CUDA accelerated simulation of needle insertions in deformable tissue

Alexandru Patriciu
1280 Main Street West, ITB A308, Hamilton, ON, Canada, L8S 4K1
E-mail: patriciu@mail.ece.mcmaster.ca

Abstract. This paper presents a stiff needle-deformable tissue interaction model. The model uses a mesh-less discretization of continuum; avoiding thus the expensive remeshing required by the finite element models. The proposed model can accommodate both linear and nonlinear material characteristics. The needle-deformable tissue interaction is modeled through fundamental boundaries. The forces applied by the needle on the tissue are divided in tangent forces and constraint forces. The constraint forces are adaptively computed such that the material is properly constrained by the needle. The implementation is accelerated using NVidia CUDA. We present detailed analysis of the execution timing in both serial and parallel case. The proposed needle insertion model was integrated in a custom software that loads DICOM images, generate the deformable model, and can simulate different insertion strategies.

1. Introduction

Needle insertions are used in medical procedures such as biopsies, Radio Frequency Ablations (RFA), or Cryo-ablations. The needle placement accuracy is important for the outcome of the procedure. Needle insertions are usually performed under image guidance which may or may not be real time. During the procedure, the tissue will deform as the needle is inserted; this may cause a shift of the target from the needle trajectory. In order to assess beforehand the accuracy of the needle insertion, it is desirable to simulate the needle insertion.

One of the first studies in needle-deformable tissue interaction was performed by DiMaio [5]. Their approach uses a linear elastic Finite Element Model (FEM) to simulate the tissue and the needle. The needle-deformable tissue interaction is modeled using a stick-slip approach. To validate the approach, an experimental system for measuring planar tissue deformation during needle insertions was developed. Goksel [7] extended that model to 3D. The model was also adapted to accommodate arbitrary meshes so that the anatomy can effectively be meshed using third-party algorithms. Nienhuys’s [13] approach was also based on FEM and uses quasi-static stick-slip friction for needle-deformable tissue interactions. It uses an iterative relaxation algorithm, and adaptive mesh resolution.

Alterovitz et al. investigated the insertion of flexible needles with beveled tip in deformable tissue [3]. They also proposed a path planning algorithm for beveled needle insertion for prostate. These studies are based on a dynamic FEM formulation and assumed linear models for tissue characteristics. In their recent work [2], they consider uncertainty and the planner computes optimal actions to maximize the probability that the needle will reach the desired target.

Misra et al. [12] proposed a new model of flexible needle-deformable tissue interaction. Their work extends the non-holonomic kinematic model proposed by Webster et. al. [14] by taking
into consideration the microscopic interaction between the bevel tip of the needle. Their results show that the models match the macroscopic needle insertions observations but discrepancies were especially large for large bevel angles. Duriez et al. [6] used complementarity constraints for the simulation of the insertion of flexible needles into soft tissue. Dehghan et al. [4] presents a needle path planning method for the insertion of rigid needles into deformable tissue. The optimization method is based on iterative simulations performed using a tissue finite element model. The experiment showed substantial decrease in the targeting error.

Within the computational mechanics community a strong research interest is devoted to the mesh-less methods that emerged in the last two decades. A comprehensive review of the mesh-less approach to continuum mechanics models is presented by Li and Liu [10]. Horton et al. [8] developed a meshless method for simulating soft organ deformation and simulated indentation of a swine brain and compared the results to experimental data. Li et al. [9] presented an adaptive meshless method (MLM) for solving deformable contact problems and validated it against analytical and numerical solutions.

In this paper we consider the interaction between stiff needles and deformable tissues. This is relevant for applications like breast biopsy. The deformable tissue is modeled using a mesh-less method (RKPM) and the needle-deformable tissue interaction is modeled using fundamental boundaries. The forces are generated either by the friction and cutting or by the needle constraints. In contrast with the approach followed by Duriez [6] the constraints are not solved explicitly but they are instead dynamically adjusted during the insertion.

The paper is organized as follows Section 2 presents the needle-deformable tissue interaction model; deformable tissue model and details regarding the parallel implementation on the Graphical Processing Units (GPU); Section 3 provides simulation results. The paper concludes with some observations and future work.

2. Stiff needle-deformable tissue interaction model

Several needle-deformable tissue interaction models have been proposed by researchers, a comprehensive survey of these models can be found in the paper by Abolhassani et al. [1] and Misra et al. [11].

In the proposed approach we model the needle-deformable tissue interaction as a fundamental boundary. The needle insertion action is performed by applying an insertion force on the needle. The interaction between the needle and the tissue comprises three types of forces - friction forces, cutting force, and constraint forces. The friction force is aligned with the needle direction and is distributed along the needle. The cutting force is aligned with the needle trajectory and it is applied by the tip of the needle. The sum of friction and the cutting forces gives the insertion force. The constraint force is orthogonal to the needle trajectory and ensures that the tissue in the neighborhood of the needle doesn’t move across the needle. From a physical perspective in order to simulate the tissue deformation during the needle insertion we need to model the deformable object subject to the interaction forces.

Figure 1 shows an object in the deformed configuration with the needle inserted in it. The needle trajectory comprises a set of points \( p_n; i = 1 \ldots NP \) represented in undeformed body coordinates. Each needle point \( p_n \) has an associated coordinate system \( \xi^i \) defined such that \( \xi^0 \) gives the direction of the needle and \( \xi^1 \) and \( \xi^2 \) are orthogonal to the needle direction. In this approach the force applied by the needle to the deformable object through friction and cutting is always aligned with axes \( \xi^0 \). Similarly, the constraint forces will be aligned with axes \( \xi^1 \) and \( \xi^2 \).

For the case of stiff needles all needle model points should lie on the needle axis. In Figure 1 \( \delta_1 \) and \( \delta_2 \) are the deviations of point \( p_n \) from the needle axis along directions \( \xi^1 \) and respectively \( \xi^2 \). Similarly, \( f_0 \) is the axial force applied by the needle, \( f_1 \) and \( f_2 \) are the constraint forces. The sum of all axial forces is equal with the insertion force. In order to have a proper configuration,
Figure 1: 2D deformable object and needle points. \((p_n_i)_{i=1...NP}\) are the needle points represented in undeformed body coordinates; \((\xi^i)_{i=1...NP}\) represent the needle coordinate systems; \((\xi^i_0)_{i=1...NP}\) are aligned with the needle axis, \((\xi^i_1)_{i=1...NP}\) are orthogonal to the needle axis; \((\delta^i)_{i=1...NP}\) represent the distances between \(p_n_i + u(p_n_i)\) and the needle axis; \((f^i)_{i=1...NP}\) are the constraint forces.

The constraint forces \(f^1_i\) and \(f^2_i\) have to be adjusted such that \(\delta^1_i \rightarrow 0\) and \(\delta^1_i \rightarrow 0\). The simulation starts with zero insertion force, the needle has only one segment inside the tissue. The insertion force is incremented and distributed as friction and cutting force - \(f^0_i\). The deformation is computed and if the needle points deviate from the desired trajectory the constraint forces are updated. This approach has the advantage that requires only a general purpose solver for a deformable object - for example a preconditioned conjugate gradient solver. The object structure will not change during the simulation, only the interaction forces will change. This is a desirable property when a parallel implementation is sought.

2.1. Adaptive Constraints Adjustment
Let’s assume that the lumped stiffness of the point \(p_n_i\) along direction \(\xi^i_j\) is \(k^i_j\); if we increment the force along the direction \(\xi^i_j\) with \(\delta f^i_j\) this should result in an incremental displacement \(\delta^i_j / k^i_j\). The friction coefficient for the segment \((p_n_i, p_{n_i-1})\) is \(\mu_i\). Furthermore, let’s assume that the function \(\text{Solve}(p_m, f, u)\) implements a solver of the deformable object that applies forces \(F^i\) along the boundary described by the points \(p_m\) and returns the displacements \(u_i\) of points \(p_n_i\). Then, an iterative algorithm that simulates the needle insertion is described by Algorithm 1.

In this approach the constraints forces propagate from one simulation iteration to the other; therefore only small adjustments are required. The compliances are also propagated and adjusted dynamically during the simulation. The compliance adjustment is performed using the following scheme.
Algorithm 1 Needle insertion model simulation workflow

Initialize vector \(pn\) with the entry point and a second point in the direction of insertion; \(NP = 2\)
Initialize constraint forces \(f^i_j \leftarrow 0\)
Initialize compliances \(k^i_j \leftarrow K_0 = 0.1\)
\(f_{\text{Insertion}} \leftarrow 0\)
while \(f_{\text{Insertion}} < f_{\text{Max}}\) do
  Distribute \(f_{\text{Insertion}}\) over the needle segments such that \(\sum f^i_0 = f_{\text{Insertion}}\).
  repeat
    Solve \((pn, f, u)\)
    Compute deviations from needle trajectory \(\delta^i_j; \ i = 1 \ldots NP; \ j = 1, 2\)
    Adjust constraint forces \(f^i_j \leftarrow f^i_j + \delta^i_j * k^i_j; \ i = 1 \ldots NP; \ j = 1, 2\)
    Adjust lumped stiffness coefficients \(k^i_j; \ i = 1 \ldots NP; \ j = 1, 2\)
  until \(RMS((\delta^1_1, \delta^1_2, \ldots, \delta^N_P 1, \delta^N_P 2)) < \epsilon\)
  if \(f_{\text{NP}}^{0} > \text{CuttingThreshold}\) then
    Add new point to the \(pn\) vector. \(NP \leftarrow NP + 1\)
  end if
  \(f_{\text{Insertion}} \leftarrow f_{\text{Insertion}} + f_{\text{Increment}}\)
end while

\[
k^i_j = 0.5 \left( k^i_j + \frac{f^i_j - f_{\text{Old}}^i_j}{\delta^i_j - \delta_{\text{Old}}^i_j} \right); \ i = 1 \ldots NP; \ j = 1, 2
\]

where \(f_{\text{Old}}^i_j\) is the previous force \(\delta_{\text{Old}}^i_j\) is the previous alignment error. The proposed approach works with any computational deformable model that can accommodate changing fundamental boundaries described in material coordinates.

2.2. Deformable Object Model

Our implementation uses a quasi-static mesh-less deformable model. Both nonlinear and linear versions were implemented. Throughout the paper the following conventions will be followed. The initial (un-deformed) coordinate system is represented by uppercase \(X\) and the deformed configuration is represented by lower case \(x\). The region occupied by the body in the initial configuration is \(\Omega_X\) and it has a boundary \(\Gamma_X; u(X, t) = \phi(X, t) - X\) is the material displacement. The non-linear elasto-static equations were discretized using the reproducing particles kernel method (RKPM). In a RKPM the deformation is defined using global interpolants [10]. The interpolants can be seen as smoothed dirac functions centered in a certain body coordinate. If there are \(N_P\) particles distributed over the body. Each particle has an associated shape function \(N_I(X): \Omega_X \rightarrow \mathbb{R}\). Then, the particle displacements can be expressed as \(u_i(X) = \sum_{I=1}^{N_P} N_I(X) d_I\). The incremental discretized equation has the form

\[
K_T(d_k) \Delta d = f_{\text{ext}} - f_{\text{int}}(d_k)
\]

where \(K_T(d_k)\) is the tangent stiffness matrix, \(f_{\text{int}}(d_k)\) is the vector of internal elastic forces in the current configuration, and \(f_{\text{ext}}\) is the force applied through the fundamental boundary. The vector of external forces \(f_{\text{ext}}\) is composed of the axial and constraint forces applied by the needle. The displacement of the needle points is returned by the solver as \(u(p_m_i) = \sum_{I=1}^{N_P} N_I(p_m_i) d_I; \ i = 1 \ldots NP\).
2.3. Parallel implementation

The parallel implementation uses a NVidia GTX-480 GPU. This GPU can be seen as very power-full parallel processing cards connected through PCIe-x16 interface to the host computer. One board has a set of multiprocessors connected to a shared memory of approximately 1.5GB. Each multiprocessor has 32 processing cores which execute the same code and can access a fast shared memory. NVidia CUDA consists of a set of C extensions that enables the programmer to launch threads on the GPU. CUDA model requires the programmer to divide the work in threads that can be executed with minimum inter-thread synchronization. Ideally, all threads should be independent. Deformable object models solvers are suitable for this type of implementations. Algorithm 2 describes the work-flow of the parallel solver.

Algorithm 2 Overall GPU solver workflow

Copy \(d\) from CPU to GPU
Copy \(f_{ext}\) from CPU to GPU

repeat

Start GPU threads for computing strains and stresses over the body; One thread will compute the stress in one particle location. The result is deposited in GPU memory.
Wait for the threads to finish
Start GPU threads for computing \(K_T\). One thread will compute the \(K_T\) block corresponding to a pair of overlapping particles. The result is deposited in GPU memory.
Start GPU threads for computing \(f_{int}\). One thread will compute the internal force applied to a particle. The result is deposited in GPU memory.
Wait for the threads to finish
Solve equation 2 for \(\Delta d_{GPU}\) using a PCG algorithm. This algorithm is also implemented on GPU.
Perform \(d_{GPU_{k+1}} \leftarrow d_{GPU_k} + \Delta d_{GPU}\) on the GPU. Compute \(\|\Delta d_{GPU}\|\).

until \(\|\Delta d\| < \epsilon\)
Copy \(d\) from GPU to CPU

The parallel algorithm performs the same high level iterations as the serial algorithm. The speed-up is achieved by implementing in parallel the tangent stiffness matrix and internal forces vector computations as well as the preconditioned conjugate gradient solver. We used a data parallelization technique to convert the serial algorithm into a parallel one. Essentially, one thread will compute the block of the \(K\) matrix corresponding to two interacting particle. Similarly, for the \(f_{int}\) vector, one thread will compute the internal force corresponding to one particle. The \(K_T\) matrix is stored as a sparse matrix in ELL format, therefore the sparse matrix vector multiplication can be performed efficiently. This approach ensured that the data transfer between the host computer memory and GPU memory was minimized. Essentially, for each nonlinear solver iteration we have to copy from the GPU memory to the device memory the vector of particle positions which has the size \(N_p \times n_{sd} \times \text{sizeof(float)}\) bytes; where \(n_{sd}\) is the number of space dimensions.

3. Results

The needle insertion in 2D models was implemented together with a GUI. The interface allows the user to load a MR image or slice. The user defines the fixed boundaries of the object then the object is discretized using a regular grid. Material properties are assigned based on the MR image contrast. For a particle, we first compute the average value of 9 surrounding pixels. If the value is higher than a threshold \(f_T\) then the material properties for that particle are assigned to fat, if they are below a certain threshold there is no material in that area. If they are in-between; they are considered stiff nodules.
Figure 2: Needle insertion simulation A) before the insertion; B) nonlinear solver after the insertion; C) before the insertion; D) linear solver after the insertion; E) Constraint forces as a function of the insertion force; Red crosses are fixed boundaries; Green circles represent particles with fat material properties; Magenta circles represent particles with stiff material properties

| Update Stresses; Assemble $K_T$ and $f_{int}$ | Serial implementation | GPU Implementation |
|------------------------------------------------|-----------------------|-------------------|
| Mean PCG algorithm time                         | 1698ms                | 2ms               |
| Mean Total Solver Time                          | 490ms                 | 5.3ms             |
|                                                 | 6s                    | 32ms              |

Table 1: Timing of different algorithm functions

Then the user has the opportunity to manually adjust the material properties for desired locations. Then, he can select the needle entry site and the insertion direction. The needle insertion simulation is executed while the MR image is warped to display the deformable object.

The material properties were assigned as follows: Lame coefficients for fat: $\lambda = 1.6$, $\mu = 0.03$; Lame coefficients for stiff insertions: $\lambda = 16$, $\mu = 0.3$; Friction coefficient fat = 0.05N/mm; Friction coefficient fat = 0.1N/mm; Cutting force fat = 0.5N; cutting force stiff insertions = 1N.

Several simulations were performed. The maximum insertion force was 3N and the insertion step force step 0.01N. The first set of simulations were performed using a nonlinear solver. Figure 2A presents the non-deformed object with the desired needle insertion direction. Figure 2B presents the object state after the needle insertion. The RMS of needle deviations at the end of the simulation was 0.08mm. The maximum displacement over the entire model was 13mm.

Table 1 presents the timing of each function of the iterative solver. The data shows that the biggest gain is in the $K_T$ matrix assembly time. The data shows that nonlinear insertions simulation can be performed with update rates larger than 20Hz.

From Table 1 it can also be noticed that a linear solver can sustain update rates of almost 200Hz. This is due to the fact that the linear solver has to execute only one PCG solver per iteration. However, Figures 2C and D show that a linear solver underestimate the deformation; this is in agreement with previously reported results. Therefore, in order to achieve accurate simulations a nonlinear model is required.
The constraints force evolution as a function of needle insertion force is presented in Figure 2E. The plot shows a smooth constraint force adaptation during the insertion. The only perturbation occurred when the needle penetrated a stiff insertion. Even in this case, the algorithm recovered and the final deviation from the straight trajectory was under $0.1 \text{mm}$.

4. Conclusions
The paper presents a needle-deformable tissue interaction model using constraint forces. The advantage of the proposed model is that it requires only a general purpose deformable object solver. This provides avenues for the parallelization of the implementation.

The paper presents also numerical results of needle insertions in 2D deformable object. The results include serial as well as GPU accelerated solvers for both nonlinear and linear elasticity. The numerical results illustrate that in order to achieve accurate simulations a nonlinear deformable model has to be used. This is in agreement with previously reported results.

The work will be extended to 3D models as well as deformable needles. More work is required to compare the simulation results against real life needle insertion experiments.

Acknowledgement: The work was partially supported by Natural Sciences and Engineering Research Council and Canadian Foundation for Innovation.

5. References
[1] N. Abolhassani, R. Patel, and M. Moallem. Needle insertion into soft tissue: A survey. Medical Engineering & Physics, 29(4):413–431, May 2007.
[2] R. Alterovitz, M. Branicky, and K. Goldberg. Motion planning under uncertainty for image-guided medical needle steering. The International Journal of Robotics Research, 27(11-12):1361–1374, 2008.
[3] R. Alterovitz, K. Goldberg, J. Pouliot, R. Taschereau, and I-Chow Hsu. Needle insertion and radioactive seed implantation in human tissues: simulation and sensitivity analysis. In Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on, pages 1793 – 1799, 2003.
[4] E. Dehghan and S. Salcudean. Needle insertion parameter optimization for brachytherapy. IEEE TRANSACTIONS ON ROBOTICS, 25:303 – 315, 2009.
[5] S.P. DiMaio and Salcudean S.E. Needle insertion modelling for the interactive simulation of percutaneous procedures. Lecture Notes in Computer Science, 2489:253–260, 2002.
[6] Christian Duriez, Christophe Gubert, Maud Marchal, Stephane Cotin, and Laurent Grisoni. Interactive simulation of flexible needles based on constraint models. In Medical Image Computing and Computer-Assisted Intervention - MICCAI 2009, volume 5762 of Lecture Notes in Computer Science, pages 291–299, 2009.
[7] O. Goksel, S. Salcudean, and S. DiMaio. 3d needle-tissue interaction simulation for prostate brachytherapy. Lecture Notes in Computer Science, 3749:827–834, 2005.
[8] A. Horton, A. Wittek, and K. Miller. Subject-specific biomechanical simulation of brain indentation using a meshless method. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2007, Pt 1, Proceedings, 4791:541–548, 2007.
[9] Q. Li and K. Lee. An adaptive meshless method for modeling large mechanical deformation and contacts. In Robotics and Automation, 2007 IEEE International Conference on, pages 1207 – 1212, 2007.
[10] Shaofan Li and Wing Kam Liu. Meshfree and particle methods and their applications. Applied Mechanics Reviews, 55(1):1–34, January 2002.
[11] S. Misra, K. Ramesh, and A. Okamura. Modeling of tool-tissue interactions for computer-based surgical simulation: A literature review. Presence, 17(S):463491, 2008.
[12] S Misra, K B Reed, B W Schafer, K T Ramesh, and A M Okamura. Mechanics of flexible needles robotically steered through soft tissue. The International Journal of Robotic Research, online, 2010.
[13] H. Nienhuy and van der Stappen A. A computational technique for interactive needle insertions in 3d nonlinear material. In IEEE International Conference on Robotics and Automation, 2004, 2004.
[14] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura. Nonholonomic modeling of needle steering. International Journal Of Robotics Research, 25(5-6):509–525, May 2006.