

New Approaches to Catheter Navigation for Interventional Radiology Simulation

C. Duriez³, S. Cotin^{1,2}, J. Lenoir¹, P. Neumann^{1,2}

¹ Sim Group, CIMIT, Cambridge, MA 02139, USA

² Harvard Medical School, Boston MA 02114, USA

³ INRIA-Futurs, Université de Lille, 59655 Villeneuve d’Ascq, FRANCE

Abstract. For over 20 years, interventional methods have improved the outcomes of patients with cardiovascular disease. However, these procedures require an intricate combination of visual and tactile feedback and extensive training periods. In this paper, we describe a series of novel approaches that have led to the development of a high-fidelity simulation system for interventional neuroradiology. In particular we focus on a new approach for real-time deformation of devices such as catheters and guidewires during navigation inside complex vascular networks. This approach combines a real-time incremental Finite Element Model, an optimization strategy based on substructure decomposition, and a new method for handling collision response in situations where the number of contacts points is very large. We also briefly describe other aspects of the simulation system, from patient-specific segmentation to the simulation of contrast agent propagation and fast volume rendering techniques for generating synthetic X-ray images in real-time.

1 Introduction

Stroke, the clinical manifestation of cerebrovascular disease, is the third leading cause of death in the United States. Each year, more than 700,000 strokes result in over 200,000 deaths [2]. Ischemic strokes can now be treated using interventional neuroradiologic therapies which rely on the insertion and navigation of catheters and guidewires through a complex network of arteries to restore blood flow. Because the treatment is delivered directly within the closed brain, using only image-based guidance, the dedicated skill of instrument navigation and the thorough understanding of vascular anatomy are critical to avoid devastating complications which could result from poor visualization or poor technique. This was the foundation for a recent decision by the FDA requiring all physicians who wish to treat carotid disease using catheter-based techniques to train to proficiency before performing high-risk procedures in the cerebral circulation. However, while most publications in the field of medical simulation have addressed issues related to laparoscopic training, few aspects of interventional radiology simulation have been explored. Aside of modeling the soft tissue deformation of arteries, other complex problems need to be solved to enable real-time, accurate simulation of such procedures. The one we address in this

paper concerns the simulation of non-linear deformations of wire-like structures under a large number of non-holonomic constraints, and the definition of such constraints to confine the catheter inside the vascular network. We also briefly mention other results in the areas of real-time fluid flow computation, real-time synthetic X-ray rendering, and patient-specific segmentation. Although some of these problems have been addressed in previous work [4, 11, 8, 1], in particular in the areas of visualization and catheter modeling, many challenging problems remain, especially when trying to reach a higher level of fidelity and accuracy in the simulation, while maintaining real-time computation performance.

2 Modeling of wire-like structures

To control the motion of a catheter or guidewire within the vascular network, the physician can only push, pull or twist the proximal end of the device. Since such devices are constrained inside the patient's vasculature, it is the combination of input forces and contact forces that allow them to be moved toward a target. The main characteristics of wire-like structures that current models attempt to capture include geometric non-linearities, high tensile strength and low resistance to bending. We previously proposed a multibody dynamics model [6] where a set of rigid elements are connected using spherical joints [4], thus mimicking the basic behavior of such devices. Another interesting approach to modeling wire-like structures was introduced by Lenoir *et al.* [9]. In this model, a one-dimensional dynamic spline model is used, providing a continuous representation. Different constraints can be defined to control the model, such as sliding through fixed locations. Although real-time computation is possible, this model does not incorporate torsional energy terms. A virtual catheter model, based on a linear elasticity, was introduced by Nowinski *et al.* [11], using a set of finite beam elements. The choice of beams for the catheter model is natural since beam equations account for cross-sectional area, cross-section moment of inertia, and polar moment of inertia, allowing solid and hollow devices of various cross-sectional geometries and mechanical properties to be modeled. The main issue with such a linear model is its inability at representing the large geometric non-linearities that occur during navigation of the catheter or guidewire within the vascular network. Another approach also directly targeted at virtual catheter or guidewire modeling was proposed by Alderliesten [1]. In this model, only bending energies are computed, assuming no elongation and perfect torque control. The model has characteristics similar to a multi-body dynamics model but integrates more complex bending energies, as well as local springs for describing the intrinsic curvature of the catheter. Although based on a more ad-hoc representation, a good level of accuracy is obtained using this model. The main drawbacks are how collision response is handled during contact with the walls of the vessel, and computation times that are not compatible with real-time requirements.

To improve the accuracy of previously proposed models, and handle geometric non-linearities while maintaining real-time computation, we have developed a new mathematical representation based on three-dimensional beam theory,

combined with an incremental approach that allows for highly non-linear behavior using a linear model, thus guaranteeing real-time performance. Further optimizations based on substructure decomposition are introduced. Finally, a new method for correctly handling contact response in complex situations where a large number of nodes are subject to non-holonomic constraints, is presented.

2.1 Incremental Finite Element Model

To model the deformation of a catheter, guidewire, or any solid body whose geometry and mechanical characteristics are similar to a wire, rod or beam, we use a representation based on three-dimensional beam theory [12], where the elementary stiffness matrix $\bar{\mathbf{K}}_e$ is a 12×12 symmetric matrix that relates angular and spacial positions of each end of a beam element to the forces and torques applied to them:

$$\bar{\mathbf{K}}_e = \frac{E}{l} \left[\begin{array}{cccccc|cccccc} A & & & & & & & & & & & \\ 0 & \frac{12I_z}{l^2(1+\Phi_y)} & & & & & & & & & & \\ 0 & 0 & \frac{12I_y}{l^2(1+\Phi_z)} & & & & & & & & & \\ 0 & 0 & 0 & \frac{GJ}{E} & & & & & & & & \\ 0 & 0 & \frac{-6I_y}{l(1+\Phi_z)} & 0 & \frac{(4+\Phi_z)I_y}{1+\Phi_z} & & & & & & & \\ 0 & \frac{6I_z}{l(1+\Phi_y)} & 0 & 0 & 0 & \frac{(4+\Phi_y)I_z}{1+\Phi_y} & & & & & & \\ \hline -A & 0 & 0 & 0 & 0 & 0 & A & & & & & \\ 0 & \frac{-12I_z}{l^2(1+\Phi_y)} & 0 & 0 & 0 & \frac{-6I_z}{l(1+\Phi_y)} & 0 & \frac{12I_z}{l^2(1+\Phi_y)} & & & & \\ 0 & 0 & \frac{-12I_y}{l^2(1+\Phi_z)} & 0 & \frac{6I_y}{l(1+\Phi_z)} & 0 & 0 & 0 & \frac{12I_y}{l^2(1+\Phi_z)} & & & \\ 0 & 0 & 0 & \frac{-GJ}{l} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{E} & & \\ 0 & 0 & \frac{6I_y}{l(1+\Phi_z)} & 0 & \frac{(2-\Phi_z)I_y}{1+\Phi_z} & 0 & 0 & 0 & \frac{6I_y}{l(1+\Phi_z)} & 0 & \frac{(4+\Phi_z)I_y}{l(1+\Phi_z)} & \\ 0 & \frac{6I_z}{l(1+\Phi_y)} & 0 & 0 & 0 & \frac{(2-\Phi_y)I_z}{1+\Phi_y} & 0 & \frac{-6I_z}{l(1+\Phi_y)} & 0 & 0 & 0 & \frac{(4+\Phi_y)I_z}{1+\Phi_y} \end{array} \right]$$

with $G = \frac{E}{2*(1+\nu)}$ where E is the Young's modulus and ν is the Poisson's ratio; A is the cross-sectional area of the beam, and l its length; I_y and I_z are cross-section moments of inertia; Φ_y and Φ_z represent shear deformation parameters and are defined as $\Phi_y = \frac{12EI_z}{GA_{sy}l^2}$ and $\Phi_z = \frac{12EI_y}{GA_{sz}l^2}$ with A_{sy} and A_{sz} the shear area in the the y and z directions.

In order to determine the stiffness property of the complete structure, a common reference frame must be established for all unassembled structural elements so that all the displacements and their corresponding forces will be referred to a common (global) coordinate system. Since the stiffness matrix $\bar{\mathbf{K}}_e$ is initially calculated in local coordinates, oriented along the frame of first node of the beam (to minimize the computing effort), it is necessary to introduce transformation matrices changing the frame of reference from a local to a global coordinate system. The first step in deriving such a transformation is to obtain a matrix

relationship between the element displacement $\bar{\mathbf{u}}$ in the local coordinate system and the element displacement \mathbf{u} in the global coordinates (see Fig 1). This relationship is expressed by the matrix equation:

$$\bar{\mathbf{u}} - \bar{\mathbf{u}}_0 = \mathbf{\Lambda} \mathbf{u} \quad (1)$$

where $\mathbf{\Lambda}$ is a matrix of coefficients obtained from the direction cosines of angles between the local and global coordinate systems and $\bar{\mathbf{u}}_0$ reflects the initial configuration of the beam

As the resulting virtual work must be independent of the coordinate system, it follows that:

$$\delta \bar{\mathbf{u}}^T \bar{\mathbf{f}} - \delta \mathbf{u}^T \mathbf{f} = 0 \Leftrightarrow \delta \mathbf{u}^T (\mathbf{\Lambda}^T \bar{\mathbf{f}} - \mathbf{f}) = 0 \Leftrightarrow \mathbf{\Lambda}^T \bar{\mathbf{f}} - \mathbf{f} = 0 \quad (2)$$

By use of previous equations, the following element force-displacement equation is obtained in global coordinates:

$$\mathbf{f} = \underbrace{[\mathbf{\Lambda}^T \bar{\mathbf{K}}_e \mathbf{\Lambda}]}_{\mathbf{K}_e} \mathbf{u} \quad (3)$$

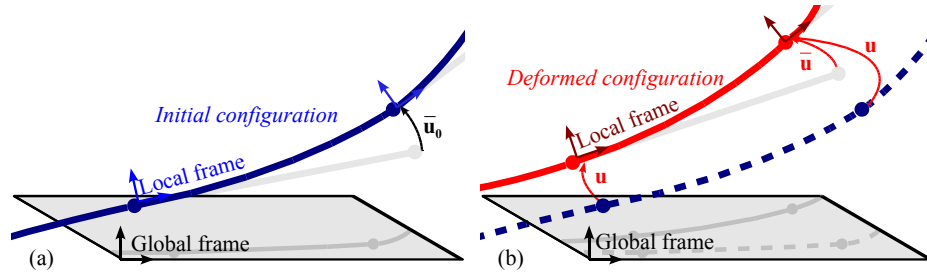


Fig. 1. Initial (blue) and deformed (red) configurations. (a) The initial configuration of a catheter could be curved. Thus, for each beam, we store an initial displacement $\bar{\mathbf{u}}_0$ based on the position of the tip node in the local frame of the base node. (b) Actual deformations can always be measured locally in the local reference frame using $\bar{\mathbf{u}}$. Displacements from initial configuration to deformed configuration are measured using \mathbf{u} in the global reference frame and by $\bar{\mathbf{u}} - \bar{\mathbf{u}}_0$ in the local frame.

To model a wire-like structure such as a catheter or guidewire, we serially link a series of beam elements (see Fig 1). As a result, for the entire structure describing a catheter or guidewire, the global stiffness matrix \mathbf{K} is computed by summing the contributions of each element, thus leading to the following equilibrium equation:

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (4)$$

where \mathbf{K} is a band matrix due to the serial structure of the model (one node is only shared by one or two elements) and \mathbf{u} represents a column matrix of displacements corresponding to external forces \mathbf{f} .

The matrix \mathbf{K} is singular unless some displacements are prescribed through boundary conditions. Such boundary conditions are naturally specified by setting the first node (base node) of the device model to a particular translation or rotation imposed by the user. The motion and deformation of the model is then entirely determined by its set displacement at the base node and the forces f applied at different nodes as the result of contacts with the vessel wall (see section 2.3). There is, however, one main drawback in using directly such a model: it is linear and as such cannot represent the geometric non-linearities that a typical wire-like object exhibits. Therefore we propose to update $\mathbf{\Lambda}$ for each beam element, and at every time step, by using the solution obtained at the previous time step. The new set of local stiffness matrices are then assembled in $[\mathbf{K}]$. This approach is different than other methods such as the co-rotational method [7] or other techniques that remove the rigid body transformation from a given configuration to remain in the linear domain [10]. Here, we do not use the initial configuration as the reference state, but instead use the previously computed solution. By controlling when each new \mathbf{K} is going to be computed, we can ensure we remain in the linear domain for each incremental step, leading to a correct, global deformation.

When using this approach, however, the model could exhibit a inelastic behavior, i.e. in the absence of forces or torques, the model would only return to the previous state, not the reference configuration. We overcome this problem by computing an elastic force $\bar{\mathbf{f}}_e$ on each beam, defined as:

$$\bar{\mathbf{f}}_e = -\gamma \bar{\mathbf{K}}_e (\bar{\mathbf{u}} - \bar{\mathbf{u}}_0) \quad (5)$$

with $0 < \gamma \leq 1$. This force is added to the external forces \mathbf{f} in the global coordinate system ($\mathbf{f}_e = \mathbf{\Lambda}^T \bar{\mathbf{f}}_e$) before solving the linear system, and it can be shown that it acts as a damping force, where γ relates to the damping coefficient of the model.

To simulate accurately a device such as a guidewire or catheter in the context of interventional neuroradiology, we need to discretize the device using a large number (from 100 to 200) of beam elements. Although solving linear systems with about 1,000 unknowns can be done in real-time using iterative methods, when integrating non-holonomic constraints, real-time computation on a single-processor workstation is no longer possible. To improve speed and handle accurately collision response, we propose the following optimizations.

2.2 Optimization using Substructures Analysis

The inverse matrix \mathbf{K}^{-1} of the stiffness matrix gives a measure of the flexibility of the model. For a force applied on one node, the induced displacement of all nodes can be obtained through the flexibility matrix. Using the flexibility representation of the behavior of the wire-like object provides also a mean

to determine the mechanical coupling that exists between nodes. As we have seen before, by using an incremental approach, the wire-like model is described by a linear relation between forces and displacements which changes at every time step. To optimize computational speed, we propose an assembling method of inverted mechanics which makes it unnecessary to assemble the mechanical properties of the devices in \mathbf{K} , therefore precluding the need to compute the whole flexibility matrix \mathbf{K}^{-1} .

Principle of Matrix substructure analysis Without loss of generality, let us assume that the nodes of the wire-like model have been ordered with the boundary (b) nodes first, followed by the internal (i) nodes. Using this ordering we can write the linear system describing the model as a block matrix system:

$$\begin{bmatrix} \mathbf{K}_{b,b} & \mathbf{K}_{b,i} \\ \mathbf{K}_{i,b} & \mathbf{K}_{i,i} \end{bmatrix} \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{bmatrix} \quad (6)$$

One can invert the system this way:

$$\begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \overbrace{[\mathbf{K}_{b,b} - \mathbf{K}_{b,i} \mathbf{K}_{i,i}^{-1} \mathbf{K}_{i,b}]^{-1}}^{\mathbf{K}_b^{-1}} & \overbrace{-\mathbf{K}_b^{-1} \mathbf{K}_{b,i} \mathbf{K}_{i,i}^{-1}}^{-\mathbf{H}_{b,i}} \\ \underbrace{-\mathbf{K}_{i,i}^{-1} \mathbf{K}_{i,b} \mathbf{K}_b^{-1}}_{\mathbf{H}_{i,b}} & \mathbf{K}_{i,i}^{-1} + \mathbf{H}_{i,b} \mathbf{K}_b^{-1} \mathbf{H}_{b,i} \end{bmatrix} \begin{bmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{bmatrix} \quad (7)$$

Previous work in the field on real-time deformable models use this result [3] to reduce computation when $\mathbf{f}_i = 0$ or only \mathbf{u}_b needs to be computed. This is often referred as condensation methods. Here, we use the inverted system in a different way. We decompose the resolution of the entire system in three steps:

- $\mathbf{u}_i^{(1)} = \mathbf{K}_{i,i}^{-1} \mathbf{f}_i$ and $\mathbf{f}_b^{(1)} = \mathbf{H}_{b,i} \mathbf{f}_i$. This step is named *boundary fixation*, because it computes the internal motion and the subsequent reaction on the boundary as if the motion of boundary was frozen.
- $\mathbf{u}_b = \mathbf{K}_b^{-1} (\mathbf{f}_b + \mathbf{f}_b^{(1)})$ and $\mathbf{u}_i^{(2)} = \mathbf{H}_{i,b} \mathbf{u}_b$. This step is named *boundary relaxation* because it computes the boundary motion due to forces on boundary and reaction forces computed during the first step. Internal displacement due to the motion of boundary is also computed.
- $\mathbf{u}_i = \mathbf{u}_i^{(1)} + \mathbf{u}_i^{(2)}$ and $(\mathbf{K}^{-1})_{i,i} = (\mathbf{K}_{i,i})^{-1} + \mathbf{H}_{i,b} \mathbf{K}_b^{-1} \mathbf{H}_{b,i}$. This step represents the *flexibility assembling* because it sums the displacements of internal node computed during step 1 and step 2.

In the case of a serially-linked structure, as the matrix \mathbf{K} is a band matrix, we can use this substructure decomposition in an accumulative way. It means that we use this methodology by sequentially applying the substructure analysis equations to all the nodes, from the base to the tip (forward), and then from the tip to the base (backward). By first processing the nodes in the forward direction, and using boundary fixation, we accumulate the reaction forces on the boundaries. By processing then the nodes in the backward direction, and using boundary relaxation, we accumulate the displacements.

Boundary fixation by accumulation We start by considering that the first node (for instance the base node) is an internal node, and that all other nodes are on the boundary. As the only boundary node directly connected to the first node is the second node, we can compute:

$$\mathbf{u}_1^{(1)} = \mathbf{K}_{1,1}^{-1} \mathbf{f}_1 \quad (8)$$

$$\mathbf{f}_{acc} = \mathbf{f}_2^{(1)} = \mathbf{H}_{2,1} \mathbf{f}_1 \quad (9)$$

These equations refer to the first step of the substructure analysis (see also Figure 2(a)) with \mathbf{f}_{acc} the accumulation force that contains the reaction force on the boundary due to internal force \mathbf{f}_1 . We also compute the new apparent stiffness of boundary node 2 using:

$$\mathbf{K}_{2,2} = \mathbf{K}_{2,2} - \mathbf{K}_{2,1} \mathbf{K}_{1,1}^{-1} \mathbf{K}_{1,2} \quad (10)$$

Then, as the influence of internal node 1 has been applied to boundary node 2, we repeat this scheme by considering node 2 as an internal node, and nodes [3...N] as boundary nodes, and so on. More generally, for a step where node n is considered an internal node, node $n+1$ will be the only boundary node directly connected to it, therefore leading to the following set of equations:

$$\mathbf{u}_n^{(1)} = \mathbf{K}_{n,n}^{-1} (\mathbf{f}_n + \mathbf{f}_{acc}) \quad (11)$$

$$\mathbf{f}_{acc} = \mathbf{H}_{n+1,n} (\mathbf{f}_n + \mathbf{f}_{acc}) \quad (12)$$

$$\mathbf{K}_{n+1,n+1} = \mathbf{K}_{n+1,n+1} - \mathbf{K}_{n+1,n} \mathbf{K}_{n,n}^{-1} \mathbf{K}_{n,n+1} \quad (13)$$

Boundary relaxation by accumulation When we reach the last node N , \mathbf{f}_{acc} represents the reaction force on N due to the force on all internal nodes (i.e. all the nodes except the last one). From this we can compute the motion of the boundary, according to the boundary relaxation step (see Figure 2(b)):

$$\mathbf{u}_N = \mathbf{K}_{N,N}^{-1} (\mathbf{f}_N + \mathbf{f}_{acc}) \quad (14)$$

We can now evaluate the motion on the internal nodes due to the motion of the boundary last node N :

$$\mathbf{u}_{N-1}^{(2)} = \mathbf{H}_{N-1,N-1} \mathbf{u}_N \quad (15)$$

Flexibility assembling by accumulation In order to compute the total displacement of $N-1$, we use the third step of the substructure analysis.

$$\mathbf{u}_{N-1} = \mathbf{u}_{N-1}^{(1)} + \mathbf{u}_{N-1}^{(2)} \quad (16)$$

Using flexibility assembling, we can compute the value of \mathbf{K}^{-1} for the node $N-1$:

$$\mathbf{K}_{N-1,N-1}^{-1} = \mathbf{H}_{N-1,N} \mathbf{K}_{N,N}^{-1} \mathbf{H}_{N,N-1} \quad (17)$$

When $N-1$ is computed, we then consider it as the boundary of all previous nodes, and compute $N-2$, and repeat until all nodes are processed.

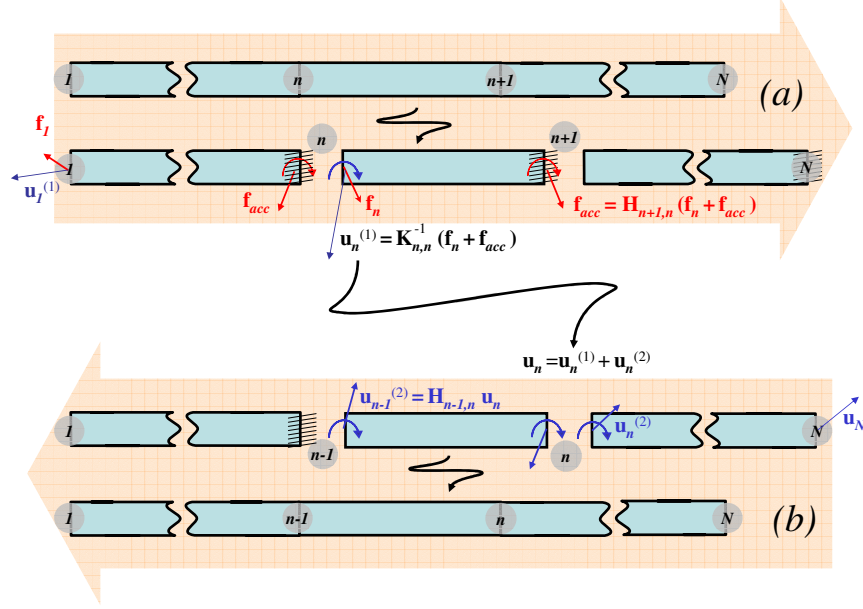


Fig. 2. (a) *Setting boundary conditions:* the object is split in a series of substructures, and local displacements $\mathbf{u}_n^{(1)}$ and force \mathbf{f}_{acc} are computed after constraining the boundary node of each substructure; (b) *Relaxing boundary conditions:* correction displacements $\mathbf{u}_n^{(2)}$ are applied recursively, starting from node N , at each internal node of each substructure.

Algorithm 1: System solving using substructure analysis

Input: \mathbf{K}, f
Output: \mathbf{u} ($= \mathbf{K}^{-1} f$)
 $\mathbf{f}_{acc} = 0;$
for $n = 1 \dots N - 1$ **do**
 $\mathbf{u}_n^{(1)} = (\mathbf{K}_{n,n})^{-1} (\mathbf{f}_n + \mathbf{f}_{acc});$
 $\mathbf{f}_{acc} += \mathbf{K}_{n+1,n} \mathbf{u}_n^{(1)}$
 $\mathbf{K}_{n+1,n+1} -= \mathbf{K}_{n+1,n} (\mathbf{K}_{n,n})^{-1} \mathbf{K}_{n,n+1}$
end
 $\mathbf{u}_N = (\mathbf{K}_{N,N})^{-1} (\mathbf{f}_N + \mathbf{f}_{acc});$
for $n = N - 1 \dots 1$ **do**
 $\mathbf{u}_n^{(2)} = (\mathbf{K}_{n,n})^{-1} \mathbf{K}_{n,n+1} \mathbf{u}_{n+1};$
 $\mathbf{u}_n = \mathbf{u}_n^{(1)} + \mathbf{u}_n^{(2)};$
end

Algorithm 2: Flexibility computation using substructure analysis

Input: \mathbf{K}
Output: $[(\mathbf{K}^{-1})_{1,1} \dots (\mathbf{K}^{-1})_{N,N}]$,
 $[\mathbf{H}_{1,2} \dots \mathbf{H}_{N,N-1}]$,
 $[\mathbf{H}_{2,1} \dots \mathbf{H}_{N-1,N}]$
for $n = 1 \dots N - 1$ **do**
 $(\mathbf{K}^{-1})_{n,n} = (\mathbf{K}_{n,n})^{-1}$;
 $\mathbf{K}_{n+1,n+1} = \mathbf{K}_{n+1,n} (\mathbf{K}_{n,n})^{-1} \mathbf{K}_{n,n+1}$;
 $\mathbf{H}_{n+1,n} = \mathbf{K}_{n+1,n} (\mathbf{K}_{n,n})^{-1}$;
end
 $(\mathbf{K}^{-1})_{N,N} = (\mathbf{K}_{N,N})^{-1}$;
for $n = N - 1 \dots 1$ **do**
 $\mathbf{H}_{n,n+1} = (\mathbf{K}_{n,n})^{-1} \mathbf{K}_{n,n+1}$;
 $(\mathbf{K}^{-1})_{n,n} = \mathbf{H}_{n,n+1} (\mathbf{K}^{-1})_{n+1,n+1} \mathbf{H}_{n+1,n}$
end

2.3 Contact Response

Another important problem in simulating catheter or guidewire navigation is solving the contacts between the virtual device and the vessels wall. Because sliding occurs at the point of contact, Lagrange multiplier techniques, simple penalty forces or quadratic programming approaches will not constrain the flexible body properly. To solve this problem, we compute the local compliance of the node in the contact space defined by \mathbf{n}_α , where \mathbf{n}_α is the normal at contact point. The local compliance of the contact α , at node n , is:

$$W_{\alpha,\alpha} = [\mathbf{n}_\alpha]^T (\mathbf{K}^{-1})_{n,n} [\mathbf{n}_\alpha] \quad (18)$$

The computation of $W_{\alpha,\alpha}$ is very efficient, since $(\mathbf{K}^{-1})_{n,n}$ is already known from algorithm 2. The problem of multiple sliding contact can be solved using a Gauss-Seidel like algorithm [5]. Considering a contact α , among m instantaneous contacts, one can write the behavior of the model in contact space:

$$\underbrace{\delta_\alpha - [W_{\alpha,\alpha}]f_\alpha}_{\text{unknown}} = \underbrace{\sum_{\beta=1}^{\alpha-1} [W_{\alpha,\beta}]f_\beta + \sum_{\beta=\alpha+1}^m [W_{\alpha,\beta}]f_\beta}_{\text{frozen}} + \delta_\alpha^{\text{free}} \quad (19)$$

where $[W_{\alpha,\beta}]$ is a compliance matrix that models the coupling between contact points α and β . For each contact α , this method solves the contact equations by considering the others contact points ($\alpha \neq \beta$) as "frozen". The new value of f_α is given by solving the Signorini's law on this contact:

$$0 \leq \delta_\alpha \perp f_\alpha \geq 0 \quad (20)$$

Let us consider the current state of a contact $(\delta_\alpha, f_\alpha)$. By removing the current contribution of the contact force f_α , we obtain a new value of δ_α that we call the

free status of the contact ($\delta_\alpha^{\text{free}}, f_\alpha = 0$). If the status shows an interpenetration ($\delta_\alpha^{\text{free}} < 0, f_\alpha = 0$), then, we should correct the force in order to solve the contact ($\delta_\alpha = 0, f_\alpha > 0$). Otherwise, the free status of the contact is correct ($\delta_\alpha \geq 0, f_\alpha = 0$). See algorithm 3 for implementation.

Algorithm 3: Signorini law

Input: W, δ, f
Output: f
 $\delta^{\text{free}} = \delta - Wf;$ /* δ^{free} gives the state of contact */
if $\delta^{\text{free}} < 0$ **then**
 $f = -\frac{\delta^{\text{free}}}{W};$ /* contact */
else
 $f = 0;$ /* no contact */
end

One of the key outcomes of the substructure analysis described above is the acceleration it provides for the computation of the right term of equation (19). We can rewrite equation (19) to show the influence of contacts other than α , and use the substructure decomposition to compute the right term of the equation:

$$\delta_\alpha - [W_{\alpha,\alpha}]f_\alpha = [\mathbf{n}_\alpha]^T \underbrace{\left[(\mathbf{K}^{-1})_{n,n} \mathbf{f}_n + \sum_{i=1}^{n-1} (\mathbf{K}^{-1})_{n,i} \mathbf{f}_i + \sum_{i=n+1}^N (\mathbf{K}^{-1})_{n,i} \mathbf{f}_i \right]}_{\text{computed by substructure decomposition}} + \delta_\alpha^{\text{free}} \quad (21)$$

Let us consider that M is the total number of contacts and N the total number of nodes, introducing substructure analysis in the Gauss-Seidel algorithm (see Algorithm 4) reduces the complexity of one iteration from $\mathcal{O}(M^2)$ to $\mathcal{O}(M + N)$ for an identical result.

3 Interventional Neuroradiology Simulation

In addition to the models and algorithms described previously, we have addressed several other aspects of the simulation of interventional radiology procedures. We briefly describe here these results, more details can be found in [13].

A first problem we addressed concerns arterial flow computation, which impacts both catheter navigation and contrast agent propagation. Blood flow in each vessel is modeled as an incompressible viscous fluid flowing through a cylindrical pipe, and can be calculated from a simplified Navier-Stokes equation called Poiseuille's Law (22). It relates the flow Q in each vessel to the pressure gradient ΔP , viscosity of the fluid η , radius r , and length L of the vessel.

$$Q = \frac{\Delta P}{R} \quad \text{with} \quad R = \frac{8\eta L}{\pi r^4} \quad (22)$$

Algorithm 4: Substructure Gauss-Seidel algorithm

```
Input:  $[(\mathbf{K}^{-1})_{1,1} \dots (\mathbf{K}^{-1})_{N,N}]$ ,  
          $[\mathbf{H}_{1,2} \dots \mathbf{H}_{N,N-1}]$ ,  $[\mathbf{H}_{2,1} \dots \mathbf{H}_{N-1,N}]$ ,  
          $[\dots, \mathbf{n}_\alpha, \dots]$ ,  $[\dots, \delta_\alpha^{\text{free}}, \dots]$   
Output:  $\mathbf{u}$ ,  $\mathbf{f}$   
 $Df = \infty$ ; /* convergence criterion */  
 $[\dots, f_\alpha, \dots] = 0$ ;  
while  $Df > \varepsilon$  do  
     $Df = 0$ ;  
     $\mathbf{f}_{acc} = 0$ ;  
    for  $n = 1 \dots N - 1$  do  
         $\mathbf{f}_{acc} += \mathbf{f}_n$ ;  
         $\mathbf{u}_n = (\mathbf{K}^{-1})_{n,n} \mathbf{f}_{acc}$ ; ( $\mathbf{u}_n$  includes contributions of  $[\mathbf{f}_1 \dots \mathbf{f}_{n-1}]$ )  
  
        /* contact solving part */  
         $\Delta \mathbf{f}_n = 0$ ;  
        forall contact  $\alpha$  on node  $n$  do  
             $\delta_\alpha = [\mathbf{n}_\alpha]^T [\mathbf{u}_n] + \delta_\alpha^{\text{free}}$ ;  
             $f_\alpha^{\text{new}} = \text{SignoriniLaw}(W_{\alpha\alpha}, \delta_\alpha, f_\alpha)$  ;  
             $\mathbf{u}_n += (\mathbf{K}^{-1})_{n,n} [\mathbf{n}_\alpha] [f_\alpha^{\text{new}} - f_\alpha]$  ;  
             $\Delta \mathbf{f}_n += [\mathbf{n}_\alpha] [f_\alpha^{\text{new}} - f_\alpha]$  ;  
             $f_\alpha = f_\alpha^{\text{new}}$  ;  
        end  
         $\mathbf{f}_{acc} += \Delta \mathbf{f}_n$ ;  
         $Df += \|\Delta \mathbf{f}_n\|$ ;  
        /* end of contact solving part */  
  
         $\mathbf{f}_{acc} = \mathbf{H}_{n+1,n} \mathbf{f}_{acc}$ ;  
    end  
     $\mathbf{u}_N = (\mathbf{K}^{-1})_{N,N} (\mathbf{f}_N + \mathbf{f}_{acc})$ ;  
     $\mathbf{u}_{acc} = (\mathbf{K}^{-1})_{N,N} \mathbf{f}_N$ ;  
    for  $n = N - 1 \dots 1$  do  
         $\mathbf{u}_{acc} += \mathbf{H}_{n,n+1} \mathbf{u}_{acc}$ ;  
         $\mathbf{u}_n += \mathbf{u}_{acc}$ ; (correction: contributions of  $[\mathbf{f}_{n+1} \dots \mathbf{f}_N]$ )  
         $\mathbf{u}_{acc} += (\mathbf{K}^{-1})_{n,n} \mathbf{f}_n$ ;  
    end  
     $Df /= \|\mathbf{f}\|$ ;  
end
```

Solving this equation for the whole vascular system is equivalent to solving a linear FEM model, and, assuming there is no deformation of the vessels, the global resistance matrix $[R]$ can be pre-inverted, allowing for real-time simulation of vascular flow. An example of flow computation is illustrated in Figure 4.

Two other important aspects which relate to visual feedback are X-ray simulation and contrast agent propagation. Both are used during interventional procedures to visualize blood vessels, as well as the anatomy. Contrast agent propagation is simulated using an advection equation where the concentration

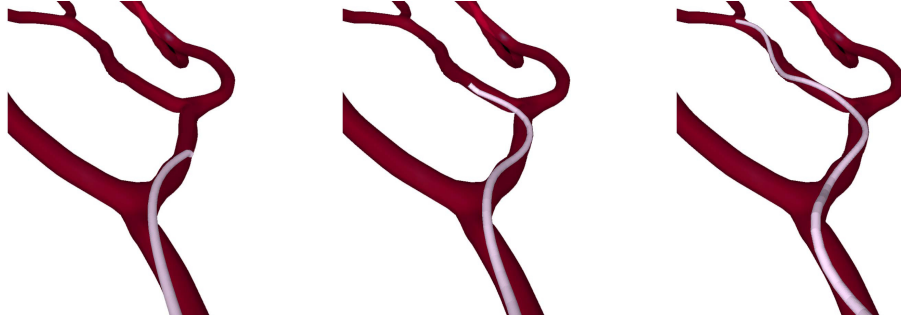


Fig. 3. Catheter navigation inside the cerebrovascular network. Complex, non-linear deformations are correctly represented by the incremental FEM model. Collision detection and collision response allow the catheter to stay within the lumen of the vessels.

of contrast agent $C(x, t)$ is a function of the injection rate $r(t)$ and the averaged laminar flow velocity $u(x, t)$.

$$\frac{\partial C(x, t)}{\partial t} + u(x, t) \frac{\partial C(x, t)}{\partial x} = r(t) \quad (23)$$

We numerically solve (23) using a finite difference scheme. The resulting angiogram is visualized using 3D particles which intensity values are a function of $C(x, t)$. Similarly, we have developed a new volume rendering approach that can render CT datasets as fluoroscopic or X-ray images, in real-time. The X-ray attenuation process simulates X-ray beam attenuation as described by a discretized Beer's Law [13]. This method greatly reduces computation times by using OpenGL and specific texture map operations.

Finally, we propose a multi-modal representation of the vascular system, generated from skeletonization of patient specific CTA datasets, as a mean to ensure consistency in the simulation. The segmentation task is particularly challenging due to the small vessel diameter and the close proximity of vessels to the skull, and thus remain semi-automatic. After applying a combination of anisotropic filtering and morphological operators, medial axis and local radius information is computed. This information is then used to automatically generate a graph of the vascular network for fluid flow computation, a set of 3D particles for contrast agent rendering, a discretization of the centerlines for contrast agent numerical computation, and a C1-continuous triangulation of the surface of the vasculature optimized for fast collision detection. An example of segmented vascular network is presented in Figure 4.

4 Conclusions and Future Work

In this paper we propose a series of new approaches for fast and accurate simulation of catheter or guidewire navigation within a vascular network. The model

we introduce for simulating wire-like structures is able to represent the complex behavior of medical devices and handles collision responses in an effective manner. We also describe results related to the development of a full simulator for interventional neuroradiology. The system runs in real-time on a single processor machine and is capable of simulating realistically key aspects of a diagnostic procedure. Current results for a 100 node model show a computation time of 25 ms for one time step (including the computation of $[\mathbf{K}]$, substructure analysis, collision detection, and contact response) on a Pentium 4 2.6 GHz processor. Going forward, we will address the issue of simulating devices such as stents and coils, and develop a set of metrics for measuring performance during training.



Fig. 4. *Left:* Color-coded vascular model obtained from patient data, warmer colors correspond to increased blood flow; *Center:* Contrast agent injection using particle-based rendering; *Right:* Real-time catheter navigation and fluoroscopic rendering.

This work was supported by the Department of Defense and CIMIT (grant numbers: 17-02-2-0006 and 21-45-22). We thank Drs. Vincent Luboz, Xunlei Wu, Karl Krissian and James Rabinov for their valuable help on anatomical segmentation.

References

1. T. Alderliesten. *Simulation of Minimally-Invasive Vascular Interventions for Training Purposes*. PhD dissertation, Utrecht University, December 2004.
2. American Heart Association. Heart and stroke facts statistics: Statistical supplement. American Heart Association, Dallas, Texas, 1999.
3. M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Computer Graphics Forum*, volume 15, pages 57–66. Eurographics, 1996.
4. S. Cotin, S. Dawson, D. Meglan, D. Shaffer, M. Ferrell, and P. Sherman. Icts, an interventional cardiology training system. In J.D. Westwood et al., editor, *Proceedings of Medicine Meets Virtual Reality*, pages 59–65. IOS Press, 2000.

5. C. Duriez, C. Andriot, and A. Kheddar. Signorini's contact model for deformable objects in haptic simulations. In *IEEE-IROS*, pages 3232–3237, 2004.
6. R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research*, 2(1):13–30, 1983.
7. C. A. Felippa. A systematic approach to the element independent corotational dynamics of finite elements. Technical Report CU-CAS-00-03, Center for Aerospace Structures, 2000.
8. U. Hofer, T. Langen, J. Nziki, F. Zeitler, J. Hesser, U. Mueller, W. Voelker, and R. Maenner. Cathi - catheter instruction system. In *Computer Assisted Radiology and Surgery (CARS)*, pages 101 – 06, Paris, France, 2002.
9. J. Lenoir, P. Meseure, L. Grisoni, and C. Chaillou. Surgical thread simulation. In *MS4CMS (Proc. of ESAIM)*, volume 12, pages 102–107, 2002.
10. Matthieu Nesme, Yohan Payan, and François Faure. Efficient, physically plausible finite elements. In J. Dingliana and F. Ganovelli, editors, *Eurographics (short papers)*, august 2005.
11. W.L. Nowinski and C.K. Chui. Simulation of interventional neuroradiology procedures. In *MIAR*, pages 87 – 94, 2001.
12. J.S. Przemieniecki. *Theory of Matrix Structural Analysis*. 1968.
13. X. Wu, V. Pegoraro, V. Lubos, P. Neumann, R. Bardsley, S. Dawson, and S. Cotin. New approaches to computer-based interventional neuroradiology training. In James D. Westwood et al., editor, *Proceedings of MMVR*, pages 602–607, 2005.