



## Real-time deformation of human soft tissues: A radial basis meshless 3D model based on Marquardt's algorithm

Jianyong Zhou<sup>a,1,\*</sup>, Zu Luo<sup>b,1</sup>, Chunquan Li<sup>b</sup>, Mi Deng<sup>c</sup>

<sup>a</sup> School of Mechatronic Engineering, Nanchang University, Jiangxi, China

<sup>b</sup> School of Information Engineering, Nanchang University, Jiangxi, China

<sup>c</sup> Department of Mathematical Sciences, New Mexico State University, Las Cruces, NM, 88003, USA



### ARTICLE INFO

#### Article history:

Received 21 May 2017

Revised 18 August 2017

Accepted 12 September 2017

#### Keywords:

Meshless method

Marquardt algorithm

Surface fitting

Virtual surgery

Real-time deformation

### ABSTRACT

**Background:** When the meshless method is used to establish the mathematical-mechanical model of human soft tissues, it is necessary to define the space occupied by human tissues as the problem domain and the boundary of the domain as the surface of those tissues. Nodes should be distributed in both the problem domain and on the boundaries. Under external force, the displacement of the node is computed by the meshless method to represent the deformation of biological soft tissues. However, computation by the meshless method consumes too much time, which will affect the simulation of real-time deformation of human tissues in virtual surgery.

**Methods:** In this article, the Marquardt's Algorithm is proposed to fit the nodal displacement at the problem domain's boundary and obtain the relationship between surface deformation and force. When different external forces are applied, the deformation of soft tissues can be quickly obtained based on this relationship.

**Results and conclusions:** The analysis and discussion show that the improved model equations with Marquardt's Algorithm not only can simulate the deformation in real-time but also preserve the authenticity of the deformation model's physical properties.

© 2017 Elsevier B.V. All rights reserved.

## 1. Introduction

At present, virtual surgery simulation has become a hot research topic in domestic and international industry, and human soft tissue modeling is one of the important research fields. The deformation of human tissues under external force can be accurately obtained through models, which contributes to the real-time interaction between a human being and model [1–5]. The following three main models are commonly used for simulating human tissues: Mass Spring Model (MSM), Finite Element Model (FEM) and Meshless Model (MM).

MSM assumes that biological tissue can be approximately represented by a system consisting of masses and springs that connect the masses. The mechanism of the system is described by its Lagrangian [6–12] (conventional mass spring model). Waters [13] pioneered a mass spring model with a regular mesh to simulate the deformation of a human face, and abstracted the facial tissues into nonlinear springs perpendicular to each other. The mass

spring model is suitable for simulating the small deformations of facial muscle tissues. Bona et al. [14] used a spring oscillator model based on the properties of linear viscoelastic materials to model brain tissues to simulate the deformation process of brain tissues during intracranial hemorrhage and tumor growth. Bourguignon and Cani proposed to add angular springs to the conventional mass spring model to describe the anisotropy of soft tissues, such as the state expressed during muscle rotation [15–17]. MSM has the advantage that the computation of the resulting model system equation is fast. However, the iterative calculations are not very robust and the model is subject to different constraints and external forces; to guarantee accuracy and correctness, the modeler often needs to manually set the system parameters (e.g. mass quality, elasticity coefficient, damping coefficient, initial velocity of mass and position of mass) by virtue of experience [18–20].

Finite element model disperses a tissue into several sub-units, and binds them into a combination through the nodes of their boundaries. The assumed approximation function in each unit is used to express the to-be-determined dependent variable in the solution domain, and an algebraic equation for solving the dependent variable is established by using the variational principle [21–23]. Cotin used a linearly interpolated tetrahedral finite element model to simulate the deformation of liver, carried out col-

\* Corresponding author.

E-mail addresses: [624235362@QQ.com](mailto:624235362@QQ.com) (J. Zhou), [waveluozu@163.com](mailto:waveluozu@163.com) (Z. Luo), [lichunquan@ncu.edu.cn](mailto:lichunquan@ncu.edu.cn) (C. Li), [dengmi@nmsu.edu](mailto:dengmi@nmsu.edu) (M. Deng).

<sup>1</sup> Jianyong Zhou and Zu Luo contribute equally to this work.

lision detection on the model, and studied the force feedback [24]. The simulated results closely matched the force feedback obtained from touching the true liver. For the material land structural properties of different soft tissues, some researchers used nonlinear viscoelastic or non-tetrahedral finite element models. For example, Zhu Qiling used the biphasic proviscoelastic (BPVE) model to abstract brain tissues as a hyper-viscoelastic body composed of incompressible viscoelastic solids and incompressible fluid [25]. The brain tissue model established by such method can be used to analyze the mechanism that causes intracranial brain tissue injury in case of head collision. Pieper [26] used the prism element finite element model to simulate the elastic deformation of fat, so the graphic interaction between a trainee and plastic fat tissues can be obtained in virtual plastic surgery. As FEM is a continuous parametric model, it is highly precise and adaptable and can simulate surfaces of various shapes and material properties. Compared with the spring oscillator model, FEM has higher complexity and more computations, and it is difficult to create a real-time simulation. In order to simplify the computation of FEM, Sascha [27] used the Fast Finite Element Model (FFEM) to achieve neck surgical navigation. Increasing the computational power through parallel computing can also effectively improve the force feedback refresh rate.

However, FEM is not without its problems. Extremely large deformations of soft tissues - distortions of the mesh cells often are difficult to calculate or have no solution. In these cases, it is often necessary to constantly re-divide the finite meshes in the finite element computation, which greatly increases the computation time and thus can no longer function as a real-time simulation of soft tissue deformation. To avoid these difficulties associated with pre-defined meshes, a meshless method has been proposed and extensively researched in recent years.

In the meshless method, soft tissue is regarded as the problem domain that has a large number of discrete nodes distributed within. A stiffness matrix is established by connecting nodes with a shape function, which approximately models the mechanical properties of soft tissues. Any node displacements obtained can be used to represent the soft tissue deformation [28,29]. Compared to FEM, the meshless methods more able to simulate deformation in real-time. It does not need to pre-process any cell meshing data in the element domain (simplifying the rendering of 3D visuals), has neither mesh distortion nor entanglement when tissue deformation is analyzed, and can analyze soft tissue deformation's uncertain or movable boundary problems without re-dividing the mesh in a new solution domain (rather than find new boundary positions for each deformation).

Since the mesh is not affected in the above process, Meshless Model (MM) can simulate large deformations of soft tissue. MM are classified by the type of shape function used: Moving least square method (MLS) shape functions, integral form shape functions, and point interpolation method (PIM) shape functions.

Applying the MLS method, Xu Shaojian [30] introduced a viscoelastic model of the liver to make it closer to bio-mechanical properties of liver soft tissue and performed collision detection experiments on the model. According to the results, the meshless method is mathematically simple, presents a good sense of deformation, and represents a significant advance in the modeling of soft tissue deformation during virtual surgery. Zhu and Yu established a mathematical model of liver and stomach surface by using a modification of MLS, Element Free Galerkin (EFG) [31]. EFG eliminates FEM's greatest source of inaccuracy: the inability to analyze large deformations. However, MM methods require careful choice of underlying mathematical model in simulating human soft tissue as equations that do not fit the boundary conditions usually have no solution [32,33].

The most representative integral form shape function is smooth particle flow (SPH) [34], an analog simulation of bleeding. Kulp

et al. [35] established a model of blood flow by modeling the velocity-flow-pressure change within one cardiac cycle, which will see potential application in clinical practice.

PIM was used by Liu et al. [36,37] to analyze the mechanical stress and strain of materials. Unlike MLS, PIM utilizes Kronecker-like penalty functions; when boundary conditions are introduced by penalty methods, the complexity of the calculations more resembles that of FEM rather than that of other meshless models. There are two types of PIMs, Polynomial Point Interpolation Method (PPIM) and Radial Point Interpolation Method (RPIM). In PPIM models, nodes must be regularly distributed and their number is strictly limited. Otherwise, the function becomes "poorly behaved" and produces a singular stiffness matrix, which cannot generate the approximation function that models mechanical properties. By contrast, RPIM models do not as strictly require regular distribution or limited nodes. Thus, RPIM is more adaptable, and the computed values are more stable. In this paper, we use global meshless method with RPIM to calculate nodal displacements because it is more simple and stable than other meshless methods in calculation.

MM can also simulate irregular deformations of soft tissue. The modeling function for such deformation is solved by obtaining the displacements for each node. However, the computation is still rather time-consuming and cannot simulate real-time deformation during virtual surgery. To address these difficulties, we propose the use of Marquardt's algorithm to fit a mathematical relation between the force on the virtual reality tissues and displacements of each node in advance. When soft tissues are put under stress in virtual surgery, their deformation can be displayed immediately by calling this fitting relation. In order to prove this method's accuracy and feasibility, the following experiment was performed. Several spatial forces (0-1 N) were applied to a 48 mm × 48 mm × 6 mm geometric model with the elastic modulus of human liver. The displacement of all nodes corresponding to a given spatial force was calculated by global meshless method. By Marquardt's algorithm, the fitting relation between force and deformation is determined, which also generates the fitting surface of each nodal displacement to display the total deformation of the model. Experimental results show that average deformation computation takes 0.1509 s and the maximum deformation error is less than 0.5 mm. An analog simulation on human liver was then performed to demonstrate the real-time feasibility in virtual surgery.

## 2. Relevant work

### 2.1. Basic theory

$$\text{Solid mechanics equation } \mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b} = 0. \text{ in } \Omega. \quad (1)$$

$$\text{Natural boundary condition } \boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{f}}. \text{ on } \Gamma_t. \quad (2)$$

$$\text{Essential boundary condition } \mathbf{u} = \bar{\mathbf{u}}. \text{ on } \Gamma_u. \quad (3)$$

In (1), the stress  $\boldsymbol{\sigma}$ , caused by body force  $\mathbf{b}$  (e.g. force of gravity), is distributed in object. In (2), the stress  $\boldsymbol{\sigma}$  is on the natural boundary  $\Gamma_t$ ,  $\mathbf{n}$  is the unit normal vector and  $\bar{\mathbf{f}}$  is the external force. In (3),  $\mathbf{u}$  denotes displacement and the known displacement  $\bar{\mathbf{u}}$  appears on essential boundary  $\Gamma_u$  when external force  $\bar{\mathbf{f}}$  and body force  $\mathbf{b}$  act on the object (see Fig. 1 and Fig. 2).  $\mathbf{L}$  is the differential operator as follow :

$$\mathbf{L} = \begin{bmatrix} \partial/\partial x & 0 & 0 & \partial/\partial z & \partial/\partial y \\ 0 & \partial/\partial y & 0 & \partial/\partial z & 0 & \partial/\partial x \\ 0 & 0 & \partial/\partial z & \partial/\partial y & \partial/\partial x & 0 \end{bmatrix}^T \quad (4)$$

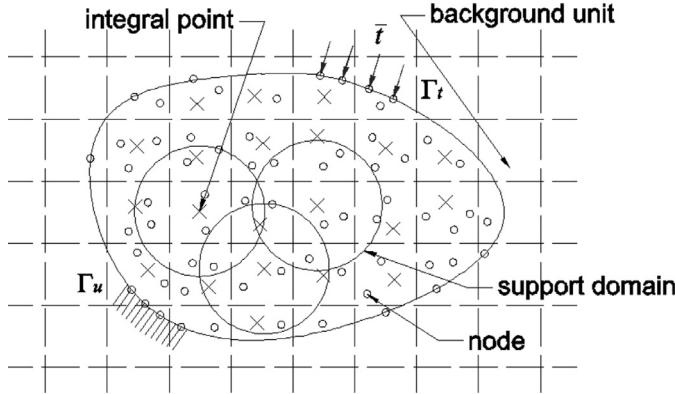


Fig. 1. 2-D meshless method.

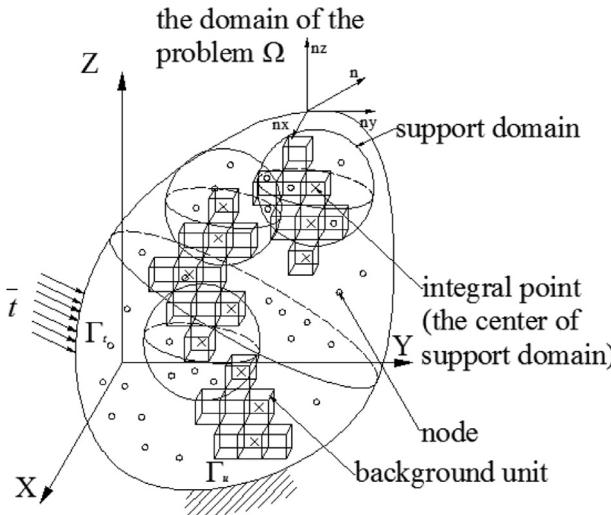


Fig. 2. 3-D meshless method.

We can get the expression of strain  $\boldsymbol{\epsilon}$  as  $\boldsymbol{\epsilon} = \mathbf{L}\mathbf{u}$ .

According to solid mechanics, the potential energy of elastic object is expressed as

$$P = P_s - W_f. \quad (5)$$

Where  $P$  is potential energy and  $P_s = \frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega$  is strain energy.  $W_f = \int_{\Omega} \mathbf{u}^T \mathbf{b} d\Omega + \int_{\Gamma_t} \mathbf{u}^T \mathbf{t} d\Gamma$  is the work done by external force  $\mathbf{t}$  and body force  $\mathbf{b}$ .

Eq. (5) is rewritten as

$$P = \frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \mathbf{u}^T \mathbf{t} d\Gamma. \quad (6)$$

In (6), stress  $\boldsymbol{\sigma}$  is obtained by  $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} = \mathbf{D}\mathbf{L}\mathbf{u}$ .  $\mathbf{D}$  denotes the elastic coefficient matrix as follow:

$$\mathbf{D} = \begin{bmatrix} D_{11} & D_{12} & D_{12} & 0 & 0 & 0 \\ D_{12} & D_{22} & D_{12} & 0 & 0 & 0 \\ D_{12} & D_{12} & D_{11} & 0 & 0 & 0 \\ D_{41} & D_{42} & D_{43} & \frac{D_{11}-D_{12}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{D_{11}-D_{12}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{D_{11}-D_{12}}{2} \end{bmatrix} \quad (7)$$

$D_{11} = \frac{E(1-\nu)}{(1-2\nu)(1+\nu)}$ ,  $D_{12} = \frac{Ev}{(1-2\nu)(1+\nu)}$ ,  $E$  is the modulus of elasticity, and  $\nu$  is Poisson's ratio.

Based on the variational principle, Eq. (6) can be expressed as

$$\begin{aligned} \delta P &= \delta \left( \frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \mathbf{u}^T \mathbf{t} d\Gamma \right) \\ &= \left( \frac{1}{2} \int_{\Omega} \delta(\boldsymbol{\epsilon}^T \boldsymbol{\sigma}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \mathbf{t} d\Gamma \right) \end{aligned} \quad (8)$$

Owing to the symmetry of matrix  $\mathbf{D}$ , there is  $\delta(\boldsymbol{\epsilon}^T \boldsymbol{\sigma}) = \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} + \boldsymbol{\epsilon}^T \delta \boldsymbol{\sigma} = 2\delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma}$ .

Eq. (8) can be rewritten as

$$\delta P = \int_{\Omega} \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \mathbf{t} d\Gamma. \quad (9)$$

When the deformed object is in mechanical equilibrium,  $\delta P$  is equal to 0.

Substituting  $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} = \mathbf{D}\mathbf{L}\mathbf{u}$  and  $\boldsymbol{\epsilon} = \mathbf{L}\mathbf{u}$  into (9), we get

$$\int_{\Omega} (\mathbf{L}\mathbf{D}\mathbf{u}^T)(\mathbf{D}\mathbf{L}\mathbf{u}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \mathbf{t} d\Gamma = 0. \quad (10)$$

## 2.2. Radial basis shape function (RBSF)

Radial Basis Function (RBF) plays an important role in RPIM. We suppose that there is scalar function  $\mathbf{u}(\mathbf{x})$  ( $\mathbf{x} = [x, y, z]^T$  or  $\mathbf{x} = [x, y]^T$ ) and nodes in the domain of the problem.

The function  $\mathbf{u}(\mathbf{x})$  is expressed as

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^n R_i(\mathbf{x}) a_i + \sum_{j=1}^m p_j(\mathbf{x}) b_j. \quad (11)$$

Where  $R_i(\mathbf{x})$  is Radial Basis Function (RBF).  $a_i$  and  $b_j$  denote coefficients,  $p_j(\mathbf{x})$  is additional monomial.

Usually, polynomial function  $\sum_{j=1}^m p_j(\mathbf{x}) b_j$  is omitted because it is inclined to make the torque matrix singular. (The torque matrix is derived from Eq. (11).)

So, the Eq. (11) is rewritten as

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^n R_i(\mathbf{x}) a_i. \quad (12)$$

In order to determine the coefficients  $a_i$  ( $i = 1, 2, \dots, n$ ), we establish support domain in which the nodes  $\mathbf{x}_k$  ( $k = 1, 2, \dots, n$ ) and center point  $\mathbf{x}$  are included (see Figs. 1 and 2). Moreover, We assume that the values of scalar function  $\mathbf{u}(\mathbf{x}_k)$  ( $k = 1, 2, \dots, n$ ) at node  $\mathbf{x}_k$  are known. As far as the RBF  $R_i(\mathbf{x})$  is concerned, function  $R_i(\mathbf{x}_k) = (r_{ik}^2 + (\alpha_c d_c)^2)^q$  ( $i, k = 1, 2, \dots, n$ ) are applied.  $r_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2}$  or  $r_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$  is the distance between node  $\mathbf{x}_i$  and  $\mathbf{x}_k$  in the same support domain. In this paper, the 1.0, 2.0, and 1.03 are selected as shape parameters  $a_c, d_c$  and  $q$ , respectively [38].

Substituting  $\mathbf{x}_k$  ( $k = 1, 2, \dots, n$ ) into Eq. (12), we get

$$\begin{bmatrix} \mathbf{u}(\mathbf{x}_1) \\ \mathbf{u}(\mathbf{x}_2) \\ \mathbf{u}(\mathbf{x}_3) \\ \vdots \\ \mathbf{u}(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} R_1(\mathbf{x}_1) R_2(\mathbf{x}_1) R_3(\mathbf{x}_1) \dots R_n(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) R_2(\mathbf{x}_2) R_3(\mathbf{x}_2) \dots R_n(\mathbf{x}_2) \\ R_1(\mathbf{x}_3) R_2(\mathbf{x}_3) R_3(\mathbf{x}_3) \dots R_n(\mathbf{x}_3) \\ \vdots \\ R_1(\mathbf{x}_n) R_2(\mathbf{x}_n) R_3(\mathbf{x}_n) \dots R_n(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}. \quad (13)$$

or

$$\mathbf{U}_s = \mathbf{R}_s \mathbf{a} \quad (14)$$

$$\mathbf{U}_s = [\mathbf{u}(\mathbf{x}_1) \mathbf{u}(\mathbf{x}_2) \mathbf{u}(\mathbf{x}_3) \dots \mathbf{u}(\mathbf{x}_n)]^T.$$

$$\mathbf{R}_s = \begin{bmatrix} R_1(\mathbf{x}_1) & R_2(\mathbf{x}_1) & R_3(\mathbf{x}_1) & \dots & R_n(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) & R_2(\mathbf{x}_2) & R_3(\mathbf{x}_2) & \dots & R_n(\mathbf{x}_2) \\ R_1(\mathbf{x}_3) & R_2(\mathbf{x}_3) & R_3(\mathbf{x}_3) & \dots & R_n(\mathbf{x}_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_1(\mathbf{x}_n) & R_2(\mathbf{x}_n) & R_3(\mathbf{x}_n) & \dots & R_n(\mathbf{x}_n) \end{bmatrix}_{(n \times n)}.$$

$$\mathbf{a} = [a_1 \ a_2 \ a_3 \ \dots \ a_n]^T.$$

Where  $\mathbf{U}_s$  is vector of function,  $\mathbf{R}_s$  denotes torque matrix,  $\mathbf{a}$  is vector of coefficient that is obtained by

$$\mathbf{a} = \mathbf{R}_s^{-1} \mathbf{U}_s. \quad (15)$$

For any point  $\mathbf{x}$  in the same support domain, Eq. (12) can be rewritten as

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= [R_1(\mathbf{x}) \ R_2(\mathbf{x}) \ R_3(\mathbf{x}) \ \dots \ R_n(\mathbf{x})] \mathbf{a}, \\ \mathbf{u}(\mathbf{x}) &= [R_1(\mathbf{x}) \ R_2(\mathbf{x}) \ R_3(\mathbf{x}) \ \dots \ R_n(\mathbf{x})] \mathbf{R}_s^{-1} \mathbf{U}_s \end{aligned} \quad (16)$$

Now, we can get Radial Basis Shape Function (RBSF)  $\phi_i(\mathbf{x})$ .

$$\begin{aligned} \Phi(\mathbf{x}) &= [R(\mathbf{x}_1, \mathbf{x}) \ R(\mathbf{x}_2, \mathbf{x}) \ R(\mathbf{x}_3, \mathbf{x}) \ \dots \ R(\mathbf{x}_n, \mathbf{x})] \mathbf{R}_s^{-1}, \\ &= [\phi_1(\mathbf{x}) \ \phi_2(\mathbf{x}) \ \phi_3(\mathbf{x}) \ \dots \ \phi_n(\mathbf{x})] \end{aligned} \quad (17)$$

### 2.3. Introduction to the global meshless method

The domain of the problem is provided with fixed background units where Gaussian integral points are determined by Gauss-Legendre formula in each unit, and a support domain of a certain size is centered there to include some nodes, as shown in Figs. 1 and 2

By RBSF, the approximate displacement  $\mathbf{u}^h(\mathbf{x})$  of any point  $\mathbf{x}$  in a three-dimension support domain is obtained as

$$\begin{aligned} \mathbf{u}^h(\mathbf{x}) &= \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \phi_1(\mathbf{x}) & 0 & 0 \dots & \phi_n(\mathbf{x}) & 0 & 0 \\ 0 & \phi_1(\mathbf{x}) & 0 \dots & 0 & \phi_n(\mathbf{x}) & 0 \\ 0 & 0 & \phi_1(\mathbf{x}) \dots & 0 & 0 & \phi_n(\mathbf{x}) \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \\ u_n \\ v_n \\ w_n \end{bmatrix} \\ \mathbf{u}^h(\mathbf{x}) &= \sum_{l=1}^n \begin{bmatrix} \phi_l(\mathbf{x}) & 0 & 0 \\ 0 & \phi_l(\mathbf{x}) & 0 \\ 0 & 0 & \phi_l(\mathbf{x}) \end{bmatrix} \begin{bmatrix} u_l \\ v_l \\ w_l \end{bmatrix} = \sum_{l=1}^n \Phi_l(\mathbf{x}) \mathbf{u}_l. \end{aligned} \quad (18)$$

where  $Y_{61}$ ,  $v_l$  and  $w_l$  are the orthogonal components of displacement at any given node,  $\phi_l(\mathbf{x})$  for  $l = 1, 2, 3, \dots, n$  are shape functions, and  $n$  is number of nodes included in the support domain.

Substituting  $\mathbf{u}^h(\mathbf{x})$  into the first term of Eq. (10), we obtain the following:

$$\begin{aligned} &\int_{\Omega} (\mathbf{L} \delta \mathbf{u}^h(\mathbf{x}))^T (\mathbf{D} \mathbf{L} \mathbf{u}^h(\mathbf{x})) d\Omega \\ &= \int_{\Omega} \left( \mathbf{L} \sum_{l=1}^n \Phi_l(\mathbf{x}) \delta \mathbf{u}_l \right)^T \left( \mathbf{D} \sum_{j=1}^n \Phi_j(\mathbf{x}) \mathbf{u}_j \right) d\Omega, \\ &= \sum_{l=1}^n \sum_{j=1}^n \delta \mathbf{u}_l^T \underbrace{\left( \int_{\Omega} (\mathbf{L} \Phi_l(\mathbf{x}))^T (\mathbf{D} \Phi_j(\mathbf{x})) d\Omega \right)}_{\mathbf{K}_{lj}} \mathbf{u}_j, \\ &= \sum_{l=1}^n \sum_{j=1}^n \delta \mathbf{u}_l^T \mathbf{K}_{lj} \mathbf{u}_j. \end{aligned} \quad (19)$$

In order to apply global meshless method to any point  $\mathbf{x}$  in the entire domain of the problem, all nodes are numbered from 1 to N and Eq. (19) is calculated in all support domains.

Thus, we get

$$\int_{\Omega} (\mathbf{L} \delta \mathbf{u}^h(\mathbf{x}))^T (\mathbf{D} \mathbf{L} \mathbf{u}^h(\mathbf{x})) d\Omega = \sum_{l=1}^N \sum_{j=1}^N \delta \mathbf{u}_l^T \mathbf{K}_{lj} \mathbf{u}_j \quad (20)$$

$$\begin{aligned} \sum_{l=1}^N \sum_{j=1}^N \delta \mathbf{u}_l^T \mathbf{K}_{lj} \mathbf{u}_j &= \delta \mathbf{u}_1^T \mathbf{K}_{11} \mathbf{u}_1 + \delta \mathbf{u}_1^T \mathbf{K}_{12} \mathbf{u}_2 + \dots + \delta \mathbf{u}_1^T \mathbf{K}_{1N} \mathbf{u}_N \\ &\quad + \delta \mathbf{u}_2^T \mathbf{K}_{21} \mathbf{u}_1 + \delta \mathbf{u}_2^T \mathbf{K}_{22} \mathbf{u}_2 + \dots + \delta \mathbf{u}_2^T \mathbf{K}_{2N} \mathbf{u}_N \\ &\quad \vdots \\ &\quad + \delta \mathbf{u}_N^T \mathbf{K}_{N1} \mathbf{u}_1 + \delta \mathbf{u}_N^T \mathbf{K}_{N2} \mathbf{u}_2 + \dots + \delta \mathbf{u}_N^T \mathbf{K}_{NN} \mathbf{u}_N \\ &= \delta \mathbf{U}^T \mathbf{K} \mathbf{U}. \end{aligned} \quad (21)$$

Where  $[\mathbf{K}_{ij}]_{1 \leq i, j \leq N}$  is a  $3 \times 3$  element stiff matrix,  $I$  and  $J$  denote the signs of different nodes in the same support domain.

Notice that if the node  $I$  and  $J$  don't belong to the same support domain, the  $\mathbf{K}_{ij}$  is equal to zero.

$\mathbf{K}$  is total stiff matrix.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{K}_{13} & \dots & \mathbf{K}_{1N} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{K}_{23} & \dots & \mathbf{K}_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{N1} & \mathbf{K}_{N2} & \mathbf{K}_{N3} & \dots & \mathbf{K}_{NN} \end{bmatrix}. \quad (22)$$

$\mathbf{U}$  is the vector of displacements of nodes.

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix}. \quad (23)$$

Similarly, substituting  $\mathbf{u}^h(\mathbf{x})$  into the second term of Eq. (10), we get

$$\begin{aligned} \int_{\Omega} \delta(\mathbf{u}^h(\mathbf{x}))^T \mathbf{b} d\Omega &= \int_{\Omega} \delta \left( \sum_{l=1}^N \Phi_l(\mathbf{x}) \mathbf{u}_l \right)^T \mathbf{b} d\Omega. \\ \sum_{l=1}^N \delta \mathbf{u}_l^T \int_{\Omega} \Phi_l^T \mathbf{b} d\Omega &= \delta \mathbf{u}_1^T \mathbf{F}_1^b + \delta \mathbf{u}_2^T \mathbf{F}_2^b + \dots + \delta \mathbf{u}_N^T \mathbf{F}_N^b. \end{aligned} \quad (24)$$

Substituting  $\mathbf{u}^h(\mathbf{x})$  into the third term of Eq. (10), we get

$$\begin{aligned} \int_{\Omega} \delta(\mathbf{u}^h(\mathbf{x}))^T \mathbf{b} d\Omega &= \int_{\Gamma_t} \delta \left( \sum_{l=1}^N \Phi_l(\mathbf{x}) \mathbf{u}_l \right)^T \mathbf{t} d\Gamma. \\ \sum_{l=1}^N \delta \mathbf{u}_l^T \int_{\Gamma_t} \Phi_l^T \mathbf{t} d\Gamma &= \delta \mathbf{u}_1^T \mathbf{F}_1^t + \delta \mathbf{u}_2^T \mathbf{F}_2^t + \dots + \delta \mathbf{u}_N^T \mathbf{F}_N^t. \end{aligned} \quad (25)$$

$$\delta \mathbf{U}^T \mathbf{F}^t.$$

Thus, Eq. (10) can be rewritten as

$$\delta \mathbf{U}^T \mathbf{K} \mathbf{U} - \delta \mathbf{U}^T \mathbf{F}^b - \delta \mathbf{U}^T \mathbf{F}^t = 0. \quad (26)$$

Omitting  $\delta \mathbf{U}^T$ , we get

$$\mathbf{K} \mathbf{U} - \underbrace{(\mathbf{F}^b + \mathbf{F}^t)}_{\mathbf{F}} = 0. \quad (27)$$

Solving Eq. (25), we get the displacements of all nodes.

Eq. (25) is a general description of the mechanics of soft tissues. To determine the stability and effectiveness of calculation of its solution, the essential boundary condition (3) is applied by penalty method as follows:

Assume node  $J$  is on the essential boundary. Then,  $\mathbf{K}_{jj}$  in total stiff matrix (22) is expressed as  $\lambda \mathbf{K}_{jj}$ , where  $\lambda \geq 10^7$ . The corresponding component  $\mathbf{F}_j$  of the total force vector  $\mathbf{F}$  is denoted as  $\mathbf{F}_j = \lambda \mathbf{K}_{jj} \bar{\mathbf{u}}_j$ . Substituting  $\lambda \mathbf{K}_{jj}$  and  $\lambda \mathbf{K}_{jj} \bar{\mathbf{u}}_j$  into the Eq. (27), we get

$$\begin{bmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1(j-1)} & \mathbf{K}_{1j} & \mathbf{K}_{1(j+1)} & \cdots & \mathbf{K}_{1N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{K}_{j1} & \cdots & \mathbf{K}_{j(j-1)} & \lambda \mathbf{K}_{jj} & \mathbf{K}_{j(j+1)} & \cdots & \mathbf{K}_{(j-1)N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{K}_{N1} & \cdots & \mathbf{K}_{N(j-1)} & \mathbf{K}_{NJ} & \mathbf{K}_{N(j+1)} & \cdots & \mathbf{K}_{NN} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_j \\ \vdots \\ \mathbf{u}_N \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \lambda \mathbf{K}_{jj} \bar{\mathbf{u}}_j \\ \vdots \\ \mathbf{F}_N \end{bmatrix}. \quad (28)$$

Because  $\lambda$  is very large, the  $j$ th line of Eq. (28) can be approximately rewritten as

$$\begin{bmatrix} 0 & \cdots & 0 & \cdots & \lambda \mathbf{K}_{jj} & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_j \\ \vdots \\ \mathbf{u}_N \end{bmatrix} = \lambda \mathbf{K}_{jj} \bar{\mathbf{u}}_j. \quad (29)$$

So, the essential boundary condition (3) is applied.

#### 2.4. Introduction to Marquardt's algorithm

The typical non-linear relation is given by:

$$y = f(x_1, x_2, \dots, x_p; b_1, b_2, \dots, b_m) + X \quad (30)$$

where  $f$  is a known non-linear function,  $x_1, x_2, \dots, x_p$  are  $p$  independent variables depending on the dimensionality of the domain of the problem such as  $x, y$  or  $z$ .  $b_1, b_2, \dots, b_m$  are  $m$  unknown parameters to be estimated,  $m$  is related to the variable order that we use 1STOP software package to perform the cross-validations [39] to obtain, and  $X$  is a random error term. If  $y$  and  $x_1, x_2, \dots, x_p$  are subject to  $n$  observations,  $n$  sets of data will be obtained:

$$(x_{i1}, x_{i2}, \dots, x_{ip}, y_i) \quad i = 1, 2, \dots, n$$

Substitute the values of the independent variables in the  $i$ th observation into Eq. (30) to obtain:

$$f(x_{i1}, x_{i2}, \dots, x_{ip}; b_1, b_2, \dots, b_m) = f(\mathbf{x}_i, \mathbf{b}) \quad (31)$$

Because  $x_1, x_2, \dots, x_p$  are known,  $f(x_i, b)$  is the function for  $b_1, b_2, \dots, b_m$ . Assign an initial value  $\mathbf{b}^{(0)} = (b_1^{(0)}, b_2^{(0)}, \dots, b_m^{(0)})$  to  $\mathbf{b}$  and expand  $f(\mathbf{x}_i, \mathbf{b})$  at  $\mathbf{b}^{(0)}$  via Taylor series, omitting the quadratic and above terms:

$$\begin{aligned} f(\mathbf{x}_i, \mathbf{b}) &= f(\mathbf{x}_i, \mathbf{b}^{(0)}) + \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_1} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} (b_1 - b_1^{(0)}) + \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_2} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} \\ &\quad \times (b_2 - b_2^{(0)}) + \dots + \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_m} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} (b_m - b_m^{(0)}) \end{aligned} \quad (32)$$

Notice that the above is a linear function for  $b_1, b_2, \dots, b_m$  where only  $b_1, b_2, \dots, b_m$  are unknowns. By the method of least squares, the residual sum of squares is:

$$\begin{aligned} J &= \sum_{i=1}^m \left\{ y_i - \left[ f(\mathbf{x}_i, \mathbf{b}^{(0)}) + \sum_{j=1}^m \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_j} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} (b_j - b_j^{(0)}) \right] \right\}^2 \\ &\quad + d \sum_{j=1}^m (b_j - b_j^{(0)})^2 \end{aligned} \quad (33)$$

where  $d \geq 0$  is a damping factor. The case where  $d=0$  reduces to the Gauss–Newton method, which is a special form of Marquardt's

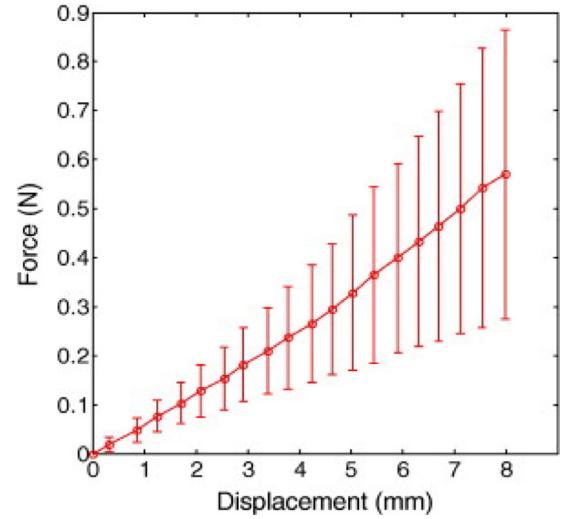


Fig. 3. Relationship between the contact depth and pressure on the surface of pork liver.

algorithm and is stricter to selection of the initial value compared to the general case.

To minimize  $J$ , set the first-order partial derivatives of  $J$  to 0 for  $b_1, b_2, \dots, b_m$  to obtain the equation set:

$$\begin{aligned} 0 &= \frac{\partial J}{\partial b_k} = 2 \sum_{i=1}^m \left[ y_i - f(\mathbf{x}_i, \mathbf{b}^{(0)}) - \sum_{j=1}^m \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_j} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} (b_j - b_j^{(0)}) \right] \\ &\quad \times \left( -\frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_k} \right) \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} + 2d(b_k - b_k^{(0)}) \\ k &= 1, 2, \dots, m. \end{aligned} \quad (34)$$

This equation set can be converted into the following form:

$$\begin{cases} (a_{11} + d)(b_1 - b_1^{(0)}) + a_{12}(b_2 - b_2^{(0)}) + \dots + a_{1m}(b_m - b_m^{(0)}) = a_{1y} \\ a_{21}(b_1 - b_1^{(0)}) + (a_{22} + d)(b_2 - b_2^{(0)}) + \dots + a_{2m}(b_m - b_m^{(0)}) = a_{2y} \\ \dots \\ a_{m1}(b_1 - b_1^{(0)}) + a_{m2}(b_2 - b_2^{(0)}) + \dots + (a_{mm} + d)(b_m - b_m^{(0)}) = a_{my} \end{cases}. \quad (35)$$

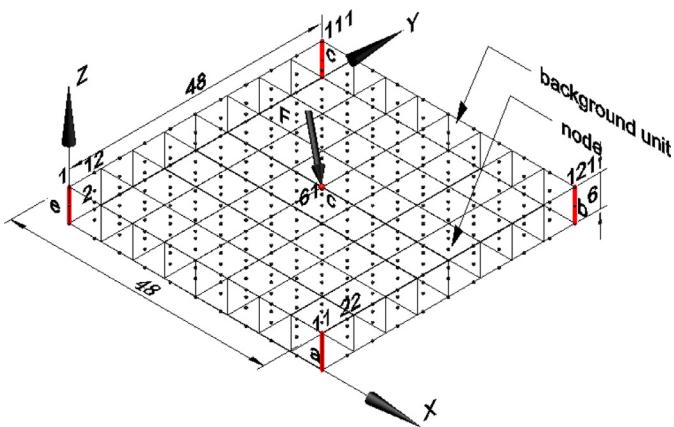
where

$$\begin{cases} a_{jk} = \sum_{i=1}^m \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_j} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_k} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} = a_{kj} & j = 1, 2, \dots, m; k = 1, 2, \dots, m. \\ a_{jy} = \sum_{i=1}^m (y_i - f(\mathbf{x}_i, \mathbf{b}^{(0)})) \frac{\partial f(\mathbf{x}_i, \mathbf{b})}{\partial b_j} \Big|_{\mathbf{b}=\mathbf{b}^{(0)}} = a_{jy} & \end{cases} \quad (36)$$

with solution

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{bmatrix} + \begin{bmatrix} a_{11} + d^{(0)} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} + d^{(0)} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} + d^{(0)} \end{bmatrix}^{-1} \begin{bmatrix} a_{1y} \\ a_{2y} \\ \vdots \\ a_{my} \end{bmatrix}. \quad (37)$$

Obviously, the solution is related to the initial values  $(b_1^{(0)}, b_2^{(0)}, \dots, b_m^{(0)})$  and  $d^{(0)}$ . Fix a maximum error ( $\delta$ ). If the value of  $|b_j - b_j^{(0)}| < \delta$  for a given value of  $j$ , the estimation is considered to be a success. Otherwise, substitute the  $b_j$  computed from the previous step into Eq. (31) as the new  $b_j^{(0)}$  and iterate the calculations of  $b_j$  as shown in Eq. (35) until the absolute difference is below  $\delta$ . Since  $a_{1y}, a_{2y}, \dots, a_{my}$  are constant values, larger values of  $d$  result in smaller values of  $|b_1 - b_1^{(0)}|, |b_2 - b_2^{(0)}|, \dots, |b_m - b_m^{(0)}|$ . Thus, the convergence can be expedited by increasing  $d$  during each new iteration. In the rare situation that  $\lim_{m \rightarrow \infty} \sum_{j=1}^m (b_j - b_j^{(0)})^2 = 0$ , large



**Fig. 4.** Deformable element soft tissue model. The red edges are fixed. Similarly, the fixed force bearing node (61) is also in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

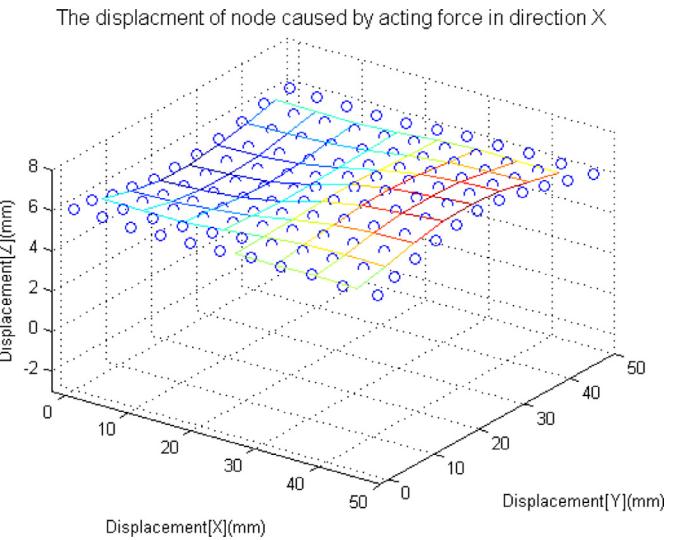
values of  $d$  increase the iterations necessary for calculation. To prevent this, during each new iteration,  $d$  should be decreased as long as the value of  $J$  (Eq. (33)) is also decreased. In practice,  $d$  should be adjusted to appropriate value (that real-timely becomes small or large) to assure the convergence of the iteration.

### 3. Method

Under the action of a spatial force  $\mathbf{F}$ , the surface deformation of a biological soft tissue model is described by the following equation:

$$S = v_0 + \Delta x + \Delta y + \Delta z + N. \quad (38)$$

where  $v_0$  is an initial coordinate  $(x_0, y_0, z_0)$  of a node on the surface when the model is free of any external force,  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  are the displacements relative to  $v_0$  in the X, Y, and Z direc-



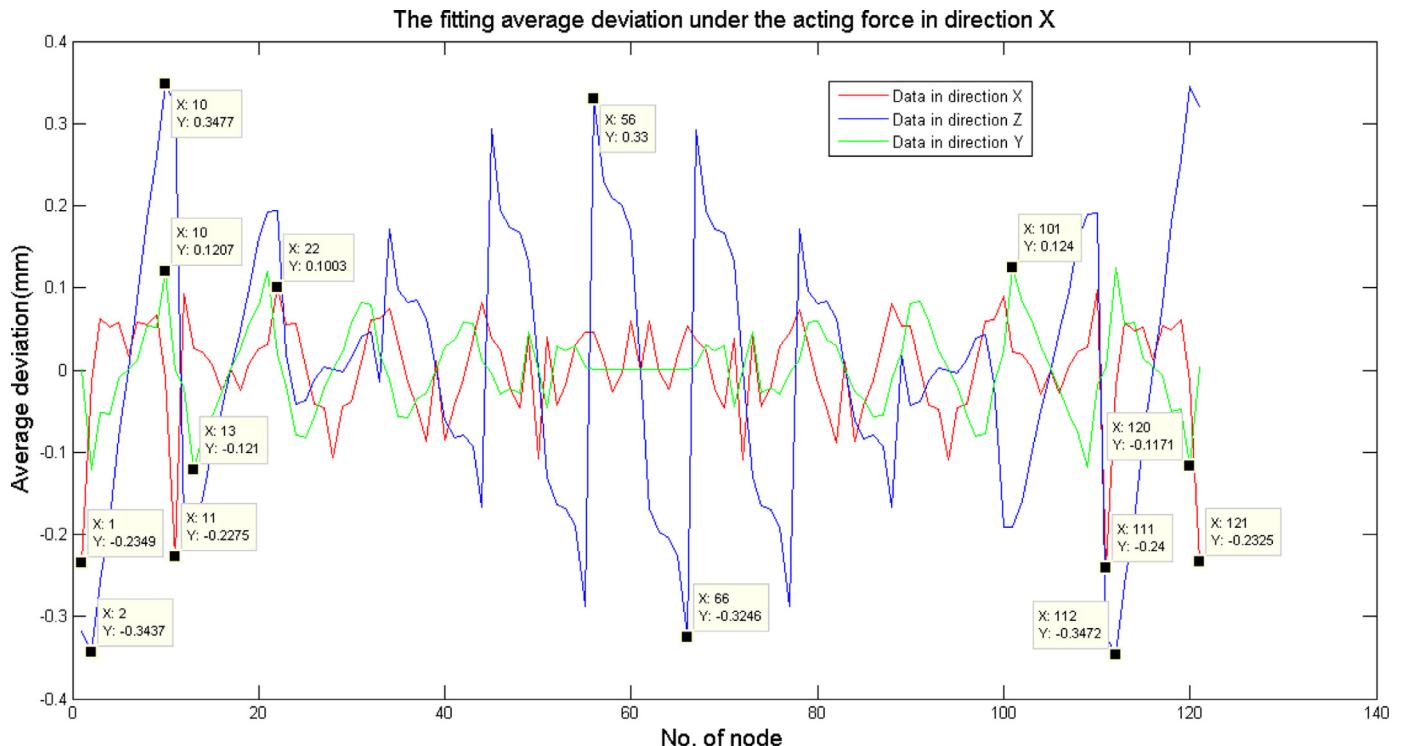
**Fig. 6.** Nodal displacement caused by spatial force in the X direction, as calculated by RPIM.

tions after the model is deformed by a force, and  $N$  is a random error term.

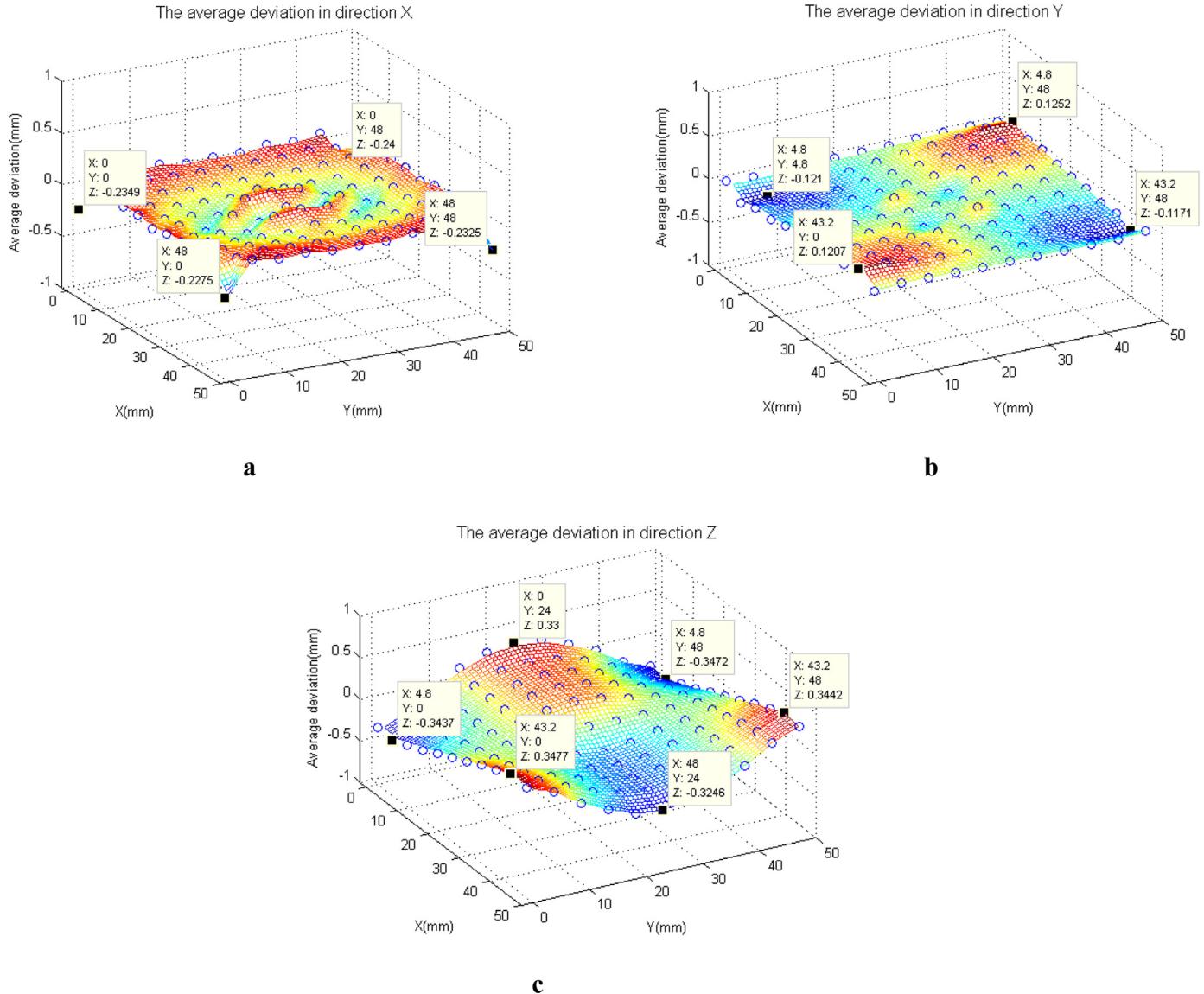
The displacement of all nodes corresponding to a given spatial force (see Fig. 4) was calculated by global meshless method in advance. And then Marquardt's algorithm is used to fit the relation between force and displacements of the nodes on surface, which also generates the fitting surface to display the total deformation of the model.

#### 3.1. The fitting relation between force and force bearing node

To establish the relation between the applied force and the deformation surface, we shall establish a fitting relation between



**Fig. 5.** Mean deviation of surface nodal displacement, as calculated by RPIM, when force is applied in the X direction.



**Fig. 7.** (a) Distribution of mean deviation of nodal displacement in the X direction when applying a X direction spatial force; (b) Distribution of mean deviation of nodal displacement in the Y direction when applying force in the X direction; (c) Distribution of mean deviation of nodal displacement in the Z direction when applying force in the X direction.

the applied force  $\mathbf{F}$  and the force bearing node  $c$  (The red node in the Fig. 4. is the force bearing node, which is defined as node  $c$ ). The displacements  $\Delta s_c$  of the force bearing node  $c$  under  $n$  different spatial forces  $\mathbf{F}$  is computed by the meshless method. The applied spatial force  $\mathbf{F}$  is decomposed into component forces  $\mathbf{F}_x$ ,  $\mathbf{F}_y$ , and  $\mathbf{F}_z$  in three directions, and the corresponding component displacements at the force bearing node  $c$  in these three directions are  $\Delta s_c = (\Delta x_c, \Delta y_c, \Delta z_c); i=1, 2, \dots, n$ . The process of applying Marquardt's algorithm for fitting is described as follows:

For  $n$  forces  $F_z$  and component displacements  $\Delta z_c$  on the force bearing node  $c$ , there exists a relation  $g_{zi}$  that satisfies  $\mathbf{F}_{zi} \rightarrow \Delta z_{ci}$ ,  $i=1, 2, \dots, n$ . Each  $g_{zi}$  has the functional form:

$$\Delta z_c = g_z(F_z^1, F_z^2, \dots, F_z^\gamma; \alpha_1, \alpha_2, \dots, \alpha_{\lambda+1}) + N \quad (39)$$

Where  $\gamma$  is the highest order of independent variables of the system,  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_h] (h=\lambda+1)$  is the system parameter, and  $N$  is the system error term.

The independent variable order  $\lambda$  of the system is determined by cross validation, which allows us to rewrite (39) as:

$$\Delta z_c = \alpha_1 + \alpha_2 F_z^1 + \dots + \alpha_{\lambda+1} F_z^\lambda. \quad (40)$$

Substituting initially estimated parameter  $\boldsymbol{\alpha}^{(0)} = [\alpha_1^{(0)}, \alpha_2^{(0)}, \dots, \alpha_h^{(0)}]$  and  $\mathbf{F}_{zi}$  for  $\mathbf{b}^{(0)}$  and  $\mathbf{x}_i$  in Eqs. (35), (36), we obtain the parameter  $\boldsymbol{\alpha}$  in Eq. (40).

By the same way, the corresponding nodal displacement functions for component forces  $F_x$  and  $F_y$  are:

$$\Delta x_c = r_1 + r_2 F_x^1 + \dots + r_{b+1} F_x^b. \quad (41)$$

$$\Delta y_c = s_1 + s_2 F_y^1 + \dots + s_{e+1} F_y^e. \quad (42)$$

### 3.2. Surface fitting

Deformation occurs when the action of a spatial force on one node changes the displacements of other nodes. To model this, we fit the relation between the displacements of the force bearing

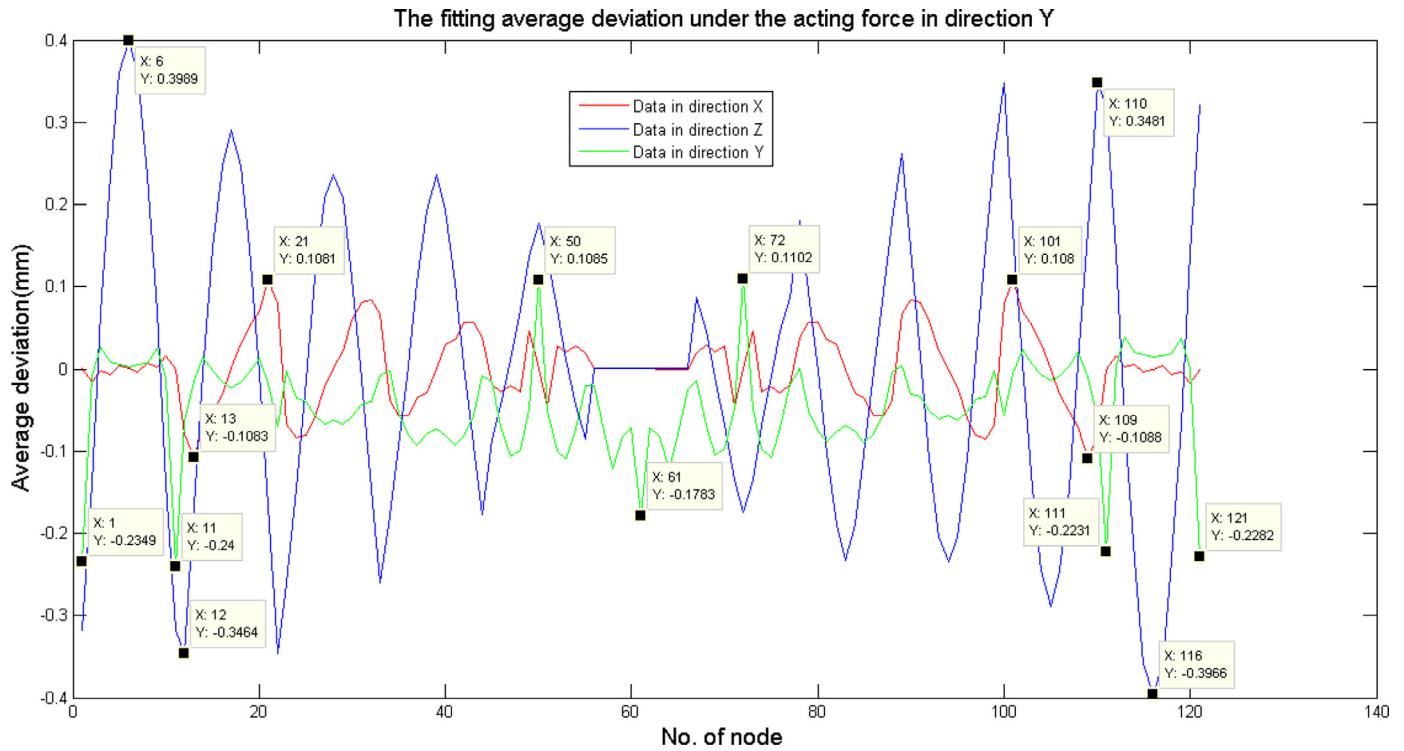


Fig. 8. Mean deviation of surface nodal displacement, as calculated by RPIM, when force is applied in the Y direction.

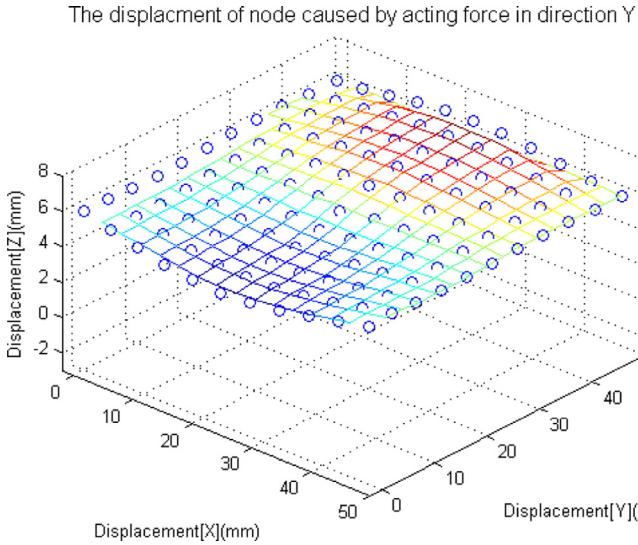


Fig. 9. Nodal displacement caused by spatial force in the Y direction, as calculated by RPIM.

node (henceforth known as original displacement) to the displacements of other nodes (henceforth known as induced displacement) on the model's surface with Marquardt's algorithm to express the deformation surface. Assume that  $n$  different spatial forces  $\mathbf{F}$  are applied to the force bearing node  $c$ . For component  $\Delta z_{ci}$  of original displacement, the corresponding induced displacement at node  $j$  ( $\Delta x_j, \Delta y_j, \Delta z_j$ )<sub>i</sub> is given by  $\Delta z_{ci} \rightarrow (\Delta x_j, \Delta y_j, \Delta z_j)_i, i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1$ .

To simplify matters, we all assume all  $m$  surface nodes have the same  $Z$  coordinate (see Fig. 4). Thus, the initial difference between nodes involves only their  $X$  and  $Y$  coordinates.

Under the action of  $\mathbf{F}_{zi}$ , the induced displacement function is:

$$\begin{cases} \Delta x_{ji} = \gamma_{z,x}(x_{jo}^1, x_{jo}^2, \dots, x_{jo}^{a_1}, y_{jo}^1, y_{jo}^2, \dots, y_{jo}^{b_1}, p_{i11}, p_{i12}, \dots, p_{i1h_1}) \\ \Delta y_{ji} = \gamma_{z,y}(x_{jo}^1, x_{jo}^2, \dots, x_{jo}^{a_2}, y_{jo}^1, y_{jo}^2, \dots, y_{jo}^{b_2}, p_{i21}, p_{i22}, \dots, p_{i2h_2}) \\ \Delta z_{ji} = \gamma_{z,z}(x_{jo}^1, x_{jo}^2, \dots, x_{jo}^{a_3}, y_{jo}^1, y_{jo}^2, \dots, y_{jo}^{b_3}, p_{i31}, p_{i32}, \dots, p_{i3h_3}) \end{cases} \quad i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1. \quad (43)$$

where  $h_1 = a_1 + b_1 + 1, h_2 = a_2 + b_2 + 1, h_3 = a_3 + b_3 + 1, j = 1, 2, \dots, m - 1, 0 \leq x_{jo}, y_{jo} \leq 48$  (here,  $x_{jo}, y_{jo}$  indicate the initial coordinate of node  $j$  when the surface is free of force).

By cross-validation, the surface deformation order  $a_1, a_2, a_3, b_1, b_2, b_3$  can be obtained.

Define the three surface deformation function parameters  $\mathbf{p}_{i1}, \mathbf{p}_{i2}, \mathbf{p}_{i3}$  as

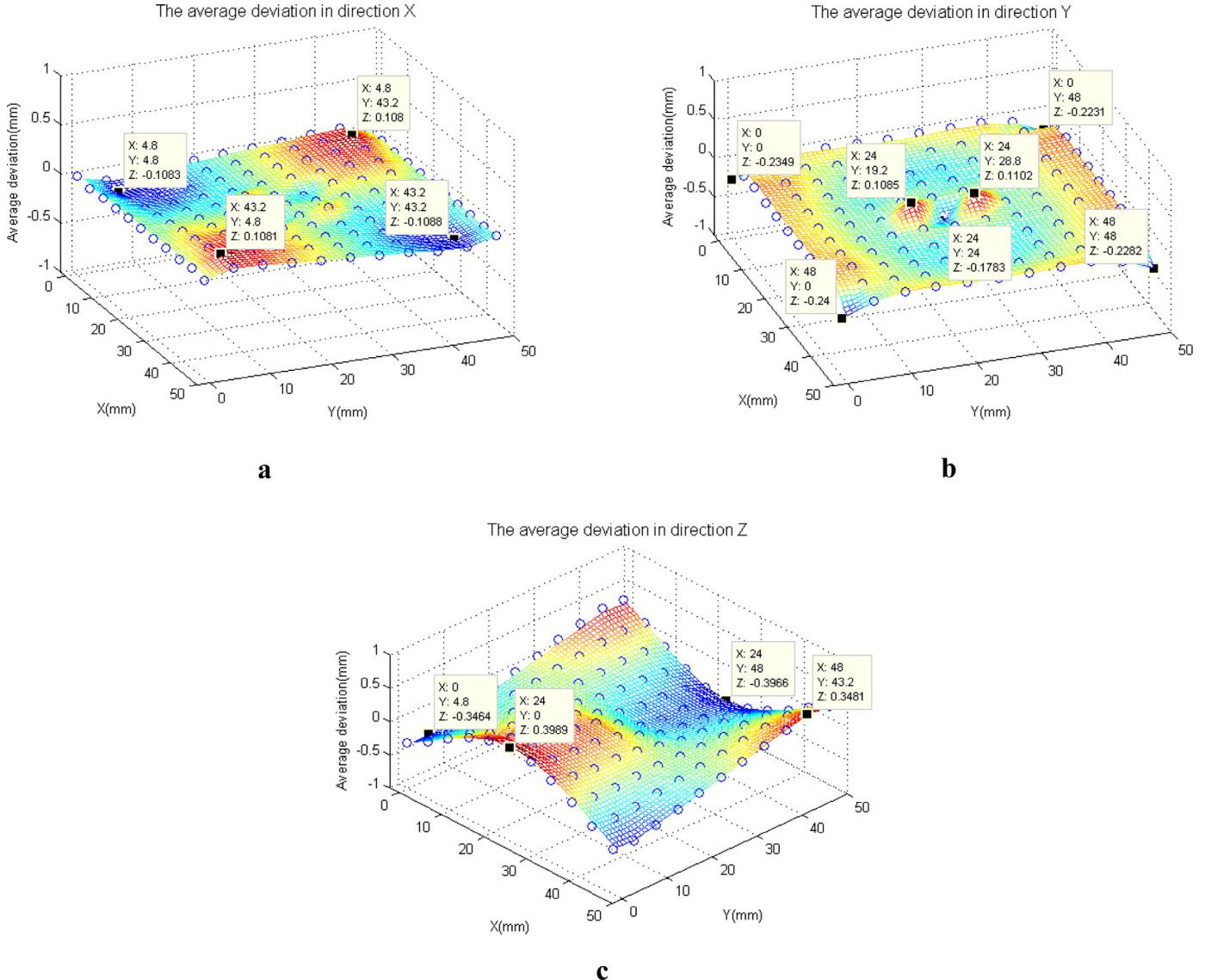
$$\mathbf{p}_{i1} = [p_{i11}, p_{i12}, \dots, p_{i1h_1}], \quad \mathbf{p}_{i2} = [p_{i21}, p_{i22}, \dots, p_{i2h_2}], \text{ and} \\ \mathbf{p}_{i3} = [p_{i31}, p_{i32}, \dots, p_{i3h_3}] \quad i = 1, 2, \dots, n$$

The Eq. (43) is rewritten as surface function:

$$\begin{cases} \Delta x = p_{i11} + p_{i12}x + \dots + p_{i1a_1+1}x^{a_1} + p_{i1a_1+2}y + \dots + p_{i1h_1}y^{b_1} \\ \Delta y = p_{i21} + p_{i22}x + \dots + p_{i2a_2+1}x^{a_2} + p_{i2a_2+2}y + \dots + p_{i2h_2}y^{b_2} \\ \Delta z = p_{i31} + p_{i32}x + \dots + p_{i3a_3+1}x^{a_3} + p_{i3a_3+2}y + \dots + p_{i2h_3}y^{b_3} \end{cases} \quad i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1 \quad (44)$$

Substituting initially estimated parameter  $\mathbf{P}_{i1}^{(0)}, \mathbf{P}_{i2}^{(0)}, \mathbf{P}_{i3}^{(0)}$  and  $(\Delta x_j + x_{jo}, \Delta y_j + y_{jo})$  for  $\mathbf{b}^{(0)}$  and  $\mathbf{x}_i$  (here  $i=j=1, 2, \dots, m-1$ ) in Eq. (35), (36), we obtain the parameter  $\mathbf{P}_{i1}, \mathbf{P}_{i2}, \mathbf{P}_{i3}$  in Eq. (44).

By the same way, the corresponding induced surface functions for component forces  $F_x$  and  $F_y$  also can be obtained. Here,  $\mathbf{P}_{i1}, \mathbf{P}_{i2}, \mathbf{P}_{i3}, \dots$  involved in all these surface functions are called as parameter  $\mathbf{P}$ .



**Fig. 10.** (a) Distribution of mean deviation of nodal displacement in the X direction when applying a Y direction spatial force; (b) Distribution of mean deviation of nodal displacement in the Y direction when applying force in the Y direction; (c) Distribution of mean deviation of nodal displacement in the Z direction when applying force in the Y direction.

### 3.3. The relation between original displacement and the surface function

Given that the force on the force bearing node is always changing,  $\mathbf{P}$  (and thus, the model's surface) will also undergo constant change. To calculate continuous parameter  $\mathbf{P}$  in real time, we use Marquardt's algorithm to find a predictive relation  $\rho$  that satisfies  $\mathbf{P} = \rho(\Delta s_c)$ . The process for finding this relation is given below:

We observe that the relation  $\rho$  satisfies,  $\Delta z_{ci} \rightarrow [p_{i1}, p_{i2}, p_{i3}], i=1, 2, \dots, n$ , where

$$\begin{cases} \mathbf{p}_{i1} = \rho_{z1}(\Delta z_c^1, \Delta z_c^2, \dots, \Delta z_c^{k_1}, \beta_{11}, \beta_{12}, \dots, \beta_{1(k_1+1)}) \\ \mathbf{p}_{i2} = \rho_{z2}(\Delta z_c^1, \Delta z_c^2, \dots, \Delta z_c^{k_2}, \beta_{21}, \beta_{22}, \dots, \beta_{2(k_2+1)}) \\ \mathbf{p}_{i3} = \rho_{z3}(\Delta z_c^1, \Delta z_c^2, \dots, \Delta z_c^{k_3}, \beta_{31}, \beta_{32}, \dots, \beta_{3(k_3+1)}) \end{cases} \quad (45)$$

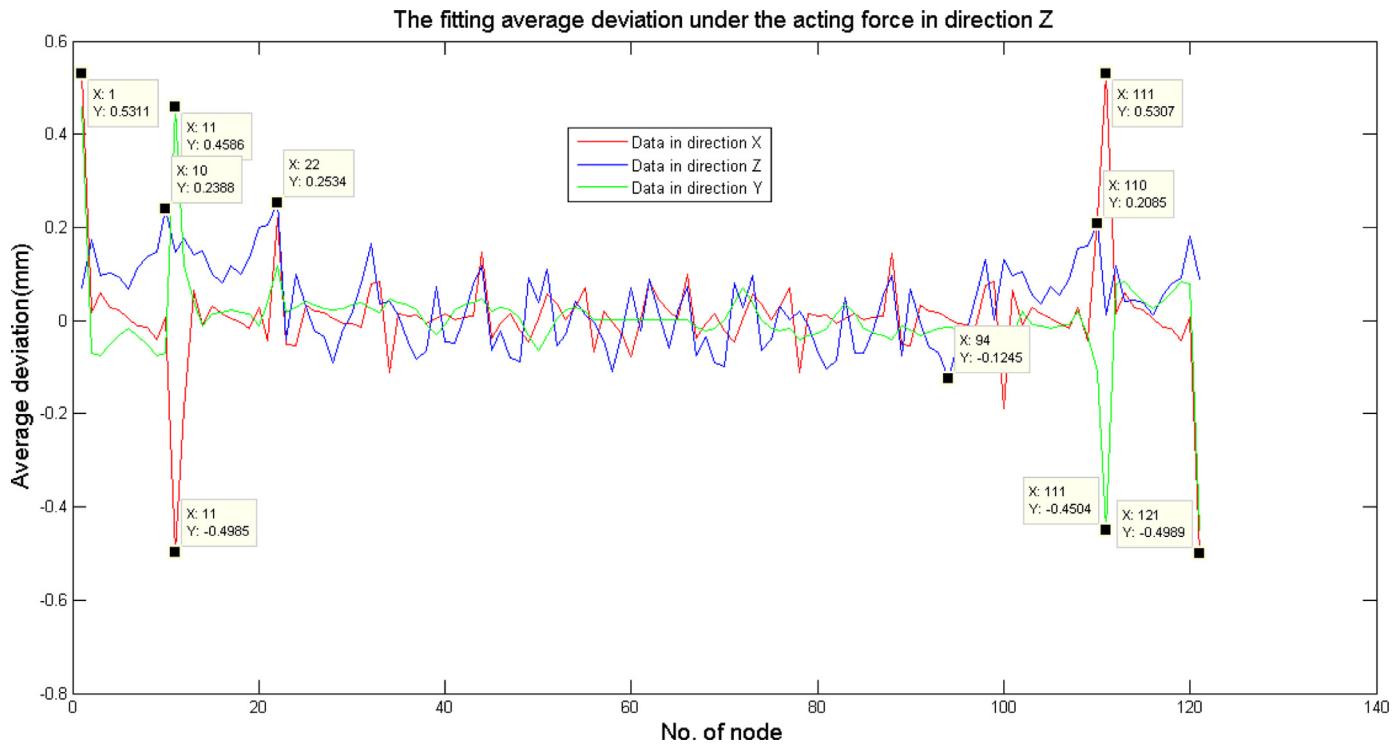
The fitting orders  $k_1, k_2, k_3$  are obtained via cross-validation.

$$\begin{aligned} \boldsymbol{\beta}_{11} &= [\beta_{111}, \beta_{112}, \dots, \beta_{11h_1}]^T, \quad \boldsymbol{\beta}_{12} = [\beta_{121}, \beta_{122}, \dots, \beta_{12h_1}]^T \\ \boldsymbol{\beta}_{21} &= [\beta_{211}, \beta_{212}, \dots, \beta_{21h_2}]^T, \quad \boldsymbol{\beta}_{22} = [\beta_{221}, \beta_{222}, \dots, \beta_{22h_2}]^T \\ \boldsymbol{\beta}_{31} &= [\beta_{311}, \beta_{312}, \dots, \beta_{31h_3}]^T, \quad \boldsymbol{\beta}_{32} = [\beta_{321}, \beta_{322}, \dots, \beta_{32h_3}]^T \end{aligned}$$

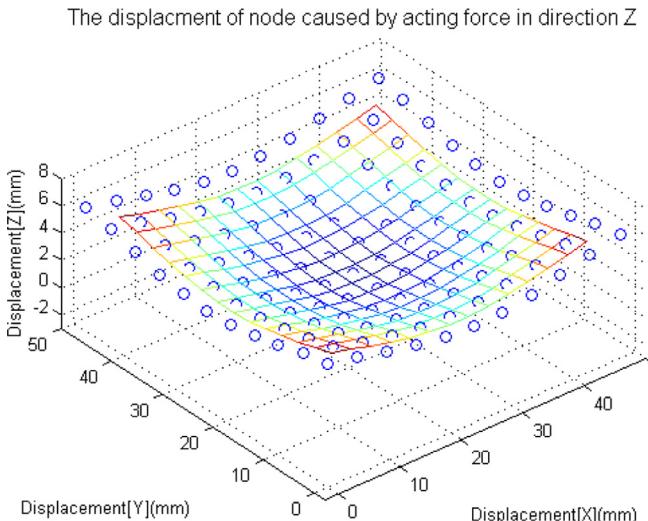
where  $\beta_{11}, \beta_{12}, \dots, \beta_{21}, \beta_{22}, \dots, \beta_{31}, \beta_{32}, \dots$  are collectively referred to as parameter  $\boldsymbol{\beta}$ . By Marquardt's Algorithm with the same bounds as [Section 3.1](#), we can solve for parameter  $\boldsymbol{\beta}$ . Substituting  $\boldsymbol{\beta}$  into [Eq. \(46\)](#), we obtain the value of parameter  $\mathbf{P}$ .

$$\begin{cases} \mathbf{p}_{i1}^T = \beta_{11} + \Delta z_c^1 \beta_{12} + \Delta z_c^2 \beta_{13} + \dots + \Delta z_c^{k_1} \beta_{1(k_1+1)} \\ \mathbf{p}_{i2}^T = \beta_{21} + \Delta z_c^1 \beta_{22} + \Delta z_c^2 \beta_{23} + \dots + \Delta z_c^{k_2} \beta_{2(k_2+1)} \\ \mathbf{p}_{i3}^T = \beta_{31} + \Delta z_c^1 \beta_{32} + \Delta z_c^2 \beta_{33} + \dots + \Delta z_c^{k_3} \beta_{3(k_3+1)} \end{cases} \quad (46)$$

The corresponding parameter  $\mathbf{P}$  can be solved from  $\Delta x_c, \Delta y_c$  in similar fashion. Thus, surface deformation induced by a continuously changing spatial force can be determined in real-time. By [Eqs. \(40\)–\(42\)](#), the original displacement can be calculated for any given spatial force's action on a force bearing node. Since  $\boldsymbol{\beta}$  is constant, for any given original displacement, [Eq. \(46\)](#) allows us to solve for  $\mathbf{P}$ , the parameter of surface fitting. Since  $\mathbf{P}$  is constant, for any given original displacement, [Eq. \(44\)](#) yields the surface created by the induced displacements.



**Fig. 11.** Mean deviation of surface nodal displacement, as calculated by RPIM, when force is applied in the Z direction.



**Fig. 12.** Nodal displacement caused by spatial force in the Z direction, as calculated by RPIM.

## 4. Results

### 4.1. Establishment of the soft tissue model

The deformable soft tissue simulated in this model is that of pork liver (see Fig. 4). The elastic modulus was calculated experimentally on a real sample in the manner described by Samur et al. [40]. A contact probe is pressed against the sample to measure the relationship between the force and displacement at the point of contact (see Fig. 3).

Select an ordered pair from the above graph (6 mm, 0.4 N) and substitute the data point into the Lee and Radok Equation [41]:

$$G = \frac{3F}{16\delta\sqrt{R\delta}} \quad (47)$$

where  $G$  is the shear modulus,  $F$  is the contact force,  $\delta$  is the contact displacement, and  $R$  is the radius of contact probe (2 mm).

To calculate the elastic modulus, we use the following equation, selecting the Poisson ratio (0.49) to be that of rubber, a similar material:

$$E = 2G(1 + \nu) \quad (48)$$

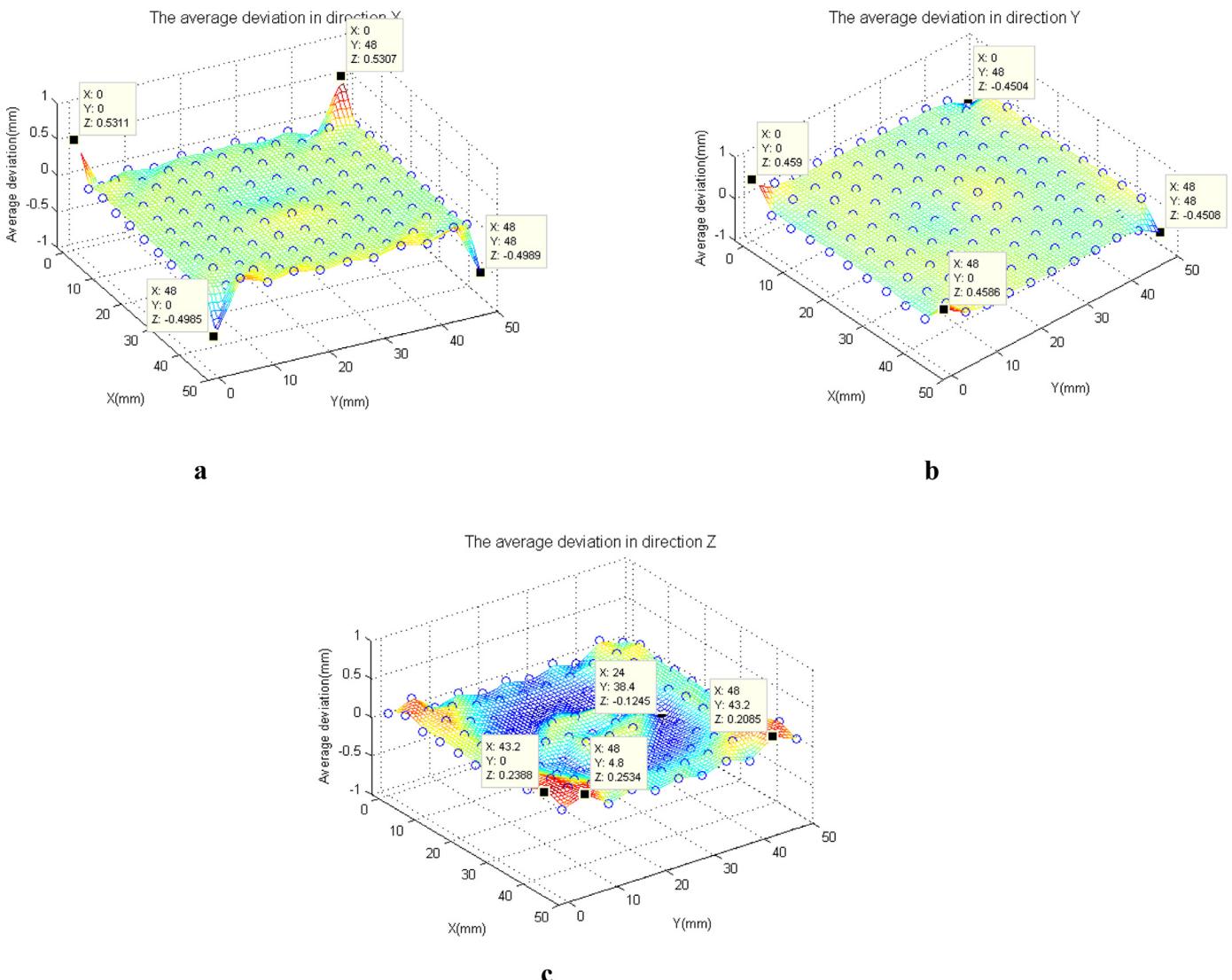
which yields  $E = 0.32$  MPa.

In the model (with dimensions  $48 \times 48 \times 6$  mm), four edges are fixed. At  $Z=0, 3$ , and  $6$  mm, there is a square ( $11 \times 11$ ) layer of nodes that is evenly dispersed. Since the upper surface of the model ( $Z=6$ ) has the most noticeable deformation, we focus our analysis only on this layer. Each node is spaced 4.8 mm from neighboring nodes of the same layer, and is given a number according to the formula  $N = (x/4.8) + 11 \times (y/4.8) + 1$ , where  $N$  is the node number,  $x$  is the X coordinate of the node, and  $y$  is the Y coordinate of the node. As a further simplification, all external forces are applied to the same force bearing node, Node 61 (which is at the center of the model). The model's problem domain is divided into 64 ( $6 \times 6 \times 6$  mm cubic) background units to determine the integral points for the Gaussian integration step of RPIM-MM.

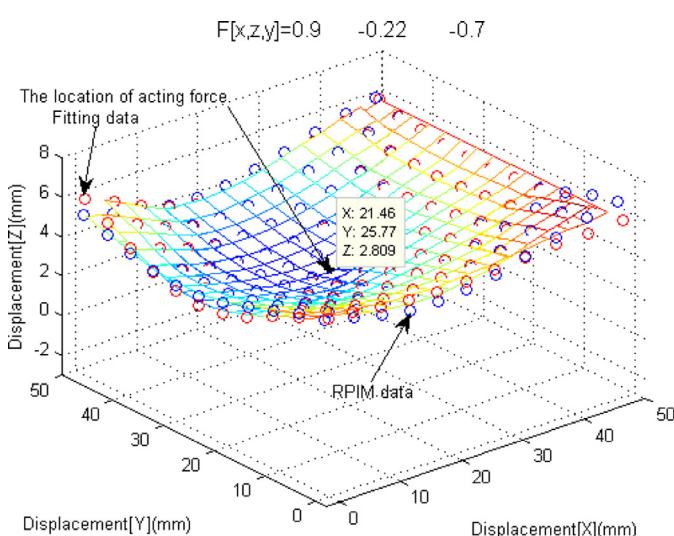
### 4.2. The application of Marquardt's algorithm

#### 4.2.1. Fitting relation between force and force bearing node

The relation between the force bearing node's displacement and the applied force is fitted first. Next, the X, Y, Z components of force are successively applied to Node 61, and corresponding displacements of Node 61 are computed by RPIM-MM. The component forces in three directions and corresponding component dis-



**Fig. 13.** (a) Distribution of mean deviation of nodal displacement in the X direction when applying a Z direction spatial force; (b) Distribution of mean deviation of nodal displacement in the Y direction when applying force in the Z direction; (c) Distribution of mean deviation of nodal displacement in the Z direction when applying force in the Z direction.



**Fig. 14.** Fitting effect.

placements are fitted as follows:

$$\begin{cases} \Delta Z_{61} = \alpha_1 + \alpha_2 F_z \\ \Delta X_{61} = r_1 + r_2 F_x \\ \Delta Y_{61} = s_1 + s_2 F_y \end{cases} \quad (49)$$

where  $\Delta Z_{61}$ ,  $\Delta X_{61}$ , and  $\Delta Y_{61}$  are the component displacements of Node 61.

The coefficients in the above equation are:

Fitting coefficients			
$\alpha_1$	0.000820000000001228	$\alpha_2$	-14.5441
$r_1$	0.000476190443162377	$r_2$	-2.82928571424066
$s_1$	0.000476190443162377	$s_2$	-2.82928571424066

#### 4.2.2. Surface deformation equation

When a spatial force of changing strength (within the range of 0–1 N) and direction is applied to Node 61, the displacements all 121 nodes are computed by RPIM-MM, and the experiment is repeated 100 times to generate 100 displacement datasets for greater accuracy. These datasets are fitted by Marquardt's Algorithm to produce the surface deformation equation as shown in Table 1 and 2.

**Table 1**

The fitted surface equation and parameters  $\beta_1$ ,  $\beta_2$  under the action of component force in direction X.

Coefficients of surface function	$\beta_1$	$\beta_2$	Generating coefficients
$p_{x1}$	-9.66E-07	0.1384048	$p_{xi} = \beta_1 + \beta_2 \Delta x_{61} \quad i=1,2,\dots,n_x+m_x$
$p_{x2}$	-9.60E-06	-0.001407	
$p_{x3}$	9.50E-07	0.0010891	
$p_{x4}$	-2.99E-08	-4.42E-05	
$p_{x5}$	4.22E-10	4.60E-07	
$p_{x6}$	4.92E-06	0.0068369	
$p_{x7}$	-4.65E-07	0.0010409	
$p_{x8}$	1.18E-08	-4.93E-05	
$p_{x9}$	-1.69E-10	5.14E-07	
Surface deformation fitting equation in X direction: $\Delta x = p_{x1} + p_{x2}x + p_{x3}x^2 + p_{x4}x^3 + p_{x5}x^4 + p_{x6}y + p_{x7}y^2 + p_{x8}y^3 + p_{x9}y^4$			
Surface deformation fitting equation in Y direction: $\Delta y = 0$ . The displacement in the Y direction is very small relative to the X and Z directions and is possibly negligible.			
$p_{z1}$	-8.08E-06	0.1875753	$p_{zi} = \beta_1 + \beta_2 \Delta z_{61} \quad i=1,2,\dots,n_z+m_z$
$p_{z2}$	9.97E-07	0.0090317	
$p_{z3}$	-4.15E-08	-0.001053	
$p_{z4}$	5.72E-10	1.46E-05	
$p_{z5}$	-1.06E-09	-1.13E-10	
Surface deformation fitting equation in Z direction: $\Delta z = p_{z1} + p_{z2}x + p_{z3}x^2 + p_{z4}x^3 + p_{z5}y$			

**Table 2**

The fitted surface equation and parameters  $\beta_1$ ,  $\beta_2$  under the action of component force in direction Z.

Coefficient of surface function	$\beta_1$	$\beta_2$	Generating coefficients
$p_{x1}$	5.12E-06	-0.06086	$p_{xi} = \beta_1 + \beta_2 \Delta z_{61} \quad i=1,2,\dots,n_x+m_x$
$p_{x2}$	9.59E-07	-0.00334	
$p_{x3}$	2.50E-07	8.48E-05	
$p_{x4}$	-3.43E-08	1.84E-05	
$p_{x5}$	4.91E-10	-6.13E-07	
$p_{x6}$	-1.72E-12	5.11E-09	
$p_{x7}$	6.83E-09	1.63E-12	
Surface deformation fitting equation in X direction: $\Delta x = p_{x1} + p_{x2}x + p_{x3}x^2 + p_{x4}x^3 + p_{x5}x^4 + p_{x6}x^5 + p_{x7}y$			
$p_{y1}$	7.42E-06	-5.26E-02	$p_{yi} = \beta_1 + \beta_2 \Delta z_{61} \quad i=1,2,\dots,n_y+m_y$
$p_{y2}$	1.52E-08	3.61E-12	
$p_{y3}$	-1.12E-05	-2.96E-03	
$p_{y4}$	2.81E-06	-2.49E-05	
$p_{y5}$	-1.75E-07	2.44E-05	
$p_{y6}$	5.16E-09	-7.52E-07	
$p_{y7}$	-4.16E-11	6.27E-09	
Surface deformation fitting equation in Y direction: $\Delta y = p_{y1} + p_{y2}y + p_{y3}y^2 + p_{y4}y^3 + p_{y5}y^4 + p_{y6}y^5 + p_{y7}x$			
$p_{z1}$	-2.05E-04	-7.96E-03	$p_{zi} = \beta_1 + \beta_2 \Delta z_{61} \quad i=1,2,\dots,n_z+m_z$
$p_{z2}$	-7.46E-06	1.94E-02	
$p_{z3}$	2.20E-06	1.23E-03	
$p_{z4}$	-8.25E-08	-6.83E-05	
$p_{z5}$	9.93E-10	7.12E-07	
$p_{z6}$	-2.57E-05	1.67E-02	
$p_{z7}$	3.01E-06	1.27E-03	
$p_{z8}$	-1.13E-07	-6.73E-05	
$p_{z9}$	6.55E-10	7.01E-07	
Surface deformation fitting equation in Z direction: $\Delta z = p_{z1} + p_{z2}x + p_{z3}x^2 + p_{z4}x^3 + p_{z5}x^4 + p_{z6}y + p_{z7}y^2 + p_{z8}y^3 + p_{z9}y^4$			

**Table 3**

Root mean square error of node displacements with units mm/N.

Force	Error		
	RMS error of node displacements in X coordinate	RMS error of node displacements in Y coordinate	RMS error of node displacements in Z coordinate
Force applied in the X direction	0.081713	0.084872	0.262261
Force applied in the Y direction	0.077761	0.099467	0.291453
Force applied in the Z direction	0.094042	0.065811	0.155073

For the Y component of planar displacement  $\Delta y_{61}$ ,  $\Delta x=0$ ,  $\Delta y$  has the coefficients of  $\Delta x$  in [Table 1](#), and  $\Delta z$  has the same coefficients as  $\Delta z$  in [Table 1](#).

The relation between the vertical component of original displacement ( $\Delta z_{61}$ ) and the surface equations are shown below.

## 5. Analysis and application of experimental results

### 5.1. Error and run-time analysis

#### 5.1.1. Analysis of fitting error

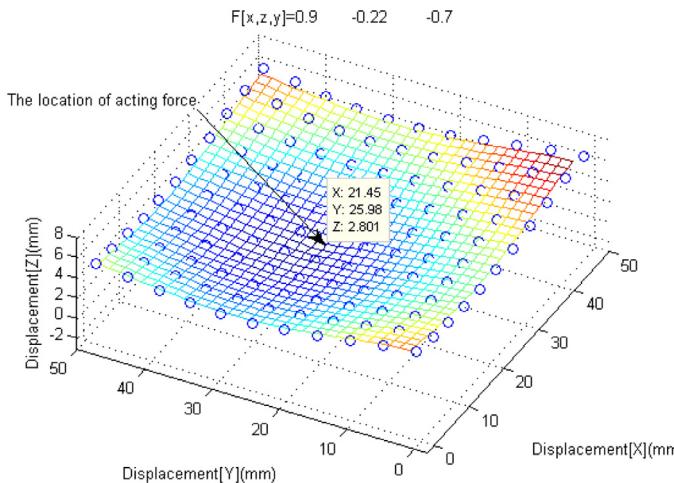
We applied 0.2, 0.4, 0.6, 0.8, and 1.0 N forces to Node 61 in each of the X, Y, and Z directions and calculated the surface dis-

### The configuration of test computer:

Modules	Type
CPU	IntelCore i7-4790 @ 3.60GHz
Memory	8 GB (DDR3L 1600MHz)
Video card	Intel HD Graphics 4600 (64 MB)

Figure 15 consists of two line graphs. The left graph shows 'Time(s)' on the y-axis (ranging from 620 to 680) versus 'Random Force' on the x-axis (ranging from 0 to 20). It contains two series: 'RPIM' (red circles) and 'RPIM-MM' (blue circles). The right graph shows 'Time(s)' on the y-axis (ranging from 0.144 to 0.158) versus 'Random Force' on the x-axis (ranging from 0 to 20). It contains two series: 'New model' (blue circles) and 'Marquardt's algorithm' (green circles).

**Fig. 15.** Comparison of time needed to calculate node displacements by RPIM-MM (left) and Marquardt's algorithm (right).



**Fig. 16.** Displacement of Node 61 obtained by our model.

placement by RPIM. The mean deviations under the action of the 5 forces are shown in Figs. 5, 8, and 11.

As shown in red for Fig. 5, the deviation of nodal displacement in the X direction ranges from -0.24 mm to 0.1003 mm. As shown in green for Fig. 5, the deviation of nodal displacement in the Y direction ranged between -0.121 mm and 0.124 mm. Such a small deviation confirms our decision to set  $\Delta y$  to be negligible. In both X and Y directions, the deviation was minimized in regions close to the central node. In the Z direction, the node displacement was large and showed an obvious regularity due to no constraint on the upper surface (see Fig. 6). The fitting equation (see Table 1) for the Z direction could fit such deformation only up to a certain degree, resulting in regularly distributed mean deviations ranging from -0.3477 mm to 0.3477 mm. Some of the large fitting error is because the corner nodes (1, 11, 111, 121) did not fully participate in the fitting (see Figs. 4 and 7a) due to fixed constraints, resulting

in large deviations of nearby nodes. A more visual approach to the data presented in Fig. 5 is given in Fig. 7a–c.

The same periodic pattern of surface nodal displacement was observed when the direction of spatial force was changed from X to Y. However, the change in direction changes the overall pattern of displacement, as reflected by the differences between Figs. 8 and 5. In the X direction, the deviations are minimized in regions close to the central node with range of -0.1088 mm to 0.1081 mm. The red X-direction curve in Fig. 8 matches the overall shape of the green Y-direction curve in Fig. 5, which supports the assumption that  $\Delta x$  is negligible. In the Y direction, the deviations ranged from -0.24 mm to 0.1102 mm. Since this is almost exactly the same range as the deviation in the X direction for Fig. 5, this supports the assumption that  $\Delta y$  has the same coefficients of fitting as  $\Delta x$  in Table 1. In the Z direction, the deviations ranged from -0.3989 mm to 0.3989 mm and were in regular distribution. As before, some of the large fitting error is due to the fixed constraints on corner nodes. A more visual approach to the data presented in Fig. 8 is given in Fig. 10a–c. The similar shape between 10a and 7b and 10b and 7a reflect the similarity of their fitting parameters.

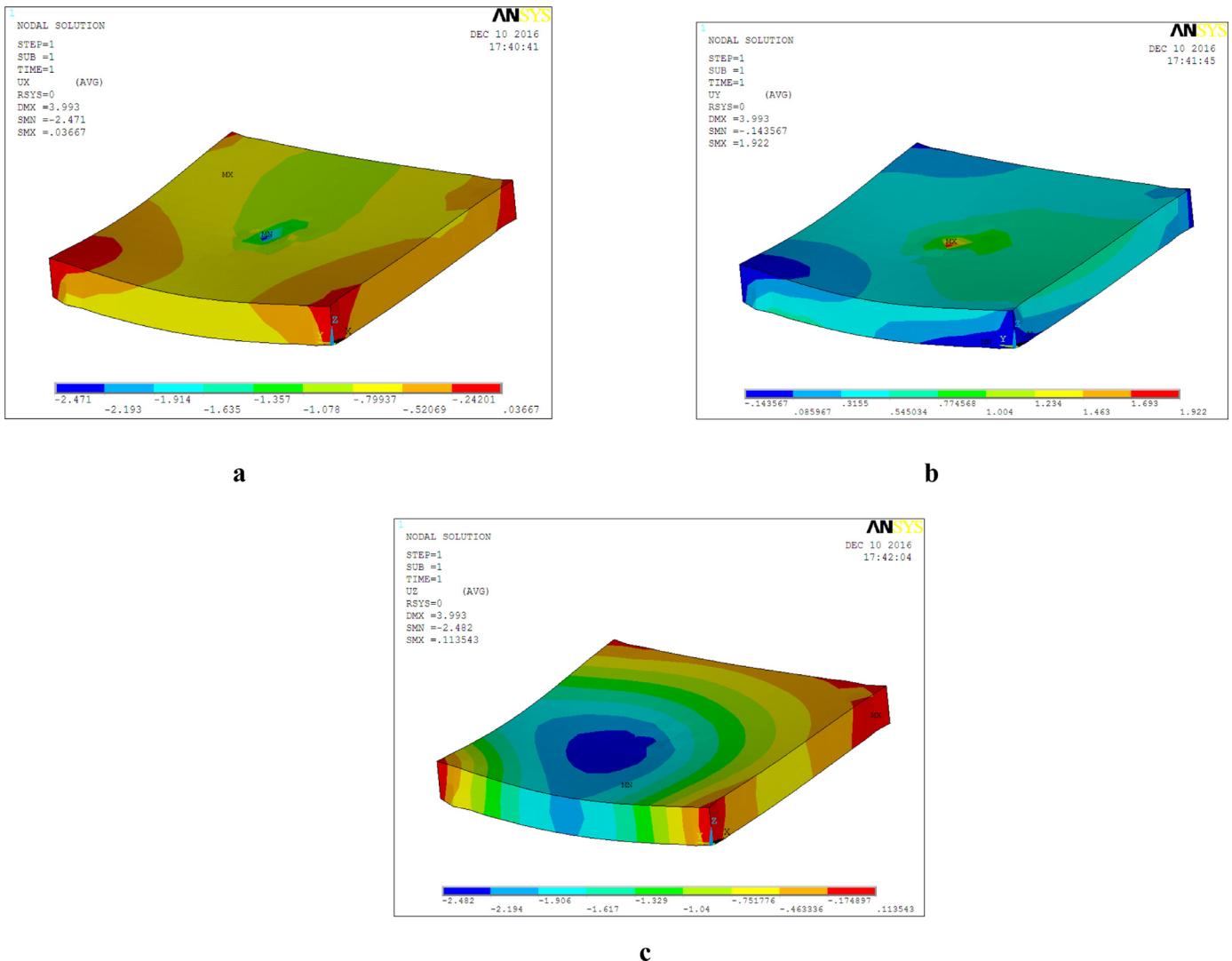
Compared to either X or Y direction spatial forces, Z direction forces have much lower deviations of displacement (see Fig. 12), which imply that the fit is better. Furthermore, unlike X or Y direction spatial forces, the deviation is minimized in regions closest to the center for all three directions of displacement. The large deviations observed near the edges in the red (X) and green (Y) curves (see Fig. 12) are due to corner nodes. The deviation in the Z direction is the lowest, ranging from -0.1245 mm to 0.2534 mm, whereas the deviations in the X direction ranged from -0.4989 mm to 0.5211 mm, and the deviations in the Y direction ranged from -0.4504 mm to 0.4586 mm. A more visual approach to the data presented in Fig. 11 is given in Fig. 13a–c. In order to assess the accuracy of the overall fitting, we computed the root mean square errors of the displacements all 121 nodes. The results are given in Table 3. For a spatial force in any given direction, the largest error was always in the Z direction. To better simulate the practical use of the model, the action of a composite spatial force with component forces of random magnitude was modeled. The results are shown in Fig. 14. For better accuracy, we directly substituted the displacement computed by the equation (Eq. (49)) for the displacements  $\Delta x_{61}$ ,  $\Delta y_{61}$ , and  $\Delta z_{61}$ . Despite the rather large Z errors (with range from 0.15 to 0.3), the fitting error was visually negligible relative to the overall deformation of the model.

#### 5.1.2. Run-time comparison

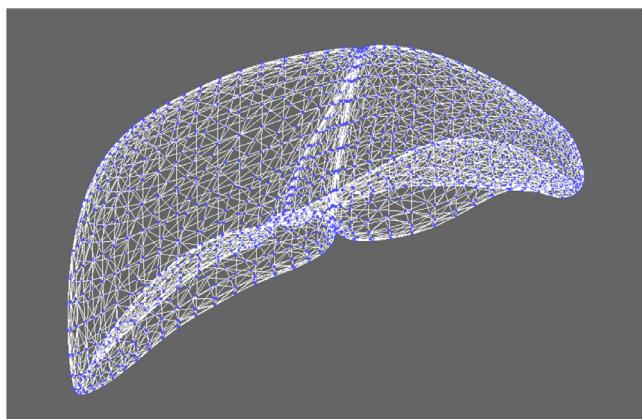
To test the real-time aspect of our model, we randomly selected 20 sets of spatial forces ranging from 0–1 N, comparing the time needed for a normal RPIM-MM algorithm and our method to calculate the resultant deformation. The mean time for the normal RPIM-MM algorithm was 647.4073 s while our method using Marquardt's algorithm had a mean time of 0.1509 s. The corresponding run-time results are shown in Fig. 15. Despite the difference in coding language (the RPIM-MM algorithm was written in Fortran and our algorithm was written in Matlab), the thousand-fold improvement in run-time clearly demonstrates the superiority of our model or method.

The configuration of test computer:

Modules	Type
CPU	IntelCore i7-4790 @ 3.60 GHz
Memory	8GB (DDR3L 1600 MHz)
Video card	Intel HD Graphics 4600 (64 MB)



**Fig. 17.** (a) X displacement of central force bearing point, as obtained by ANSYS; (b) Y displacement of central force bearing point, as obtained by ANSYS; (c) Z displacement of central force bearing point, as obtained by ANSYS.



**Fig. 18.** Triangular patch made up of nodes.

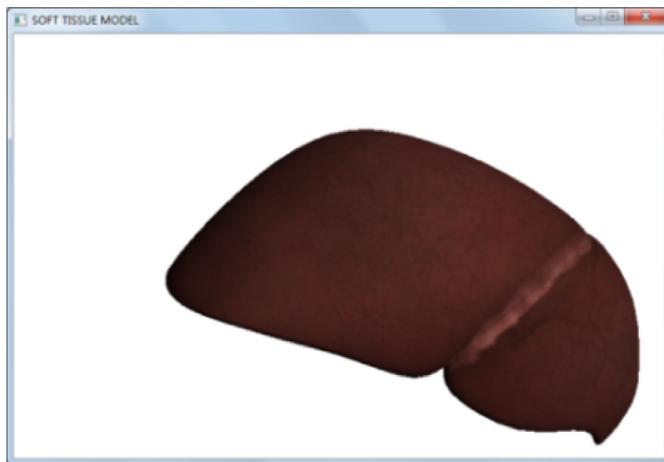
## 5.2. Validation by ANSYS

We used ANSYS, a standard modeling software, to validate our model's accuracy. In order to simplify the problem, both our model and ANSYS adopted an elastic modulus of 0.32 MPa and a Poisson's

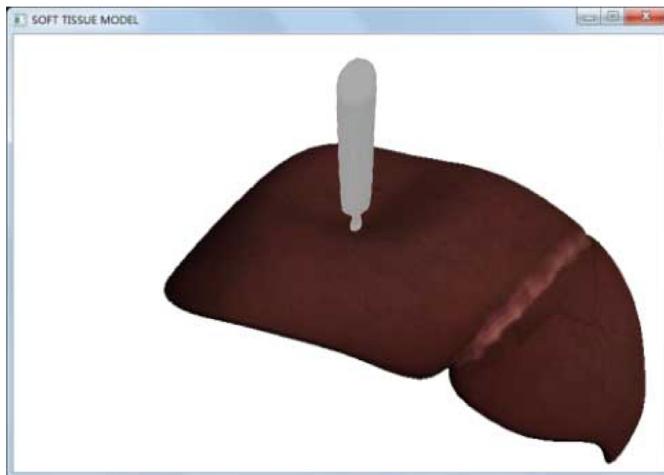
ratio of 0.49. A force is applied to the center of the model surface. In the test, we applied a force with component magnitudes (0.9, -0.7, -0.22). In general, the deformation at the force bearing point was the most noticeable and thus was chosen as the point of comparison. In our model's simulation (Fig. 16), the coordinates of the central point after the application of force are (21.45, 25.98, 2.801), meaning the displacement is (-2.55, 1.98, -3.199). In the ANSYS simulation (Fig. 17a-c), the displacement is (-2.471, 1.922, -2.482). Using the ANSYS model as the reference, the relative error as an ordered triple is (3.1%, 3%, 28%), which matches the error trends calculated in Table 3. In practice, the large Z-direction relative error can be reduced by adjusting the parameters in the radial basis function (refer to  $ac$ ,  $dc$ , and  $q$ ) to recalculate the node's displacement in a new fitting equation.

## 5.3. Simulation of liver deformation

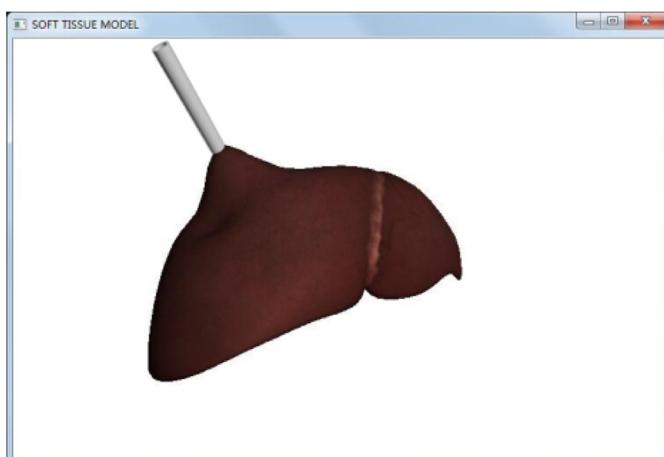
To simulate deformation of the liver by our soft tissue model, we consider the surface of the liver as a problem domain scattered with nodes whose displacements can be computed by the meshless method. Applying our model, we establish a fitting relation between force and liver surface deformation. When different forces are applied to node(s), the original and induced displacements can



**Fig. 19.** Rendered image before the application of force. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).



**Fig. 20.** Rendered image of deformation from pressed nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).



**Fig. 21.** Rendered image of deformation from stretched nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

be obtained through this relationship. In the liver model, there are 1986 surface nodes arranged in a triangular patch (Fig. 18), which is then rendered in OpenGL (Fig. 19) to obtain a liver morphology. The deformation of the liver under various types of force is shown in Figs. 20 and 21.

## 6. Conclusion and future work

In this paper, we fitted nodal displacements computed by the meshless method augmented with Marquardt's algorithm to determine the relationship between force and the deformed surface in a simulation of soft tissue deformation. This improves the run-time enough to allow for real time simulations. Experimental analysis of the model shows that the error and run-time meets the visual requirements for use in virtual surgery. In addition, the model was validated by the current standard modeling software, which demonstrates the accuracy of our model for general modeling of soft tissue. In the future, we would like to use this method to analyze the anisotropic performance of simulated soft tissue and study tissue cutting.

## Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and are grateful to G.R.Liu and Y.T.Gu for writing the code for the meshless method used in our work.

## Funding

The work was supported by a grant from the National High Technology Research and Development Program of China (863 Program) (No. 2013AA013804). And this work was also supported by the Science and Technology Department of Jiangxi Province of China under Grants 20161ACB21007, 20171BBE50071, and 20171BAB202033, and by the Education Department of Jiangxi province of China under Grant GJJ14228.

## Competing interests

The authors declare that they have no conflict of interest.

## References

- [1] S.-M. Cristina, C.G. Gómez, C.C. Parra, Virtual reality simulation training and assisted surgery: AYRA - virtual and physical biomodels in surgery, in: Proceedings of the 2012 18th international conference on virtual systems and multimedia, VSMM 2012: Virtual Systems in the Information Society, 2012, doi:10.1109/VSMM.2012.636595.
- [2] F. Te-Yung, W. Pa-Chun, Evaluation of a haptics-based virtual reality temporal bone simulator for anatomy and surgery training, Comput. Methods Progr. Biomed. 113 (2014) 674–681.
- [3] L.C. Kiang, S. Kenneth, Computer-based virtual reality simulator for phacoemulsification cataract surgery training, Virtual Reality 18 (2014) 281–293.
- [4] H. Pheng-Ann, C. Chun-Yiu, A virtual-reality training system for knee arthroscopic surgery, IEEE Trans. Inf. Technol. Biomed. 8 (2004) 217–227.
- [5] C. Teeranoot, Modeling soft-tissue deformation prior to cutting for surgical simulation: Finite element analysis and study of cutting parameters, IEEE Trans. Biomed. Eng. 54 (2007) 349–359.
- [6] H. Dongjin, T. Wen, D. Youdong, W. Taoruan, C. Yimin, An interactive 3D preoperative planning and training system for minimally invasive vascular surgery, in: Proceedings-12th international conference on computer-aided design and computer graphics, CAD/Graphics 2011, 2011, doi:10.1109/CAD/Graphics.2011.40.
- [7] S.V. Gaizka, A. Iker, C.J. Tomas, Cubical mass-spring model design based on a tensile deformation test and nonlinear material model, IEEE Trans. Visual. Comput. Graph. 18 (2012) 228–241.
- [8] W. Shuguo, C. Lili, An unfixed-elasticity mass spring model based simulation for soft tissue deformation, IEEE Int. Conf. Mechatron. Autom. (2014), doi:10.1109/ICMA.2014.6885714.
- [9] T. Halic, S. Kockara, Soft tissue deformation and optimized data structures for mass spring methods, in: Proc. IEEE in. conf bioinformatics BioEng, 2009, doi:10.1109/BIBE.2009.64.

- [10] A. Bummo, K. Jung, Dynamic measurement and modeling of soft tissue behavior with an indentation device using indenters of various shapes, *Key Eng.* 326–328 (2006) 781–784.
- [11] S. Xu, X.P. Liu, H. Zhang, L. Hu, A nonlinear viscoelastic tensor-mass visual model for surgery simulation, *IEEE Trans. Instrum. Measur.* 60 (2011) 14–20.
- [12] U. Meier, O. López, C. Monserat, Real-time deformable models for surgery simulation, *Comput. Methods Progr. Biomed.* 77 (2005) 183–197.
- [13] K. Waters, A physical model of facial tissue and muscle articulation derived from computer tomography data, *Visualization in Biomedical Computing (VBC'92)*, vol.574, Chapel Hill, N.C., 1992.
- [14] S.D. Bona, L. Ludovico, A simulation model for analyzing brain structure deformations, *Phys. Med. Biol.* 48 (2003) 4001–4022.
- [15] D. Bourguignon, C. Marie-paule, Controlling anisotropy in mass-spring systems, 11th eurographics workshop in interlaken, 2000, doi:[10.1007/978-3-7091-6344-3\\_9](https://doi.org/10.1007/978-3-7091-6344-3_9).
- [16] X.P. Liu, S. Xu, H. Zhang, L. Hu, A new hybrid soft tissue model for visio-haptic simulation, *IEEE Trans. Instrum. Measur.* 60 (2011) 3570–3581.
- [17] C. Monserat, U. Meier, M. Alcaiz, F. Chistea, M.C. Juan, A new approach for the real-time simulation of tissue deformations in surgery simulation, *Comput. Methods Progr. Biomed.* 64 (2001) 77–85.
- [18] P. Patete, M.I. Iacono, M.F. Spadea, et al., A multi-tissue mass-spring model for computer assisted breast surgery, *Med. Eng. Phys.* 35 (2013) 47–53.
- [19] A.V. Gelder, Approximate simulation of elastic membranes by triangulated spring meshes, *J. Graph. Tools* 3 (1998) 21–41.
- [20] W. Mollemans, F. Schuttyser, Van Cleynenbreugel J, Tetrahedral mass spring model for fast soft tissue deformation, *Lect. Notes Comput. Sci.* 2673 (2003) 145–154.
- [21] Y. Guisheng, L. Yanbo, Soft tissue modeling using tetrahedron finite element method in surgery simulation, *Int conf inf sci eng*, 2009, doi:[10.1109/ICISE.2009.1071](https://doi.org/10.1109/ICISE.2009.1071).
- [22] G. Zayang, Y. Shihui, A FEM-based direct method for material reconstruction inverse problem in soft tissue elastography, *Comput. Struct.* 88 (2010) 1459–1468.
- [23] L. XiangYun, Y. ZhiYong, Soft tissue parameter measurement based on pressure acquisition and FEM model, *J. Northeastern. Univ.* (2015), doi:[10.3969/j.issn.1005-3026.2015.09.007](https://doi.org/10.3969/j.issn.1005-3026.2015.09.007).
- [24] S. Cotin, H. Delingette, Real-time elastic deformations of soft tissues for surgery simulation, *IEEE Trans. Visual. Comput. Graph.* 5 (1999) 62–73.
- [25] Q.L. Zhu, J.K. Francis, Dynamic biphasic poroviscoelastic model simulation of hydrated soft tissues and its potential applications for brain impact study, *ASME Bioeng. Div. Publ. BED* 50 (2001) 835–836.
- [26] S. Pieper, J. Rosen, Interactive graphics for plastic surgery: a task-level analysis and implementation, *Comput. Graph.* 25 (1992) 127–134.
- [27] S. Sascha, B. Oliver, Deformable modeling of the cervical spine for neurosurgical navigation, *Int. Congr. Ser.* 1268 (2004) 455–460.
- [28] L. Mao, M. Karol, Biomechanical model for computing deformations for whole-body image registration: a meshless approach, *Int. J. Numer. Methods Biomed. Eng.* (2016), doi:[10.1002/cnm.2771](https://doi.org/10.1002/cnm.2771).
- [29] Y. Zou, P.X. Liu, Q. Cheng, P. Lai, C. Li, A new deformation model of biological tissue for surgery simulation, *IEEE Trans. Cybern.* (2016), doi:[10.1109/TCYB.2016.2560938](https://doi.org/10.1109/TCYB.2016.2560938).
- [30] X. Shaojian, C. Guodon, Research of liver deformation based on moving least square of meshless method, *Comput. Simul. China* 32 (2015) 408–412.
- [31] L. Zhu, Y. Yu, An improved meshless method in virtual surgery simulation, in: *Proceedings - 2014 international conference on virtual reality and visualization*, 2014, doi:[10.1109/ICVRV.2014.49](https://doi.org/10.1109/ICVRV.2014.49).
- [32] Z. Zhang, K.M. Liew, Y. Cheng, Analyzing 2D fracture problems with the improved element-free Galerkin method, *Eng. Anal. Boundary Elem.* (2008), doi:[10.1016/j.enganabound.2007.08.012](https://doi.org/10.1016/j.enganabound.2007.08.012).
- [33] P. Krysl, T. Belytschko, Analysis of thin shells by the element-free Galerkin method, *Int. J. Solids Struct.* 33 (1996) 3057–3080.
- [34] K. Areti, H. Kelli, SPH for incompressible free-surface flows. Part II: performance of a modified SPH method, *Comput Fluids* 86 (2013) 510–536.
- [35] K. Scott, M. Gao, Z. Shaoting, Practical patient-specific cardiac blood flow simulations using SPH, in: *Proceedings - international symposium on biomedical imaging*, 2013, doi:[10.1109/ISBI.2013.6556604](https://doi.org/10.1109/ISBI.2013.6556604).
- [36] C.M. Tiago, V.M.A. Leito, Application of radial basis functions to linear and non-linear structural analysis problems, *Comput. Math. Appl.* 51 (2006) 1311–1334.
- [37] F.F. Yan, X. Ting, Z. Hui, A dual reciprocity hybrid radial boundary node method based on radial point interpolation method, *Comput. Mech.* 45 (2010) 541–552.
- [38] G.R. Liu, Y.T. Gu, *An introduction to meshfree methods and their programming*, 1st ed., Jinan: Shandong University Press, China, 2007.
- [39] L. Yong, L. Shizhong, Preventing over-fitting of cross-validation with kernel stability, *Lect. Notes Comput. Sci.* (2014) 290–305 8725 LNAI.
- [40] E. Samur, M. Sedef, C. Basdogan, L. Avtan, O. Duzgun, A robotic indenter for minimally invasive measurement and characterization of soft tissue response, *Med. Image Anal.* 11 (2007) 361–373.
- [41] E.H. Lee, J.R.M. Radok, The contact problem for viscoelastic bodies, *J. Appl. Mech.* 27 (1996) 438–444.