Biological Tissue Movement

Oleksandr Hubanov
Vilnius Gediminas Technical University

November 14, 2019

Research aim

Mechanic equivalent of fibrous tissue could be described as system of 1D rods. Example of such flat 2D system are shown on figure 2.

Figure 1: 1D Rod system in global coordinate system

System consists of discrete elements e_1, e_2 and e_3. All elements are connected to each other through nodes n_2, n_3 and to special points through n_1 and n_4. Each element e_n of system has own orientation in global coordinate system. From schematic representation of node comes that all vector variables of node should be calculated in global coordinate system and element variables in own local coordinate system(figure 2). For transformation between coordinate systems direction cosine matrix (DCM)(1) can be used.

$$DCM = \begin{bmatrix} cos(X,x) & cos(X,y) & cos(X,z) \\ cos(Y,x) & cos(Y,y) & cos(Y,z) \\ cos(Z,x) & cos(Z,y) & cos(Z,z) \end{bmatrix} \tag{1}$$

where $\{X,Y,Z\}$ is global coordinate system and $\{x,y,z\}$ is local coordinate system.

According to primitive scheme of node 2, mass of each node can be calculated, like sum of half mass of each element, which acting in node. $m_n = \sum_e m_e/2$

In case that node does not have external interrupt, like pressure or other applied force, schematic represent of node can be as on figure 2.

Figure 2: Extracted node from system

Mathematical model of discrete system is expressed by equations of nodes motion. As elements is 1D, nodes will be 1D as well. All system acting in global coordinate system $\{X,Y,Z\}$ and each element acting in own local coordinate system $\{x,y,z\}$.

Mass Spring Model

Linear deformation

For investigating motion of any mechanical system comes to integration motion equation of discrete parts of this system. In case of MSM, system described as nodes connected to each other by 1D rod elements2. (5). For this propose need to express all acting forces in node(2) for each node in relation to their application place.

$$\vec{F}_n = \vec{F}_{ext} + \sum \vec{F}_{elem} \times [DCM] + \vec{F}_{press} \times [DCM] \tag{2}$$

F_{ext} is external load force, applied to node in global coordinates. Value of this force for each time step is loaded from list of loads.

F_{press} is external pressure and can be described like force applied to element in local coordinates.

F_{elem} is sum of internal forces of each element, which acting in node. From each element counts only half of force to node, other half going to neighbour node. In case of 1D element system, internal force of each element can be express like axial force and it is equal to integral of stress over area:

$$N(x) = \int_A \sigma dA \tag{3}$$

For 1D rod system F_{elem} can be expressed like:

Is it has to be divided by 2? in this case how to calc mass of node?

$$F_{elem} = \sum_e N_e(x)/2 \tag{4}$$

The motion of nodes can be expressed by Newton's equation of motion5. As 1D element was choosed as discrete element, only the normal component of the translational motion is considered, the equation reduces to

$$m\ddot{\vec{x}} + k\vec{x} + c\dot{\vec{x}} = \vec{F}_n \tag{5}$$

where F_n is force acting in node n, m – mass of node and $\ddot{x}$ is acceleration, k – stiffness coefficient of node, x is displacement of node and c – damping coefficient of node, $\dot{x}$ is velosity of node, initial conditions are: $x(0) = 0$ and $\dot{x}(0) = V_0$. According to 2, kx are described by $\sum F_{elem}$, $c\dot{x}$ are described by F_{press} and motion equantion will become to:

$$\vec{F}_{press} + \sum F_{elem} + m\ddot{\vec{x}} = \vec{F}_n \tag{6}$$

where F_{elem} is sum of node's elements forces, m – mass of node and $\ddot{x}$ is acceleration. In static case $m\ddot{x}$ equal to zero, due zero acceleration, because no external force is applied and equantion of node is balansed. But if external force are applied, motion will start, motion equation will become not balansed7, mass start to act as inertia.

$$\vec{F}_{press} + \sum \vec{F}_{elem} + m\ddot{\vec{x}} = F_n + \vec{F}_{ext} \tag{7}$$

Equations of motion for Euler's scheme of integration can be described like:

$$\ddot{\vec{x}}(t + \Delta t) = \vec{F}_n/m \tag{8}$$

$$\vec{\dot{x}}(t + \Delta t) = \vec{\dot{x}}(t) + \vec{\ddot{x}}(t)\Delta t \tag{9}$$

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{\dot{x}}(t)\Delta t \tag{10}$$

Element force becomes from physical deformation of element. In linear case of study, deformation of element much less compare to element dimensions. It is expressed by linear geometry equation(11), which showing relation between initial length of element and length in Δt state.

$$\varepsilon = \frac{dU}{dx} = \frac{l(\Delta t) - l(t)}{l(t)} \tag{11}$$

According to Hook law $\sigma = \varepsilon E$ and linear geometry equation (11), inner force can be described as:

$$N(x) = \int\limits_{Al} \varepsilon E dl dA = EA \int\limits_{l} \varepsilon dl = \frac{EA}{l(t)} * (l(t + \Delta t) - l(t)) \tag{12}$$

where $l(\Delta t)$ is current length of element, $l(t)$ length of element at previuous time moment, E – Young's modulus for element material, A – volume of element. To be able to integrate equation of motion, need to express deformation in equation (12) by differences between displacements of nodes, to which element is connected:

$$N(x) = \frac{EA}{l(t)} * (x_i(t) - x_j(t)) \tag{13}$$

Let's try to describe minimal possible way to get simulation of such structure as shown on figure 1.

```
initialization;
// Preform integration nodes movement over time
// Initial state is x(0) = ẋ(0) = ẍ(0) = 0
// Iterate over time
while t < t_end − 1 do
    // Iterate over nodes
    foreach node in nodes do
        // get external force loading
        F_ext = getFext(node, t);
        // push F_ext into current node force F_n
        F_n = F_n + F_ext;
        // calc F_elm from previous displacement
        foreach link in neighbours do
            F_elm = getFelm(node, link);
            // convert F_elm from local to global coordinates
            F_elm G = F_elm × [DCM];
            // push F_elm G into current node force F_n
            F_n = F_n + F_elm G;
        end
        // calc pressure force
        F_press = getFpress(node, t);
        // convert F_elm from local to global coordinates
        F_press G = F_press × [DCM];
        // push F_elm G into current node force F_n
        F_n = F_n + F_press G;
        // integrate collected F_n to get derivatives of x for t + Δt
        [x(t + Δt), ẋ(t + Δt), ẍ(t + Δt)] = integrate(F_n, x(t), ẋ(t), ẍ(t) )
    end
end
```