GPU-Accelerated Hydroelastic Contact via Signed Distance Fields

Lennart Röstel[1,2,*], Yang Liu[3], Jessica Yin[1], Miguel Zamora[1],
Miles Macklin[1], Philipp Reist[1], Tobias Widmer[1]

***Abstract*— Accurately and efficiently simulating frictional contact is a key challenge in robotic manipulation. Commonly used point-contact methods are efficient but lack area-dependent effects and can introduce artifacts. The hydroelastic contact model [1] improves upon this by spatially resolving the full contact surface; however, current approaches are computationally expensive, especially for non-convex, high-resolution geometries. Addressing these challenges, we present a novel, GPU-accelerated formulation of hydroelastic contact based entirely on Signed Distance Fields (SDFs). We extract the isopressure surface between colliding objects via marching cubes on the stiffness-weighted difference of SDFs, while a hierarchical pruning strategy identifies isosurface-containing grid cells. For efficient integration into commonly used solvers, we also devise a contact reduction scheme that produces a small set of contact constraints capturing net wrenches from the original contact patch. Our end-to-end GPU-accelerated implementation is part of the Newton project[1], enabling simulation of complex, geometry-dependent contact phenomena with high fidelity and computational efficiency.**

I. Introduction

Closing the sim-to-real gap in robotic manipulation hinges on the ability to reproduce realistic contact behavior in simulation. Among recent efforts to improve contact modeling fidelity, the hydroelastic contact model proposed by Elandt et al. [1] offers a compelling compromise between fidelity and efficiency for simulating contact between rigid objects. Rather than resolving contacts at isolated contact points, the hydroelastic model defines a pressure field over the interior of each body. When two bodies interpenetrate, the model then produces a distributed force estimate by calculating the surface inside the overlap volume where the two pressure fields are equal. This formulation offers two notable advantages: contact forces vary continuously with respect to changes in contact geometry and the resulting net wrenches reflect the extended spatial structure of the contact surface. Masterjohn et al. [2] extended this model to a velocity-level time-stepping formulation with polygonal tessellation of the contact surface, simplifying its integration into solvers commonly used in robotics such as in Drake [3].

However, the current implementation in Drake carries practical limitations. The pressure field is represented on tetrahedral volume meshes, which are non-trivial to generate for arbitrary geometries — Drake can automatically generate them only for convex primitives, while non-convex objects require user-supplied VTK files. Moreover, detecting intersecting tetrahedra pairs and computing their intersection planes becomes computationally expensive as mesh resolution increases, which can limit scalability for detailed geometries and contact-rich manipulation tasks.

More fundamentally, although hydroelastic contact requires a volumetric representation, tetrahedral meshes are not the only option. Signed distance fields offer a volumetric description that is typically easier to generate from arbitrary geometries and is well suited to regular-grid, GPU-parallel computation. For point-contact-based approaches, SDF-based representations have recently gained traction: Macklin et al. [4] proposed a method for robust collision detection between rigid bodies by local optimization over surface triangle meshes against SDFs. Narang et al. [5] extend this approach by introducing contact-reduction and leveraging GPU-acceleration for high-throughput contact simulation in robotic assembly tasks. MuJoCo [6], [7] recently introduced native SDF-based contacts that compute contact points via gradient descent on the SDF intersection.

However, these prior SDF-based contact approaches do not aim at recovering the full distributed contact surface and thus lack area-dependent effects. Our method combines the geometric generality of SDF representations with the physically grounded distributed-force model of hydroelastic contact.

II. Method

We present a novel approach to hydroelastic contact computation that is purely based on signed distance fields (SDFs). In short, we propose to

1) Represent each object with a grid-sampled SDF, which can be robustly and efficiently generated for arbitrary geometries.
2) Construct a triangle representation of the isopressure surface between pairs of colliding objects via marching cubes [8] on the stiffness-weighted difference of SDFs.
3) Discretize and reduce the contact surface to a compact set of contact points suitable for downstream solvers, while preserving instantaneous net forces and frictional moment capacity from the full contact patch.

For a visual overview of the approach, also see figure 1. We now describe each stage in more detail.

A. Isosurface Generation

We represent each object's geometry by an SDF $\phi(\mathbf{x})$ sampled on a regular three-dimensional grid of axis-aligned cubic cells (voxels). The SDF gives the signed distance from

[1]NVIDIA; [2]TUM, Germany; [3]EPFL, Switzerland.
[*]Work done during an internship at NVIDIA.
[1]`github.com/newton-physics/newton`

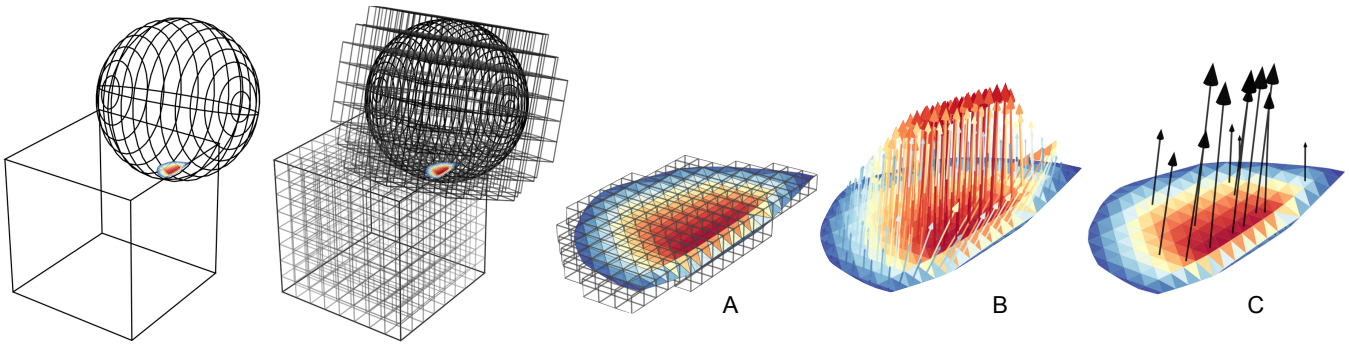

Fig. 1. Stages of the proposed SDF-based hydroelastic contact generation pipeline. **Left:** The collision between two objects is described by a contact surface within their volume of intersection. We represent each shape as a (narrow-band) grid-sampled SDF in the frame of the object. **A)** The contact surface is obtained by applying marching cubes to the difference of stiffness-weighted SDFs, iterating over the voxels which contain the isosurface. **B)** Each triangle of the resulting isosurface defines a force contribution or an equivalent point-contact constraint. **C)** The set of contact points is reduced to a smaller number of representative point contacts (here $274 \rightarrow 17$), suitable for efficient processing by downstream solvers.

a point $\mathbf{x} \in \mathbb{R}^3$ to the object's surface, with $\phi(\mathbf{x}) < 0$ inside the object and $\phi(\mathbf{x}) > 0$ outside.

For two colliding objects A and B with SDFs $\phi_A(\mathbf{x})$ and $\phi_B(\mathbf{x})$ and hydroelastic moduli k_A and k_B , respectively, we define the scalar field of stiffness-weighted difference

$$g(\mathbf{x}) = k_A\,\phi_A(\mathbf{x}) - k_B\,\phi_B(\mathbf{x}). \tag{1}$$

In the hydroelastic contact model, finding the zero-level set of g is equivalent to computing the isopressure surface for pressure fields that increase linearly with penetration depth from each object's surface.

Thus, the hydroelastic *contact surface* $\mathcal{S}$ is defined as

$$\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^3 \mid g(\mathbf{x}) = 0, \phi_A(\mathbf{x}) < 0, \phi_B(\mathbf{x}) < 0\right\}. \tag{2}$$

Since $\mathcal{S}$ is defined as an isosurface, it can be naturally tessellated into a triangle mesh using marching cubes.

Because the two colliding objects may have SDF grids with different resolutions, we use the finer of the two as the marching-cubes grid. On this grid, we sample g at voxel corners, query the coarser SDF by trilinear interpolation where needed, and apply marching cubes to extract a triangulated approximation of the contact surface.

As naively running marching cubes on *all* voxels of the grid can become expensive in dense 3D grids, we first filter for the voxels that contain the isosurface (*isovoxels*) in a broad/midphase:

1) We detect overlapping bounding boxes (given by SDF extents) of potentially colliding objects A and B.
2) We divide the SDF into *blocks* of $8 \times 8 \times 8$ voxels. We evaluate $g(\mathbf{x}_T)$ at the center of the block and compare it to its stiffness-corrected bounding radius $R_T = r(k_A + k_B)$ where r is the blocks geometric half-diagonal. If $|g(\mathbf{x}_T)| > R_T$, we prune all voxels in the block, as we know they cannot contain the isosurface.
3) We repeat step 2) recursively for subblocks of sizes $4 \times 4 \times 4$, $2 \times 2 \times 2$ and $1 \times 1 \times 1$.

We then apply marching cubes only to the remaining isovoxels to extract the triangulated isosurface. While $g(\mathbf{x})$ is valid outside the overlap volume as well, the actual contact surface is obtained by keeping only the triangles whose vertices satisfy $\phi_A(\mathbf{x}) < 0$ and $\phi_B(\mathbf{x}) < 0$.

The result is a triangle-mesh approximation of the contact surface $\mathcal{S}$, defined by a set of vertices $\mathbf{v}_j$ and triangular faces, along with the pressure value $p(\mathbf{v}_j) = k_A\,\phi_A(\mathbf{v}_j) = k_B\,\phi_B(\mathbf{v}_j)$ at each vertex.

The computation is well suited to GPU parallelization: voxels can be processed independently, and evaluating g requires only local SDF queries and trilinear interpolation. In our implementation, we further accelerate computations and reduce memory footprint using sparse narrow-band SDFs.

B. Contact Discretization

While the tessellated isopressure surface from section II-A could be directly used to derive contact forces for use in ODE-based solvers [1], integration into efficient, commonly used velocity-level solvers requires discretization of the contact surface into point contacts. Following [2], we place a contact constraint $(\mathbf{x}_c, \hat{\mathbf{n}}, \phi_0, k_{\text{eff}})$ at the centroid $\mathbf{x}_c$ of each triangle, with contact normal $\hat{\mathbf{n}}$ given by the triangle face, a surrogate penetration depth

$$\phi_0 = \phi_A(\mathbf{x}_c) + \phi_B(\mathbf{x}_c), \tag{3}$$

and an effective stiffness that accounts for the triangle area a as

$$k_{\text{eff}} = \frac{k_A\,k_B}{k_A + k_B}\,a. \tag{4}$$

The equilibrium normal force $f_{n,i} = k_{\text{eff}}|\phi_{0,i}|$ represented by each point contact is exactly the integral of the linear pressure field over its triangle area, since centroid evaluation is exact for linear functions. The discretized contacts also retain the smooth dependence of hydroelastic contact forces on relative configuration, since under marching cubes triangles appear and disappear at zero area.

For speculative contact generation it is useful to additionally retain non-penetrating contacts within a positive margin d_{gap}, i.e., where $\phi_A(\mathbf{x}) < d_{\text{gap}}$ and $\phi_B(\mathbf{x}) < d_{\text{gap}}$.

C. Contact Reduction and Wrench Matching

In principle, the set of contact points from section II-B (one per isosurface triangle) can be passed directly to any downstream solver for implicit handling of the compliant contact constraints. This is feasible when contact constraints are resolved in parallel on a GPU, as in the Jacobi-style XPBD solver [9] used in Newton [10]. However, most widely used solvers, such as MuJoCo [7], scale superlinearly in the number of contact constraints, making it impractical to retain all generated contacts. To address this, we introduce a *contact reduction* step that compresses the full contact set into a compact set of *representative* contacts while preserving the instantaneous (equilibrium) net wrenches across the original contact patch.

Given the set of raw contacts between a colliding pair AB, we reduce them to a compact representative set as follows:

1) **Normal binning.** We define N_b bin normals $\hat{\mathbf{n}}_b$ uniformly distributed on the unit sphere via icosahedral subdivision. Each contact i with normal $\hat{\mathbf{n}}_i$ is assigned to the bin $b^* = \arg\max_b \langle \hat{\mathbf{n}}_i, \hat{\mathbf{n}}_b \rangle$.
2) **Per-bin aggregation.** For each bin b, we compute the aggregate surrogate force

$$\mathbf{F}_b = \sum_{i \in b} f_{n,i}\, \hat{\mathbf{n}}_i, \tag{5}$$

the pressure-weighted centroid $\bar{\mathbf{x}}_b = \sum_{i \in b} p_i \mathbf{x}_i \,/\, \sum_{i \in b} p_i$, and the friction-induced maximum resistive moment about the bin normal

$$M_b = \sum_{i \in b} \mu\, f_{n,i}\, |\mathbf{r}_i \times \hat{\mathbf{n}}_i|, \tag{6}$$

where $\mathbf{r}_i = \mathbf{x}_i - \bar{\mathbf{x}}_b$, along with the second-moment statistics of the contact distribution within the bin.
3) **Spatial selection.** Within each bin b, we construct N_u direction vectors $\mathbf{u}_j$ uniformly spaced in the tangent plane of $\hat{\mathbf{n}}_b$. For each direction, we retain the contact maximizing a heuristic scoring function measuring the projected displacement from the patch centroid, weighted by penetration depth

$$i_j^* = \arg\max_{i \in b} \left(\langle \mathbf{x}_i - \bar{\mathbf{x}}_b,\, \mathbf{u}_j \rangle |\phi_{0,i}| \right). \tag{7}$$

This ensures that the selected contacts match the spatial extents of the contact patch for stable support while also contributing meaningfully to wrenches. Additionally, to guarantee that the regions of maximum pressure are consistently captured, we select the deepest contacts inside each voxel of a coarse grid on the object's bounding box.
Notably, the selection operates independently on each contact in the bin and does not assume convexity or even connectivity of the contact patch, making it applicable to complex contact geometries.
4) **Wrench matching.** We adjust the selected contacts so that their aggregate surrogate force matches the original bin force $\mathbf{F}_b$. Since contacts within a bin share similar normals and the force model is linear in k_{eff}, this admits a closed-form solution in two steps. First, all selected contacts in the bin are assigned a uniform stiffness chosen so that the total force magnitude is preserved: $k_{\text{eff}}^b = \|\mathbf{F}_b\| \,/\, \sum_{j \in \text{selected}} |\phi_{0,j}|$. Second, the normals of the selected contacts are uniformly rotated such that the direction of their aggregate force aligns with $\mathbf{F}_b$:

$$\hat{\mathbf{n}}_j \leftarrow \mathbf{R}_b\, \hat{\mathbf{n}}_j, \qquad \mathbf{R}_b = \text{rot}\left(\frac{\mathbf{F}_{\text{sel}}}{\|\mathbf{F}_{\text{sel}}\|} \rightarrow \frac{\mathbf{F}_b}{\|\mathbf{F}_b\|} \right), \tag{8}$$

where $\mathbf{F}_{\text{sel}} = \sum_{j \in \text{selected}} k_{\text{eff}}^b\, \phi_{0,j}\, \hat{\mathbf{n}}_j$ is the aggregate force of the selected contacts after stiffness reassignment.
Additionally, to preserve the torsional friction capacity of the original patch, the friction coefficients μ_j of the selected contacts are scaled non-uniformly based on their distance to the pressure-weighted centroid $\bar{\mathbf{x}}_b$, such that the maximum friction-induced moment about the bin normal approximately matches M_b (*Moment Matching*). The non-uniform scaling allows simultaneously preserving the net tangential friction capacity of the original patch.

III. Simulation Results

We demonstrate the performance of our GPU-based implementation in Warp [11] on three contact-intensive simulation scenes (Figure 2), showing that the SDF-based representation can efficiently handle detailed, highly nonconvex geometries. Table I reports the accumulated speedup over real time and the number of generated contacts. To isolate the effect of contact reduction, we report nut-and-bolt metrics both with and without reduction enabled: with the selection step the contact count decreases by more than $200\times$, yielding a substantial increase in simulation speed (benchmarked on an NVIDIA RTX 5090).

Fig. 2. Simulation scenes used for benchmarking. **Left:** an M20 nut-and-bolt assembly [5] **Center:** a pile of 91 colliding nonconvex bunny meshes, **Right:** a Franka Panda robot grasping a pen and placing it into a cup. For the nut-and-bolt and Panda scenes, we use MjWarp [12] and simulate 128 and 512 independent instances in parallel, respectively; for the Bunny Pile, we use Newton's XPBD.

TABLE I
CONTACT SIMULATION METRICS

Scene		# Contacts	Acc. Realtime Factor
nut-and-bolt	reduced	18k	141×
	unreduced	4M	18×
bunny pile	reduced	14k	5.0×
Panda-pen	reduced	57k	104×

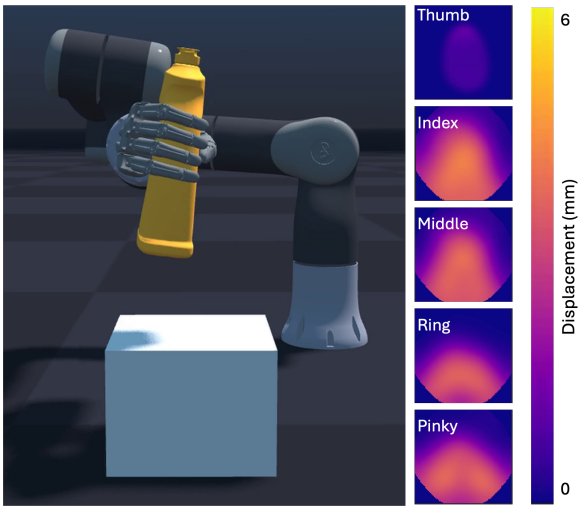

Fig. 3. Simulated visuotactile sensing on the Sharpa Wave hand grasping a bottle, with the resulting per-pixel displacement maps.

IV. Application: Tactile Sensor Simulation

The isosurface extracted in Section II-A provides a spatially resolved contact surface that can be directly queried for tactile simulation—an application where existing fast simulators are limited by their underlying point-contact models [13]–[15]. Inspired by [16], we implement a visuotactile sensor model for the Sharpa Wave hand by raycasting from each taxel's outer surface inward along its local normal against the full (unreduced) hydroelastic contact surface.

The intersection depth yields a per-pixel elastomer displacement map (Fig. 3) that mirrors the real sensor's output. Because the hydroelastic model resolves the spatial distribution of pressure and area-dependent friction effects, the resulting tactile signals capture richer contact modes than point-contact-based alternatives.

V. Qualitative Verification of Area-Resolved Contact Dynamics

In most simulators, area-dependent frictional effects must be specified explicitly — e.g., MuJoCo's [7] torsional friction is a per-geom parameter unrelated to contact geometry. The hydroelastic model instead *resolves* contact area from the SDF, so torsional capacity emerges from the spatial extent of the reconstructed isopressure surface. We test whether this geometric resolution yields physically plausible dynamics in a pen-dropping scenario (Fig. 4 (Left)).

A parallel-jaw gripper grasps a bar away from its center of mass. During gradual release, the resulting gravity torque causes the bar to rotate. Intuitively, a smaller contact area provides less torsional friction capacity, which allows the bar to begin pivoting earlier during the release and thus accumulate more angular velocity before detachment. Despite this simple expectation, these transient dynamics, occurring within ∼50ms, are remarkably difficult to model and simulate faithfully [17]. As shown in [18], combining limit surface models [19], [20] with rigid-body dynamics produces pathological Zeno-like oscillations during the release. Only

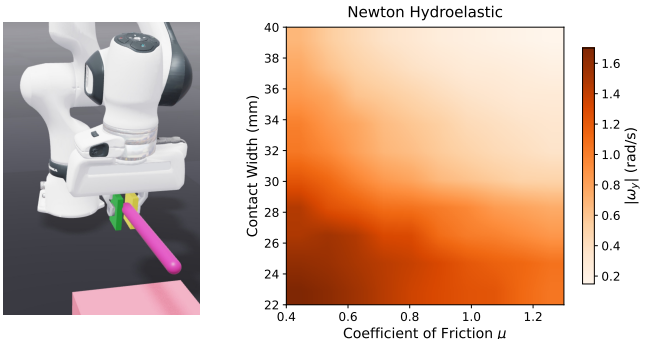

Fig. 4. (Left) Pen-dropping simulation setup. (Right) Sensitivity of the release angular velocity $|\omega_y|$ at detach.

specific remedies (e.g., viscous smoothing with implicit integration) resolve this instability, yet none of these are built into the hydroelastic formulation tested here. The question is whether the distributed pressure field, discretized into point contacts via the method of Section II, can nonetheless produce physically plausible release dynamics.

Following [21], we sweep friction coefficient μ and contact width, with the latter varied by shifting the bar within the gripper (the CoM offset relative to the patch center is held fixed so gravity torque is unchanged). The hydroelastic model automatically resolves the resulting contact patch from the finger and bar SDFs. Fig. 4 (Right) shows two trends: smaller μ and smaller contact area both yield faster pivoting at detachment, consistent with ILS predictions [18] under the same sweep (ILS is the most accurate model in that work's real-robot throwing benchmark). The landscape is smooth and monotonic, without the spurious structure that plagues classical limit-surface models in the transient regime. The contact area governing these dynamics is not a tuned parameter (as in the ILS model) but emerges directly from the SDF geometry, enabling physically consistent predictions across novel objects and grasps without per-scenario specification.

VI. Future Work

We plan to validate sim-to-real transfer of manipulation policies trained with hydroelastic contact, including policies that leverage the tactile sensing model of Section IV, and to develop system identification guidelines to tune our contact model with real-world data. On the algorithmic side, hardware texture sampling for SDF queries could substantially accelerate isosurface extraction [5], and we intend to stress-test the contact reduction scheme as well as its interplay with different solver formulations under more challenging configurations and edge-cases. Differentiability of our SDF-based implementation is an exciting avenue for future work, with preliminary experiments showing promising gradients to optimize for k. Finally, we aim for a more comprehensive benchmark against Drake's tetrahedral hydroelastic model [3], acknowledging that direct comparison is challenging due to differences in underlying formulations.

We are excited about the many downstream applications that our implementation can enable, and we hope it will help advance research in contact-rich manipulation.

REFERENCES

[1] R. Elandt, E. Drumwright, M. Sherman, and A. Ruina, "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 8238–8245.

[2] J. Masterjohn, D. Guoy, J. Shepherd, and A. Castro, "Velocity level approximation of pressure field contact patches," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 593–11 600, 2022.

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

[4] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and Z. Corse, "Local optimization for robust signed distance field collision," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 3, no. 1, pp. 1–17, 2020.

[5] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu *et al.*, "Factory: Fast contact for robotic assembly," *arXiv preprint arXiv:2205.03532*, 2022.

[6] Google DeepMind, "MuJoCo: Multi-joint dynamics with contact," https://github.com/google-deepmind/mujoco/releases/tag/3.0.0, Oct. 2023, version 3.0.0. [Online]. Available: https://github.com/google-deepmind/mujoco/releases/tag/3.0.0

[7] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[8] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87. New York, NY, USA: Association for Computing Machinery, 1987, p. 163–169. [Online]. Available: https://doi.org/10.1145/37401.37422

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

[10] The Newton Contributors, "Newton: GPU-accelerated physics simulation for robotics and simulation research," 2025. [Online]. Available: https://github.com/newton-physics/newton

[11] M. Macklin, "Warp: A High-performance Python Framework for GPU Simulation and Graphics," Mar. 2022, nVIDIA GPU Technology Conference (GTC). [Online]. Available: https://github.com/NVIDIA/warp

[12] Google DeepMind, "Mujoco warp," accessed: 2026-05-29. [Online]. Available: https://github.com/google-deepmind/mujoco_warp

[13] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra, "Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," in *IEEE Robotics and Automation Letters*, vol. 7, no. 2. IEEE, 2022, pp. 3930–3937.

[14] Z. Si and W. Yuan, "Taxim: An example-based simulation model for gelsight tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2361–2368, 2022.

[15] I. Akinola, J. Xu, J. Carius, D. Fox, and Y. Narang, "Tacsl: A library for visuotactile sensor simulation and learning," *IEEE Transactions on Robotics*, 2025.

[16] D. P. Leins, F. Patzelt, and R. Haschke, "Hydroelastictouch: simulation of tactile sensors with hydroelastic contact surfaces," *arXiv preprint arXiv:2501.08077*, 2025.

[17] M. Antali and P. L. Varkonyi, "The nonsmooth dynamics of combined slip and spin motion under dry friction," *Journal of Nonlinear Science*, vol. 32, no. 4, p. 58, 2022.

[18] Y. Liu and A. Billard, "On transient release dynamics in robot throwing: A sliding pivot model," *IEEE Transactions on Robotics*, vol. 42, pp. 132–151, 2025.

[19] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction part 1. limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, 1991.

[20] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996.

[21] B. Acosta, W. Yang, and M. Posa, "Validating robotics simulators on real-world impacts," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6471–6478, 2022.