Real-time Error Control for Surgical Simulation

Huu Phuoc Bui, Satyendra Tomar, Hadrien Courtecuisse, Stéphane Cotin, Stéphane Bordas

HAL Id: hal-01514621

https://hal.inria.fr/hal-01514621

Submitted on 26 Apr 2017

Real-time Error Control for Surgical Simulation

Huu Phuoc Bui*†, Satyendra Tomar†, Hadrien Courtecuisse*, Stéphane Cotin‡ and Stéphane Bordas†§

*University of Strasbourg, CNRS, ICube, F-67000 Strasbourg, Fance

†University of Luxembourg, Research Unit of Engineering Science, L-1359 Luxembourg, Luxembourg

‡Inria Nancy Grand Est, 54603 Villers-les-Nancy, France

§Cardiff University, School of Engineering, Queens Buildings, The Parade, Cardiff CF24 3AA, Wales, UK

Abstract—Objective: **To present the first real-time a posteriori error-driven adaptive finite element approach for real-time simulation and to demonstrate the method on a needle insertion problem.** *Methods*: **We use corotational elasticity and a frictional needle/tissue interaction model. The problem is solved using finite elements within SOFA[1]. The refinement strategy relies upon a hexahedron-based finite element method, combined with a posteriori error estimation driven local h-refinement, for simulating soft tissue deformation.** *Results*: **We control the local and global error level in the mechanical fields (e.g. displacement or stresses) during the simulation. We show the convergence of the algorithm on academic examples, and demonstrate its practical usability on a percutaneous procedure involving needle insertion in a liver. For the latter case, we compare the force displacement curves obtained from the proposed adaptive algorithm with that obtained from a uniform refinement approach.** *Conclusions*: **Error control guarantees that a tolerable error level is not exceeded during the simulations. Local mesh refinement accelerates simulations.** *Significance*: **Our work provides a first step to discriminate between discretization error and modeling error by providing a robust quantification of discretization error during simulations.**

Index Terms—**Finite element method, real-time error estimate, adaptive refinement, constraint-based interaction.**

I. Introduction

A. Motivation

Real-time simulations are becoming increasingly common for various applications, from geometric design [1], [2] to medical simulation [3]. Our focus is on real-time simulation of the interaction of a surgeon or interventional radiologist with deformable organs. Such simulations are useful to both, help surgeons train, rehearse complex operations or/and to guide them during the intervention. In time, reliable simulations could also be central to robotic surgery.

A number of factors are concurrently involved in defining the "accuracy" of surgical simulators: mainly the modeling error and the discretization error. Most work in the area has been looking at the above sources of error as a compounded, lumped, overall error. Little or no work has been done to discriminate between modeling error (e.g. needle-tissue interaction, choice of constitutive models) and discretization error (use of approximation methods like FEM). However, it is impossible to validate the complete surgical simulation approach and, more importantly, to understand the sources of

error without evaluating both the discretization error and the modeling error.

The first ingredient in any mechanical simulation is the ability to simulate the deformation of the solid of interest. This deformable solid mechanics problem is usually solved by finite element method (FEM) [4] or meshless/meshfree methods [5], which are used to discretize the equilibrium equations. It is usually uneconomical or prohibitively expensive to use a fixed mesh for such simulations. Indeed, coarse meshes are sufficient to reproduce "smooth" behavior, whereas "non-smooth" behavior such as discontinuities engendered by cuts or material interfaces, singularities, boundary layers or stress concentrations require a finer mesh. Adequate approaches are thus needed to refine the discretization in these areas.

Yet, existing numerical methods used in surgical simulation use either a fixed discretization (finite element mesh, meshfree point cloud, reduced order method), or adapt the mesh using heuristics [6], [7]. To our knowledge, no approach is currently able to adapt the finite element mesh based on rational a posteriori error estimates [8].

Our objective is thus to devise a robust and fast approach to local remeshing for surgical simulations. To ensure that the approach can be used in clinical practice, the method should be robust enough to deal, as realistically as possible, with the interaction of surgical tools with the organ, and fast enough for real-time simulations. The approach should also lead to an improved convergence so that an "economical" mesh is obtained at each time step. The final goal is to achieve optimal convergence and the most economical mesh, which will be studied in our future work.

In this paper, we propose and benchmark a local mesh refinement and coarsening approach which is based on the estimation of the discretization error incurred by FEM in the solution of a corotational model representing soft tissues. The general ideas presented here can be used directly in geometric design based on deformable models. Our proposed approach has similitudes with the octree approaches of [9].

B. Error in numerical simulations

It is useful to first review the various sources of error in numerical simulations. The first error source arises when a mathematical model is formulated for a given physical problem: this is known as the modeling error. The second error arises upon discretization of this mathematical model, for example using FEM or meshfree methods. Finally, numerical

error is incurred because of the finite precision of computers and round off errors. In this paper, we focus on the second source of error, namely discretization error. We therefore assume that the model we use is descriptive of reality, i.e. we are solving the right problem, and we ask ourselves the question whether "we are solving the problem right", in other words, correctly.

The main difficulty in answering this question comes from the fact that an exact solution, to which the numerical solution could be compared, is generally not available. Different approaches exist to address this problem, which are reviewed in the literature, see e.g., [4], [8], [10]. Simple methods available in practice to indicate the error distribution can be categorized into two classes: recovery-based and residual-based.

The first class of indicators assumes that the exact solution (of stresses) is smooth enough (at least locally). They rely on the construction of an "improved" numerical solution from the raw numerical solution, to which the raw numerical solution can be compared. Where these two solutions are significantly different (above certain threshold), the error level is high and the mesh should be refined, and where these two solutions are close together, the mesh can be kept unchanged or coarsened. This idea was proposed by Zienkiewicz and Zhu in [11], and its asymptotic convergence to the exact error is studied in [12], [13].

The second class of indicators relies on the computation of the residual of the governing equations within each computational cell, typically each finite element. These "residual-based" error indicators lead to mesh refinement where the solution leads to large residuals, and keep the mesh constant or coarsen it where the element residuals are relatively small, compared to a given tolerance. These estimates were first proposed by Babuska and Rheinboldt in [14].

Based on these local error estimators, mesh refinement methodologies can be devised to derive mesh adaptation, see e.g. [10] for a recent comprehensive presentation. This requires two key ingredients: *a marking strategy* that decides which elements should be refined, and *a refinement rule* that defines how the elements are subdivided. For element marking, we use the *maximum strategy*, see Section III for details. Other strategies, such as bulk/equilibration strategy or percentage strategy, see e.g. [10], can also be used.

C. Simulation of percutaneous operations

Needle-based percutaneous procedures are an important part of modern clinical interventions such as biopsy, brachytherapy, cryotherapy or regional anesthesia. The success of these procedures depends on good training and careful planning to optimize the path to the target, while avoiding critical structures [15]. In some instances the procedure can also be assisted by robotic devices. Unfortunately, natural tissue motion (due to breathing, for instance), and deformation (due to needle insertion) generally lead to incorrect or inefficient planning [15]. To address these issues, one must rely on an accurate simulation of needle insertion. For most problems, computational speed is also very important, since the simulation is at the core of an optimization algorithm (for the needle path) or a robotic control loop.

The main works on needle insertion (see the survey by [16]) propose to model the interaction between the needle and soft tissues using FEM. In the various methods proposed in the literature, three main research directions have been followed: soft tissue model, flexible needle model and needle-tissue interactions. The needle model is usually not an issue, both in terms of modeling choice and computational cost. For instance, in [17], authors report computation times of a few milliseconds for a FEM needle model composed of 50 serially-linked Timoshenko beam elements. Soft tissue models are usually based on FEM, and rely on linear or non-linear constitutive laws [16].

A large body of work covers the modeling and simulation of soft tissue deformation, even under real-time computation constraints. But overall, the interaction model between the needle and tissue remains a major challenge. It combines different physical phenomena, such as puncturing, cutting, sliding with friction and Poynting's effect. To capture the essential characteristics of these interactions, existing methods usually rely on experimental force data and remeshing techniques in order to align nodes of the FEM mesh with the needle path. In [17], a constraint-based approach, avoiding remeshing, was used to simulate needle-tissue interactions. However, the simulations did not account for realistic anatomical details. In addition, Misra *et al.* [18] showed that needle steering, which occurs when using asymmetric needle tips, can be modeled using microscopic observations of needle-tissue interactions.

Unfortunately, if no assumption can be made about the region of the domain where the needle will be inserted, simulations involving very detailed meshes become very slow, which is a real issue in the context presented above. Error-controlled real-time simulation of needle insertion is thus an unsolved problem whose solution requires tackling a number of difficulties:

- developing needle-tissue models. A review of cutting simulation is provided in [19]
- using these models within discrete approaches like FEM, mesh-free methods, or others
- accelerating the simulation (advanced hardware, model order reduction)
- validating the needle-tissue interaction model combined with discrete solution (are we solving the right problem?)
- verifying the discrete solution, i.e. controlling the discretization error associated with the discrete model (are we solving the problem right?)

In this paper, we propose to focus on the last point above, with the aim to model needle-tissue interactions using an adaptive meshing strategy driven by simple a posteriori error estimation techniques. Similarly to [17], we do not require the mesh to conform to the needle path. Mesh subdivision is only introduced as a means to improve the accuracy of the needle-tissue interactions. Our mesh refinement method is guided by the stress/energy error estimate resulting from the needle-tissue interaction and imposed boundary conditions: elements of the mesh are subdivided when a numerical error threshold is reached. The subdivision process is completely reversible, i.e. refined elements are set back to their initial

topology when refinement is no longer needed. Our refinement approach does not rely on a usual octree structure (see also [20]), thus allowing a variety of subdivision schemes that are well suited for needle insertions, as detailed in Section III. Using this approach, interactive computation times can be achieved while detailed tissue motion near the needle shaft or tip can be computed. This opens new possibilities for fast simulations of flexible needle insertion in soft tissues. We illustrate the convergence study of the adaptive refinement scheme and some of the possible scenarios in Section V.

II. MODEL AND DISCRETIZATION

In this section, we describe the model and the discretization approach, which are used for needle and soft tissue interaction.

A. Problem statement

During the needle insertion, three types of constraints are defined, see Fig. 1. Coulomb's friction law is used to describe
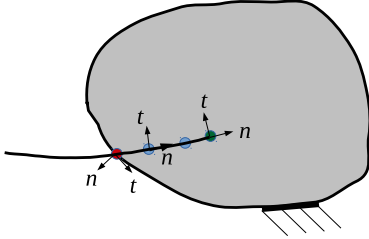

Fig. 1: Three types of constraints between the needle and soft tissue: surface puncture (in red), needle tip constraint (in green) and needle shaft constraints (in blue). A local coordinate system n-t is defined at each constraint point.

frictional contacts within these three types of constraints. First, a puncture constraint is defined between the needle tip and the tissue surface. This constraint satisfies the Kuhn-Tucker condition in the direction n (normal to the tissue surface)

$$\delta_n \geq 0, \qquad \lambda_n \geq 0, \qquad \delta_n \cdot \lambda_n = 0, \tag{1}$$

where δ_n denotes the distance between the needle tip and the tissue surface in the direction n, and λ_n denotes the contact force in that direction. Let λ_{n0} represent the puncture strength of the tissue. The Kuhn-Tucker condition expresses that the contact force only exists when the needle tip is in contact with the tissue surface. When the contact force is higher than a threshold (puncture strength of the tissue)

$$\lambda_n > \lambda_{n0}, \tag{2}$$

the needle can penetrate into the tissue. In the tangent direction t, Coulomb's friction law is considered in order to take into account the stick/slip between the needle tip and tissue surface

$$\lambda_t < \mu\lambda_n \quad \text{(stick)}; \quad \lambda_t = \mu\lambda_n \quad \text{(slip)}, \tag{3}$$

where μ denotes the friction parameter.

Second, a needle tip constraint is defined at the tip of the needle as soon as it penetrates into the tissue. Depending on the relationship between the contact forces in the normal direction n (along the needle shaft) and in the tangent direction

t (see Fig. 1), the needle tip can cut and go through the tissue or not

$$\lambda_n < \mu\lambda_t + \lambda_{n0} \text{ (stick)}; \quad \lambda_n \geq \mu\lambda_t + \lambda_{n0} \text{ (cut and slip)}. \tag{4}$$

Finally, needle shaft constraints are defined along the needle shaft so that the needle shaft is enforced to follow the insertion trajectory created by the advancing needle tip. Again, the Coulomb's friction law is applied to these constraints to represent the stick and sliding contact between the tissue and the needle shaft

$$\lambda_n < \mu\lambda_t \text{ (stick)}; \quad \lambda_n = \mu\lambda_t \text{ (sliding)}. \tag{5}$$

B. Strong form

We model both, the tissue and the needle, as dynamic deformable objects. Thus, they can be regarded as dynamic elastic solids, and the governing equations of the model are formulated as

$$\text{div}\,\boldsymbol{\sigma} + \mathbf{b} + \boldsymbol{\lambda} = \rho\ddot{\mathbf{u}} \qquad \text{in } \Omega \tag{6a}$$

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\text{grad}\,\mathbf{u} + (\text{grad}\,\mathbf{u})^T\right) \tag{6b}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \boldsymbol{\nu}) \tag{6c}$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t; \tag{6d}$$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u, \tag{6e}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\mathbf{b}$ is the body force vector, ρ is the mass density, $\boldsymbol{\epsilon}$ is the strain tensor, $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_n)$ is the vector of internal variables, $(\dot{})$ denotes the partial derivative with respect to time, $\mathbf{n}$ denotes the outward unit normal vector on Γ_t, and $\boldsymbol{\lambda}$ denotes the contact force between the needle and the tissue. The object domain and boundary conditions are shown in Fig. 2a.
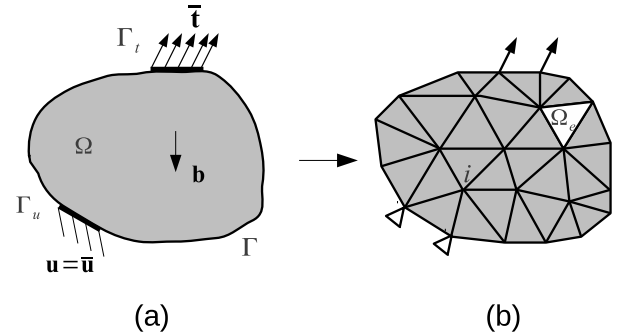

Fig. 2: A body Ω subjected to a traction $\bar{\mathbf{t}}$ on its boundary part Γ_t, a body force $\mathbf{b}$, and an imposed displacement $\bar{\mathbf{u}}$ on boundary part Γ_u (a); Simplified illustration of FEM discretization (b).

C. Spatial and temporal discretization

1) Space discretization: The basic idea of FEM is to discretize the domain Ω into finite elements Ω_e, $e = 1, 2, \ldots, N_e$, by N_n nodes, as depicted in Figure 2b. Based on the discretization concept, see e.g., [21], [22], we obtain the discrete problem of the element e

$$\mathbf{M}_e\ddot{\mathbf{u}}_e + \mathbf{C}_e\dot{\mathbf{u}}_e + \mathbf{f}_e(\boldsymbol{\sigma}) = \mathbf{f}_e^{ext}, \tag{7}$$

where $\mathbf{M}_e$ is the element mass matrix, $\mathbf{K}_e$ is the element stiffness matrix, $\mathbf{C}_e$ is the damping matrix, $\mathbf{f}_e^{ext}$ is the external force applied to the element e, while the internal force $\mathbf{f}_e(\boldsymbol{\sigma})$ reads

$$\mathbf{f}_e(\boldsymbol{\sigma}) = \int_{\Omega_e} \mathbf{B}_e^T \boldsymbol{\sigma} \, d\Omega = \int_{\Omega_e} \mathbf{B}_e^T \mathbf{E} \mathbf{B}_e \mathbf{u}_e \, d\Omega \qquad (8)$$
$$= \mathbf{K}_e \cdot \mathbf{u}_e = \mathbf{K}_e \cdot (\mathbf{x}_e - \mathbf{x}_{0e})$$

where $\mathbf{B}_e$ is the strain-displacement matrix, $\mathbf{E}$ is the fourth-order stiffness tensor, $\mathbf{x}_e$ and $\mathbf{x}_{0e}$ denote the current and initial position of the element e, respectively. However, using Eq. (8) results in inaccuracy for large rotations problems, which is observed by artificially inflated deformation of the elements. To overcome this, Felippa *et al.* [23] decomposed the deformation gradient of the element into the rigid and deformation parts, and the element nodal internal force becomes

$$\mathbf{f}_e = \mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e^T \mathbf{x}_e - \mathbf{x}_{0e}), \qquad (9)$$

where $\mathbf{R}$ stands for the element rotation matrix of the element local frame with respect to its initial orientation, being updated at each time step. Using this corotational formulation results in no visual artifacts.

The global mass, stiffness and damping matrices of the system can then be assembled from the element ones, and Eq. (7) can be rewritten to a global system equation as

$$\mathbf{M}\mathbf{a} = \mathbf{f}(\mathbf{x}, \mathbf{v}), \qquad (10)$$

where $\mathbf{a} = \ddot{\mathbf{u}}$, $\mathbf{x}$, $\mathbf{v} = \dot{\mathbf{u}}$ are the acceleration, position and velocity vectors, respectively, and $\mathbf{f}(\mathbf{x}, \mathbf{v}) = \mathbf{f}^{ext} - \mathbf{K}\mathbf{u} - \mathbf{C}\mathbf{v}$ represents the net force (the difference of the external and internal forces) applied to the object.

In our simulations, a diagonally lumped mass matrix is employed, and the stiffness matrix $\mathbf{K}$ is computed based on the co-rotational FE formulation described above, which allows large rotations for both, needle as well as tissue. For higher accuracy of the computed strain field, the soft tissue domain is discretized using hexahedral elements. To avoid the complex issue of generating an exact hexahedral mesh of the domain, we use a mesh that does not conform to the boundary of the domain, as in Immersed Boundary Method [24]. The needle, on the other hand, is modeled using serially-linked beam elements, as in [17]. In this case, each node of the needle has 6 degrees of freedom (3 translations and 3 rotations), while the tissue model only uses 3 translational degrees of freedom per node.

Since the FEM formulation is based on the discretization of the physical domain, it naturally introduces the discretization error in the result. To control this error source, in Section III we present an adaptive refinement scheme.

2) Time discretization: For temporal discretization, we use an implicit backward Euler scheme [25], which is described as follows

$$\dot{\mathbf{u}}_{t+\tau} = \dot{\mathbf{u}}_t + \tau \ddot{\mathbf{u}}_{t+\tau}; \quad \mathbf{u}_{t+\tau} = \mathbf{u}_t + \tau \dot{\mathbf{u}}_{t+\tau}, \qquad (11)$$

where τ denotes the time step. Inserting Eq. (11) into Eq. (10) gives the final discrete system

$$\underbrace{(\mathbf{M} - \tau\mathbf{C} - \tau^2\mathbf{K})}_{\mathbf{A}} d\mathbf{v} = \underbrace{\tau \mathbf{f}(\mathbf{x}^t, \mathbf{v}^t) + \tau^2 \mathbf{K}\mathbf{v}^t}_{\mathbf{b}} \qquad (12)$$

where $d\mathbf{v} = \mathbf{v}_{t+\tau} - \mathbf{v}_t$.

After solving (12) for $d\mathbf{v}$, the position and velocity are updated for needle and tissue as

$$\mathbf{v}_{t+\tau} = d\mathbf{v} + \mathbf{v}_t; \quad \mathbf{x}_{t+\tau} = \mathbf{x}_t + \tau\mathbf{v}_{t+\tau}. \qquad (13)$$

D. Constraint enforcement for needle-tissue interaction

To take into account the needle-tissue interaction when they are in contact, a constrained dynamic system is solved for the needle and the tissue. Eq. (12) then becomes

$$\mathbf{A}d\mathbf{v} = \mathbf{b} + \mathbf{J}\boldsymbol{\lambda}, \qquad (14)$$

where $\boldsymbol{\lambda}$ denotes Lagrange multipliers representing the interaction forces between the needle and the tissue, and $\mathbf{J}$ provides the direction of the constraints. Different types of constraints between needle and tissue are used, and solving their interaction is detailed in Section IV.

Remark Combining more advanced and clinically relevant needle-tissue interaction is straightforward in our approach.

III. ERROR ESTIMATE AND ADAPTIVE REFINEMENT

To achieve faster and more accurate FEM simulations, different adaptive techniques have been proposed in the literature. Octree-based approaches [26] are the most common, but the refinement procedure is limited to cubic elements which are recursively subdivided into eight finer elements. To overcome this limitation, more generic remeshing techniques [27]–[29] have been proposed. However, they are complex to implement, and may lead to ill-shaped elements. Our template-based refinement algorithm is designed to be independent of the type of element (tetrahedra, hexahedra, others), and produces a high quality mesh (thanks to the well-shaped elements of the predefined template).

Starting with an initial, relatively coarse mesh (as required to achieve real-time simulation), a criterion based on a posteriori error estimate is evaluated to drive the local refinement. The elements where the stress increases, i.e. $d\boldsymbol{\sigma}/dt > 0$, are considered for refinement, and the elements where the stress decreases, i.e. $d\boldsymbol{\sigma}/dt < 0$, are taken to a lower refinement (coarsening) level. We define the approximate error of an element Ω_e as

$$\eta_e = \sqrt{\int_{\Omega^e} (\boldsymbol{\epsilon}^h - \boldsymbol{\epsilon}^s)^T (\boldsymbol{\sigma}^h - \boldsymbol{\sigma}^s) d\Omega}, \qquad (15)$$

which is the energy norm of the distance between the FEM solution (denoted by h) and an improved solution (denoted by s) obtained by the Zienkiewicz-Zhu smoothing procedure [11]. Among the elements with increasing stress, only those elements, where the error exceeds the predefined threshold, are subdivided (refined). Similarly, among the elements with decreasing stress, only those elements, where the error is smaller than the above threshold, are coarsened. Note that we are not limited by the regularity of the mesh and can start from any (reasonable) heterogeneous mesh as a starting point prior to refinement.

A. Zienkiewicz-Zhu error estimate

Using the superconvergent patch recovery (SPR) procedure [11], the smoothed stress field $\boldsymbol{\sigma}^s$ is recovered from the stresses computed at the element center. The idea of this technique is based on the fact that the stress and strain at the superconvergent points (at element center in the case of linear hexahedral elements) are accurate with higher order than at the element nodes, and these values are employed to recover the nodal stress and strain within the least squares sense. A 2D representation of a patch of 8 hexahedral elements is shown in Fig. 3. For each component σ_j^h of the FEM solution $\boldsymbol{\sigma}^h$, the nodal recovered stresses are computed by defining a polynomial interpolation within the element patch as

$$\tilde{\sigma}_j^s = \mathbf{P}\mathbf{a}_j = [1\,x\,y\,z\,xy\,yz\,zx\,xyz][a_1\,a_2\,a_3\,a_4\,a_5\,a_6\,a_7\,a_8]_j^T. \quad (16)$$

Let $\mathbf{P}_k \equiv \mathbf{P}(x_k, y_k, z_k)$. To determine the unknowns $\mathbf{a}_j$ we minimize, for 8 sampling points k of an element patch,

$$\Pi = \sum_{k=1}^{8} \left[\sigma_j^h(x_k, y_k, z_k) - \mathbf{P}_k \mathbf{a}_{j_k} \right]. \quad (17)$$

This minimization results in finding $\mathbf{a}_j$ by

$$\mathbf{a}_j = \mathbf{A}^{-1}\mathbf{b} \quad (18)$$

where

$$\mathbf{A} = \sum_{k=1}^{8} \mathbf{P}_k^T \mathbf{P}_k \text{ and } \mathbf{b} = \sum_{k=1}^{8} \mathbf{P}_k^T \sigma_j^h(x_k, y_k, z_k) \quad (19)$$

Once $\mathbf{a}_j$ is available, the nodal recovered stress values are obtained by simply employing Eq. (16) with $\mathbf{P}$ evaluated at the corresponding node. [2]
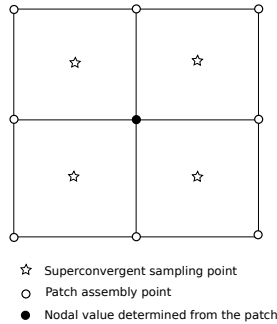

Fig. 3: Smoothed gradient obtained from an element patch.

B. Element marking strategy

After obtaining error distribution across all the elements, we employ the maximum strategy to select those elements which must be refined for the next level mesh. In this strategy, only those elements, where error (see Eq. (15)) is higher than certain threshold, are refined. Let $\eta_M = \max_e \eta_e$, where η_e is defined in Eq. (15). We mark an element for refinement if

$$\eta_e \geq \theta \eta_M \text{ with } 0 < \theta < 1. \quad (20)$$

---

[2]Similar to the displacements, the recovered stresses $\boldsymbol{\sigma}^s$ can also be obtained using element shape functions $\boldsymbol{\sigma}^s = \mathbf{N}\tilde{\boldsymbol{\sigma}}^s$.

Other marking strategies, such as bulk/equilibration strategy, or percentage strategy, see e.g. [10], can also be used. However, the maximum strategy described above is the cheapest among all, and hence, it is preferred for our use. In the maximum strategy, a large value of θ leads to small number of elements marked for refinement, and small value of θ leads to large number of elements marked for refinement. In our studies presented in Section V, we set $\theta = 0.3$.

C. Template-based adaptive h-refinement

Once the refinement criterion is satisfied within an element, the element is replaced by several elements according to a predefined template. The template is simply a set of nodes and an associated topology, defined using an isoparametric formulation. The template nodes are added by using their natural coordinates. The position $\mathbf{x}_j$, in Cartesian coordinates, of the new node j is defined as $\mathbf{x}_j = \mathbf{x}_i \xi_i^j$, where the Einstein summation convention is applied on the nodes i of the removed element ($i = 1, 2, \ldots 8$ for hexahedral elements). The shape function ξ_i^j is computed from the barycentric coordinates of template node j with respect to the node i. The procedure is summarized below:

1) Remove the element to be refined
2) Add template nodes and then template elements using the element shape functions
3) Update the topology of the global mesh
4) Compute stiffness matrix of new elements
5) If needed, update the mass and damping matrices

It is worth mentioning that if, after refinement, a new element fulfills the refinement criterion, it can be refined again, using the same predefined template. This results in a multi-resolution mesh (see Fig. 4). Conversely, if the coarsening criterion is satisfied in already refined elements, the coarsening procedure is applied by simply removing respective fine elements, and updating the associated matrices.
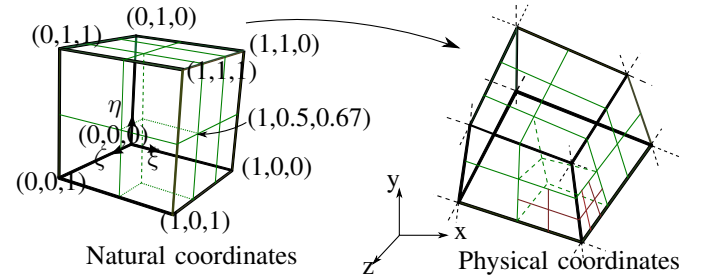

Fig. 4: Adaptive subdivision process: each element to be subdivided is topologically transformed in its reference shape, using a template expressed in natural coordinates. Cartesian coordinates in the mesh are computed using the element shape functions. The process can be applied recursively, and is completely reversible.

D. T-junction handling

Since elements are refined by using templates, regardless of their neighboring elements, some T-junctions (incompatible nodes or hanging nodes) are generated. To avoid discontinuities at the T-junctions during the simulation, these nodes

need to be handled in a special way, in which T-junction nodes are considered as slave of other independent (master) degrees of freedom (DOFs). One of the possible options is to use Lagrange multipliers, but this approach increases the total number of DOFs (as to solve the unknown Lagrange multipliers in addition), and usually leads to ill-conditioned systems. In our approach, we follow the method proposed by [30], which considers only the reduced system (without T-junctions) when solving for the new positions. Let $\mathbf{T}$ denote the transformation matrix from the reduced system to the full one (with T-junctions). The matrix $\mathbf{T}$ contains the barycentric coordinates of the T-junctions (slaves) with respect to their masters, and contains 1 for all other normal DOFs. The reduced system matrix A_r is then computed from the full system matrix A_f as

$$\mathbf{A}_r = \mathbf{T}^T \mathbf{A}_f \mathbf{T}. \tag{21}$$

The nodal forces in the reduced space is computed from the full space as $\mathbf{f}_r = \mathbf{T}^T \mathbf{f}_f$. The reduced system $\mathbf{A}_r d\mathbf{v}_r = \mathbf{f}_r$ is solved to find $d\mathbf{v}_r$, the difference of velocity between current and previous time step. Once $d\mathbf{v}_r$ is available, the difference of velocity in the full space are easily deduced as $d\mathbf{v}_f = \mathbf{T} d\mathbf{v}_r$. The latter is employed to update the new position and velocity of the object as in Eq. (13).

A heuristic example is shown in Fig. 5 to explicitly illustrate the method for T-junction handling, especially how the full and reduced systems are defined, resulting in the computation of the transformation matrix between them. After subdivision, node 7 is a T-junction in the full system (see Fig. 5a). Within this heuristic illustration, considering that each node has only one DOF and the static condition is applied, the displacement of the node 7 is expressed from those of node 2 and 3 as

$$u_7 = 0.5u_2 + 0.5u_3. \tag{22}$$

Fig. 5b shows the reduced system where the T-junction node 7 is not considered. The displacement fields between these full and reduced systems are expressed (by taking into account Eq. (22)) through the transformation matrix $\mathbf{T}$ as

$$\underbrace{\begin{Bmatrix} u_0 \\ u_1 \\ \vdots \\ u_7 \end{Bmatrix}}_{\mathbf{u}_f} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{T}} \underbrace{\begin{Bmatrix} u_0 \\ u_1 \\ \vdots \\ u_6 \end{Bmatrix}}_{\mathbf{u}_r}. \tag{23}$$

The transformation matrix $\mathbf{T}$ for the general 3D case where each node has three DOFs is built straightforwardly from this example.

Assuming we are using a non-linear constitutive law (e.g. hyperelastic) or a co-rotational FE formulation, the system matrix needs to be updated at each time step. Consequently, the local updating of the topology has a very limited impact on the computation. The main overhead comes from the T-junction
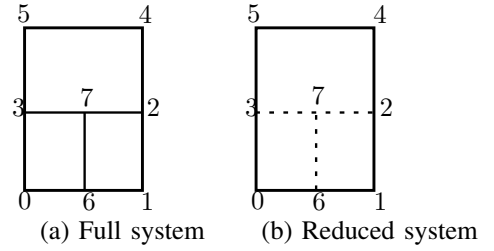

Fig. 5: Illustration of T-junction handling method on a schematic example.

handling, but it is somewhat compensated by the reduced dimensions of the linear system to be solved (although the reduced matrix is denser than the initial one). Our experience has shown that if we consider that about 10% of the mesh elements are subdivided, approximately 20% of the nodes in the resulting mesh are T-junctions. Obviously, the number of T-junctions depends strongly on the template mesh used by the refinement, and also on the fact that the elements are subdivided locally within one or several regions.

IV. NEEDLE-TISSUE INTERACTION ALGORITHM

To model the interaction of needle and tissue, we consider two different constraints: penetration (puncture) and sliding [17]. To avoid remeshing when modeling needle-tissue interaction, we use the same constraints based approached as described in [17]. However, unlike [17], we solve the constrained system differently. Before entering the tissue, the needle-tissue constraint is only created when the needle tip is in contact with the tissue surface. This penetration constraint is represented mathematically as $\mathcal{P}(\mathbf{x}_n, \mathbf{x}_t) \geq 0$, where $\mathbf{x}_n$, and $\mathbf{x}_t$ stand for the position of the needle, and tissue, respectively. Immediately after entering the tissue, sliding constraint between the needle and tissue is created along the needle trajectory as $\mathcal{S}(\mathbf{x}_n, \mathbf{x}_t) = 0$. When friction is considered, both $\mathcal{P}$ and $\mathcal{S}$ are nonlinear. The constraints $\mathcal{P}$ and $\mathcal{S}$ between the tissue (denoted by subscript 1) and needle (denoted by subscript 2) are expressed through the global coordinate system using Lagrange multipliers $\boldsymbol{\lambda}$ as follows

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{J}_1^T \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{J}_2^T \\ \mathbf{J}_1 & \mathbf{J}_2 & \mathbf{0} \end{pmatrix} \begin{Bmatrix} d\mathbf{v}_1 \\ d\mathbf{v}_2 \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{0} \end{Bmatrix}, \tag{24}$$

where $\mathbf{A}_1$ and $\mathbf{A}_2$ are the system matrices for the soft tissue and needle, respectively; $\mathbf{J}_1$ and $\mathbf{J}_2$ account for contraint directions between the needle and tissue. In the local coordinate system attached to the needle, constraints between needle and tissue are only expressed in two directions orthogonal to the needle shaft, resulting in a needle-tissue sliding constraint. The expressions of these constraints in the global coordinate system $\mathbf{J}_1$ and $\mathbf{J}_2$ are then built by transforming the local constraint expressions from the local coordinate system to the global one. However, formulating the problem as (24) leads to a non positive definite global matrix, which makes the system challenging to solve. An alternative approach, as proposed in [15], [17], is to solve the interaction problem

in three steps: predictive motion (no interaction constraints), constraint solving, and corrective motion. However, this alternative requires the computation of matrix inverse $\mathbf{A}_1^{-1}$ and $\mathbf{A}_2^{-1}$. This approach is time consuming, especially for large systems. Unlike this method, we solve the constrained problem iteratively by using the augmented Lagrangian method [31]

$$(\mathbf{A} + \mathbf{J}^T\mathbf{W}\mathbf{J})d\mathbf{v}^{k+1} = \mathbf{b} - \mathbf{J}^T\boldsymbol{\lambda}^k \tag{25a}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \mathbf{W}\mathbf{J}d\mathbf{v}^{k+1}, \tag{25b}$$

where $\mathbf{W}$ is the penalty weight matrix with finite values. The advantage of this method is that the exact solution of the needle-tissue interaction can be obtained, as compared to the penalty method (see e.g. [32]), and no additional DOFs are needed, as compared to the classical Lagrange multiplier method [33]. A critical feature of this approach is that the system matrix in (25a) is positive definite, therefore iterative solvers, such as the conjugate gradient, can be used efficiently.

It is worth stressing that using the augmented Lagrangian method for solving needle-tissue interaction, combined with the T-junction handling in the tissue, is straightforward. Indeed, as mentioned above, it is sufficient to solve (25a) in the reduced space, and $d\mathbf{v}^{k+1}$ is easily computed from the reduced solution.

V. RESULTS AND DISCUSSIONS

To demonstrate the efficiency of our method, we present several numerical studies. We first present the convergence of the stress error on a typical L-shaped domain. The motivation for this test is because of its localized nature, i.e. stress is concentrated at the corner of the domain, which mimics the localized scenario of needle insertion. To demonstrate the computational advantage of adaptive refinement over uniform refinement, we also present the computational time for this problem. Then, to point out the benefits of a local mesh refinement in needle insertion simulation, we study a needle insertion scenario with friction to show the impact of local refinement on the displacement field around the needle shaft and also on the needle-tissue interaction force profile. Finally, a more complicated scenario is simulated i.e. insertion of a needle into a liver which is undergoing breathing motion. In our simulations, the needle and soft tissue follow a linear elastic constitutive law, associated to a co-rotational FE formulation.

A. Convergence study

To show the advantage of error-controlled adaptive refinement scheme, as compared to the uniform mesh refinement, a convergence study is performed on a 3D L-shaped domain. As shown in Fig. 6, the L-shaped domain is clamped at the right boundary, and simply supported in the vertical direction at the top boundary. The dimension is set to $L = 4$ and the thickness of the domain is $L/2 = 2$. Young's modulus, and Poisson's ratio of the tested material are 1×10^3, and 0.3, respectively. The domain is subjected to a uniformly distributed traction force on the left surface boundary.

Starting with the mesh having $8 \times 8 \times 4$ hexahedral elements (excluding the $4 \times 4 \times 4$ corner elements), two types of refinements are performed. The first one, called uniform refinement, consists of subsequently subdividing every hexahedral element in to 8 smaller elements. In the second approach, called adaptive refinement, only those elements which satisfy the marking condition (20) are refined (subdivided in to 8 smaller elements).
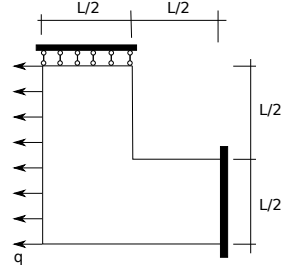

Fig. 6: Boundary conditions of the L-shaped domain test.

We define the relative error η as

$$\eta = \frac{\sqrt{\sum_{e=1}^{N_e} \int_{\Omega^e} (\boldsymbol{\epsilon}^h - \boldsymbol{\epsilon}^s)^T (\boldsymbol{\sigma}^h - \boldsymbol{\sigma}^s)\mathrm{d}\Omega}}{\sqrt{\sum_{e=1}^{N_e} \int_{\Omega^e} (\boldsymbol{\epsilon}^h)^T \boldsymbol{\sigma}^h \mathrm{d}\Omega}}. \tag{26}$$

In Fig. 7, we show the plots of the relative error versus the number of DOFs for uniform and adaptive refinement. We see that for the uniform refinement, the relative error η converges with a slope of 0.21, which corresponds to the theoretical slope of $2/9$ for singular problems in 3D. By comparison with the uniform refinement, the adaptive refinement converges with a higher slope (0.31). Clearly, to achieve certain expected error of the simulation, the adaptive refinement needs fewer DOFs than the uniform refinement.
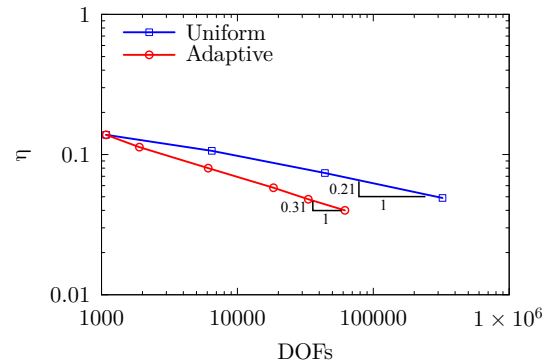

Fig. 7: Convergence of the relative error, comparison between uniform and adaptive refinements.

To demonstrate the performance of adaptive (local) refinement in terms of computational time, as compared to uniform (full) refinement of the mesh, with the same expected relative error $\eta = 8\%$, we again studied the L-shaped domain problem. The result is reported in Table I. The local refinement decreases the number of DOFs by a factor 5.9 (7473 vs. 44064) associated with a computational speed-up of $16\times$ (736.41 ms vs. 12140.8 ms).

In view of above observations, it is a strong argument to support the employment of adaptive refinement scheme while limiting the discretization error in real-time simulations.

	DOFs	Time		Total time
		TC	MS	
Full refinement	44064	x	12140.8	12140.8
Local refinement	7473	319.24	417.17	736.41

TABLE I: Computational time (in $10^{-3}s$). TC: Topological Changes, MS: System Matrix Solve.

B. Impact of local mesh refinement on displacement field

We now present the results of a simulation of needle insertion into a homogeneous 3D tissue model. For this study, we consider the Young's modulus of 10^8 for the needle, and 10^3 for the tissue, whereas the Poisson's ratio is taken as 0.4 for both. The friction coefficient between the needle and the tissue is set to 0.9. The displacement field due to frictional interactions with the needle, viewed from the xy plane of the tissue, is shown in Fig. 8.

It is shown (by the nonlinear variation of the displacement field in the vicinity of the needle) that when the mesh is adaptively refined near the needle shaft, the needle-tissue interaction is captured as good as in the case of full refinement (see Fig. 8d). Indeed, closer the position is to the needle shaft, higher the obtained displacement field. Conversely, when a coarse element is used, and is not refined during the simulation, the above behavior is not reproduced within the element (Fig. 8a). It is important to point out that the refinement using anisotropic template, as in Fig. 8b, is very relevant since it generates fewer DOFs than using the isotropic template (Fig. 8c), while still catching the nonlinear displacement field. Note that the refinement is not restricted to one element, but only one level of refinement is used. It also happens that the criterion used in this simulation leads to refining only one element.

C. Impact of local mesh refinement on needle-tissue interaction

In order to gain insight into the nonlinear behavior of the needle-tissue interaction around the needle shaft, and to exhibit the effect of the adaptive refinement on a trade-off between computational time and precision, a needle insertion simulation into a phantom tissue test is carried out, see Fig. 9. For this study, we consider the Young's modulus of 50 MPa for the needle, and 10 MPa for the tissue, whereas the Poisson's ratio is taken as 0.4 for the tissue, and 0.3 for the needle, respectively. Note that, in a separate study for the tissue (not reported here for brevity reasons), we also run the simulation for Young's modulus 2 MPa and Poisson's ratio 0.49, and the results are qualitatively same as those reported in the above-mentioned case. Again, a linear elastic model based on corotational formulation is employed for the needle as well as the tissue. The dimension of the tissue is $4 \times 2 \times 2$ cm. The needle length and radius are of 3.2 cm and 0.1 cm, respectively.

Three meshing schemes are employed: a coarse mesh with resolution $10 \times 5 \times 5$ nodes, a fine mesh with resolution $20 \times 10 \times 10$ nodes and an adaptive mesh (starting with the coarse mesh $10 \times 5 \times 5$ nodes and adaptively refining the mesh during the simulation). Within the adaptive meshing scheme, the mesh

refinement is again piloted by the error estimate described in Section III.

To investigate the sensitivity of the needle-tissue interaction parameters (frictional coefficient and puncture strength) on the resulting mesh adaptation, and thus on computational output, two scenarios are studied. The first concerns varying the puncture strength σ_{n0} parameter (0 N, 10 N, and 20 N), while keeping the same frictional coefficient $\mu = 0.5$ between the tissue and the needle shaft. The second is dedicated to study the influence of the frictional coefficient by setting it to 0.1, 0.3 and 0.5, while keeping the puncture strength unchanged ($\sigma_{n0} = 10$ N). Within these two scenarios, the frictional coefficient on the tissue surface is set to 0.8.

Fig. 10 shows the plots of the integrated interaction force along the needle shaft versus the displacement of the needle tip for the first scenario. The second scenario is depicted in Fig. 11.

It shows that when the contact force between the needle tip and the tissue surface is higher than the tissue puncture strength, the needle penetrates into the tissue. Right after this penetration event, a relaxation phase can be observed that induces a decreasing force at the needle bases. Thereafter, it is observed that as the needle moves forward, the interaction force increases due to the increasing frictional force along the needle shaft (which is directly proportional to the insertion distance). Only when the contact force at the needle tip is greater than the cutting strength of the soft tissue, the needle cuts the tissue and continues going ahead. Immediately after this cutting action, the relaxation phase is observed anew. This behavior is periodically observed during the needle insertion. These observations are clearly shown in Fig. 12a, which is obtained by zooming in Fig. 10b. A typical behavior with distinguished phases is presented in Fig. 12b.

It is observed that under mesh refinement the resulting global behavior of needle-tissue system is less stiff. This is explained by the fact that beneath mesh refinement, a greater displacement field is obtained, which results from the needle-tissue interaction (as also observed in Section V-B). It is also shown that using the adaptive refinement scheme, the interaction needle-tissue behavior is close to those when using a fine mesh (see Fig. 10). As can be seen in Fig. 13, an interesting observation is that the number of DOFs in the adaptive refinement simulation is significantly fewer than that of using the fine mesh. This obviously results in an important gain in terms of computational time. Indeed, the simulation using the adaptive refinement mesh runs at nearly 45 FPS compared to 4 FPS of that using the uniform fine mesh.

It is seen from Fig. 11 that the smaller the frictional coefficient is, the more the behavior of the adaptive refinement scheme differs from that of the simulation using the fine mesh. It aligns nicely with the fact that smaller friction force does not lead to mesh refinement around the needle shaft. Indeed, as seen in Fig. 14, the refinement in the case of frictional coefficient $\mu = 0.1$ is mostly due to the penetration force at the tissue surface. Note that, for a fair comparison, the θ coefficient in Eq. (20) was chosen as $\theta = 0.3$ for all friction coefficients. However, it is also possible to use lower values of θ for lower values of the friction coefficient (such as $\mu = 0.1$),
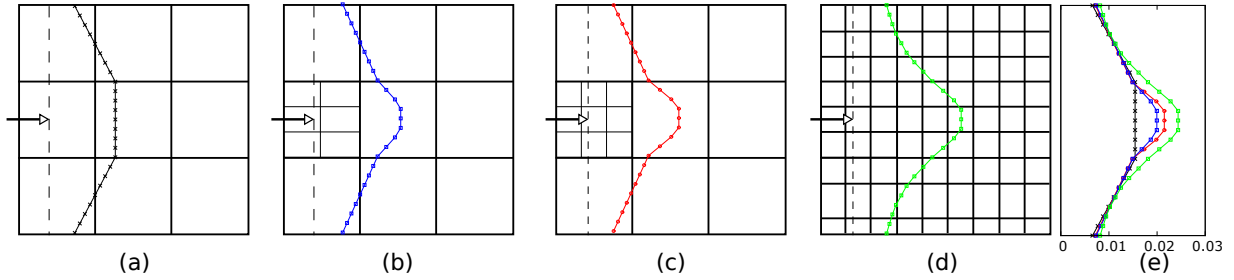
Fig. 8: Variation of tissue displacement resulting from friction during needle insertion, measured along a vertical line located at the needle tip. (a) The un-refined mesh; (b) adaptive refinement using an anisotropic template $2 \times 3 \times 3$; (c) adaptive refinement using an isotropic template $3 \times 3 \times 3$; and (d) full refinement. The graph (e) shows the benefits of the anisotropic refinement.
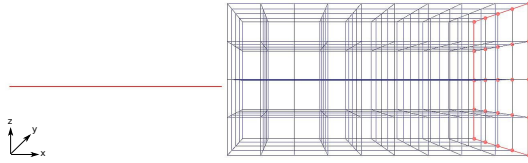

Fig. 9: Schematic representation of needle insertion simulation into a phantom tissue. The phantom tissue is clamped at the right surface.

which leads to a higher degree of local refinement. We also remark that when uniformly very fine meshes ($30 \times 15 \times 15$ and $40 \times 20 \times 20$ nodes) are used (not shown here for brevity reasons), the force-displacement responses converge. However, simulations on such fine meshes are computationally very expensive, and because of the required constraint of real-time simulation, such meshes can not be employed. The proposed adaptive refinement allows us to perform real-time simulations while keeping the error in control.

D. Application to liver

The method proposed in this paper is now applied to a liver model undergoing breathing motion, to mimic a typical case of radio-frequency ablation of a tumor. The same Young's modulus and Poisson's ratio for the needle and the tissue as in Section V-C are employed. The frictional coefficient is set to 0.5 when the needle is inserted and to 0.1 when it is pulled back (the performance of the adaptive refinement scheme does not depend on the ratio between the frictional coefficients during insertion and retraction). The puncture force at the tissue surface is set to 10 N. Induced by error estimate (Eq. (20)), the needle insertion and constraints applied to the liver lead to refinements in different regions. The initial mesh has 1179 DOFs. When the needle advances into the liver, combining with the motion of the liver due to breathing effect, the mesh is progressively refined to accurately take in to account the interaction of needle and liver. The maximum number of DOFs when the needle is completely inserted into the liver is 2961. And, when the needle is steadily pulled back, the mesh is then progressively coarsened until the needle is completely outside of the liver. Thereafter, the refinement process is now only due to the movement of the liver by breathing effect and imposed boundary conditions. The number of DOFs at this

stage is 1509. By applying this adaptive refinement/coarsening procedure, it is not only guaranteed that the discretization error is fully controlled, but the computational cost is also kept as small as possible. Indeed, without adaptive remeshing procedure applied on the initial mesh, the simulation runs at 35 FPS while the discretization error is 12%, whereas when the adaptive refinement is performed, it runs at 22 FPS while decreasing the discretization error to 6%. Note that these frame rates result not only from computational resolutions of needle, tissue and their interactions but also from their visualization cost.

In order to investigate the benefits of the adaptive refinement scheme when the needle is inserted and retracted into the liver phantom, tests with uniform and adaptive refinement schemes are carried out. Within the uniform refinement case, a coarse mesh with 723 DOFs and a fine mesh with 3894 DOFs are used for the liver discretization. Whereas upon the adaptive refinement scenario, the simulations start with the coarse mesh 723 DOFs and is adaptively refined by two schemata: each marked element is refined into (i) $2 \times 2 \times 2$ elements, and (ii) into $3 \times 3 \times 3$ elements. The integrated needle-tissue interaction force along the needle shaft is plotted versus the needle tip displacement when the needle is inserted and pulled back, see Fig. 16.

It is observed that when the needle is outside the tissue, there is no interaction force between them. This is also detected when the needle is completely retracted from the tissue. It is clearly shown that the needle-tissue interaction depends strongly on the mesh used (especially if the mesh is coarse). The mesh influence reveals a stronger effect on the insertion stage than the pullback one. This is fully understood by the fact that the frictional coefficient between the needle shaft and the tissue is more important during the insertion steps than that during the pullback ones (0.5 versus 0.1, respectively). Using the coarse mesh, the puncture force at the tissue surface is not well captured compared to the case where the fine mesh or adaptive refined mesh is employed, see Fig. 16. Under mesh refinement around the needle shaft guided by error estimate, the needle-tissue interaction converges to the solution of the fine mesh. However, the maximum number of DOFs when using the adaptive refinement schemes $2 \times 2 \times 2$ and $3 \times 3 \times 3$ is 1071 and 2193 respectively. Therefore, as observed in Section V-C, using the adaptive refinement scheme results in significantly fewer DOFs compared to the employment

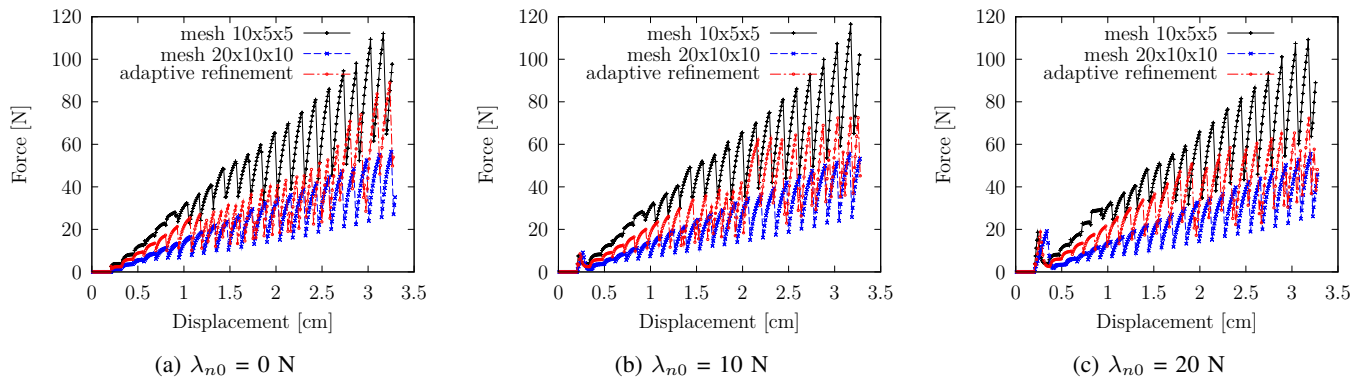(a) $\lambda_{n0} = 0$ N (b) $\lambda_{n0} = 10$ N (c) $\lambda_{n0} = 20$ N

Fig. 10: Comparison of needle-tissue interaction forces along the needle shaft within two cases: without refinement (with different mesh resolutions) and with adaptive refinement. The penetration strength is varied while keeping the same frictional coefficient $\mu = 0.5$ between the needle shaft and the soft tissue.
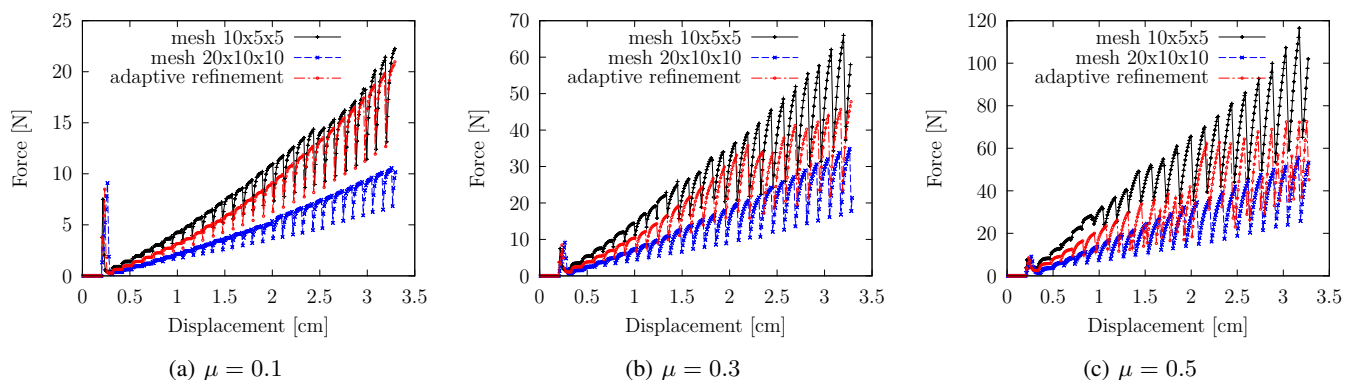

(a) $\mu = 0.1$ (b) $\mu = 0.3$ (c) $\mu = 0.5$

Fig. 11: Comparison of needle-tissue interaction forces along the needle shaft within two cases: without refinement (with different mesh resolutions) and with adaptive refinement. The frictional coefficient between the needle shaft is varied while keeping the same penetration strength (10 N) at the tissue surface.

(a) (b)
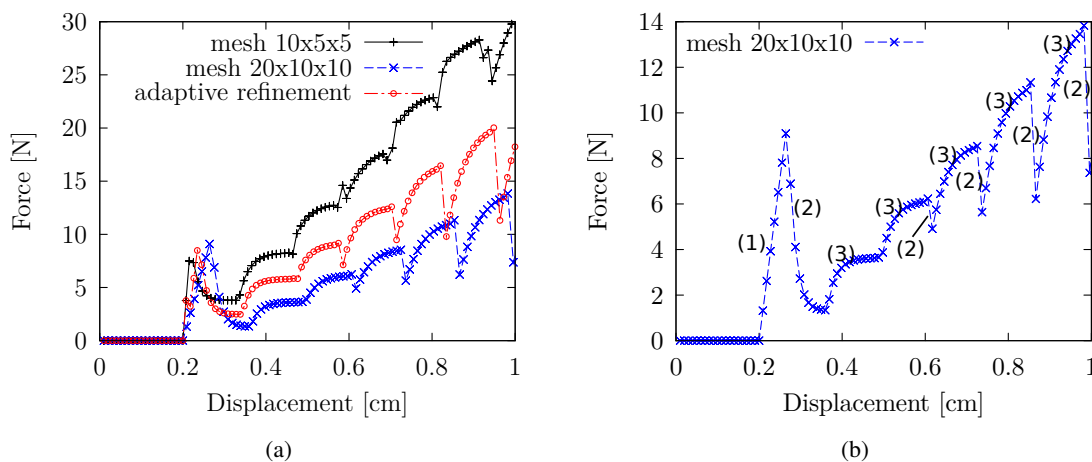
Fig. 12: (a) Puncture, cutting and relaxation behaviors are shown by a zoom-in plot from Fig. 10b. (b) The typical behavior is shown in phases. Phase (1): The needle is puncturing the tissue surface. Phase (2): Just after the penetration event, the relaxation occurs. Phase (3): The interaction force increases due to the fact that frictional force increases with insertion distance. When the needle tip has cut the tissue to advance forward, the relaxation occurs again.

of uniform fine mesh. All of these lead to the conclusion that even starting with the coarse mesh but employing the adaptive refinement scheme, the needle-tissue interaction can be simulated more precisely compared to the coarse mesh, and with significantly lower computational cost compared to the uniform fine mesh.
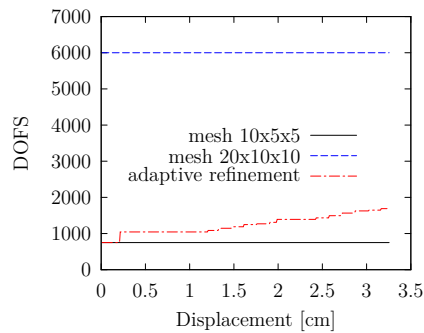
Fig. 13: Number of DOFs during the needle insertion of the simulation in Fig. 10b.

VI. Conclusion and perspectives

This paper contributes a structured approach to answering the important but rarely tackled question of accuracy in surgical simulation.

The novelty of our paper is to drive local adaptive mesh refinement during needle insertion by a robust a posteriori estimate of the discretization error.

This can be seen as a first step to control the error associated with acceleration methods in needle surgical simulations, and to separate the modeling (are we solving the right problem?) and discretization error (are we solving the problem right?).

Verification of the discrete scheme is guaranteed in our approach because the a posteriori estimate asymptotically converges to the exact error. As we use an implicit approach, we also control the error on the equilibrium equations. As such, assuming a proper material model and kinematics for the problem, we can guarantee the accuracy of the solution. This is not the case in explicit time stepping approaches.

Validation of the approach is not considered. We focus here only on one source of error (discretization). Whilst this is a limitation, we do believe that quantifying discretization errors separately to modeling errors is necessary to devise accurate and clinically-usable surgical simulators and to better understand the resulting simulation results.

A natural direction for research, building on recent work on data-driven simulations is to devise error-controlled approaches able to learn from data as it is acquired during the simulation [34]–[36]. In such a paradigm, the model would adapt to the real situation, as opposed to being driven by a continuous indirect comparison, as is the case in this work, with an unknown exact solution. In turn, such a data-driven approach would facilitate patient-specific simulations, which were not considered here. We are currently investigating such directions through Bayesian inference for parameter identification and model selection [37] and uncertainty quantification approaches [38].

Acknowledgements

References

[1] A. Nealen *et al.*, "Physically based deformable models in computer graphics," in *Computer graphics forum*, vol. 25, no. 4, 2006, pp. 809–836.

[2] Y. Wang *et al.*, "Linear subspace design for real-time shape deformation," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 57, 2015.

[3] H. Courtecuisse *et al.*, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical Image Analysis*, vol. 18, no. 2, pp. 394 – 410, 2014.

[4] O. Zienkiewicz *et al.*, *The finite element method: Its basis and fundamentals*. Elsevier, 2013, vol. 1.

[5] V. P. Nguyen *et al.*, "Meshless methods: a review and computer implementation aspects," *Mathematics and computers in simulation*, vol. 79, no. 3, pp. 763–813, 2008.

[6] X. Wu *et al.*, "Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes," in *Computer Graphics Forum*, vol. 20, no. 3, 2001, pp. 349–358.

[7] M. Müller *et al.*, "Stable real-time deformations," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 49–54.

[8] M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*. John Wiley & Sons, 2011, vol. 37.

[9] M. Seiler *et al.*, "Robust interactive cutting based on an adaptive octree simulation mesh," *The Visual Computer*, vol. 27, no. 6-8, pp. 519–529, 2011.

[10] R. Verfürth, *A posteriori error estimation techniques for finite element methods*, ser. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2013.

[11] O. Zienkiewicz and J. Zhu, "The superconvergent patch recovery (spr) and adaptive finite element refinement," *Computer Methods in Applied Mechanics and Engineering*, vol. 101, no. 1, pp. 207–224, 1992.

[12] C. Carstensen and S. Bartels, "Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM," *Math. Comp.*, vol. 71, no. 239, pp. 945–969 (electronic), 2002.

[13] S. Bartels and C. Carstensen, "Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. II. Higher order FEM," *Math. Comp.*, vol. 71, no. 239, pp. 971–994 (electronic), 2002.

[14] I. Babuška and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM J. Numer. Anal.*, vol. 15, no. 4, pp. 736–754, 1978.

[15] N. Hamzé *et al.*, "Preoperative trajectory planning for percutaneous procedures in deformable environments," *Computerized Medical Imaging and Graphics*, vol. 47, pp. 16–28, 2016.

[16] N. Abolhassani *et al.*, "Needle insertion into soft tissue: A survey," *Medical Engineering and Physics*, vol. 29, pp. 413–431, 2007.

[17] C. Duriez *et al.*, "Interactive simulation of flexible needle insertions based on constraint models," in *Lecture Notes in Computer Science*, vol. 5762, no. PART 2, 2009, pp. 291–299.

[18] S. Misra *et al.*, "Mechanics of flexible needles robotically steered through soft tissue," *Int. J. Rob. Res.*, vol. 29, no. 13, pp. 1640–1660, Nov. 2010.

[19] J. Wu *et al.*, "Physically-based simulation of cuts in deformable bodies: A survey." in *Eurographics (State of the Art Reports)*, 2014, pp. 1–19.

[20] C. Dick *et al.*, "A hexahedral multigrid approach for simulating cuts in deformable objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1663–1675, 2011.

[21] G. R. Liu and S. S. Quek, "Chapter 3 - Fundamentals for Finite Element Method," in *The Finite Element Method (Second Edition)*, second edition ed., G. R. Liu *et al.*, Eds. Oxford: Butterworth-Heinemann, 2014, pp. 43–79.
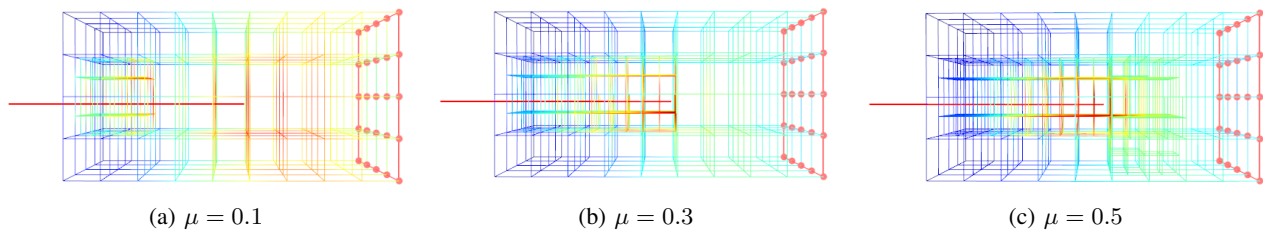
(a) $\mu = 0.1$ (b) $\mu = 0.3$ (c) $\mu = 0.5$

Fig. 14: Refinement patterns (colored by stress level) of the adaptive scenario while using different frictional coefficients.
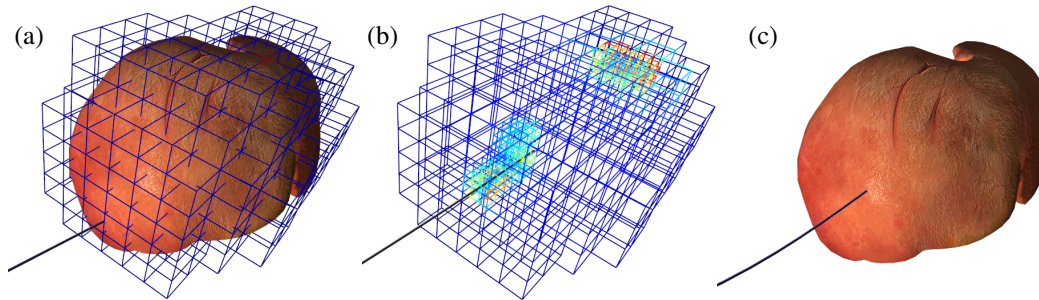

(a)   (b)   (c)

Fig. 15: (a) Simulation of needle insertion in a liver; (b) using dynamic mesh refinement scheme driven by error estimate; (c) visual depiction. The simulation runs at 22 Hz (on a 4GHz processor).
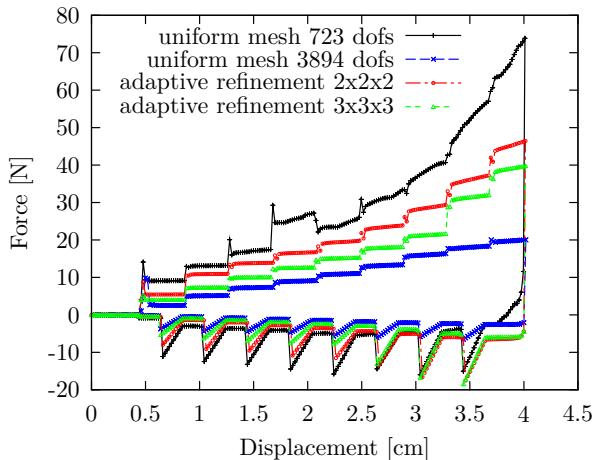

Fig. 16: Needle-liver phantom interaction force during needle insertion and pullback. The interaction force varies due to advancing friction and tissue cutting strength, globally increases (with positive values) during the insertion stage. At 4 cm of the needle-tip displacement, the needle is retracted and the interaction force changes the direction, varies due to retrograding friction, globally decreases and gets zero when the needle is completely pulled out. These form a loop profile for force-displacement curve resulting from needle insertion-retraction.

[22] O. Zienkiewicz and R. Taylor, *The Finite Element Method: Solid mechanics*, ser. Referex collection.Mecánica y materiales. Butterworth-Heinemann, 2000.

[23] C. Felippa and B. Haugen, "A unified formulation of small-strain corotational finite elements: I. theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 2124, pp. 2285 – 2335, 2005.

[24] A. Pinelli *et al.*, "Immersed boundary method for generalised finite volume and finite difference navier-stokes solvers," *Journal of Computational Physics*, vol. 229, no. 24, pp. 9073–9091, 2010.

[25] D. Baraff and A. Witkin, "Large steps in cloth simulation," in *Proceedings of SIGGRAPH*, 1998, pp. 43–54.

[26] D.-Y. Kwak and Y.-T. Im, "Hexahedral mesh generation for remeshing in three-dimensional metal forming analyses," *Journal of Materials Processing Technology*, vol. 138, no. 13, pp. 531 – 537, 2003.

[27] D. Koschier *et al.*, "Adaptive Tetrahedral Meshes for Brittle Fracture Simulation," in *Symposium on Computer Animation*, V. Koltun and E. Sifakis, Eds. The Eurographics Association, 2014.

[28] D. Burkhart *et al.*, "Adaptive and Feature-Preserving Subdivision for High-Quality Tetrahedral Meshes," *Computer Graphics Forum*, vol. 29, no. 1, pp. 117–127, 2010.

[29] C. Paulus *et al.*, "Virtual cutting of deformable objects based on efficient topological operations," *The Visual Computer*, vol. 31, no. 6-8, pp. 831–841, 2015.

[30] E. Sifakis *et al.*, "Hybrid simulation of deformable solids," in *Proc. Symposium on Computer Animation*, 2007, pp. 81–90.

[31] H. Uzawa and K. J. Arrow, "Iterative methods for concave programming," in *Preference, production, and capital*. Cambridge University Press, 1989, pp. 135–148.

[32] I. Babuška, "The finite element method with penalty," *Mathematics of Computation*, vol. 27, no. 122, pp. 221–228, 1973.

[33] P. Papadopoulos and J. M. Solberg, "Recent Advances in Contact Mechanics A Lagrange multiplier method for the finite element solution of frictionless contact problems," *Mathematical and Computer Modelling*, vol. 28, no. 4, pp. 373–384, 1998.

[34] M. Seiler *et al.*, "Enriching coarse interactive elastic objects with high-resolution data-driven deformations," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2012, pp. 9–17.

[35] ——, "Data-driven simulation of detailed surface deformations for surgery training simulators," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 10, pp. 1379–1391, Oct 2014.

[36] M. U. Seiler *et al.*, "Efficient Transfer of Contact-Point Local Deformations for Data-Driven Simulations," in *Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 2014.

[37] H. Rappel *et al.*, "Bayesian inference for the stochastic identification of elastoplastic material parameters: Introduction, misconceptions and additional insight," *CoRR*, vol. abs/1606.02422, 2016. [Online]. Available: http://arxiv.org/abs/1606.02422

[38] P. Hauseux *et al.*, "Accelerating monte carlo estimation with derivatives of high-level finite element models," *Computer Methods in Applied Mechanics and Engineering*, pp. –, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782516313470