

# Image-based point cloud Hamiltonian dynamic analysis for biomechanical systems

A. I. Nadareyshvili<sup>1</sup>, K.B. Chandran<sup>2</sup>, Jia Lu<sup>3\*</sup>

<sup>1</sup>*Mechanical Engineering Research Institute, Russian Academy of Science, Moscow, Russia.*

<sup>2</sup>*Department of Biomedical Engineering, the University of Iowa, Iowa City, IA 52242, U.S.A.*

<sup>3</sup>*Department of Mechanical and Industrial Engineering, Center for Computer Aided Design,  
The University of Iowa, Iowa City, IA 52242, U.S.A.*

## Abstract

The paper presents a discrete method for the dynamic analysis of biomechanical systems. The method works directly on a point cloud representation of a material domain. An elastic body is represented by a set of interacting particles, and the dynamic behavior is described by Hamilton's equations. A first order symplectic integrator, which also conserves the total linear and angular momentum, is utilized for numerical integration. The discrete formulation and integration method are presented in detail. Numerical examples are presented to show the numerical properties and demonstrate the method.

**Keywords:** Point-cloud method; image-based analysis; Hamiltonian system; symplectic integration; moving least square approximation.

## 1. INTRODUCTION

The finite element method (FEM) is undoubtedly the most commonly used method for solving partial differential equations arising in engineering analysis. Utilities of FEM in biomechanical systems and biomedical applications are common. A salient feature of FEM is that it divides a material domain into discrete elements (meshing). Nowadays, meshing a regular geometry is a routine task, and a variety of mesh generation programs are either freely or commercially available. However, generating a high quality, analysis-ready FEM mesh for biological structures still presents considerable challenges. Biological objects often have intricate geometric features that require local mesh refinements, which in turn require user's interference. So far, mesh

\*Corresponding author. Email address: jia-lu@uiowa.edu. Tel: +1-319 3356405. Fax: +1-319 3545669.

generation for biological systems is semi-automatic at the best, and remains as the most of the time consuming step in the process of FEM analysis.

Motivated by the need of alleviating the burden of mesh generation in biomechanical analysis, the authors' group has been developing methods that interface directly with point cloud representations of continuum bodies [1, 2, 3, 4]. In contrast mesh generation, a point cloud can be readily extracted from medical images, and the process of which can be made fully or nearly fully automatic. In [1, 2], we proposed a discrete method of analysis that essentially treats a continuum material body as a set of interacting points. The underlying motivation is akin to that of meshfree methods [5, 6]; however, our method is a further simplification as it eliminates continuous approximation of the primary variables. The method utilizes a generalized finite difference to compute the pointwise gradient or strain, which is subsequently employed into a weak form to derive the discrete governing equations. As demonstrated in [1, 2], the method possesses the same accuracy and convergence rate of the low order finite elements, and is locking-free for nearly incompressible materials. However, implementations in [1, 2], and particularly in [4], are still mesh-based; a Voronoi tessellation or a simplex mesh is required to partition the domain and determine the so-called point volume which is needed in the discrete weak form.

In this paper, we advance the discrete method in two directions. First, we introduce a moving least square (MLS) algorithm for computing the pointwise strain; with the technique, the need for tessellation is circumvented. Second, we introduce a Hamiltonian formulation for the dynamic analysis of the ensuing discrete system. The overall goal is to develop a fully automated procedure that pipelines from image segmentation to analysis with no or minimum user interference. Since the discrete model obtained from medical images are usually large, we intend to use only low order explicit methods for numerical integration. To this end, the dynamics of discrete system is formulated by first order differential equations (Hamiltonian system). The elastic potential is still computed from continuum constitutive equations, and thus enables a direct application of standard tissue constitutive equations. The kinetic energy comprises of pointwise contributions from the point velocity and mass (the mass of a small material volume represented by a point). For regular grids, as ones derived from medical images, the point

volume can be readily inferred from the pixel dimension. To achieve long-term stability in numerical integration, we use symplectic integrators, in particular a first order symplectic integrator. Interested readers are referred to [7,8] and references therein for in-depth coverage on symplectic flow in Hamiltonian systems and symplectic integration in engineering analysis.

The paper starts with the introduction of a MLS approximation for computing the gradient of a function over an arbitrary grid. This gradient operator is utilized to compute the deformation gradient (or strain) of a displacement field. The Hamiltonian representation is introduced in Section 4. The symplectic integrator we utilize turns out to identically conserve the total linear momentum and angular momentum, and these conserving properties are also discussed. A method of point cloud segmentation is introduced in Section 5. Some quasi-static benchmark tests are conducted to demonstrate the properties of the method. An example of heartvalve dynamic analysis is contained in Section 6.

## 2. MLS APPROXIMATION OF GRADIENT

Consider a material body represented by a set of discrete points. Let  $\mathbf{x}_I$  denote the coordinate of the point  $I$ , assume that each point is associated with a set of neighboring points and let  $N_I$  be the set of neighbors of the point  $I$ . Introduce the notations  $\Delta u_{IJ} = u_J - u_I$  and  $\mathbf{R}_{IJ} = \mathbf{x}_J - \mathbf{x}_I$ . We seek a vector  $\mathbf{g}_I$  that approximates the gradient of a continuous function  $u(\mathbf{x})$  at the point  $I$ . The vector is determined by a local least square problem that minimizes the function

$$J = \sum_{J \in N_I} w_{IJ} (\mathbf{g}_I \cdot \mathbf{R}_{IJ} - \Delta u_{IJ})^2 \quad (1)$$

where  $w_{IJ}$  are weights,  $w_{IJ} > 0$ . The extreme of  $J$  is given by

$$\mathbf{0} = \frac{\partial J}{\partial \mathbf{g}_I} = \left[ \sum_{J \in N_I} w_{IJ} \mathbf{R}_{IJ} \otimes \mathbf{R}_{IJ} \right] \mathbf{g}_I - \sum_{J \in N_I} w_{IJ} \mathbf{R}_{IJ} \Delta u_{IJ} \quad (2)$$

Let us assume that the point  $I$  has at least three neighbors not coplanar with  $\mathbf{x}_I$  and among these three, no two of them are co-linear with  $\mathbf{x}_I$ . A point satisfying this requirement is called a (well-connected) regular point, otherwise a hanging point. In this work we assume that all points are regular. For a regular point, the matrix  $\sum_{J \in N_I} w_{IJ} \mathbf{R}_{IJ} \otimes \mathbf{R}_{IJ}$  is invertible. It follows from (2) that

$$\mathbf{g}_I = \sum_{J \in N_I} \left[ \sum_{K \in N_I} w_{IK} \mathbf{R}_{IK} \otimes \mathbf{R}_{IK} \right]^{-1} w_{IJ} \mathbf{R}_{IJ} \Delta u_{IJ} \quad (3)$$

Introducing the vector interpolants

$$\mathbf{R}_{IJ}^* = w_{IJ} \left[ \sum_{K \in N_I} w_{IK} \mathbf{R}_{IK} \otimes \mathbf{R}_{IK} \right]^{-1} \mathbf{R}_{IJ} \quad (4)$$

we can write the discrete gradient as

$$(\nabla^h u)_I \equiv \mathbf{g}_I = \sum_{J \in N_I} \mathbf{R}_{IJ}^* (u_J - u_I) \quad (5)$$

We can also write (5) in a homogenized form. To this end, let

$$\mathbf{R}_{II}^* = - \sum_{J \in N_I} \mathbf{R}_{IJ}^* \quad (6)$$

and expand the neighbor set  $N_I$  to include the point  $I$  itself, namely  $\bar{N}_I = N_I \cup \{I\}$ . With these definitions we write

$$(\nabla^h u)_I = \sum_{J \in \bar{N}_I} \mathbf{R}_{IJ}^* u_J \quad (7)$$

## 2.1 Properties of the gradient operator

1. The interpolants satisfy the relation

$$\sum_{J \in N_I} \mathbf{R}_{IJ}^* \otimes \mathbf{R}_{IJ} = \mathbf{I} \quad (8)$$

**Proof:**

$$\sum_{J \in N_I} \mathbf{R}_{IJ}^* \otimes \mathbf{R}_{IJ} = \left[ \sum_{K \in N_I} w_{IK} \mathbf{R}_{IK} \otimes \mathbf{R}_{IK} \right]^{-1} \left[ \sum_{J \in N_I} w_{IJ} \mathbf{R}_{IJ} \otimes \mathbf{R}_{IJ} \right] = \mathbf{I} \quad (9)$$

This property ensures that the discrete gradient is linearly exact, as shown below.

2. Linear consistency. The gradient operator can exactly reproduce the homogeneous gradient of any linear function.

**Proof.** Consider a linear function  $u^{lin} = \alpha_0 + \alpha_1 x + \alpha_2 y + \alpha_3 z$  where  $\alpha_0, \alpha_1, \alpha_2, \alpha_3$  are arbitrary constants. For this function

$$\begin{aligned} u_J - u_I &= a_1(x_J - x_I) + a_2(y_J - y_I) + a_3(z_J - z_I) \\ &= (\mathbf{x}_J - \mathbf{x}_I) \cdot (a_1 \mathbf{e}_x + a_2 \mathbf{e}_y + a_3 \mathbf{e}_z) = \mathbf{R}_{IJ} \cdot \nabla u^{lin} \end{aligned} \quad (10)$$

where  $e_x, e_y, e_z$  are the basis vectors. Substituting this result into (5) yields

$$(\nabla^h u^{lin}) = \sum_{j \in N_I} \mathbf{R}_{IJ}^* (\mathbf{R}_{IJ} \cdot \nabla u^{lin}) = \left[ \sum_{j \in N_I} \mathbf{R}_{IJ}^* \otimes \mathbf{R}_{IJ} \right] \nabla u^{lin} = \nabla u^{lin} \quad (11)$$

Here the equality (8) is used in the last step.

3. The discrete gradient is first order accurate in the sense that

$$(\nabla^h u)_I = \nabla u(\mathbf{x}_I) + O(h_I) \quad (12)$$

where  $h_I$  is the characteristic length of the cell  $I$ , say  $h_I = \max_{j \in N_I} (\|\mathbf{R}_{IJ}\|)$ .

**Proof.** Let us assume that the function  $u(x, y, z)$  admits the Taylor series expansion at  $\mathbf{x}_I$ , so that

$$u_J - u_I = u_{,x}(x_J - x_I) + u_{,y}(y_J - y_I) + u_{,z}(z_J - z_I) + O(h_I^2) = \mathbf{R}_{IJ} \cdot \nabla u(\mathbf{x}_I) + O(h_I^2) \quad (13)$$

Substituting the Taylor expansion into Eq. (5), we find

$$(\nabla^h u)_I = \sum_{j \in N_I} \mathbf{R}_{IJ}^* [\mathbf{R}_{IJ} \cdot \nabla u(\mathbf{x}_I) + O(h_I^2)] = \nabla u(\mathbf{x}_I) + O(h_I) \quad (14)$$

The first derivative is exactly preserved due to the linear exactness.

## 2.2 Least square gradient on point cloud

The least square gradient (7) applies to an arbitrary point cloud as long as the neighbor set of each point can be properly defined. In an arbitrary point cloud, the identification of neighbors is a nontrivial task. However, for medical images there is a straightforward way to assign neighbors, as images have an underlying grid structure. For simplicity, let us assume that we work directly on the pixel resolution. Each point in this case is the geometric center of a pixel (see [4]). Assume further that the pixels that belong to the material body of interested have been segmented out. We can define the neighbors of a point as those of the adjacent pixels. Depending on how many adjacent pixels exist, there are four topological cases ranging from three to six neighbors. These cases are depicted in Fig. 1.

Note that it is permissible to include some diagonal points in the neighbor set. In this way, a fully surrounded point can have a minimum of six to a maximum of twenty-six neighbors. We conducted numerical tests to evaluate the accuracy and convergence rate of different neighboring schemes (see Section 6.1). It was found that the definition above provided a reasonable trade-off

between accuracy and computation cost. For this reason the diagonal points are not included in the neighbor set in this study.

It is a good place to remark on the weights  $w_{IJ}$ . It is possible to assign weights based on distance. In fact, one could use a distance criterion to identify the neighbor set. Here, as we have defined the neighbor set by other means, and within the set the distances do not vary greatly, the use of variable weights does not have significant advantage. For computational simplicity, a uniform weight  $w_{IJ} = 1$  is employed.

For each of the four topological cases, the least square gradient works out to be:

$$(\nabla^h u)_I = \frac{u_{i,j,k} - u_{i,j-1,k}}{h} \mathbf{e}_x + \frac{u_{i+1,j,k} - u_{i,j,k}}{h} \mathbf{e}_y + \frac{u_{i,j,k} - u_{i,j,k-1}}{d} \mathbf{e}_z \quad (15a)$$

$$(\nabla^h u)_I = \frac{u_{i,j,k} - u_{i,j-1,k}}{h} \mathbf{e}_x + \frac{u_{i+1,j,k} - u_{i,j,k}}{h} \mathbf{e}_y + \frac{u_{i,j,k+1} - u_{i,j,k-1}}{2d} \mathbf{e}_z \quad (15b)$$

$$(\nabla^h u)_I = \frac{u_{i,j,k} - u_{i,j-1,k}}{h} \mathbf{e}_x + \frac{u_{i+1,j,k} - u_{i,j,k}}{2h} \mathbf{e}_y + \frac{u_{i,j,k+1} - u_{i,j,k-1}}{2d} \mathbf{e}_z \quad (15c)$$

$$(\nabla^h u)_I = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2h} \mathbf{e}_x + \frac{u_{i,j+1,k} - u_{i,j-1,k}}{2h} \mathbf{e}_y + \frac{u_{i,j,k+1} - u_{i,j,k-1}}{2d} \mathbf{e}_z \quad (15d)$$

Here,  $h$  is the pixel width and  $d$  is the distance between slices. Notably, for some cases the gradient notably does not depend on the function value at  $\mathbf{x}_{I,J,K}$ .

In the kinetic formulation later, each point represents a small material volume. For each of the four cases, the ‘‘point volume’’ is depicted by the shaped parallelepiped in Fig.1. Notably, for a fully surrounded point (case (d)), the point volume equals to the pixel volume,  $V^*$ . We write

$$V_I = V^* \quad (16)$$

Other cases can be readily determined.

### 2.3 Zero energy modes

Weak form formulations based on discrete gradient are prone to having spurious singular or low energy modes arising from improper approximation of the gradient. The nature of this kind of instability has been clearly delineated in [3]. The authors introduced a stabilization scheme [3],

which utilizes the so-called sub-cell gradient and eliminates the low energy modes by penalizing the difference between the point strain and the sub-cell strains. The scheme is shown to be effective in stabilizing the discrete method.

For image-based application considered here, a possible remedy is to use the sub-cell strain directly. A sub-cell is 1/8 of a regular cell containing 4-node, as the one in Fig. 1a. A regular 7-node cell contains 8 sub-cells, each having a volume of  $V^*/8$ . In the stabilized formulation, the nodal gradient at  $\mathbf{x}_{I,J,K}$  will be computed using Eqn. (15a). In this way, the nodal gradient will have different values depending in which sub-cell the point lies in, but the other aspects of the formulation remain intact. This strategy can be readily implemented. For this work, however, the simulations were conducted using the formulation (i.e. Eqn. (15)) without stabilization treatment.

### 3. APPROXIMATION OF DEFORMATION GRADIENT

Consider the finite strain deformation of a material domain. To be consistent with the convention in finite elasticity, from this point on we use  $\mathbf{x}$  to denote the current position of a material point, the dependent variable. The referential position (independent variable) is denoted by  $\mathbf{X}$ . The gradient operators are defined with respect to  $\mathbf{X}$ , e.g.  $\mathbf{R}_{IJ} = \mathbf{X}_J - \mathbf{X}_I$ . In the discrete setting, the deformation map  $\varphi: \mathbf{x} = \varphi(\mathbf{X})$  is defined by pointwise values of displacement. The discrete deformation gradient at point  $I$  is given by

$$\mathbf{F}_I^h := (\text{Grad}^h \varphi)_I = \sum_{J \in N_I} (\mathbf{x}_J - \mathbf{x}_I) \otimes \mathbf{R}_{IJ}^* \quad (17a)$$

Or, in the homogeneous form

$$\mathbf{F}_I^h = \sum_{J \in \bar{N}_I} \mathbf{x}_J \otimes \mathbf{R}_{IJ}^* \quad (17b)$$

Due the identity (8), the discrete gradient in Eq. (17) exactly preserves any homogeneous gradient, including as a special case the rigid body rotation. In fact, if  $\mathbf{x}_J - \mathbf{x}_I = \mathbf{F}^c \mathbf{R}_{IJ} \forall J \in N_I$  for a constant but otherwise arbitrary  $\mathbf{F}^c$ , we have

$$\mathbf{F}_I^h = \sum_{J \in N_I} (\mathbf{F}^c \mathbf{R}_{IJ}) \otimes \mathbf{R}_{IJ}^* = \mathbf{F}^c \left[ \sum_{J \in N_I} \mathbf{R}_{IJ} \otimes \mathbf{R}_{IJ}^* \right] = \mathbf{F}^c \quad (18)$$

This property is critical for the conserving properties of numerical integrator introduced later. In addition, the discrete gradient is properly invariant under superposed rigid body motions. Consider the motion

$$\mathbf{x}^+ = \mathbf{Q}(t)\chi(\mathbf{X}, t) + \mathbf{a}(t) \quad (19)$$

which differs from the motion  $\mathbf{x} = \chi(\mathbf{X}, t)$  by a rigid body motion. In the continuum case, the deformation gradient follows  $\mathbf{F}^+ = \mathbf{Q}\mathbf{F}$ . This relation is preserved in the discrete setting, as

$$(\mathbf{F}_I^h)^+ = \sum_{J \in N_I} (\mathbf{x}_J^+ - \mathbf{x}_I^+) \otimes \mathbf{R}_{IJ}^* = \mathbf{Q}\mathbf{F}_I^h \quad (20)$$

The implication of this property will be discussed below.

We can proceed to define other finite strains measures on the basis of (17). For example,

$$\mathbf{C}_I^h = (\mathbf{F}_I^h)^T (\mathbf{F}_I^h), \quad (21)$$

Geometric quantities like this will be used in constitutive equations. The spatial gradient of a Lagrangian function  $f = f(\mathbf{X})$  can be deduced with the aid of the chain rule,

$$(\nabla^h f)_I = (\mathbf{F}_I^h)^{-T} (Grad^h f)_I \quad (22)$$

It follows that

$$(\nabla^h f)_I = \sum_{J \in N_I} (f_J - f_I) \mathbf{r}_{IJ}^*, \text{ where } \mathbf{r}_{IJ}^* = (\mathbf{F}_I^h)^{-T} \mathbf{R}_{IJ}^* \quad (23)$$

This relation is critical in bridging the Lagrangian and Euler descriptions.

#### 4. POINT CLOUD AS A HAMILTONIAN SYSTEM

Describing a point cloud of an elastic body as a Hamiltonian system enables us to draw upon a rich set of analytical and numerical tools for analysis. The Hamilton's approach, as is well known, solves a system of two first-order ODEs (Hamiltonian system) instead of the one second-order ODE (Newtonian system). In this formulation, many properties of the system, particularly conservation laws, can be easily revealed. Analytical solutions for Hamilton's equations are exception rather than the rule. This means that there are very few integrable Hamiltonian systems, and generally one utilizes numerical methods to solve the equations. In this work, we focus on symplectic integrators. Symplectic integrators were historically developed and used in



molecular dynamics and celestial mechanics, but the application to mechanical systems is becoming popular recently [8]. The symplectic integrator preserves the sum of the projected 2-areas in the phase space [8] and this feature is believed to help the long-term stability of the numerical integration.

#### 4.1 Hamiltonian

Let the material domain be described by  $n$  points; each point represents a small material volume  $V_I$  with the understanding that the volumes are non-overlapping and  $\sum_{I=1}^n V_I \approx V$  where  $V$  is the volume of the continuum body. The position vectors in the current configuration are denoted by  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ . Let  $m_I$  be the mass of the point  $I$ , the linear momentum of this point is  $\mathbf{p}_I = m_I \dot{\mathbf{x}}_I$ . The total kinetic energy of the discrete system is

$$T = \sum_{I=1}^n \frac{p_I^2}{2m_I} \quad (24)$$

The elastic potential of the system comprises of pointwise contributions from each material volume. Let  $w$  denote the material's strain energy density (per unit reference volume) which depends on the deformation gradient, and assume that each material volume undergoes a uniform deformation given by the pointwise deformation gradient, the pointwise strain energy is

$$w_I = V_I w(\mathbf{F}_I^h) \quad (25)$$

The sum of the pointwise energy sums gives the total elastic potential

$$W = \sum_{I=1}^n V_I w_I(\mathbf{F}_I^h) \quad (26)$$

where  $\mathbf{F}_I^h$  is the point-wise deformation gradient computed according to equation (17). Upon using the discrete gradient, the elastic potential reduces to a function of  $\mathbf{x}$ . We write

$$W = W(\mathbf{x}) \quad (27)$$

At the absence of external forces, the Hamiltonian is

$$H(\mathbf{x}, \mathbf{p}) = T(\mathbf{p}) + W(\mathbf{x}) \quad (28)$$

where  $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$ . If external forces exist and are conservative, the potential of which will be added to  $H$ .

Let  $\Omega$  in  $\mathbf{R}^{2n}$  be the phase space domain endowed with the canonical symplectic structure  $\omega = \sum dx_i \wedge dp_i$ . The dynamics of the system is governed by

$$\dot{\mathbf{x}} = \frac{\partial H(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}, \quad \dot{\mathbf{p}} = -\frac{\partial H(\mathbf{x}, \mathbf{p})}{\partial \mathbf{x}} \quad (29)$$

The derivative of  $H$  to  $\mathbf{x}$  proceeds as follows. Introduce  $\mathbf{f}_I = -\frac{\partial H(\mathbf{x}, \mathbf{p})}{\partial \mathbf{x}_I}$ . By chain rule,

$\mathbf{f}_I = -\frac{\partial W}{\partial \mathbf{F}^h} \cdot \frac{\partial \mathbf{F}^h}{\partial \mathbf{x}_I}$ . The functions that contain  $\mathbf{x}_I$  are  $w_I$  and  $w_J$ ,  $J \in N_I$ . Recalling the expression

(17b), we find

$$\mathbf{f}_I = -\sum_{J \in N_I} V_J \frac{\partial w}{\partial \mathbf{F}_J^h} \mathbf{R}_{JI}^* - V_I \frac{\partial w}{\partial \mathbf{F}_I^h} \mathbf{R}_{II}^* \quad (30)$$

The derivative  $\frac{\partial w}{\partial \mathbf{F}}$  gives the first Poila-Kirchhoff stress  $\boldsymbol{\tau}$ . In term of the expanded point set  $\bar{N}_I$ , we write

$$\mathbf{f}_I = -\sum_{J \in \bar{N}_I} V_J \boldsymbol{\tau}_J \mathbf{R}_{JI}^* \quad (31)$$

The computation can be implemented in exactly the same manner as the finite element method. At the ‘‘element’’ level, the ‘‘element nodal force’’  $V_I \boldsymbol{\tau}_I \mathbf{R}_{IJ}^*$  for each ‘‘node’’ in the ‘‘point element’’ is computed. Subsequently, the element forces are assembled over all ‘‘elements’’ to yield the global force vector.

### Linear elasticity

For small strain case, the discrete strain at point  $I$  is

$$\boldsymbol{\varepsilon}_I^h = (\nabla_s^h \mathbf{u})_I = \frac{1}{2} \sum_{J \in N_I} [(\mathbf{u}_J - \mathbf{u}_I) \otimes \mathbf{R}_{IJ}^* + \mathbf{R}_{IJ}^* \otimes (\mathbf{u}_J - \mathbf{u}_I)] \quad (32)$$

where  $\mathbf{u}_I := \mathbf{x}_I - \mathbf{X}_I$  is the nodal displacement. The energy density is quadratic in  $\boldsymbol{\varepsilon}_I^h$ , written as  $w_I = \frac{1}{2} \boldsymbol{\varepsilon}_I^h \cdot \mathbf{D} \boldsymbol{\varepsilon}_I^h$ . The total elastic energy of the system is

$$W = \sum_{I=1}^n \frac{V_I}{2} \boldsymbol{\varepsilon}_I^h \cdot \mathbf{D} \boldsymbol{\varepsilon}_I^h \quad (33)$$

Again the computation follows exactly that of the finite element method. Introduce the point-wise Cauchy stress

$$\boldsymbol{\sigma}_I = \mathbf{D} \boldsymbol{\varepsilon}_I^h \quad (34)$$

A straight forward computation shows

$$\mathbf{f}_I = - \sum_{J \in \bar{N}_I} V_J \boldsymbol{\sigma}_J \mathbf{R}_{JI}^* \quad (35)$$

## 4.2 Integration Method

A variety of numerical methods can be used to integrate the Hamiltonian system (Eqn. (29)). Our priority in selecting integration method is placed on algorithmic simplicity and capability to handle large system. Since a pixel resolution point cloud normally contains a large number of points, we prefer explicit methods in order to avoid the computation of stiffness matrix. Further, we would like to use only first order methods. Based on the reported properties of various first order algorithms for Hamiltonian systems, we selected the following one-step explicit symplectic scheme [9]:

$$\mathbf{x}_{(k+1)} = \mathbf{x}_{(k)} + \Delta t \mathbf{M}^{-1} \mathbf{p}_{(k)}, \quad \mathbf{p}_{(k+1)} = \mathbf{p}_{(k)} - \Delta t \left. \frac{\partial W}{\partial \mathbf{x}} \right|_{(k+1)} \quad (36)$$

where the subscript  $(k)$  indicates the time step. Here  $\mathbf{M}$  is the diagonal mass matrix consisting of the pointwise mass. In computation the time integration amounts to conducting two explicit updates in one time step:

1. Given  $\mathbf{x}_{(k)}$  and  $\mathbf{p}_{(k)}$ , update the nodal position using Eqn. (36)<sub>1</sub>.
2. Compute  $\frac{\partial W}{\partial \mathbf{x}}$  at  $\mathbf{x} = \mathbf{x}_{(k+1)}$ , and update the linear momentum using Eqn. (36)<sub>2</sub>.

The time step has to meet the condition:

$$\Delta t = \frac{h}{c} \quad (37)$$

Where,  $h$  is the minimum of space resolution,  $c$  is a speed of sound for the considering material. For hyperelastic solids, the speed of sound is inferred from the ground state elasticity constants.

### 4.3 Conservation properties of the integrator

The integration algorithm (36), albeit very simple, has the following conservation properties:

1. Conserves the total linear momentum at the absence of external forces;
2. Conserves the total angular momentum at the absence of external forces;
3. Conserves the canonical 2-form:  $\omega = \sum dx_i \wedge dp_i$ .

**Proof:**

1. The total linear momentum  $\mathbf{P}$  in the system is  $\mathbf{P} = \sum_I^n \mathbf{p}_I$ . From (36),

$$\mathbf{P}_{(k+1)} = \mathbf{P}_{(k)} - \Delta t \sum_I^n \left. \frac{\partial W}{\partial \mathbf{x}_I} \right|_{(k+1)} \quad (38)$$

Since the elastic potential initially depends on the deformation gradient  $\mathbf{F}$  and the latter is algorithmically invariant under a rigid body translation  $\mathbf{x}_I \rightarrow \mathbf{x}_I + \mathbf{a}$  (Section 3), the total energy  $W$ , eventually a function of nodal positions, possesses the property  $W(\mathbf{x} + \mathbf{a}) = W(\mathbf{x})$ . Here  $\mathbf{x} + \mathbf{a}$  means applying the translation  $\mathbf{a}$  to every members of  $\mathbf{x}$ . Taking the derivative of  $W(\mathbf{x} + \mathbf{a})$  with respect to  $\mathbf{a}$  and evaluating the derivative at  $\mathbf{a} = \mathbf{0}$ , we find

$$\mathbf{0} = \left. \frac{\partial W(\mathbf{x} + \mathbf{a})}{\partial \mathbf{a}} \right|_{\mathbf{a}=\mathbf{0}} = \sum_I^n \frac{\partial W}{\partial \mathbf{x}_I} \quad (39)$$

Upon recalling (38), Eqn (39) concludes  $\mathbf{P}_{(k+1)} = \mathbf{P}_{(k)}$ .

2. The total angular momentum is  $\mathbf{L} = \sum_I^n \mathbf{x}_I \times \mathbf{p}_I$ . From (38), after a straight forward computation,

$$\mathbf{L}_{(k+1)} = \mathbf{L}_{(k)} - \Delta t \left( \sum_I^n \mathbf{x}_I \times \frac{\partial W}{\partial \mathbf{x}_I} \right) \Big|_{(k+1)} \quad (40)$$

The energy density  $w$  satisfies the standard invariance requirement  $w(\mathbf{QF}) = w(\mathbf{F})$  for any rotation tensor  $\mathbf{Q}$ . This condition is preserved numerically since the discrete deformation gradient satisfies the invariance property (20). Together, they indicate that  $W(\mathbf{Qx}) = W(\mathbf{x})$  for

any rotation tensor  $\mathbf{Q}$ . Consider a one-parameter family of rotation tensor  $\mathbf{Q}(\varepsilon)$  with  $\mathbf{Q}(0) = \mathbf{I}$  and let  $\boldsymbol{\Omega} = \frac{d\mathbf{Q}}{d\varepsilon} \mathbf{Q}^T$  be the (skew-symmetric) spin tensor. Taking the derivative of  $W(\mathbf{Q}(\varepsilon)\mathbf{x})$  with respect to with respect to  $\varepsilon$  and evaluating the derivative at  $\varepsilon = 0$ ,

$$0 = \left. \frac{\partial W(\mathbf{Q}(\varepsilon)\mathbf{x})}{\partial \varepsilon} \right|_{\varepsilon=0} = \sum_I^n \frac{\partial W}{\partial \mathbf{x}_I} \cdot (\boldsymbol{\Omega} \mathbf{x}_I) = \boldsymbol{\omega} \cdot \left( \sum_I^n \mathbf{x}_I \times \frac{\partial W}{\partial \mathbf{x}_I} \right) \quad (41)$$

where  $\boldsymbol{\omega}$  is the axial vector of  $\boldsymbol{\Omega}$ . Since the above equation holds for an arbitrary vector  $\boldsymbol{\omega}$ , we proved that  $\sum_I^n \mathbf{x}_I \times \frac{\partial W}{\partial \mathbf{x}_I} = 0$ , which in turn concludes  $\mathbf{L}_{(k+1)} = \mathbf{L}_{(k)}$ .

3. The fact that (36) is symplectic has been established in [9].

This explicit symplectic integrator is known has a good long-time stability [9].

#### 4.4 Extension to non-Hamiltonian system

Although the integration method is developed in the context of Hamiltonian systems, it is straightforward to extend the method to systems containing non-conservative forces, in particular damping. Consider the case of linear damping for which the pointwise damping force is give by  $\mathbf{f}_I = -c_I \mathbf{v}_I = -(c_I / m_I) \mathbf{p}_I$ . We can also write the damping force globally as

$$\mathbf{f} = -\mathbf{C}\mathbf{M}^{-1}\mathbf{p} \quad (42)$$

At the presence of this force, Eqn (36)<sub>2</sub> is modified as

$$\mathbf{p}_{(k+1)} = \mathbf{p}_{(k)} - \Delta t \left. \frac{\partial W}{\partial \mathbf{x}} \right|_{(k+1)} - \Delta t \mathbf{C}\mathbf{M}^{-1} \mathbf{p}_{(k+1)} \quad (43)$$

Thus,

$$\mathbf{p}_{(k+1)} = \left[ \mathbf{I} + \Delta t \mathbf{C}\mathbf{M}^{-1} \right]^{-1} \left( \mathbf{p}_{(k)} - \Delta t \left. \frac{\partial W}{\partial \mathbf{x}} \right|_{(k+1)} \right) \quad (44)$$

The algorithm becomes semi-implicit. The update for the position vector  $\mathbf{x}$  remains explicit but the solution for the momentum becomes implicit. Nevertheless, the computation remains simple as both  $\mathbf{C}$  and  $\mathbf{M}$  are diagonal. A numerical example of damped system is provided in Section 6.2.

## 5. DERIVATION OF POINT CLOUD FOR IMAGES

We employed a simple gray-scale based binary segmentation method to extract points that are believed to belong to the interested material domain. The images were processed using the automated binary segmenting/morphologic techniques integrated in MATLAB [10]. Using these functions the images were transformed to binary analogues, where “1” – means tissue cell, “0” – nothing (Fig. 2). Each point represents the small volume of material in the corresponding pixel (or voxel in 3D). The set of segmented points forms a point cloud representation of the material domain of interest. An examples of segmentation is presented in Fig. 2.

### 5.1 Point connectivity

For a given point, the neighbors are the adjacent points shown in Fig. 1. These are the points that energetically interacting with the given point. A simple search algorithm is used to identify the neighbors. The cross-layer neighbors “ $i,j,k-1$ ”, “ $i,j,k$ ” and “ $i,j,k+1$ ” are detected first followed by the same-layer neighbors. A total of four topological cases can be identified in this manner. The morphological operations of binary images “**spur**” and “**fill**” exclude other singular records. The physical spacing of between points are computed from the image resolution.

### 5.2 Discrete surface

Despite the discrete nature of the model, we still need to construct the boundary surface for the purpose of applying traction boundary tractions (pressures for example). The boundary surface is formulated using polygons connecting the boundary nodes on two neighboring layers, ( $k, k+1$ ). A scheme was implemented to produce triangle, quadrilateral and mixed polygons (Fig. 3 c, d) depending on the relative position of the opposite nodes. Polygons are recorded as a counter-clockwise sequence of vertexes. This method produces a pixel-resolution boundary surface. The surface is non-smooth and is not suited for contact analysis, but suffices for applying pressure load.

## 6. NUMERICAL RESULTS

### 6.1 Evaluation of MLS approximation

**6.1.1 Patch test.** The least square formulation can identically pass the patch test on a uniform point cloud. To illustrate that, we considered a straight rectangle elastic beam of dimensions  $600 \times 50 \times 100$  (Fig. 4a). The beam is linear elastic with  $E = 1000$ ,  $\nu = 0.3$ , which is subjected to a uniform axial traction of  $s_3 = 0.1$  on one end, and the other end is constrained against the axial displacement. Other constraints just enough to eliminates lateral rigid body motion and axial rotation are applied. Static equilibrium is governed by the equation  $\frac{\partial W}{\partial \mathbf{x}} = \frac{\partial \Pi^{ext}}{\partial \mathbf{x}}$  where  $\Pi^{ext}$  is the potential of the external force. For linear elasticity the equation reduces to a linear system of equations of nodal displacements. The axial stress is found to be uniform and exactly equals to 0.1. The stress contour is plotted in Fig. 4b.

**6.1.2 Accuracy and convergence.** We compared the performance of the standard 7-node cell with two variations that include some diagonal neighbors. The physical problem, material parameters, and results are presented on Table 1. Although not plotted, all three schemes exhibit roughly the same rate of convergence. The 15-node cell appears to have the best accuracy; however, considering the computation cost (the 7-node cell and its topological variants are least expensive because closed-form formulae are available), the 7-node cell provides a reasonable trade-off between accuracy and efficiency. If the accuracy is of primary concern, the 15-node cell is recommended.

## 6.2 Cantilever beam

As a testing problem for dynamic analysis we consider the deformation of 3D rubber cantilever beam (Fig. 5) of sizes  $600 \times 50 \times 100$  mm made of neo-Hookean hyperelastic material for which the energy function is given in Eqn. (45).

$$W = 0.5G(I - 2\log(\det \mathbf{F}) - 3) + 0.5K(\log(\det \mathbf{F}) - 1)^2 \quad (45)$$

Here  $I$  is a first invariant of deformation tensor  $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ . The material parameters are  $G=3.35E6$  Pa,  $K = 1.12E8$  Pa. The beam is loaded by impulse tip force  $F$  (Fig. 6c), and fixed on the other end,

$$u_x = u_y = u_z = 0 \quad \text{at} \quad x = 0 \quad (46)$$

The beam point-cloud model contains 165 points. The integration time step  $\Delta t = 1e-5$  s. The transient response of a tip node A simulated by FEM code (FEAP) and the explicit discrete point

cloud method (EDPCM) in time interval  $s$  plotted in Fig. 6. The results compare reasonably well with the finite element prediction. Figs. 7 and 8 show the time-history of the beam's total energy and motion of node A in phase surface, correspondently. The observable jitter in the total energy is typical of symplectic integration, but the mean energy is notably stable over the integration period ( $2 \times 10^6$  steps). The phase-plane trajectory in Fig. 8 also indicates that the numerical response is asymptotically stable.

Fig. 9 shows the response at the presence of viscous force with a pointwise damping coefficients of  $c = 0.02$ . The response was computed using formulae (36)<sub>1</sub> and (44). The solution was compared with FEM results and a good agreement was found.

### **6.3 An artificial aortic heart valve**

An image based point cloud model organization technology was used to simulate dynamics of artificial aortic heart valve (Fig.10). An isotropic nonlinear neo-Hookean hyperelastic model (45) had been used with material property:  $G=0.93E6$  Pa,  $K=9.06E6$  Pa. These properties are inferred from the material model used in [11]. The pressure difference of acting on both side of valve is applied on the inner surface of valve (Fig. 11). The simulation was performed over the time interval  $t \in [0, 0.15]$  sec, corresponding to open phase of cardio circle. The point-model consists of 161574 points. A small time step,  $\Delta t = 1e-7$  sec, was used. The results of EDPCM, implicit FEM (NASTRAN) are compared (Fig. 12, 13). By a visual inspection, the displacement predictions compare very well. The stress contours have different color schemes, but a close look at the values indicates that the two sets of results are in reasonably good agreement. For example, in the fully open position the stress in the belly region in both models falls into the range of  $7.0E3 - 6.0E4$  Pa. The patterns of distribution are also similar to each other.

## **7. CONCLUDING REMARKS**

We introduced a point cloud method for image-based dynamic analysis in biological systems. The method takes as the geometric input a point cloud derived from medical images, and describes the cloud as a discrete Hamiltonian system. Information necessary to formulate the discrete system is readily available from the images. Continuum constitutive equations normally used in FEM analysis are directly applied. We suggested an explicit integration scheme using a



symplectic momentum conserving integrator. This integrator, being explicit, can readily handle large systems and appears to have a good long-term stability.

The proposed method has the following attributes:

- The entire process can be carried out with minimal user interference, or even fully automatically;
- The formulation is very simple and can be easily implemented. It bypasses continuum approximation of the displacement field and thus greatly simplifies the computation.
- Low computer memory requirement. Medium to relatively large problem (500K-1M degree-of-freedoms) can be readily solved in common desktop computers.

A major limitation of the method is the CPU time. The discrete model generally contains a large number degree of freedoms ( $>10^6$ ) and the time step is typically very small. This leads to long CPU times. Nevertheless, the computation structure is inherently parallel and a significant speed-up is expected in a parallel platform.

**ACKNOWLEDGEMENTS.** This work was conducted while the first author (A.I.N.) was visiting the Department of Biomedical Engineering, the University of Iowa. The visit was funded by the Fulbright Scholar Program, and this support is gratefully acknowledged. A.I.N. also thanks the BME Department of the University of Iowa for providing him with office and computing supports.

## REFERENCES

1. J. Lu, J. Qian, and W. Han. Discrete gradient method in solid mechanics. *International Journal for Numerical Methods in Engineering*, 74:619–641, 2008
2. J. Lu and J. Qian, Discrete gradient method over polygon mesh. *International Journal for Numerical Methods in Engineering*, 78:505-527, 2009
3. J. Qian and J. Lu, A stabilization formulation for discrete gradient method. *Communications in Numerical Methods in Engineering*, DOI: 10.1002/cnm.1337.

4. J. Qian and J. Lu, Point-cloud method for image-based biomechanical stress analysis. *International Journal for Numerical Methods in Biomedical Engineering*, in review.
5. T. Belytschko, Y. Krongauz, D. Organ D, M. Fleming, and P. Krysl, Meshless methods: An overview and recent developments, *Comput. Methods Appl. Mech. Eng.* 139, 3–48, 1996.
6. S. Li and W. K. Liu, Meshfree and particle methods and their applications. *Appl. Mech. Rev.* 55, 1-34, 2002.
7. V. I. Arnold: *Mathematical Methods of Classical Mechanics*, Springer, New York, 2nd edition, 1988
8. J. E. Marsden, G. W. Patrick, and W. F. Shadwick (eds.), *Integration algorithms and classical mechanics*, American Mathematical Society, Providence, RI, 1996
9. R. D. Skeel, G. Zhang and T. Schlick, A family of symplectic integrators: stability accuracy, and molecular dynamics applications, *SIAM J. Sci. Comput.* 18, no. 1, 203-222, 1997
10. MATLAB 7. The MathWorks Web Site Resources, 2009
11. R. M. Haj-Ali, L. P. Dasi, H. S. Kim, J. Choi, and H. W. Leo, and A. P. Yoganathan, Structural simulations of prosthetic tri-leaflet aortic heart valves. *J. Biomechanics*, 41: 1510-1519, 2008

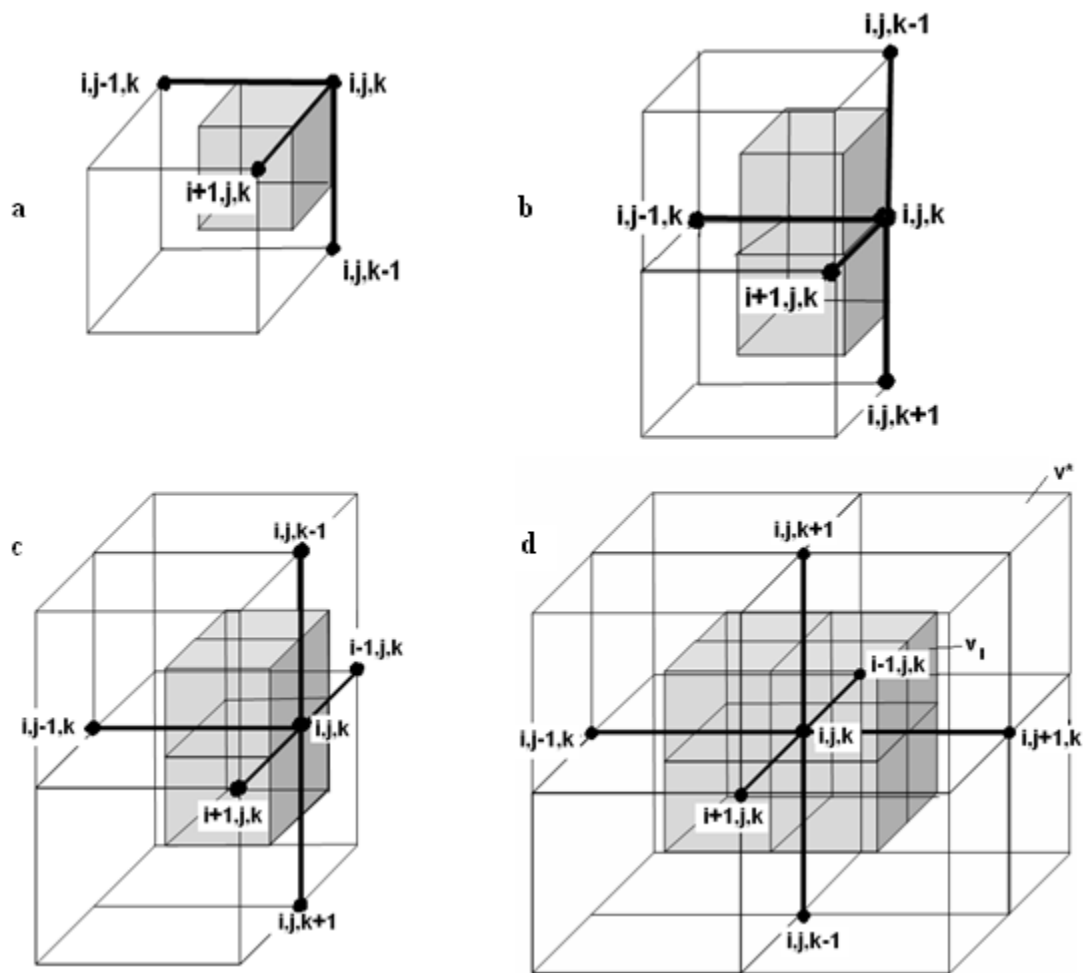


Fig.1. Schematics of neighbor sets and point volume for four topological cases

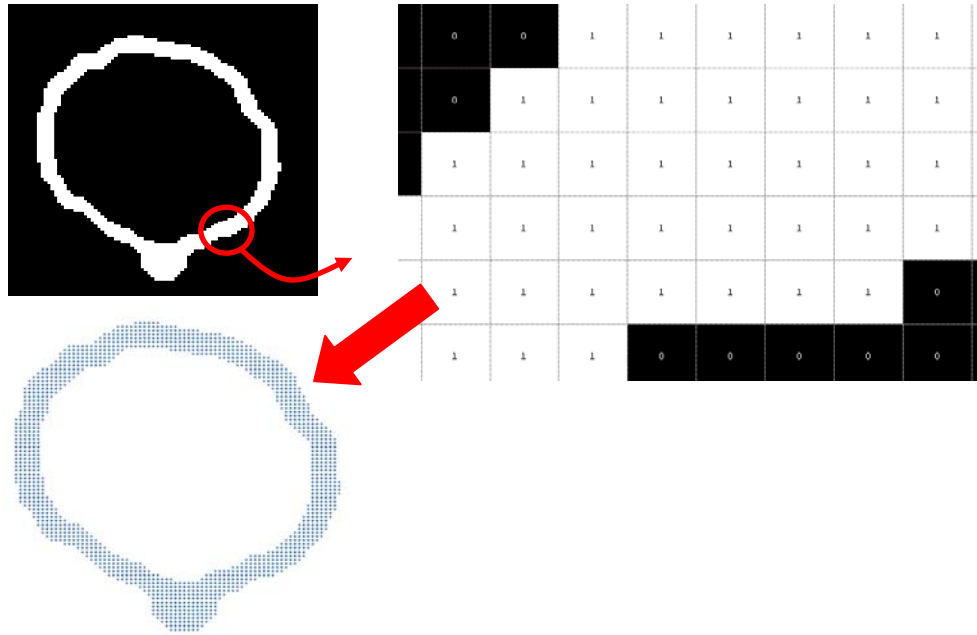


Fig. 2. Illustration of image segmentation

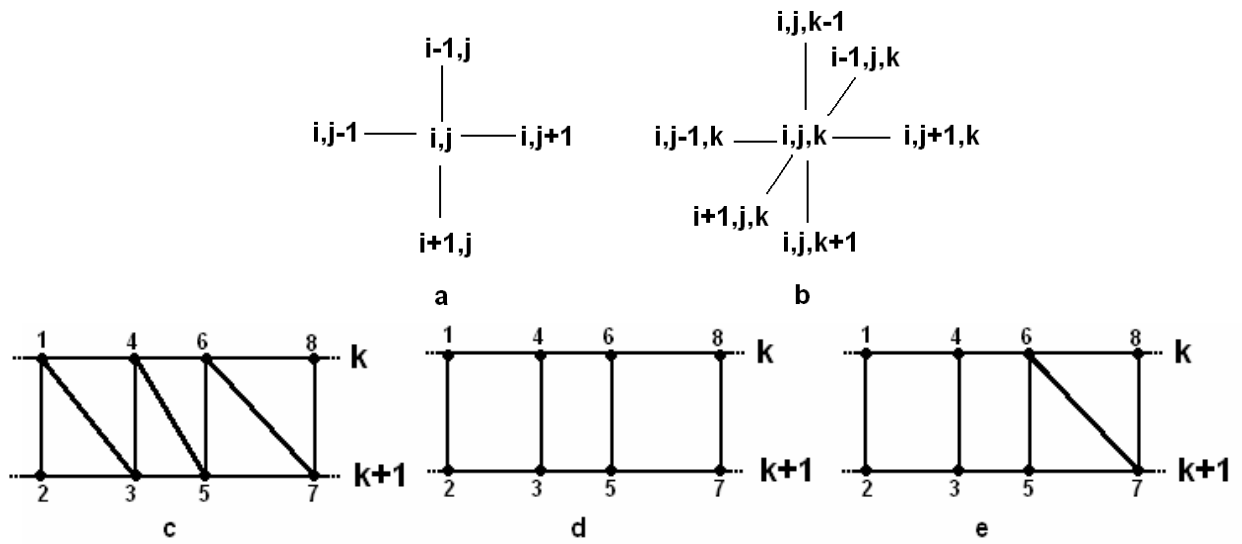


Fig. 3. Connectivity in point-cloud model for **a** – 2D, **b** -3D case.  
 Surface polygons: **c** - triangle, **d**- quadrilateral, **e**- mixed

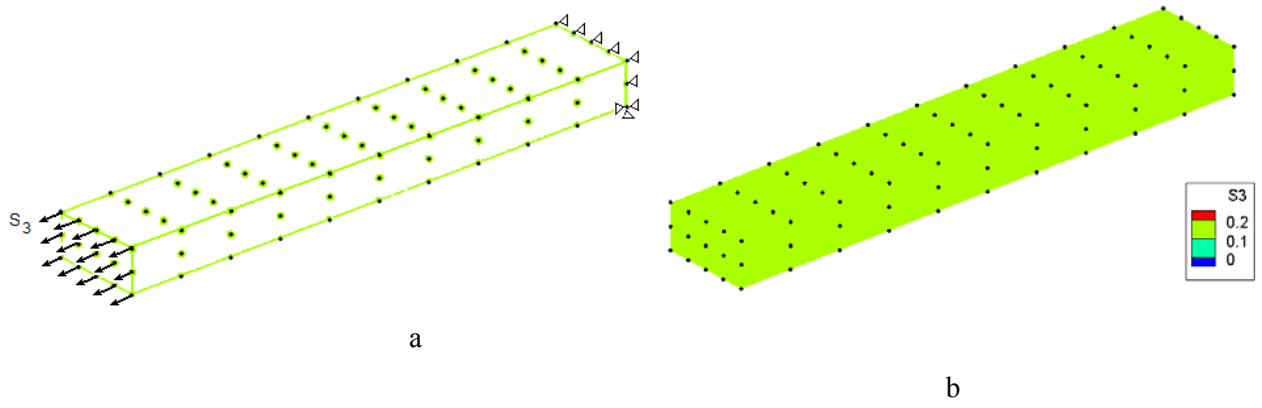


Fig.4. Patch test

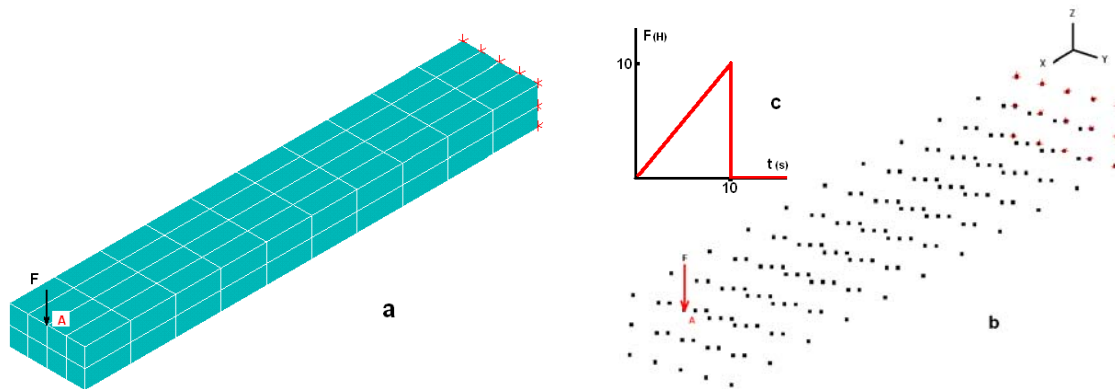


Fig. 5. 3D finite element (a) and point cloud (b) models of a cantilever beam loaded by a tip force

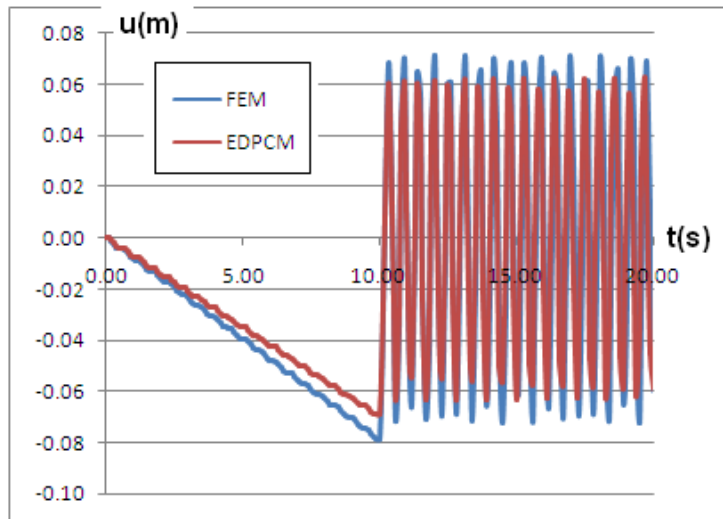


Fig. 6. The displacement of point A for time  $t=[0, 20]$ sec simulated by FEM and EDPCM



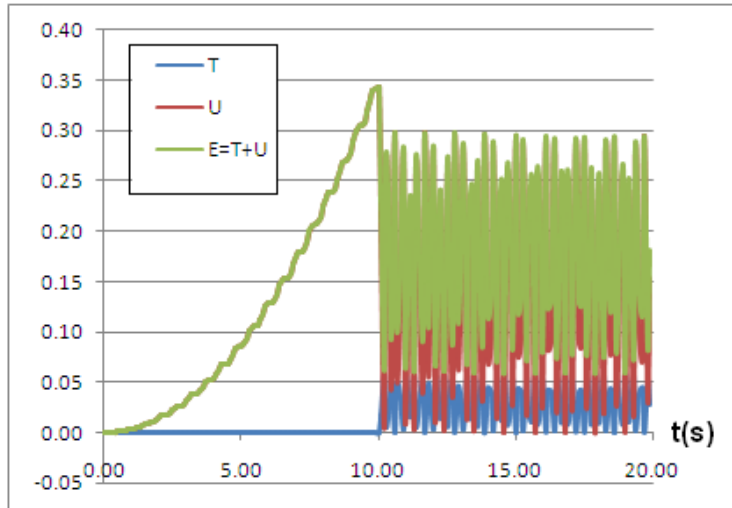


Fig. 7. Energy changes in time interval  $t=[0, 20]$  s;  
T – Kinetic energy, U- Potential energy, E- Total energy

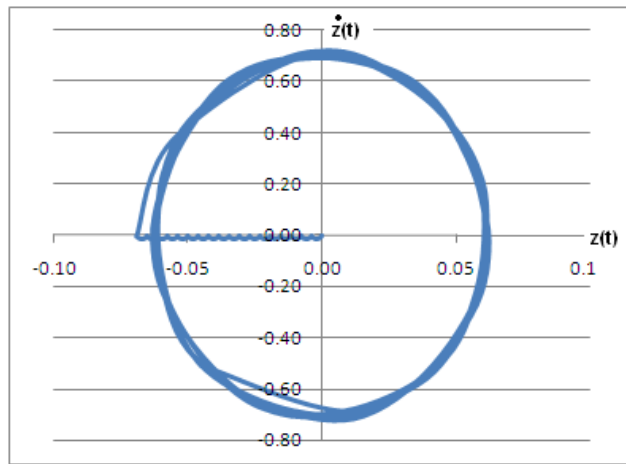


Fig. 8. The X-Z plane motion of point A, phase space trajectory

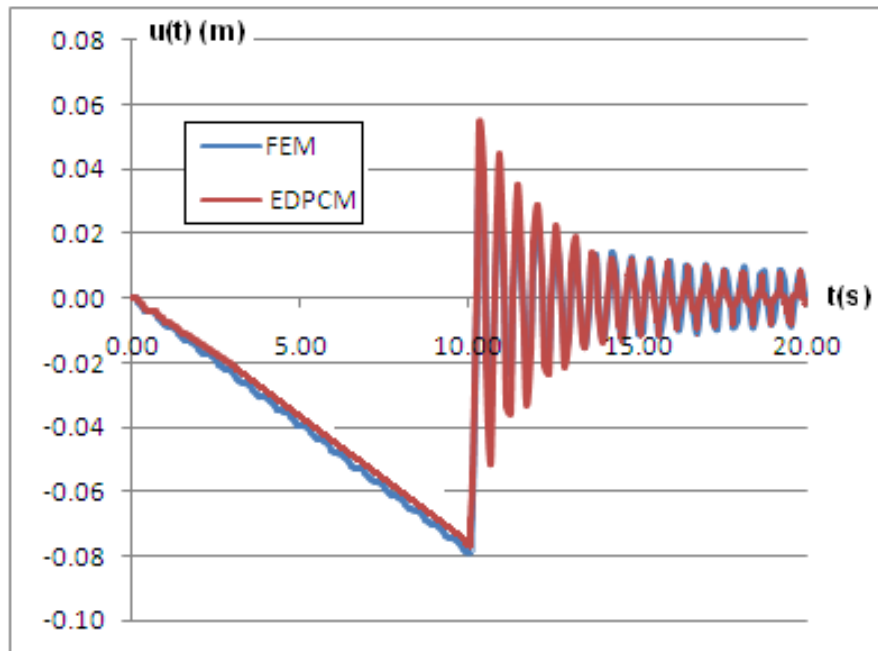


Fig.9. The displacement of point A for time  $t=[0, 20]$ sec simulated by FEM and EDPCM for Non-Hamiltonian system with damping ratio  $c = 0.02$

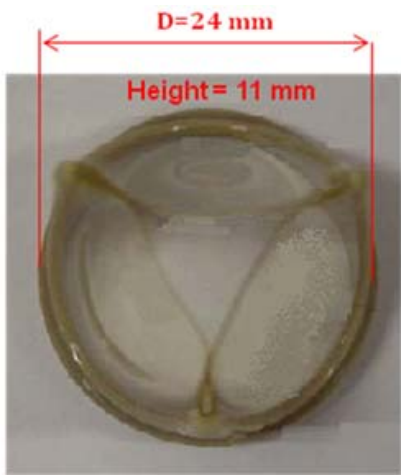


Fig. 10. Tri-leaflet polymeric aortic heart valve

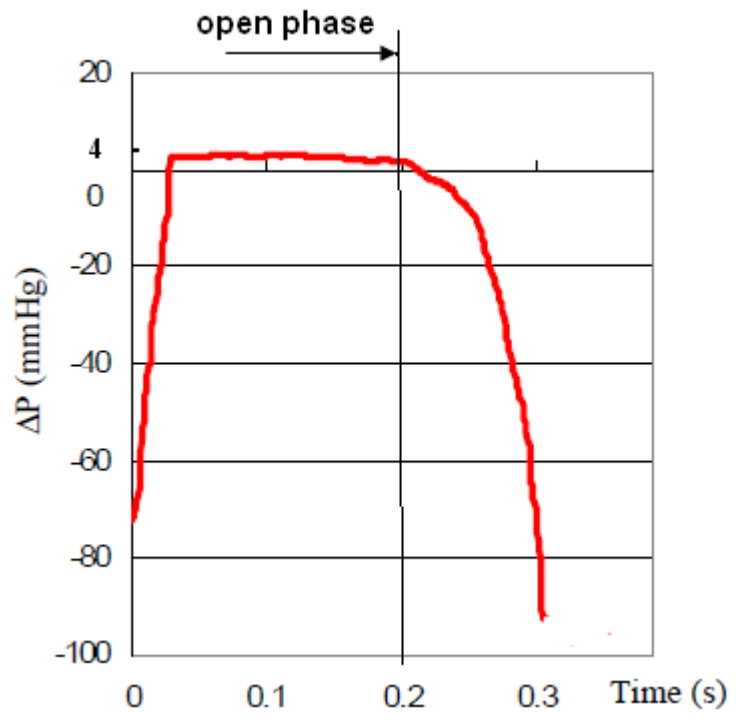


Fig. 11. Pressure profile

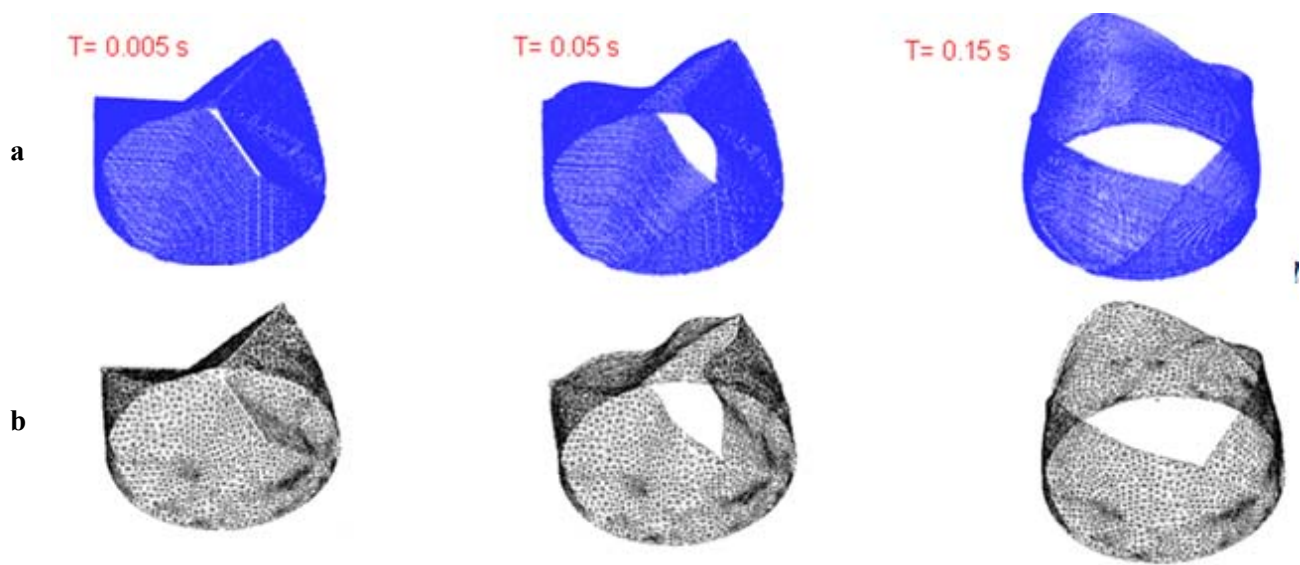


Fig.12. Deformation of an aortic artificial heart valve. Simulation performed by **a**- EDPCM , **b** - FEM (NASTRAN). The configuration at  $t=0.15\text{ s}$  corresponds to the fully opened position.

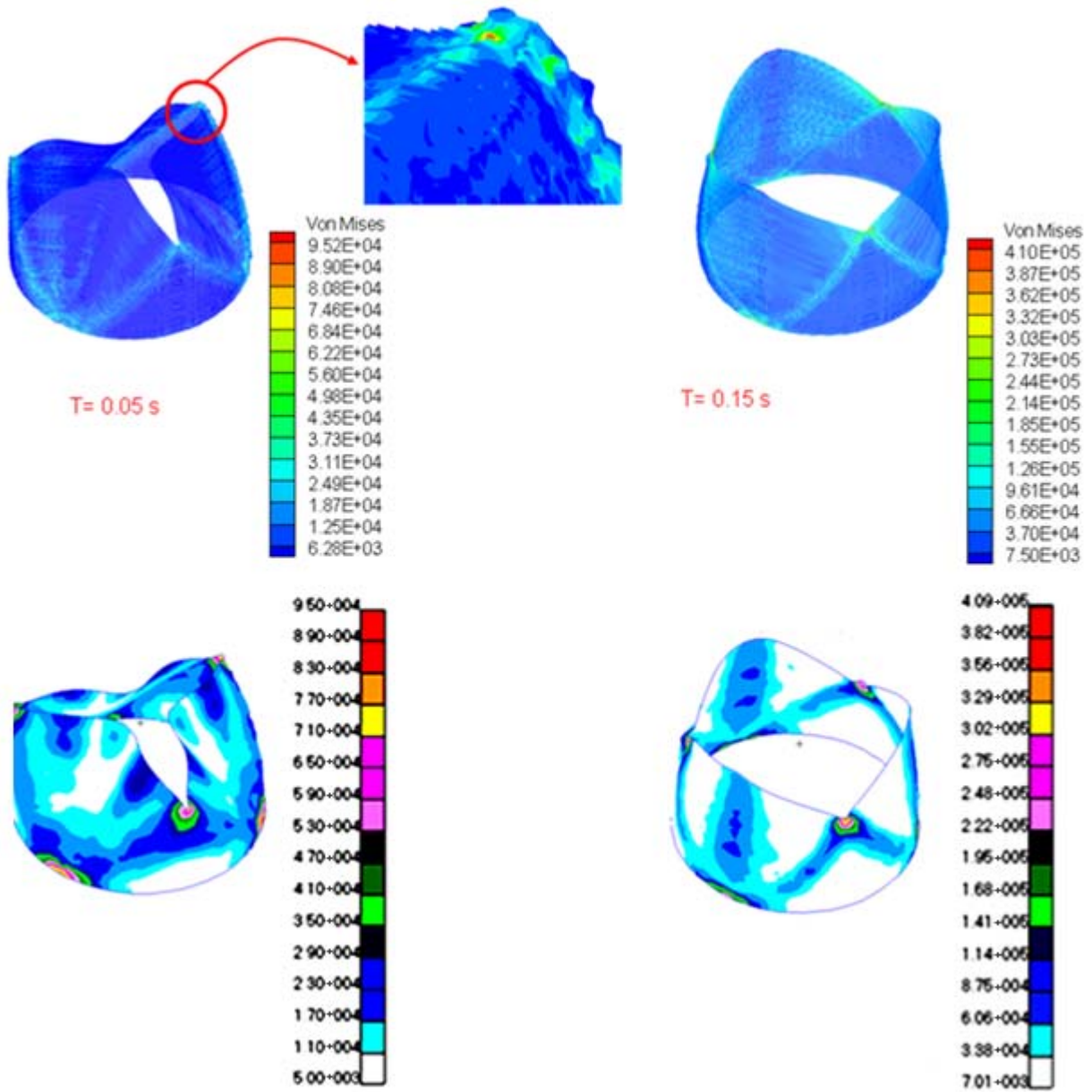


Fig. 13. Von Mises stress. Upper row: EDPCM; Lower row: FEM (NASTRAN)

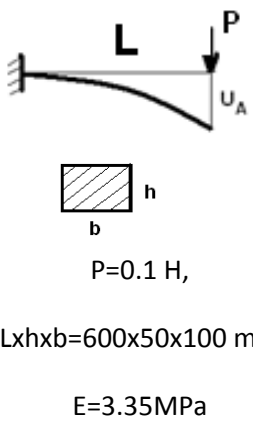
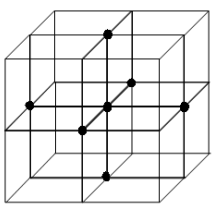
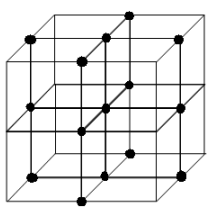
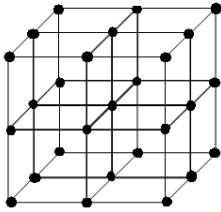
	Deflection $U_A$ (mm)		
	7 node	15 node	27 node
<b>Number of nodes in directions of length (L) x thickness (h) x width (b)</b>			
21x3x5=315	1.29	2.49	4.74
21x5x5=525	1.72	2.16	2.52
21x7x5=735	1.84	2.09	2.24
<b>Analytical <math>U_A=PL^3/3EI</math></b>	2.06		

Table 1. Accuracy and convergence rate for various neighbor sets