A Convex Formulation of Material Points and Rigid Bodies with GPU-Accelerated Async-Coupling for Interactive Simulation

Chang Yu*[1], Wenxin Du*[1], Zeshun Zong[1], Alejandro Castro[2], Chenfanfu Jiang[1], Xuchen Han[2]

I. INTRODUCTION

With the rise of robotics foundation models [1], simulation has become an indispensable tool for both policy training and evaluation [2, 3]. Simulations enable generating large-scale training data at a fraction of the cost of real-world collection and facilitate evaluating model checkpoints in controlled environments, ensuring repeatability. In these settings, modern physics simulators for robotics must satisfy several key requirements to be effective: (i) **strong numerical stability guarantees**, (ii) **sufficient accuracy to capture real-world success and failure modes**, (iii) **computational efficiency to support data collection and policy evaluation at scale**, and (iv) **the ability to model diverse environments**.

Among existing physics simulators for robotics, Drake [4] and MuJoCo [5] are well-regarded for their strong robustness guarantees and high accuracy required for manipulation tasks. Both employ a convex formulation of frictional contact, ensuring stability and global convergence [6, 7]. However, despite recent advancements in deformable body simulation within the convex framework [8–10], these simulators struggle to efficiently simulate deformable bodies with a large number of degrees of freedom (DoFs) at interactive rates. This limitation arises partly from the need to solve poorly-conditioned convex optimization problems that necessitates direct linear solvers with $O(n^3)$ complexity [7]. The inclusion of deformable bodies exacerbates this challenge, as it significantly increases the number of DoFs—often by orders of magnitude—making efficient solutions increasingly difficult. Moreover, the inherently serial nature of direct linear solvers precludes efficient parallelization on GPUs, further restricting scalability.

Among various methods for simulating deformable materials, the **Material Point Method (MPM)** has gained traction due to its ability to handle large deformations and topology changes [11]. However, existing simulators struggle to achieve both robustness and efficiency [10]. In this paper, we propose a novel convex formulation for coupling MPM with rigid bodies through frictional contact, designed for efficient GPU parallelization. We prove the stability and global convergence of our frictional contact model. Furthermore, we demonstrate that our method efficiently simulates a wide range of materials while preserving the high accuracy required for robotic manipulation tasks. We make our method available in the **open-source robotics toolkit, Drake**.

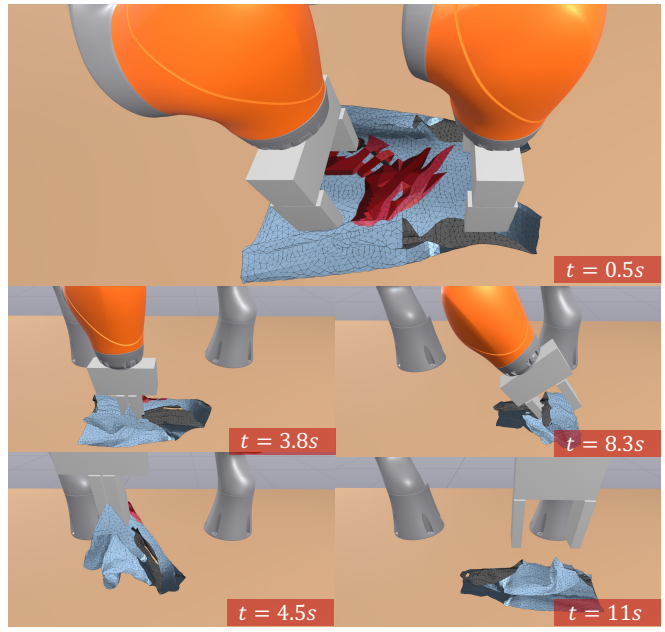

Fig. 1: A challenging T-shirt folding task demonstrating the accuracy, speed, and robustness of our method.

II. MATHEMATICAL FORMULATION

A. Asynchronous Time-Splitting Scheme

We discretize time into intervals of size Δt to advance the system dynamics from t_n to the next time step $t_{n+1} := t_n + \Delta t$ using the governing equation:

$$\mathbf{M}(\mathbf{q})(\mathbf{v}^{n+1} - \mathbf{v}^n) = \Delta t\, \mathbf{k}(\mathbf{q}, \mathbf{v}) + \hat{\mathbf{J}}^T(\mathbf{q})\hat{\boldsymbol{\gamma}}(\mathbf{q}, \mathbf{v}), \quad (1)$$

where $\mathbf{q}$ and $\mathbf{v}$ are generalized positions and velocities, $\boldsymbol{\gamma}$ is the contact and friction impulses, $\mathbf{k}$ represents all non-contact forces, and $\hat{\mathbf{J}}$ is the contact Jacobian. Using subscripts d for MPM DoFs and r for rigid body DoFs, we define:

$$\begin{aligned} \mathbf{q}^T &= [\mathbf{q}_r^T,\ \mathbf{q}_d^T], & \mathbf{v}^T &= [\mathbf{v}_r^T,\ \mathbf{v}_d^T], \\ \mathbf{k}^T &= [\mathbf{k}_r^T,\ \mathbf{k}_d^T], & \mathbf{M} &= \mathrm{diag}(\mathbf{M}_r,\ \mathbf{M}_d). \end{aligned} \quad (2)$$

Specifically, $\mathbf{q}_d$ and $\mathbf{v}_d$ denote the MPM *particle* positions and *grid* velocities ; $\mathbf{k}_d$ includes elastic-plastic and external forces on MPM DoFs; $\mathbf{q}_r$ and $\mathbf{v}_r$ are the articulated rigid body joint positions and velocities; $\mathbf{k}_r$ captures Coriolis terms and gravity on rigid body DoFs.

* equal contribution.

[1]{changyu1,wenxindu,zeshunzong,cffjiang}@ucla.edu, AIVC Laboratory, UCLA, USA.
[2]{alejandro.castro,xuchen.han}@tri.global, Toyota Research Institute, USA.

We further separate contact between MPM and rigid bodies from contact among rigid bodies as:

$$\hat{\gamma} = \begin{pmatrix} \gamma_d \\ \gamma_r \end{pmatrix}, \quad \hat{\mathbf{J}} = \begin{pmatrix} \mathbf{J}_r & \mathbf{J}_d \\ \mathbf{J}_{rr} & \mathbf{0} \end{pmatrix}.$$

Notably, we do not explicitly model contact forces among MPM DoFs, as these are naturally resolved through the hybrid Eulerian/Lagrangian representation with constitutive and plasticity modeling [12, 13].

To achieve efficient GPU parallelization without sacrificing stability, we integrate elastic forces explicitly using symplectic Euler, while treating stiff frictional contact forces implicitly via backward Euler. However, symplectic Euler requires smaller time step sizes than backward Euler for stability, motivating our asynchronous time-splitting approach:

$$\mathbf{M}_r(\mathbf{q}_r^n)(\mathbf{v}_r^{n+1} - \mathbf{v}_r^n) = \Delta t\, \mathbf{k}_r(\mathbf{q}_r^n, \mathbf{v}_r^n) + \mathbf{J}_{rr}^T \boldsymbol{\gamma}_r(\mathbf{v}_r^{n+1}) + \mathbf{J}_r^T \sum_{k=0}^{N-1} \boldsymbol{\gamma}_d^{n,k}, \tag{3}$$

$$\mathbf{M}_d(\mathbf{q}_d^{n,k})(\mathbf{v}_d^{n,k+1} - \mathbf{v}_d^{n,k}) = \frac{\Delta t}{N}\, \mathbf{k}_d(\mathbf{q}_d^n) + \mathbf{J}_d^T \boldsymbol{\gamma}_d^{n,k}, \tag{4}$$

for $k = 0, \ldots, N-1$, with $\mathbf{v}_d^{n,0} = \mathbf{v}_d^n$, and $\mathbf{v}_d^{n,N} = \mathbf{v}_d^{n+1}$.

Here, $\boldsymbol{\gamma}_d^{n,k} = \boldsymbol{\gamma}_d(\mathbf{v}_r^n, \mathbf{v}_d^{n,k+1})$ is the frictional impulse between rigid bodies and MPM at substep k. Notice that in Eq. (4), we decompose a single time step into N substeps. Elastic forces $\mathbf{k}_d$ are integrated explicitly to facilitate parallelization, whereas contact forces $\boldsymbol{\gamma}_d$ are integrated implicitly with respect to $\mathbf{v}_d^{n,k+1}$ for stability. Throughout all substeps in a single timestep, the rigid body velocity is fixed at $\mathbf{v}_r^n$. The contact impulses at each substep are accumulated and applied to rigid body DoFs in Eq. (3). This approach results in a weak coupling between rigid bodies and MPM DoFs, in contrast to the strong coupling scheme proposed by [10]. We evaluate the accuracy of our weak coupling scheme in Section III.

B. Convex Formulation

The integration of rigid DoFs in Eq. (3) follows the same methodology as described in [14]. We refer readers to that work for detailed explanations and implementation practices. Here, we focus on the integration of Eq. (4) for MPM DoFs. For notational simplicity, we drop the subscript d and the superscript n in Eq. (4), reducing it to an algebraic difference equation that advances a substep of MPM:

$$\mathbf{M}(\mathbf{v}^{k+1} - \mathbf{v}^k) = \frac{\Delta t}{N}\mathbf{k} + \mathbf{J}^T \boldsymbol{\gamma}(\mathbf{v}_c). \tag{5}$$

Here, $\mathbf{v}_c = \mathbf{J}\mathbf{v}^{k+1} + \mathbf{b}_r$ denotes the relative contact velocity between particles and the rigid body they are in contact with, expressed in the contact frame, where the contact frame is a local frame with the z-axis aligned with the contact normal. The term $\mathbf{b}_r = \mathbf{J}_r \mathbf{v}_r^n$ represents the bias velocity from the rigid body in the contact frame.

We solve Eq. (5) in two stages by performing another time-splitting. First, we compute the *free motion velocity*, $\mathbf{v}^*$, which is the velocity the MPM DoFs would attain in the absence of contact forces:

$$\mathbf{M}(\mathbf{v}^* - \mathbf{v}^k) = \frac{\Delta t}{N}\mathbf{k}. \tag{6}$$

This step is equivalent to a standard explicit MPM step. We adopt the moving least-squares formulation described in [15] for this computation.

In the second stage, we compute the *post-contact velocity* $\mathbf{v}^{k+1}$ by solving:

$$\mathbf{M}(\mathbf{v}^{k+1} - \mathbf{v}^*) = \mathbf{J}^T \boldsymbol{\gamma}(\mathbf{v}_c). \tag{7}$$

To ensure global convergence, we reformulate Eq. (7) as a convex optimization problem [14]:

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \ell_p = \arg\min_{\mathbf{v}} \frac{1}{2}\|\mathbf{v} - \mathbf{v}^*\|_{\mathbf{M}}^2 + \ell_c(\mathbf{v}_c). \tag{8}$$

The term $\ell_c(\mathbf{v}_c)$ is the contact potential energy, defined such that $\boldsymbol{\gamma}(\mathbf{v}_c) = -\partial \ell_c / \partial \mathbf{v}_c$. With mass lumping, $\mathbf{M}$ is diagonal and positive definite, ensuring that problem (8) is strongly convex as long as the frictional contact model is designed with a convex ℓ_c. We adopt the *lagged contact model* in [14].

C. Quasi-Newton Solver

A key advantage of our weak coupling scheme is that, instead of requiring the Schur complement of the Jacobian of the *momentum residual* as in [10]—which is unfriendly to GPU parallelization—our formulation replaces it with the diagonal mass matrix, which is trivial to parallelize. However, the Hessian $\hat{\mathbf{H}}$ of ℓ_p still contains off-diagonal entries from the Hessian of ℓ_c. Given the massive parallelization capabilities of GPUs, it is more efficient to trade additional solver iterations for better parallelization (see Section III-A). Therefore, we adopt a quasi-Newton strategy by approximating $\hat{\mathbf{H}}$ with its 3×3 block diagonal counterpart $\mathbf{H}$ to avoid the inherently serial Cholesky factorization of $\hat{\mathbf{H}}$. We leverage the convergence criteria from [7] and [10] to ensure robustness against large mass ratios and enable fair comparison in Section III-A. The resulting $\boldsymbol{\gamma}$ is accumulated into Eq. (3).

III. RESULTS

We present several test cases to showcase the efficiency, accuracy, and robustness of our method. All simulations are run on a system with an Intel(R) Xeon(R) CPU E5-2690 v4 processor (56-core) and 128 GB of RAM, and an RTX 4090 with 24 GB device memory. All simulations are solved to convergence with $\varepsilon_r = 5 \times 10^{-2}$ and ε_a set to machine epsilon unless otherwise specified.

A. Rolling an Elastoplastic Dough

We reproduce the complex dough rolling example from [10] with our asynchronous time-splitting scheme, as shown in Fig. 2. The friction coefficient is 1.0 between the dough and the rolling pin.

We compare the runtime performance of our method with [10] under various tolerance criteria in Fig. 2. For both methods, we set the absolute tolerance ε_a to machine epsilon

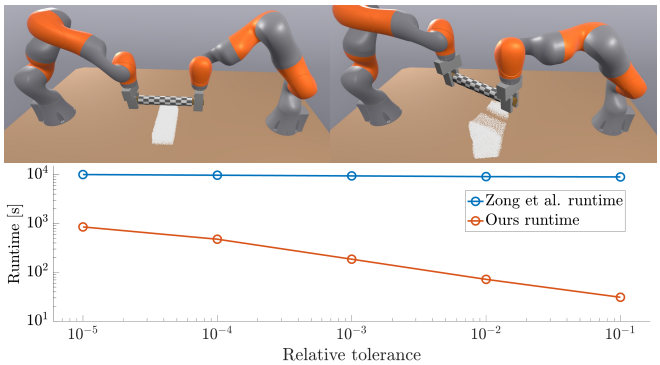

Fig. 2: (**Top**) Flattening a piece of dough with a rolling pin. Our coupling scheme captures the rolling pin's friction-driven rotation and the dough's deformation. (**Bottom**) We compare the runtime between ours and Zong et al. [10] under various relative tolerances.

and vary the relative tolerance ε_r from 10^{-5} to 10^{-1}. Both methods are simulated with a time step size $\Delta t = 10$ ms, and our method uses $N = 10$ substeps. Across all tolerance levels, our method consistently outperforms [10], achieving at least a 10× speed-up and demonstrating strong scalability as the convergence criteria are relaxed, making it well-suited for interactive-rate simulations. With $\varepsilon_r = 10^{-1}$, our method attains a 500× speed-up compared to [10], reaching a 59% real-time rate (defined as simulation time divided by wall-clock time).

B. Cloth and rigid bodies

Another key advantage of our weak coupling scheme is that it eliminates the requirement for a convex energy density in the constitutive model of MPM, as mandated by [10] to maintain convexity in the optimization problem (8). This flexibility unlocks the potential of MPM to model a broader range of materials.

In this experiment, we implement the method from [16] to model cloth with MPM. The frictional contact and self-collision of the cloth are automatically handled by the MPM grid. The simulation setup involves a 42 cm^2 cloth modeled with a Young's modulus of $E_{\text{cloth}} = 3.2 \times 10^6$ Pa, a Poisson's ratio of $\nu = 0.4$, and a density of $\rho_{\text{cloth}} = 1.5 \times 10^3$ kg/m^3. The four corners of the cloth are fixed with boundary conditions to form a cradle. A KUKA LBR iiwa 7 arm, equipped with an anthropomorphic Allegro hand, picks up two rigid boxes and places them on the cloth. Each box has a side length of 8 cm and a density of $\rho_{\text{box}} = 10^3$ kg/m³. The friction coefficient is 0.2 between the boxes and the cloth. The cloth corners are then moved inward to wrap around the rigid boxes, and finally, two corners are released, allowing the boxes to fall out.

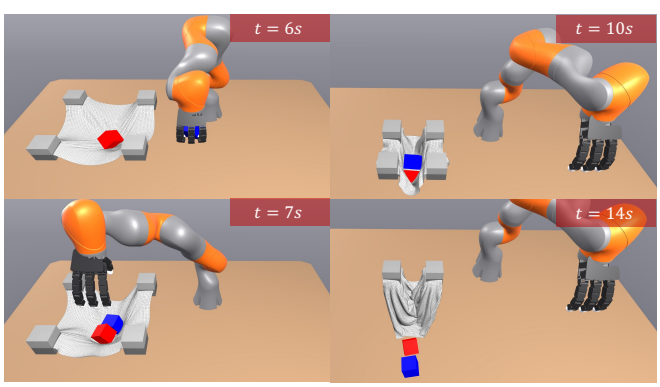

Fig. 3: Pick and place rigid boxes into a deformable cradle.

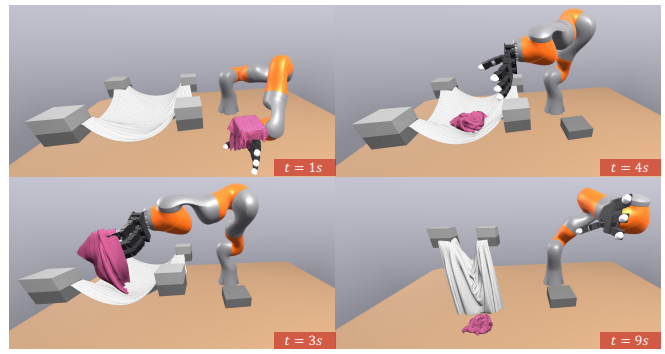

Fig. 4: A robot picks up a piece of cloth and throws it in a "laundry bag" . Our method resolves the complex cloth-on-cloth collisions and cloth self-collisions at interactive rate.

C. Laundry

Similar to the setup in Section III-B, we replace the two boxes with another piece of cloth to mimic a common household scenario of moving clothes into a laundry bag. The robot follows a prescribed trajectory to grasp the cloth and throw it into the laundry bag. We then release the grip on two corners of the laundry bag, allowing the cloth to fall out naturally. The simulation achieves a real-time rate of 26.5%, demonstrating that our method efficiently handles intense cloth-on-cloth collisions and self-collision scenarios.

D. Folding and Unfolding a T-Shirt

We demonstrate the accuracy and robustness of our method with a challenging T-shirt folding task (Fig. 1). The T-shirt mesh consists of 4,171 vertices and 7,987 faces, and is simulated with a Young's modulus of $E_{\text{cloth}} = 10^5$ Pa, a Poisson's ratio of $\nu = 0.3$, and a density of $\rho_{\text{cloth}} = 10^3$ kg/m³. The robot setup features two PD-controlled KUKA LBR iiwa 7 arms equipped with custom parallel grippers.

Our method accurately captures frictional interactions between the gripper and the T-shirt, enabling smooth task execution. The T-shirt undergoes two folds, resulting in up to eight stacked layers. Our approach effectively handles cloth self-collisions and rigid-MPM interactions, preventing cloth self-penetration throughout the simulation. This ensures the T-shirt is successfully unfolded without artifacts, showing the robustness of our method.

We refer readers to the supplemental video for the full trajectories of all experiments.

References

[1] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, p. 02783649241281508, 2023. 1

[2] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve *et al.*, "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," *Proceedings of the National Academy of Sciences*, vol. 118, no. 1, p. e1907856118, 2021. 1

[3] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang *et al.*, "Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks," *arXiv preprint arXiv:2412.18194*, 2024. 1

[4] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," https://drake.mit.edu, 2019. 1

[5] E. Todorov, "MuJoCo," http://www.mujoco.org. 1

[6] ——, "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6054–6061. 1

[7] A. M. Castro, F. N. Permenter, and X. Han, "An unconstrained convex formulation of compliant contact," *IEEE Transactions on Robotics*, 2022. 1, 2

[8] K. Bayes, M. Bennice, T. Erez, E. Frey, C. K. Fu, N. Gileadi, A. Quaglino, B. Tabanpour, Y. Tassa, and S. Tunyasuvunakool, "Mujoco 3," 2023. [Online]. Available: https://github.com/google-deepmind/mujoco 1

[9] X. Han, J. Masterjohn, and A. Castro, "A convex formulation of frictional contact between rigid and deformable bodies," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6219–6226, 2023.

[10] Z. Zong, C. Jiang, and X. Han, "A convex formulation of frictional contact for the material point method and rigid bodies," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1831–1838. 1, 2, 3

[11] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023. 1

[12] X. Han, T. F. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran, "A hybrid material point method for frictional contact with diverse materials," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 2, no. 2, pp. 1–24, 2019. 2

[13] C. Jiang, T. Gast, and J. Teran, "Anisotropic elastoplasticity for cloth, knit and hair frictional contact," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–14, 2017. 2

[14] A. Castro, X. Han, and J. Masterjohn, "A theory of irrotational contact fields," *arXiv preprint arXiv:2312.03908*, 2023. 2

[15] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, "A moving least squares material point method with displacement discontinuity and two-way rigid body coupling," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018. 2

[16] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The affine particle-in-cell method," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–10, 2015. 3