IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 July 2025, accepted 22 September 2025, date of publication 1 October 2025, date of current version 21 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3616629

RESEARCH ARTICLE

Group-Based Corotational FEM for Real-Time Large Deformation Simulation

SIYU WANG, YUNXIU XU, (Graduate Student Member, IEEE), AND SHOICHI HASEGAWA
Department of Information and Communication Engineering, Institute of Science Tokyo, Yokohama 226-8501, Japan

Corresponding author: Siyu Wang (siw131@haselab.net)

ABSTRACT Real-time simulation of large deformations in soft bodies has long faced a trade-off between computational efficiency and physical accuracy. This paper presents a novel Local Linear Corotated Finite Element Method that addresses this challenge through a hybrid strategy combining direct and iterative solvers. Our method decomposes the simulation domain into element groups; the dynamics within each group are resolved using a pre-computed direct method, while inter-group interactions are handled by efficient iterative constraints. This domain decomposition enables significant pre-computation, replacing runtime equation solving with fast matrix-vector multiplications. Experimental results demonstrate that the method achieves a better balance of speed and accuracy, demonstrating higher fidelity than similarly fast methods. The method supports anisotropic materials and achieves low volume change (2.95% error) under nearly incompressible conditions. Furthermore, its group-based architecture exhibits high scalability on multi-core processors, reducing computation time to 19.6 ms for a 120,000-tetrahedron model using 64 threads. These characteristics make our method well-suited for demanding applications such as surgical simulation, haptic feedback systems, and real-time digital twins, where both accuracy and performance are critical.

INDEX TERMS Corotational finite element method, computational mechanics, deformable simulation, domain decomposition, finite element analysis, haptic feedback, large deformation, parallel computing, position-based dynamics, real-time simulation, soft body physics, virtual reality.

I. INTRODUCTION

Advancements in virtual reality technology and the emergence of Digital Twins have underscored the critical need for accurate and robust real-time simulations of large deformations in soft bodies. Such simulations are essential for enabling realistic interactions within virtual environments, particularly in applications ranging from gaming and entertainment to engineering design and scientific visualization.

Medical simulation represents a particularly demanding application domain, encompassing surgical training, patient-specific preoperative planning, and intra-operative guidance. These scenarios impose stringent and often conflicting requirements: haptic feedback loops demand real-time update rates, while clinical validity requires high geometric fidelity. Bridging the gap between these real-time performance needs and the demand for physical accuracy is a central challenge in the field and the primary motivation for our work.

To address these performance requirements, recent developments in real-time simulation have produced highly robust and computationally efficient methods well-suited for interactive applications. These include Position-Based Dynamics (PBD) [1], [2], Extended Position-Based Dynamics (XPBD) [3], Projective Dynamics [4], and Vertex Block Descent (VBD) [5]. While these constraint-based approaches are notable for their robustness and speed, they often compromise physical accuracy, limiting their applicability in high-fidelity scenarios such as engineering analysis or medical simulations.

The fundamental limitation of constraint-based methods stems from the slow propagation of constraint information through the mesh when using local iterative solvers. As model resolution or material stiffness increases, convergence requires substantially more iterations. Under fixed

The associate editor coordinating the review of this manuscript and approving it for publication was Su Yan.

computational budgets, this slow convergence produces significant residual errors and visual artifacts, particularly affecting global deformation modes. While increasing the iteration count can improve accuracy, it directly reduces computational speed.

In contrast, direct solvers avoid convergence errors but become computationally prohibitive for large systems. An effective compromise involves solving middle-sized linear systems that balance efficiency and accuracy. Building on this insight, our method decomposes the simulation domain into element groups, each forming a manageable linear subproblem. Through strategic approximations, we precompute and simplify these linear systems, eliminating runtime equation solving in favor of efficient matrix-vector multiplications.

Our approach groups multiple tetrahedra and applies constraints only at shared vertices between groups. This significantly reduces the iteration count required for convergence while maintaining accuracy through accumulated constraint forces that efficiently handle inter-group interactions.

While constraint-based methods prioritize speed, Finite Element Methods (FEM) have long been favored in engineering and medical applications due to their superior accuracy. To address the computational challenges of nonlinear FEM, Corotated FEM was introduced [6], [7], with subsequent research focusing on acceleration techniques including Operator Splitting [8] and Warp Cancelling [9].

Our work builds upon these corotational principles with a key distinction: instead of computing rotations at the element level, we apply a single rotation to entire groups of tetrahedra. This group-level approximation is central to our method's efficiency. The name ''Local Linear Corotated FEM'' reflects our core contributions: ''local'' refers to domain decomposition into groups.

By constraining interactions only at group boundaries, our hybrid approach combines the accuracy of FEM with the efficiency of constraint-based methods. This bridges the gap between fast methods like XPBD [2] and traditional corotated FEM, achieving both computational efficiency and physical accuracy for real-time large deformation simulations.

II. RELATED WORK

Real-time large-deformation simulation has evolved through multiple research paradigms, each addressing different aspects of the fundamental trade-off between computational efficiency and physical accuracy. While significant progress has been made, persistent challenges remain, creating a distinct research gap that our work aims to address. This section reviews the key methodological approaches, highlighting their contributions and, critically, their limitations to position our proposed method.

A. CONSTRAINT-BASED ITERATIVE METHODS

Constraint-based methods [10] have gained popularity for real-time applications due to their computational robustness and ease of implementation. Position-Based Dynamics (PBD) [1], a pioneering approach, updates particle positions by iteratively projecting geometric constraints. The method's versatility has led to widespread adoption across diverse applications, including rigid body dynamics [11], fluid simulation [12], and real-time sponge modeling [13]. While effective for achieving interactive frame rates, this approach deviates from the principles of continuum mechanics, making it dependent on tuning and iteration counts for stability rather than on physical material parameters. Building on PBD, Bender et al. [2] introduced a method for simulating continuum materials. Extended Position-Based Dynamics (XPBD) [3] addresses some of PBD's limitations by introducing compliance parameters that correspond to well-defined elastic potentials, making constraint stiffness independent of time step and iteration count. However, XPBD still relies on iterative constraint solving and suffers from slow constraint propagation through the mesh. For materials with high stiffness or in high-resolution models, achieving convergence within computational budgets remains challenging, particularly for global deformation modes that require information to propagate across many elements.

Projective Dynamics (PD) [4] represents a significant architectural improvement by reformulating implicit Euler integration using specially designed energy potentials that enable local-global alternating optimization. In the local step, each constraint is projected onto its constraint manifold independently, while the global step finds a compromise between all projections while accounting for inertia and external forces. This approach bridges the gap between FEM and Position-Based Dynamics by providing a rigorous continuum mechanics foundation while maintaining computational efficiency through the local-global structure. More recent methods like Vertex Block Descent (VBD) [5] further close the gap by converging to implicit Euler integration solutions through block coordinate descent. Despite its improved physical fidelity over PBD/XPBD, VBD is still fundamentally an iterative solver. Its convergence rate degrades with increasing material stiffness, and it can struggle to resolve global deformation modes within the tight time constraints of real-time haptic feedback loops.

Other techniques, such as Shape Matching [14], [15], offer unconditional stability and high speed by enforcing shape preservation on clusters of vertices. However, they sacrifice physical accuracy to such an extent that they are primarily suitable for applications where only visual plausibility is required, not mechanical fidelity. In summary, while constraint-based methods offer speed and robustness, their reliance on iterative schemes and simplified physical models fundamentally limits their accuracy and makes them ill-suited for engineering or medical applications where high fidelity to continuum mechanics is paramount.

B. FINITE ELEMENT AND COROTATIONAL METHODS

In contrast to constraint-based approaches, Finite Element Methods (FEM) [16] are the gold standard for physically accurate simulation. By directly solving discretized continuum mechanics equations, FEM ensures strict adherence to material properties and mechanical laws. However, the computational cost of standard nonlinear implicit FEM, which requires solving large, non-linear systems of equations at each time step, is prohibitive for real-time applications.

To bridge this gap, Corotational FEM was introduced as a compromise, linearizing the problem by separating the deformation of each element into a rigid rotation and a small, linear strain component [6], [7]. This allows the use of pre-computed linear stiffness matrices, significantly improving performance. Subsequent research focused on enhancing the stability and efficiency of this framework, for example by eliminating ghost forces arising from inconsistent rotations between elements [17].

Further acceleration techniques have been developed, including Operator Splitting [8], which decouples different energy components for more efficient processing, and Warp Cancelling [9], which avoids re-factorizing the system matrix at every step. Despite these significant advances, a fundamental bottleneck persists: for large-scale systems with hundreds of thousands of elements, the need to assemble and solve a global system matrix, even if linearized, remains computationally expensive. Consequently, even the most advanced corotational FEM methods struggle to achieve the sub-millisecond update rates required for haptic feedback on high-resolution models.

C. DOMAIN DECOMPOSITION AND SUBSPACE METHODS

To address the scalability challenge of global FEM, domain decomposition strategies partition the simulation domain into smaller, more manageable subdomains. The core idea is to solve these subdomains in parallel using efficient direct solvers and then enforce continuity at their interfaces.

Subspace methods represent one popular variant of this approach. Techniques introduced by Barbić and Zhao [18] and others [19], [20] pre-compute reduced-order modal bases for each subdomain, drastically reducing the degrees of freedom. While this enables significant speed-ups, it fundamentally limits the simulation's expressive power. The pre-computed bases enable efficient simulation by constraining deformations to a reduced space, making these methods well-suited for real-time applications such as character animation.

Other approaches focus on coupling full-rank subdomains. Huang et al. [21] demonstrated accelerated computation by partitioning the domain, while recent GPU-based methods have achieved impressive performance through parallel preconditioning [22]. However, the central challenge in these methods lies in the interface coupling. Interface coupling can be handled either by introducing Lagrange multipliers [23], [24], which leads to computationally expensive saddle-point systems, or by forming a Schur complement [25], [26], which requires repeated assembly and factorization of interface matrices. In contrast, our PBD-inspired iterative coupling is matrix-free, introduces no additional degrees of freedom, and converges with small interface errors.

D. OUR APPROACH: BRIDGING PARADIGMS

The preceding review reveals a set of trade-offs. Constraint-based methods are fast but physically inaccurate. Global FEM is accurate but slow. Domain decomposition methods improve scalability but often introduce implementation complexity. Our work is specifically designed to navigate these trade-offs by creating a hybrid architecture that synergistically combines the strengths of these different paradigms.

Our work addresses these trade-offs with a novel hybrid solver that merges two paradigms. By partitioning the domain into element groups, our method applies the accuracy of corotational FEM within each group using efficient, pre-computed direct solvers. For inter-group coupling, we replace the costly assembly and solution of a global constraint matrix with a fast, PBD-inspired iterative process at shared vertices. This two-level architecture avoids the primary bottlenecks of its parent methods: it circumvents the global solve of traditional FEM while ensuring the core dynamics are governed by high-fidelity physics, not the simplified iterative projections common in PBD. The result is a method that retains the physical accuracy of FEM with the scalability needed for real-time simulation of large, complex models.

III. CONTRIBUTION

We propose a novel Corotated FEM method that operates on a group of tetrahedrons, achieving an optimized balance between iterative and direct solvers to address the trade-offs between computational efficiency and residual error. This method is specifically designed for applications where both real-time performance and physical fidelity are critical, such as in medical simulation. Our approach eliminates the need to solve linear equations in the backward Euler method by enabling time stepping through matrix multiplications with current state variables. This significantly reduces computational costs while maintaining accuracy. Moderate group sizes allow for efficient matrix multiplications and parallelizable operations. By grouping multiple tetrahedrons into manageable groups and applying constraints only to shared vertices, we significantly reduce the number of iterations. Inter-group constraints are handled through accumulated constraint forces, ensuring computational efficiency while maintaining accuracy. Unlike methods where each element is handled individually, which results in a higher number of constraints and iterations, our group-based formulation reduces the number of iterations while retaining precision. Additionally, the original stiffness matrix form in FEM is preserved, ensuring compatibility with any material properties supported by linear stiffness FEM, including anisotropic materials. By addressing challenges such as excessive elements and constraints in iterative methods like PBDs or Gauss-Seidel, and avoiding the computational overhead of dense inverse matrices in direct methods, our

TABLE 1. Mathematical notation.

Symbol	Definition
m	Number of groups
G_j	Group j, where $j = 1, 2, \ldots, m$
N_j	Number of vertices in group j
$\mathbf{x}_j^n \in \mathbb{R}^{3N_j}$	Position vector for group j at time step n
$\mathbf{v}_j^n \in \mathbb{R}^{3N_j}$	Velocity vector for group j at time step n
$\mathbf{M}_j \in \mathbb{R}^{3N_j \times 3N_j}$	Mass matrix for group j
$\mathbf{K}_j^0 \in \mathbb{R}^{3N_j \times 3N_j}$	Stiffness matrix for group j in reference configuration
$\mathbf{R}_j \in SO(3)$	Rotation matrix for group j
$\mathbf{f}_{j,c} \in \mathbb{R}^{3N_j}$	Inter-group constraint forces for group j
$\mathcal{S}$	Set of all shared vertices between groups
$\mathcal{N}(l)$	Set of groups sharing vertex l
$\kappa > 0$	Constraint stiffness parameter
$\Delta t > 0$	Time step size

method provides an alternative for real-time simulations and other computationally intensive applications. The application scenarios for this method are further discussed in Section V-E.

IV. METHODOLOGY

A. NOTATION AND PROBLEM SETUP

Before presenting the method details, we establish the mathematical notation used throughout this work. Table 1 summarizes the key symbols and their definitions.

The proposed method consists of two computational phases: pre-computation and runtime calculation.

1) PRE-COMPUTATION PHASE

The algorithm first partitions the tetrahedral mesh into groups G_j where $j = 1, 2, \ldots, m$. For each group G_j, we compute:

- The mass matrix $\mathbf{M}_j \in \mathbb{R}^{3N_j \times 3N_j}$, where N_j is the number of vertices in group j;
- The local stiffness matrix $\mathbf{K}_j^0 \in \mathbb{R}^{3N_j \times 3N_j}$ in the reference configuration;
- The constant system matrix $\mathbf{A}_j$ and its inverse $\mathbf{A}_j^{-1}$ as defined in Equation 13;
- The identification of shared vertices S_j between adjacent groups.

2) RUNTIME PHASE

At each time step $n \rightarrow n + 1$, the algorithm performs the following sequence:

1) *Prediction step:* Compute predicted positions $\tilde{\mathbf{x}}_j^{n+1} \in \mathbb{R}^{3N_j}$ for each group using the explicit prediction step (Equation 8);
2) *Rotation extraction:* Extract group rotation matrices $\mathbf{R}_j^n \in SO(3)$ via Shape Matching followed by analytical polar decomposition;
3) *Constraint initialization:* Initialize constraint forces $\mathbf{f}_{j,c}^{n+1,0} \in \mathbb{R}^{3N_j}$ using warm start from the previous time step;
4) *Constraint iterations:* Iterate to resolve inter-group constraints using the scheme defined in Equations 19–21;
5) *State update:* Update final positions $\mathbf{x}_j^{n+1}$ and velocities $\mathbf{v}_j^{n+1}$ for all vertices.

B. GROUP DIVISION

The computational mesh is divided into smaller groups of elements using a coordinate-axis-aligned partitioning strategy. While more sophisticated mesh partitioning strategies such as METIS [27] and balanced graph partitioning methods [28] could potentially yield better load balancing, we adopted this straightforward approach for its implementation simplicity and to facilitate reproducibility. This choice allows us to isolate and evaluate the core contribution of this work—the hybrid solver architecture—independent of partitioning scheme complexities. The impact of more advanced partitioning strategies on performance is therefore considered a direction for future work. Each group should contain a manageable number of elements to avoid negatively affecting computational speed. This segmentation method treats each resultant group as an independent entity, adopting a local coordinate system where the centroid of each group serves as both the origin and the pivot for rotation.

C. STIFFNESS MATRIX

Similar to other corotational methods, such as [29] and [30], our approach also computes an approximation to the exact stiffness matrix, a technique highlighted in [31]. Omitting certain computational terms in real-time applications can improve efficiency, though calculating the exact stiffness matrix enhances stability and robustness. Herein, we pre-calculate the stiffness matrix K_j^0 for each group, where $j = 1, 2, \ldots m$ denotes the index of the group.

D. DYNAMIC EQUATION

1) LOCAL COROTATED FEM FORMULATION

In this study, we assign FEM-based Hooke's Law to the Second law of motion as in Eq. (1).

$$\mathbf{M}_j\ddot{\mathbf{x}}_j + \mathbf{C}_j\dot{\mathbf{x}}_j + \mathbf{K}_j(\mathbf{x}_j - \mathbf{x}_{0,j}) = \mathbf{f}_j \tag{1}$$

where $\dot{\mathbf{x}}_j$ and $\ddot{\mathbf{x}}_j$ are first and second-order differences of $\mathbf{x}_j$. $\mathbf{M}_j$ and $\mathbf{C}_j$ are the mass matrix and damping matrix. Our strategy is to simplify the computations as much as possible by approximating the matrices to block-scalar ones. The mass matrix is constructed using mass lumping [32], written as $\mathbf{M} = \mathrm{diag}[m_1, m_1, m_1, m_2, m_2, m_2, \ldots, m_N, m_N, m_N]$ (N represents number of vertices in the group) while the damping matrix is based on Rayleigh Damping [33]. This approach helps to create a block-scalar damping matrix and minimize the number of updated terms. $\mathbf{f}_j$ is external force, including between group constraint forces $\mathbf{f}_{j,c}$ and other added forces $\mathbf{f}_{j,ext}$ other than constraint forces ($\mathbf{f}_j = \mathbf{f}_{j,c} + \mathbf{f}_{j,ext}$). Furthermore, Eq. (1) can be transformed by the first derivative as follows

$$\mathbf{M}_j\dot{\mathbf{v}}_j = -\mathbf{C}_j\mathbf{v}_j - \mathbf{K}(\mathbf{x}_j - \mathbf{x}_{0,j}) + \mathbf{f}_j. \tag{2}$$

To make the simulation stable, we use the Implicit Euler method to discretize Eq. (2).

$$\mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \Delta t \mathbf{v}_j^{n+1}, \tag{3}$$

$$\mathbf{M}_j \mathbf{v}_j^{n+1} = \mathbf{M}_j \mathbf{v}_j^n + \Delta t(-\mathbf{C}_j \mathbf{v}_j^{n+1} - \mathbf{K}_j(\mathbf{x}_j^{n+1} - \mathbf{x}_j^0) + \mathbf{f}_j^{n+1}). \tag{4}$$

In cases of large deformations, the stiffness matrix must be recalculated at each time step. This recalculation, especially when dealing with high-dimensional stiffness matrices, incurs significant computational cost. To make the stiffness matrix precomputed, we adopted the corotated method [6], along with local coordinates for each group. Then the elastic force can be written as:

$$\begin{aligned}&\mathbf{K}_j(\mathbf{x}_j^{n+1} - \mathbf{x}_j^0)\\ &\quad = \mathbf{R}_j^{n+1}\mathbf{K}_j^0 \left((\mathbf{R}_j^{n+1})^{-1}(\mathbf{x}_j^{n+1} - \mathbf{x}_{j,cm}) - (\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0)\right),\end{aligned} \tag{5}$$

where $\mathbf{x}_{j,cm}^0$ is the initial center of mass and $\mathbf{x}_{j,cm}$ is center of mass in run-time calculations. The rotation matrix $\mathbf{R}_j^{n+1}$ is updated each time step, and the stiffness matrix $\mathbf{K}_j^0$ is constant. Substituting Eq. (5) into (4), we get

$$\begin{aligned}\mathbf{M}_j \mathbf{v}_j^{n+1} = \mathbf{M}_j \mathbf{v}_j^n + \Delta t \big(&- \mathbf{C}_j \mathbf{v}_j^{n+1}\\ &- \mathbf{R}_j^{n+1}\mathbf{K}_j \big((\mathbf{R}_j^{n+1})^T \left(\mathbf{x}_j^{n+1} - \mathbf{x}_{j,cm}^{n+1}\right)\\ &- \left(\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0\right) \big) + \mathbf{f}_j^{n+1}\big).\end{aligned} \tag{6}$$

We then use $\mathbf{C'}_j = \Delta t \mathbf{C}_j$ to simplify. Then substitute Eq. (6) into Eq. (3). We have

$$\begin{aligned}(\mathbf{M}_j + \mathbf{C}'_j)\mathbf{x}_j^{n+1} = {}& (\mathbf{M}_j + \mathbf{C}'_j)\mathbf{x}_j^n\\ &+ \Delta t \mathbf{M}_j \mathbf{v}_j^n - \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j \Big((\mathbf{R}_j^{n+1})^T\\ &\left(\mathbf{x}_j^{n+1} - \mathbf{x}_{j,\mathrm{cm}}^{n+1}\right) - \left(\mathbf{x}_j^0 - \mathbf{x}_{j,\mathrm{cm}}^0\right)\Big)\\ &+ \Delta t^2 \mathbf{f}_j^{n+1}.\end{aligned} \tag{7}$$

We also use predicted position $\bar{\mathbf{x}}_j^{n+1}$(in Eq. (8)) to represent positions adding external force, and our iterative position constraints will correct it.

$$\begin{aligned}(\mathbf{M}_j + \mathbf{C'}_j)\bar{\mathbf{x}}_j^{n+1} = {}& (\mathbf{M}_j + \mathbf{C'}_j)\mathbf{x}_j^n\\ &+ \Delta t \mathbf{M}_j \mathbf{v}_j^n + \Delta t^2 \mathbf{f}_{j,ext}^{n+1}.\end{aligned} \tag{8}$$

Then, substituting Eq. (8) into Eq. (7) and, rearranging terms we arrive at Eq. (9), denoted as

$$\begin{aligned}&\Big(\mathbf{M}_j + \mathbf{C}'_j + \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T\\ &\quad -\Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \mathbf{M}_{j,cm}\Big)\, \mathbf{x}_j^{n+1}\\ &\quad = \left(\mathbf{M}_j + \mathbf{C}'_j\right) \bar{\mathbf{x}}_j^{n+1} + \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0 \left(\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0\right)\\ &\quad + \Delta t^2 \mathbf{f}_{j,c}^{n+1}.\end{aligned} \tag{9}$$

Here, we use $\mathbf{x}_{j,cm} = \mathbf{M}_{j,cm}\bar{\mathbf{x}}_j^{n+1}$, where $\mathbf{M}_{j,cm}$ is the mass distribution matrix of each group. Although this equation may be solved as it is, the following substitutions are made to reduce the computational error. We solve $\Delta \mathbf{x}^{n+1} = \mathbf{x}^{n+1} - \bar{\mathbf{x}}^{n+1}$ rather than $\mathbf{x}^{n+1}$. Then, the equation becomes

$$\begin{aligned}&\Big(\mathbf{M}_j + \mathbf{C}'_j + \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T\\ &\qquad -\Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \mathbf{M}_{j,cm}\Big)\, \Delta\mathbf{x}_j^{n+1}\\ &\quad = \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0 \left(\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0\right)\\ &\qquad - \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \bar{\mathbf{x}}_j^{n+1}\\ &\qquad + \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \mathbf{M}_{j,cm}\bar{\mathbf{x}}_j^{n+1}\\ &\qquad + \Delta t^2 \mathbf{f}_{j,c}^{n+1}, \quad j = 1, \ldots, m.\end{aligned} \tag{10}$$

Here, $\mathbf{f}_c^{n+1}$ means constraint force given by other groups. In this formulation, the rotation matrix is derived by the local coordinates of each group's points using Shape Matching [14] and then decomposed into a 3 × 3 rotation matrix $\mathbf{R}_{j,block}$ using the Analytical Polar Decomposition (APD) method [8]. This rotation matrix is expanded to the entire group, represented as $\mathbf{R}_j^{n+1} = \mathrm{diag}[\mathbf{R}_{j,block}^{n+1}, \mathbf{R}_{j,block}^{n+1}, \ldots, \mathbf{R}_{j,block}^{n+1}]$ (each vertex has one block). The advantage of shape matching lies in its elimination of ghost forces and the fact that APD is straightforward to implement and fast and stable. In Eq (10), the term $(\mathbf{M}_j + \mathbf{C}'_j)^{-1}\Delta t^2 \mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \mathbf{M}_{j,cm}\bar{\mathbf{x}}_j^{n+1}$ is always zero. This is because $\mathbf{M}_{j,cm}\bar{\mathbf{x}}_j^{n+1}$ represents the displacements of the centre of mass, which are uniform across all points. Therefore, the rotation of $\mathbf{M}_{j,cm}\bar{\mathbf{x}}_j^{n+1}$ by $(\mathbf{R}_j^{n+1})^T$ results in a uniform displacement, which does not induce any strain for the stiffness matrix. We eliminate this term to accelerate calculation further. Then Eq. (10) becomes

$$\begin{aligned}&\Big(\mathbf{M}_j + \mathbf{C}'_j + \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T\\ &\qquad -\Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \mathbf{M}_{j,cm}\Big)\, \Delta\mathbf{x}_j^{n+1}\\ &\quad = \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0 \left(\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0\right)\\ &\qquad - \Delta t^2 \mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T \bar{\mathbf{x}}_j^{n+1}\\ &\qquad + \Delta t^2 \mathbf{f}_{j,c}^{n+1}, \quad j = 1, \ldots, m.\end{aligned} \tag{11}$$

2) SIMPLIFICATION AND PRE-COMPUTATION STRATEGIES

To define the damping matrix $\mathbf{C}$, we use Rayleigh Damping [33], $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$. We observed that both left and right sides contain unconstant repeated terms $\mathbf{R}_j^{n+1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T$. Eliminating these terms would reduce matrix multiplications and thus accelerate computations. However, to eliminate rotation terms, $\mathbf{M}_j + \mathbf{C}'_j$ must be block-scalar. Therefore, we set $\beta = 0$ [34], [35] in the Rayleigh Damping, making the damping matrix also block-scalar ($\mathbf{C} = \alpha\mathbf{M}$). This approximation is valid, especially in scenarios where low-frequency vibrations dominate, as the damping effect is primarily governed by the mass-proportional term ($\alpha\mathbf{M}$). Setting $\beta = 0$ simplifies calculations while maintaining sufficient accuracy, ensuring the model's efficiency and effectiveness for real-time simulations. Using a property of the orthogonal matrix ($(\mathbf{R}_j^{n+1})^T = (\mathbf{R}_j^{n+1})^{-1}$), we rearranged the equation and eliminated the common term $\mathbf{R}_j^{n+1}$ on both sides. Finally, the equation to be solved becomes

$$\Delta\mathbf{x}_j^{n+1} = \mathbf{R}_j^{n+1}\left(\mathbf{A}_j\right)^{-1}\mathbf{b}_j^{n+1}, \tag{12}$$

where

$$\begin{aligned}\mathbf{A}_j &= \mathbf{I} + \Delta t^2(\mathbf{M}_j + \mathbf{C}_j')^{-1}\mathbf{K}_j^0 \\ &\quad - \Delta t^2(\mathbf{M}_j + \mathbf{C}_j')^{-1}\mathbf{K}_j^0\mathbf{M}_{j,cm}, \\ \mathbf{b}_j^{n+1} &= \Delta t^2(\mathbf{M}_j + \mathbf{C}_j')^{-1}\mathbf{K}_j^0\left(\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0\right) \\ &\quad - \Delta t^2(\mathbf{M}_j + \mathbf{C}_j')^{-1}\mathbf{K}_j^0(\mathbf{R}_j^{n+1})^T\bar{\mathbf{x}}_j^{n+1} \\ &\quad + \Delta t^2(\mathbf{M}_j + \mathbf{C}_j')^{-1}(\mathbf{R}_j^{n+1})^{-1}\mathbf{f}_{j,c}^{n+1}.\end{aligned} \tag{13}$$

In this equation, $\mathbf{A}_j$ is constant and can be pre-calculated.

Next, we describe how constraint forces between groups are determined. To keep the model's integrity as we divide it into several groups, constraints that keep the nodes shared by the adjacent groups in the same position are added. G_j, G_k are adjacent groups. I_l^{jk} and I_l^{kj} are the index of the common vertex for the two groups. Here $\mathbf{x}_{G_j,I_l^{jk}}^{n+1}$ is a 3-dimensional vector for the vertex position, which is taken from $\mathbf{x}_j^{n+1}$. Assuming there are h common vertices, the constraints become

$$\mathbf{x}_{G_j,I_l^{jk}}^{n+1} = \mathbf{x}_{G_k,I_l^{kj}}^{n+1} \qquad l = 1, 2, \ldots, h, \tag{14}$$

where h is the number of shared vertices.

E. ITERATIVE CONSTRAINT SOLVING

1) CONSTRAINT MODEL DEFINITION

The constraint forces $\mathbf{f}_{j,c}$ are essential for maintaining the integrity of the object across group boundaries. Let $\mathcal{S}$ denote the set of all shared vertices between groups, and let $\mathcal{N}(l)$ be the set of groups that share vertex $l \in \mathcal{S}$.

For each shared vertex l, we define position consistency constraints that ensure the vertex has the same position when viewed from all groups that contain it. Let $\mathbf{x}_{j,l}^{n+1}$ denote the position of shared vertex l as computed by group j at time step $n+1$. The constraint function for vertex l is:

$$\mathbf{C}_l(\mathbf{x}^{n+1}) = \left\{\mathbf{x}_{j,l}^{n+1} - \mathbf{x}_{k,l}^{n+1} : \forall j, k \in \mathcal{N}(l), j \neq k\right\} = \mathbf{0}. \tag{15}$$

This constraint ensures that all groups sharing vertex l predict the same position for that vertex. The complete constraint system consists of all such constraints for all shared vertices:

$$\mathbf{C}(\mathbf{x}^{n+1}) = \left\{\mathbf{C}_l(\mathbf{x}^{n+1}) : \forall l \in \mathcal{S}\right\} = \mathbf{0}, \tag{16}$$

where $\mathbf{x}^{n+1} = [\mathbf{x}_1^{n+1}, \mathbf{x}_2^{n+1}, \ldots, \mathbf{x}_m^{n+1}]^T$ represents the concatenated position vectors of all groups.

2) ITERATIVE CONSTRAINT RESOLUTION

Solving the full coupled system of equations for all groups simultaneously would be computationally expensive and would negate the benefits of domain decomposition. Instead, we employ an iterative scheme that alternates between updating group positions and enforcing inter-group constraints.

a: CONSTRAINT FORCE MODEL

We model the constraint forces as spring-like forces that drive shared vertices toward a mean position. For each shared vertex l, this mean position $\bar{\mathbf{x}}_l$ is computed by averaging the positions predicted by all groups that share this vertex:

$$\bar{\mathbf{x}}_l^{n+1,s} = \frac{1}{|\mathcal{N}(l)|}\sum_{j\in\mathcal{N}(l)} \mathbf{x}_{j,l}^{n+1,s}, \tag{17}$$

where s denotes the constraint iteration index.

b: ITERATIVE ALGORITHM

At each time step $n+1$, we perform a fixed number of inner iterations $s = 0, 1, \ldots, S_{\max}$. The algorithm proceeds as follows:

Initialize constraint forces. We employ a warm start strategy by setting the initial constraint forces to the converged values from the previous time step:

$$\mathbf{f}_{j,c}^{n+1,0} = \mathbf{f}_{j,c}^{n}. \tag{18}$$

This leverages temporal coherence in constraint forces to accelerate convergence.

c: DRIFT PREVENTION THROUGH HYBRID ARCHITECTURE

Unlike pure constraint-based methods such as PBD, our approach exhibits reduced drift when using warm starts due to the hybrid combination of direct solvers and constraint enforcement. In traditional PBD methods, all deformation response relies on iterative constraint corrections, where each correction is approximate and errors accumulate over time when using warm starts. Our method fundamentally differs by employing exact direct solutions within each group using the pre-computed matrix $(\mathbf{A}_j)^{-1}$, ensuring that intra-group dynamics satisfy the underlying physical equations precisely. The constraint forces serve primarily to maintain inter-group continuity rather than to approximate the entire deformation response. This architectural separation means that errors in constraint force estimation do not propagate through the intra-group solutions, effectively isolating drift accumulation. The direct solver component acts as a stabilizing foundation that maintains physical correctness even when constraint force estimates from warm starts contain residual errors.

The following steps are performed in each iteration s:

Solve for position update. First, we solve for the position update $\Delta\mathbf{x}_j^{n+1,s}$ for each group independently using the current estimate of the constraint force $\mathbf{f}_{j,c}^{n+1,s}$:

$$\Delta\mathbf{x}_j^{n+1,s} = \mathbf{R}_j^{n}(\mathbf{A}_j)^{-1}\mathbf{b}_j^{n+1,s}. \tag{19}$$

Algorithm 1 Calculation Scheme

```
Pre-Calculation
 Create Groups G
for Each Group $G_j$ in Groups $G$ do
    for all tetrahedrons $j_e$ ($E_{je} \in G_j$) do
        compute $\mathbf{V}_{je}$          ▷ Volume of each tetrahedron
        compute $\mathbf{M}_{je} \leftarrow \rho \mathbf{V}_{je}$
        compute $\mathbf{K}_{je}$
    end for
    compute $\mathbf{M}_j$ using $\mathbf{M}_{je}$
    compute $\mathbf{K}_j^0$ using $\mathbf{K}_{je}$
    compute $\mathbf{C}_j'$ using $\mathbf{M}_j$ with Rayleigh Damping
    compute $\mathbf{A}_j^{-1}$ using Eq. (13)
    $\mathbf{f}_{j,c}^{n+1,0} \leftarrow 0$
end for
Run-time Calculation
for time step $n$ do
    for Groups $G_j$ do
        compute $\bar{\mathbf{x}}_j^{n+1}$ using Eq. (8) ▷ Predicted Position
        compute $\mathbf{R}_j^{n+1}$                 ▷ Rotation Matrix
    end for
    for Iteration $s$ do
        for Groups $G_j$ do
            compute $\mathbf{b}_j^{n+1,s}$ using Eq. (13)
            compute $\Delta\mathbf{x}_j^{n+1,s}$ using Eq. (19)
            update positions $\mathbf{x}_j^{n+1,s} = \bar{\mathbf{x}}_j^{n+1} + \Delta\mathbf{x}_j^{n+1,s}$
        end for
        for shared vertex $l \in \mathcal{S}$ do
            compute consensus position $\mathbf{x}_l^{n+1,s}$
        end for
        for Groups $G_j$ do
            compute constraint force corrections $\delta\mathbf{f}_{j,l}^s$
                using Eq. (20)
            update $\mathbf{f}_{j,c}^{n+1,s+1}$ using Eq. (21)
        end for
    end for
    for Groups $G_j$ do
         Update Position $\mathbf{x}_j^{n+1}$
         Update Velocity $\mathbf{v}_j^{n+1}$
    end for
end for
```

Compute consensus positions. This gives updated positions $\mathbf{x}_j^{n+1,s} = \bar{\mathbf{x}}_j^{n+1} + \Delta\mathbf{x}_j^{n+1,s}$ for each group. For each shared vertex l, we compute the consensus position as defined in Eq. (17).

Update constraint forces. The constraint force correction for shared vertex l in group j is computed as:

$$\delta\mathbf{f}_{j,l}^s = \kappa(\bar{\mathbf{x}}_l^{n+1,s} - \mathbf{x}_{j,l}^{n+1,s}), \tag{20}$$

where $\kappa > 0$ is the constraint stiffness parameter that controls the strength of the coupling between groups. This correction drives each group's prediction for vertex l toward the consensus position.

TABLE 2. Calculation time for different methods.

Tetra	Ours	LCFEM	WC	OS	XPBD	Shape
1300	0.67 ms	8.93 ms	1.43 ms	0.34 ms	1.15 ms	3.23 ms
120,000	65 ms	6507 ms	117 ms	31 ms	28 ms	66 ms
358,000	210 ms	26598 ms	460 ms	94 ms	128 ms	242 ms

Accumulate constraint forces. The total constraint force for group j is updated by accumulating corrections from all shared vertices:

$$\mathbf{f}_{j,c}^{n+1,s+1} = \mathbf{f}_{j,c}^{n+1,s} + \sum_{l \in S_j} \delta\mathbf{f}_{j,l}^s, \tag{21}$$

where $S_j \subseteq \mathcal{S}$ is the set of shared vertices belonging to group j.

The simulation scheme is shown in Algorithm 1.

V. RESULTS AND DISCUSSION

The evaluation has three aspects: speed, accuracy, and universality. We first compared the computational speed with other methods, including Linear Corotated FEM (LCFEM, implemented in VegaFEM), Operator Splitting (OS), Warp Cancelling (WC), XPBD, and Shape Matching. Then, we compared the speed without grouping and with various group numbers, and finally, we analyzed the impact of multithreading on performance acceleration. For all aspects, including speed and accuracy, we aimed to establish a guide for optimal group numbers and performed comparisons with other methods. We used non-linear FEM as a benchmark for accuracy comparisons. To address the need for large deformation evaluation beyond simple gravity-based scenarios, we also conducted additional tests involving stretching, substantial twisting, and compression-recovery scenarios across diverse geometric configurations, as demonstrated in Fig. 2. The simulations shown in the first two sections were performed on one AMD Ryzen 9 7940HS processor with a Radeon Graphics 780M processor. The third section was conducted with an EPYC 9654QS CPU. Fig. 1 shows an Armadillo divided into 40 groups (4 × 5 × 2), with a fixed right hand, which drops down by gravity. The Young's Modulus and Poisson Ratio are $10^6 Pa$ and 0.45.

A. SPEED

In this section, we evaluate the peak computational performance of our method. Therefore, the constraint solver iteration count was intentionally set to a low value (2 iterations) to measure the best-case scenario for speed, which is critical for real-time applications. The time step Δt was set to 0.01s. We selected materials with Young's Modulus of $10^6 Pa$ and Poisson Ratio of 0.45 across all methods. For XPBD and Shape Matching, we set the iteration number as 1 and substeps number as 5, aligning with common practices for performance-focused comparisons.

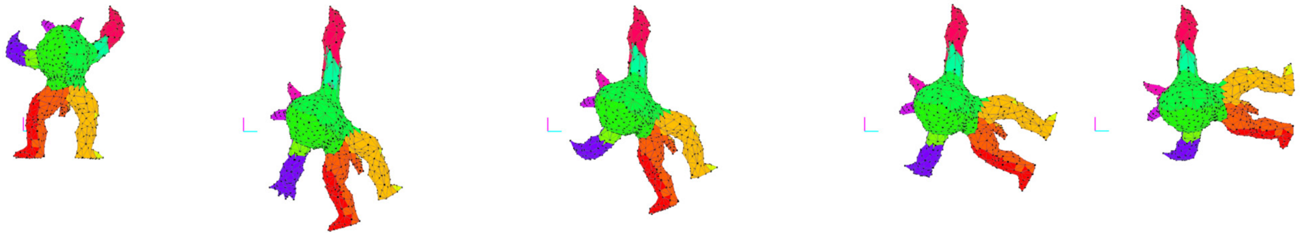

FIGURE 1. Sequential frames of an armadillo model, fixed at the right hand, undergoing free fall under gravity. This visualization showcases the robustness of our simulation method in handling complex deformations and dynamic interactions within a detailed mesh structure, highlighting its effectiveness in accurately replicating the physics of real-time large deformation. Simulation coefficients: Young's Modulus = 100,000 Pa, Poisson Ratio = 0.3, $\kappa = 200$, 5 iterations.

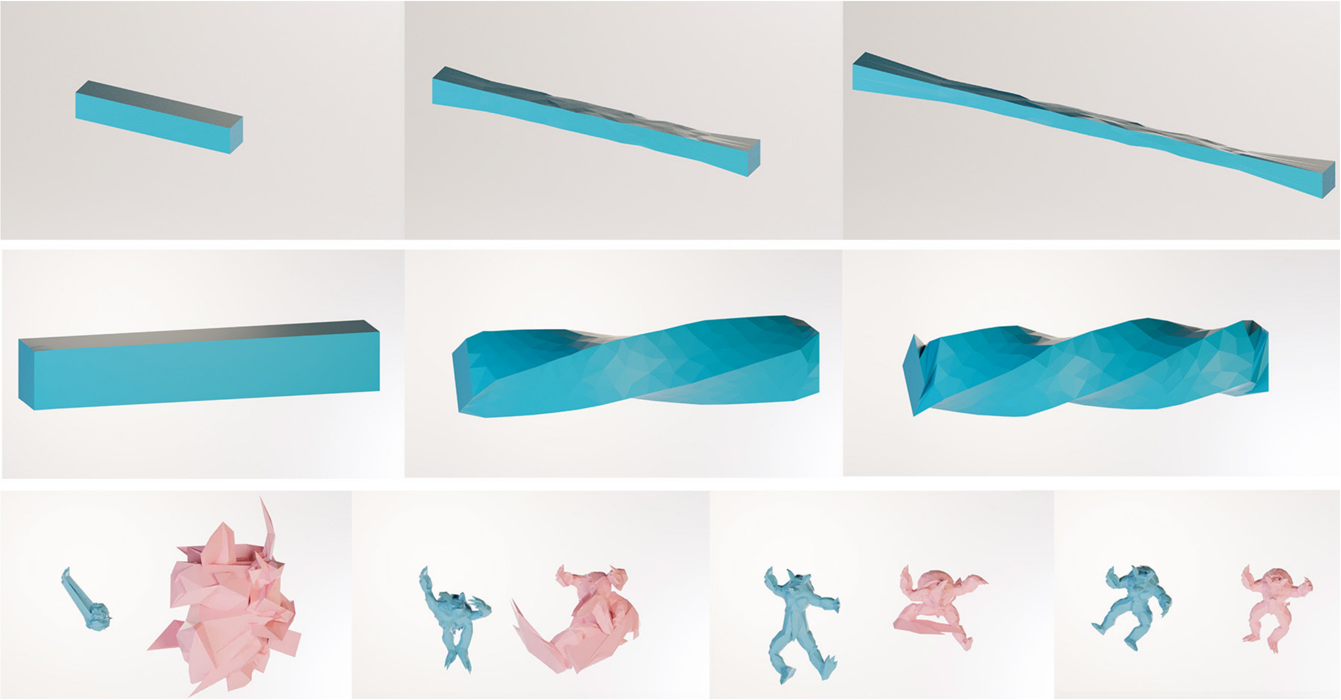

FIGURE 2. Large deformation test scenarios demonstrating the robustness of our method across diverse geometric configurations and loading conditions. Top row: extreme stretching deformation; middle row: twisting induced by opposite rotations on the two end faces about the object's principal axis, while all other rigid-body motion for these faces is constrained; bottom row: complex compression and stretching with deformation recovery. These tests showcase the method's capability to handle large deformations.

TABLE 3. Calculation time for our method and operator splitting using multi-threading.

Method	120,000 Tetra	360,000 Tetra
Run-time (ms)		
Our Method	19.61	71.63
Operator Splitting	31.61	122.64
Precomputation Time (s)		
Our Method	5	16
Operator Splitting	10	84

1) COMPARISON WITH PREVIOUS METHODS

The proposed method was compared with LCFEM, WC, OS, XPBD, and Shape Matching across various tetrahedral mesh sizes. Among these methods, LCFEM, WC and OC are kinds of corotated FEM. For the WC method [9], performance differences between CPUs were assessed by referring to benchmarks and estimating computation speed on the test machine. VBD was excluded from the comparison as it requires GPU computation. The fastest computational results were selected across different group configurations. The results are shown in Table 2.

The proposed method demonstrated significant computational efficiency advantages over the LCFEM at all tested tetra numbers, with calculation times much lower than LCFEM's at every scale. As the number of tetrahedra increases, the method's speed advantage over LCFEM becomes even more pronounced. The proposed method also outperformed WC across all element sizes, showing faster results even as the tetra number becomes large. However, OS is faster than our method at all tetra numbers. In comparison to XPBD, the proposed method was generally

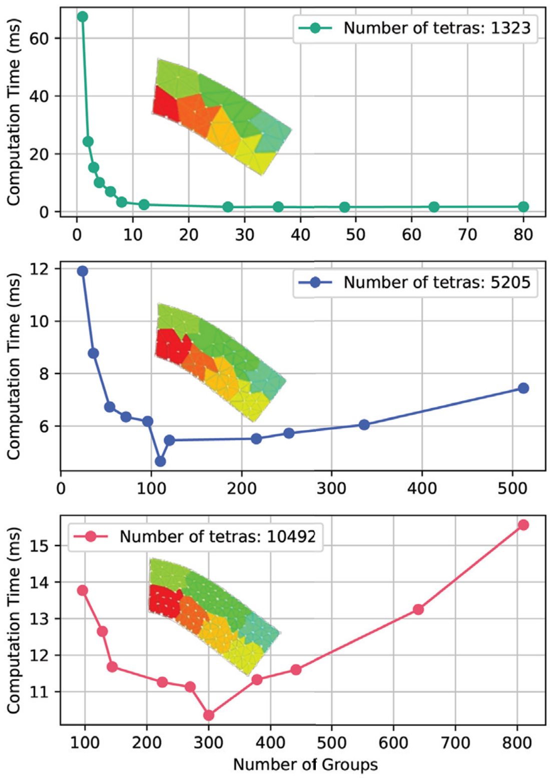

FIGURE 3. Comparison of speed across three numbers of tetras.

slower, though it consistently remained slightly faster than Shape Matching at all tested scales.

2) SPEED ACROSS DIFFERENT GROUP SIZES

The performance evaluation focused specifically on the impact of group count on computational time. The evaluation results are shown in Fig. 3.

The computation time for the simulations generally decreases sharply as the number of groups increases, then rises after a certain point. This trend suggests there is an optimal range of group numbers where computation time is minimized.

This optimal group size exists due to competing computational factors. Small groups suffer from poor parallelization efficiency and increased inter-group constraint overhead, as the computational workload per group becomes insufficient while the number of shared vertices requiring constraint enforcement grows significantly. Large groups, conversely, lead to expensive matrix operations that scale as $O(n^3)$ and reduce parallelizability. The boundary convergence behavior also affects performance, as constraint information must propagate across multiple group boundaries, requiring more iterations when groups are too numerous or poorly configured.

3) SPEED PERFORMANCE REGARDING PARALLEL COMPUTING

To fully explore the performance of parallel computing, Calculation speed was evaluated and compared with Operator Splitting using an EPYC 9654QS CPU (192 threads). The models tested in this section are cantilever beams with tetra numbers 120,000 and 360,000. For both methods, parallel computing was enabled using 64 threads for the proposed method. The results are shown in Table 3.

While the proposed method did not outperform Operator Splitting in the previous single-threaded tests, it demonstrates performance when utilizing multiple threads for both pre-computation and run-time calculations.

TABLE 4. Average error between different methods and ground truth. Top: $E = 10^6 Pa$, Bottom: $E = 10^9 Pa$.

Tetra Size	LCFEM	Our Method	Op	VBD
5084	0.00092	0.02449	0.17485	0.17393
12005	0.00147	0.01699	0.15797	0.1621

Tetra Size	LCFEM	Our Method	Op	VBD
5084	0.0005	0.02804	0.17307	0.17969
12005	0.0007	0.03454	0.15443	0.17634

4) DISCUSSION

The proposed method achieves significant acceleration over LCFEM through the use of multiple groups and demonstrates better performance compared to Warp Cancelling. According to their results [9], their method becomes slower when materials become stiffer. The proposed method does not exhibit this performance degradation. However, it is slower than XPBD, Operator Splitting, but slightly faster than Shape Matching, particularly for high-resolution meshes where handling large matrices becomes challenging.

The results indicate a balance between speed and group numbers. With fewer groups, large-scale matrices slow down the acceleration. As group numbers increase, the computation speeds up due to smaller matrix scales, but too many groups introduce excessive constraints and iterations, reducing the speed again. Practical implementation requires finding the optimal number of groups, to maximize calculation speed.

Additionally, the direction and uniformity of the group divisions affect computation speed. For example, in the case of a Stanford Bunny model including 6k tetras, a subdivision of 27 × 3 × 1 is the slowest (53 ms per frame). A subdivision of 9×9×1 spends 24 ms per frame compared to a 9 × 3 × 3 grouping (20 ms per frame), suggesting that more uniform subdivisions lead to faster computations. This may result from common vertex numbers generated by different groupings. Thus, achieving a balance among speed, number of groups, and division shape is crucial.

Regarding multi-thread computation, the proposed method outperforms Operator Splitting when utilizing a CPU with many threads. Additional threads do not significantly accelerate Operator Splitting, whereas the proposed method exhibits substantial performance gains with increased thread count. This performance difference arises from the group-based calculation structure, which is inherently well-suited for

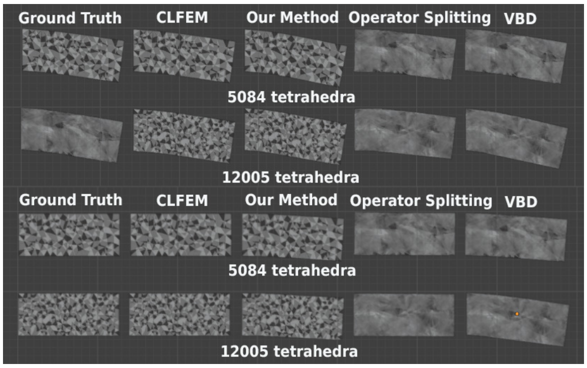

FIGURE 4. Comparison result across all methods. The top image shows the results using $E = 10^6 Pa$, and the bottom image shows the results using $E = 10^9 Pa$.

multi-threaded execution. The accelerated pre-computation phase also reduces scene initialization time, benefiting applications requiring rapid loading.

B. ACCURACY

1) COMPARISION WITH PREVIOUS METHODS

To evaluate the test accuracy, we compared our method against several existing approaches, including Linear Corotated FEM, Operator Splitting, and VBD. In contrast to the speed evaluation, the focus here is on fidelity, thus a higher iteration count (5 iterations) was used for both our method and VBD to ensure sufficient convergence for a fair comparison of accuracy. We conducted experiments using two different tetrahedral mesh resolutions: 5084 and 12,005 elements. Additionally, we tested two different Young's modulus values, 10^6 and 10^9, while keeping the density at $1000kg/m^3$ and the Poisson's ratio at 0.45. XPBD and Shape Matching were excluded from the comparison as they lack physical accuracy, and Warp Cancelling was omitted due to the unavailability of its implementation. The test scenario involved a cantilever beam ($0.5 \times 0.2 \times 0.2$) fixed on the left side, subjected to gravity.

The accuracy of each method was assessed by comparing simulation results to Nonlinear FEM, which served as the ground truth. Specifically, we computed the average error per vertex. For the proposed method, the tetrahedral groupings—($4 \times 2 \times 2$) for the 5084-element mesh and ($6 \times 4 \times 4$) for the 12,005-element mesh—were selected based on performance profiling to balance the costs of intra-group computation and inter-group constraint handling. The parameter κ was set to 100 for the 12,005 tetrahedra mesh and 1000 for the 5084 tetrahedra mesh, following the tuning guidelines described in Section V-F. Both the proposed method and VBD method utilized five iterations to maintain consistency between experiments.

Results comparing the methods at different numbers of tetras and Young's Modulus are shown in Fig. 4 and Table 4.

The proposed method and LCFEM exhibit higher accuracy across both tetrahedra sizes and Young's Modulus. In a qualitative comparison based on the visual results in Fig. 4, the deformation patterns of both the proposed method and LCFEM closely match the Nonlinear FEM ground truth. In contrast, the results from Operator Splitting and VBD show significant deviations. This suggests that our method correctly reproduces the physical deformation behavior. Quantitatively, as shown in Table 4, LCFEM has the lowest average error. The error of our method is lower than that of Operator Splitting and VBD, but higher than that of LCFEM. Operator Splitting and VBD display similar error levels, with marginal differences in performance. Their errors slightly decrease with higher mesh resolution but remain consistently larger than those of LCFEM and our method. Operator Splitting performs better when the material becomes stiff. VBD shows softer results than other methods in higher stiffness cases.

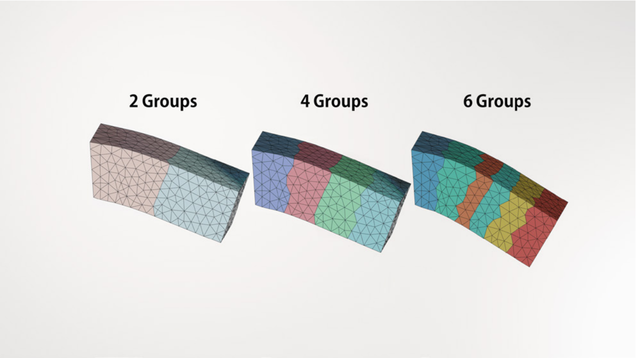

FIGURE 5. The results were that the model was divided into 2 (left), 4 (middle), and 6 (right) groups in the x direction.

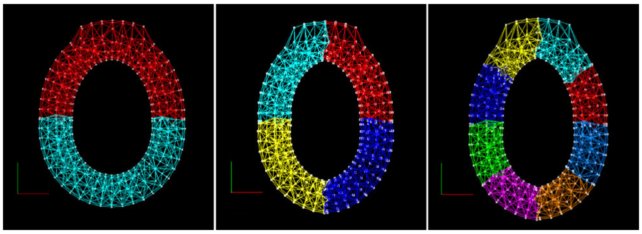

FIGURE 6. Visual results of the ring divided into 2, 4 and 8 groups uniformly along the circumferential direction (treated here as the Z-axis equivalent in our XYZ grouping scheme).

2) GROUP DIVISION

Subsequently, the impact of group division on accuracy was evaluated of the simulation result. The beam was divided into 2, 4, and 6 groups in the X direction to see their deformed states. Then, the effect of using 2 and 4 groups was tested in the Y direction for a ring fixed at the top vertices. The Young's Modulus and Poisson Ratio used in this section are the same as the previous one. κ for each beam model are 2500, 2000 and 1500 respectively. For the rings, κ are 2000, 1000 and 800 respectively. Iteration counts are 5. For the beam, gravity is added, and for the ring, 2 times gravity is added in order to get more significant deformations. The results are shown in Fig. 5 and Fig. 6.

TABLE 5. Volume results for different methods.

Method	Volume	Error Rate (%)
Our Method	0.01941	2.95
Operator Splitting	0.02078	3.899
Linear Corotated FEM	0.01927	3.650

TABLE 6. Average distance between groups.

Groups	Average Distance
01	0.016
12	0.010
23	0.005

When the model is divided into 2 and 4 groups in the x direction, the differences in the deformation patterns are almost indistinguishable. However, when divided into 6 groups, the deformation shows slightly more noticeable differences compared to the previous groupings. For the ring model, we partitioned it along the direction of gravity. Since the model is fixed at the top and deforms under gravity, the primary deformation gradient is vertical. This partitioning strategy was chosen to test the convergence of inter-group constraints under high, uniform stress. As shown in Fig. 6, increasing the number of groups in the circumferential direction from 2 to 8 leads to more natural-looking results, demonstrating the solver's ability to handle these conditions with a sufficient number of partitions.

3) DISCUSSION

The results show that our method's deformation patterns are visually consistent with those of LCFEM and the Nonlinear FEM ground truth, demonstrating qualitative accuracy over the Operator Splitting and VBD methods. This indicates that our method captures the overall deformation behavior.

While our method shows favorable accuracy compared to several accelerated techniques, Table 4 indicates a larger error margin when compared to the high-fidelity Nonlinear FEM ground truth. This discrepancy primarily arises from two fundamental simplifications inherent to our approach. The first is the use of a single, averaged rotation for an entire group of tetrahedra. This approximation, while key to our performance, cannot perfectly capture complex, localized deformations, leading to deviations from the element-wise calculations of standard FEM. The second source of error is the iterative nature of our inter-group constraint solver. Although effective, it may not achieve full convergence, leaving small residual errors at the boundaries between groups. In contrast, traditional FEM formulates and solves a single global system, thereby avoiding these specific sources of approximation error and achieving higher precision, albeit at a significantly greater computational cost.

Both our method and LCFEM consistently maintained accurate deformation patterns across all tetrahedron counts. In contrast, the Operator Splitting and VBD methods exhibited significant variations. This divergence in performance underscores a critical advantage of our approach: its ability to maintain accuracy even as the number of model elements increases, an attribute shared with LCFEM but lacking in the Operator Splitting and VBD methods.

Furthermore, our method strikes a highly favorable balance between accuracy and computational efficiency, consistently outperforming the Operator Splitting and VBD methods in terms of accuracy. Additionally, it closely matches the results of Linear Corotated FEM, which is well known for its accuracy but incurs a higher computational cost.

Several key factors contribute to the performance of our method in capturing accurate deformations.

Unlike the Operator Splitting method, which relies on simplified volume constraints, our approach derives its equations directly from fundamental displacement-force relationships. This foundation enables a more physically accurate representation of material behavior, capturing deformation nuances that simplified constraints might overlook.

Our method also employs a local, iterative position-based constraint strategy to ensure convergence at shared points between groups. This approach preserves the overall structural integrity of the model while allowing for realistic localized deformations. It effectively balances global consistency with local flexibility, an essential feature for accurate large-deformation simulations. In contrast, the VBD method applies element-wise constraints, which pose a challenge when the material becomes stiff or the mesh resolution increases. Under such conditions, VBD requires a larger number of iterations to converge. Given that we use only five iterations, VBD struggles to achieve convergence for each element by the end of the simulation.

To address this, our method adopts an iterative strategy for resolving inter-group constraints, inspired by Position-Based Dynamics. Because our approach involves fewer constraints than VBD, it converges more easily with a small number of iterations. This enables a gradual refinement of the model's state, ensuring continuity and physical plausibility throughout the simulation.

Group division also impacts accuracy. Theoretically, the most accurate deformation would be achieved if each element were treated as an individual group. However, this scenario results in constraints being applied to every element, which significantly increases computational complexity. Our method differs from such approaches by considering a balance between computational efficiency and the number of groups. As discussed in the speed discussion section above, different group divisions lead to varying simulation speeds, with an optimal configuration that maximizes performance. Excessive subdivision into smaller groups can reduce computational speed and hinder model convergence.

C. VOLUME CONSERVATION

Volume conservation is crucial in physics simulations to ensure realism and accuracy, as it upholds fundamental

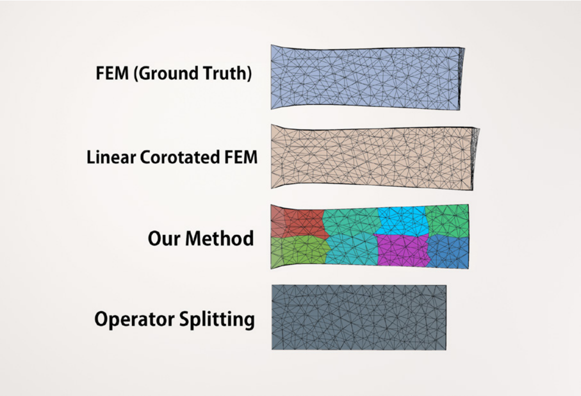

FIGURE 7. Comparing stretching results of the three methods with FEM as the ground truth. Our method shows detailed stretching results, closely aligning with the FEM ground truth. LCFEM also displays noticeable deformation, whereas the Operator Splitting method fails to display accurate stretching.

physical laws. The volume conservation capabilities of the proposed method were evaluated, Corotated Linear FEM, and Operator Splitting, which are all Corotated FEM. The same drag forces as the vertices' gravity were applied to each vertex, while the left side was fixed. The Young's Modulus was set to 20,000 Pa and Poisson Ratio to 0.49 to test the ability to finely conserve volume under such a nearly incompressible condition. κ used here is 200 and the iteration count is 5. The group division used was $4 \times 2 \times 2$.

1) RESULTS

The visual results are presented in Fig. 7. After deformation, the volume was calculated for each method, the error rates between the deformed and initial volumes, and recorded the average norms of neighboring groups in the x direction. The results are shown in Tables 5 and 6. The initial volume of the beam was 0.02, and its length was 0.5.

As shown in Table 5, the volume results indicate minimal differences across the methods, with our method and LCFEM showing a slight volume decrease, while the Operator Splitting method exhibits a slight increase. However, these differences are not significant.

The visual results, presented in Fig. 7, highlight that our method provides more precise and uniform stretching results, closely aligning with the Non-linear FEM ground truth. Both our method and LCFEM exhibit noticeable deformation through stretching, with our method preserving volume more effectively than Corotated Linear FEM. The Operator Splitting method, however, fails to display an accurate cross-section decrease, deviating significantly from the FEM results.

Additionally, Table 6 shows the average distances between neighbouring groups in the x-direction. These results suggest that the proposed method maintains group cohesiveness, considering the overall length of the beam.

2) DISCUSSION

The volume conservation test results further corroborate the accuracy of our method. This advantage is visually reinforced in our stretching simulations. The proposed method demonstrates more detailed stretching results, closely mirroring the FEM ground truth. The Operator Splitting method, however, fails to accurately capture stretching behaviour, showing significant deviations from the FEM results. A possible reason for this observation is that the iterative constraint solver, by enforcing constraints through averaging vertex positions at group boundaries, may regularize the deformation in a way that better preserves volume during stretching scenarios.

However, it is important to acknowledge that corotational formulations, including our method, have inherent limitations regarding volume conservation. Unlike hyperelastic models that can enforce exact volume preservation through appropriate constitutive relationships, corotational methods linearize the material response around the current deformed configuration, which can lead to volume drift over time, particularly in scenarios involving large deformations or long simulation durations. While our results show relatively small volume errors (2.95%), this represents a fundamental limitation of the corotational framework rather than a perfect volume conservation property.

The ability of our method to maintain volume, especially under high Poisson ratios (as demonstrated with a Poisson ratio of 0.49), is particularly noteworthy. This capability is crucial for simulating nearly incompressible materials, which are common in many real-world applications.

Furthermore, the stretching behaviour observed in our method, which aligns closely with FEM results, indicates that the proposed approach can effectively capture detailed deformation modes. This is critical for applications requiring high fidelity in deformation behaviour, such as virtual surgery simulations or detailed material testing in virtual environments. However, there are several factors that need careful consideration to optimize our method.

The accuracy of our method can be influenced by the number of groups used in the simulation. When a high number of groups is employed to enhance computational efficiency, it becomes increasingly challenging to ensure full convergence of shared vertices across all groups. This can result in small gaps or discontinuities between shared vertices, potentially leading to inaccuracies, particularly in regions experiencing large forces.

While our use of Shape Matching and APD for rotation calculations offers computational benefits, it may not capture highly localized or complex non-linear deformations as accurately as more granular, element-wise rotation calculations. However, our approach of applying rotations to local groups rather than globally helps mitigate this limitation to a significant extent.

The effectiveness of our method relies partly on the appropriate tuning of constraint forces between groups. Excessive constraint forces can lead to instability or unrealistic stiffness,

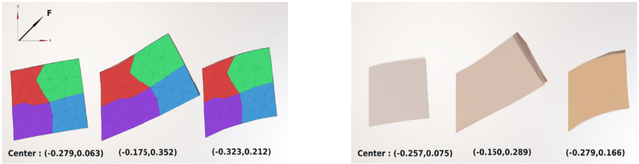

FIGURE 8. Results of Using Isotropic and Anisotropic Stiffness using Our Method (Left) And LCFEM (Right). The left column has a homogeneous Young's Modulus K in all three directions. The middle has 0.1K on the x-axis and K on the y and z axes. The right has 0.1K on the y-axis and K on the x and z-axis. The proposed method in the middle column shows better deformation.

while insufficient forces may compromise model integrity or result in excessive softness. Finding the optimal balance requires careful consideration and may vary depending on the specific simulation requirements.

D. ANISOTROPIC MATERIALS

In this section, the proposed method was tested using orthotropic material parameters to evaluate its universality. A cube model was employed and initially applied an isotropic Young's Modulus of 10^5 Pa. Forces were applied at a 45-degree angle upwards in the positive x direction. Next, the Young's Modulus was changed in the x direction to 10^4 Pa, keeping the y and z directions unchanged, and applied the same forces. Finally, the Young's Modulus was set in the y direction to 10^4 Pa, with the x and z directions remaining at 10^5 Pa, and applied the same forces. In all cases, the model was divided into $2 \times 2 \times 2$ groups. Fig. 8 shows the deformation results for the cube model with varying anisotropic material parameters, compared with LCFEM. The left image represents the cube with a homogeneous Young's Modulus of 10^5 Pa in all directions. The middle image shows the cube with Young's Modulus in the x direction changed to 10^4 Pa, resulting in noticeable stretching in the x direction, highlighting reduced stiffness. The right image depicts the cube with Young's Modulus in the y direction set to 10^4 Pa, showing a different deformation pattern with larger deformation in the y direction, demonstrating decreased stiffness. These results illustrate the impact of anisotropic material parameters on the deformation behaviour of the cube under applied forces.

1) DISCUSSION

In the previous section, we simulated the stretch of an orthotopic material and observed its anisotropy by changing its Young's Modulus in the x and y directions. The results indicate that altering Young's Modulus in the x and y directions, respectively, leads to corresponding changes in deformation. Therefore, it has been demonstrated that the method can simulate some anisotropic materials by reconstructing the stiffness matrix. Compared with the Operator Splitting Method, since its compliant volume constraint did not consider anisotropy, it cannot realize simulations of anisotropic materials based on their current implementation. The current method can be extended to fit various materials.

E. METHODOLOGICAL TRADE-OFFS AND APPLICATION SCENARIOS

Our experimental results define a specific application area for the proposed method: real-time medical simulations that require high-fidelity haptic feedback for complex biological tissues. This scenario, common in surgical training and pre-operative planning, imposes a unique set of simultaneous constraints that exposes the limitations of existing techniques. While PBD-based methods offer the requisite speed for interactivity, their compromised physical accuracy is unacceptable where realistic force feedback and deformation are paramount for effective training. Conversely, traditional FEM provides the necessary fidelity but is computationally prohibitive for real-time interaction.

The proposed method is engineered to thrive in this challenging middle ground. When compared to other fast corotational techniques, its advantages become particularly salient. For instance, simulating nearly incompressible tissues like the liver or brain (a high Poisson's ratio scenario) is a known weakness for many real-time approaches. The proposed method, however, demonstrates better volume conservation (Table 5, Fig. 7), which is essential for realistic tissue response. Furthermore, its inherent support for anisotropic materials (Fig. 8) allows for the accurate modeling of fibrous structures like muscles and tendons, a capability often absent in other accelerated FEM implementations such as Operator Splitting. This unique capacity to couple real-time performance with high-fidelity simulation of complex material models defines its ideal use case. The method's group-based architecture also scales exceptionally well on modern multi-core CPUs (Table 3), making it a future-proof choice for the high-performance workstations typically used in medical research and clinical settings. In essence, this work provides a tailored solution not as a universal replacement, but as the preferred tool for a domain where the stringent demands for speed, accuracy, and material complexity converge.

F. PARAMETER TUNING GUIDELINES

The stability and accuracy of the simulation depend significantly on the inter-group constraint stiffness parameter, κ. Selecting an appropriate value is crucial, as it must be large enough to enforce inter-group continuity without introducing numerical instability. The value of κ is systematically related to several key factors. Specifically, it should be scaled proportionally to the material's stiffness (Young's Modulus, E) and the magnitude of external loads, as greater forces require stronger constraints. Conversely, the value can be reduced for denser meshes, where lower individual vertex mass requires less force to ensure continuity. Similarly, as the number of groups increases, the required κ value per constraint generally decreases to avoid over-stiffening the system.

In practice, we follow a two-stage tuning process. First, we determine an initial order-of-magnitude estimate for κ based on these relationships. Then, we perform fine-tuning by running short simulations and observing the behavior at group interfaces. If gaps appear, κ is incrementally increased until visual continuity is restored. This systematic approach ensures that the chosen κ for each experiment is not arbitrary but is instead a choice that balances physical realism with computational stability.

We will explore the precise mathematical relationship between κ, the mass of individual points, the number of points in each group, and the stiffness matrix in future work.

VI. CONCLUSION

In this paper, we introduced the Local Corotated FEM, an approach developed for real-time simulations that require high physical fidelity, such as those in medical training and visualization. The method addresses a specific set of application requirements where neither traditional, slower FEM nor faster, less accurate constraint-based methods are fully suitable. By decomposing the simulation domain into manageable groups and combining direct solvers with an iterative constraint mechanism, the method balances computational speed and accuracy.

Our quantitative evaluations demonstrate the method's performance in its target application. It achieves interactive frame rates (e.g., 65 ms for a 120,000 tet mesh) while maintaining a high degree of physical accuracy. This level of precision is important for applications like surgical planning. The method is also effective in simulating nearly incompressible materials (2.95% volume error at a 0.49 Poisson ratio) and supports anisotropic materials. The group-based architecture is highly parallelizable and shows effective scaling on multi-core CPUs (e.g., a 19.6 ms frame time for 120,000 tets on 64 threads). This affinity for CPU parallelism stems from the method's structure, which decomposes the problem into a moderate number of computationally significant, independent tasks (one per group), aligning well with the task-parallel architecture of modern multi-core processors. While a GPU implementation could be explored, our approach is well-suited for execution on multi-core CPU systems, which may be readily available in various computing environments.

In summary, the Local Corotated FEM provides a viable option for applications where the cost of inaccuracy is high and real-time interaction is necessary. Its demonstrated performance and features make it an applicable method for developing medical simulators, haptic systems, and digital twins.

VII. LIMITATIONS & FUTURE WORK

While our Local Corotated FEM method demonstrates significant advancements in real-time large deformation simulation, it is not without limitations.

The proposed method inherits certain fundamental limitations from the corotational framework. The corotational approach does not inherently support volume conservation and relies on external constraints or stabilization techniques to maintain volume during deformation. Additionally, our linearized stiffness matrix approximation, while computationally efficient, neglects the rotational components in the Hessian computation [36]. This simplification can lead to reduced accuracy in scenarios involving large rotations or complex deformation modes, where the neglected rotational terms would otherwise contribute significantly to the system's response. The pre-computed constant stiffness matrices exacerbate this limitation, as they cannot adapt to changing material orientations during simulation, potentially causing unrealistic behavior in scenes with substantial rotational motions or complex multi-directional deformations.

The current group division strategy employs coordinate-axis-aligned partitioning for computational simplicity and implementation efficiency. However, this approach may not be optimal for all geometries or loading conditions. Alternative strategies, such as spectral partitioning based on mesh connectivity, force-field-aware divisions, or adaptive clustering based on deformation patterns, could potentially improve accuracy and convergence behavior. The choice of coordinate-axis division, while straightforward to implement, may create artificial boundaries that do not align with the natural stress distribution in the material.

The proposed method is designed for specific application scenarios and may not be suitable for all simulation requirements. The method may fail or exhibit reduced accuracy in cases involving extremely large deformations where the linearization assumptions break down, contact-heavy scenarios where the group-based approach cannot efficiently handle numerous contact constraints, or materials with severe anisotropy where our group-level rotation approximation becomes inadequate. Additionally, the method's performance degrades when the characteristic length scale of deformation features becomes smaller than the typical group size.

The iterative nature of our inter-group constraint solver introduces convergence considerations that affect both accuracy and performance. The current implementation uses a Jacobi-like iteration scheme for inter-group constraints, which, while parallelizable, may converge more slowly than a Gauss-Seidel approach for certain problem configurations.

The Jacobi-like approach was chosen primarily for its parallelization advantages in multi-core CPU environments. Since each group's constraint update can be computed independently using information from the previous iteration, this enables efficient parallel execution across multiple threads. In contrast, Gauss-Seidel requires sequential processing as each group's update depends on the most recent values from previously updated groups, limiting parallel scalability. The choice between these iteration schemes involves a trade-off: Jacobi iterations enable better parallelization but may require more iterations for convergence, while Gauss-Seidel iterations can converge faster but are inherently sequential.

However, the Jacobi-like approach can introduce boundary artifacts when insufficient iterations are used. These manifest as small gaps or discontinuities between adjacent groups, particularly visible in regions experiencing large forces or rapid deformation. The slower convergence characteristic of Jacobi methods means that constraint information propagates more gradually across group boundaries, potentially leaving residual errors that appear as visual artifacts. These effects can be mitigated by increasing the iteration count or adjusting the constraint stiffness parameter κ, though this trades computational efficiency for visual quality. The optimal choice depends on the specific hardware architecture, problem size, and desired accuracy.

A key challenge lies in the fine-tuning of constraint forces between groups. Excessive forces can lead to simulation instability or unrealistic stiffness, while insufficient forces may compromise model integrity or result in excessive softness. Accumulation can decrease static error for constraint accuracy, but dynamic errors might still occur. Future work will focus on developing more robust and adaptive methods for constraint force adjustment, along with guidelines to address dynamic errors effectively. Another area for improvement is the use of Shape Matching and APD for group rotations. While computationally efficient, this approach may not capture highly localized or extreme deformations as accurately as element-wise rotation calculations. Future research aims to explore hybrid approaches that could maintain efficiency while improving accuracy for large rotations. The preliminary work with orthotropic material properties opens up possibilities for simulating a wider range of anisotropic materials. In future research, future work plans to expand the method to accommodate more complex material anisotropies. Additionally, future research intends to investigate the integration of the method with haptic devices, exploring its potential in applications requiring real-time force feedback.

REFERENCES

[1] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, “Position based dynamics,” *J. Vis. Commun. Image Represent.*, vol. 18, no. 2, pp. 109–118, 2007.

[2] J. Bender, D. Koschier, P. Charrier, and D. Weber, “Position-based simulation of continuous materials,” *Comput. Graph.*, vol. 44, pp. 1–10, Nov. 2014.

[3] M. Macklin, M. Müller, and N. Chentanez, “XPBD: Position-based simulation of compliant constrained dynamics,” in *Proc. 9th Int. Conf. Motion Games*, Oct. 2016, pp. 49–54.

[4] S. Bouaziz, S. Martín, T. Liu, L. Kavan, and M. V. Pauly, “Projective dynamics: Fusing constraint projections for fast simulation,” in *Proc. Seminal Graph. Papers, Pushing Boundaries*, 2023, pp. 787–797.

[5] A. H. Chen, Z. Liu, Y. Yang, and C. Yuksel, “Vertex block descent,” *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–16, Jul. 2024.

[6] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, “Stable real-time deformations,” in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2002, pp. 49–54.

[7] M. Müller and M. Groß, “Interactive virtual materials,” in *Proc. Graph. Interface*, 2004, pp. 239–246.

[8] T. Kugelstadt, D. Koschier, and J. Bender, “Fast corotated FEM using operator splitting,” in *Proc. Comput. Graph. Forum*, vol. 37, 2018, pp. 149–160.

[9] F. Hecht, Y. J. Lee, J. R. Shewchuk, and J. F. O’Brien, “Updated sparse Cholesky factors for corotational elastodynamics,” *ACM Trans. Graph.*, vol. 31, no. 5, pp. 1–13, Aug. 2012.

[10] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, “Physically based deformable models in computer graphics,” *Comput. Graph. Forum*, vol. 25, pp. 809–836, Dec. 2006.

[11] C. Deul, P. Charrier, and J. Bender, “Position-based rigid-body dynamics,” *Comput. Animation Virtual Worlds*, vol. 27, no. 2, pp. 103–112, 2014.

[12] M. Macklin and M. Müller, “Position based fluids,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–12, Jul. 2013.

[13] V. Burkus, A. Kárpáti, G. Klár, and L. Szécsi, “Real-time sponge and fluid simulation,” in *Proc. Eurographics*, 2022, pp. 81–84.

[14] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, “Meshless deformations based on shape matching,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 471–478, Jul. 2005.

[15] M. Müller, M. Macklin, N. Chentanez, and S. Jeschke, “Physically based shape matching,” in *Proc. Comput. Graph. Forum*, vol. 41, 2022, pp. 1–7.

[16] J. N. Reddy, *Introduction to the Finite Element Method*. New York, NY, USA: McGraw-Hill, 2019.

[17] J. Georgii and R. Westermann, “Corotated finite elements made fast and stable,” *VRIPHYS*, vol. 8, pp. 11–19, May 2008.

[18] J. Barbič and Y. Zhao, “Real-time large-deformation substructuring,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–8, Jul. 2011.

[19] X. Wu, R. Mukherjee, and H. Wang, “A unified approach for subspace simulation of deformable bodies in multiple domains,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–9, Nov. 2015.

[20] T. Kim and D. L. James, “Physics-based character skinning using multi-domain subspace deformations,” in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, Aug. 2011, pp. 63–72.

[21] J. Huang, X. Liu, H. Bao, B. Guo, and H.-Y. Shum, “An efficient large deformation method using domain decomposition,” *Comput. Graph.*, vol. 30, no. 6, pp. 927–935, Dec. 2006.

[22] B. Wu, Z. Wang, and H. Wang, “A GPU-based multilevel additive schwarz preconditioner for cloth and deformable body simulation,” *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–14, Jul. 2022.

[23] C. Farhat, “A Lagrange multiplier based divide and conquer finite element algorithm,” *Comput. Syst. Eng.*, vol. 2, nos. 2–3, pp. 149–156, 1991.

[24] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen, “FETI-DP: A dual-primal unified feti method—Part I: A faster alternative to the two-level feti method,” *Int. J. Numer. Methods Eng.*, vol. 50, no. 7, pp. 1523–1544, 2001.

[25] J. Cros, “38. A preconditioner for the Schur complement domain decomposition method,” in *Proc. Domain Decomposition Methods Sci. Eng.*, 2003, p. 373.

[26] R. Li, Y. Xi, and Y. Saad, “Schur complement-based domain decomposition preconditioners with low-rank corrections,” *Numer. Linear Algebra Appl.*, vol. 23, no. 4, pp. 706–729, Aug. 2016.

[27] G. Karypis and V. Kumar, “Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices,” Dept. of Comput. Sci., Univ. of Minnesota, Minneapolis, MN, USA, Tech. Rep. TR 97-061, 1997.

[28] P. Sanders and C. Schulz, “Think locally, act globally: Highly balanced graph partitioning,” in *Proc. Int. Symp. Experim. Algorithms*, 2013, pp. 164–175.

[29] O. Etzmuss, M. Keckeisen, and W. Strasser, “A fast finite element solution for cloth modelling,” in *Proc. 11th Pacific Conf. Comput. Graph. Appl.*, 2003, pp. 244–251.

[30] V. Burkus, A. Kárpáti, G. Klár, and L. Szécsi, “Real-time sponge and fluid simulation,” in *Proc. Eurographics*, N. Pelechano and D. Vanderhaeghe, Eds., Geneva, Switzerland: The Eurographics Association, 2022, pp. 81–84, doi: 10.2312/egs.20221038.

[31] J. Barbic, “Exact corotational linear fem stiffness matrix,” USC, Los Angeles, CA, USA, Tech. Rep., 2012.

[32] Y. Voet, E. Sande, and A. Buffa, “A mathematical theory for mass lumping and its generalization with applications to isogeometric analysis,” *Comput. Methods Appl. Mech. Eng.*, vol. 410, May 2023, Art. no. 116033.

[33] M. Liu and D. G. Gorman, "Formulation of Rayleigh damping and its extensions," *Comput. Struct.*, vol. 57, no. 2, pp. 277–285, Oct. 1995.

[34] X. Chen, J. Duan, and Y. Li, "Mass proportional damping in nonlinear time-history analysis," in *Proc. 3rd Int. Conf. Mater., Mech. Manuf. Eng.*, 2015, pp. 567–571.

[35] Ý. Y. Zhuang and J. Canny, "Real-time simulation of physically realistic global deformation," Dept. Comput. Sci. Division, Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/CSD-04-1323, 2004.

[36] T. Kim and D. Eberle, "Dynamic deformables: Implementation and production practicalities," in *Proc. ACM SIGGRAPH Courses*, Aug. 2020, pp. 1–182.

SIYU WANG received the M.S. degree in mechanical engineering from the University of California, San Diego, in 2018. She is currently pursuing the Ph.D. degree with the Department of Information and Communication Engineering, Institute of Science Tokyo, Yokohama, Japan. Her research interests include soft body simulations using finite element methods and virtual reality.

YUNXIU XU (Graduate Student Member, IEEE) received the B.S. degree from Beijing Information Science and Technology University, Beijing, China, and the M.S. degree in engineering from the Department of Information and Communication Engineering, Institute of Science Tokyo, Yokohama, Japan, in 2022, where he is currently pursuing the Ph.D. degree. His research interests include haptic and interaction in virtual reality.

SHOICHI HASEGAWA received the D.Eng. degree in computational intelligence and systems from the Institute of Science Tokyo, Tokyo, Japan. Since 2010, he has been an Associate Professor with the Institute of Science Tokyo. He was an Associate Professor with The University of Electro-Communications, Tokyo. His research interests include haptic renderings, real-time simulations, interactive characters, soft and entertainment robotics, and virtual reality.