XR-oriented Medical Elastodynamics: Developing a Versatile 2D Energy-based FEM Framework and Its Applications*

Xu Wang[1,*], Atsushi Konno[1]

[1]Hokkaido University, Sapporo, Hokkaido, 060-0814, Japan

Abstract

Real-time elastodynamic simulations in Extended Reality (XR) environments show promise for supporting remote emergency medical procedures during disasters. However, determining the optimal combination of energy models and computational frameworks for real-time 3D elastodynamic simulations is complex due to the multitude of existing methods and their potential combinations. This paper introduces a 2D energy-based Finite Element Method (FEM) prototyping framework designed to facilitate comparative analysis of various energy models across different computational frameworks. To enable the reproduction of disaster scenarios in 2D space, we propose an algorithm for semi-automatic conversion of arbitrary geometries into 2D FEM-compatible meshes. Using this framework, we simulated a leg trapped under collapsed building debris, computing and visualizing stress distributions to aid medical decision-making. Our framework facilitates the implementation of common energy models and enables efficient comparison of their combinations with various real-time computation frameworks, aiding the selection of optimal approaches for XR-oriented medical elastodynamics simulations.

Keywords

Extended Reality, Medical Elastodynamics, Finite Element Method, Energy-based Models, Disaster Medicine

1. Introduction

In recent years, multiple applications of XR technologies in medical practices have led to notable advancements in remote disaster healthcare [1, 2]. During major disasters, physical barriers may substantially impact the provision of professional medical treatment for victims. One solution is combining elastodynamic simulations with XR technology, which has multiple benefits for medical treatment and medical education training [1]. One challenge is that there are several existing energy models and solvers, each of which has its own strengths and limitations. Determining the optimal combination of these elements for a specific medical scenario is a non-trivial task, requiring thorough comparative analysis and performance evaluation. To address this challenge, we propose a 2D energy-based FEM prototyping framework. This framework is designed to compare various energy models (e.g., Saint Venant-Kirchhoff (StVK), Co-rotational, and Neo-Hookean) across different computational frameworks (e.g., Explicit FEM, Position-Based Dynamics (PBD) [3], and Extended Position-Based Dynamics (XPBD) [4]). It serves as a stepping stone towards more complex 3D simulations. By focusing on 2D representations, we can reduce computational complexity while still capturing essential physical behaviors, allowing for rapid prototyping and evaluation of different approaches.

To demonstrate the practical applicability of our framework, we modeled a case study of a leg trapped under collapsed building debris. This simulation computed and visualized Von Mises stress distributions, providing critical data that could inform medical decision-making in real-world disaster response situations. The contributions of this paper are:

- A versatile 2D energy-based FEM prototyping framework
- An algorithm for semi-automatic conversion of arbitrary geometries into 2D FEM-compatible meshes
- A comparative analysis of common energy models and their combinations with various real-time computation frameworks
- A case study demonstrating the application of the framework in simulating a disaster scenario with potential medical implications

This paper is organized as follows: Section 2 reviews related work in the fields of medical elastodynamics. Section 3 describes the methodology behind our 2D energy-based FEM framework and the semi-automatic mesh conversion algorithm. Section 4 presents the results of our case study and comparative analysis. Finally, Section 5 discusses the limitations of our framework and summarizes future works.

APMAR'24: The 16th Asia-Pacific Workshop on Mixed and Augmented Reality, Nov. 29-30, 2024, Kyoto, Japan

*Corresponding author.
wang.xu@ist.hokudai.ac.jp (X. Wang);
konno@ssi.ist.hokudai.ac.jp (A. Konno)

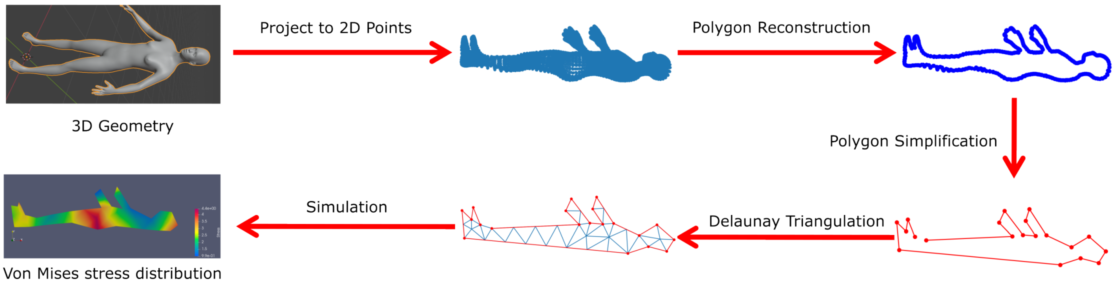

Figure 1: Workflow for converting a 3D human body geometry to 2D FEM-compatible meshes.

2. Related Works

2.1. Finite Element Based Method

The finite element method has a broad range of applications across various fields, including but not limited to engineering, physics, and biomedical sciences [5, 6]. Due to its ability to accurately simulate the characteristics of deformable objects and represent multiple complex materials and structures, it has been widely used in 3D elastodynamics to simulate brittle and ductile fracture of materials [7, 8, 9].

In the field of medical elastodynamics, FEM has found numerous applications. For example, Sase et al. [10] proposed a GPU-accelerated FEM-based approach for simulating brain fissure opening in surgical procedures. They also introduced a volume embedding method [11] to preserve complex topological structures during surgical simulations, and a penalty based method for rigid-deformable objects coupling [12]. In these cases, all of their methods employed the Corotational FEM [13] for rapid computation of organ deformation simulations. Our study extends this line of research by comparing the performance of the Corotational energy model with St. Venant-Kirchhoff (StVK) and Neo-Hookean energy models. Unlike previous studies, we utilize explicit FEM rather than implicit solvers. This approach allows us to simply evaluate the efficacy of different energy models in the context of real-time medical elastodynamics simulations.

2.2. Position Based Method

Position-Based Dynamics (PBD) is a real-time physics-oriented simulation framework initially proposed by Müller et al. in the computer graphics community [3]. Unlike other physics-based simulation methods, PBD primarily focuses on converting physics models into constraint forms and solving them. Although this framework has proven capable of achieving accuracy comparable to physics-based models through multiple iterations of computation, developers primarily use in real-time physical simulations with fewer iterations to achieve satisfactory visual effects. Macklin et al. [4] identified limitations in the PBD framework when simulating elastic objects, as the stiffness of materials depends on the timestep. In response, they introduced an approximate implicit Euler method called extended PBD (XPBD).

Given the stability and ease of parallelization of the PBD framework, numerous applications have emerged for medical elastodynamics simulations. For instance, Tai et al. [14] developed a virtual surgical training system in Augmented Reality (AR), using XPBD for soft tissue elastodynamic simulations. Camara et al. [15] utilized the PBD-based library NVIDIA FleX to develop a simulation platform for exploring optimal material properties and other parameters. Moreover, Yu et al. [16] employed PBD methods in a Virtual Reality (VR) environment to develop a real-time medical education training system. While these studies demonstrate the direct application of PBD/XPBD in medical simulations, our framework develops energy model-based PBD and XPBD approaches, building upon the FEM-PBD method proposed by Bender et al. [17].

3. Energy-based FEM Prototyping Framework

This section will describe the implementation of our developed 2D energy-based FEM prototyping framework. First, we will describe the computation of various energy models when employing explicit FEM methods. Subsequently, we will detail the calculation of different energy models within the position-based approach. Finally, we will introduce the semi-automatic mesh conversion algorithm for 2D scenarios.

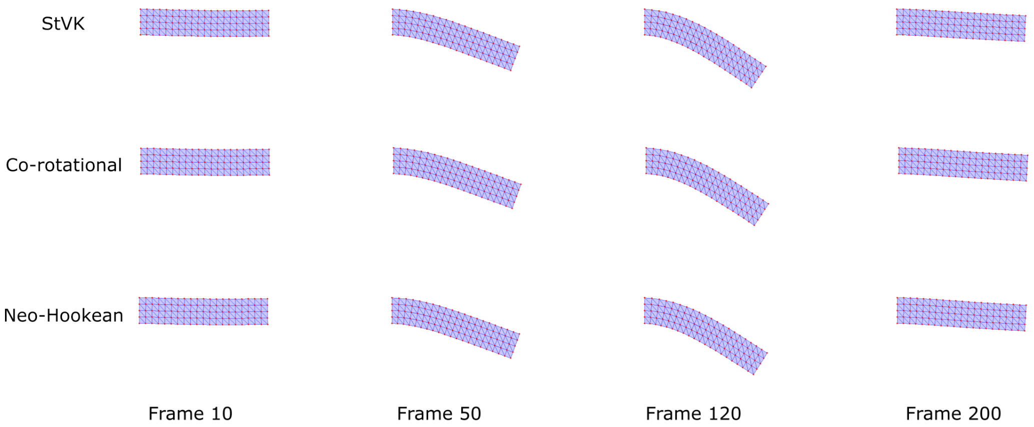

Figure 2: Comparison of three energy models (StVK, Corotational, and Neo-Hookean) in an explicit FEM framework for case A.

3.1. Energy Model in Explicit FEM

Deformation Map In continuum mechanics, the deformation of a material can be conceptualized as an affine mapping $\phi(\cdot)$ from the material space to the deformed world space [5]. This mapping allows us to describe the position of any point in the deformed world space. Specifically, for a material point $\bar{\boldsymbol{x}}$ in the material space, its position $\boldsymbol{x}$ in the world space is given by:

$$\begin{aligned} \boldsymbol{x} &= \phi(\bar{\boldsymbol{x}}) = \boldsymbol{F}\bar{\boldsymbol{x}} + \boldsymbol{U} \\ \boldsymbol{F} &= \frac{\partial \phi(\bar{\boldsymbol{x}})}{\partial \bar{\boldsymbol{x}}} \\ &= \frac{\partial}{\partial \bar{\boldsymbol{x}}}(\boldsymbol{F}\bar{\boldsymbol{x}} + \boldsymbol{U}) \end{aligned} \tag{1}$$

where $\boldsymbol{F}$ is the deformation gradient and $\boldsymbol{U}$ is a translation matrix.

This formulation provides a fundamental basis for analyzing material deformation in our 2D energy-based FEM framework. It enables us to track the movement and deformation of each point in the material over time, which is crucial for computing various energy models and simulating elastodynamic behavior.

Elastic Energy The energy models Ψ used in elastodynamics calculations can be viewed as evaluation functions that measure the degree of material deformation. These can be expressed as [13, 18, 19, 20, 21]:

$$\begin{aligned} \Psi_{\text{StVK}} &= \mu \|\boldsymbol{E}\|_{\text{F}}^2 + \frac{\lambda}{2}(\text{tr}(\boldsymbol{E}))^2 \\ \Psi_{\text{Co-rotational}} &= \mu \|\boldsymbol{F} - \boldsymbol{R}\|_{\text{F}}^2 + \frac{\lambda}{2}(\text{tr}(\boldsymbol{R}^T\boldsymbol{F} - \boldsymbol{I}))^2 \\ \Psi_{\text{Neo-Hookean}} &= \frac{\mu}{2}(\text{tr}(\boldsymbol{F}^T\boldsymbol{F}) - 3) - \mu \log(\det(\boldsymbol{F}^T\boldsymbol{F})) \\ &\quad + \frac{\lambda}{2}\log^2(\det(\boldsymbol{F}^T\boldsymbol{F})) \\ \boldsymbol{E} &= \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}) \\ \mu &= \frac{k}{2(1+\nu)} \\ \lambda &= \frac{k\nu}{(1+\nu)(1-2\nu)} \end{aligned} \tag{2}$$

where $\boldsymbol{R}$ is a matrix representing the pure rotational part of $\boldsymbol{F}$, which must satisfy $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$. $\boldsymbol{E}$ denotes Green's strain tensor. $\text{tr}(\cdot)$ represents the trace of a matrix. The Lamé parameters, denoted as μ and λ, characterize the elastic properties of the material. These coefficients are directly related to two fundamental material constants: Young's modulus k and Poisson's ratio ν. $\det(\cdot)$ denotes the determinant of a matrix. $\|\cdot\|_{\text{F}}^2$ represents the squared Frobenius norm.

Explicit FEM In the explicit FEM framework, we can derive the nodal forces $\boldsymbol{f}$ for each independent element based on various energy models Ψ (i.e., elastic potential

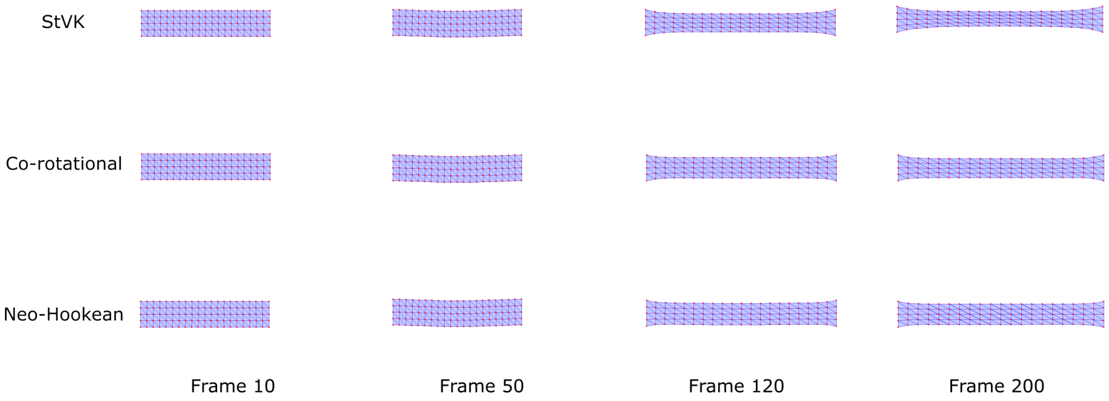

Figure 3: Comparison of three energy models (StVK, Corotational, and Neo-Hookean) in an explicit FEM framework for case B.

energy) as follows:

$$\begin{aligned} \boldsymbol{f} =& -a\frac{\partial\Psi}{\partial\boldsymbol{x}} = -a\frac{\partial\Psi}{\partial\boldsymbol{F}}\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}} \\ =& -a\boldsymbol{P}(\boldsymbol{F})\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}} \end{aligned} \tag{3}$$

Where a represents the area of the element in its undeformed configuration. $\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}$ is an energy-independent matrix that can be derived directly from the defined element. $\boldsymbol{P}(\boldsymbol{F})$ denotes the first Piola-Kirchhoff stress tensor, which can be derived from Equation 2.

Based on the derivation of the aforementioned energy models, we can efficiently decompose the calculation of nodal forces for each element in the explicit FEM method into energy-independent and energy-dependent components. This decomposition facilitates the seamless substitution of various energy models within the explicit FEM framework, enabling straightforward validation of their performance and behavior. The algorithm for the 2D explicit FEM is presented in Algorithm I.

3.2. Energy Model in PBD

To integrate the FEM method into the PBD framework, we primarily developed our framework using the position-based energy reduction method proposed by Bender et al. [17] Additionally, we attempted to implement the energy constraints within the XPBD method [4] and conducted comparative experiments.

The position-based energy reduction method aims to determine position adjustments $\Delta\boldsymbol{x}$ such that the energy function $\boldsymbol{E}(\boldsymbol{x}+\Delta\boldsymbol{x})$ is minimized to zero. For an individual constraint, the position adjustment $\Delta\boldsymbol{x}$ is derived by resolving the linearized equation. This process can be represented as follows:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{x}+\Delta\boldsymbol{x}) \approx \boldsymbol{E}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}\boldsymbol{E}^T(\boldsymbol{x})\Delta\boldsymbol{x} = 0 \\ \nabla_{\boldsymbol{x}}\boldsymbol{E} = \int\frac{\partial\Psi}{\partial\boldsymbol{x}}\mathrm{d}\bar{\boldsymbol{x}} = \int\frac{\partial\Psi}{\partial\boldsymbol{F}}\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}\mathrm{d}\bar{\boldsymbol{x}} \\ = \int\boldsymbol{P}(\boldsymbol{F})\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}\mathrm{d}\bar{\boldsymbol{x}} \end{aligned} \tag{4}$$

To constrain $\Delta\boldsymbol{x}$ to align with the direction of $\nabla_{\boldsymbol{x}}\boldsymbol{E}$, a Lagrange multiplier λ is introduced, which is defined such that:

$$\lambda = -\frac{\boldsymbol{E}(\boldsymbol{x})}{\sum w_j|\nabla_{\boldsymbol{x}_j}\boldsymbol{E}(\boldsymbol{x})|^2} \tag{5}$$

Algorithm I 2D Explicit FEM Algorithm

```
for each finite element $\boldsymbol{D}$ do
    compute $\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}$
    compute $\boldsymbol{P}(\boldsymbol{F})$
    compute $\boldsymbol{f}$ (Equation (3))
    accumulate forces $\boldsymbol{f}$ for each vertex in $\boldsymbol{D}$
end for
$\boldsymbol{a}^n \leftarrow \boldsymbol{M}^{-1}(\boldsymbol{f}+\boldsymbol{f}_{\text{ext}})$
$\boldsymbol{v}^{n+1} \leftarrow \boldsymbol{v}^n + \Delta t\boldsymbol{a}^n$
$\boldsymbol{x}^{n+1} \leftarrow \boldsymbol{x}^n + \Delta t\boldsymbol{v}^n$
damp velocities $\boldsymbol{v}^{n+1}$
```

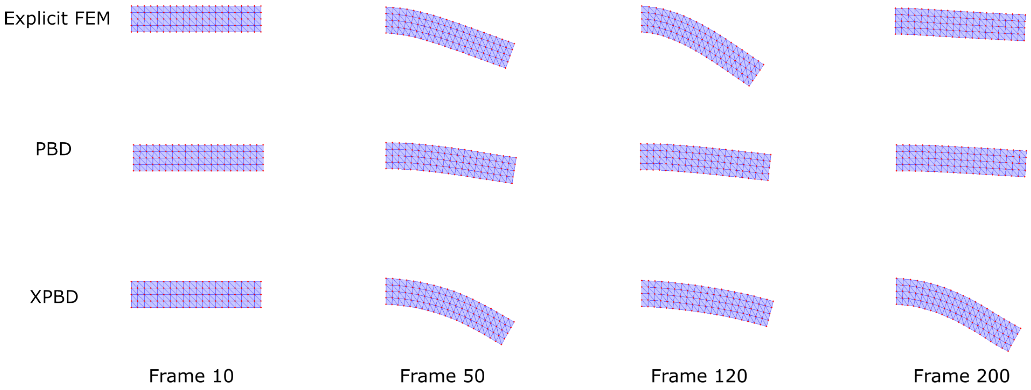

Figure 4: Comparison of StVK energy model performance under simulation scenario case A, implemented in explicit FEM, PBD, and XPBD frameworks.

where $w_i = \frac{1}{m_i}$ represents the inverse of the particle mass, and the position correction for each particle is determined by:

$$\Delta \boldsymbol{x}_i = w_i \lambda \nabla \boldsymbol{E}(\boldsymbol{x}) \tag{6}$$

In contrast to the energy-based FEM framework, the energy-based PBD framework requires a priori conversion of all vertices of each element into a particle-based data structure, ensuring that each vertex possesses independent mass, velocity, and position information. As for XPBD method can be conceptualized as a solver that approximates an implicit Euler solution. It differs from the PBD framework in its computation of position corrections, as illustrated below:

$$\begin{aligned} \Delta \boldsymbol{x}_i =& \boldsymbol{M}^{-1} \nabla \boldsymbol{E}(\boldsymbol{x})^T \Delta\lambda \\ \Delta\lambda =& \frac{-\boldsymbol{E}(\boldsymbol{x}) - \alpha\lambda}{\nabla \boldsymbol{E} \boldsymbol{M}^{-1} \nabla \boldsymbol{E}^T + \alpha} \end{aligned} \tag{7}$$

where α is a compliance coefficient. The algorithm for the energy-based XPBD is presented in Algorithm II.

3.3. Mesh Conversion

To facilitate the simulation of complex and irregular geometric shapes in elastodynamics within our developed framework, we propose a computational pipeline capable of transforming arbitrary geometries into 2D FEM-compatible meshes. The following sections elucidate the step-by-step implementation process of this computational pipeline.

Project 3D Geometries to 2D Points To project 3D geometries onto a 2D plane, we first establish a camera coordinate system using three vectors: camera position $\boldsymbol{c}$, focal point $\boldsymbol{f}$, and up vector $\boldsymbol{u}$. From these, we derive the orthonormal basis: $\hat{\boldsymbol{f}} = \frac{\boldsymbol{f}-\boldsymbol{c}}{|\boldsymbol{f}-\boldsymbol{c}|}$, $\hat{\boldsymbol{r}} = \frac{\boldsymbol{f}\times\boldsymbol{u}}{|\boldsymbol{f}\times\boldsymbol{u}|}$, $\hat{\boldsymbol{u}} = \hat{\boldsymbol{r}} \times \hat{\boldsymbol{f}}$ We then construct the camera-to-world transformation matrix $\boldsymbol{T}$:

$$\boldsymbol{T} = \begin{bmatrix} r_\mathrm{x} & u_\mathrm{x} & -f_\mathrm{x} & C_\mathrm{x} \\ r_\mathrm{y} & u_\mathrm{y} & -f_\mathrm{y} & C_\mathrm{y} \\ r_\mathrm{z} & u_\mathrm{z} & -f_\mathrm{z} & C_\mathrm{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

For a point $\boldsymbol{p} = [x\ y\ z]$ in world coordinates, we convert it to camera coordinates p' using:

$$\boldsymbol{p}' = [\mathrm{x}\ y\ z\ 1] \cdot \boldsymbol{T}^{-1} \tag{9}$$

Finally, we project $\boldsymbol{p}' = (x', y', z', w')$ onto the 2D plane as $\boldsymbol{p}_{\mathrm{proj}} = (x', y')$.

Polygon Reconstruction from 2D Point Clouds After projecting the 3D geometry to 2D, we obtain a large set of 2D point data that needs to be processed. Our next task is to reconstruct the polygon shape information based on these 2D point data. While the most straightforward approach typically involves using a convex hull algorithm to obtain the contour information of these points, this method struggles to handle concave polygon shapes. Therefore, we opted to employ the alpha shapes algorithm [22] for 2D polygon shape reconstruction. An experimental result of applying the alpha shape algorithm can be found in Figure 1.

Table 1
Vertical Displacement of Cantilever Beam Tip Node (mm)

Energy Model	Frame 10	Frame 50	Frame 120	Frame200
StVK	49	1269	1948	246
Co-rotational	49	1270	1880	272
Neo-Hookean	49	1271	1865	279

Generation of FEM-Compatible Meshes from 2D Polygon Directly converting the polygons into FEM-compatible meshes is challenging due to the massive number of vertices in polygons. This is because numerous vertices allow for accurate representation of the original geometric object shape, while they complicate the Delaunay triangulation process, which is essential for producing FEM-computable meshes. As an example of complex polygon shapes, Delaunay triangulation often generates numerous small-area triangles. These small triangles can impact both the stability and computational efficiency of FEM solvers. To overcome this challenge, we implement a polygon simplification step before applying Delaunay triangulation. The Douglas-Peucker algorithm primarily forms the basis of the polygon simplification process; we choose it for its efficiency in reducing the number of vertices while preserving essential shape characteristics. For Delaunay triangulation, we additionally incorporate maximum area and minimum angle constraints to produce FEM-compatible meshes. The complete mesh conversion process is illustrated in Figure 1.

4. Experimental Results

Case A In this case study, we simulate a cantilever beam (5 m long, 1 m high), which is fixed at its left end, and utilize explicit FEM frameworks with different energy models to test its deformation under gravity and evaluate computational performance. The following experimental parameters were used: a Young's modulus of 1000 Pa, a Poisson's ratio of 0.3, a mass of 1 kg, and a time step of 0.001 s. As illustrated in Figure 2, the simulation results show that there are no significant differences in visual effects across the different models for the cantilever beam deformation. Furthermore, to precisely capture the similarities and differences among these energy models, we recorded the variations in the vertical displacement of the cantilever beam tip node, as presented in Table 1. The statistical results indicate that the displacement amounts generated through deformation are relatively similar for the Co-rotational and Neo-Hookean models.

Algorithm II 2D Energy Based XPBD Algorithm

```
$\boldsymbol{v}^{n+1} \leftarrow \boldsymbol{v}^n + \Delta t \boldsymbol{M}^{-1} \boldsymbol{f}_{\text{ext}}$
$\boldsymbol{x}^{n+1} \leftarrow \boldsymbol{x}^n + \Delta t \boldsymbol{v}^{n+1}$
$\lambda \leftarrow 0$
for iter := 1 to maxIterations do
    for each energy constraint $\boldsymbol{E}(\boldsymbol{x})$ do
        compute $\Delta\lambda$
        compute $\Delta\boldsymbol{x}_i$
        $\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta\boldsymbol{x}$
        $\lambda \leftarrow \lambda + \Delta\lambda$
    end for
end for
$\boldsymbol{v}^{n+1} \leftarrow \frac{1}{\Delta t}(\boldsymbol{x}^{n+1} - \boldsymbol{x}^n)$
damp velocities $\boldsymbol{v}^{n+1}$
```

As for the computational performance, the StVK model and the Neo-Hookean model require approximately 14 seconds to compute 2000 steps, whereas the corotational method necessitates nearly 20 seconds. Additionally, we conducted an experiment to test the stability of different energy models by increasing the Young's modulus and reducing the time step. The results of this experiment demonstrated that all of the energy models can achieve stable simulation for the cantilever beam.

Given these results, the StVK model emerges as the optimal choice for this cantilever beam case study. It not only ensures simulation stability across different parameter settings but also demonstrates the highest computational efficiency.

Case B In this case study, we initially fix both ends of the bar-shaped elastic body. Subsequently, we apply a stretching operation to the nodes on the right side of the elastic body. The fundamental experimental parameters remain consistent with those in Case A. The final experimental results are presented in Figure 3. Examining the longitudinal contraction of the elastic body post-stretching, we observe that the StVK model exhibits more pronounced contraction compared to the corotational and Neo-Hookean models. However, when we attempted to test the stability of various energy models by increasing the Young's modulus and reducing the time step, the StVK model required a smaller time step than the corotational and Neo-Hookean models to achieve stable simulations. From these observations, we can conclude that the StVK energy model is the optimal choice when simulating elastic bodies with lower Young's module in 2D scenarios. However, when simulating objects

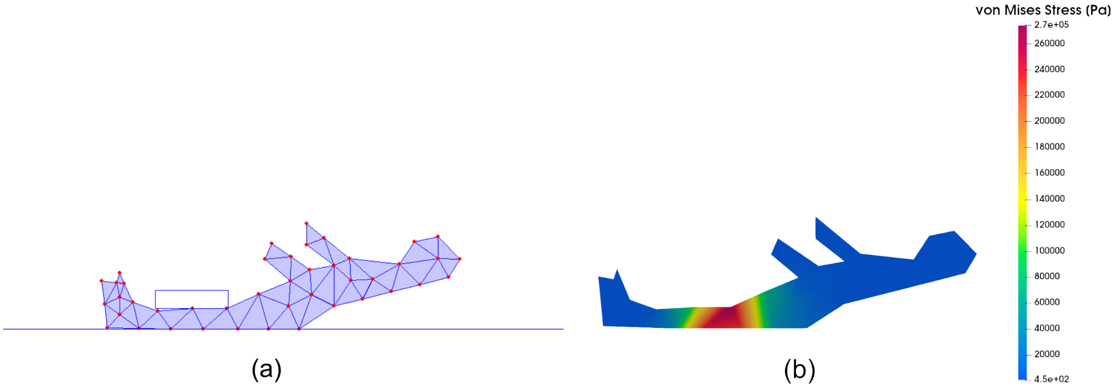

Figure 5: 2D simulation of a leg trapped under debris (case D). (a) Simulation scenario. (b) Visualization of von Mises stress distribution.

with higher Young's module and prioritizing computational performance and stability, the corotational and Neo-Hookean models prove more suitable.

Case C In this case study, we adopt the same experimental setup as in Case A. However, we will evaluate and compare the performance of the StVK model implemented within three different computational frameworks: explicit FEM, PBD, and XPBD. The simulation results in Figure 4 show that even when using the same energy model, elastic bodies exhibit different behaviors under PBD and XPBD frameworks. It should be noted here that the deformation under the PBD framework is not significant. This is because the material stiffness in the PBD framework depends on the time step. From this observation, we should be cautious when using PBD-based frameworks in medical dynamics simulations, especially when simulation accuracy is required.

Case D In this case study, we utilized the mesh conversion method described in Section 3.3 to project a 3D human body model onto a 2D plane. This projection enabled us to conduct an elastodynamic experiment simulating a simplified scenario where a human leg is trapped under building debris. For this study, we used the following parameters: Young's modulus of 1.85 MPa, Poisson's ratio of 0.3, total mass of 70 kg, and time step of 0.0001 s. We employed the Neo-Hookean energy model and Explicit FEM framework as our solver. Through visualization techniques, we were able to illustrate the stress distribution on the leg when compressed by the debris. This approach demonstrates the practical application of our framework in simulating medical scenarios relevant to disaster response.

5. Conclusion

In this paper, we develop an energy model-based 2D FEM framework for XR-oriented medical elastodynamics. The main goal of developing this framework is to assist in determining the optimal combination of energy models and computational frameworks for real-time 3D elastodynamic simulations. Through comparative experiments conducted in 2D space across 4 case studies, we analyze the performance of various energy models and elastodynamic frameworks, subsequently identifying their applicable simulation scenarios. Furthermore, to enable our proposed framework to simulate elastodynamics of more complex geometries, we introduce an algorithm for semi-automatic conversion of arbitrary geometries into 2D FEM-compatible meshes. We successfully employ the resulting meshes in dynamic simulations. However, it should be noted that our framework does not include parallel versions of the PBD and XPBD approaches, which limits our ability to demonstrate potential performance advantages in our comparative experiments. In the future, we would like to implement parallel algorithms for each approach and incorporate an implicit FEM solver for comparing the stability of existing methods.

Recently, deep learning-based FEM approaches [23, 24, 25, 26] have emerged and demonstrated superior accuracy and performance compared to classical FEM methods, with some implementations achieving real-time performance. However, our literature review reveals that these approaches have primarily been validated in lower-dimensional spaces, with most demonstrations limited to 1D and a few in 2D scenarios. While authors suggest that their methods can be extended to 3D applications, the performance and interactive capabilities in such scenarios remain unverified. In contrast, all energy-based methods examined in our comparative study have been proven effective and stable in 3D implementations, which is not yet the case for deep learning-based approaches. This observation partly motivates our choice of methods for comparison. As future work, we plan to investigate the stability of deep learning-based FEM methods and evaluate their effectiveness in 3D scenarios, particularly for XR-oriented medical dynamics applications.

Acknowledgments

This work was supported by Innovative Science and Technology Initiative for Security Grant Number JPJ004596, ATLA, Japan.

References

[1] V. R. Curran, X. Xu, M. Y. Aydin, O. Meruvia-Pastor, Use of extended reality in medical education: An integrative review, Medical Science Educator 33 (2023) 275–286. URL: https://doi.org/10.1007/s40670-022-01698-4. doi:10.1007/s40670-022-01698-4.

[2] D. Jones, S. Fealy, D. Evans, R. Galvez, Editorial: The use of extended realities providing better patient outcomes in healthcare, Frontiers in Medicine 11 (2024). URL: https://www.frontiersin.org/journals/medicine/articles/10.3389/fmed.2024.1380046. doi:10.3389/fmed.2024.1380046.

[3] M. Müller, B. Heidelberger, M. Hennix, J. Ratcliff, Position based dynamics, J. Vis. Comun. Image Represent. 18 (2007) 109–118. URL: https://doi.org/10.1016/j.jvcir.2007.01.005. doi:10.1016/j.jvcir.2007.01.005.

[4] M. Macklin, M. Müller, N. Chentanez, Xpbd: position-based simulation of compliant constrained dynamics, in: Proceedings of the 9th International Conference on Motion in Games, 2016, pp. 49–54.

[5] K.-J. Bathe, Finite element procedures, Klaus-Jurgen Bathe, 2006.

[6] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, Butterworth-Heinemann, Oxford, 2013. doi:10.1016/C2009-0-24909-9.

[7] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson, Physically based deformable models in computer graphics, in: Computer graphics forum, volume 25, Wiley Online Library, 2006, pp. 809–836.

[8] J. F. O'brien, J. K. Hodgins, Graphical modeling and animation of brittle fracture, in: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999, pp. 137–146.

[9] J. F. O'brien, A. W. Bargteil, J. K. Hodgins, Graphical modeling and animation of ductile fracture, in: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 2002, pp. 291–294.

[10] K. Sase, A. Fukuhara, T. Tsujita, A. Konno, Gpu-accelerated surgery simulation for opening a brain fissure, Robomech Journal 2 (2015) 1–16.

[11] K. Sase, T. Tsujita, A. Konno, Embedding segmented volume in finite element mesh with topology preservation, in: Medical Image Computing and Computer-Assisted Intervention-MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III 19, Springer, 2016, pp. 116–123.

[12] K. Sase, T. Tsujita, A. Konno, Haptic rendering of contact between rigid and deformable objects based on penalty method with implicit time integration, in: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2016, pp. 1594–1600.

[13] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, B. Cutler, Stable real-time deformations, in: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2002, pp. 49–54.

[14] Y. Tai, J. Shi, J. Pan, A. Hao, V. Chang, Augmented reality-based visual-haptic modeling for thoracoscopic surgery training systems, Virtual Reality & Intelligent Hardware 3 (2021) 274–286.

[15] M. Camara, E. Mayer, A. Darzi, P. Pratt, Soft tissue deformation for surgical simulation: a position-based dynamics approach, International Journal of Computer Assisted Radiology and Surgery 11 (2016) 919–928. URL: https://doi.org/10.1007/s11548-016-1373-8. doi:10.1007/s11548-016-1373-8.

[16] P. Yu, J. Pan, H. Qin, A. Hao, H. Wang, Real-time suturing simulation for virtual reality medical training, Computer Animation and Virtual Worlds 31 (2020) e1940. doi:https://doi.org/10.1002/cav.1940.

[17] J. Bender, D. Koschier, P. Charrier, D. Weber, Position-based simulation of continuous materials, Computers & Graphics 44 (2014) 1–10.

[18] J. Bonet, R. D. Wood, Nonlinear continuum mechanics for finite element analysis, Cambridge university press, 1997.

[19] G. Irving, J. Teran, R. Fedkiw, Invertible finite elements for robust simulation of large deformation, in: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2004, pp. 131–140.

[20] E. Sifakis, J. Barbic, Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction, in: Acm siggraph 2012 courses, 2012, pp. 1–50.

[21] T. Kim, D. Eberle, Dynamic deformables: imple-

mentation and production practicalities, in: ACM SIGGRAPH 2020 Courses, 2020, pp. 1–182.

[22] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, Alpha shapes: definition and software, volume 63, 1995.

[23] J. Jung, K. Yoon, P.-S. Lee, Deep learned finite elements, Computer Methods in Applied Mechanics and Engineering 372 (2020) 113401. URL: https://www.sciencedirect.com/science/article/pii/S0045782520305867. doi:https://doi.org/10.1016/j.cma.2020.113401.

[24] R. Phellan, B. Hachem, J. Clin, J.-M. Mac-Thiong, L. Duong, Real-time biomechanics using the finite element method and machine learning: Review and perspective, Medical Physics 48 (2021) 7–18.

[25] R. E. Meethal, A. Kodakkal, M. Khalil, A. Ghantasala, B. Obst, K.-U. Bletzinger, R. Wüchner, Finite element method-enhanced neural network for forward and inverse problems, Advanced Modeling and Simulation in Engineering Sciences 10 (2023) 6.

[26] M. Lienen, S. Günnemann, Learning the dynamics of physical systems from sparse observations with finite element networks, arXiv preprint arXiv:2203.08852 (2022).