial x_j}$$

Example:

–divergence of a vector field $\mathbf{u}(x, y, z) = (u_x(x, y, z), u_y(x, y, z), u_z(x, y, z))$

$$div\mathbf{u} = (\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}) = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}) \cdot (u_x, u_y, u_z)^\top = \boldsymbol{\nabla} \cdot \mathbf{u} = u_{i,i}$$

STATIC LINEAR ELASTICITY

Kinematics

Kinetics

Constitutive equation

KINEMATICS: DEFORMATION

- deformation field: vector field defined in each point

$$x = X + u(x, y, z)$$

- deformation gradient: 2nd order tensor defined in each point

$$F = I + \nabla u$$

- decomposition of deformation gradient to rotation and stretch tensors

$$F = RU = VR: \qquad R^{-1} = R^\top$$

- right Cauchy-Green deformation tensor (square of local change)

$$C = F^\top F = I + \nabla u + \nabla u^\top + \nabla u^\top \nabla u$$

- alternative: left Cauchy-Green deformation tensor

$$B = FF^\top = I + \nabla u + \nabla u^\top + \nabla u^\top \nabla u$$

KINEMATICS: STRAIN

- **strain:** a description of deformation in terms of relative displacement of particles in the body that excludes rigid-body motions
- different measures of strain: Green, Biot, Almansi, logarithmic strain
- Green strain tensor:

geometric non-linearity

$$E = \frac{1}{2}(C - I) = \frac{1}{2}(\nabla u + \nabla u^\top + \nabla u^\top \nabla u)$$

- linearization:

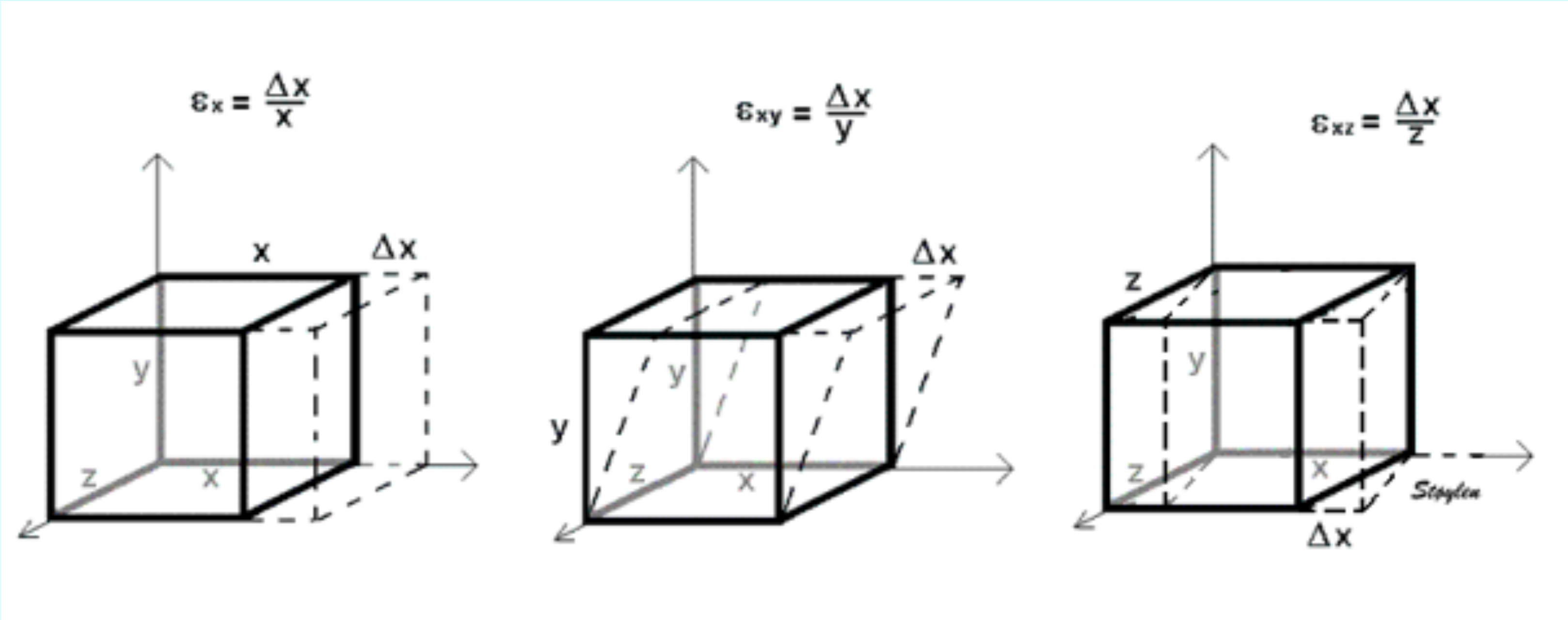
$$\varepsilon = e = \tfrac{1}{2}(\nabla u + \nabla u^\top)$$

$$\begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y}\right) & \frac{\partial u_y}{\partial y} & \frac{1}{2}\left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}\right) & \frac{1}{2}\left(\frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z}\right) & \frac{\partial u_z}{\partial z} \end{bmatrix}$$

KINEMATICS: STRAIN

- components of strain: diagonal + shear strains:

$$\begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y}\right) & \frac{\partial u_y}{\partial y} & \frac{1}{2}\left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}\right) & \frac{1}{2}\left(\frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z}\right) & \frac{\partial u_z}{\partial z} \end{bmatrix}$$



ELASTICITY-BASED MODELING

Kinematics

Strain – Displacement

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^\top)$$

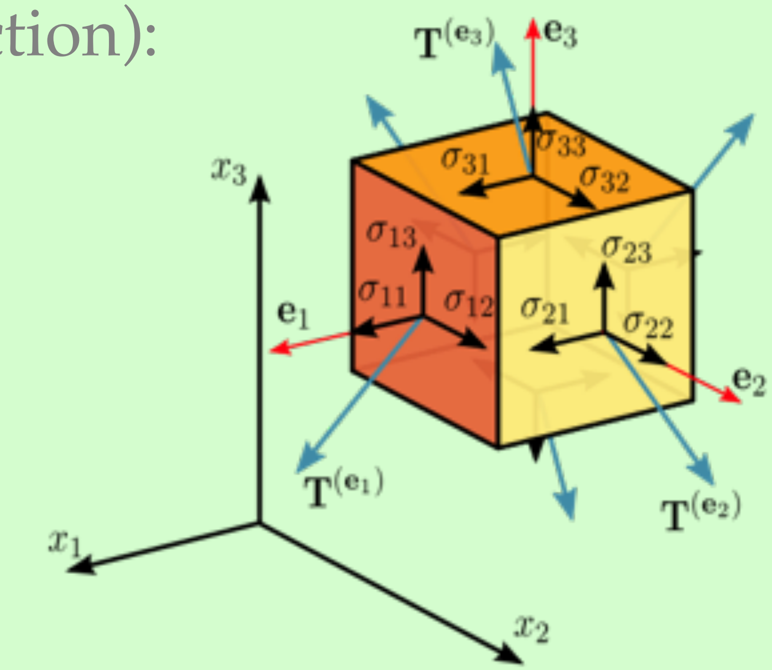
Constitutive equation

Kinetics

KINETICS: STRESS

- **stress:** internal forces that neighboring particles of a continuous material exert on each other
- Cauchy (true) stress tensor: 2nd order tensor that completely define stress at a point
- relates a unit length vector and stress vector: $\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$
- the components of stress vector (surface traction):

$$t_i = \frac{dg_i}{dS}$$



$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \equiv \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \equiv \begin{bmatrix} \sigma_{x} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{y} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{z} \end{bmatrix}$$

STRESS TENSOR

- **stress:** internal forces that neighboring particles of a continuous material exert on each other

- Cauchy (true) stress tensor: 2nd order tensor that completely defines stress at a point

- **conservation of linear momentum**: in static equilibrium, it satisfies equilibrium equation in each point (**b** being the body forces)

$$div\boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{i.e.,} \quad \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{i.e.,} \quad \sigma_{ij,j} + b_i = 0$$

- **conservation of angular momentum:** symmetry (6 components instead of 9)

$$\sigma_{ij} = \sigma_{ji}$$

$$\tau_{xy} = \tau_{yx}$$

$$\tau_{xz} = \tau_{zx}$$

$$\tau_{yz} = \tau_{zy}$$

ELASTICITY-BASED MODELING

Kinematics

Strain – Displacement

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^\top)$$

Constitutive equation

Kinetics

Stress in static equilibrium

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

$$\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$$

CONSTITUTIVE EQUATION

- **Cauchy elastic material:** stress is a function of strain
- **linear elasticity:** stress is a **linear** function of strain
- **Hooke law:** the relation between stress (2nd order tensor) and strain (2nd order tensor) is a 4th order tensor

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \quad \text{i.e.,} \quad \boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}$$

- in general, C has 81 components: however, symmetry of strain and stress reduces the number of components to 21
- for isotropic and homogeneous material, number of parameters is reduced to two Lamé coefficients:

$$\boldsymbol{\sigma} = \lambda \mathbf{I} tr(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon}$$

MATERIAL PARAMETERS

$$\boldsymbol{\sigma} = \lambda \mathbf{I} tr(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon}$$

- in tensorial notation (with Einstein summation convention):

$$\sigma_{ij} = \lambda\delta_{ij}\varepsilon_{kk} + 2\mu\varepsilon_{ij} = \lambda\delta_{ij}u_{k,k} + \mu(u_{i,j} + u_{j,i})$$

- Lamé coefficients: the second is sometimes called shear modulus (G)

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \qquad \mu = \frac{E}{2+2\nu}$$

- where
  - **E** is the Young's modulus [Pa]: stiffness of the material
  - nu is the Poisson's ratio: incompressibility of the material <0,0.5

ELASTICITY-BASED MODELING

Kinematics

Strain – Displacement

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\top})$$

Constitutive equation

Kinetics

Stress, static equilibrium

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

$$\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$$

Stress-strain relation

$$\boldsymbol{\sigma} = \lambda \mathbf{I} tr(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon}$$

PUTTING IT ALL TOGETHER

$$\boldsymbol{\varepsilon} = \tfrac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^\top) \qquad \boldsymbol{\sigma} = \lambda \mathbf{I} tr(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon}$$

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

$$\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$$

- **Navier-Cauchy equation** (see the proof performed by components on LinearElasticity@Wikipedia):

$$(\lambda + \mu)\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{u}) + \mu \boldsymbol{\nabla}^2 \mathbf{u} + \mathbf{b} = 0$$

- tensor notation:

$$(\lambda + \mu) u_{j,ij} + \mu u_{i,jj} + b_i = 0$$

- per component: $K \in \{x, y, z\}$

$$(\lambda + \mu)\frac{\partial}{\partial K}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right) + \mu\left(\frac{\partial^2 u_K}{\partial x^2} + \frac{\partial^2 u_K}{\partial^2 y^2} + \frac{\partial^2 u_K}{\partial z^2}\right) + b_K = 0$$

THE PROBLEM TO SOLVE

- the body given by a continuous domain $\tilde{\Omega}$ with boundary $\tilde{\Gamma} = \partial\tilde{\Omega}$
- Navier-Cauchy equation holds for every point of the domain (***f_i*** being body forces per unit volume)

$$(\lambda + \mu)u_{j,ij} + \mu u_{i,jj} + b_i = 0$$

- **essential boundary conditions** ***has to be defined*** on a part of the boundary (to choose the particular solution of N.-C. PDE

$$u_i^p = \bar{u}_i^p \quad \text{for} \quad p \in \tilde{\Gamma}_E \quad \text{where} \quad \tilde{\Gamma}_E \subset \tilde{\Gamma} \quad \text{and} \quad \tilde{\Gamma} = \partial\tilde{\Omega}$$

- **natural boundary conditions** ***can be defined*** on a part of the boundary (i.e., tractions ***T*** along normal ***n*** in point *p*)

$$T_i^p = \sigma_{ij} n_j^p \quad \text{for} \quad p \in \tilde{\Gamma}_N \quad \text{where} \quad \tilde{\Gamma}_N \subset \tilde{\Gamma} \quad \text{and} \quad \tilde{\Gamma} = \partial\tilde{\Omega}$$

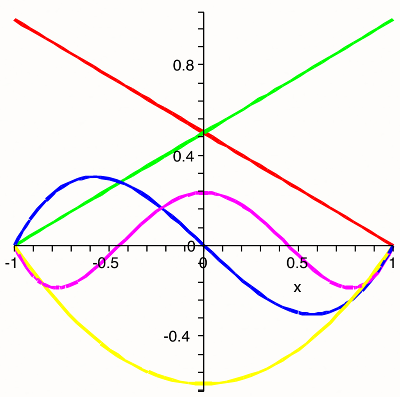
CONTINUOUS VS. DISCRETE SOLUTION II

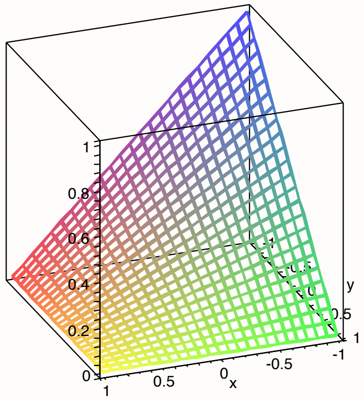
- the only feasible way – **discretization:** approximate the original continuous quantities by discrete (piecewise) functions:

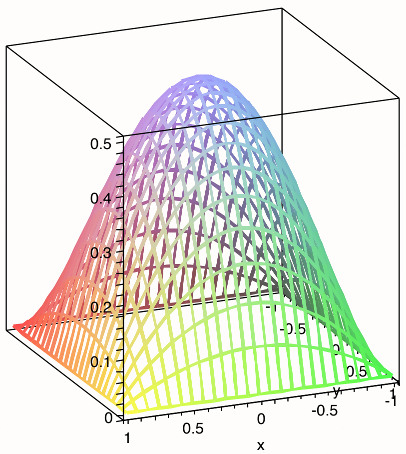
$$\mathbf{u}(\mathbf{x}) \approx \sum_n \mathbf{U}_n \varphi_n(\mathbf{x}) \qquad \frac{\partial \mathbf{u}(\mathbf{x})}{\partial x} \approx \sum_n \mathbf{U}_n \frac{\partial \varphi_n(\mathbf{x})}{\partial x}$$

- central role of the interpolation (basis, shape, blending) functions
- required properties:
  - local support:
    the function is non-zero only inside the element
  - bound to a node **n**:

$$\varphi_n(\mathbf{x}_m) = \delta_{nm}$$







FINITE ELEMENT METHOD

First appeared in 40s and 50s (civil engineering, aeronautics).

1. **Weak formulation** of the continuous differential problem
   – integration over domain and multiplication by test functions
2. **Discretization**
   – discretization of the domain by the elements
   – discretization of the variable and the operator
   – integration over element volume (quadratures)
3. Global **assembling** of the algebraic system of equations
   – imposing the compatibility between the elements
4. Imposition of the **essential boundary conditions**
5. **Numerical solution** of the algebraic system

EXAMPLE: STATIC LINEAR ELASTICITY (SLE)

Given relations (in tensor notation)

Newton's law (kinetics)

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

linearized strain (kinetics)

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

linear material (constitutional law)

$$\sigma_{ij} = \lambda e_{kk}\delta_{ij} + 2\mu e_{ij}$$

Weak form of the Newton's equation (Lax-Milgram lemma)

–***integration*** over the volume

–multiplication by a ***test functions*** w_i

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

The integral over volume allows to distribute the derivatives

–application of chain rule

–**divergence theorem**

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

– no derivative of the stress tensor

– the only derivative applied to the test function on the left side

– $\mathbf{t_i}$: tractions defined over the surface $\partial\Omega$ (**natural boundary conditions**)

SLE: DISCRETIZATION AND GALERKIN METHOD

The actual weak form: $$\int_\Omega \sigma_{ij} w_{i,j} d\Omega = \int_\Omega b_i w_i d\Omega + \int_{\partial\Omega} t_i w_i d\Gamma$$

where: $$\sigma_{ij} = \lambda e_{kk}\delta_{ij} + 2\mu e_{ij} \qquad e_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$$

Domain discretization by elements ***e***: $\tilde{\Omega} \approx \Omega = \biguplus_e \Omega_e$

– element ***e*** given by ***N*** nodes

– each element "equipped" with interpolation functions $\varphi^{en}(x,y,z)$

– index ***n***: node of the element (therefore ***N*** interpolation functions per element)

Galerkin method: use the same interpolation functions to discretize the test functions ***w*** and the solution ***u*** over an element ***e***:

$$w_i = \varphi^{en} W_i^{en} \qquad u_i = \varphi^{en} U_i^{en}$$

(note: no summation over ***e***!)

Example of derivative:

$$w_{i,j} = \varphi^{en}_{,j} W_i^{en}$$

SLE: GALERKIN METHOD II

Discretized week form: $$\sum_e \int_{\Omega_e} \sigma_{ij}\varphi^{en}_{,j} W^{en}_i d\Omega = \sum_e \int_{\Omega_e} b_i \varphi^{en} W^{en}_i d\Omega + \int_{\partial\Omega_e} t_i \varphi^{en} W^{en}_i d\Gamma$$

where: $$\sigma_{ij} = \lambda e_{kk}\delta_{ij} + 2\mu e_{ij} \qquad e_{ij} = \frac{1}{2}(\varphi^{en}_{,j} U^{en}_i + \varphi^{en}_{,i} U^{en}_j)$$

Galerkin method: the equations hold for any virtual displacement *W_i*:

$$\sum_e \int_{\Omega_e} (\sigma_{ij}\varphi^{en}_{,j} d\Omega)\, \cancel{W^{en}_i} = \sum_e \left(\int_{\Omega_e} b_i \varphi^{en} d\Omega + \int_{\partial\Omega_e} t_i \varphi^{en} d\Gamma\right) \cancel{W^{en}_i}$$

For each element *e*, we have the local equation:

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

where: $$\sigma_{ij} = \lambda \varphi^{ne}_{,k} U^{ne}_k \delta_{ij} + \mu(\varphi^{en}_{,j} U^{en}_i + \varphi^{en}_{,i} U^{en}_j)$$

SLE: THE ELEMENT EQUATION

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

where: $$\sigma_{ij} = \lambda \varphi^{ne}_{,k} U^{ne}_k \delta_{ij} + \mu(\varphi^{en}_{,j} U^{en}_i + \varphi^{en}_{,i} U^{en}_j)$$

Right-hand side:

– we consider tractions to be zero and

– body forces to be constant w.r.t. space

$$b_i \int_{\Omega_e} \varphi^{ne} d\Omega$$

Left-hand side:

– clearly linear in ***U*** being the unknown displacements in nodes n=1...N

$$\int_{\Omega_e} \lambda \varphi^{ne}_{,k} U^{ne}_k \delta_{ij} + \mu(\varphi^{en}_{,j} U^{en}_i + \varphi^{en}_{,i} U^{en}_j)\varphi^{en}_{,j} d\Omega$$

– since linear, the left-hand side can be re-organized to $K^{en}_{ij} U^{en}_j$

VOIGT NOTATION

Left-hand side:

$$\int_{\Omega_e} \sigma_{ij} \varphi^n_{,j} d\Omega \qquad \text{with} \qquad \begin{aligned} \sigma_{ij} &= \lambda e_{kk} \delta_{ij} + 2\mu e_{ij} \\ e_{ij} &= \frac{1}{2}(\varphi^n_{,j} U^n_i + \varphi^n_{,i} U^n_j) \end{aligned}$$

– the tensor notation has been useful to derive the final form
– for implementation purposes, Voigt notation is usually employed where 3x3 symmetric 1-order tensor is stored as 6x1 vector:

$$\begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{12} & T_{22} & T_{23} \\ T_{13} & T_{23} & T_{33} \end{bmatrix} = \begin{bmatrix} T_1 & T_6 & T_5 \\ . & T_2 & T_4 \\ . & . & T_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

SLE: STRESS-STRAIN MATRIX **D**

Applying the Voigt notation to the stress–strain relation $\sigma_{ij} = \lambda e_{kk}\delta_{ij} + 2\mu e_{ij}$ results in following matrix equation (derivation is straightforward:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{pmatrix} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} e_{11} \\ e_{22} \\ e_{33} \\ 2e_{12} \\ 2e_{13} \\ 2e_{23} \end{pmatrix}$$

The matrix in the middle is 6x6 stress-strain matrix (denoted further as **D**).

Before encoding the rest into matrices we have to choose the interpolation functions!

$$\int_{\Omega_e} \sigma_{ij}\varphi^n_{,j} d\Omega \qquad e_{ij} = \frac{1}{2}(\varphi^{en}_{,j}U^{en}_i + \varphi^{en}_{,i}U^{en}_j)$$

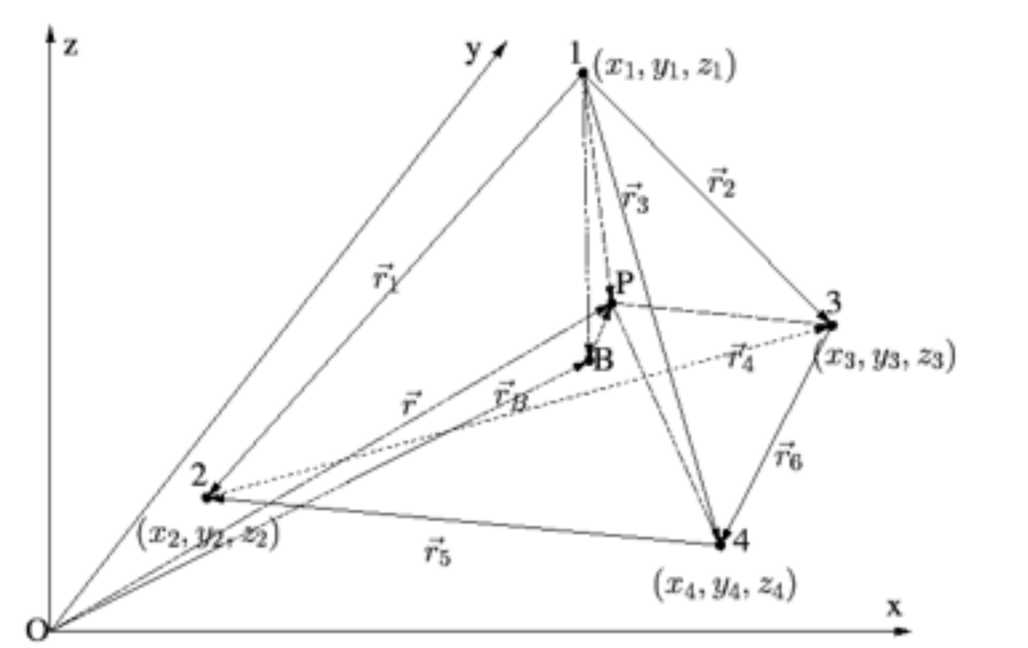
Note that only derivatives of interpolation functions appear in the formulation.

P1: TETRAHEDRAL LINEAR ELEMENT

– ***tetrahedral***: simplex in 3D having four nodes
– ***linear*** since we choose linear interpolation functions:

$$\varphi(x, y, z) = a + b(x) + c(y) + d(z)$$

(a general linear function in 3D)



– how to find the coefficients a,b,c,d? Recall the basic property of an interpolation function:

$$\varphi^i(x_j, y_j, z_j) = \delta_{ij} \qquad i, j \in 1, \ldots N$$

(the value of an interpolation function associated to a node ***i*** is 1 when evaluated in that node [x_i, y_i, z_i] and zero in any other node [x_j,y_j,z_j])

SLE&P1: COMPUTING THE SHAPE FUNCTIONS

Linear P1 (Lagrangian) tetrahedral element

– putting the condition into a matrix form gives:

$$\begin{pmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \left(\begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

– denoting **V** the matrix on the left (***nodal matrix***), 4 instances of coefficients corresponding to 4 interpolation functions (associated to each node) can be computed as columns of the $\mathbf{V}^{-1}$ (recall the requirements for mesh quality!)
– recall also that only derivatives of interpolation functions are present in the formulation (so only coefficients b,c,d) will be used

SLE&P1: STRAIN-DISPLACEMENT MATRIX $\mathbf{B}$

Using the Voigt notation and assuming the linear P1 tetrahedra used for discretization, the left-hand side

$$\int_{\Omega_e} \sigma_{ij}\varphi^n_{,j} d\Omega \qquad \sigma_{ij} = \lambda e_{kk}\delta_{ij} + 2\mu e_{ij} \qquad e_{ij} = \frac{1}{2}(\varphi^{en}_{,j}U^{en}_i + \varphi^{en}_{,i}U^{en}_j)$$

can be rewritten in matrix form as:

$$\int_{\Omega_e} \mathbf{B}_e^\top \mathbf{D}_e \mathbf{B}_e d\Omega$$

$$\mathbf{D_e} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}$$

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

SLE&P1: LOCAL STIFFNESS MATRIX

What about the integration?
– recall that only derivatives of shape functions appear in the formulation
– since interpolation functions are linear, only coefficients b,c,d appear in the matrices
– therefore, the integrand is constant (does not depend on x,y,z)
– integration of a constant over a tetrahedron is computed by multiplication of the constant by the volume of the tetrahedron
– the volume of a tetrahedron is given by determinant of nodal matrix: $\mathcal{V}_e = \frac{|\mathbf{V}_e|}{6}$
– the final form is therefore:

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}_e^\top \mathbf{D}_e \mathbf{B}_e d\Omega = \frac{|\mathbf{V}_e|}{6} \mathbf{B}_e^\top \mathbf{D}_e \mathbf{B}_e$$

– the local matrices $\mathbf{K_e}$ are assembled into a global matrix $\mathbf{K}$
– the contribution from different elements to the same node are added (*globalization matrix*)

ASSEMBLING THE GLOBAL SYSTEM

- the procedure now gives 12x12 matrix (4x4 block matrix where each block (i,j) corresponds to stiffness relation between nodes n and m (n,m=1…4)
- global assembly:
  - mapping for each node from local to global indices: (e,n) -> $\mathbb{n}$
  - the block ($\mathbb{n}$,$\mathbb{m}$) from matrix associated to element e is added to the global block at position ($\mathbb{n}$,$\mathbb{m}$) in the global matrix
  - usually is done directly during the computation of local matrix
  - the global matrix is a 3Nx3N block matrix where N is the total number of DOFs (and 3N is thus the number of degrees of freedom)

BOUNDARY CONDITIONS

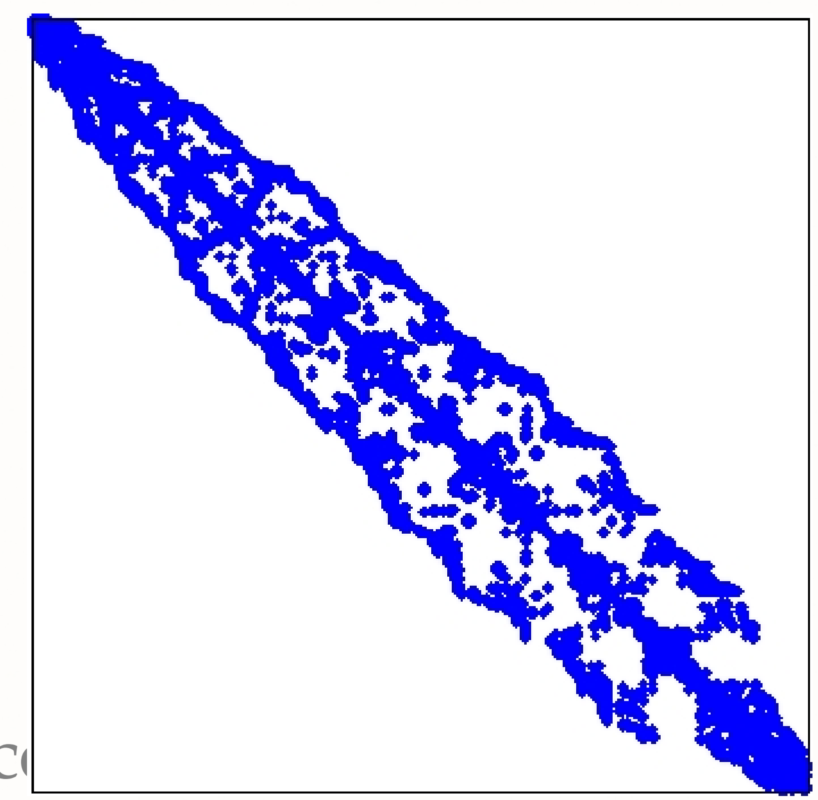
- choosing a particular solution (otherwise **K** singular)
- several options to impose a Dirichlet boundary condition u_i=V
  - **elimination** (projection):
    — left side: K(i,k) = K(k,i) = 0 for all k ≠ i, K(i,i) = 1
    — right side: f(i) = V ("pseudo-loads")
    — not very flexible and difficult to parallelize
  - **penalization**: adding a penalization term to impose the boundary condition (reduces the "quality" of matrix in terms of the condition number)
  - **Lagrange multipliers**: changes the properties of the matrix (larger, possibly indefinite)

THE GLOBAL STIFFNESS MATRIX

- linear relation between forces (**f**) and displacements (**u**):

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

- encoding relations between nodes
- highly sparse (<3% of non-zero)
  - non-zero blocks only for combinations of nodes connected by a mesh edge
  - suitable representation [**i j K**]
  - efficient matrix–vector multiplication
- regular after the imposition of boundary c



- symmetric, positive-definite, sparsity pattern depends on node numbering (can be improved e.g. by Metis)

PRACTICAL MATRIX MANIPULATION

- sparse matrices generated from the FE formulation
  - only a small fraction of entries non-zero (<3%)
  - system of N nodes in 3D results in size of $(3N)^2$
  - practical example: 10000 nodes in double (4B): 3.4GB
  - but 3.3GB are zeros...
  - common format: **i j A_{ij}** (137MB, 2 x int + 1 x double)
  - row vs. column compressed
  - sometimes storing both representations can be practical

SYSTEMS OF *LINEAR* EQUATIONS

- scalar case: $ax = b \quad \rightarrow \quad x = \dfrac{b}{a}$

- vectorial case: $\mathbf{Ax} = \mathbf{b} \quad \rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

- properties of **A** (considered being a square matrix)
  - regular matrix: inverse $\mathbf{A}^{-1}$ exists
  - symmetric: equals to transpose, $\mathbf{A}^{T} = \mathbf{A}$
  - positive-definite: $\mathbf{z}^{T}\mathbf{Az}$ is positive for a vector **z** (eigenvalues)
  - orthogonal matrix: $\mathbf{A}^{T} = \mathbf{A}^{-1}$ (representation of rotations)

DIRECT SOLUTION OF LINEAR SYSTEM

- solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$
- direct solutions: the inverse $\mathbf{A}^{-1}$ computed explicitly as factorization
  - for cases when you need to recompute Ax=b′ for another b′
  - 2 phases: decomposition (factorisation), solution (back-substitution)
  - **Cholesky decomposition**: **A = LL^T** (L lower triangular matrix): *symmetric positive-definite matrices*, most optimal (num. of operation)
  - **LDL decomposition: A = LDL^T (D** diagonal), works for some *indefinite matrices* where Cholesky fails
  - **LU decomposition**: (U upper triangular matrix), general case, modified Gaussian elimination (Doolittle, Crout algorihms, pivoting)

ITERATIVE SOLUTION OF LINEAR SYSTEM

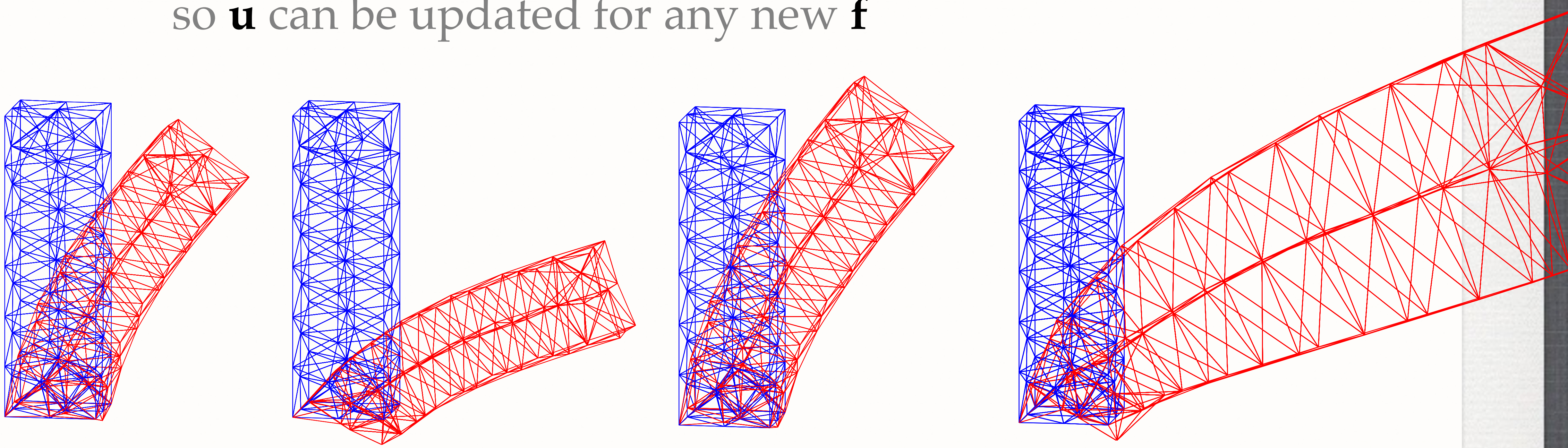
- solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$
  - will depend on properties of A
- iterative solutions: the inverse $\mathbf{A}^{-1}$ is not assembled explicitly
  - start with an estimation $\mathbf{x}^{(0)}$ and iterate until $|\mathbf{A}\mathbf{x}^{(i)} - \mathbf{b}| < \mathbf{e}$ (stopping criterium usually more complicated, absolute vs. relative residual)
  - **conjugate gradients (CG)**: for symmetric, positive-definite matrices (see *Shewchuk: Conjugate gradients without agonizing pain)*
  - **bi-conjugate gradient (BiCG)**: generalization for non-symmetric
  - **generalized minimal residual (GMRES)**: any regular matrix
  - **preconditioned versions:** approximation of $\mathbf{A}^{-1}$

ISSUES WITH LINEAR ELASTICITY

- after imposition of the boundary conditions, the system can be solved

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

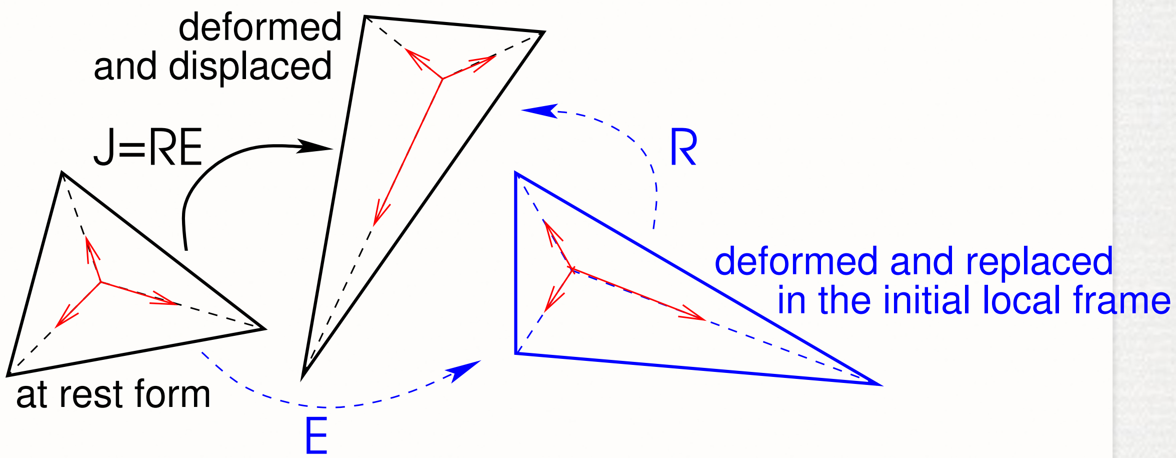
  - iterative: even the matrix **K** does not have to be assembled
  - direct: the both **K** and $\mathbf{K}^{-1}$ are assembled and stored explicitly, so **u** can be updated for any new **f**



- linearized Green strain does not work for large deformations

TOWARDS NONLINEAR: CO-ROTATIONAL FORMULATION

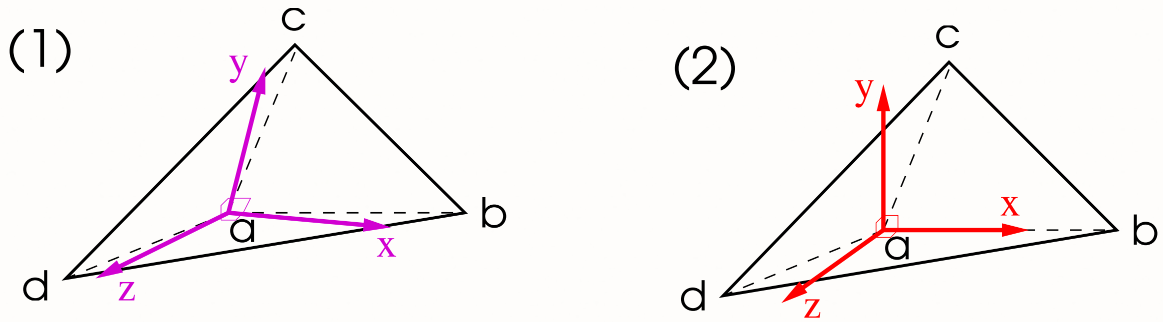
- an extremely successful approach in soft-tissue modeling allowing for large displacements (but supposing small strains)
  - *C.Felippa: A systematic approach to the element-independent corotational dynamics of finite elements, 2000*
- uses the linear-elasticity but co-rotational strain
- the simulation is performed in small steps and in each step:



  - the actual deformation of every element e is decomposed into rigid and deformable components w.r.t. the initial configuration
  - the rigid component is given by a rotation $\mathbf{R_e}$ of the component
  - the local stiffness matrix $\mathbf{K_e}$ is updated as $\mathbf{R}_e^\top \mathbf{K}_e \mathbf{R}_e$

CO-ROTATIONAL FORMULATION II

- the matrix **K** is not constant anymore (**K** => **K**(**u**))
  - the rotational matrices $\mathbf{R_e}(\mathbf{u})$ depend on the actual **u**
  - in each step, Newton-Raphson method should be performed, actually, works quite stably even if only one iteration is performed
- the decomposition can be performed by various methods
  - choosing the basis
  - polar(1), QR(2), SVD



- although the large deformations are simulated realistically, only small strains are handled correctly
- more information about the implementation in SOFA:
  - *M.Nesme et al.: Efficient, physically plausible finite elements, 2005*