Inverse Nonlinear Finite Element Methods for Surgery Simulation and Image Guidance

Philip Pratt
p.pratt@imperial.ac.uk

May 18, 2008

Institute of Biomedical Engineering
Imperial College, London

Abstract

Nonlinear finite element methods are described in which cyclic organ motion is implied from 4D scan data. The equations of motion corresponding to an explicit integration of the total Lagrangian formulation are reversed, such that the sequence of node forces which produces known changes in displacement is recovered. The forces are resolved from the global coordinate system into systems local to each element, and at every simulation time step are expressed as weighted sums of edge vectors. In the presence of large deformations and rotations, this facilitates the combination of external forces, such as tool-tissue interactions, and also positional constraints. Applications in the areas of surgery simulation and minimally invasive robotic interventions are discussed, and the methods are illustrated using CT images of a pneumatically-operated beating heart phantom.

1 Introduction

The future of minimally invasive robotic surgery lies not only in the mechanical evolution of better telemanipulator systems, but also in the development of advanced software tools that facilitate surgical training, patient-specific intraoperative rehearsal, and the seamless integration of preoperative and intraoperative imaging, of various modalities, through augmented reality techniques. The image-constrained biomechanical modelling (ICBM) approach is a key technology which promises to realise the goal of allowing surgeons to alternate between full surgical simulation, endoscopic views enhanced in real-time through the same simulation constrained by imaging data, and completion of the intervention itself. To that end, this paper extends previous work [6] by showing how intrinsic cyclic tissue motion can be inferred from 4D scan data and combined with externally induced motion and other constraints using a *nonlinear* finite element model. Taking known changes in node displacements over time, the finite element model inverted such that the sequence of node forces responsible for the motion can be recovered. These forces are then resolved from the global coordinate system into systems local to each element, thereby expressing them in terms of local geometry. That way, in the presence of large deformations and rotations, external forces and other constraints can be combined when forward simulation is performed.

Accurate modelling of soft tissue deformation represents a significant challenge, since the constitutive behaviour of such tissue is known to be both nonlinear and time-varying, and the assumptions made in linear,

small-strain formulations are not valid, particularly when considering large deformations. Miller *et al.* [5] present the total Lagrangian explicit dynamics (TLED) algorithm, which offers the possibility of accurate simulation at interactive rates. In contrast to the *updated* Lagrangian finite element formulation, the *total* Lagrangian formulation [1, 10] expresses stress and strain measures in terms of the reference configuration, and thus many quantities can be either completely or partially precomputed. Furthermore, an explicit integration scheme coupled with element and node-wise storage enable efficient implementation, particularly when GPU hardware [9] is employed. Several inverse finite element simulations have been described in the literature where deformations are known *a priori*. In particular, Kruggel and Tittgemeyer [4] use an inverse finite element model of linear elasticity to derive a force field given an observed deformation of the brain. Kauer [3] also uses an inverse model to calibrate the properties of a visco-elastic material given experimental pressure data and resulting tissue deformations.

2 Methods

The tetrahedral finite element mesh comprises N nodes, and therefore has at most $3N$ degrees of freedom. In general, the equations of motion are expressed in terms of the $3N$ displacements from the initial mesh configuration, i.e. $\mathbf{U} = [u^0\, u^1 \ldots u^{3N-1}]^\top$, and following the notation of Bathe [1], are written in semi-discrete form as

$$\mathbf{M}\,{}^t\ddot{\mathbf{U}} + \mathbf{C}\,{}^t\dot{\mathbf{U}} + {}^t_0\mathbf{F} = {}^t\mathbf{R} \tag{1}$$

where $\mathbf{U}$, ${}^t\dot{\mathbf{U}}$ and ${}^t\ddot{\mathbf{U}}$ are the displacement, velocity and acceleration vectors, respectively, $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, ${}^t_0\mathbf{F}$ is the vector of nodal reaction forces equivalent to the element stresses, and ${}^t\mathbf{R}$ is the vector of externally applied, time-varying forces. The damping matrix is assumed to be proportional to the mass matrix, i.e. $\mathbf{C} = \alpha\mathbf{M}$, where α is the damping coefficient. The mass matrix is assumed to be constant, and is diagonalised to facilitate explicit integration.

2.1 Total Lagrangian Formulation

In the total Lagrangian formulation of the finite element method, quantities are expressed in terms of the reference configuration. Considering an individual element i, the nodal reaction forces are computed as an integral over the element volume, as follows

$${}^t_0\mathbf{F}^{(i)} = \int_{{}^0V^{(i)}} {}^t_0\mathbf{B}_L^\top\, {}^t_0\hat{\mathbf{S}}\; d^0V \tag{2}$$

where ${}^t_0\mathbf{B}_L$ is the full strain-displacement matrix and ${}^t_0\hat{\mathbf{S}}$ is the vector of 2$^{\text{nd}}$ Piola-Kirchoff stresses. The latter depend on the element deformation and the choice of material constitutive law. For an assemblage of elements, the nodal reaction forces are accumulated in accordance with the mesh's element-node relationships.

2.2 Explicit Central Difference Integration

The motion from which forces are to be inferred is assumed to be cyclic, spanning a period of T seconds. Successor and predecessor functions of time t are defined, using time step duration Δt, as follows.

$$next(t) = \begin{cases} t + \Delta t & \text{if } t < T - \Delta t \\ 0 & \text{otherwise} \end{cases} \qquad prev(t) = \begin{cases} t - \Delta t & \text{if } t > 0 \\ T - \Delta t & \text{otherwise} \end{cases} \tag{3}$$

From these definitions, the central finite-difference approximations to the first and second-order time derivatives of the displacement vector yield the following expressions for velocity and acceleration.

$$ {}^{t}\dot{\mathbf{U}} \approx \tfrac{1}{2\Delta t}\left[{}^{next(t)}\mathbf{U} - {}^{prev(t)}\mathbf{U}\right] \qquad {}^{t}\ddot{\mathbf{U}} \approx \tfrac{1}{\Delta t^2}\left[{}^{next(t)}\mathbf{U} - 2\,{}^{t}\mathbf{U} + {}^{prev(t)}\mathbf{U}\right] \tag{4} $$

Thus, over a single cycle, the fully discretised equations of motion take the form shown in (5). Note that the nodal reaction forces must be calculated at every time step. At the expense of some restriction on time step magnitude, the explicit scheme avoids the iterative solution of the displacements at the next time step, which would otherwise be extremely computationally expensive.

$$ {}^{next(t)}\mathbf{U} \approx \frac{2}{2+\alpha\Delta t}\left[\frac{\Delta t^2}{\mathbf{M}}({}^{t}\mathbf{R} - {}^{t}_{0}\mathbf{F}) + 2\,{}^{t}\mathbf{U} - (\frac{\alpha\Delta t}{2} - 1)\,{}^{prev(t)}\mathbf{U}\right] \tag{5} $$

Over multiple cycles, this displacement update rule is extended in the conventional manner.

2.3 Recovering Forces from Displacements

By inverting equation (5), one can write the discretised applied force in terms of the displacement, velocity and acceleration vectors, the nodal reaction forces, and other known quantities. By construction, if one were then to solve the equations of motion and apply these forces at the appropriate times, one would recover the original cyclic motion exactly and indefinitely. Note that the recovered forces are expressed in the global coordinate system.

The force recovery process, and therefore also the forward simulation, are initialised by precomputing the spatial derivatives of element shape functions, the element Jacobian determinants, constant strain-displacement matrices ${}^{t}_{0}\mathbf{B}_{L0}$, and the diagonalised mass matrix. Furthermore, deformation gradient tensors can be factorised into two parts, depending on global node positions and displacements, respectively, and the former can also be precomputed for efficiency. Subsequently, the following two-stage calculation is performed at each simulation time step.

For each integration point in each element:

- Calculate deformation gradient tensor ${}^{t}_{0}\mathbf{X}$
- Calculate strain-displacement matrix ${}^{t}_{0}\mathbf{B}_{L} = {}^{t}_{0}\mathbf{B}_{L0}\,{}^{t}_{0}\mathbf{X}^{\top}$
- Calculate 2nd Piola-Kirchoff stress vector ${}^{t}_{0}\hat{\mathbf{S}}$
- Accumulate element nodal reaction forces ${}^{t}_{0}\mathbf{F}^{(i)}$ to give node totals ${}^{t}_{0}\mathbf{F}$

For each node:

- Invert displacement update step to recover external node force:

$$ {}^{t}\mathbf{R} \approx \frac{\mathbf{M}}{\Delta t^2}\left[(1+\frac{\alpha\Delta t}{2})\,{}^{next(t)}\mathbf{U} - 2\,{}^{t}\mathbf{U} + (1-\frac{\alpha\Delta t}{2})\,{}^{prev(t)}\mathbf{U}\right] + {}^{t}_{0}\mathbf{F} \tag{6} $$

For forward simulation, the node-wise update stage is replaced by the following:

- Determine external forces to be applied to each node (e.g. due to virtual tool-tissue interaction)
- Add to the recovered forces and use equation (5) to update displacements
- Apply displacement constraints (e.g. anchor points)

2.4 Local Force Resolution

In order to combine recovered and external forces, the former must be expressed not in the global coordinate system, but for each node in terms of its local surrounding geometry. With the introduction of external forces, the geometry may deviate from the original cyclic motion through potentially large-scale deformations and rotations. By resolving recovered forces locally, they are made to act in the appropriate direction in conjunction with externally induced motion.

The recovered force acting on a particular node in the mesh is assumed to originate from the elements which contain that node. Indeed, an approximation is made whereby the force receives an equal contribution from each such element. For a given element at each point in time, the edge vectors from the node in question to the other three nodes in that element define a local basis in terms of which that element's fraction of the node force can be expressed. This amounts to equating the force to a weighted sum of those edge vectors and solving for the weights. Subsequently, the weights are further computed over the node's other parent elements, ultimately building a set of weights that links all the recovered forces locally to the geometry of the entire mesh.

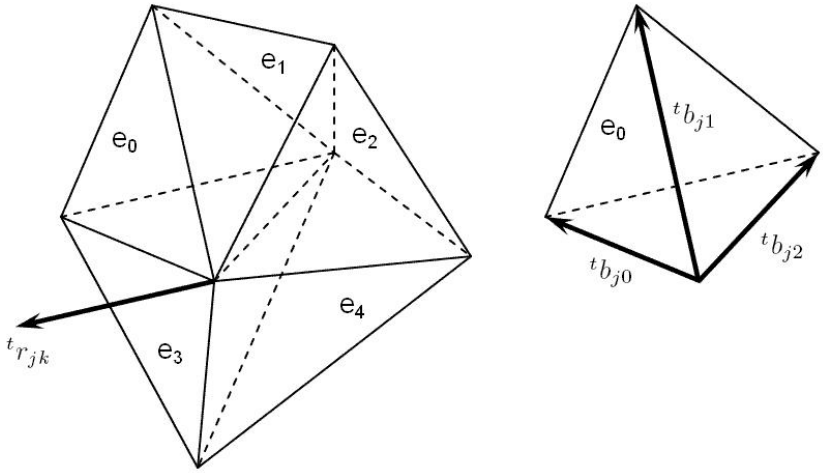

Figure 1: Force resolution using local geometry

Labelled with coordinate indices j, figure 1 (left) depicts at time t a typical node k with its recovered force $^{t}r_{jk}$, and the five surrounding elements $e_0, \ldots, e_4$ to which it belongs. In general, a node will be common to M_k elements. Figure 1 (right) illustrates the first element e_0, and the three edge vectors $^{t}b_{j0}$, $^{t}b_{j1}$ and $^{t}b_{j2}$ which, with the node itself, define the geometry of the element at that instant. In order to express the required fraction of the force in terms of local mesh geometry, its components in the global coordinate system are equated to weighted combinations of these edge vectors, as shown in (7).

The weights are determined by direct inversion of the left-hand-side matrix. This process is repeated for the element's other three nodes, and subsequently over all nodes in the mesh. In order to ensure that the magnitudes of the recovered forces, expressed as functions of current element edge vectors, remain within reasonable bounds and do not give rise to simulation instability, recovered forces are normalised at each time step to have the same magnitude as those implied from the original motion where no externally applied forces are present.

$$\begin{bmatrix} {}^{t}b_{00} & {}^{t}b_{01} & {}^{t}b_{02} \\ {}^{t}b_{10} & {}^{t}b_{11} & {}^{t}b_{12} \\ {}^{t}b_{20} & {}^{t}b_{21} & {}^{t}b_{22} \end{bmatrix} \begin{bmatrix} {}^{t}w_0 \\ {}^{t}w_1 \\ {}^{t}w_2 \end{bmatrix} = \frac{1}{M_k} \begin{bmatrix} {}^{t}r_{0k} \\ {}^{t}r_{1k} \\ {}^{t}r_{2k} \end{bmatrix} \tag{7}$$

3 Results

The force recovery and resolution techniques are illustrated using data taken from scans of a beating heart phantom, using an isotropic, hyperelastic neo-Hookean tissue model. The Chamberlain Group CABG phantom, illustrated in figure 2 (left), was scanned at 54 bpm with a Philips 64-slice CT scanner, producing 10 uniformly-spaced phases. The first of these was manually segmented and converted into a tetrahedral mesh using the SimBio-Vgrid [2] mesh generator. Figure 2 (right) shows the interaction between the resulting mesh and a virtual tool. The Image Registration Toolkit [7, 8] was used to create a sequence of 3D tensor product cubic B-spline deformations, mapping the initial mesh onto each phase in turn. Cyclic cubic B-splines, defined using 6 uniformly spaced knots, were then used to interpolate mesh node positions over time. The material constitutive law is given as a strain energy density in equation (8), from which [10] the 2$^{\text{nd}}$ Piola-Kirchoff stress tensor elements S_{ij} can be derived. Here, C_{ij} is the right Cauchy-Green deformation tensor, I and J are its first and third invariants, respectively, and λ and μ are Lamé constants, defined in terms of Young's modulus E and Poisson's ratio ν.

$$W(I,J) = \frac{1}{2}\mu(I-3-2\ln J) + \frac{1}{2}\lambda(J-1)^2 \tag{8}$$

$$S_{ij} = \mu(\delta_{ij} - C_{ij}^{-1}) + \lambda J(J-1)C_{ij}^{-1} \tag{9}$$

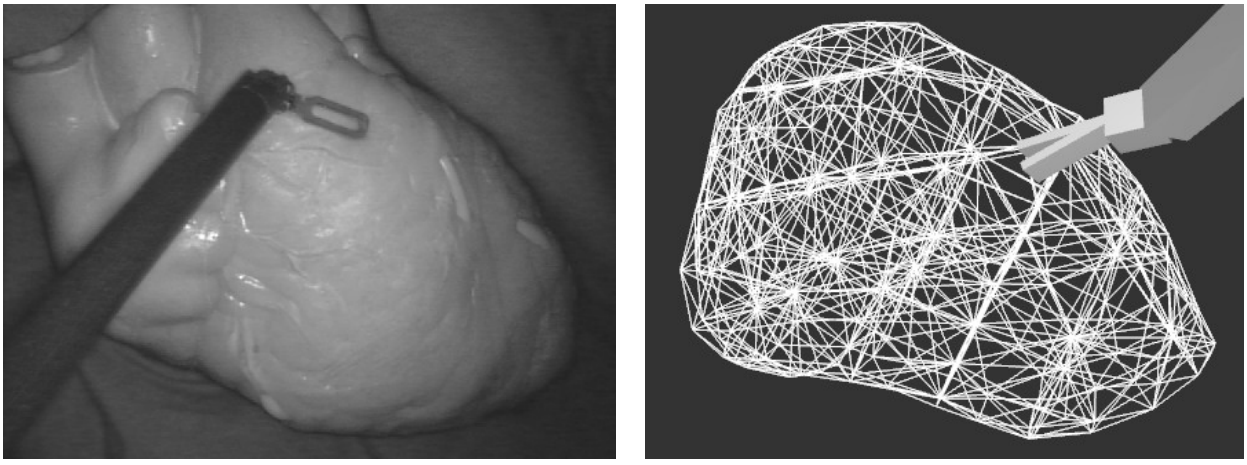

Figure 2: Beating heart phantom and tetrahedral FEM mesh with virtual tool

The following constants were used in the simulation: Young's modulus $E = 3.0\text{E}+03\,\text{Pa}$; Poisson's ratio $\nu = 0.45$; material density $\rho = 1.0\text{E}+03\,\text{kg/m}^3$; and mass damping coefficient $\alpha = 7.5\text{E}+01$. Thus, $\lambda \approx 9.31\text{E}+03\,\text{Pa}$ and $\mu \approx 1.03\text{E}+03\,\text{Pa}$. The equations of motion were integrated using a time step of $\Delta t = 0.001$ seconds.

Figure 3 shows the motion of a typical surface node as a result of applying recovered forces alone. Unlike the heart phantom itself, several nodes in the base of the mesh are deliberately anchored, and hence it takes a short period of time to converge to an exactly repeatable motion. Figures 4 and 5 show the individual effects of displacement constraints simulating respiratory motion, and externally applied forces simulating tool-tissue interaction, respectively. Finally, figure 6 illustrates how all three motions are combined by the model. The underlying beating motion is apparent throughout the simulation.

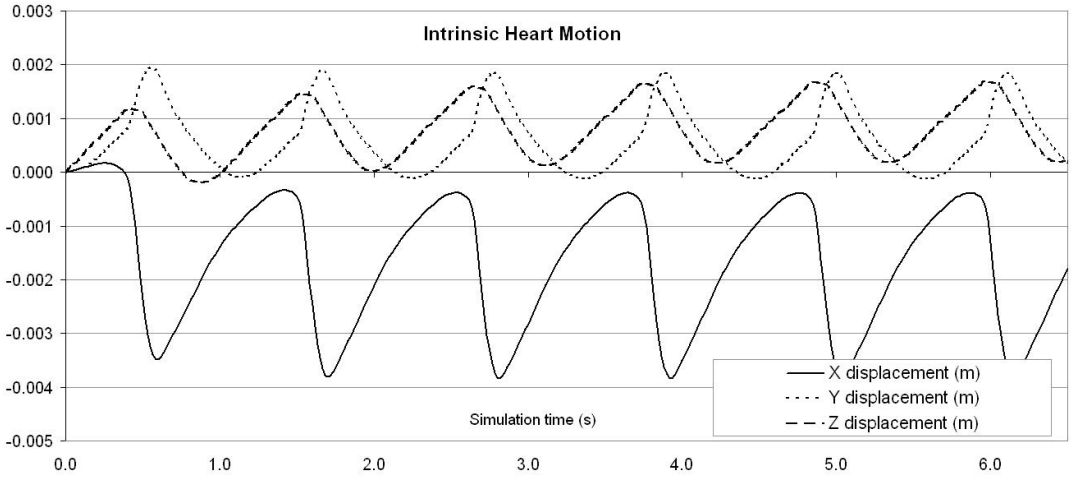

Figure 3: Intrinsic motion

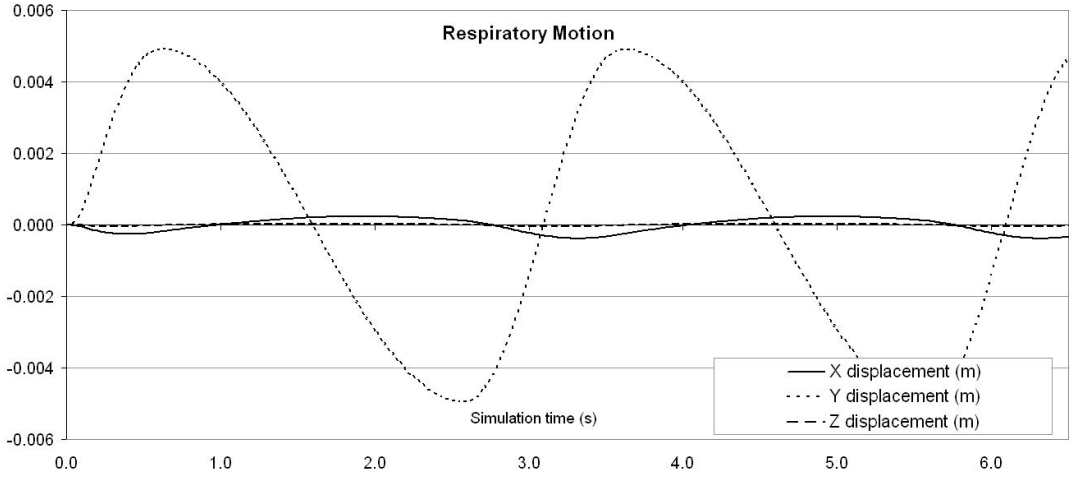

Figure 4: Respiratory motion

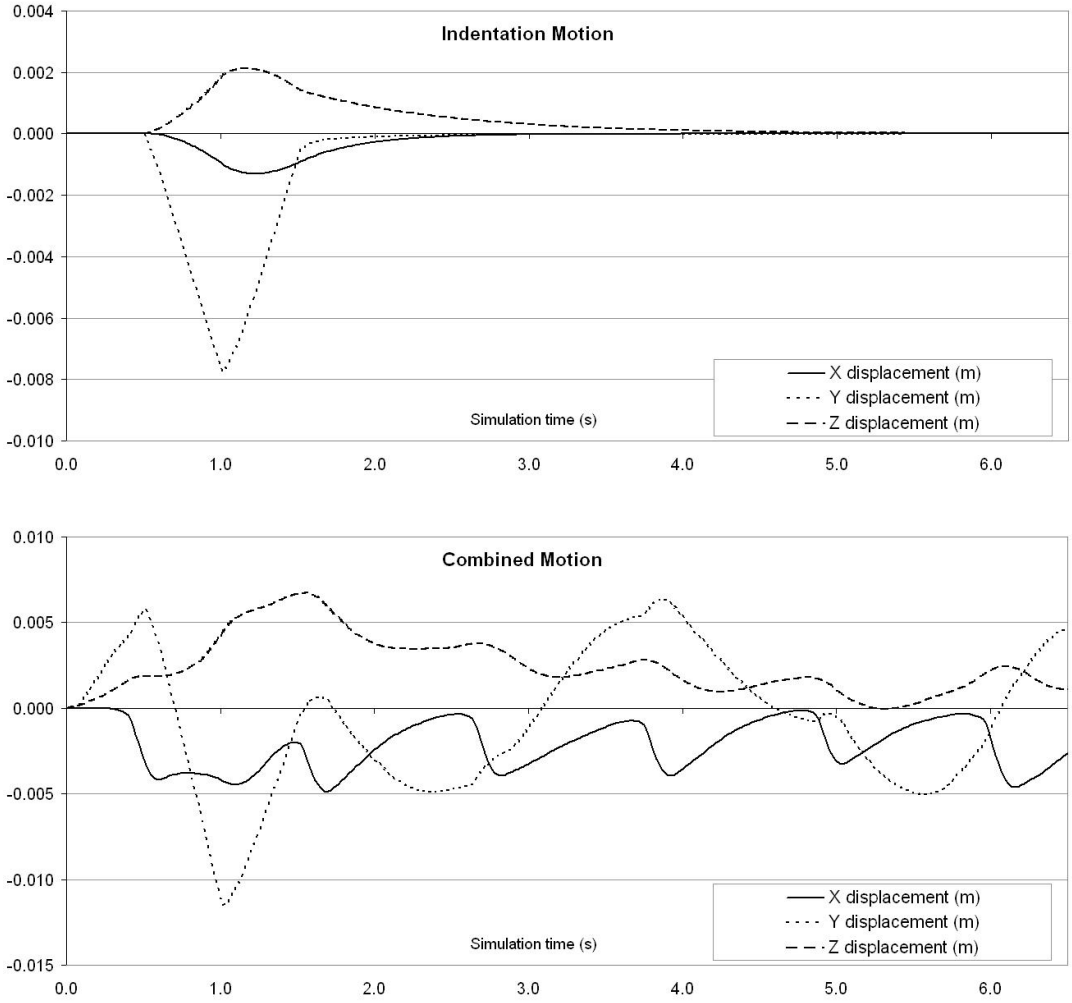

Figure 6: Combined intrinsic, respiratory and indentation motion

4 Conclusion

This paper describes a novel technique for constructing nonlinear finite element simulations with cyclical motion recovered from 4D tomographic scan data, whereby external forces and positional constraints can be combined to produce realistic composite behaviour. The technique has immediate applications in the field of patient-specific surgery simulation, and will also form the basis of an image-constrained biomechanical modelling approach to intraoperative image guidance. Future work includes validation of the technique with real patient data, an efficient GPU implementation, anisotropic weighting during the force recovery procedure, and the introduction of weight interpolation and scaling such that the frequency and intensity of the cyclic motion can be modified.

Acknowledgements

The Image Registration Toolkit was used under licence from IXICO Ltd.. The authors would like to thank staff at St. Mary's and the Royal Brompton hospitals.

References

[1] Klaus-Jürgen Bathe. *Finite element procedures*. Prentice Hall India, 1st edition, 2007. 1, 2

[2] Guntram Berti. Image-based unstructured 3D mesh generation for medical applications. In *European Congress on Computational Methods in Applied Sciences and Engineering - ECCOMAS 2004*. University of Jyväskylä, Department of Mathematical Information Technology, 2004. 3

[3] Martin Kauer. *Inverse Finite Element Characterization of Soft Tissues with Aspiration Experiments*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 2001. 1

[4] F. Kruggel and M. Tittgemeyer. Inverse Biomechanical Models of the Brain. In *Proceedings of the 2001 Bioengineering Conference*, pages 837–838. The American Society of Mechanical Engineers, New York, 2001. 1

[5] Karol Miller, Grand Joldes, Dane Lance, and Adam Wittek. Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in Numerical Methods in Engineering*, 23:121–134, 2007. 1

[6] Philip Pratt, Fernando Bello, Eddie Edwards, and Daniel Rueckert. Interactive finite element simulation of the beating heart for image-guided robotic cardiac surgery. In *Medicine Meets Virtual Reality 16*, pages 378–383. IOS Press, 2008. 1

[7] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Non-rigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999. 3

[8] J. A. Schnabel, D. Rueckert, M. Quist, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit, F. A. Gerritsen, D. L. G. Hill, and D. J. Hawkes. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. In *MICCAI 2001*, pages 573–581. Springer, 2001. 3

[9] Zeike A. Taylor, Mario Cheng, and Sébastien Ourselin. Real-time nonlinear finite element analysis for surgical simulation using graphics processing units. In *MICCAI 2007, Part I*, pages 701–708. Springer, 2007. 1

[10] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, Volume 2: Solid Mechanics*. Butterworth-Heinemann, 5th edition, 2000. 1, 3