Progressive Simulation for Cloth Quasistatics

JIAYI ERIS ZHANG, Adobe & Stanford University, USA
JÈRÈMIE DUMAS, Adobe, USA
YUN (RAYMOND) FEI, Adobe, USA
ALEC JACOBSON, Adobe & University of Toronto, Canada
DOUG L. JAMES, Stanford University, USA
DANNY M. KAUFMAN, Adobe, USA

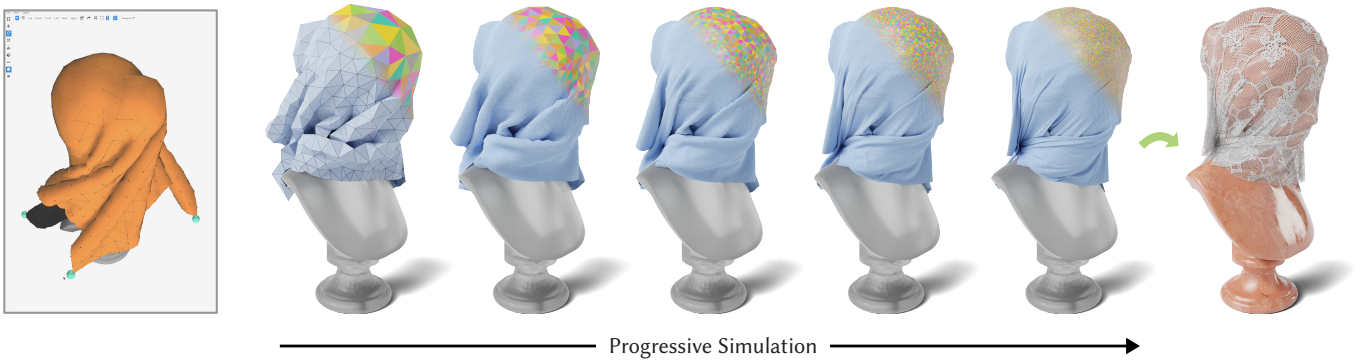

Fig. 1. **Progressive simulation of cloth interactively draped on a marble bust.** (Left) Progressive Cloth Simulation (PCS) provides custom coarse-level simulation of cloth quasistatics (here with 1.5K triangles) where users can easily perform interactive physics-based manipulation to achieve a desired draping outcome. (Middle) At any time, PCS resolves a progressively finer sequence of cloth meshes with self-consistent deformations at equilibrum (shown here with 5 levels). Furthermore, (Right) the resulting finest-level cloth solution (here with 371K triangles) is a fully converged simulation of the underlying cloth model's equilibrium, complete with high-fidelity wrinkles, folds, and intersection-free frictional contact behavior that is consistent with the solutions previewed by the coarser levels. By enabling fine-level cloth simulation consistent with the original coarse-level interactive model, PCS provides an interactive and predictive way to perform high-fidelity cloth simulation.

The trade-off between speed and fidelity in cloth simulation is a fundamental computational problem in computer graphics and computational design. Coarse cloth models provide the interactive performance required by designers, but they can not be simulated at higher resolutions ("up-resed") without introducing simulation artifacts and/or unpredicted outcomes, such as different folds, wrinkles and drapes. But how can a coarse simulation predict the result of an unconstrained, high-resolution simulation that has not yet been run?

We propose Progressive Cloth Simulation (PCS), a new forward simulation method for efficient *preview* of cloth quasistatics on exceedingly coarse triangle meshes with consistent and progressive improvement over a hierarchy of increasingly higher-resolution models. PCS provides an efficient coarse previewing simulation method that predicts the coarse-scale folds and wrinkles that will be generated by a corresponding converged, high-fidelity C-IPC simulation of the cloth drape's equilibrium. For each preview PCS can generate an increasing-resolution sequence of *consistent* models that progress towards this converged solution. This successive improvement can then be interrupted at any point, for example, whenever design parameters are updated. PCS then ensures feasibility at all resolutions, so that predicted solutions remain intersection-free and capture the complex folding and buckling behaviors of frictionally contacting cloth.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Progressive Simulation, Multiresolution, Cloth Simulation, Contact Mechanics

ACM Reference Format:
Jiayi Eris Zhang, Jèrèmie Dumas, Yun (Raymond) Fei, Alec Jacobson, Doug L. James, and Danny M. Kaufman. 2022. Progressive Simulation for Cloth Quasistatics . *ACM Trans. Graph.* 41, 6, Article 218 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555510

Authors' addresses: Jiayi Eris Zhang, Adobe & Stanford University, USA, eriszhan@stanford.edu; Jèrèmie Dumas, Adobe, USA, jedumas@adobe.com; Yun (Raymond) Fei, Adobe, USA, yfei@adobe.com; Alec Jacobson, Adobe & University of Toronto, Canada, jacobson@cs.toronto.edu; Doug L. James, Stanford University, USA, djames@cs.stanford.edu; Danny M. Kaufman, Adobe, USA, dannykaufman@gmail.com.

0730-0301/2022/12-ART218 $15.00
https://doi.org/10.1145/3550454.3555510

1 INTRODUCTION

A dilemma all simulation tasks face is the fundamental trade-off between speed and fidelity. This balancing act is especially tricky when it comes to cloth simulation. High-fidelity methods remain slow but are increasingly expressive with simulation results that closely mimic and even, in some cases, accurately depict the world

around us [Romero et al. 2021]. Fidelity, in turn, offers controllability and expressive range: changes in known, real-world parameters (e.g., materials) result in corresponding, expected changes in simulated shape. On the other hand, high-speed methods for cloth simulation are now reaching interactive rates for extreme-scale cloth meshes, e.g., millions of vertices stepped in less than a second [Wang 2021]. For now, however, there is no "free lunch." High resolution, on its own, does not support nor imply high-fidelity results. The astonishing speeds reached by these methods currently require many simplifying and limiting assumptions in the simulation model to ensure computation stays within budget. In turn, these modeling choices significantly restrict the fidelity and expressive range of the material properties, geometries, and contact behaviors we can simulate with them.

In the long run, research aims to bring these two disparate threads into synchrony. In the short term, however, almost all physical modeling tasks must confront the need to 1) explore the physical impact of design variations (changing material properties, shape, boundary conditions, etc.) and 2) as-rapidly-as-possible obtain reliable simulation feedback to iterate on these design choices. This decision pattern is replicated across applications that range from computational experiments and design optimization to scene layout and cloth modeling.

In this work, we aim to leverage the expressiveness of high-fidelity simulation while minimizing its burden of slow computation. Our goal is to enable rapid and even interactive exploration of design parameters while retaining the expressiveness and accuracy of slow, high-fidelity cloth simulation methods.

To do so we propose Progressive Cloth Simulation (PCS), a coarse-mesh previewing, quasistatic cloth simulator with frictional contact and (when required) strain limiting whose output is progressively improved, at any specified configuration, resolution-by-resolution with consistent shape and folds across all levels. At the preview level, applications can apply PCS to explore changes in material parameters and boundary conditions with quick feedback, using exceedingly coarse meshes (see Figure 5). Then, when satisfactory results are achieved, PCS enables progressive improvement of the predicted equilibrium drape up to a finest-resolution, *converged* solution of the underlying cloth simulation model (see Figure 1). This is in contrast to control-based methods which maintain consistency by constraining a fine-resolution solution to resemble the coarse model result.

Two endemic failure modes in cloth simulation are intersections from self-contact and membrane locking in materials that are close to inextensible. To address these potential issues, we build PCS upon the barrier-based Codimensional Incremental Potential Contact (C-IPC) [Li et al. 2021] cloth model, which importantly provides a guarantee of non-intersection and, when needed, strain-limit satisfaction. We design PCS progression to carefully preserve these constraints, at all resolutions, while guiding the progressive improvement of the cloth simulation towards a convergent, high-resolution, intersection-free solution.

PCS provides consistent solutions across levels by leveraging two key insights. First, we obtain high-quality, artifact-free, fine-level previewing geometries, at each coarse-level approximation, by biasing their solutions with efficiently computed shell forces and energies evaluated on the finest-level model. Second, we then initialize each finer-level's solve with *safe* (intersection-free and strain-limit-satisfying) prolongations of these converged, artifact-free solutions from the prior-level's coarser model and so ensure feasible progression to nearby solutions at each successive level.

In turn, the overhead for simulating PCS's *coarsest-level* "preview" simulation mode, over that of a standard coarse-mesh solve, is just the easily parallelized gradient and energy evaluations of the fine-level membrane and bending energies, resulting in minimal additional simulation costs for exploring preview solutions over varying scene parameters and/or user manipulations of BCs.

Correspondingly, while not the primary focus of this work, for directly simulating *finest-level* equilibrium solutions (when parameters and BCs are preselected), PCS's progressive solver also provides a significant advantage. Here we observe a well-over 10x speedup over well-engineered Newton solves (minimizing total energy) to directly obtain a final, converged, high-resolution drape (even when no preview is desired). See §6.1.

Contributions. PCS achieves consistent contact- and wrinkle-aware up-resing of cloth with practical, high-quality results via interaction that, to our knowledge, are impossible with prior methods. We demonstrate PCS's efficacy with a wide range of stress tests and an interactive demo across challenging deformations, severe contact-conditions, and real-world cloth material parameters. Our technical contributions include

- A nonlinear, coarse-level simulation model enriched with fine-level shell energies and forces that 1) eliminates membrane-locking artifacts in coarse-level simulations, and 2) strongly biases progressive solutions across levels to a consistent final drape;
- A safe prolongation method that resolves exact contact-handling (preventing intersections) and strain-limits for all refinement operations;
- A multiresolution progressive simulation-solver framework that provides efficient solutions of each coarse-level's enriched nonlinear shell model, and a final, fully converged, and so high-quality, solution for the finest-level drape geometries;
- Interactive editing of all scene parameters (See §5.3), at all times, by stable interruptions of the progressive solver at any level;
- Enabling significant speedup (e.g., 10x) over well-engineered Newton codes when directly solving for fine-resolution static drape solutions.

Together these simple and effective contributions provide high-quality (intersection-free, strain-limited, and converged) as-consistent-as-possible nonlinear solution previews at all levels of PCS's progressive hierarchy.

2 RELATED WORK

Simulating thin-shell mechanics remains a long-standing focus in both computational mechanics and computer graphics, where efforts have especially focused on the efficient modeling of cloth [Baraff and Witkin 1998; Bridson et al. 2002; Grinspun et al. 2003; Harmon et al. 2009; Li et al. 2020; Narain et al. 2012; Terzopoulos et al. 1987; Volino and Thalmann 2000]. Pipelines for shell simulation most commonly adopt implicit time-integration methods [Baraff and Witkin 1998; Bridson et al. 2002; Kim 2020; Li et al. 2020, 2018; Narain et al. 2012;

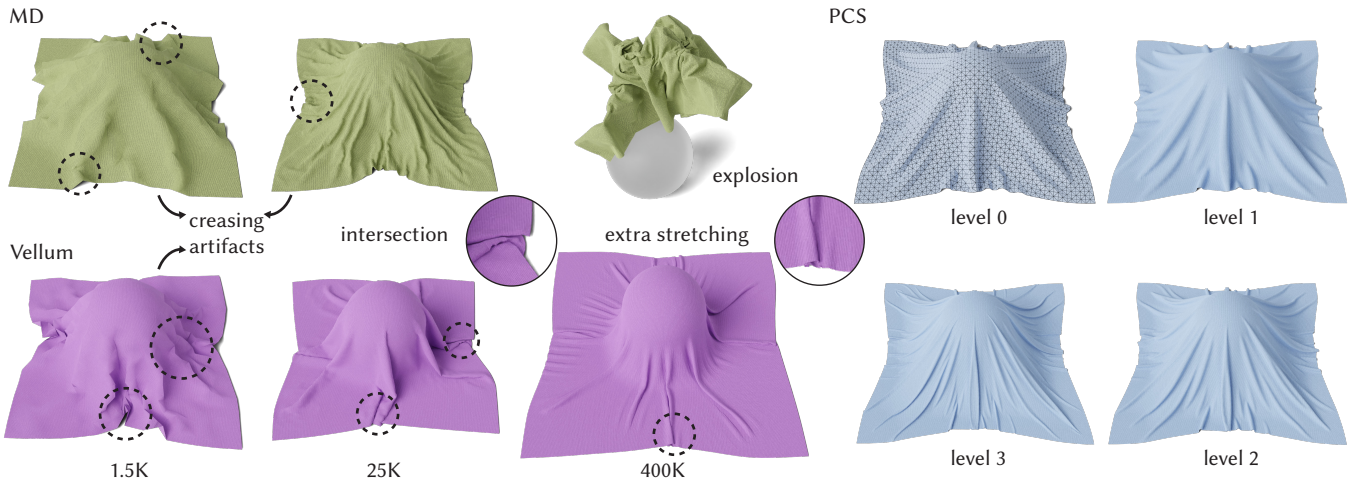

Fig. 2. **Consistent and stable cloth up-resing.** Designing with interactive coarse cloth models and then re-running at a higher resolution can lead to undesirable and unpredictable outcomes in traditional cloth solvers. For example, (Left) both Marvelous Designer (MD, in green) and Vellum (in purple) produce numerous undesirable creasing artifacts at low mesh resolutions, which are notably absent in (Right) the coarse level-0 PCS solution due to its enrichment with fine-scale cloth forces and energy evaluations. Furthermore, trying to "up-res" coarse cloth designs by then re-running them at higher resolutions leads to unpredictable results: (Left) both MD and Vellum exhibit dramatically different folds across resolutions, as well as artifacts such as intersections and inconsistent material stretching in Vellum, and explosive instabilities in MD. In contrast, (Right) PCS achieves a high-fidelity, artifact-free, fine-scale cloth simulation consistent with the interactive coarse mesh preview.

Otaduy et al. 2009; Tang et al. 2016, 2018] with a wide range of filters often applied to help improve contact processing [Bridson et al. 2002; Harmon et al. 2008] and limit strain [Goldenthal et al. 2007; Narain et al. 2012].

Speed, Fidelity and Expressiveness. To further accelerate performance, high-speed methods offering new formulations [Bender et al. 2013; Bouaziz et al. 2014; Daviet 2020; Ly et al. 2020; Zhang et al. 2019] and leveraging multi-core and GPU architectures [Li et al. 2020; Schmitt et al. 2013; Selle et al. 2008; Tang et al. 2013, 2016, 2018] have demonstrated the ability to simulate high-resolution meshes at ever-improving speeds, and even, in the last few years, at increasingly interactive rates [Designer 2022; Li et al. 2020; SideFX 2022; Tang et al. 2018; Wang 2021; Wu et al. 2020]. At the same time, increasingly high-fidelity, and so controllably expressive methods for shells have enabled ever-improving simulations of cloth with advances in both constitutive modeling [Chen et al. 2018; Clyde et al. 2017; Guo et al. 2018; Jiang et al. 2017; Miguel et al. 2012; Narain et al. 2012; Weischedel 2012] and contact [Harmon et al. 2009; Li et al. 2018, 2021; Narain et al. 2013; Vouga et al. 2011]. In turn, each new generation of these methods provides simulations of complex shell behaviors not possible previously.

These two classes of methods have, not surprisingly, very different characteristic trade-offs. Recent, high-fidelity methods, while able to accurately capture complex shell mechanics, are nowhere near to achieving interactive-rate simulations for practical mesh resolutions, even for well-optimized implementations [Chen et al. 2021; Li et al. 2021]. On the other hand, in order to achieve performance, recent advances in high-speed methods require many simplifying and limiting assumptions that ensure computation stays within budget [Designer 2022; SideFX 2022; Wang 2021; Wu et al. 2020]. These assumptions significantly restrict the range of material properties, geometries, and interaction behaviors that can be simulated and so, in turn, limit their expressive-range [Li et al. 2021; Wang 2021], e.g., see Figure 2.

We seek to leverage the expressiveness of these high-fidelity simulation methods while minimizing the burden of slow computation. Here, an outstanding goal is to enable interactive exploration of scene parameters for application in design, computational experiments, surrogate sampling, machine learning, garment draping, staging, robotics, layout, and many other tasks while retaining the expressiveness and accuracy of slow, high-fidelity cloth simulation methods. We propose PCS as a new, exceedingly coarse-mesh surrogate simulator (in our experiments, generally well over a 250X reduction in mesh size) for high-resolution cloth models that generates a corresponding hierarchy of progressively refined simulation meshes.

Enrichment. To achieve a balance between efficiency and expressiveness, many recent explorations focus on enriching coarse (and so quickly simulated) meshes with post-processed, higher-resolution wrinkle and fold details. These methods either hallucinate geometry from computed subspaces [Gillette et al. 2015; Hahn et al. 2014; Jin et al. 2020; Kavan et al. 2011; Müller and Chentanez 2010; Zurdo et al. 2012] or else augment details by imposing a secondary elasticity model [Chen et al. 2021]. However, in all such approaches, augmented details are again, as in high-speed methods, limited in their expressiveness via restrictions imposed by interpolation, one-way coupling and/or missing contact interactions [Chen et al. 2021]. At

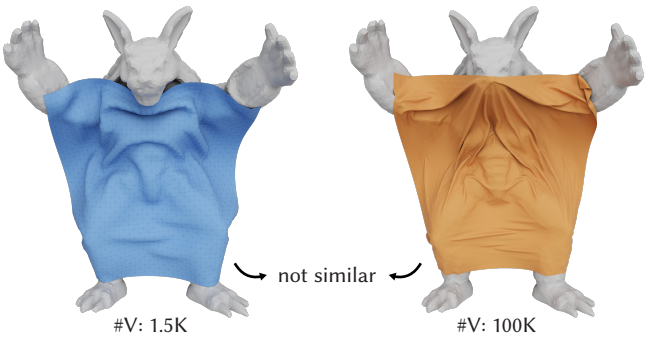

Fig. 3. **Previsualization results,** using coarse (Left) simulation, give "previews" that are generally far from final (Right) simulation results that are then produced by slow, high-resolution simulations using the same selected settings.

the same time, these methods also often require significant preprocessing, depend on a corpus of data [Lahner et al. 2018; Santesteban et al. 2019], and/or impose prohibitive runtime costs [Chen et al. 2021].

Previsualization. A longstanding alternative then is previsualization provided by fast, coarse-mesh simulation utilized as a preview. This preview mode is used to explore scene variations (e.g., geometry, materials, boundary conditions) until results look good. Then, once these scene settings are finalized, previsualization is complete, and we commit to simulating an upsampled high-resolution geometry with the same parameters. While this latter, final "hero" simulation will be slow to run, the hope is that it will directly provide an appropriately consistent, high-quality analog to the user's hand-crafted, coarse-level simulation. In practice, however, this rarely works out as planned. Issues with this strategy are well-documented and well-understood: models that differ significantly in resolution provide very different accuracies and so solution results between fine and coarse models can correspondingly diverge (see Figure 3) by large amounts [Bergou et al. 2007]. In turn, for cloth, these problems are only exacerbated by the bifurcations introduced as higher resolutions enable more buckling, folding and contact interactions.

Constraints. To address these challenges, the TRACKS method [Bergou et al. 2007] augments fine-mesh simulations with moment-based constraints that track the deformed shape of input target coarse-resolution geometries in an average sense. However, to do so, the target input must also provide the fine simulator with a choice (generally automated via heuristics) of subdomain cells to define the weighted-average (centroid) constraints. Differing cell choices result in different output behaviors in the final fine-mesh simulation, with hard-to-predict changes in results. In turn, such hard constraints pull fine-level output away from the unconstrained local minima that define the natural equilibria of a cloth drape. At the same time, the constraints themselves can not and do not ensure that local deformations in the coarse simulation result will be captured by the fine [Bai et al. 2016].

Progressive Refinement. Here, our goal is to invert the problem rather than constraining high-quality, fine-level simulation results to mimic low-quality, coarse simulation output. We seek a preview simulation method whose results, *by construction*, predict the coarse-scale folds and wrinkles generated by a corresponding *unconstrained*, high-fidelity, high-quality cloth simulation using the same input scene settings. Moreover, this predictive ability should hold across all reasonable cloth material parameter variations, without restriction, rather than for a limited, customized-for range. To do so, however, there are numerous challenges to address, including, perhaps the most pressing riddle: *how can a coarse simulation predict the result of an unconstrained, high-resolution simulation that has not yet been run?*

Towards this goal, we begin with the strategy of *progressive refinement* proposed by Umetani et al. [2011] for their Sensitive Couture (SC) method. Umetani and colleagues propose a vision for simulation that, analogous to progressive geometry refinement [Hoppe 1996], hierarchically and successively improves solutions starting from a coarse simulation and ending in a final, high-quality simulation result. Analogous to the *linear* Cascadic Multigrid method [Bornemann and Deuflhard 1996], SC solves a succession of increasingly finer *nonlinear* cloth drape problems, passing the converged solution at each coarser level (via upsampling) as the warm start to the next finer-level problem. This simple, nonlinear multi-level approach has since been adopted as a module within diverse cloth simulation algorithms, see e.g., We et al. [2020] and Wang [2021]. However, here we observe a fundamental problem with the method: cascading a series of warm starts in this way neither ensures consistency (see Figure 6) across increasing resolution models (e.g., nonconvexity enables arbitrary and equally correct "fold jumps" across levels) nor avoids unacceptable artifacts (e.g., kinks) in the final, high-resolution mesh solutions; see Figure 4. We discuss and analyze these artifacts further in §6.

Multigrid Methods and Subdivision Models. To improve the fundamental bottleneck of *linear* system solves in simulating shell models, a range of custom multigrid methods [Tamstorf et al. 2015; Wang et al. 2018; Xian et al. 2019] are actively being pursued. Similarly, subdivision-based models continue to attractively offer improved accuracy of underlying shell model discretizations [Cirak et al. 2000; Green et al. 2002; Grinspun et al. 2002; Kopaničáková et al. 2019; Thomaszewski et al. 2005]. PCS tackles distinctly different and complementary goals from both these classes of methods. It targets the discrete shells simulation of a high-resolution, finest-level triangle mesh (neither simulating with a subdivision-based shell model nor limit surface) and does not apply multigrid preconditioning to accelerate linear system solves. In principle, PCS could thus be combined with these prior methods in future work for further improvement.

Nonlinear Multilevel Methods. Taking a step beyond *linear* multigrid methods, SC's cascadic method [Umetani et al. 2011] fits well within the broader scope of nonlinear multilevel (NML) optimization methods [Henson 2003; Ho et al. 2019]. Here a broad range of hierarchical methods, some designed as nonlinear analogs of well-known multigrid methods [Briggs et al. 2000], have been proposed to solve variational problems. Trust-region-based NML methods [Gratton et al. 2008] have recently been applied to improve convergence when solving cloth simulations without contact handling [Kopaničáková et al. 2019]. However, with the added heterogeneity

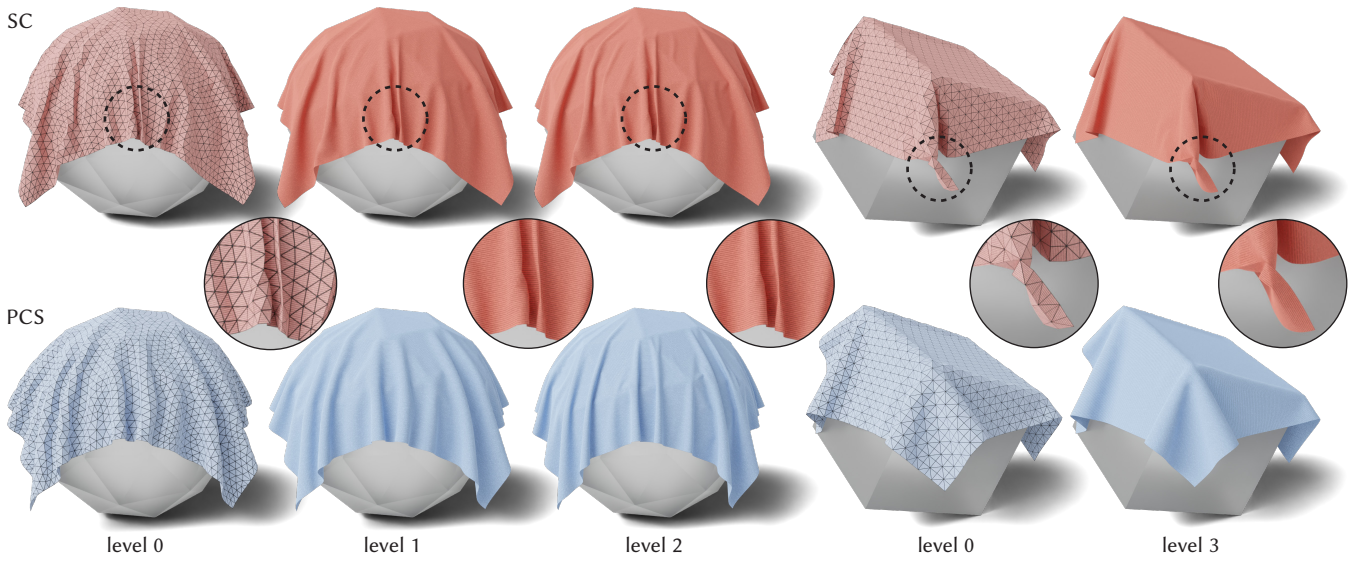

Fig. 4. **Avoiding cascadic artifacts.** (Top) Using Sensitive Couture (SC) cascadic method to hierarchically and successively refine a coarse solution can lead to unnatural coarse-level locking artifacts (sharp and unnatural creasing and buckling errors) being inherited by fine-scale solutions. (Bottom) In contrast, PCS uses fine-scale information to avoid artifacts in the coarse-scale solution, and thereby estimates a consistent fine-scale solution free of these artifacts.

and stiffness introduced by contact handling, it is unclear whether multilevel methods can in fact improve over a well-implemented Newton solver. In our exploratory testing, we have found that NML solvers offer neither speed-up nor improved convergence over a well-engineered Newton solver for shell draping with contact. At the same time, these methods also do not provide previews from their coarse approximations. Individual lower-resolutions, when extracted prior to full convergence, have no objective requiring these lower-resolution scratchpads to provide reasonable previews prior to fine-level convergence. Meanwhile, existing multilevel methods cannot resolve frictional contact nor strain-limit constraints. In contrast, PCS efficiently solves an enriched equilibrium, subject to frictional contact and strain-limits, at each level, to convergence, which obtains effective, high-quality previews. PCS then passes these solutions on, with safety, to warm-start each next-level solve. At the same time, PCS also provides significant speed-up over Newton solvers in directly reaching the final, high-resolution shell-drape equilibrium.

3 FORMULATION

We target the simulation of a high-resolution triangulated domain with n nodes stored in a vector $x \in \mathbb{R}^{3n}$. Our goal then is to enable controllable, expressive and *progressive* cloth modeling that

(1) obtains high-quality (and so convergent, non-interpenetrating, and artifact-free), fine resolution simulations of cloth drape geometries, in any configuration, across a full range of real-world material parameters;
(2) provides efficient previewing of this simulated cloth's drape in any configuration, utilizing a coarse-resolution mesh, that is *consistent* with the corresponding final, *converged* fine-level simulated configuration; and
(3) enables progressive and consistent resolution improvement of the solution (with increasing cost), starting from the coarse mesh preview and ending with the converged, fine-mesh geometry.

3.1 Hierarchy

To enable this progressive workflow, for each simulation we construct (see §5 for details) a nested triangle mesh hierarchy indexed in increasing resolution by subscript $l \in [0, L]$, where x_l and $\bar{x}_l \in \mathbb{R}^{3n_l}$ are respectively the vectors of deformed and rest positions of the n_l nodes in the mesh at level l, with corresponding triangulation indexed in $\mathcal{T}_l$. To be abundantly clear, x_0 then stores the deformed positions of the coarsest mesh, and x_L is the finest-resolution positions of the target mesh for our final, converged high-quality simulation output.

For the hierarchy, we introduce linear prolongation (full column rank) operators that refine nodes from levels l to $l+1$ as

$$P_{l+1}^{l} \in \mathbb{R}^{3n_{l+1} \times 3n_l},$$

while the corresponding, linear projection operators, via least squares (mapping nodes from levels l to $l-1$) are

$$\Pi_{l-1}^{l} = \left((P_l^{l-1})^T (P_l^{l-1}) \right)^{-1} (P_l^{l-1})^T.$$

To simplify the discussion, we will continue to designate finest-level resolution quantities without decoration, so that, e.g., $x = x_L$, $\bar{x} = \bar{x}_L$, and $n = n_L$.

3.2 Incremental Potential Problems

We equip each simulation mesh with membrane (Ψ), bending (Φ), contact barrier (B), friction (D), and, when required, strain-limiting potential energies (S). Here we use neo-Hookean membrane [Chen

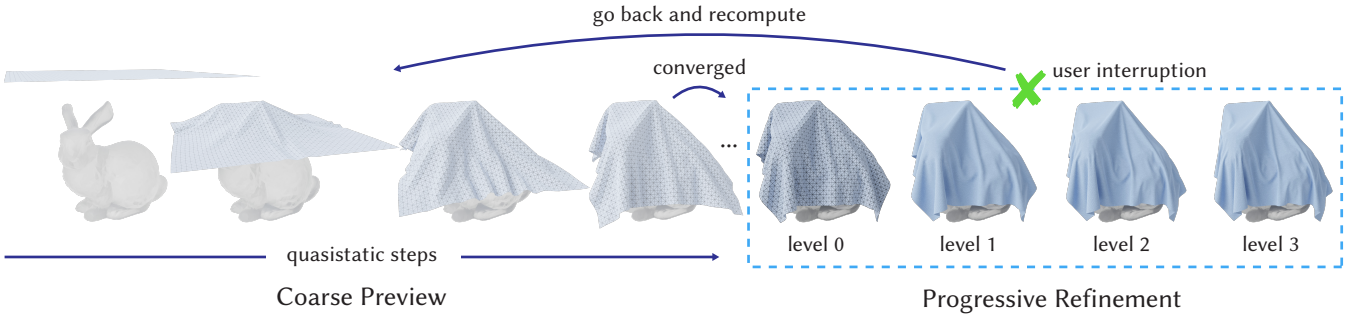

Fig. 5. **PCS workflow.** With quick feedback PCS enables rapid exploration of materials and boundary conditions at the preview level, using exceedingly coarse meshes. Then, when satisfactory results are achieved, PCS enables progressive improvement of the predicted equilibrium drape up to a finest-resolution, converged solution of the underlying cloth simulation model. Progressive simulation can be interrupted at any point to change conditions; for example to update design and scene parameters.

et al. 2018] and discrete hinge bending [Grinspun et al. 2003; Tamstorf and Grinspun 2013] for shell elastics, and C-IPC [Li et al. 2021] barriers for contact, friction and strain limiting.

Here we focus on simulating the *equilibria* of shells in frictional contact. Stable *equilibria* of these discrete shell models are the local minimizers of the total potential energy,

$$E_l(x) = E_l(x, \bar{x}, u), \tag{1}$$

constructed from the sum of the above potentials, $E_l = \Psi_l + \Phi_l + B_l + D_l + S_l$. Here u collects material and boundary condition scene parameters that can be varied by user or application (see §5.3).

Quality and accuracy. A *convergent* and therefore high-quality equilibrium solution of the physics for level l's spatial discretization is then given by a *feasible* (thus nonpenetrating and strain-limit satisfying) geometry x_l^* satisfying $\|\nabla E_l(x_l^*)\| \leq \epsilon$, where we use the Newton decrement norm from C-IPC. Progressive simulation should obtain these convergent solutions across wide variations in material properties and/or boundary conditions.

Exploration via quasistatic stepping. Given a current, nonequilibrium configuration, x^t, we can step towards a stable state by time stepping its gradient flow with implicit Euler. Applying large time steps, h (generally $h \in [0.05, 0.5]$s) to enhance implicit Euler's numerical dissipation, this amounts to computing forward quasistatic position updates from "time steps" t to $t+1$ by the minimization of an updated *incremental* potential,

$$x^{t+1} = \underset{x}{\operatorname{argmin}} \frac{1}{2h^2}\|x - x^t\|_M^2 + E(x, \bar{x}, u^{t+1}). \tag{2}$$

Here x^t is a last step's simulated shape and u^{t+1} collects the time-varying scene parameters that drive forward change by moving configurations out of equilibrium in the simulation steps (see §5 below). For unchanged u, time stepping can be repeated until $x^{t+1} = x^t$, and so a new equilibrium is reached. Note: we reserve superscripts t on nodal vectors to index quasistatic time steps; however, later we will use alternate superscripts to denote stages within our solver's optimization process.

Consistency. While we could hypothetically use the above quasistatic process (or direct minimization) to solve for each level l's equilibria independently, as discussed in §2, solutions across levels will unacceptably diverge from each other and so will be unsuitable for preview or refinement. Here we must next address the fundamental problem that individual, independent solutions at different levels are generally inconsistent, see Figure 6.

We say solutions across levels are *consistent* when overall shape and wrinkle patterns are similar across scales. In the following we will evaluate this empirical notion of consistency in two ways:

Qualitative visual comparisons: Here we refer to figures, supplemental videos, and supplemental geometries (we provide all output meshes for detailed consideration) to demonstrate PCS's significant improvement over both alternatives: SC's cascadic approach, which exhibits unsightly and nonphysical artifacts, and Newton-based direct simulation with even more inconsistencies.

Quantitative analysis: We derive in §6.2, to our knowledge, a first quantitative measure for this consistency distance via mean curvature. Across all examples, we observe that PCS improves on this consistency measure over both alternatives significantly.

See §6 and our supplemental for details.

4 PROGRESSIVE SIMULATION

We seek a one-way, nonlinear multiresolution solver that provides, at each step t, a sequentially improving hierarchy of self-consistent simulation solutions i.e.,

$$x_0^t \approx \Pi_0^1 x_1^t, \ \ldots, \ x_l^t \approx \Pi_l^{l+1} x_{l+1}^t, \ \ldots, \ x_{L-1}^t \approx \Pi_{L-1}^L x_L^t,$$

along *with* requiring our final, finest resolution solution is a *converged* equilibrium solution of the underlying cloth model satisfying $\|\nabla E(x_L^t)\| \leq \epsilon$. At the same time, each individual solution, x_l^t, at all levels of resolution $l \in [0, L]$ must independently provide a stable solution for forward simulation. We divide these tasks into two solver phases (see Figure 5):

Refinement, $x_l^t \rightarrow x_{l+1}^t$: progressive spatial improvement of the solution from level l to level $l+1$, for a fixed set of conditions u^t; and

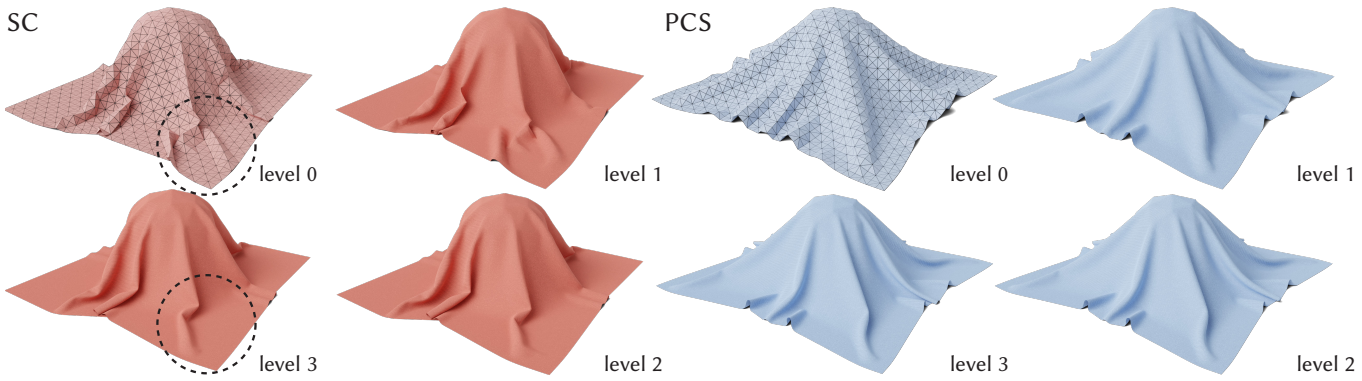

Fig. 6. **Avoiding refinement inconsistencies.** (Left) Cascadic methods like Sensitive Couture (SC) can suffer from inconsistent folds across levels. For example, the crumpled corner on levels 0 and 1 bifurcates into a different fold on levels 2 and 3. (Right) In contrast, PCS provides multi-level consistency so that the original coarse-scale folds are progressively refined predictively.

Preview, $x_l^t \rightarrow x_l^{t+1}$**:** quasistatic advancement of the solution, at level l, from time step t to $t+1$, over possibly varying u.

Refinement simulation ensures progressive improvement, with consistency, towards the converged fine-level solution, for fixed conditions, u^t, given at a time t. *Preview* simulation continues the complementary, forward exploration over continued changes in conditions encoded in time-varying parameters, $u^t \rightarrow u^{t+1}$; see §5 for details on parameters controlled by u and their implementation. While, as we will see, preview stepping with changing parameter exploration is possible at any level of refinement, we focus our evaluation here solely on preview stepping for the coarsest level.

4.1 Coarse-Level Energies

At each coarsened-level $l < L$ we construct a mixed-resolution objective, F_l, as a *proxy energy* for the finest-level potential energy,

$$F_l(x_l) = C_l(x_l) + G(P^l x_l), \tag{3}$$

with coarse, barrier-based potentials collected in

$$C_l(x_l) = B_l(x_l) + D_l(x_l) + S_l(x_l),$$

enforcing contact and strain-limit feasibility on the current level-l's geometry, and shell membrane and bending potentials,

$$G(P^l x_l) = \Psi(P^l x_l) + \Phi(P^l x_l),$$

evaluated at the finest level, using repeated prolongation to the finest scale, $P^l x_l \equiv P_L^{L-1} \cdots P_{l+1}^l x_l$.

Later we will also employ per-level inertial energies,

$$K_l(x_l, y_l) = \frac{1}{2h^2} \|x_l - y_l\|_{M_l}^2, \tag{4}$$

to promote continuity across quasistatic time steps (effective step size h) and spatial consistency across resolutions during refinement. Here the free parameter, y_l, in K_l will differ based on the step and simulation mode. See Sections 4.2, 4.3 and 4.5 for respective details.

4.2 Refinement Simulation

To progress from level $l-1$ to level $l \in [1, L-1]$, at fixed time step $t+1$, we begin with the last level's equilibrium solution for parameters u^{t+1} given by x_{l-1}^{t+1}. We then solve for a *local* minimizer, and so an equilibrium, of the proxy energy

$$x_l^{t+1} = \underset{x_l}{\operatorname{argmin}}\, F_l(x_l). \tag{5}$$

To solve each minimization in (5) we construct a custom, Newton-type descent method, detailed below in §4.4. However, not all minima are equally acceptable. Critical to progressive improvement in each level's optimization is the initialization of the Newton solve.

There are then two key and generally competing requirements for initialization. First, nonconvexities of the shell and contact potentials imply that each level's energy supports many equally physically valid minima (e.g., consider period shifts in cloth folds). In turn, successive Newton iterations descend to a local minima in a basin containing their initializer. Second, in order to obtain a *feasible* (defined as noninterpenetrating and strain-limit-satisfying) minimizer, we require a correspondingly feasible initializer, which may not be satisfied by an otherwise ideal choice.

Here we seek to bias each new level's minimizer to be near to the last level's ***prolonged solution***, $x_l^p = P_l^{l-1} x_{l-1}^{t+1}$. However, as we cover in the next section, while the prolonged solution x_l^p may be an ideal choice to bias the minimization process towards consistency, it will not satisfy feasibility. This feasibility issue holds for $l > 0$, while all coarsest deformations, x_0^t, remain feasible at start of refinement (see §4.5) with the presumption that our starting, coarsest configuration, $x_0^{t=0}$, is feasible.

4.3 Safe Prolongation for Refinement

We start with a feasible initial state at the very first time step of preview simulation. All following time steps must remain feasible throughout. In turn, as we employ barrier based methods, all iterates within our Newton solver (see §4.4), will also preserve feasibility. However, the missing link is that all successive optimizations of F_l, for $l > 0$, must also initialize their Newton solve with a *safe* prolongation of x_{l-1}^{t+1} that is feasible.

Given the solution x_{l-1}^{t+1}, at level $l-1$, the prolonged solution $x_l^p = P_l^{l-1} x_{l-1}^{t+1}$, is our target to initialize the next solve of our level-l proxy energy (3). However, even though mapped from a level $l-1$

feasible point, x_{l-1}^{t+1}, there are no guarantees that x_l^p, will satisfy nonintersection and strain limits at the next level l.

To construct a close-by, feasible initializer we begin by finding a *safe* starting point x_l^s, at level l, that is guaranteed feasible and then search along $d = x_l^p - x_l^s$ to find the closest feasible point to the unconstrained prolongation x_l^p. To compute the safe start we begin by applying barycentric ***up**sampling*, $x_l^{up} = U_l^{l-1} x_{l-1}^{t+1}$ to match the next level's mesh connectivity. As this is a purely topological update that does not affect the carrier geometry, non-intersection constraints are preserved.

For cloth materials that also require strain-limits, we construct a feasible-start optimization. We begin by leveraging the above observation that each coarse triangle domain in the fine mesh geometry, x_l^{up}, roughly satisfies the strain-limit, since it does so exactly at the prior level $l-1$. To take advantage of this we first fix all "even" vertices, i.e., those copied over from level $l-1$, in x_l^{up}. We then iteratively solve for new positions on just the remaining upsampled, "odd" vertices via a Newton-type optimizer that pulls our upsampled solution, x_l^{up}, towards our prolongated target, x_l^p, while preserving all constraints currently satisfied, and improving those that are not. At each iteration we visit all triangles in level l. We place all faces $f \in \mathcal{T}_l$ currently satisfying the strain limit in the feasible set, $\mathcal{F}$, and the remainder, over-the-limit faces, are set in the complement $\mathcal{F}^c$. We then form the energy

$$K_l(x, x_l^p) + B_l(x) + S_l|_{\mathcal{F}}(x) + A_l|_{\mathcal{F}^c}(x),$$

where B_l and S_l respectively ensure that no intersections nor new strain violations are possible, K_l penalizes change away from the prolongation, and a restricted ARAP [Chao et al. 2010; Sorkine and Alexa 2007] energy,

$$A|_{\mathcal{F}^c}(x) = \kappa_a \sum_{f \in \mathcal{F}^c} \sum_{i \in [1,2]} (\sigma_{f_i} - 1)^2,$$

then exactly provides the needed strain-limiting penalty to pull each triangle's principal stretches, σ_{f_i}, below their limits. With descent steps decreasing $\mathcal{F}^c$, the resulting optimization then generates the feasible safe-starting geometry, x^s. See Algorithm 1 for details.

Algorithm 1 Feasible Initialization

procedure FeasibleSolve(x_{l-1})
  $x^p \leftarrow P_l^{l-1} x_{l-1}$
  define $K_l(\cdot) = K_l(\cdot, x^p)$
  $x_l \leftarrow U_l^{l-1} x_{l-1}$
  $\mathcal{F} \leftarrow \emptyset$
  $\mathcal{F}^c \leftarrow \mathcal{T}_l$
  do
    for $f \in \mathcal{F}^c$ **do**
      if $\sigma_{f_{i\in[1,2]}} <$ strain limit **then**
        $\mathcal{F}^c \leftarrow \mathcal{F}^c \setminus f$
        $\mathcal{F} \leftarrow \mathcal{F} \cup f$
    $C \leftarrow$ ComputeConstraintSet(x_l)
    define $J(x_l) = K_l(x_l) + B_l(x_l) + S_l|_{\mathcal{F}}(x_l) + A_l|_{\mathcal{F}^c}(x_l)$
    $H \leftarrow$ ProjectPD$\left(\nabla^2 J_l(x_l)\right)$
    $H \leftarrow$ ProjectEvenVertices(H)
    $g \leftarrow$ ProjectEvenVertices$\left(\nabla J(x_l)\right)$
    $d \leftarrow -H^{-1} g$
    $\alpha \leftarrow \min\left(1, \text{StepSizeFilter}(x_l, d, C)\right)$
    $\alpha \leftarrow$ LineSearch(F_l, x_l, d, α)
    $x_l \leftarrow x_l + \alpha d$
  while $\|d\|_2 > \epsilon_i$
  $C \leftarrow$ ComputeConstraintSet(x_l)
  $d \leftarrow x^p - x_l$
  $\beta \leftarrow \min\left(1, \text{StepSizeFilter}(x_l, d, C)\right)$
  $x_l \leftarrow x_l + \beta p$
  return x_l

Algorithm 2 Optimization for Level-l Proxy Steps

procedure ProxyStep(x_l, y_l, ϵ, u, max_iter)
  UpdateSystem(u) ▷ update materials, geometry and BCs
  define $K_l(\cdot) = K_l(\cdot, y_l)$
  $i \leftarrow 0$
  while $i <$ max_iter **do**
    $C \leftarrow$ ComputeConstraintSet(x_l)
    $H \leftarrow \frac{1}{h^2} M_l +$ ProjectPSD$\left(\nabla^2 E_l(x_l)\right)$
    $g \leftarrow (P^l)^T \nabla G(P^l x_l) + \nabla C_l(x_l) + \nabla K(x_l)$
    $d \leftarrow -H^{-1} g$
    if $\|d\|_2 > \sqrt{n_l} h \epsilon$ **then** break
    $\alpha \leftarrow \min\left(1, \text{StepSizeFilter}(x_l, d, C)\right)$
    $\alpha \leftarrow$ LineSearch(F_l, x_l, d, α)
    $x_l \leftarrow x_l + \alpha d$
    $i \leftarrow i + 1$
    if $i >$ max_iter **then** break
  eq $\leftarrow$ (i == 0) ▷ equilibrium at x_l if no iterations taken
  return (x_l, eq)

4.4 Minimizing Coarse Proxy Energies

While we could directly apply a projected Newton's method to solve each optimization of the proxy energy in (5), this would require the computation at each iteration of the $n \times n$ global Hessian contribution, $\nabla^2 G(P^l x_l)$, from the bending and membrane energies. Combining with fill-in from its subsequent reduction to $(P^l)^T \nabla^2 G(P^l x_l) P^l$, this would further result in evaluation, assembly and linear-solve costs comparable to that of the original, fine-level model solve we are seeking to avoid in the first place.

At the same time, waiting for a final result of this potentially costly optimization per level, with no intermediate preview of output, means that significant time could be wasted during refinement that could better be spent on new parameter explorations. To provide many intermediate and stable updates for previewing (and so opportunities to change parameters) during each level's refinement we apply quasistatic stepping via *approximate* solves of the incremental potentials formed by the sum of the proxy and inertial energies, $F_l + K_l$.

To solve each such proxy step we apply an inexact Newton solve by preconditioning the proxy and inertial energy's gradient with

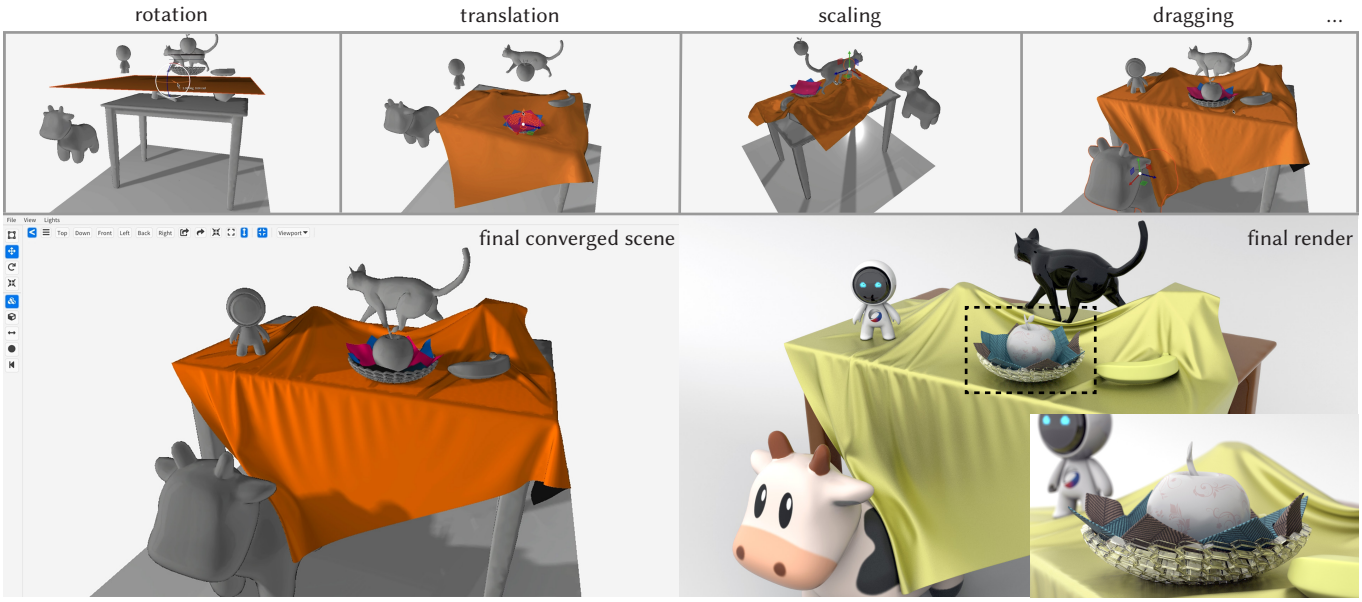

Fig. 7. **Still life design with PCS.** (Top) PCS enables rapid design and layout via a diverse range of interactions, including handle-based manipulation and material editing. With PCS's robust collision-processing users can also directly apply transforms to complex collision objects to intuitively push cloth around and between shapes. Here layers of cloth are interactively draped on a table and into a bowl while scene shapes are pushed against them. As equilibrium is reached for edits, PCS automatically starts progressive simulation to finer-level solutions. Any new changes interrupt this process and allow for fine-tuning. When no further edits are desired, PCS proceeds to a (Bottom left) converged, finest-level model providing (Bottom right) high-quality, non-intersecting results for tightly nested and complex geometries. See our supplemental video for more details.

the projected Hessian of the *unmodified, coarse-model's* incremental potential, $E_l + K_l$. Please see Algorithm 2 and §5 for details. This process scales all barrier gradients with their corresponding second-order information, while we use the sparse, coarse membrane and bending Hessians to approximate the second-order information for their fine-level gradient counterparts.

In practice we observe rapid, Newton-like convergence for this solver, close to that of directly solving a coarse-mesh optimization with Newton iteration; see §6.2. *The overhead for solving the proxy energy, over that of a standard coarse-mesh solve, is then just the easily parallelized gradient and energy evaluations of the fine-level membrane and bending energies.* In turn, each individual proxy step need not be solved accurately. In the extreme, as we cover in the next section, even *a single iteration* per proxy-step solve provides both rapid convergence to equilibria as well as *stable*, easily interruptible, intermediate previews during refinement.

4.5 Progressive Simulation

Coarse-level *preview* simulation then follows refinement simulation closely. For preview stepping, however, no additional preprocessing is required to initialize simulation. A preview time step from t to $t+1$ at level $l < L$ (in our implementation solely at coarsest level $l = 0$) always begins with a prior feasible state x_l^t, at the same level. This configuration either comes from a prior time-step solve just applied (and so is guaranteed feasible by all prior simulation solves at level l), or else is sourced from a solution cached at the start of the last refinement solve. If the latter case, this cached state is then applied as x_l^t to restart preview simulation whenever this refinement is interrupted–more on this below. We then simulate the preview-level system forward via quasistatic time stepping towards equilibrium of the proxy energy with updates in parameters u applied at the start of each step's solve.

For progressive simulation we generally time step previews, via Algorithm 3, at the coarsest level ($l = 0$). We then sequentially move from preview solution, x_0^t to x_0^{t+1} with each step's (possibly updated) material and boundary condition parameters, u^{t+1}. We continue this process until, at some time t, we reach an equilibrium preview configuration, x_0^t, that we are (e.g., based on application or user feedback) interested in refining (start_refinement()).

At this point we trigger a refinement simulation sequence to sequentially solve across increasing resolution levels $l \in [1, L]$. After each level's solve we process and pass the current solution, via safe prolongation, on to start the next. At any point in this refinement simulation process we may be interrupted (end_refinement()), e.g., via user interaction changing parameters, u. If so, we then switch back to preview simulation starting with the cached coarse preview state.

If, however, during refinement we reach the finest level L, we are then, by construction, directly minimizing the target finest-level total energy, E, via our quasistatic stepping of the fine-level incremental potential (2). On convergence, we then reach a consistent solution, x_L^t, satisfying the equilibrium, $\|\nabla E(x_L^t)\| < \epsilon$, at the finest level, for parameters u^t. See Algorithm 3 for details.

Stability. As constructed, minimizers x_l, of both preview and refinement time steps are stable forward integrations with implicit

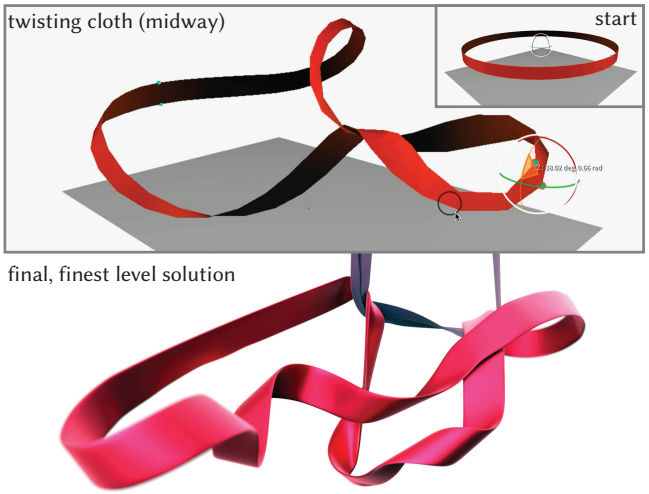

Fig. 8. **Twisted ribbon.** PCS enables handle-based cloth manipulation with contact to interactively twist cloth against itself, rapidly forming complex, tightly wound geometries from simple input shapes.

Algorithm 3 Progressive Simulation

procedure PCS(x_0^{start}, ϵ, step_iters)
  $x_0^t \leftarrow x_0^{\text{start}}$
  do
    $(x_0^{t+1}, \text{eq}) \leftarrow \text{ProxyStep}(x_0^t, x_0^t, \epsilon, u^t, \text{step\_iters})$
    $t \leftarrow t + 1$
  while !eq
  if start_refinement() **then**
    $l \leftarrow 1, x_{l-1} \leftarrow x_0^t$
    do
      $x_l \leftarrow \text{FeasibleSolve}(x_{l-1}^t)$
      $x^p \leftarrow P_l^{l-1} x_{l-1}$
      $(x_l, \text{eq}) \leftarrow \text{ProxyStep}(x_l, x^p, \epsilon, u^t, \text{step\_iters})$
      do
        $(x_l, \text{eq}) \leftarrow \text{ProxyStep}(x_l, x_l, \epsilon, u^t, \text{step\_iters})$
      while !eq and !end_refinement()
      $l \leftarrow l + 1$
    while $l \leq L$

Euler. To clarify this, consider that each solve's optimality gives,

$$x_l = y_l - h^2 M_l^{-1} \nabla C(x_l) - h^2 M_l^{-1} P^{l^T} \nabla G(P^l x_l), \tag{6}$$

which is simply the implicit Euler update of a first-order system. Here this perspective additionally offers the interpretation of membrane and bending energy forces as reduced-order models with the prolongation operator as basis. Steps with $y_l = x_l^t$ are simply, as proposed, forward quasistatic time steps updating from last configuration. Initial refinement steps, with $y_l = P_l^{l-1} x_{l-1}^{t+1}$, are similarly implicit Euler steps updating from level $l-1$ to level l. Our choice of implicit Euler, together with our application of a large time-step size, h, ensures stable, strongly dissipative steps that quickly pull each step towards equilibrium.

Approximate Solves to Equilibria. While it is thus tempting to solve each individual time step solve accurately, aggressively small (even single) numbers of iterates per step maintain stability *and rapid convergence to equilibria* while offering immediate visual feedback at each update on the evolving solution at each level of refinement. Here each displacement applied by Algorithm 2 guarantees decrease in potential energy (via line-searching), preserves non-intersection and strain-limits (via step-size filtering), while inertial energy provides quadratic damping, maintaining consistency across steps. Finally, irrespective of number of iterations per-step applied, convergence to equilibrium is achieved when an applied step converges to tolerance at the start of its first iteration.

Convergence. PCS's finest-level solution is then a fully converged local minimum. We define convergence precisely via the scaled Newton decrement of the finest level's total energy, and therefore the PCS solution satisfies, by construction, the same physical accuracy that would be achieved by simulating the finest-scale model directly. It's also important to clarify that intermediate coarse preview solutions do not simply solve equilibrium of the same physical system on coarser meshes. Instead they solve equilibrium (via Equation 5) of the enriched physical system designed to better approximate the fine model on the coarse mesh.

5 ALGORITHM AND IMPLEMENTATION

5.1 Prolongation

For prolongation operators, P_{l+1}^l we choose a single refinement step of a boundary-modified Loop subdivision that keeps boundary vertices along shell edges unchanged. While, in the long term, we expect improvements via alternative prolongations, in §6 we cover the significant advantages of using Loop refinement over the tempting alternative of direct in-plane upsampling. When rest domains are flat, as standard in draping and paneled cloth simulation, we then trivially construct our hierarchy by repeated application of modified Loop subdivision until we reach our finest-level rest mesh. For shapes with non-flat rest-shape we currently again begin with a coarse starting shape and define the final, high-resolution mesh via repeated modified Loop subdivision. Alternately reverse subdivision [Hassan and Dodgson 2005; Liu et al. 2021] could be applied for curved shells going from coarse to fine. However, this, and investigations of other, alternate prolongation strategies remain next steps for further developing PCS.

5.2 Optimization with Barriers

In order to minimize with C-IPC's barrier energies we require two nonstandard tools within our optimization solvers. First, prior to energy-based evaluations with the barriers we apply proximity detection to *compute constraint sets* that define the non-zero barrier energies for all sufficiently close surface-to-surface primitive pairs. Second, we apply *step-size filtering* to find, via continuous collision detection and strain evaluation, largest possible feasible size steps along prescribed nodal displacement vectors that will ensure strain-limits and nonintersection are maintained.

5.3 Time-Varying Conditions

We drive changes in the quasistatic simulation over times $t, t+1, \ldots$ by variations (encoded in time-varying u) in material properties, boundary conditions (both Dirichlet and collision geometries), external forces, and handles. For details please see §4 in our supplemental.

5.4 User-in-the-loop Interaction

PCS supports a diverse range of offline and one-shot applications where direct simulation and exploration of the above parameters can be efficiently enabled with rapid, predictive previews. In the next section we demonstrate a few of these. However, we also and especially focus on *interactive*, user-in-the-loop parameter exploration and scene manipulation with PCS. To do so we build and demonstrate a prototype interactive tool that allows interactive control of the scene via manipulation, creation, and deletion of handles and collision geometries, as well as updates to material parameters. See Figures 8 and 17 and our supplemental video. Colliders and handles are controlled by transform widgets–we support rotation, scaling (uniform and nonuniform) and translation–to manipulate the cloth drape and setting. Material manipulation is supported by a drop-down of real-world material presets, directly taken from measurement [Penava et al. 2014], and the option to further custom-tune material parameters to refine drape detailing or apply alternate materials.

At the end of each preview mode manipulation we reach a converged equilibrium state. We then cache this coarse solution and proceed with the progressive simulation from that state. At each manipulation we revert to the cached coarse solution, swap in all changed parameters, and then restart the preview mode simulation. This allows the optimization to efficiently warm-start the preview simulation and generally rapidly obtains the new preview solution, allowing PCS to then quickly initiate the progressive simulation steps.

Handle and geometry transformations are applied immediately and asynchronously from the simulation and are then communicated to the simulation as target goals for the BC constraints and penalty energy. They are satisfied incrementally by stepping the underlying preview simulator, which includes both the collision geometry and all cloth simulation domains, towards the target goals. Because PCS (and the underlying C-IPC model) ensures feasibility in each step this allows us to provide both unconstrained live editing of cloth and colliders (thus avoiding user frustration) while also guaranteeing intersection-free (and when desired strain-limited) final state for the drapes at all levels of progressive refinement to ensure high-quality output.

6 EVALUATION

We implement our methods in C++, applying PARDISO [Bollhöfer et al. 2020], compiled with Intel MKL LAPACK and BLAS for linear solves and Eigen for remaining linear algebra routines [Guennebaud et al. 2010]. For robust line-search filtering we evaluate continuous collision detection queries with a spatial-hash culled ACCD [Li et al. 2021].

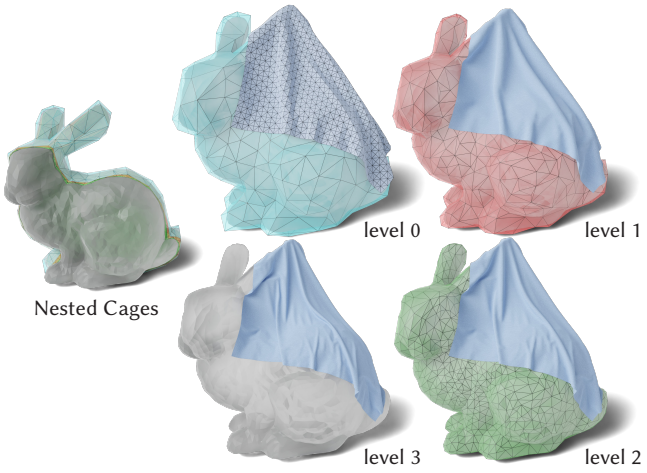

Fig. 9. **Nested cages.** For complex collision geometries we employ Nested Cages [Sacht et al. 2015] that are precomputed and then refined jointly with our cloth simulation meshes. Nesting allows us to safely swap in each finer collision geometry, while preserving safe initialization.

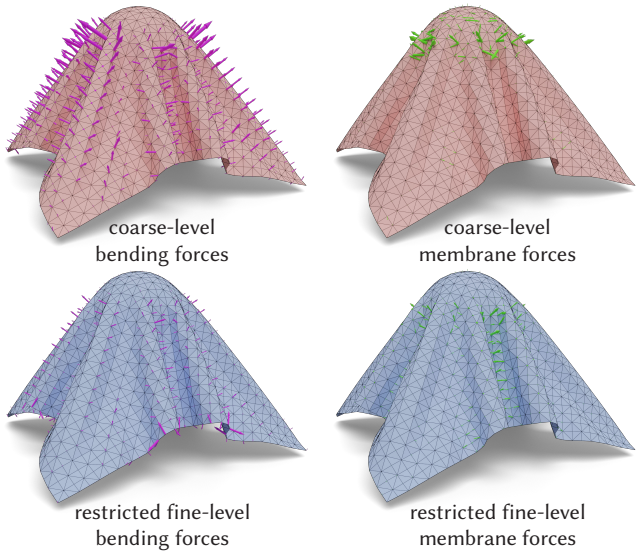

Fig. 10. **Coarse v.s. PCS cloth forces.** Visualization of coarse-level bending and membrane forces using (Top) the coarse-mesh force evaluation versus (Bottom) PCS forces obtained by evaluating fine-scale cloth forces on prolongated geometry followed by subsequent projection back to the coarse scale. The latter avoids ill-scaled values, which can generate locking and numerical artifacts.

6.1 Comparisons

Sensitive Couture. We begin by considering the output of Sensitive Couture's (SC) cascadic refinement. The original SC method applies only a simple penalty-spring-based contact model and solely resolves contacts between cloth and collision objects (neglecting self-contact). In our implementation, for fair side-by-side comparison in terms of timing and capabilities, we have upgraded SC with

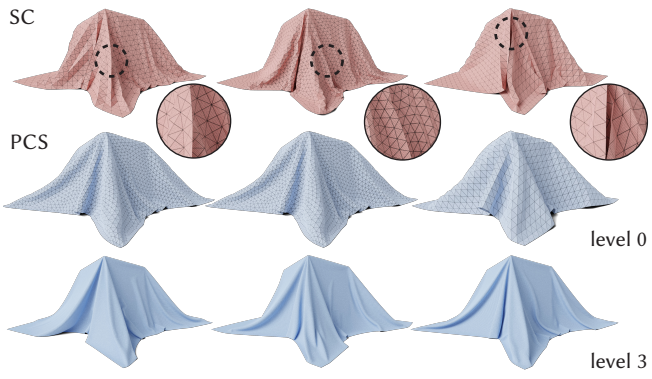

Fig. 11. **Changing base triangulation.** Each column presents a different base-mesh triangulation. SC suffers from creasing artifacts regardless of mesh, while, PCS coarse simulations remain artifact free, and consistent with their converged fine counterparts across different tesselations.

Fig. 12. **Comparison to direct, high-resolution simulation.** (Left, in blue) PCS achieves a high-fidelity, fine-scale solution when draping the (Top) armadillo and (Bottom) bunny. (Right, in orange) Initializing a fine-level Newton solve directly from the coarse (level-0) mesh also achieves a fine-scale solution with comparable numerical fidelity but does not provide the predictive coarse-to-fine deformation consistency of PCS.

the same C-IPC contact model we use for PCS, enabling SC to apply accurate resolution of frictional-contact, self-collision, and, as needed, strain-limiting. In enabling C-IPC modeling (and likewise without), we observe that SC generates significant artifacts across examples with standard cloth material parameters.

Here, for SC, we consistently observe coarse-level simulations exhibiting expected coarse-shell membrane-locking artifacts with exceedingly sharp and kinked edges as well drapes stuck in unnatural fold-overs. These artifacts worsen for thin and stiff materials–standard for cloth materials. Then, as SC passes these converged solutions at each coarser level to the next finer-level solve, these artifacts are passed successively up to finer levels in the hierarchy; see Figures 4 and 11 for representative examples and our supplemental materials for many more. Here we see even the final, fine-level solutions do not escape from the most severe locking artifacts of the coarser SC levels. On the other hand, in other examples we also occasionally (less often) observe that, as SC progresses to finer levels, simply initializing a next Newton solve from the last is insufficient to maintain fold consistency – in these cases we see solutions jump across levels with the larger-scale folds in coarser previews lost altogether, and new ones appearing (see Figure 6). In contrast, in Figures 4 and 6 we observe that, for the same materials and settings, PCS avoids these locking artifacts in both coarse and fine results, while obtaining consistent, larger-scale fold geometries that refine with finer-detail wrinkling and softer folds as we progress in resolution to the finest-level, high-quality result. See also Figure 10 for a direct comparison of the shell forces generated by the standard coarse-level model (as applied by SC) and those generated by PCS's coarse-level model. At the same time, the overall cost of simulating a coarse-level preview with PCS is certainly greater than the direct simulation of the coarse-level model applied by SC and the more standard previsualization approach. Here, however, this cost overhead for PCS over a direct coarse-level simulation is moderate and easily parallelized. Specifically this overhead is just the additional gradient evaluations of membrane and bending energies on the finest-level mesh. This enables interactive previewing and parameter changes (e.g., on a Macbook Pro) even when sampling from highest-level meshes with well over 400K triangles. See below in §6.2 where we analyze the relative cost of this overhead for PCS vs direct coarse-level preview as well as PCS's scaling as its finest-level resolution increases.

Speed-up over Direct, Fine-Mesh Simulation. PCS also enables significant speed-up over well-engineered Newton codes when directly solving for fine-resolution static draping solutions. This significantly expands the application of PCS to additionally include simulations where initial conditions and scene parameters are already pre-detemined and so no parameter exploration is required (See Figure 12). Here PCS's performance to achieve a high-resolution, converged solution of a drape is faster than a simulation of the targeted high-resolution mesh directly solved via Newton, and avoids the numerous issues, inconsistencies and artifacts we see in standard "high-speed" methods (see immediately below for an analysis of industry-standard fast-cloth solvers). Here we consider the drape of a 370K triangle mesh onto the dragon model and a floor, comparing a direct C-IPC solve at high-resolution with a five-level PCS progression. See Figure 13. When we apply C-IPC's standard quasistatic time-stepping [Li et al. 2021] the speed-up for PCS in this example to achieve a converged highest-resolution drape over the direct C-IPC solve is well over two-orders of magnitude. Here then, to get a better baseline we comparably equip the C-IPC quasistatic stepper to take advantage of the same simple, yet highly-effective observation proposed here for PCS: that IPC quasistatic solves can much more effectively be solved by single-iteration time-steps solves (ending with line-search and filter); recall §4.5 for discussion. While this proposed change is simple, it is also, to our knowledge, also entirely new for IPC, and results in significant speed-up for direct IPC-based drape computation. For the above dragon drape experiment, extending this single-iteration stepping for drape solves to IPC obtains an order of magnitude speed-up for C-IPC direct solves. Nevertheless, here the PCS speed-up to the final drape still retains an over 10X speed-up over this modified C-IPC . Here C-IPC converges in 1,248s and PCS in 122s (MacBook M1 Max (64 GB)) with the bulk of PCS's

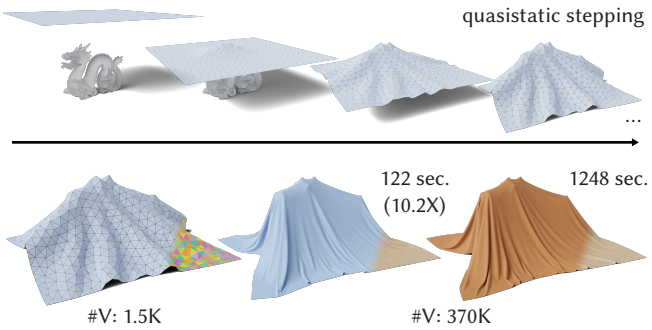

Fig. 13. **Dragon drop test.** PCS progression also significantly speeds up direct "one-shot" simulation tasks when scene parameters are already known. (Blue) a five-level PCS simulation of a cloth drape on a dragon converges to 370K triangle drape solution with a 10X speed-up over a direct simulation solve (Orange) of the high-resolution model.

180 optimization steps taken at coarsest resolution and the number of its steps monotonically decreasing per level down to 15 steps at the finest level. In contrast, C-IPC requires fewer steps total (135) but with significant cost per step. In turn, as we progress to even larger target resolutions and contact numbers, the increasing-cost of high-resolution IPC steps over coarse PCS steps (both dominated by linear solves), the effective speed-up of PCS over C-IPC only grows, while certainly, for more modest-scale scenes, the relative advantage decreases.

Fast Cloth Solvers: Vellum and Marvelous Designer. There is a wide and ever-growing range of fast cloth solvers currently available. Here, as representatives for comparison, we consider two popular industrial solutions, Houdini's Vellum cloth solver [2022] and Marvelous Designer (MD) [2022] as representative cloth solvers for respectively Position-Based Dynamics [Macklin et al. 2016; Müller et al. 2007] and implicit time-integration [Baraff and Witkin 1998; Bridson et al. 2002] methods. We also note that while we do not yet utilize the GPU for PCS, highly optimized commercial cloth-solver solutions like these two do—we thus evaluate Vellum and MD on an Intel i9-9900K @ 3.60GHz with a GTX 2080Ti. As a test case we consider a cloth-on-sphere drop at 1.5K, 25K, and 400K resolutions, simulated with Young's: 8.21e5Pa, Poisson: 0.243, thickness: 3.2e-4m and density: 472.6 kg/m^3. For MD, we are able to apply comparable material parameters and otherwise keep all simulation parameters at default settings. Vellum, on the other hand, does not provide standard moduli, other material parameters, nor units. Instead, following standard artist-practice in Vellum, we use the low-resolution model (obtaining fast simulation) and iterate over Vellum material parameters (otherwise leaving default settings) to find the best visual match to the C-IPC solution for the same. The hand-tuning of both stiffness settings (stretch and bending) and algorithmic collision parameters (iteration counts) to try and match material behavior with the cloth-on-sphere drape is time-consuming. See Figure 2 for visual results and comparison with comparable resolution levels of the PCS solution. For Vellum we see a range of moderate to extreme jagged and irregular folding artifacts, along with small tight non-physical tangles caused by self-intersection. As we scale up towards 400K triangles, not only do we obtain completely different (inconsistent) wrinkle patterns with additional artifacts (unnatural fold pattens, kinking, intersections), we also see that the effective material behavior changes with extreme stretching, which, in turn would require prohibitively expensive multiple high-resolution simulation passes to try and find a set of stiffness and algorithm settings to again best approximate a targeted real-world cloth behavior. For MD results, the story is similar across the first two resolutions. However, as we go to the finest level we also observe large ground-contact instabilities causing the mesh to explode upwards (bouncing artifacts). Experimentation with all exposed contact and collision parameters leaves this behavior unchanged. Finally, we also observe that as we try to model thinner materials (and so soften bending stiffness) the disagreement between Vellum's and MD's coarse and fine models is heightened further.

6.2 Progressive Simulation

Consistency Analysis. As discussed above, the primary failure of SC's cascadic approach leads to unsightly and nonphysical artifacts. On the other hand, the long-standing objection to direct (re-)simulation at high resolution (aside from the attendant slow runtimes also analyzed above) lacks consistency. Similarly, we have, until now, usefully and loosely discussed obtaining wrinkle and fold consistency across resolutions and compared consistency qualitatively. Here we now take a first stab towards formally proposing a geometric measure of this consistency. To do so we measure the total integrated squared difference of a curvature measure between two displacements of the same base domain. Our displacements are represented by meshes at differing subdivision resolutions. We first apply modified Loop subdivision to the coarser of the two until it matches the fine resolution. Then we estimate a curvature measure at vertices. The difference of these per-vertex values is then integrated using the base domain's mass matrix to arrive at our scalar quantity. Here we evaluate this distance measure over a collection of pairings of coarse and fine simulation for PCS and direct simulation, as well as some (rarer) inconsistency cases generated by SC (see e.g., Figure 6), across four curvature measures. Here the results are clear, across all examples and all curvature measures we observe a significant, generally close-to or greater-than an order of magnitude smaller measure for PCS across all examples between coarse and finest for direct simulation as well as across all levels of refinement for SC. Please see our supplemental for details.

Choice of Base Prolongation. We have investigated several prolongation operators. As our default, we use Loop subdivision [Loop 1987] with boundaries fixed for our base prolongation method. This choice is harmonious with using in-plane midpoint upsampling for computing the full, safe prolongation (see §4.3) because both share the same connectivity. Loop subdivision smooths out sharp creases and this is welcome in our cloth simulations as we expect finer levels to either smooth out existing coarse wrinkles or evolve new high-frequency wrinkles not realizable on the coarser level. We fix the boundary so that the parametric domain of the cloth remains fixed, preserving any sharp boundary corners defined at the coarsest level. Forgoing smooth subdivision and using midpoint upsampling directly as prolongation is *safe* in terms of feasibility, but we have found that it leads more frequently to unwanted creases remaining

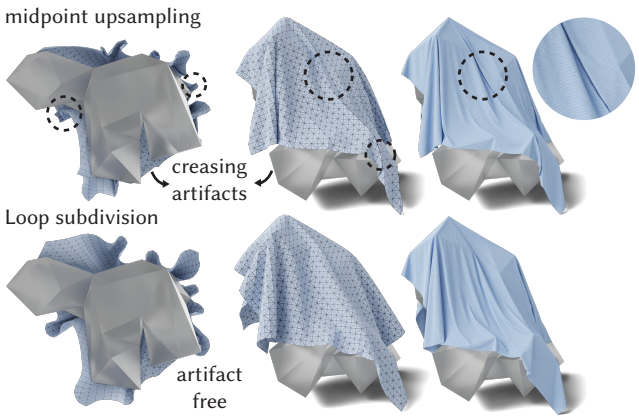

Fig. 14. **In-plane upsampling v.s. Loop for Prolongation.** (Top) Applying in-plane midpoint upsampling as a prolongation operator is *safe* from a feasibility perspective but can lead to coarse creasing artifacts propagating into fine resolution solutions. (Bottom) Using modified, smooth Loop subdivision as a prolongation alleviates this issue.

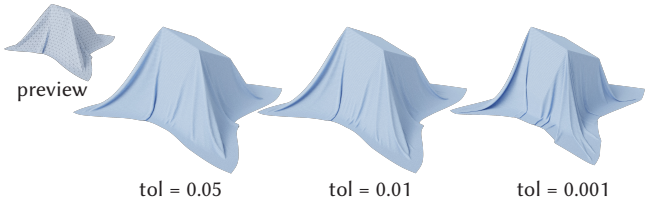

Fig. 15. **Tolerance refinement.** PCS allows users to design and visualize high-fidelity, fine-scale cloth results at lower, draft tolerance (tol) settings, then dial down the error for the very final result. In this example, lower tol values result in even more detailed folds in the fine-scale solution.

even after relaxation on finer levels. In Figure 14, we demonstrate these creasing artifacts and how Loop subdivision alleviates them.

Overhead and Comparative Convergence. PCS minimizes an enriched shell energy to provide consistent coarse-level previewing at the cost of fine-level membrane and bending gradient evaluations. Two questions to consider are how does this affect convergence for these proxy solves, and what is the overhead cost over directly simulating with un-enriched coarse model? We compare the PCS coarse-level solves with corresponding direct solutions of coarse-level energies across drape examples taken from Figures 2, 4, 6 and 14. Here the PCS solutions sample from fine-level meshes ranging from 92-371K triangles. Across this set, we observe that PCS preview simulation on the coarse level is on average 56% slower than direct simulation on the coarse level; however, we also observe that PCS converges a bit faster (on average 14%).

Scaling. A natural follow-up question to then ask is how does timing for PCS's preview stepping change as we increase the resolution of the finest mesh in our hierarchy? Revisiting the dragon-drop test from the last section (see Figure 13), we now consider an increasing range of levels all starting with the same base 1.5K triangle mesh with respectively 3 (23K triangles at finest level), 4 (93K), 5 (371K) and 6 (1.4M) levels. Here, on a MacBook M1 Max (64 GB) we see that timing per step increases respectively from an average of 0.1s with 3 and 4 levels, to 0.14s for 5 levels, and 0.3s for 6 levels.

Material Variations. PCS enables direct and easy exploration for both plug-and-play with real-world reported material parameters as well as direct material parameter manipulation. Here, starting from the otherwise same initial conditions, PCS simulates material settings for silk, denim, and wool. We also simulate a default material, Youngs: 8.21e5 Pa, Poission 0.243, with three different scalings of bending stiffness. As we see in Figure 16 a broad range of material effects are generated with consistent drapes across progressive resolutions for all materials.

Quality. Throughout all our experiments and demos, we apply a default C-IPC [Li et al. 2021] convergence tolerance of 0.01 for termination across all implemented methods–this is consistent for all SC, direct solves via C-IPC, and PCS comparisons. Following Li et al. [2021], we observe that this tolerance is generally a good setting for capturing detailed, high-quality folds and draping in static solutions. That said, PCS allows further solution improvement by decreasing this tolerance. Specifically, this solely increases the cost of the final (generally offline) solve at the finest level. In Figure 15 we illustrate the impact that decreasing this tolerance, solely for the final level convergence, has on the cloth-on-cube drape starting from a larger tolerance of 0.05 where detailed folds that only then appear at a tolerance of 0.01 are missing. Similarly, moving to a more extreme tolerance of 0.001 further enriches with small detail changes. As the final-level preview's tolerance can be decreased without requiring re-simulation at any of the coarser levels, PCS enables users to choose to lower the tolerance for finer quality *after* inspecting the current solution's geometry upon its convergence to the currently specified tolerance.

Interactive Manipulation and Exploration. We have instrumented a series of interactive manipulations with our prototype interactive PCS demonstration. Please see our supplemental videos for live screen captures of these sessions and Figures 1, 7 and 8 for editing, progressive refinements and final geometries.

7 LIMITATIONS AND CONCLUSION

We have presented PCS – a new method for the *progressive simulation* of quasistatic cloth draping. There are many following directions for future improvement and generalization. Most immediately, while we empirically observe that PCS provides significantly better qualitative and quantitative consistency, we have no guarantee nor proof that PCS will do so. For example, there can certainly be challenging configurations, e.g., highly twisted/crumpled cloth, for which coarse triangulations may miss fine-mesh wrinkling behavior. Extending and guaranteeing PCS consistency remain important future avenues of investigation. More broadly, we begin the development of progressive simulation here in the context of quasistatic cloth simulation, which is the first step towards a more general framework for progressive simulation modeling. Extending to shells could be straightforward in theory; however, in practice, designing hierarchies and prolongation operators on curved domains is nontrivial [Liu et al. 2021]. Similar challenges likewise will need to be addressed in future work for extension to volumetric solids. Our

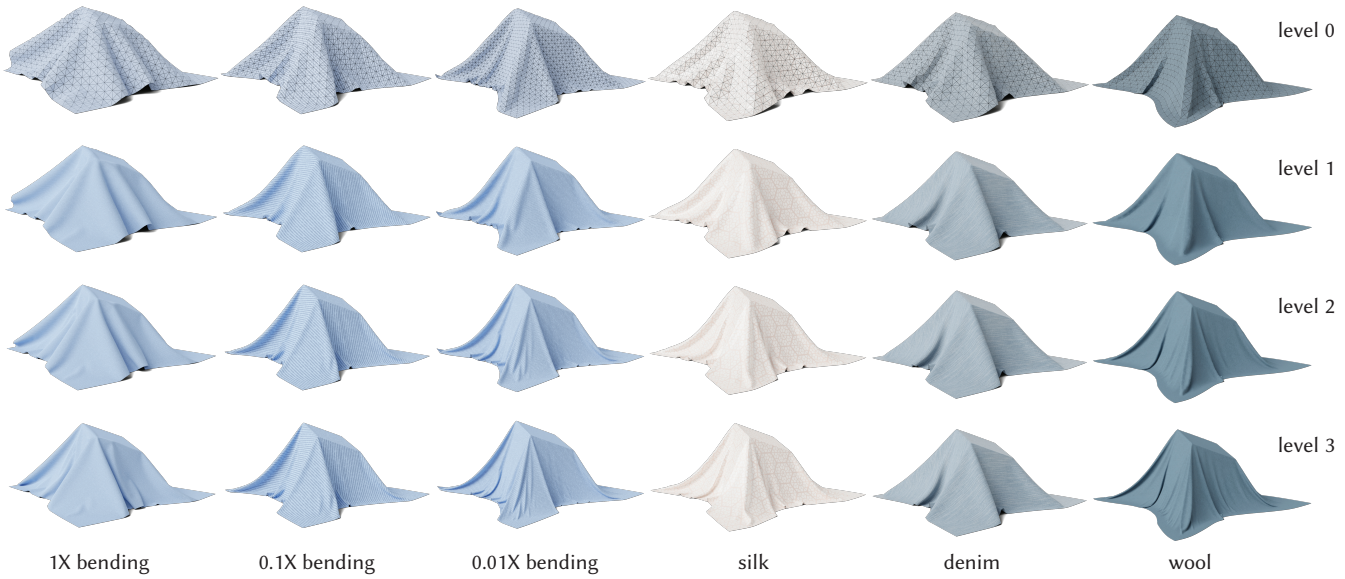

Fig. 16. **Material gallery.** PCS allows users to prototype and rapidly preview the effect of different cloth materials. (Left to Right) Seven different materials are applied to the same draping scene, resulting in a wide range of distinctive wrinkling behaviors that are progressively simulated (Top to Bottom) with increasing resolution previews.

2D domains are determined by the coarse-level geometry, when in practice cloths (e.g., clothing) will often have fine-scale domain boundaries and elaborate seam constraints, which we do not currently handle. Many materials (cloth or otherwise) exhibit not just elastic behavior but also plasticity: this too remains exciting future work. In some ways, the quasistatic scenario we treat here is *more* challenging than dynamics simulation, where momentum helps keep updates between timesteps small. Nevertheless, it is likewise a promising direction to consider how to add dynamics to Progressive simulations, particularly given how performative our approach is compared to vanilla Newton's method. Finally, our progressive simulations are a natural fit for dataset generation in supervised learning of cloth simulation upsampling.

ACKNOWLEDGMENTS

Jiayi Eris Zhang is supported by a Stanford Graduate Fellowship. Alec Jacobson is funded in part by the Canada Research Chairs Program and a Sloan Fellowship.

REFERENCES

Yunfei Bai, Danny M. Kaufman, C. Karen Liu, and Jovan Popović. 2016. Artist-directed dynamics for 2D animation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–10.

David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 43–54.

Jan Bender, Daniel Weber, and Raphael Diziol. 2013. Fast and stable cloth simulation based on multi-resolution shape matching. *Computers & Graphics* 37, 8 (2013), 945–954.

Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. 2007. Tracks: toward directable thin shells. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 50–es.

Matthias Bollhöfer, Olaf Schenk, Radim Janalik, Steve Hamm, and Kiran Gullapalli. 2020. State-of-the-art sparse direct solvers. In *Parallel Algorithms in Computational Science and Engineering*. Springer, 3–33.

Folkmar A Bornemann and Peter Deuflhard. 1996. The cascadic multigrid method for elliptic problems. *Numer. Math.* 75, 2 (1996), 135–152.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)* 33, 4 (2014), 1–11.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. on Graph.* 21 (05 2002).

William L Briggs, Van Emden Henson, and Steve F McCormick. 2000. *A multigrid tutorial*. SIAM.

Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. 2010. A simple geometric model for elastic deformations. *ACM transactions on graphics (TOG)* 29, 4 (2010).

Hsiao-Yu Chen, Arnav Sastry, Wim M van Rees, and Etienne Vouga. 2018. Physical simulation of environmentally induced thin shell deformation. *ACM Trans. Graph. (TOG)* 37, 4 (2018), 1–13.

Zhen Chen, Hsiao-Yu Chen, Danny M. Kaufman, Mélina Skouras, and Etienne Vouga. 2021. Fine Wrinkling on Coarsely Meshed Thin Shells. *ACM Transactions on Graphics (TOG)* 40, 5 (2021), 1–32.

Fehmi Cirak, Michael Ortiz, and Peter Schröder. 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Internat. J. Numer. Methods Engrg.* 47, 12 (2000), 2039–2072.

David Clyde, Joseph Teran, and Rasmus Tamstorf. 2017. Modeling and data-driven parameter estimation for woven fabrics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–11.

Gilles Daviet. 2020. Simple and scalable frictional contacts for thin nodal objects. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 61–1.

Marvelous Designer. 2022. https://www.marvelousdesigner.com

Russell Gillette, Craig Peters, Nicholas Vining, Essex Edwards, and Alla Sheffer. 2015. Real-time dynamic wrinkling of coarse animated cloth. In *Proceedings of the 14th ACM SIGGRAPH/eurographics symposium on computer animation*. 17–26.

Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient simulation of inextensible cloth. In *ACM SIGGRAPH 2007 papers*. 49–es.

Serge Gratton, Annick Sartenaer, and Philippe L Toint. 2008. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization* 19, 1 (2008), 414–444.

Seth Green, George Turkiyyah, and Duane Storti. 2002. Subdivision-based multilevel methods for large scale engineering simulation of thin shells. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*. 265–272.

Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Citeseer, 62–67.

Eitan Grinspun, Petr Krysl, and Peter Schröder. 2002. CHARMS: A simple framework for adaptive simulation. *ACM transactions on graphics (TOG)* 21, 3 (2002), 281–290.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3.

Qi Guo, Xuchen Han, Chuyuan Fu, Theodore Gast, Rasmus Tamstorf, and Joseph Teran. 2018. A material point method for thin shells with frictional contact. *ACM Trans. Graph. (TOG)* 37, 4 (2018), 1–15.

Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, Robert W Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross. 2014. Subspace clothing simulation using adaptive bases. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9.

David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous contact mechanics. In *ACM Trans. on Graph. (TOG)*, Vol. 28. ACM.

David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 papers*. 1–4.

Mohamed F Hassan and Neil A Dodgson. 2005. Reverse subdivision. In *Advances in multiresolution for geometric modelling*. Springer, 271–283.

Van Emden Henson. 2003. Multigrid methods nonlinear problems: an overview. In *Proceedings of SPIE*, Vol. 5016. 36–48.

Chin Pang Ho, Michal Kočvara, and Panos Parpas. 2019. Newton-type multilevel optimization method. *Optimization Methods and Software* (2019), 1–34.

Hugues Hoppe. 1996. Progressive Meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 99–108.

Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans. Graph. (TOG)* 36, 4 (2017).

Ning Jin, Yilin Zhu, Zhenglin Geng, and Ronald Fedkiw. 2020. A Pixel-Based Framework for Data-Driven Clothing. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 135–144.

Ladislav Kavan, Dan Gerszewski, Adam W Bargteil, and Peter-Pike Sloan. 2011. Physics-inspired upsampling for cloth simulation in games. In *ACM SIGGRAPH 2011 papers*. 1–10.

Theodore Kim. 2020. A Finite Element Formulation of Baraff-Witkin cloth. In *Symposium on Computer Animation*.

Alena Kopaničáková, Rolf Krause, and Rasmus Tamstorf. 2019. Subdivision-based nonlinear multiscale cloth simulation. *SIAM Journal on Scientific Computing* 41, 5 (2019), S433–S461.

Zorah Lahner, Daniel Cremers, and Tony Tung. 2018. DeepWrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 667–684.

Cheng Li, Min Tang, Ruofeng Tong, Ming Cai, Jieyi Zhao, and Dinesh Manocha. 2020. P-Cloth: Interactive Cloth Simulation on Multi-GPU Systems using Dynamic Matrix Assembly and Pipelined Implicit Integrators. *ACM Transaction on Graphics (Proceedings of SIGGRAPH Asia)* 39, 6 (December 2020), 180:1–15.

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E Brown, and Laurence Boissieux. 2018. An implicit frictional contact solver for adaptive cloth simulation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–15.

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph.* 40, 4, Article 170 (jul 2021), 24 pages.

Hsueh-Ti Derek Liu, Jiayi Eris Zhang, Mirela Ben-Chen, and Alec Jacobson. 2021. Surface Multigrid via Intrinsic Prolongation. *ACM Trans. Graph.* 40, 4, Article 80 (jul 2021), 13 pages.

Charles Loop. 1987. *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis. University of Utah.

Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective dynamics with dry frictional contact. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 57–1.

Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: Position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*. 49–54.

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A Otaduy, and Steve Marschner. 2012. Data-driven estimation of cloth simulation models. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library.

Matthias Müller and Nuttapong Chentanez. 2010. Wrinkle Meshes.. In *Symposium on Computer Animation*. Madrid, Spain, 85–91.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.

Rahul Narain, Tobias Pfaff, and James F. O'Brien. 2013. Folding and Crumpling Adaptive Sheets. *ACM Trans. Graph.* 32, 4, Article 51 (jul 2013), 8 pages.

Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph.* 31, 6, Article 152 (nov 2012), 10 pages.

Miguel Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus Gross. 2009. Implicit Contact Handling for Deformable Objects. *Comp. Graph. Forum* 28 (04 2009).

Željko Penava, Diana Šimić-Penava, and Ž Knezic. 2014. Determination of the elastic constants of plain woven fabrics by a tensile test in various directions. *Fibres & Textiles in Eastern Europe* (2014).

Victor Romero, Mickaël Ly, Abdullah-Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. 2021. Physical validation of simulators in Computer Graphics: A new framework dedicated to slender elastic structures and frictional contact. *ACM Transactions on Graphics* (2021).

Leonardo Sacht, Etienne Vouga, and Alec Jacobson. 2015. Nested Cages. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–14.

Igor Santesteban, Miguel A Otaduy, and Dan Casas. 2019. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 355–366.

Nikolas Schmitt, Martin Knuth, Jan Bender, and Arjan Kuijper. 2013. Multilevel Cloth Simulation using GPU Surface Sampling. *VRIPHYS* 13 (2013), 1–10.

Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. 2008. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE transactions on visualization and computer graphics* 15, 2 (2008), 339–350.

SideFX. 2022. *Houdini Vellum*. https://www.sidefx.com/products/houdini/

Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*.

Rasmus Tamstorf and Eitan Grinspun. 2013. Discrete bending forces and their Jacobians. *Graphical models* 75, 6 (2013), 362–370.

Rasmus Tamstorf, Toby Jones, and Stephen F McCormick. 2015. Smoothed aggregation multigrid for cloth simulation. *ACM Trans. Graph. (TOG)* 34, 6 (2015), 1–13.

Min Tang, Ruofeng Tong, Rahul Narain, Chang Meng, and Dinesh Manocha. 2013. A GPU-based streaming algorithm for high-resolution cloth simulation. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 21–30.

Min Tang, Huamin Wang, Le Tang, Ruofeng Tong, and Dinesh Manocha. 2016. CAMA: Contact-aware matrix assembly with unified collision handling for GPU-based cloth simulation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 511–521.

Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation. *ACM Transaction on Graphics (Proceedings of SIGGRAPH Asia)* 37, 6 (November 2018), 204:1–10.

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 205–214.

Bernhard Thomaszewski, Markus Wacker, and Wolfgang Straßer. 2005. A consistent bending model for cloth simulation with corotational subdivision finite elements. (2005).

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive Couture for Interactive Garment Modeling and Editing. *ACM Trans. Graph.* 30, 4, Article 90 (jul 2011), 12 pages.

Pascal Volino and N Magnenat Thalmann. 2000. Implementing fast cloth simulation with collision response. In *Proceedings Computer Graphics International 2000*. IEEE.

Etienne Vouga, David Harmon, Rasmus Tamstorf, and Eitan Grinspun. 2011. Asynchronous variational contact mechanics. *CMAME* 200, 25-28 (2011).

Huamin Wang. 2021. GPU-Based Simulation of Cloth Wrinkles at Submillimeter Levels. *ACM Trans. Graph.* 40, 4, Article 169 (jul 2021), 14 pages.

Zhendong Wang, Longhua Wu, Marco Fratarcangeli, Min Tang, and Huamin Wang. 2018. Parallel Multigrid for Nonlinear Cloth Simulation. *Computer Graphics Forum* (2018).

Clarisse Weischedel. 2012. A discrete geometric view on shear-deformable shell models. (2012).

Longhua Wu, Botao Wu, Yin Yang, and Huamin Wang. 2020. A Safe and Fast Repulsion Method for GPU-Based Cloth Self Collisions. *ACM Trans. Graph.* 40, 1, Article 5 (dec 2020), 18 pages.

Zangyueyang Xian, Xin Tong, and Tiantian Liu. 2019. A Scalable Galerkin Multigrid Method for Real-time Simulation of Deformable Objects. *ACM Trans. Graph. (TOG)* 38, 6 (2019).

Juyong Zhang, Yue Peng, Wenqing Ouyang, and Bailin Deng. 2019. Accelerating ADMM for Efficient Simulation and Optimization. *ACM Trans. Graph.* 38, 6, Article 163 (nov 2019), 21 pages.

Javier S Zurdo, Juan P Brito, and Miguel A Otaduy. 2012. Animating wrinkles by example on non-skinned cloth. *IEEE Transactions on Visualization and Computer Graphics* 19, 1 (2012), 149–158.