Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 184 (2007) 52–62
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Modelling 3D semi-deformable tubes in real time

A.N.F. Klimowicz, M.D. Mihajlović *

School of Computer Science, The University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom

Abstract

Numerical simulation of deformable objects is an important problem in biomechanical engineering and computational science. A significant class of problems in this area requires that the simulation results are presented both haptically and graphically in real time (an example being a medical training simulator based on virtual reality). In such cases various types of trade-offs between accuracy and efficiency are implemented. Numerical procedures for modelling deformable objects, with respect to their efficiency, mainly belong to the following two classes: interactive methods (that are fast but have moderate accuracy), and continuum mechanics based methods (that are accurate, but generally not fully interactive). In this paper we present a numerical method based on oriented splines for the numerical simulation of semi-deformable tubes. The numerical results demonstrate both the accuracy and the interactivity of the proposed method, making it a suitable component of complex systems for interactive virtual reality simulations of biomechanical systems.

Keywords: Linear elastic tubes; Oriented splines; Lagrange equations; Newmark method; Cholesky factorisation

1. Introduction

Accurate numerical modelling and simulation of deformable objects is an important problem in mechanical engineering. When such models represent a constitutive part of complex interactive virtual reality systems (for example in medical training simulators [1,2,4,19,22]), in addition to the accuracy requirement, they must be capable of responding in real time to the operator constraints, both in graphics and in haptics [22]. This requirement is formalised as the minimisation of end-to-end lag time [24]. End-to-end lag time represents the delay between user's actions and the display of the results of these actions. Obviously, the lag time is application dependent, but some common contributions can be identified, numerical simulation time being one of them.

In this paper we are interested in numerical modelling of flexible tubes. There exists a host of applications in mechanical and biomechanical engineering where such models occur (the examples include common bile duct exploration [1], and simulations of various biological tissues such as intestines [7], blood vessels [8,10], and bronchioles [9]). The conventional numerical modelling techniques for simulation of elastic tubes can be

* Corresponding author.
E-mail addresses: klimowia@cs.man.ac.uk (A.N.F. Klimowicz), milan@cs.man.ac.uk (M.D. Mihajlović).

0096-3003/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2005.12.072

broadly classified into the following two categories: physically based models and interactive (or real-time) models. The first category includes the accurate continuum mechanics-based techniques, such as the finite element method (FEM) [27], and the boundary element method (BEM) [17]. The methods from the second category are also known as interactive models. In these models speed and latency are important, at the cost of low physical accuracy. Typical examples include mass-spring models [1] and deformable spline-based models [7,20].

The FEM can facilitate problems defined on complex domain geometries and which involve different material properties, and produces the numerical solution to an arbitrary degree of accuracy. However, due to large discrete problem sizes, it cannot achieve the necessary performance which enables update rates needed for interactive graphical rendering (20–30 Hz) and interactive haptics force feedback (200–1000 Hz) [2]. More recently a considerable amount of research effort has been invested in adapting the classical FEM for real-time simulations in biomechanical engineering. Some modifications include precomputation of some data [1,2] (this approach works in the case of small deformations and fixed meshes), problem condensation [3] (''calculate what you need'' principle), replacement of large deformations and/or non-linear anisotropic behaviour of living tissue by static deformations and/or isotropic models [1,22], modal (frequency) analysis of the solution (computing only the dominant Fourier modes of the solution) [1] (for a similar approach see [13]).

An alternative physically based method for simulation of deformable objects is the BEM [12]. The BEM is naturally confined to compute the updates on a surface. Thus, the method should be fast, providing that only a few boundary conditions change. The method's interactivity still depends on pre-processing, which involves precomputing of a number of different system responses (Green's functions), followed by a low-rank update solution reconstruction [12]. However, boundary nature of the BEM makes modelling of material properties, such as anisotropy, difficult.

Particle based techniques represent the continuum by a finite set of masses interconnected by springs and dampers. Such models are relatively simple to implement and computationally less expensive. However, these models have problems with numerical accuracy and numerical stiffness and often depend upon a number of parameters that are difficult to tune. Moreover, human tissues have complex anatomies and exhibit complex material properties (anisotropy, creep, visco-elasticity), making particle models less suitable in this context. Particle-based techniques exhibit good efficiency in simulating interactions between the objects and non-organic material bodies (such as the surgical instruments, catethers [1], sutures [2]). Finally, numerical methods based on splines are generally non-physical and are used to primarily achieve more realistic graphical rendering (e.g. NURBS [18]).

In this paper we introduce a numerical method for simulation of tube dynamics based on oriented splines. The method represents a trade-off between the physically based and interactive modelling. The key idea is to use two conventional splines in the context of the Lagrange equations (for an overview of different types of splines that can be used in this context see [20]). Material properties (such as mass distribution) are associated to each spline segment. A linear visco-elastic tube is discretised with respect to the spatial variables using oriented splines, yielding a system of ordinary differential equations in time. Numerical solution of this system is obtained by the simultaneous application of the Newmark method (the position and the rotation unknowns are decoupled and the appropriate equations can be treated independently).

The paper is organised as follows. Section 2 covers the details of the mathematical model. In Section 3 we present the implementation details, and Section 4 presents the simulation results in terms of accuracy, execution time and memory cost. The results confirm the effectiveness and interactivity of the proposed model. Finally, Section 5 is a discussion of some possible extensions and improvements of the model.

2. Mathematical model of an elastic tube

We want to model a semi-deformable tube in 3D. The tube geometry is defined by extruding a non-deformable ring in 2D (Fig. 1, right) along a deformable oriented spline curve in space (Fig. 1, left). A simplified version of this approach was introduced in [7,20]. We first give a brief overview of the spline functions used in this context. For further details see [5].

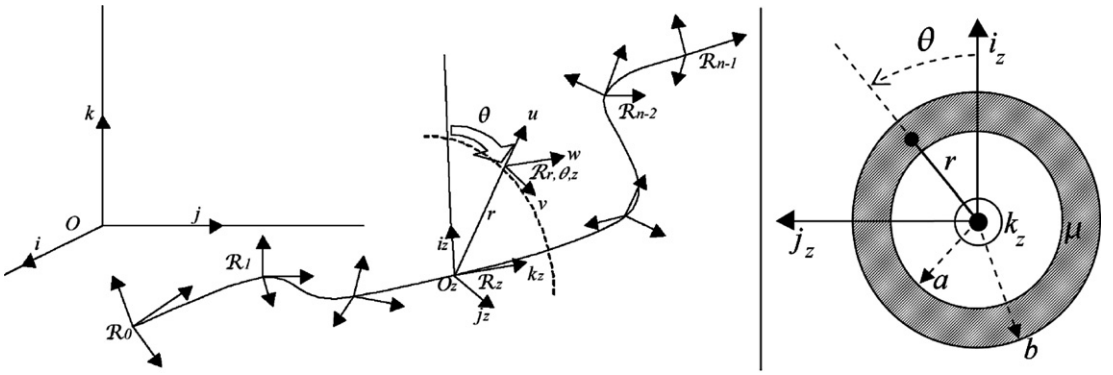

Fig. 1. A deformable oriented spline curve in space (left) and a non-deformable 2D ring (right).

2.1. The splines and the tube geometry

A spline curve in a 3D space $\vec{\mathbb{E}}$ is a piecewise polynomial function $\vec{f} : \mathbb{R} \to \vec{\mathbb{E}}$ such that for all $z \in \mathbb{R}$

$$\vec{f}(z) = \begin{cases} \vec{f}_i(z) & \text{if } \exists i \in [0, n-2] \text{ such that } z \in [z_i, z_{i+1}], \\ \vec{0} & \text{otherwise.} \end{cases} \tag{1}$$

In (1) $z_0 < z_1 < \cdots < z_{n-1}$ are n knots[1] which define $n-1$ segments. On each of these segments ($i = 0, \ldots, n-2$) the corresponding spline can be written [5]

$$\vec{f}_i(z) = \varphi_0\left(z_0 + \frac{z - z_i}{z_{i+1} - z_i}(z_1 - z_0)\right)^{\mathrm{T}} \xi_i + \varphi_1\left(z_0 + \frac{z - z_i}{z_{i+1} - z_i}(z_1 - z_0)\right)^{\mathrm{T}} \xi_{i+1}. \tag{2}$$

In Eq. (2) each node $\xi_0, \xi_1, \ldots, \xi_{n-1} \in \mathbb{R}^p$ gathers p scalar parameters and $\varphi_0(z), \varphi_1(z) \in \mathbb{R}[z]^{p,3}$ are the basis polynomial matrices which degree corresponds to that of the spline. If we assume the uniform knots $z_i = i$, the spline segments defined by (2) simplify to become

$$\vec{f}_i(z) = \varphi_0(z - i)^{\mathrm{T}} \xi_i + \varphi_1(z - i)^{\mathrm{T}} \xi_{i+1}. \tag{3}$$

In (3) we assume that $\varphi_0(0) = \varphi_1(1)$ and $\varphi_0(1) = \varphi_1(0) = 0$. If we introduce the notation

$$\lfloor z \rfloor_{[\alpha,\beta]} = \begin{cases} \alpha & \text{if } z < \alpha, \\ \beta & \text{if } z > \beta, \\ \lfloor z \rfloor & \text{otherwise} \end{cases} \tag{4}$$

for $z \in \mathbb{R}$, where $\lfloor \cdot \rfloor$ is the standard "floor" function, we can rewrite (1) on $[0, n-1]$ as

$$\vec{f}(z) = \varphi(z)^{\mathrm{T}} \xi. \tag{5}$$

In (5) the nodes are represented by $\xi \in (\mathbb{R}^p)^n$, while the shape function $\varphi(z) \in (\mathbb{R}[z]^{p,3})^n$ is such that

$$\varphi_i(z) = \delta_{\lfloor z \rfloor_{[0,n-2]},i} \cdot \varphi_0\left(z - \lfloor z \rfloor_{[0,n-2]}\right) + \delta_{\lfloor z \rfloor_{[0,n-2]}+1,i} \cdot \varphi_1\left(z - \lfloor z \rfloor_{[0,n-2]}\right) \tag{6}$$

for $i = 0, \ldots, n-1$, where $\delta_{i,j}$ is the Kronecker delta function.

The linear tube geometry is described by an ordered pair of splines, that are referred to as an oriented spline. Then the tube is represented by an extrusion of generalised cylindrical frames with parameters (r, θ, z) along the oriented spline (Fig. 1). The frames along the axis are generated using one spline curve for their position vector and another one for their rotation vector. The position spline models the axis, while the rotation spline completes its virtual skeleton. This approach should give more accurate representation of

[1] The term "knot" is standard in NURBS [18].

the tube than the approaches used in [7,20] that are based only on spline approximation of the position vectors. Similar approach (approximation of both the translation and the rotation degrees of freedom) in the context of FEM is presented in [1].

For each of the nodes $(\xi,\eta)\in(\mathbb{R}^p)^n\times(\mathbb{R}^q)^n$ of the oriented spline

$$(\varphi(z),\psi(z))\in(\mathbb{R}[z]^{p,3})^n\times(\mathbb{R}[z]^{q,3})^n, \tag{7}$$

the mappings

$$\xi\mapsto[\overrightarrow{OO_z}]_{\mathscr{B}}=\varphi(z)^{\mathrm{T}}\xi,\quad \eta\mapsto[\overrightarrow{\Theta\Theta_z}]=\psi(z)^{\mathrm{T}}\eta, \tag{8}$$

define the translation vector $\overrightarrow{OO_z}$ and the rotation vector $\overrightarrow{\Theta\Theta_z}$ from a Galilean referential $\mathscr{R}(O;\mathscr{B})$ to the solid $\mathscr{R}_z=(O_z;\mathscr{B}_z)$ with parametric abscissa $z\in[0,n-1]$ (see Fig. 1). This specification can be represented by the following homogeneous transition matrices

$$[\mathscr{R}_z]_{\mathscr{R}}=\begin{pmatrix}[\mathscr{B}_z]_{\mathscr{B}} & [\overrightarrow{OO_z}]_{\mathscr{B}}\\ \vec{0} & 1\end{pmatrix}, \tag{9}$$

where the rotation matrix $[\mathscr{B}_z]_{\mathscr{B}}$ from $\mathscr{B}$ to $\mathscr{B}_z$ is given by the Rodrigues formula

$$[\mathscr{B}_z]_{\mathscr{B}}=[\cos\omega_z]_3+\sin\omega_z\cdot\widehat{\mathbf{u}}_z+(1-\cos\omega_z)\mathbf{u}_z\mathbf{u}_z^{\mathrm{T}}. \tag{10}$$

In (10), $\omega_z=\|\overrightarrow{\Theta\Theta_z}\|$, $\mathbf{u}_z=[\overrightarrow{\Theta\Theta_z}]/\omega_z$ if $\omega_z\neq 0$ and 0 otherwise, and $\widehat{\mathbf{u}}_z$ is the pre cross-product matrix of $\mathbf{u}_z$. If $\mathbf{u}=(u_0\ u_1\ u_2)^{\mathrm{T}}$, then

$$\begin{pmatrix}u_0\\ u_1\\ u_2\end{pmatrix}^{\wedge}=\begin{pmatrix}0 & -u_2 & u_1\\ u_2 & 0 & -u_0\\ -u_1 & u_0 & 0\end{pmatrix}. \tag{11}$$

Then, full geometry of the system $[\mathscr{R}_{r,\theta,z}]_{\mathscr{R}}=[\mathscr{R}_z]_{\mathscr{R}}[\mathscr{R}_{r,\theta,z}]_{\mathscr{R}_z}$ can be obtained from the state vectors (ξ,η) with regards to the polar coordinates $(r,\theta)\in[a,b]\times[0,2\pi]$, $0<a<b$ in $\mathscr{R}_z$. The matrix $[\mathscr{R}_{r,\theta,z}]_{\mathscr{R}_z}$ defines the cylindrical frame as (see Fig. 1)

$$[\mathscr{R}_{r,\theta,z}]_{\mathscr{R}_z}=\begin{pmatrix}\cos\theta & -\sin\theta & 0 & r\cos\theta\\ \sin\theta & \cos\theta & 0 & r\sin\theta\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}. \tag{12}$$

2.2. Kinetic study

The tube is given as a material solid $\mathscr{R}(t)$ in space and time with an uniform parametric mass distribution $\hat{\mu}>0$ (expressed in kg m^{-2} u^{-1}, where u is the parametric unit along the axis). Note that, in general, mass distribution $\mu=\hat{\mu}/\|\varphi'(z)^{\mathrm{T}}\xi\|$ can be space/time dependent. The kinetic energy of a deformable tube with a generalised cylindrical geometry is

$$T=\frac{1}{2}\int\int\int_{M\in\mathscr{R}(t)}[\dot{\overrightarrow{OM}}]_{\mathscr{B}}^{\mathrm{T}}\cdot[\dot{\overrightarrow{OM}}]_{\mathscr{B}}\hat{\mu}\,\mathrm{d}r\cdot r\cdot\mathrm{d}\theta\cdot\mathrm{d}z. \tag{13}$$

In (13) $\overrightarrow{OM}$ is the position vector of any point of the tube, and $\mathscr{B}$ is the referential basis. Dot notation assumes partial differentiation with respect to time ($\dot{\overrightarrow{OM}}=\partial(\overrightarrow{OM})/\partial t$). We consider the case of non-deformable material slice ($\dot{r}=0$ and $\dot{\theta}=0$) of a circular shape. For this case the velocity field can be written as

$$[\dot{\overrightarrow{OM}}]_{\mathscr{B}}=\varphi(z)^{\mathrm{T}}\dot{\xi}(t)-[\mathscr{B}_z(t)]_{\mathscr{B}}\begin{pmatrix}r\cos\theta\\ r\sin\theta\\ 0\end{pmatrix}^{\wedge}\psi(z)^{\mathrm{T}}\dot{\eta}(t). \tag{14}$$

By substituting (14) into (13) we get the following expression for the kinetic energy:

$$T = \frac{1}{2}\hat{\mu}\left[\dot{\xi}(t)^{\mathrm{T}}\tilde{\mathbf{A}}\dot{\xi}(t) + \dot{\eta}(t)^{\mathrm{T}}\tilde{\mathbf{B}}\dot{\eta}(t) + \dot{\eta}(t)^{\mathrm{T}}\tilde{\mathbf{C}}\langle t,\eta\rangle\dot{\xi}(t)\right]. \tag{15}$$

For the definition formulas of the matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}\langle t,\eta\rangle$ see [14]. In the parameter space, the matrix $\hat{\mu}\tilde{\mathbf{A}}$ is the linear inertia matrix, $\hat{\mu}\tilde{\mathbf{B}}$ is the angular inertia matrix, and $\hat{\mu}\tilde{\mathbf{C}}\langle t,\eta\rangle$ is a coupling matrix. In the case of a fully closed circular ring (Fig. 1) we obtain

$$\begin{aligned}
\tilde{\mathbf{A}} &= \pi(b^2 - a^2)\int_0^{n-1}\varphi(z)\varphi(z)^{\mathrm{T}}\,\mathrm{d}z,\\
\tilde{\mathbf{B}} &= \frac{\pi}{4}(b^4 - a^4)\int_0^{n-1}\psi(z)\mathrm{diag}(1,1,2)\psi(z)^{\mathrm{T}}\,\mathrm{d}z,\\
\tilde{\mathbf{C}}\langle t,\eta\rangle &= 0,
\end{aligned} \tag{16}$$

thus yielding the following symmetric positive definite (SPD) quadratic form for the kinetic energy:

$$T = \frac{1}{2}\begin{pmatrix}\dot{\xi}(t)^{\mathrm{T}} & \dot{\eta}(t)^{\mathrm{T}}\end{pmatrix}\begin{pmatrix}\hat{\mu}\tilde{\mathbf{A}} & 0\\ 0 & \hat{\mu}\tilde{\mathbf{B}}\end{pmatrix}\begin{pmatrix}\dot{\xi}(t)\\ \dot{\eta}(t)\end{pmatrix}. \tag{17}$$

2.3. Static study

If we assume that the material spline is an isotropic, linear, elasto-static object [25], the finite displacements along the axis are

$$\left[\Delta(\overrightarrow{OO_z(t)})\right]_{\mathscr{B}} = \varphi(z)^{\mathrm{T}}\Delta\xi(t), \quad \left[\Delta(\overrightarrow{\Theta_z(t)})\right] = \psi(z)^{\mathrm{T}}\Delta\eta(t). \tag{18}$$

In (18) $\Delta x = x - x_{\mathrm{ref}}$, where x_{ref} is a reference state for x with respect to the referential $\mathscr{R}$. Then, the deformation energy can be written [25]

$$U = \frac{1}{2}\int_0^{n-1}\hat{k}\left(\frac{\partial}{\partial z}\left[\Delta\overrightarrow{OO_z(t)}\right]_{\mathscr{B}}\right)^2\mathrm{d}z + \frac{1}{2}\int_0^{n-1}\hat{\ell}\left(\frac{\partial}{\partial z}\left[\Delta\overrightarrow{\Theta_z(t)}\right]\right)^2\mathrm{d}z \tag{19}$$

where $\hat{k},\hat{\ell} > 0$ are the uniform parametric stiffness constants (in $\mathrm{N\,m^{-1}\,u^{-1}}$ and $\mathrm{N\,m^{-1}\,rad^{-1}\,u^{-1}}$, respectively). The first term in (19) represents the generalisation of the spring energy, and the second term is the generalisation of the bending energy of a beam. The expression (19) can be rewritten as the SPD quadratic form

$$U = \frac{1}{2}\begin{pmatrix}\Delta\xi(t)^{\mathrm{T}} & \Delta\eta(t)^{\mathrm{T}}\end{pmatrix}\begin{pmatrix}\mathbf{K} & 0\\ 0 & \mathbf{L}\end{pmatrix}\begin{pmatrix}\Delta\xi(t)\\ \Delta\eta(t)\end{pmatrix}, \tag{20}$$

with $\mathbf{K} = \hat{k}\hat{\mathbf{D}}$ and $\mathbf{L} = \hat{\ell}\hat{\mathbf{E}}$ being the linear and the angular stiffness matrices, respectively, and

$$\hat{\mathbf{D}} = \int_0^{n-1}\varphi'(z)\varphi'(z)^{\mathrm{T}}\,\mathrm{d}z, \quad \hat{\mathbf{E}} = \int_0^{n-1}\psi'(z)\psi'(z)^{\mathrm{T}}\,\mathrm{d}z. \tag{21}$$

From (21) it follows that both matrices $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ are SPD.

We assume that the tube is subjected to both conservative and non-conservative mechanical actions. As conservative external mechanical actions we consider volume forces and torques (that act in the interior of the tube), and surface forces and torques that act on the tube boundaries. The virtual work of the conservative external mechanical actions can be expressed as (see [14] for more details)

$$\delta W = \mathbf{S}^{\mathrm{T}}\delta\xi(t) + \mathbf{M}^{\mathrm{T}}\delta\eta(t). \tag{22}$$

The only non-conservative volume mechanical action is the viscous friction slider defined as $\vec{F}_c^{\,*} = -v\dot{\overrightarrow{OM}}$, where v is the coefficient of viscous dissipation. The virtual work of the non-conservative mechanical actions can be expressed as

$$\delta W^* = \mathbf{S}^{*\mathrm{T}}\delta\xi(t) + \mathbf{M}^{*\mathrm{T}}\delta\eta(t), \tag{23}$$

with $\mathbf{S}^* = -\hat{v}\tilde{\mathbf{A}}\dot{\xi}$ and $\mathbf{M}^* = -\hat{v}\tilde{\mathbf{B}}\dot{\eta}$, where $\hat{v}$ is a uniform parametric viscosity (in $\mathrm{N\,m^{-3}\,u^{-1}\,s}$).

2.4. Dynamic study and the Lagrange equations

In the Introduction we emphasised that our approach uses splines in the context of the Lagrange equations. The Lagrange equations are equations of motion which include the kinetic energy, the potential energy and the dissipative action. When solved numerically, the approximate solution represents the nodal displacements. The displacements are approximated by segregating the spatial and the time variables as $\mathbf{Y}(x,y,z,t) = \Phi(x,y,z)^{\mathrm{T}}\mathbf{X}(t)$, where $\mathbf{X}(t)$ are the generalised unknown displacements and $\Phi(x,y,z)$ are the basis functions of the spatial approximation. The general form of the Lagrange equations reads

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{\mathbf{X}}}\right) - \frac{\partial T}{\partial \mathbf{X}} + \frac{\partial U}{\partial \mathbf{X}} = \mathbf{Q} + \mathbf{Q}^{*}. \tag{24}$$

In (24) T and U are the kinetic and the deformation energy defined by (17) and (20), while $\mathbf{Q}$ and $\mathbf{Q}^{*}$ are the generalised conservative and non-conservative action vectors expressed by $\delta W = \mathbf{Q}^{\mathrm{T}}\delta\mathbf{X}$ and $\delta W^{*} = \mathbf{Q}^{*\mathrm{T}}\delta\mathbf{X}$. By substituting (17) and (20) into (24), we finally get the Lagrange equations for motion of the linear tube

$$\hat{\mu}\tilde{\mathbf{A}}\frac{\mathrm{d}^2(\Delta\xi)}{\mathrm{d}t^2} + \hat{\nu}\tilde{\mathbf{A}}\frac{\mathrm{d}(\Delta\xi)}{\mathrm{d}t} + \mathbf{K}(\Delta\xi) = \mathbf{S}, \tag{25}$$

$$\hat{\mu}\tilde{\mathbf{B}}\frac{\mathrm{d}^2(\Delta\eta)}{\mathrm{d}t^2} + \hat{\nu}\tilde{\mathbf{B}}\frac{\mathrm{d}(\Delta\eta)}{\mathrm{d}t} + \mathbf{L}(\Delta\eta) = \mathbf{M}. \tag{26}$$

If we introduce the notation

$$\mathbf{X} = \begin{pmatrix} \Delta\xi \\ \Delta\eta \end{pmatrix}, \quad \tilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{S} \\ \mathbf{M} \end{pmatrix}, \tag{27}$$

into (25), (26), we finally get

$$\hat{\mu}\begin{pmatrix} \tilde{\mathbf{A}} & 0 \\ 0 & \tilde{\mathbf{B}} \end{pmatrix}\ddot{\mathbf{X}} + \hat{\nu}\begin{pmatrix} \tilde{\mathbf{A}} & 0 \\ 0 & \tilde{\mathbf{B}} \end{pmatrix}\dot{\mathbf{X}} + \begin{pmatrix} \mathbf{K} & 0 \\ 0 & \mathbf{L} \end{pmatrix}\mathbf{X} = \tilde{\mathbf{Q}}. \tag{28}$$

Eq. (28) represents a linear system of second-order ordinary differential equations. The coefficient matrices in (28) are positive definite. Note that the equations for the position and rotation unknowns in (28) are decoupled and can be solved independently. This is in contrast with the FEM approach described in [1], where the coupling between these unknowns introduces a significant computational overhead.

3. Implementation details

The tube model that we suggest is based on oriented splines. This means that we use two splines $\varphi(z)$ and $\psi(z)$ to define a position and an orientation within the "snake-space" (Fig. 1). In general, the splines $\varphi(z)$ and $\psi(z)$ can be different, but for the sake of simplicity we use a single interpolation function $\varphi(z)$ for both the position and the orientation. In our implementation we use the Hermite interpolation splines at order 2 (thus, $\varphi_i(z) \in C^2[z_i, z_{i+1}]$) [5,6]. This choice represents a suitable trade-off between the numerical accuracy and smoothness of the solution. For this choice good numerical accuracy can be achieved with relatively few degrees of freedom, thus making the method interactive. The level of accuracy is also sufficient for the haptics feedback. On the other hand, the spline choice guarantees the curvature continuity at every node, which is helpful in adaptive tessellation, texturing and other visualisation issues, as well as in collision detection.

The spline nodes are defined as follows:

$$\xi_i(t) = \left(\left[\vec{\xi}_i^{(0)}(t)\right]_{\mathscr{B}}^{\mathrm{T}} \quad \left[\vec{\xi}_i^{(1)}(t)\right]_{\mathscr{B}}^{\mathrm{T}} \quad \left[\vec{\xi}_i^{(2)}(t)\right]_{\mathscr{B}}^{\mathrm{T}} \right)^{\mathrm{T}}, \tag{29}$$

$$\eta_i(t) = \left(\left[\vec{\eta}_i^{(0)}(t)\right]_{\mathscr{B}}^{\mathrm{T}} \quad \left[\vec{\eta}_i^{(1)}(t)\right]_{\mathscr{B}}^{\mathrm{T}} \quad \left[\vec{\eta}_i^{(2)}(t)\right]_{\mathscr{B}}^{\mathrm{T}} \right)^{\mathrm{T}}, \quad i = 0, \ldots, n-1. \tag{30}$$

In (29), (30) $[\vec{\xi}_i^{(k)}(t)]_{\mathscr{B}}$ and $[\vec{\eta}_i^{(k)}(t)]_{\mathscr{B}}$, $k = 0, 1, 2$ are 3×1 matrices of the kth derivative of the translation and rotation vectors for the ith node, respectively, in the basis $\mathscr{B}$, at time t. Thus, each node is described by 18 degrees of freedom. The shape functions from (6) are the Hermite splines at order 2 defined by[2]

$$\varphi_0(z) = \begin{pmatrix} [-6z^5 + 15z^4 - 10z^3 + 1]_3 \\ [-3z^5 + 8z^4 - 6z^3 + z]_3 \\ \left[-\frac{1}{2}z^5 + \frac{3}{2}z^4 - \frac{3}{2}z^3 + \frac{1}{2}z^2\right]_3 \end{pmatrix}, \tag{31}$$

$$\varphi_1(z) = \begin{pmatrix} [6z^5 - 15z^4 + 10z^3]_3 \\ [-3z^5 + 7z^4 - 4z^3]_3 \\ \left[\frac{1}{2}z^5 - z^4 + \frac{1}{2}z^3\right]_3 \end{pmatrix}. \tag{32}$$

With this choice of basis functions we need to assemble the coefficient matrices in (28) using the formulas (16) and (21). Since the basis functions have only the local support (each spline segment corresponds to a single tube element), the matrix assembly procedure is that of the FEM. Consequently, all the integrals of the basis functions defined between 0 and $n - 1$ can be calculated elementwise. By using the isoparametric mapping we can reduce the integration segment to $[0, 1]$ in all cases. Then the integrals are calculated numerically using the Gauss–Legendre quadrature rule. In our implementation we use the NAG routines D01BCF and D01EBF [16] to accomplish this task. The order of the quadrature rule is selected optimally with respect to the degree of the polynomial functions we need to integrate, thus computing the matrix elements within the machine accuracy. We use Horner's algorithm to evaluate basis polynomials (32) and their first derivatives. This enables us to calculate all the matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\mathbf{K}$, $\mathbf{L}$ in (28) simultaneously (see [14, Section 4.1.1] for full details). Due to a local support of the basis set, the coefficient matrices are banded, with a constant bandwidth depending on the number of parameters for each node, regardless of the number of tube elements. The band itself is nearly dense. Thus we do not expect a substantial amount of fill-in when applying direct methods to the solution of the linear system. Moreover, memory requirements for our model will scale linearly with the problem size. These facts offer a prospect of having an optimal algorithm for the problem that we consider here.

Numerical solution of the system of ordinary differential Eq. (28) is done by the simultaneous application of the Newmark method [27] to the position and orientation equation with the appropriate boundary conditions (BCs). The time variable is discretised uniformly with a time step $\mathrm{d}t > 0$, and we assume the initial conditions $\mathbf{X}_0$ and $\dot{\mathbf{X}}_0$ are known. We need to impose a suitable set of BCs, the simplest case being to impose known dynamics to some nodes (e.g. the node fixed to the referential $\mathscr{R}$ gives $\xi_k = 0$, $\dot{\xi}_k = 0$, $\ddot{\xi}_k = 0$, $\eta_k = 0$, $\dot{\eta}_k = 0$, $\ddot{\eta}_k = 0$). In order to take into account the BCs, the effect of the constrained nodes is added to the generalised actions in (28), and the rows and the columns of the matrices that correspond to the degrees of freedom associated with the constrained nodes are deleted. The (β_1, β_2)-Newmark scheme computes the unknown values $\mathbf{X}_{i+1}$, $\dot{\mathbf{X}}_{i+1}$, $\ddot{\mathbf{X}}_{i+1}$ from the known values $\mathbf{X}_i$, $\dot{\mathbf{X}}_i$, $\ddot{\mathbf{X}}_i$ by

$$\begin{pmatrix} \hat{\mu}\tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \hat{\mu}\tilde{\mathbf{B}} \end{pmatrix}\ddot{\mathbf{X}}_{i+1} + \begin{pmatrix} \hat{v}\tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \hat{v}\tilde{\mathbf{B}} \end{pmatrix}\dot{\mathbf{X}}_{i+1} + \begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix}\mathbf{X}_{i+1} = \tilde{\mathbf{Q}}_{i+1}, \tag{33}$$

$$\dot{\mathbf{X}}_{i+1} = \dot{\mathbf{X}}_i + \mathrm{d}t((1 - \beta_1)\ddot{\mathbf{X}}_i + \beta_1\ddot{\mathbf{X}}_{i+1}), \tag{34}$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathrm{d}t\dot{\mathbf{X}}_i + \frac{\mathrm{d}t^2}{2}((1 - \beta_2)\ddot{\mathbf{X}}_i + \beta_2\ddot{\mathbf{X}}_{i+1}). \tag{35}$$

If we substitute (34) and (35) into (33) we obtain two independent linear systems

$$\begin{pmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{pmatrix}\ddot{\mathbf{X}}_{i+1} = \tilde{\mathbf{Q}}_{i+1} - \begin{pmatrix} \hat{v}\tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \hat{v}\tilde{\mathbf{B}} \end{pmatrix}\dot{\mathbf{X}}^*_{i+1} - \begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix}\mathbf{X}^*_{i+1}, \tag{36}$$

[2] The notation $[x]_3$ used in (32) assumes a 3×3 diagonal matrix with x along the diagonal.

where

$$\dot{\mathbf{X}}^*_{i+1} = \dot{\mathbf{X}}_i + \mathrm{d}t(1-\beta_1)\ddot{\mathbf{X}}_i, \quad \mathbf{X}^*_{i+1} = \mathbf{X}_i + \mathrm{d}t\dot{\mathbf{X}}_i + \frac{\mathrm{d}t^2}{2}(1-\beta_2)\ddot{\mathbf{X}}_i, \tag{37}$$

$$\mathbf{F} = (\hat{\mu} + \beta_1 \cdot \mathrm{d}t \cdot \hat{v})\tilde{\mathbf{A}} + \left(\beta_2 \frac{\mathrm{d}t^2}{2}\right)\mathbf{K}, \tag{38}$$

$$\mathbf{G} = (\hat{\mu} + \beta_1 \cdot \mathrm{d}t \cdot \hat{v})\tilde{\mathbf{B}} + \left(\beta_2 \frac{\mathrm{d}t^2}{2}\right)\mathbf{L}. \tag{39}$$

Then from (34) we get

$$\dot{\mathbf{X}}_{i+1} = \dot{\mathbf{X}}^*_{i+1} + \beta_1 \cdot \mathrm{d}t \cdot \ddot{\mathbf{X}}_{i+1}, \tag{40}$$

and from (35)

$$\mathbf{X}_{i+1} = \mathbf{X}^*_{i+1} + \beta_2 \frac{\mathrm{d}t^2}{2}\ddot{\mathbf{X}}_{i+1}. \tag{41}$$

Notice that the position and the orientation at time t_i are given by

$$\begin{pmatrix} \xi_i \\ \eta_i \end{pmatrix} = \begin{pmatrix} \xi_{\mathrm{ref}} \\ \eta_{\mathrm{ref}} \end{pmatrix} + \mathbf{X}_i. \tag{42}$$

The choice $\beta_1 = \beta_2 = \frac{1}{2}$ in (36)–(41) gives an unconditionally stable implicit scheme [27].

The matrices **F** and **G** in (36) are SPD. This means that the two independent linear systems in (36), which solution is needed at each time step, can be solved effectively by computing a Cholesky factorisation of the matrices **F** and **G**. This needs to be done only once before Newmark's scheme, as (38), (39) indicates that the matrices **F** and **G** remain unchanged throughout the time integration. The last property is a consequence of the linear character of the system (28). During the time integration only two forward and backsubstitutions are needed to compute (36) at each time step. In our implementation we use the Bunch–Kaufman variant of the banded Cholesky factorisation from the NAG library [16] (it is an LDL^{T} factorisation).

Our primary objective in this work was to develop an efficient numerical procedure for the simulation of visco-elastic tubes in real time, rather than more challenging and complex task of developing a complete virtual reality simulator with graphics and haptics feedback (although our model satisfies all preconditions to be a part of such a system). However, in order to visualise our results, we use optimal drawing primitives from OpenGL [26]. The drawing mode GL_TRIANGLE_STRIP is used to form a tube surface from a series of conjoined triangles (see the graphical results in Section 4). In our implementation no attempt was made to optimise the performance of graphical output. For some possibilities, see Section 5.

4. Numerical results

In this section we demonstrate the numerical accuracy and efficiency of our model. As a representative example, we consider a straight tube which is initially deformed into a stressed position. The tube is fixed at one boundary. When no gravity is taken into account, the tube returns to its reference configuration after passing through a few damped oscillations around the equilibrium. Due to a specific set of initial and boundary conditions, the tube motion in this example will be planar. However, this example represents a typical benchmark in mechanical engineering which enables validation of the model. More complicated combinations of initial and boundary conditions usually produce 3D motion of the tube, and our model can accommodate such cases. The tube parameters in our experiment are $a = 0.32$ m, $b = 0.4$ m, $L = 12$ m, $\hat{k} = 1.2$ N m^{-1} u^{-1}, $\hat{\mu} = 0.8$ kg m^{-2} u^{-1}, $\hat{\ell} = 1.0$ N m rad^{-1} u^{-1}, and $\hat{v} = 0.5$ N m^{-3} u^{-1} s. We measure the execution time per one Newmark step and the maximal relative errors in the positions and the orientations as functions of the problem size. In order to estimate the solution accuracy, we compare the solutions computed with various numbers of tube elements N_{elt} with a reference solution obtained with $N_{\mathrm{elt}} = 128$. Computations with larger number of elements produce the solutions that are essentially the same as the one with 128 elements. The accuracy is estimated during the transition period, well away from the equilibrium configuration. We adopted equidistant

Table 1
Efficiency and accuracy of the tube model based on oriented splines as a function of the problem size

N_{elt}	N_{dof}	T (ms)	M (kB)	ε_{ps}	ε_{or}
16	306	1.497	118	0.3487	0.1418
32	594	3.055	226	0.1493	0.1248
64	1170	6.640	442	0.0498	0.0693
96	1746	9.444	658	0.0166	0.0448
128	2322	13.124	874	–	–

The execution time T is in ms per Newmark step, memory requirements M are in kB, and position (ε_{ps}) and orientation (ε_{or}) relative errors are recorded after 1000 time steps.

time steps of $dt = 0.005$ s. With such time discretisation, the equilibrium for the problem under consideration occurs after approximately 6000 time steps. Our error estimates are sampled after 1000 time steps. As the equilibrium is approached, these errors asymptotically approach 0.

The model is implemented in C++. The code was compiled with MS Visual C++ 6.0 and was run under Windows XP on a PC with a single Intel Pentium 4 CPU at 1.8 GHz and 512 MB of RAM. The results are summarised in Table 1. In the table N_{elt} denotes the number of tube elements, and N_{dof} is the total number of degrees of freedom. T is the CPU clock time (in *ms*) needed to do the computations within one Newmark iteration, and M is the memory (in kB) used to store the data. Relative errors in position ε_{ps} and in orientation ε_{or} are computed as the maximum difference between the approximate solutions for the frames R_i of the 128-element tube, and frames R_j of the N_{elt} tube, where $j = i \cdot N_{elt}/128$. The position error is normalised with respect to the tube length L, while the orientation error is normalised with respect to 2π.

From Table 1 it can be confirmed that the execution times and memory requirements scale linearly with the problem size. Thus our method has optimal solver property. From the execution times it can be concluded that the computation time per time step is sufficiently small to allow real-time simulations for problem sizes of approximately 50 elements ($N_{dof} = 918$) if we require haptics feedback at 200 Hz, and for problem sizes of approximately 500 elements (corresponding to $N_{dof} = 9180$) if we require only graphical output at 20 Hz. For the case $N_{elt} = 50$ both position and orientation relative errors are approximately 5%, what is well within the range of tolerances admissible for these types of models. Problem sizes that allow real-time modelling are actually the upper limits, as the execution times from Table 1 do not include time spent to produce graphics or haptics output. The execution times from Table 1 does not include the one-off cost of computing the LDL^T factorisation of the matrices $\mathbf{F}$ and $\mathbf{G}$ from (36). For the case $N_{elt} = 128$ the time needed to create the tube object (which include memory allocations, assembly of the matrices, application of the BCs, and computation of the LDL^T factorisation) is 80 ms. Thus, the startup delay does not present a considerable overhead, and it can be barely noticed by a human operating a simulator. This is in contrast with the FEM and the BEM for the real-time simulations [1,12], that require a formidable off-line computation prior to the real-time simulation.

Finally, in Fig. 2 we present the screen shots produced by OpenGL of 3 tubes obtained with different number of elements $N_{elt} = 1, 2, 4$, corresponding from bright ($N_{elt} = 1$) to dark ($N_{elt} = 4$) colour of the tube.

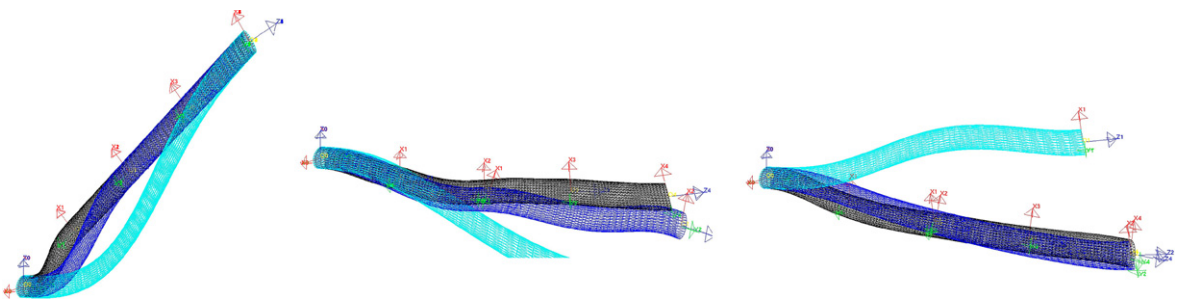

Fig. 2. Screen shots of the tubes with 1, 2, 4 elements (from bright to dark colour) at time steps $t = 0$ s, $t = 2$ s and $t = 4$ s (from left to right).

The screen shots are taken from the simulation of the model problem described in this section at the times $t = 0$ s, $t = 2$ s, and $t = 4$ s (from left to right). From the figure it can be seen that the simulations match relatively closely even for relatively small number of tube elements (the exception is 1 element, when the simulation is fairly inaccurate).

5. Discussion and conclusions

In this paper we present a numerical method for simulation of semi-deformable tubes based on oriented splines, which are applied in context of the Lagrange equations. This model is suitable for inclusion in complex medical training simulators based on virtual reality, or other systems where numerical modelling in real time is an essential requirement. Such choice of basis functions for spatial discretisation is motivated by the requirement to achieve sufficient accuracy in real time, and to enable realistic graphical output and haptics feedback. Over the past years a substantial research effort was invested to upgrade standard physically based numerical methods, such as the FEM and the BEM to work under the real-time constraints. The updates are usually based on moving a considerable amount of computational work off-line [1,12]. This approach restricts considerably the number of applications for which satisfactory results can be obtained. Our approach is similar to that of the FEM in the sense that we subdivide a tube into a (moderate) number of elements. Then the position and the orientation within each element is approximated by a Hermite spline at order 2. When this spatial discretisation is used in context of the Lagrange equations, we obtain an uncoupled system of ODEs in position and orientation variables. Time integration using Newmark's method require the solution of two linear systems at each step. The main advantage of our approach, comparing to the related FEM [1] is that the two linear systems are decoupled and can be solved efficiently. Several modelling techniques for elastic objects found in the literature [7,20] are based on splines. However, these techniques are based on approximation of the position variables only. In our approach we consider both the position and the orientation as the unknown degrees of freedom. This increases robustness and accuracy of our methodology (see Table 1), making it comparable to that of the FEM. In addition, smoothness of the spline basis functions is beneficiary for the graphical rendering issues. In our case the only off-line computation is related to the assembly of the tube object and two LDL^{T} factorisations of the matrices in (38), (39). As we have seen in Section 4, this cost is not prohibitive.

At the end we give some potential modifications which can make the proposed model more efficient. First, all the matrices in (28) can be assembled in a FEM fashion (elementwise). These computations are independent and can be done in parallel. The two LDL^{T} factorisations can be done independently and in parallel as well. There is a host of efficient parallel library codes (see for example [11]) targeted for such applications. Replacing the direct solver by an iterative Krylov method [21] preconditioned by the algebraic multigrid (AMG) [23] can be an alternative. However, our experience in the context of FE modelling of linear elasticity problems is that real benefits of an iterative solver become obvious when discrete problem size exceeds 10^4 degrees of freedom [15]. Thus, direct solution methods seem to be optimal in this context. When this model is used as a part of a more complex simulator, further parallelisation of the tasks is possible as suggested in [24]. For example, numerical computation can be done on one group of computers, while graphics and haptics output can be generated on the other group of machines, presumably with special graphical cards and optimised software for graphical applications. Communication between the machines can be realised by the MPI. However, this would incur communication delays, especially on shared networks.

References

[1] C. Basdogan, C.-H. Ho, M.A. Srinivasan, Virtual environments for medical training: graphical and haptic simulation of laparoscopic common bile duct exploration, IEEE/ASME Trans. Mechatronics 6 (3) (2001) 269–285.
[2] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, S. Weghorst, Real-time finite element modeling for surgery simulation: an application to virtual suturing, IEEE Trans. Visual Comp. Graphics 10 (3) (2004) 314–325.
[3] M. Bro-Nielsen, Fast finite elements for surgery simulation, Stud. Health Techn. Inform. 39 (1997) 395–400.
[4] S.L. Delp, J.P. Loan, M.G. Hoy, F.E. Zajac, E.L. Topp, J.M. Rosen, An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures, IEEE Trans. Biomed. Eng. 37 (8) (1990) 757–767.
[5] P. Dierckx, Curve and Surface Fitting with Splines, Oxford University Press, Oxford, 1993.

[6] M. Attéia, J. Gaches, Approximation Hilbertienne. Splines-Ondelettes-Fractales, EDP Sciences, Les Ulis, 1999.
[7] L. France, J. Lenoir, A. Angelidis, P. Meseure, M.-P. Cani, F. Faure, C. Chaillou, A layered model of a virtual human intestine for surgery simulation, Medical Image Anal. 9 (2) (2005) 123–132.
[8] A.L. Hazel, M. Heil, Steady finite Reynolds number flows in three-dimensional collapsible tubes, J. Fluid Mech. 486 (2003) 79–103.
[9] A.L. Hazel, M. Heil, Three-dimensional airway reopening: The steady propagation of a semi-infinite bubble into a buckled elastic tube, J. Fluid Mech. 478 (2003) 47–70.
[10] M. Heil, O.E. Jensen, Flow in deformable tubes and channels: theoretical models and biological applications, in: T.J. Pedley, P.W. Carpenter (Eds.), Flow in Collapsible Tubes and Past Other Highly Compliant Boundaries, Kluwer, Dodrecht, 2003, pp. 15–50.
[11] HSL: A collection of Fortran codes for large-scale scientific computation, 2002 (see <http://hsl.rl.ac.uk/>).
[12] D.L. James, D.K. Pai, ArtDefo: Accurate Real Time Deformable Objects, in: Proc. ACM SIGGRAPH 1999, pp. 65–72.
[13] B.N. Khoromskij, G.E. Mazurkevich, G. Wittum, Frequency filtering for elliptic interface problems with Lagrange multipliers, SIAM J. Sci. Comput. 21 (2) (1999) 421–440.
[14] A.N.F. Klimowicz: Accurate real time deformable tubes, MSc. thesis, The University of Manchester, 2003. Available from: <http://www.cs.manchester.ac.uk/cnc/students/aklimowicz/>.
[15] M.D. Mihajlović, S.Z. Mijalković, Efficiency study of the "black-box" component decomposition preconditioning for discrete stress analysis problems, in: Proc. ICCS 2004, Lect. Notes in Comput. Sci., vol. 3037, 2004, pp. 97–104.
[16] Numerical Algorithms Group: NAG Manual, Fortran Library Mark 20, 2002.
[17] F. Paris, J. Cañas, Boundary Element Method: Fundamentals and Applications, Oxford University Press, Oxford, 1997.
[18] L. Piegl, On NURBS: a survey, Comput. Graphics Appl. 11 (1) (1991) 55–71.
[19] S. Piper, J. Rosen, D. Zeltzer, Interactive graphics for plastic surgery, Comput. Graphics 20 (4) (1986) 55–64.
[20] Y. Rémion, J.-M. Nourrit, D. Gillard, A dynamic animation engine for generic spline objects, J. Visual. Comput. Animat. 11 (2000) 17–26.
[21] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelphia, 2003.
[22] M.A. Sagar, D. Bullivant, G.D. Mallinson, P.J. Hunter, A virtual environment and model of the eye for surgical simulation, in: Proc. 21st Conf. Comp. Graphics and Interact. Tech., 1994, pp. 205–212.
[23] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math. 128 (2001) 281–309.
[24] V.E. Taylor, J. Chen, T.L. Disz, M.E. Papka, R. Stevens, Interactive virtual reality in simulations: exploring lag time, IEEE Comput. Sci. Eng. 3 (4) (1996) 46–54.
[25] G. Wempner, Mechanics of Solids with Applications to Thin Bodies, McGraw-Hill, New York, 1973.
[26] M. Woo, J. Neider, T. Davis, OpenGL Programming Guide, Addison-Wesley, London, 1997.
[27] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method, fifth ed., vol. 1, Butterworth-Heinemann, Oxford, 2000.