Simulation Methods for Multiphysics Phenomena in Visual Computing

Fabian Löschner, Stefan Rhys Jeske, José Antonio Fernández-Fernández, and Jan Bender

RWTH Aachen University, Germany

Figure 1: *Complex multiphysics simulations of different materials interacting with each other.* ***Top:*** *(Left) Honey is poured onto a stretched, elastic fabric [LLH*24]. (Right) A snowball fractures into clumps after impact with an obstacle [YLL*24].* ***Bottom:*** *(Left) Two creatures run from a flood wave through highly viscous mud, breaking through a wall of rigid bodies [GPB*19]. (Right) Drifting around a corner, a rigid car with deformable tires as well as suspension and steering modeled by motors and joints is simulated with frictional contact [FFLW*25].*

Abstract

Physics simulation is a cornerstone of many computer graphics applications, ranging from video games and virtual reality to visual effects and computational design. The number of techniques for physically-based modeling and animation has thus skyrocketed over the past few decades, facilitating the simulation of a wide variety of materials and physical phenomena. These course notes provide an in-depth introduction to multiphysics simulation methods for computer graphics, covering the mathematical foundations, key algorithms, and practical considerations behind the most widely used approaches. We focus on methods developed by the computer graphics community for simulating various physical phenomena and materials – including rigid and deformable bodies, fluids, and granular materials – as well as the interactions between them. For each method, we present the underlying mathematical framework with detailed derivations and discuss how different materials and coupling strategies fit into the formulation. A selection of software frameworks that offer out-of-the-box multiphysics modeling capabilities is also presented. Finally, we touch on emerging trends in physics-based animation, including machine learning-based methods which have become increasingly popular in recent years.

Contents

1. Introduction

Physics-based simulation is a key component of computer graphics, powering applications from video games and virtual reality to visual effects and computational design. In this tutorial, we provide an in-depth introduction to multiphysics simulation techniques developed by the graphics community, covering the simultaneous simulation of interacting physical phenomena such as rigid and deformable bodies, fluids, and granular materials (see Figure 1). For each method, we present the underlying mathematical framework with detailed derivations and equations, discuss how different materials and physical phenomena are represented, and show how models can be coupled.

Work in the field can be classified into two fundamental modeling approaches – *unified models* and *coupling techniques*. Unified models treat multiphysics systems with a single discretization across different material types and a monolithic mathematical formulation, which facilitates strong coupling and simplifies implementation. However, this approach may not provide the most suitable model for every material or phenomenon in a given simulation [TLZ*24]. Coupling techniques take a modular design perspective, permitting the use of specialized simulation methods for different materials and phenomena. For example, mesh-free Lagrangian descriptions are beneficial for tracking large deformations, whereas Eulerian representations excel at computing spatial derivatives and enforcing global constraints such as incompressibility. The modular approach enables best-in-class models but can introduce challenges such as additional boundary handling costs and weaker inter-model coupling.

We organize the material covered in this tutorial into four main categories based on the general modeling approach.

- Section 2 presents energy-based models in which materials and physical phenomena are modeled using scalar energy potentials, permitting formulations as unconstrained optimization problems.
- Section 3 introduces constraint-based multiphysics models that use the concept of constraints as the fundamental building block for physically-based animation.
- Section 4 presents Lagrangian methods using mesh-free point-based discretizations.
- Section 5 discusses Eulerian and hybrid simulation methods involving grid-based discretizations.

The distinction between energy-based and constraint-based methods is not always clear cut. For example, constraint-based methods may also rely on strain energies to model deformables while energy-based formulations may account for (soft) constraints using penalty energies. Broadly speaking, we consider a model to be energy-based if its formulation permits application in unconstrained optimization, covering strain energies, barrier potentials, penalty functions, and incremental potentials such as Projective Dynamics. Simulation methods that directly support hard equality or inequality constraints, such as Position Based Dynamics, are covered in Section 3.

Section 6 discusses a selection of multiphysics frameworks that expose combined simulation capabilities in a single platform. Section 7 presents emerging trends in physics-based simulation before we conclude in Section 8.

2. Energy-Based Multiphysics Modeling

Instead of approaching multiphysics systems with a unified discretization, in this section we focus on a unified formulation in terms of scalar energy potentials. If all physical systems can be represented only in terms of such energies, we obtain a strongly coupled, fully implicit formulation that can be solved as a monolithic optimization problem. Each physical effect—elasticity, contact, gravity, and others—contributes an additive term to a single global objective, the *incremental potential*, whose minimizer solves the discrete equations of motion. Casting time integration as optimization rather than root-finding has important practical advantages: it enables globalization techniques such as backtracking line search that make Newton-type solvers substantially more robust, and the

Figure 2: *A selection of energy-based multiphysics simulations with strong coupling.* ***Left:*** *A robotic hand modeled by rigid bodies, joints and motors compresses an elastic cube [FFLL*24].* ***Middle:*** *A scroll modeled as a Cosserat shell is pushed open by rigid bodies through frictional contact [LFFJB24].* ***Right:*** *A crane modeled as a multibody system lifts metallic cubes using magnetic forces that are strongly coupled with frictional contact [WFFJB24].*

additive structure provides a natural interface for multiphysics coupling, since all effects are assembled into and solved through the same objective. This optimization-based time integration approach has a long history [RO99] and is well established in computer graphics [KYT*06, MTGG11, LBOK13, GSS*15]. In recent years, it gained significantly in popularity, for example with the introduction of barrier methods for handling of deformables coupled with frictional contact [LFS*20, LKJ21] and multibody dynamics [FLS*21, CLL*22]. Notably, many constraints that traditionally require explicit enforcement, such as contact non-penetration and joint limits, can be incorporated directly into the unconstrained optimization through penalty or barrier energies. Although an energy-based formulation is well suited for these models, dissipative effects such as damping and friction require adaptations, since they do not derive from a potential in the unknown positions alone, and coupling with fluids or granular media usually has to fall back to operator splitting or weak coupling. Still, energy-based and optimization-based approaches can be applied to a wide range of multiphysics applications as shown in Figure 2.

We want to highlight that a formulation as an unconstrained optimization problem is possible for many phenomena and has advantages in terms of complexity and robustness of the numerical methods (see, e.g., [LFS*20]). However, more complex simulators that seek to combine different physical systems may still choose to combine such energy-based models with hard constraints and apply constrained optimization, for example, to enforce incompressibility in solid-fluid coupling (see, e.g., [TB22]).

In the following, we first lay out the mathematical foundations of optimization-based time integration and its numerical building blocks, including Newton's method, derivative assembly, penalty and barrier constraints, line search, convergence criteria, and linear solves (see Section 2.1). We then present recent works on concrete physical systems and effects as well as their coupling (see Section 2.2). Finally, we explore related methods that are built upon optimization-based integration but aim to improve performance for interactive applications in comparison to a full Newton method, notably Vertex Block Descent (see Section 2.3) and Projective Dynamics (see Section 2.4).

2.1. Foundations of Energy-Based Simulation

In this section, we introduce the foundations of energy-based modeling and optimization-based solvers. This viewpoint is particularly appealing in multiphysics settings because many different physical effects can be expressed in a *unified* way through scalar potentials. Elasticity, gravity, contact, pressure, and other effects can often be modeled by additive energy terms. These contributions are then assembled into a single global objective whose stationary points correspond to the discrete equations of motion. All effects contribute directly to the same objective, and can therefore be solved simultaneously within one nonlinear optimization problem.

2.1.1. Optimization Time Integration

To cast dynamics as an optimization problem, we begin by writing time integration itself in variational form. Consider the equations of motion

$$\mathbf{M}\dot{\mathbf{v}} = \sum_i \mathbf{f}_i(\mathbf{x}, \mathbf{v}), \tag{1}$$

where $\mathbf{M}$ is the global mass matrix, $\dot{\mathbf{v}}$ is the acceleration, and $\mathbf{f}_i$ are the forces acting on the system. These forces may be internal or external, and may in general depend on both positions $\mathbf{x}$ and velocities $\mathbf{v}$.

Let Δt denote the time step, and let $(\mathbf{x}^{\text{prev}}, \mathbf{v}^{\text{prev}})$ and $(\mathbf{x}, \mathbf{v})$ denote the state at the beginning and end of the step, respectively. For simplicity, we first consider Euclidean degrees of freedom, such as nodal positions in a particle system or a deformable body discretization. Using Backward Euler, we write

$$\dot{\mathbf{v}} \approx \frac{\mathbf{v} - \mathbf{v}^{\text{prev}}}{\Delta t}, \qquad \mathbf{v} \approx \frac{\mathbf{x} - \mathbf{x}^{\text{prev}}}{\Delta t}. \tag{2}$$

For non-Euclidean configuration spaces, such as rigid body rotations, the same variational viewpoint still applies, but the kinematic update must be written in a form consistent with the underlying representation. Substituting these expressions into Eq. (1), and assuming for now that the forces are evaluated at the end of the step, yields

$$\mathbf{M}\frac{\mathbf{x} - \mathbf{x}^{\text{prev}} - \Delta t\,\mathbf{v}^{\text{prev}}}{\Delta t^2} - \sum_i \mathbf{f}_i(\mathbf{x}, \mathbf{v}) = \mathbf{0}. \tag{3}$$

A state $\mathbf{x}$ satisfying Eq. (3) is a state that satisfies the discrete balance of forces for the current time step. One may attempt to solve this nonlinear system directly, for instance using Newton's method. However, solving a nonlinear root-finding problem directly can be fragile: Newton steps may overshoot, enter invalid intermediate states, or fail to converge when started far from the solution. These difficulties are particularly pronounced in stiff and strongly nonlinear simulation problems.

For a prominent subset of such systems, however, the same step can be reformulated as a nonlinear optimization problem. Assume that the forces are conservative, so that each force can be written as

$$\mathbf{f}_i(\mathbf{x}) = -\nabla \phi_i(\mathbf{x}), \tag{4}$$

for some potential energy ϕ_i. Then the Backward Euler step is equivalent to minimizing the incremental potential

$$E(\mathbf{x}) = \frac{1}{2\Delta t^2}(\mathbf{x}-\tilde{\mathbf{x}})^T \mathbf{M}(\mathbf{x}-\tilde{\mathbf{x}}) + \sum_i \phi_i(\mathbf{x}), \tag{5}$$

where $\tilde{\mathbf{x}} = \mathbf{x}^{\text{prev}} + \Delta t \mathbf{v}^{\text{prev}}$ is the inertial position. This equivalence follows from the stationarity condition $\nabla E(\mathbf{x}) = \mathbf{0}$, which recovers Eq. (3) for conservative forces.

The time step update can therefore be written as

$$\mathbf{x} = \arg\min_{\mathbf{x}} E(\mathbf{x}), \tag{6}$$

$$\mathbf{v} = \frac{\mathbf{x}-\mathbf{x}^{\text{prev}}}{\Delta t}. \tag{7}$$

This *variational* form is very useful in practice. Rather than solving a nonlinear root-finding problem directly, we can minimize a scalar objective. This immediately enables the use of globalization techniques such as backtracking line search or trust-region strategies [NW06], which make Newton-like methods substantially more robust. In particular, line search enforces sufficient decrease of the objective and prevents overly aggressive steps, which is crucial in the presence of strong nonlinearities such as large deformations or contact.

The energy-based viewpoint also provides a natural interface for multiphysics simulation. Each physical effect contributes an additive term to the same objective, and the total gradient and Hessian are simply the sums of the corresponding per-effect contributions. This leads to a conceptually simple and implementation-friendly framework in which elasticity, inertia, contact, and other effects are handled in a uniform manner.

Unfortunately, not all physically relevant forces can be incorporated directly into this framework. Dissipative or more general non-conservative effects do not, in general, derive from a scalar potential in the unknown state variables. This includes many common models for friction, damping, and plastic flow, and is often reflected by a force Jacobian that is not symmetric. Nevertheless, approaches have been proposed to bring such effects to optimization-based solvers, for example through lagged or split treatments of selected terms.

Although Eq. (5) was derived for Backward Euler, the variational viewpoint is not limited to this integrator. Incremental potential formulations have also been developed for other implicit schemes, including implicit Newmark [KMOW00, LFS*20], the implicit midpoint method [DLK18], and higher-order schemes such as the trapezoidal rule, BDF2, and TR-BDF2 [BOFN18, CLL*22].

2.1.2. Newton's Method

We will use Newton's method to minimize the incremental potential $E(\mathbf{x})$ due to the robustness and fast local convergence it provides. It makes direct use of first- and second-order derivatives and shows great robustness and fast local convergence, which is particularly effective for the stiff nonlinear problems that arise in practice.

At a current iterate $\mathbf{x}$, Newton's method builds the local quadratic model

$$m_k(\Delta\mathbf{x}) = E(\mathbf{x}) + \nabla E(\mathbf{x})^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \nabla^2 E(\mathbf{x}) \Delta\mathbf{x}, \tag{8}$$

and chooses the search direction $\Delta\mathbf{x}$ as its minimizer. This yields the linear system

$$\nabla^2 E(\mathbf{x})\, \Delta\mathbf{x} = -\nabla E(\mathbf{x}). \tag{9}$$

The iterate is then updated as

$$\mathbf{x}^{(k+1)} = \mathbf{x} + \alpha \Delta\mathbf{x}, \tag{10}$$

where $\alpha \in (0,1]$ is typically chosen by line search.

From the viewpoint of mechanics, the gradient $\nabla E(\mathbf{x})$ is the residual force of the current state. A stationary point satisfies $\nabla E(\mathbf{x}) = \mathbf{0}$, which is precisely the condition of discrete force balance. The Hessian $\nabla^2 E(\mathbf{x})$ is the local stiffness, that is, the linearization of the residual forces with respect to the degrees of freedom. Newton's method can therefore be interpreted as repeatedly solving a linearized equilibrium problem around the current iterate. It converges in one iteration for a quadratic objective, and typically converges quadratically once sufficiently close to a regular minimizer.

The quality of Newton's method depends strongly on the regularity of the objective. At a minimum, C^1 continuity is required so that the gradient is well defined. In practice, C^2 continuity is desirable, since Newton's method relies on the Hessian to build a local quadratic model. If the objective is discontinuous, optimization progress may stall entirely, and line search may get stuck at the discontinuity. The construction of smooth and well-behaved energies is a central theme in simulation.

Through this document, we will visit important aspects of this optimization process: the assembly of gradients and Hessians, Hessian projection to positive definiteness, line search conditions, convergence criteria, and inexact linear solves. The overall procedure is summarized in Algorithm 1.

2.1.3. Global Derivatives and Assembly

The global potential is usually an aggregation of different potential energies *sources*, each of which is itself a sum of many *local* contributions:

$$E(\mathbf{x}) = \sum_{p\in\mathcal{P}} \sum_{e\in\mathcal{T}_p} \phi_p(\mathbf{x}_e). \tag{11}$$

Here, $\mathcal{P}$ denotes the different classes of potentials, such as inertia, elasticity, or contact, and $\mathcal{T}_p$ is the corresponding set of local contributions. The vector $\mathbf{x}_e$ contains the subset of degrees of freedom

Algorithm 1 Newton's method for optimization of the Backward Euler incremental potential.

1: $\mathbf{x} \leftarrow \tilde{\mathbf{x}} = \mathbf{x}^{\text{prev}} + \Delta t \mathbf{v}^{\text{prev}}$ ▷ initialize with inertial position
2: **while** not converged **do**
3: assemble gradient $\nabla E(\mathbf{x})$ and Hessian $\nabla^2 E(\mathbf{x})$ ▷ Sec. 2.1.3
4: solve $\nabla^2 E(\mathbf{x}) \Delta \mathbf{x} = -\nabla E(\mathbf{x})$ ▷ Eq. (9)
5: find step size α by line search ▷ Sec. 2.1.6
6: $\mathbf{x} \leftarrow \mathbf{x} + \alpha \Delta \mathbf{x}$
7: **end while**
8: $\mathbf{v} \leftarrow (\mathbf{x} - \mathbf{x}^{\text{prev}})/\Delta t$

associated with contribution e. Since each local contribution only depends on a small part of the full unknown vector $\mathbf{x}$, we may write

$$\mathbf{x}_e = \mathbf{P}_e \mathbf{x}, \tag{12}$$

where $\mathbf{P}_e$ is a selection matrix that extracts the relevant entries from the global vector.

For example, in a tetrahedral discretization of an elastic body, the total strain energy is the sum of the strain energies of the individual tetrahedra,

$$E_{\text{strain}}(\mathbf{x}) = \sum_{e \in \mathcal{T}_{\text{strain}}} \phi_{\text{strain}}(\mathbf{x}_e), \tag{13}$$

where $\mathcal{T}_{\text{strain}}$ is the set of tetrahedra and $\mathbf{x}_e$ contains the degrees of freedom of tetrahedron e.

A single simulation problem may also contain multiple sets of degrees of freedom. This is especially common in multiphysics settings, where, for example, rigid and deformable bodies may be solved together, even though they may have different representations of state. To keep the notation simple, we will continue to write the global unknown as $\mathbf{x}$, even though it refers to the stacked vector of all degrees of freedom

$$\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n)^T. \tag{14}$$

Although the optimization problem is global, each local contribution only "sees" a few unknowns. The global coupling arises because different local terms may depend on overlapping degrees of freedom: for example, two neighboring tetrahedra share nodes, and those same nodes may also appear in contact primitives during a collision. This overlap is reflected in the derivatives: local force contributions accumulate in the global gradient, and local stiffness contributions accumulate in the global Hessian. Consequently, physical effects such as elasticity and contact are solved in a fully coupled manner through the same global minimization problem.

2.1.4. Local-to-global derivatives.

Since the global potential is an aggregation, its derivatives are the sums of the derivatives of the individual local contributions:

$$\nabla E(\mathbf{x}) = \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{T}_p} \nabla \phi_p(\mathbf{x}_e), \tag{15}$$

$$\nabla^2 E(\mathbf{x}) = \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{T}_p} \nabla^2 \phi_p(\mathbf{x}_e). \tag{16}$$

However, these expressions must be interpreted carefully: each local gradient and Hessian act only on the subset of degrees of freedom associated with their corresponding contribution. For example, a linear tetrahedron with four nodes contributes a local gradient of size 12×1 and a local Hessian of size 12×12. Another tetrahedron produces derivatives of the same size, but with respect to a different subset of vertices. The process of inserting these local derivatives into the global gradient and Hessian according to their local-to-global mapping is called *assembly*.

Using the selection matrix notation from Eq. (12), this mapping can be written compactly through the chain rule as

$$\nabla E(\mathbf{x}) = \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{T}_p} \mathbf{P}_e^T \nabla \phi_p(\mathbf{x}_e), \tag{17}$$

$$\nabla^2 E(\mathbf{x}) = \sum_{p \in \mathcal{P}} \sum_{e \in \mathcal{T}_p} \mathbf{P}_e^T \nabla^2 \phi_p(\mathbf{x}_e) \mathbf{P}_e. \tag{18}$$

Local element derivatives are scattered into the global derivatives. Since most local potentials only involve a small number of unknowns, the global Hessian is sparse, that is, most of its entries are zero. Moreover, the sparsity pattern is not always fixed: dynamic contact, changing constraints, or adaptive remeshing may modify the connectivity of the problem over time.

When multiple sets of degrees of freedom are present in multiphysics simulation, the global derivatives naturally acquire a *superblock* structure:

$$\nabla E(\mathbf{x}) = \begin{pmatrix} \nabla_{\mathbf{x}_1} E \\ \nabla_{\mathbf{x}_2} E \\ \vdots \\ \nabla_{\mathbf{x}_n} E \end{pmatrix}, \tag{19}$$

$$\nabla^2 E(\mathbf{x}) = \begin{pmatrix} \nabla^2_{\mathbf{x}_1 \mathbf{x}_1} E & \nabla^2_{\mathbf{x}_1 \mathbf{x}_2} E & \cdots & \nabla^2_{\mathbf{x}_1 \mathbf{x}_n} E \\ \nabla^2_{\mathbf{x}_2 \mathbf{x}_1} E & \nabla^2_{\mathbf{x}_2 \mathbf{x}_2} E & \cdots & \nabla^2_{\mathbf{x}_2 \mathbf{x}_n} E \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^2_{\mathbf{x}_n \mathbf{x}_1} E & \nabla^2_{\mathbf{x}_n \mathbf{x}_2} E & \cdots & \nabla^2_{\mathbf{x}_n \mathbf{x}_n} E \end{pmatrix}. \tag{20}$$

Contributions that depend only on one set of degrees of freedom describe energy variations internal to a single subsystem. These include, for example, inertial or elastic terms, as well as contact terms internal to one subsystem. Off-diagonal Hessian blocks, on the other hand, describe couplings between different subsystems, such as contact, attachments, or fluid–structure interaction.

2.1.5. Penalty Constraints

Within an energy-based formulation, constraints are often incorporated through additional *penalty* or *barrier* terms. Common examples include contact, strain-limiting, joint constraints and joint limits in rigid-body systems, and prescribed boundary conditions. This preserves the unified optimization structure introduced above: constraints contribute to the same global potential and are handled by the same Newton solve.

Let $C(\mathbf{x})$ be a scalar constraint function such that $C(\mathbf{x}) = 0$ corresponds to exact satisfaction of the constraint. A penalty energy can then be defined as

$$E_c(\mathbf{x}) = \frac{k_c}{2} C(\mathbf{x})^2, \tag{21}$$

where $k_c > 0$ is the penalty stiffness. Its gradient produces a restoring force that increases with the amount of violation,

$$\mathbf{f}_c = -\nabla E_c(\mathbf{x}) = -k_c C(\mathbf{x}) \nabla C(\mathbf{x}). \tag{22}$$

Thus, the larger the constraint violation, the stronger the corrective force.

For inequality constraints, however, corrective forces should act only when the constraint is violated. If the feasible set is defined by $C(\mathbf{x}) \geq 0$, the corresponding penalty may be written as

$$E_c(\mathbf{x}) = \begin{cases} 0, & C(\mathbf{x}) \geq 0, \\ -\frac{k_c}{3} C(\mathbf{x})^3, & C(\mathbf{x}) < 0. \end{cases} \tag{23}$$

A cubic term is used here because it provides a C^2-continuous transition at activation, which is generally beneficial for the convergence of Newton's method.

In some cases, the constraint defines an additional *hard limit* of the admissible state, as in contact with non-penetration guarantees. For such constraints, logarithmic barrier energies are often used, since they can produce arbitrarily large corrective forces as the constraint approaches the boundary of admissibility, at the price of increased nonlinearity.

2.1.6. Line Search

Line search is central to the success of Newton's method in complex multiphysics problems, particularly in the presence of contact. Newton's method provides a search direction $\Delta\mathbf{x}$ based solely on local information at the current iterate $\mathbf{x}$. This direction may therefore be unaware of any nearby but currently inactive stiff barrier constraint, and may point directly across it. In such a case, the full Newton step is clearly unacceptable. More generally, far from the minimizer, the local quadratic model may simply be too inaccurate for the full step to be reliable. In all such situations, taking the full step may lead to overshooting or even complete solver failure.

For this reason, Newton's method is almost always combined with a line search, which replaces the full update by

$$\mathbf{x}^{(k+1)} = \mathbf{x} + \alpha\Delta\mathbf{x}, \tag{24}$$

where the step length $\alpha \in (0, 1]$ is chosen to satisfy suitable acceptance conditions.

The first requirement is usually *sufficient decrease*. Given a descent direction $\Delta\mathbf{x}$, one seeks a step length such that

$$E(\mathbf{x} + \alpha\Delta\mathbf{x}) \leq E(\mathbf{x}) + c_1 \alpha \nabla E(\mathbf{x})^T \Delta\mathbf{x}, \tag{25}$$

where $c_1 \in (0, 1)$ is a small constant. This is the classical *Armijo condition* [NW06]. It ensures that the accepted step decreases the objective by an amount commensurate with the predicted local descent, and therefore prevents overly aggressive updates.

A standard strategy to enforce this condition is *backtracking*. One starts with $\alpha = 1$, corresponding to the full Newton step, and repeatedly reduces it, typically by a constant factor $\rho \in (0, 1)$, until the acceptance condition is satisfied. This mechanism retracts a step from regions of very large penalty or barrier growth, such as a step deep into a contact potential.

In physical simulation, however, decrease of the objective is not the only concern. The iterates must also remain *admissible*. Depending on the application, this may exclude inverted finite elements, penetrations, or other invalid configurations. In contact simulation, this is commonly obtained using continuous collision detection (CCD) to truncate the step in order to keep it admissible. A simpler alternative is to test admissibility during backtracking itself, for example by testing for intersection-freeness, although this does not prevent tunnelling.

Finally, for very stiff problems, line search may become numerically delicate near convergence, since the change in total energy can approach floating-point accuracy. In this regime, direct evaluation of the Armijo condition may become noisy and reject otherwise acceptable steps. A standard remedy is to complement the direct energy comparison by a local Taylor approximation; see [LLFF*23].

2.1.7. Convergence and Inexactness

In computer graphics applications, Newton's method is rarely driven to very high accuracy. Instead, one usually accepts an *inexact* solution once the remaining error is sufficiently small for the purpose of the simulation. This is an important design choice: solving too accurately increases computational cost with little visible benefit, while terminating too early may introduce excessive numerical dissipation, poor force balance, and mechanically implausible behavior. Moreover, the convergence criterion itself should remain meaningful across disparate materials and interactions, especially in multiphysics settings.

Validity-preserving mechanisms such as IPC make early termination substantially safer. A search terminated before full convergence may still exhibit large force imbalances, but it remains collision-free and is therefore generally admissible as the starting point for the next time step.

Common stopping criteria include the following:

- **Residual norm.** Terminate when
$$\|\nabla E(\mathbf{x})\| < \varepsilon_{\text{res}}.$$
This is the standard criterion in nonlinear optimization, since the gradient measures the remaining force imbalance.
- **Newton step norm.** Terminate when
$$\|\Delta\mathbf{x}\| < \varepsilon_{\text{step}}.$$
This measures the size of the update and therefore how much the current iterate is still changing geometrically.
- **Fixed iterations.** Terminate after a prescribed number of Newton iterations.

From a mechanical viewpoint, the residual norm is the most direct measure of convergence, since it quantifies the remaining force imbalance. However, for inexact solves, it is often difficult to choose a coarse residual tolerance that both correlates with visual outcome and remains meaningful across very different stiffness scales. Consider, for example, a dryer machine rotating a piece of cloth inside. A residual force of a given magnitude may correspond to a negligible geometric error in the stiff and heavy rigid components, but to a large visible deformation in the cloth.

In this setting, the Newton step norm is often more practical because it measures the size of the correction being applied. In fact, the Newton step is quite literally the residual weighted by the inverse of mass and stiffness (see Eq. 9). This relative scaling of the force residual to the local stiffness makes it more consistent across mixed-material and multiphysics settings. For better temporal consistency, the tolerance is often scaled with the time step:

$$\|\Delta \mathbf{x}\| < \Delta t\, \varepsilon_{\text{vel}}.$$

Note that neither the residual nor the step norms are guaranteed to decrease monotonically during a Newton search. Nevertheless, it is usually reasonable to expect the accepted steps to become progressively smaller as the iterates approach stationarity.

Finally, while terminating after a fixed number of iterations provides predictable runtime, it is not tied to any direct measure of convergence or mechanical accuracy.

The appropriate choice ultimately depends on the goals of the application, but both the stopping criterion and its tolerance should be chosen deliberately. Stopping too late wastes work, while stopping too early reduces accuracy. Consistency across models, discretizations, time steps, and parameter choices should therefore be considered carefully.

2.1.8. Projection to Positive Definite

A central requirement for Newton's method is that the search direction be a *descent direction*. Recall the Newton system

$$\nabla^2 E\, \Delta \mathbf{x} = -\nabla E. \tag{26}$$

If the Hessian is symmetric positive definite (SPD) at the current iterate, that is,

$$\mathbf{r}^T \nabla^2 E\, \mathbf{r} > 0 \qquad \forall \mathbf{r} \neq \mathbf{0}, \tag{27}$$

then the local quadratic model is strictly convex and the Newton direction is guaranteed to be a descent direction. Indeed, multiplying the Newton equation by $\Delta \mathbf{x}^T$ gives

$$\Delta \mathbf{x}^T \nabla E(\mathbf{x}) = -\Delta \mathbf{x}^T \nabla^2 E(\mathbf{x}) \Delta \mathbf{x}, \tag{28}$$

implying

$$\Delta \mathbf{x}^T \nabla E(\mathbf{x}) < 0. \tag{29}$$

Thus, the Newton step is aligned with the negative gradient and can be used safely within a line search.

In practice, however, Hessians arising in physical simulation are not always positive definite and therefore will generally not guarantee descend directions. For this reason, many simulation methods replace the true Hessian by a positive definite approximation, thereby guaranteeing descent while remaining as faithful as possible to the original model. This process is commonly referred to as *projection to positive definite*.

A standard approach is to filter, for example by clamping, the negative eigenvalues of the element Hessians prior to assembly. Consider the eigendecomposition of the Hessian of element e,

$$\mathbf{H}^e = \mathbf{Q}^e \mathbf{D}^e \left(\mathbf{Q}^e\right)^T, \tag{30}$$

where the columns of $\mathbf{Q}^e$ are the eigenvectors of $\mathbf{H}^e$ and $\mathbf{D}^e$ is a diagonal matrix containing the corresponding eigenvalues. Applying clamping, the projected element Hessian is

$$\widehat{\mathbf{H}}^e = \mathbf{Q}^e \widehat{\mathbf{D}}^e \left(\mathbf{Q}^e\right)^T, \tag{31}$$

with

$$\widehat{D}^e_{ii} = \max(D^e_{ii}, \varepsilon), \tag{32}$$

for some $\varepsilon > 0$. Since a sum of SPD matrices is SPD, the resulting assembled global matrix is also SPD. This approach is commonly referred to as Projected Newton [TSIF05].

In some cases, however, projecting all element Hessians at every iteration introduces unnecessary distortion of the global Hessian and adds computational overhead. This is particularly relevant in problems with natural regularization, for example in dynamics, where the inertial term already contributes positive curvature. In such settings, it can be beneficial to apply projection only when needed. In particular, Project-On-Demand Newton [LLFF*23] projects local Hessians only when the assembled global Hessian is detected to be indefinite, and Progressively Projected Newton [FFLB26] selectively projects contributions in regions more likely to induce indefiniteness.

Beyond clamping negative eigenvalues, other spectral modifications have also been explored. A common alternative is *eigenvalue mirroring*, in which negative eigenvalues are reflected to positive values by an absolute-value operation rather than clamped to a floor [CLL*24]. Related approaches interpolate between clamping and mirroring in order to balance robustness and fidelity to the original curvature [CLF25, CLJ*24]. These variants have shown strong results in simulations of near-incompressible materials undergoing extreme deformation.

A related but conceptually distinct approach is to avoid the true Hessian altogether and instead build a positive semidefinite approximation from the outset. The most common example is the Gauss–Newton method, which arises when the objective is written as a sum of squared residuals. Its Hessian approximation discards second-order residual terms and retains only a matrix of the form $\mathbf{J}^T\mathbf{J}$, which is positive semidefinite by construction.

2.1.9. Linear System Solves

Once the gradient and Hessian have been assembled, each Newton iteration requires the solution of the linear system in Eq. (9). In large-scale simulation, the assembly of the Hessian and the solution of the linear system often account for a large fraction of the total runtime. Equally important, the linear solve is usually *inexact*: in practice, one does not seek a highly accurate solution of the Newton system at every iteration, but rather a level of accuracy commensurate with the current quality of the quadratic model.

As discussed in Section 2.1.3, the assembled Hessian is typically very sparse, and its sparsity pattern may change over time due to contact activation, remeshing, or changing constraints. Practical implementations therefore rely on sparse formats that support efficient assembly and matrix–vector products. Popular choices include Blocked Compressed Row Storage (BCRS), where each nonzero stores a small dense block such as 3×3 in three-dimensional nodal discretizations, and ELL-type formats, which

are attractive for their regular memory access and parallel efficiency. In all cases, assembly and sparse linear algebra are usually memory-bound, making data layout and access critical for performance.

When the Hessian has been made positive definite, as discussed in the previous section, one may use solvers designed for SPD systems. Direct factorizations such as Cholesky or LDLT are robust and accurate, and are often attractive for small problems or for simulations so stiff that iterative methods become ineffective. When iterative solvers are applicable, the standard choice is the Preconditioned Conjugate Gradient (PCG) method, equipped with a good preconditioner. At a minimum, diagonal or block-diagonal preconditioners should be used; stronger alternatives such as incomplete factorizations may be worthwhile when their setup cost is justified.

Inexact Newton Solves A key practical advantage of iterative linear solvers is that they support inexact Newton methods naturally. Far from the minimizer, solving the Newton system to very tight tolerances is usually wasteful, since the quadratic model is itself the limiting factor for progress. It is then preferable to compute a larger number of cheaper Newton steps. Near the solution, however, more accurate solves are desirable in order to recover fast local convergence.

This strategy is commonly described in terms of a *forcing sequence*, which tightens the accuracy of the inner linear solve as the outer Newton iteration progresses. A common practical surrogate is to terminate CG once its relative residual has been reduced by a fixed factor, e.g.

$$\|\mathbf{r}_k\| \leq \eta \, \|\mathbf{r}_0\|, \tag{33}$$

with $\eta \approx 10^{-4}$. This is simple and effective in practice, and often provides the desired behavior across Newton iterations. Strictly speaking, however, it is not a forcing sequence in the classical sense, where the inner tolerance is tied explicitly to the nonlinear Newton residual rather than fixed a priori. Nevertheless, this relative-residual criterion is widely used and it provides a good balance between efficiency and convergence in simulation.

In summary, the linear system solve should not be viewed as an isolated numerical subroutine. It is tightly coupled to sparse representation, assembly, Hessian definiteness, preconditioning, and the inexactness strategy of the outer Newton method. A good implementation balances all of these aspects simultaneously.

2.1.10. Differentiability

The optimization-based simulation framework described above requires a certain degree of differentiability, which is likewise needed in many inverse design and parameter identification applications. When the underlying models and solvers are sufficiently smooth, one can compute sensitivities of the simulation output with respect to its parametrization. This makes it possible not only to simulate a system for fixed parameters, but also to adjust those parameters so that the simulation matches a desired outcome. Here we briefly state the main technique used to compute such sensitivities, and refer the reader to the course notes by Coros et al. [CMTT21] for a more complete treatment.

Let $\theta \in \mathbb{R}^N$ denote a vector of parameters, and let $\mathbf{x}(\theta)$ denote the converged simulation state obtained by minimizing the incremental potential. We define a loss $\mathcal{L}(\mathbf{x}(\theta), \theta) \in \mathbb{R}$ and seek to solve

$$\theta = \operatorname{argmin}_\theta \mathcal{L}(\mathbf{x}(\theta), \theta). \tag{34}$$

To use gradient-based optimization methods, we need the total derivative $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta}$. Differentiating every operation inside Newton's method is possible in principle, for example by automatic differentiation, but is often impractical for large-scale problems. Instead, one typically applies the *Implicit Function Theorem*.

We have already established that the solution of the nonlinear mechanical problem satisfies

$$\nabla_{\mathbf{x}} E(\mathbf{x}, \theta) = \mathbf{0}. \tag{35}$$

Then the total derivative of the loss can be written as

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta} = \frac{\partial \mathcal{L}}{\partial \theta} + \lambda^T \frac{\partial^2 E}{\partial \mathbf{x} \partial \theta}, \quad \text{for} \quad \left(\frac{\partial^2 E}{\partial \mathbf{x}^2}\right)^T \lambda = -\left(\frac{\partial \mathcal{L}}{\partial \mathbf{x}}\right)^T, \tag{36}$$

where λ is the adjoint state. This expression connects the simulation degrees of freedom $\mathbf{x}$ and the parameters θ through mixed derivatives of the physical model and the loss. Importantly, evaluating Eq. (36) requires only one additional linear solve after the forward simulation has converged, regardless of how many Newton iterations were needed.

Finally, Eq. (36) describes the sensitivity of the solution of a single nonlinear solve. In dynamic simulation, however, we must solve a sequence of such problems. The corresponding sensitivities must then be propagated through that sequence, which leads to a sequence of adjoint computations, typically applied in reverse time order.

2.1.11. Frameworks

We have seen that optimization time integrators provide a convenient way to unify complex multiphysics problems through a simple mathematical interface: a global scalar objective and a collection of degrees of freedom. Their implementation, however, can still be highly involved. A complete system may require many local potentials, several coupled sets of degrees of freedom, first- and second-order derivatives, sparse assembly, positive-definite projection, line search, and robust, efficient linear solves. Implementing, optimizing, testing, and integrating all these components into a correct and efficient solver is a substantial engineering task. Several frameworks have therefore been proposed to facilitate the construction of such systems by taking over large parts of the differentiation and solution process from a concise problem definition.

Representative examples include TinyAD [SBB*22], SymX [FFLW*25], and the mesh-based simulation framework of Herholz et al. [HSK24]. These may be viewed as lower-level mathematical engines on top of which a multiphysics solver can be built. TinyAD is a lightweight C++ library for first- and second-order differentiation of sparse mesh-based problems, and is particularly attractive for its simplicity and ease of integration. It was introduced in the context of geometry processing and provides simple solver facilities while relying heavily on the Eigen library. SymX targets energy-based simulation more directly, and provides a more extensive infrastructure for the ingredients discussed

throughout this tutorial. The authors report strong performance on complex nonlinear and multiphysics problems, including higher-order FEM, rigid bodies, and frictional contact. Herholz et al. [HSK24] pursue a similar goal, although their system is not publicly released.

At a higher level, there are also full multiphysics solvers available. STARK [FFLL*24] is built on top of SymX, thereby removing differentiation from the manual implementation pipeline. It provides ready-to-use models for FEM, rigid bodies, and frictional contact. As such, it can serve as a useful starting point for researchers and practitioners who wish to develop new models within an already integrated simulation environment.

Alternatively, the IPC Toolkit [Fo20] provides a modular C++ implementation of the core IPC building blocks that will be introduced in the following section, including barrier potentials, friction, and continuous collision detection, and can be integrated into existing simulation pipelines. PolyFEM [SDG*19] is a finite element library that supports a wide range of element types and formulations, including higher-order bases and mixed methods, and can be combined with IPC Toolkit.

2.2. Energy-Based Materials and Phenomena

Any materials or physical effects that admit a formulation in terms of an energy or scalar potential can be easily combined to obtain a strongly coupled multiphysics simulator by summing their contributions and any interaction potentials in the global incremental potential from Equation (5). As a result, these models benefit from the numerical methods and their stability as outlined before. In the following, we give a brief overview of recent works that use this approach to model specific effects or to perform coupled simulations.

2.2.1. Elastic Deformables

The simulation of elastic deformables naturally fits into an optimization-based framework. Continuous material models for deformable bodies are typically defined in terms of a scalar strain energy density function $\Psi(\mathbf{F})$ depending on the local deformation gradient $\mathbf{F}$. The total strain energy of a body occupying reference domain Ω_0 is obtained by integrating this density over the volume,

$$\phi_{\text{strain}}(\mathbf{x}) = \int_{\Omega_0} \Psi\big(\mathbf{F}(\mathbf{x},\mathbf{X})\big)\,\mathrm{d}\mathbf{X}, \tag{37}$$

where $\mathbf{X}$ denotes material (reference) coordinates. This strain energy can be added directly as one of the potential terms ϕ_i in the incremental potential of Equation (5).

To evaluate this integral numerically, a spatial discretization is required. Using the Finite Element Method (FEM) with a Lagrangian view, the domain is partitioned into elements (e.g. tetrahedra in 3D), and the integral is approximated using numerical quadrature,

$$\phi_{\text{strain}}(\mathbf{x}) \approx \sum_{e\in\mathcal{T}_{\text{strain}}} \sum_{q=1}^{Q_e} w_{eq}\,\Psi\big(\mathbf{F}(\mathbf{x}_e,\mathbf{X}_{eq})\big), \tag{38}$$

where Q_e is the number of quadrature points in element e, w_{eq} are the corresponding quadrature weights, $\mathbf{X}_{eq}$ are the quadrature point locations in the reference configuration, and $\mathbf{x}_e = \mathbf{P}_e\mathbf{x}$ are the nodal positions of element e extracted via the selection operator from Eq. (12). The deformation gradient $\mathbf{F}(\mathbf{x}_e,\mathbf{X}_{eq})$ at each quadrature point is computed from the element's nodal positions using the shape function derivatives evaluated at $\mathbf{X}_{eq}$. For higher-order elements with non-constant deformation gradients, multiple quadrature points per element are necessary to accurately capture the variation of the strain energy density across the element.

Different element geometries can be used in practice, such as tetrahedra, hexahedra or even polyhedra [WBG07, BHSH*22]. However, in graphics the most common choice is linear tetrahedra, where the shape functions are linear and the deformation gradient is therefore constant within each element. A single quadrature point per element (with weight $w_e = V_e^0$, the reference volume) is then sufficient, and Eq. (38) simplifies to

$$\phi_{\text{strain}}(\mathbf{x}) \approx \sum_{e\in\mathcal{T}_{\text{strain}}} V_e^0\,\Psi\big(\mathbf{F}_e(\mathbf{x}_e)\big). \tag{39}$$

For a linear tetrahedron with nodes $\mathbf{x}_0,\dots,\mathbf{x}_3$, the constant deformation gradient is given by

$$\mathbf{F}_e = \begin{pmatrix}\mathbf{x}_1-\mathbf{x}_0 & \mathbf{x}_2-\mathbf{x}_0 & \mathbf{x}_3-\mathbf{x}_0\end{pmatrix}\mathbf{D}_e^{-1}, \tag{40}$$

where $\mathbf{D}_e = \begin{pmatrix}\mathbf{X}_1-\mathbf{X}_0 & \mathbf{X}_2-\mathbf{X}_0 & \mathbf{X}_3-\mathbf{X}_0\end{pmatrix}$ is formed from the reference positions and its inverse can be precomputed. In either case, each element contribution takes the role of a local potential in the assembly from Eq. (11), and contributes its gradient and Hessian to the global system as described in Section 2.1.4.

Commonly used examples of hyperelastic material models that can be formulated in this way include the Corotated material model [SHST12, MZS*11], the Neohookean and Stable Neohookean materials [SGK18] and the ARAP model [LCK22, SA07]. Non-classical material models such as micropolar materials with rotational degrees of freedom can also be considered in this framework as shown by Löschner et al. [LFFJ*23, LFFJB24]. While FEM is a sophisticated numerical method that supports higher-order discretizations [SHD*18, LLK*20, SHG*22], linear elements were successfully used for very efficient methods and possibly interactive applications, for example the Fast Corotated FEM approach by Kugelstadt et al. [KKB18]. For more details on deformable solids and their spatial discretization using FEM, we recommend the courses of Kim and Eberle [KE22] and Sifakis and Barbič [SB12b].

2.2.2. Contact Potentials

The coupling of deformable solids between themselves and to other materials is typically realized through frictional contact. In the context of unconstrained optimization-based integration, *Incremental Potential Contact* (IPC) [LFS*20] gained significant popularity in recent years. IPC is designed to enable intersection-free simulations and is based on a fully implicit and unconstrained formulation of contact. The overall procedure is summarized in Algorithm 2.

The key idea is to define admissibility in terms of unsigned distances $d_k(\mathbf{x})$ between non-adjacent surface primitive pairs $k \in \mathcal{C}$, where $\mathcal{C}$ is the set of all non-incident point-triangle and non-adjacent edge-edge pairs on the surface mesh. A configuration is intersection-free if $d_k(\mathbf{x}) > 0$ for all $k \in \mathcal{C}$. To enforce this within

Algorithm 2 Incremental potential contact (IPC) with Newton's method.

$\mathbf{x} \leftarrow \tilde{\mathbf{x}} = \mathbf{x}^{\text{prev}} + \Delta t \mathbf{v}^{\text{prev}}$ ▷ initialize with inertial position
while not converged **do**
  detect active contact set $\hat{\mathcal{C}}(\mathbf{x}) = \{k \in \mathcal{C} : d_k(\mathbf{x}) \leq \hat{d}\}$
  assemble $E(\mathbf{x}) = E_{\text{inertia}} + E_{\text{elastic}} + \kappa \sum_{k \in \hat{\mathcal{C}}} b(d_k, \hat{d}) + \phi_{\text{friction}}$
  assemble gradient $\nabla E(\mathbf{x})$ and Hessian $\nabla^2 E(\mathbf{x})$
  project $\nabla^2 E(\mathbf{x})$ to positive definiteness
  solve $\nabla^2 E(\mathbf{x}) \Delta \mathbf{x} = -\nabla E(\mathbf{x})$ ▷ Eq. (9)
  $\alpha_{\text{CCD}} \leftarrow$ largest step preserving intersection-freeness via CCD
  $\alpha \leftarrow$ backtracking line search starting from α_{CCD} ▷ Eq. (25)
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha \Delta \mathbf{x}$
  update barrier stiffness κ adaptively
end while
$\mathbf{x} \leftarrow \mathbf{x}$
$\mathbf{v} \leftarrow (\mathbf{x} - \mathbf{x}^{\text{prev}})/\Delta t$

unconstrained optimization, IPC augments the incremental potential from Eq. (5) with a sum of smoothly clamped barrier potentials (see Section 2.1.5) over all primitive pairs,

$$\phi_{\text{contact}}(\mathbf{x}) = \kappa \sum_{k \in \mathcal{C}} b\big(d_k(\mathbf{x}), \hat{d}\big), \tag{41}$$

where $\kappa > 0$ is an adaptively chosen barrier stiffness, $\hat{d} > 0$ is a user-specified distance threshold controlling geometric accuracy of the contact, and $b(d, \hat{d})$ is a C^2-continuous, smoothly clamped log-barrier defined as

$$b(d, \hat{d}) = \begin{cases} -(d - \hat{d})^2 \ln\left(\frac{d}{\hat{d}}\right), & 0 < d < \hat{d}, \\ 0, & d \geq \hat{d}. \end{cases} \tag{42}$$

This barrier is exactly zero for $d \geq \hat{d}$, providing a smooth clamping that allows efficient pruning of distant pairs, and tends to infinity as $d \to 0$, generating arbitrarily large repulsive forces to prevent intersection. Because b vanishes for distant primitives, only the active constraint set $\hat{\mathcal{C}}(\mathbf{x}) = \{k \in \mathcal{C} : d_k(\mathbf{x}) \leq \hat{d}\}$ contributes non-zero terms, making evaluation efficient despite the quadratic number of potential pairs. Each barrier term plays the role of a local potential in the assembly from Eq. (11), contributing its gradient and Hessian to the global system. The resulting contact forces are given by the barrier derivatives $\lambda_k = -\kappa \partial b / \partial d_k$, which are guaranteed to be non-negative (repulsive) by construction.

To guarantee that the intersection-free invariant is maintained not just at the solution but throughout all Newton iterations, IPC augments the backtracking line search (see Section 2.1.6) with continuous collision detection (CCD): before accepting a step $\alpha \Delta \mathbf{x}$, CCD determines the largest feasible step size α_{CCD} along the search direction such that no primitive pair intersects along the linear trajectory from $\mathbf{x}$ to $\mathbf{x} + \alpha_{\text{CCD}} \Delta \mathbf{x}$. The line search then backtracks starting from α_{CCD} to find sufficient energy decrease, ensuring that every iterate remains admissible. Combined with a non-inverting elastic energy density such as neo-Hookean, this yields a trajectory that is intersection- and inversion-free at every solver iteration.

Further contributions of IPC include a smoothed and lagged friction potential (discussed below in the context of damping and friction) and an adaptive barrier stiffness update for κ that improves conditioning. Subsequent works aim to resolve spurious tangential forces in the original formulation [DLCT24], ensure convergence under refinement [LFS*23] and extend it to high-order (curved) meshes [FJZ*23]. More recently, Huang et al. [HPF*25] proposed the *Geometric Contact Potential* (GPC), a barrier formulated as a surface integral that addresses several limitations of IPC simultaneously: it eliminates spurious contact forces in rest configurations by construction, ensures convergence under mesh refinement, supports both smooth and piecewise-smooth (sharp-featured) surfaces in a unified framework, and allows substantially larger barrier extents without artificial stiffness.

While the original IPC framework did not target interactive applications, recent works aim to fill this gap by proposing efficiency and robustness improvements such as better convergence using cubic barriers with dynamic stiffness [And24], improved conditioning using a barrier-augmented Lagrangian [GLY*24], Preconditioned Conjugate Gradient solvers [SCBL24], improved Hessian projection and filtering in a Gauss-Newton method [HCLK24], localized CCD updates and improved Parallelization [LLJ*23] or integration into the Projective Dynamics framework [LMY*22] which all advertise fast GPU implementations.

Alternatively, if the requirement of fully intersection-free configurations can be relaxed, various penalty-based collision energies as investigated by Shi and Kim [SK23] or by Macklin et al. [MEM*20] with accurate penetration-depths obtained by e.g. the method of Chen et al. [CDY23] can also be applied in the context of optimization-based integration. These barrier and penalty energy-based approaches naturally fit in an optimization framework and provide strongly coupled contact. For more details on contact and friction handling in computer graphics, we recommend the course by Andrews et al. [AEF22].

2.2.3. Damping, Friction and Plasticity

So far, all forces we have considered are *conservative*: they derive from a scalar potential energy that depends only on the configuration $\mathbf{x}$. This is what makes the optimization-based formulation work: every force contributes a term to the incremental potential from Eq. (5), and minimizing the sum solves the equations of motion. However, many important physical effects are *dissipative*: damping removes kinetic energy, and friction opposes relative sliding motion. These forces generally depend on velocities and do not derive from a potential in the unknown positions alone, so they cannot simply be added as another energy term.

A useful framework for modeling such effects is the *dissipation potential* $R(\mathbf{x}, \mathbf{v})$, a non-negative scalar function of positions and velocities whose negative velocity gradient yields the dissipative force, $\mathbf{f}_d = -\nabla_{\mathbf{v}} R(\mathbf{x}, \mathbf{v})$ (see [BOFN18, MSAO18]). Many common models, including Coulomb friction and Rayleigh damping, can be cast in this form. The challenge for optimization-based integration is that R typically depends on both $\mathbf{x}$ and $\mathbf{v}$, so there is no straightforward way to incorporate it into the incremental potential. In the special case that the dissipation potential depends only on velocities, however, a corresponding objective term $\phi_d = \Delta t\, R(\mathbf{v})$ can

be defined directly, which motivates various *lagging* strategies that freeze the position dependence at a known state to reduce R to a function of velocities alone.

Rayleigh Damping Rayleigh damping is a widely used model that defines the damping force as a linear combination of mass- and stiffness-proportional terms,

$$\mathbf{f}_d = -(\alpha\mathbf{M} + \beta\mathbf{K}(\mathbf{x}))\,\mathbf{v}, \tag{43}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are the mass and stiffness damping coefficients, and $\mathbf{K}(\mathbf{x}) = \nabla^2\Psi(\mathbf{x})$ is the elastic stiffness matrix. This corresponds to the dissipation potential

$$R(\mathbf{x},\mathbf{v}) = \tfrac{1}{2}\,\mathbf{v}^T(\alpha\mathbf{M} + \beta\mathbf{K}(\mathbf{x}))\,\mathbf{v}. \tag{44}$$

Since the stiffness matrix $\mathbf{K}$ depends on positions, this dissipation potential depends on both $\mathbf{x}$ and $\mathbf{v}$, and a corresponding objective term for the incremental potential cannot be directly defined. Gast et al. [GSS*15] resolve this by *lagging* the position dependence: the stiffness matrix is evaluated at the known positions from the start of the time step, $\mathbf{K}(\mathbf{x}^{\text{prev}})$, yielding the lagged dissipation potential

$$R^{\text{lag}}(\mathbf{v}) = \tfrac{1}{2}\,\mathbf{v}^T(\alpha\mathbf{M} + \beta\mathbf{K}(\mathbf{x}^{\text{prev}}))\,\mathbf{v}. \tag{45}$$

Because R^{lag} depends only on velocities, and using the Backward Euler substitution $\mathbf{v} = (\mathbf{x} - \mathbf{x}^{\text{prev}})/\Delta t$, the corresponding objective term for the incremental potential becomes

$$\phi_{\text{damp}}(\mathbf{x}) = \frac{1}{2\,\Delta t}\,(\mathbf{x} - \mathbf{x}^{\text{prev}})^T(\alpha\mathbf{M} + \beta\mathbf{K}(\mathbf{x}^{\text{prev}}))\,(\mathbf{x} - \mathbf{x}^{\text{prev}}). \tag{46}$$

For energy-based simulation of crowds, Karamouzas et al. [KSNG17] instead argue that a fully implicit formulation of Rayleigh damping is desirable and show that omitting the lagging still results in forces that are first-order accurate w.r.t. the time step size. Brown et al. [BOFN18] extend this to Coulomb friction, strain rate damping and other models but demonstrate that this first-order approximation results in forces that are not invariant under rigid transformations. They introduce a correction term to remedy the resulting angular momentum loss. However, they apply a local-global optimization approach using ADMM which solves individual potentials locally (see Projective Dynamics in Section 2.4), and a fully implicit formulation for an unconstrained global optimization problem was not considered. Alternatively, for strain rate-based damping, an energy can be formulated by directly discretizing the strain rate in time which was used to simulate viscous threads [BAV*10] and sheets [BUAG12].

Coulomb Friction A major contribution of the IPC framework [LFS*20] is a variational treatment of Coulomb friction within unconstrained optimization. The physical model is based on the Maximal Dissipation Principle (MDP), which states that frictional forces maximize the rate of dissipation in relative tangential motion subject to the Coulomb constraint. For each active contact pair $k \in \hat{\mathcal{C}}$, a tangential sliding basis $\mathbf{T}_k(\mathbf{x}) \in \mathbb{R}^{3n\times 2}$ is constructed in the plane orthogonal to the contact normal. The local relative sliding displacement is then $\mathbf{u}_k = \mathbf{T}_k(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^{\text{prev}}) \in \mathbb{R}^2$, and the sliding velocity is $\mathbf{v}_k = \mathbf{u}_k/\Delta t$. Maximizing the dissipation rate subject to the Coulomb constraint yields the friction force

$$\mathbf{F}_k(\mathbf{x},\lambda) = \mathbf{T}_k(\mathbf{x})\,\underset{\boldsymbol{\beta}\in\mathbb{R}^2}{\arg\min}\,\boldsymbol{\beta}^T\mathbf{v}_k \quad \text{s.t.} \quad \|\boldsymbol{\beta}\| \leq \mu\lambda_k, \tag{47}$$

where $\lambda_k = -\kappa\,\partial b/\partial d_k$ is the (non-negative) contact force magnitude from the barrier potential and μ is the friction coefficient. This friction force is bimodal: when $\|\mathbf{v}_k\| > 0$ (sliding), we have $\mathbf{F}_k = -\mu\lambda_k\mathbf{T}_k\mathbf{u}_k/\|\mathbf{u}_k\|$; when $\|\mathbf{v}_k\| = 0$ (sticking), the friction direction is not uniquely defined.

Two challenges prevent direct inclusion of these friction forces in unconstrained optimization: the non-smoothness at the transition between sticking and sliding, and the absence of a well-defined dissipation potential whose spatial gradient generates friction forces. To address the first issue, IPC introduces a C^1-smooth approximation of the friction magnitude function using a velocity threshold $\epsilon_v > 0$ below which contacts are treated as static,

$$f_1(y) = \begin{cases} -\dfrac{y^2}{\epsilon_v^2\Delta t^2} + \dfrac{2y}{\epsilon_v\Delta t}, & y \in (0,\,\epsilon_v\Delta t), \\ 1, & y \geq \epsilon_v\Delta t. \end{cases} \tag{48}$$

This yields smooth friction forces $\mathbf{F}_k = -\mu\lambda_k\mathbf{T}_k\,f_1(\|\mathbf{u}_k\|)\,\mathbf{u}_k/\|\mathbf{u}_k\|$ that transition continuously from zero (sticking) to the full Coulomb magnitude (sliding).

To address the second issue and obtain a potential whose gradient generates these friction forces, IPC lags the tangential sliding basis $\mathbf{T}_k^n$ and contact force magnitude λ_k^n at their values from the previous nonlinear solve (or previous time step). With these lagged quantities held fixed, the friction contribution to the incremental potential can be written as a sum over all active contact pairs,

$$\phi_{\text{friction}}(\mathbf{x}) = \sum_{k\in\hat{\mathcal{C}}} \mu\lambda_k^n\,D(\mathbf{x},\mathbf{T}_k^n), \tag{49}$$

where D is a per-contact dissipation potential that integrates the smoothed friction magnitude and whose negative gradient with respect to $\mathbf{x}$ recovers the smoothed friction force with the lagged quantities. This lagged potential can be added directly to the incremental potential from Eq. (5) and resolved within the same Newton solve. To improve the accuracy of the friction direction and magnitude, the lagged quantities $\mathbf{T}_k^n$ and λ_k^n can be updated by solving successive minimizations within each time step.

Similar approximations were used in the penalty formulation by Macklin et al. [MEM*20]. Outside of simple test cases, however, it is not recommended to update the lagged quantities during optimization as there are no convergence guarantees in general. Larionov et al. [LLA*24] claim that this lagging can lead to inaccurate and time step dependent behavior, especially close to the slip threshold and demonstrate that force-based simulations with a fully implicit friction formulation can be beneficial for simulations where accurate frictional behavior is important.

Plasticity For plasticity, Li et al. [LLJ22] present implicit augmented energy densities of finite strain von Mises and Drucker-Prager plasticity models as well as hardening and viscoelasticity models and evaluate them in MPM and FEM frameworks. However, their approach is currently limited to the St. Venant-Kirchhoff elastic material model, while the Drucker-Prager model and hardening model require additional fixed-point iterations around the optimization problem as their yield strain is fully implicit.

2.2.4. Shells and Rods

In addition to volumetric deformables, codimensional deformables including cloth, shells, rods and hair are an important aspect of multimaterial simulations. They can also often be described by energies that can be added as contributions to the incremental potential formulation. The IPC method was also extended to provide strongly coupled frictional contact between codimensional and volumetric deformables [LKJ21].

For cloth or shells, an established approach in computer graphics is to combine a membrane deformation model for in-plane forces with a bending energy for out-of-plane deformations. The total shell energy then takes the form

$$\phi_{\text{shell}}(\mathbf{x}) = \phi_{\text{membrane}}(\mathbf{x}) + \phi_{\text{bend}}(\mathbf{x}), \tag{50}$$

where ϕ_{membrane} is obtained by integrating a membrane energy density over the surface and ϕ_{bend} is a discrete bending energy. This total shell energy enters directly as one of the potential terms ϕ_i in the incremental potential of Equation (5). Common choices for the membrane part include the Baraff-Witkin model [Kim20, BW98] or Koiter's model [CSvRV18], while established bending energies include the Discrete Shells model [GHDS03, BMF03, TG13] and the quadratic and cubic shell variants for inextensible surfaces [BWH*06, GGWZ07]. In the following, we briefly introduce the Baraff-Witkin and Discrete Shells models as representative examples. Analytic eigenanalyses of such membrane and bending energies, which are required for Hessian projection to positive definiteness, were presented by Kim [Kim20], Haomiao and Kim [WK23] and Wang et al. [WYW23].

Baraff-Witkin Model The Baraff-Witkin model [BW98] is a widely used membrane energy for cloth that measures stretching and shearing separately along the warp and weft thread directions of the fabric. As shown by Kim [Kim20], it can be expressed as an anisotropic strain energy density using the invariants

$$I_5(\mathbf{a}) = \mathbf{a}^T\mathbf{F}^T\mathbf{F}\mathbf{a}, \qquad I_6(\mathbf{a},\mathbf{b}) = \mathbf{a}^T\mathbf{F}^T\mathbf{F}\mathbf{b}, \tag{51}$$

where $\mathbf{F}$ is the deformation gradient and $\mathbf{a}$, $\mathbf{b}$ are directions in the 2D material space. The stretching energy density is given by

$$\Psi_{\text{stretch}} = \frac{k_s}{2}\left[\left(\sqrt{I_5(\mathbf{a}_u)} - b_u\right)^2 + \left(\sqrt{I_5(\mathbf{a}_v)} - b_v\right)^2\right], \tag{52}$$

where $\mathbf{a}_u$, $\mathbf{a}_v$ are orthogonal unit vectors along the warp and weft directions (the standard basis in material space), k_s is the stretching stiffness, and b_u, b_v are rest-state stretch parameters (typically 1). Using I_5 with these specific directions measures stretch independently along each thread direction rather than in the principal strain directions, which is physically consistent with woven fabrics. The shearing energy density penalizes changes of the angle between the two thread directions,

$$\Psi_{\text{shear}} = \frac{k_{\text{shear}}}{2} I_6(\mathbf{a}_u,\mathbf{a}_v)^2, \tag{53}$$

and plays a critical role in producing characteristic wrinkling behavior under extension.

Discrete Shells Bending Model While membrane energies resist in-plane deformations, a separate bending energy is needed to penalize out-of-plane deformations (i.e. curvature changes) of the surface. The Discrete Shells model [GHDS03, BMF03] defines a bending energy by measuring the change in dihedral angles along mesh edges. The discrete flexural energy is defined as a sum over all interior edges of the triangle mesh,

$$\phi_{\text{bend}}(\mathbf{x}) = k_B \sum_{e\in\mathcal{E}} \frac{\|\bar{\mathbf{e}}\|}{\bar{h}_e}\left(\theta_e(\mathbf{x}) - \bar{\theta}_e\right)^2, \tag{54}$$

where k_B is the bending stiffness, θ_e is the dihedral angle at edge e in the deformed configuration, $\bar{\theta}_e$ is its counterpart in the undeformed (rest) configuration, $\|\bar{\mathbf{e}}\|$ is the edge length in the rest configuration, and $\bar{h}_e$ is related to the heights of the two triangles sharing the edge. By storing the rest angles $\bar{\theta}_e$, this model naturally supports curved rest configurations such as hats, bent paper, or pre-shaped garments.

Strain Limiting In physical cloth, large in-plane deformations are strongly resisted, and many fabrics are nearly inextensible. Achieving this behavior in simulation by simply using a very stiff membrane model leads to numerical difficulties: on practical mesh resolutions, such stiff membranes can introduce *membrane locking*, a numerical artifact where the material appears artificially resistant to bending [LKJ21]. A common workaround is to use a relatively soft membrane energy and impose separate *strain limits* that prevent excessive stretching.

Strain limiting can be formulated as an energy potential within the optimization-based framework. In the C-IPC approach [LKJ21], the singular values σ_i of the deformation gradient $\mathbf{F}$ of each element are constrained to lie below an upper bound s (the strain limit), and this constraint is enforced through a C^2-continuous barrier, analogous in spirit to the contact barrier from Eq. (41). A second parameter $\hat{s} < s$ defines an activation threshold below which no forces are generated. To make the barrier independent of the specific choice of s and $\hat{s}$, the strain is mapped to the normalized variable

$$y = \frac{\hat{s} - \sigma_i^t}{s - \hat{s}}, \tag{55}$$

which is positive when the strain is below the threshold and approaches -1 as $\sigma_i^t \to s$. The smoothly clamped barrier is then defined as

$$b(y) = \begin{cases} -y^2\ln(1+y), & y < 0, \\ 0, & y \geq 0. \end{cases} \tag{56}$$

This function is exactly zero when the strain remains below the activation threshold ($y \geq 0$) and diverges as $y \to -1$ (i.e. $\sigma_i^t \to s$), guaranteeing that the strain limit is never exceeded. The resulting strain-limiting potential is integrated over the cloth volume to ensure mesh-resolution independent behavior,

$$\phi_{\text{SL}}(\mathbf{x}) = \kappa_s \sum_{t\in\mathcal{T}} \sum_i V^t\, b(\sigma_i^t), \tag{57}$$

where κ_s is the barrier stiffness and V^t is the reference volume of element t. Because the barriers are unilateral—they only activate when the strain threshold is exceeded—they do not over-constrain the system and thus avoid triggering locking. Furthermore, since the strain-limiting potential is resolved simultaneously with all other forces in the optimization, it avoids the compression, jittering, and numerical softening artifacts that can arise from time-step

splitting approaches where strain limits are enforced as a separate projection step.

Continuum-based Shell Models Instead of manually combining stretch and bending energies, continuum-based shell models, such as Kirchhoff-Love shells, inherit their bending stiffness from an underlying volumetric material model [WB23]. Although Kirchhoff-Love shells typically require either sophisticated C^1 continuous [TWS06, WHP11] or non-conforming [KGBG09] discretizations, it is possible to apply results from discrete differential geometry [Wei12] to construct an efficient piecewise linear discretization [WB23]. In contrast to Kirchhoff-Love shells which are rigid to transverse shearing, Cosserat shells are shear deformable and introduce additional degrees of freedom (directors) which can also be incorporated in an energy-based formulation [LFFJB24]. As an alternative to codimensional approaches, thick shells can also be simulated with volumetric prism elements and reduced integration to avoid locking [CXY*23].

Hair and Rods The main challenge of modeling hair and rods is to capture the bending and torsion of their one-dimensional geometry in three-dimensional space. Common approaches include the shear-deformable Cosserat rods [ST07, KS16] as well as shear-rigid Kirchhoff rods [HB23] as used in the Discrete Elastic Rods model [BWR*08]. These models are based on an orthonormal frame attached to the centerline of the geometry to measure bending and torsion. Depending on the model, the orthonormal frame might be explicitly represented by additional degrees of freedom or is only implicitly used during derivation. Shi et al. [SWP*23] recognized that established rod models are unsuitable for simulation of tightly coiled hair and proposed an energy-based model defined directly in terms of the vertex positions of the hair.

2.2.5. Rigid Bodies and Multibody Systems

Rigid bodies and articulated multibody systems are ubiquitous in multiphysics simulations, appearing as tools, mechanisms, characters and environmental objects that interact with deformable materials through frictional contact. Unlike deformable bodies, rigid bodies do not require a volumetric discretization, but their rotational degrees of freedom introduce challenges for optimization-based formulations that we discuss in the following.

Rigid Body Dynamics To simulate rigid bodies in the context of optimization-based integration, Macklin et al. [MEM*20] present a formulation based on generalized positions (translation and orientation of the bodies) and a kinematic map [BET14] which maps linear and angular velocities to time derivatives of the generalized positions. This enables rigid bodies and deformables to be solved in a unified way. To define a corresponding potential, the kinematic map is approximated to be constant during a time step and gyroscopic forces are integrated explicitly. Contacts are resolved using penalty energies.

Ferguson et al. [FLS*21] propose a formulation that avoids these approximations by parametrizing each rigid body i with a translation vector $\mathbf{q}_i \in \mathbb{R}^3$ and a rotation vector $\theta_i \in \mathbb{R}^3$, where the rotation matrix is recovered via Rodrigues' formula $\mathcal{R}(\theta_i)$. Because Rodrigues' formula maps arbitrary vectors to valid rotation matrices, orthogonality does not need to be imposed as a constraint, yielding a fully unconstrained optimization problem. The incremental potential splits into a translational and a rotational part. The translational potential is the standard inertial term,

$$E_{\mathbf{q}}(\mathbf{q}) = \sum_i \frac{m_i}{2\Delta t^2} \|\mathbf{q}_i - \tilde{\mathbf{q}}_i\|^2, \tag{58}$$

where $\tilde{\mathbf{q}}_i = \mathbf{q}_i^t + \Delta t\, \dot{\mathbf{q}}_i^t + \Delta t^2 m_i^{-1} \mathbf{f}_i$ is the inertial position. For the rotational part, following the simplified form of Romero et al. [RSCO25], the rotational inertial potential is

$$E_{\mathbf{R}}(\mathbf{R}) = -\frac{1}{\Delta t^2} \sum_i \mathrm{tr}\big(\mathbf{R}_i \mathbf{J}_i^l \tilde{\mathbf{R}}_i^T\big), \tag{59}$$

where $\mathbf{J}_i^l$ is the local inertia tensor of body i and $\tilde{\mathbf{R}}_i$ is the explicit rotation prediction. Instead of using angular velocities, the prediction is expressed in terms of incremental rotation steps $\Delta\mathbf{R}_i^t = \mathbf{R}_i^t (\mathbf{R}_i^{t-1})^T$ as $\tilde{\mathbf{R}}_i = (2\mathbf{I}_3 - (\Delta\mathbf{R}_i^t)^T)\,\mathbf{R}_i^t$, which avoids the need for explicit velocity variables. The full time step is then obtained by minimizing the combined incremental potential

$$\begin{aligned} \arg\min_{\theta,\mathbf{q}} \; & E_{\mathbf{q}}(\mathbf{q}) + E_{\mathbf{R}}(\mathcal{R}(\theta)) \\ & + \phi_{\text{contact}}(\mathbf{x}(\mathcal{R}(\theta),\mathbf{q})) + \phi_{\text{friction}}(\mathbf{x}(\mathcal{R}(\theta),\mathbf{q})), \end{aligned} \tag{60}$$

where the contact and friction potentials from Eqs. (41) and (49) are evaluated on the world-space surface positions $\mathbf{x}$ obtained from the rigid transformations. This formulation generalizes the IPC method to guarantee intersection-free contact with strong coupling between rigid bodies and deformables. Because rigid bodies undergo curved trajectories during a time step, CCD queries must account for these nonlinear paths, which is more involved than the linear CCD used for deformables [FLS*21]. Romero et al. [RSCO25] improve on this approach by storing rotations directly as matrices and computing derivatives in the local Lie algebra tangent space at the identity, where the exponential map reduces to simple linear and quadratic approximations. This avoids the complex and poorly conditioned derivatives of the Rodrigues formula required by Ferguson et al. [FLS*21], resulting in simpler, exact expressions and significantly improved convergence of the Newton solver.

Affine Body Dynamics Reasoning that the idealized assumption of fully rigid bodies is not always necessary, Lan et al. [LKL*22] propose Affine Body Dynamics (ABD) which relaxes the rigid body constraint to an affine transformation penalized by a very stiff orthogonality potential. Intuitively, an affine body can be understood as embedding the entire object inside a single linear finite element [LLK*20]: just as a linear tetrahedron maps its interior through an affine function with a spatially constant deformation gradient $\mathbf{F}$, ABD represents the whole body by one global affine map, so the deformation gradient is uniform over the body. Each affine body b is accordingly described by 12 degrees of freedom: a translation vector $\mathbf{p}_b \in \mathbb{R}^3$ and a linear transformation matrix $\mathbf{A}_b \in \mathbb{R}^{3\times3}$, stored as a generalized coordinate vector $\mathbf{q}_b = (\mathbf{p}_b^T, \mathbf{a}_1^T, \mathbf{a}_2^T, \mathbf{a}_3^T)^T \in \mathbb{R}^{12}$, where $\mathbf{a}_i$ are the rows of $\mathbf{A}_b$. A material point with rest position $\bar{\mathbf{x}}_k$ in the body frame is mapped to world coordinates via

$$\mathbf{x}_k = \mathbf{A}_b \bar{\mathbf{x}}_k + \mathbf{p}_b = \mathbf{J}(\bar{\mathbf{x}}_k)\,\mathbf{q}_b, \qquad \mathbf{J}(\bar{\mathbf{x}}) = \begin{pmatrix} \mathbf{I}_3 & \mathbf{I}_3 \otimes \bar{\mathbf{x}}^T \end{pmatrix}, \tag{61}$$

where $\mathbf{J}$ is a constant Jacobian matrix. This is directly analogous to the FEM shape function derivatives from Eq. (40): the deformation gradient of the affine body is simply $\mathbf{F} = \mathbf{A}_b$, and because this single-element view has constant $\mathbf{F}$, all quantities that would normally be integrated over many elements reduce to a single evaluation per body. Because the coordinates are affine (flat), velocities are linear in $\dot{\mathbf{q}}_b$ and the generalized mass matrix $\mathbf{M}_b = \int_{\Omega_b} \rho \mathbf{J}(\bar{\mathbf{x}})^T \mathbf{J}(\bar{\mathbf{x}})\, \mathrm{d}\Omega$ is constant, avoiding the nonlinear Coriolis terms that arise in standard rigid body dynamics.

To keep the body nearly rigid, a stiff orthogonality potential penalizes deviations of $\mathbf{A}_b$ from a rotation matrix,

$$\phi_\perp(\mathbf{q}_b) = \kappa V_b \|\mathbf{A}_b \mathbf{A}_b^T - \mathbf{I}_3\|_F^2, \tag{62}$$

where κ is a large penalty stiffness and V_b is the volume of the body. This potential can be evaluated efficiently as a polynomial in the rows of $\mathbf{A}_b$ without requiring matrix decompositions. The per-body incremental potential then takes the form

$$E_b(\mathbf{q}_b) = \frac{1}{2\Delta t^2} (\mathbf{q}_b - \tilde{\mathbf{q}}_b)^T \mathbf{M}_b (\mathbf{q}_b - \tilde{\mathbf{q}}_b) + \phi_\perp(\mathbf{q}_b), \tag{63}$$

where $\tilde{\mathbf{q}}_b = \mathbf{q}_b^t + \Delta t\, \dot{\mathbf{q}}_b^t + \Delta t^2 \mathbf{M}_b^{-1} \mathbf{f}_b$ is the inertial state, analogous to Eq. (5). A key advantage of ABD is that the affine map produces linear point trajectories within each time step, which greatly simplifies continuous collision detection compared to the curved trajectories of true rigid bodies, significantly reducing its cost while still ensuring intersection-free simulations.

Recently, Huang et al. [HLL*25] demonstrate how to simulate affine bodies and deformable bodies in a unified system and propose a hash-based two-level reduction strategy to improve performance of contact Hessian assembly for GPU implementations. Previously, this assembly process could present a problematic bottleneck as the degrees of freedom of a single affine body might be involved in thousands of collisions.

Multibody Systems Finally, rigid-, affine- or deformable bodies are often coupled with joints and motors to form multibody systems. Chen et al. [CLL*22] propose a unified approach for modeling articulation joints within the framework of unconstrained optimization. Their key insight is to categorize joint constraints into three types—linear equality, nonlinear equality and inequality constraints—each handled by a different mechanism that preserves the unconstrained optimization structure.

Joint constraints are formulated in terms of *attachment points* defined via barycentric coordinates on the bodies' degrees of freedom, $\mathbf{p}^i = \sum_{j=1}^{N^i} \alpha_j \mathbf{x}_j^i$, which makes the approach applicable to rigid, affine and deformable bodies alike. *Linear equality constraints* of the form $\mathbf{c}_i^T \mathbf{x} = s_i$ are enforced exactly through a change of variables $\mathbf{x} = \mathbf{U}\mathbf{z}$ that reduces the system to a lower-dimensional space where the constraints are satisfied by construction. Joints formulated in this way include point connections ($\mathbf{p}^1 = \mathbf{p}^2$), hinges (two shared points defining a rotation axis), and sliding constraints (restricting displacement to be orthogonal to a plane normal). *Nonlinear equality constraints*, such as a distance constraint maintaining a fixed length l between two points,

$$c(\mathbf{x}) = \|\mathbf{p}_1 - \mathbf{p}_2\| - l = 0, \tag{64}$$

cannot be eliminated by a linear coordinate transformation and are instead enforced through a stiff quadratic penalty potential,

$$\phi_{\mathrm{eq}}(\mathbf{x}) = \frac{k}{2} c(\mathbf{x})^2. \tag{65}$$

Finally, *inequality constraints* such as joint limits are formulated as $c(\mathbf{x}) \geq 0$ and enforced using barrier potentials,

$$\phi_{\mathrm{ineq}}(\mathbf{x}) = \kappa b(c(\mathbf{x})), \tag{66}$$

where b is a smoothly clamped log-barrier analogous to the contact barrier from Eq. (41). This is a natural fit: just as the contact barrier prevents intersection by generating arbitrarily large repulsive forces as the distance approaches zero, the joint barrier prevents the constraint from being violated. All inequality joints are formulated purely in terms of bounded distances. For example, upper and lower distance bounds are expressed as $c(\mathbf{x}) = d_u - \|\mathbf{p}_1 - \mathbf{p}_2\| \geq 0$ and $c(\mathbf{x}) = \|\mathbf{p}_1 - \mathbf{p}_2\| - d_l \geq 0$, respectively, and rotation limits for hinges are converted from dihedral angle bounds to equivalent arc-distance constraints, avoiding trigonometric functions. All of these joint potentials are added as additional terms to the global incremental potential from Eq. (5) and are resolved simultaneously with elasticity, contact and friction.

Beyond traditional joint types, Chen et al. [CDGB19] introduce energy-based point, curve and surface connections to couple different codimensional and volumetric models which also supports non-manifold deformable bodies.

2.2.6. Multiphysics Phenomena

Besides deformables and multibody systems, other physical effects can also be incorporated into optimization-based formulations. For thin shells discretized with FEM, Chen et al. [CSvRV18] simulate the effects of heating and wetting, like curling paper or shape-changing pasta.

Another aspect that can be considered in multiphysics systems are forces due to magnetic effects. For inverse design of magnetically actuated thin shells, Chen et al. [CNZ*22] couple thin shell deformations with a linear, ideal hard-magnetic potential. Westhofen et al. [WFFJB24] propose an incremental potential formulation that supports strong coupling of linear magnetic rigid bodies with frictional contact considering para- and diamagnetic effects. They demonstrate much more robust simulations in comparison to previous weakly coupled approaches.

2.2.7. Fluids and coupling

While energy-based formulations could facilitate coupling between deformables and fluids, full strong coupling is hard to achieve. To our knowledge, an energy-based, fully implicit discretization of fluid dynamics with e.g. SPH does not exist. Instead, an operator splitting approach is a viable alternative, such as the approach proposed by Xie et al. [XLYJ23] for the coupling of SPH fluids and FEM deformables. For fluids, they use a semi-implicit approach to formulate pressure and viscosity as conservative forces and to obtain corresponding potentials. For the coupling, they introduce a contact proxy of the deformables into the SPH solver as well as a fluid contact proxy into the incremental potential of the subsequent solid and frictional contact solve.

Takahashi and Batty [TB22] proposed a monolithic solver for strongly coupled fluids, rigid bodies and deformable elastic solids with frictional contact based on constrained optimization. Fluids are discretized with APIC [JSS*15] (see Section 5.6) using a variational formulation of viscosity [BB08, LBB17] reformulated as a compliant constraint in combination with a hard constraint for incompressibility. Deformables are discretized with FEM and formulated using an incremental potential as shown in Equation (5). Fluid-solid coupling is achieved through a cut-cell approach with volume fractions while contacts between solids and rigid bodies are formulated as hard constraints. All these contributions are combined into a single constrained optimization problem with primal variables for fluid velocity, elastic displacement and rigid body velocities and dual variables or Lagrange multipliers in terms of viscous stress, pressure and contact force. The system is solved using a custom Newton-type solver leveraging the structure of the problem which was improved from previous, less general monolithic solvers [TB20, TB21].

A different approach for the implicit simulation of fluid dynamics are Lagrangian mesh-based discretizations using FEM. Misztal et al. [MEB*14] used this approach to simulate immiscible multi-phase flow and perform an explicit advection step followed by an implicit solve for all other forces such as inertia, viscosity and surface tension. For the discretization they also discretize ambient space around the fluid. A similar formulation was used by Clausen et al. [CWSO13] but they also handle advection within the implicit solve. They also support coupling with elastic and plastic materials through explicit collision forces and they only mesh the materials themselves, without ambient space. This Lagrangian mesh-based approach, however, requires involved remeshing operations for merging, splitting and refinement. Also Lagrangian in nature but working on simplicial complexes, i.e. collections of points, lines, triangles and tetrahedra, Zhu et al. [ZQC*14] proposed a codimensional approach for the simulation of surface tension. They extended this approach to simulate non-Newtonian fluids [ZLQF15].

Coupling of MPM and FEM as introduced by Li et al. [LLH*24] is described in Section 5.6.

2.3. Vertex Block Descent

Newton's method applied to an incremental potential demonstrates great stability, even for large time steps. However, it is a relatively costly approach. In computer graphics, there has been particular interest to parallelize solver algorithms and better leverage GPU hardware for interactive or learning applications. Recently, Chen et al. [CLYY24] introduced *Vertex Block Descent* (VBD) as a block coordinate descent algorithm [Wri15] for minimizing the variational energy. Their approach solves a per vertex optimization problem. Compared to the global variational formulation of Backward Euler from Equation (5), VBD introduces local variational energies

$$E_i(\mathbf{x}) = \frac{m_i}{2\Delta t^2}||\mathbf{x}_i - \tilde{\mathbf{x}}_i||^2 + \sum_{j\in\mathcal{E}_i} \phi_j(\mathbf{x}), \tag{67}$$

where $\mathcal{E}_i$ is the set of all elements (e.g., FEM elements or constraints) that act on vertex i. Minimizing this local energy with Newton's method leads to a series of linear problems with 3×3 matrices that can be solved analytically. The local solves are guaranteed to reduce the global variational energy - however, the method does not necessarily converge to the same solution as Newton's method. They demonstrate impressive simulations of elastic deformables coupled with rigid bodies, frictional contact, damping and simple constraints. VBD is a very recent method, and although its applicability to a broader range of multiphysics simulations has yet to be demonstrated, we expect that future research will extend the technique for simulating new phenomena and material behaviors.

The local view and vertex-centric reformulation in VBD gives significant opportunities for parallelization (e.g. on GPUs [Mac22]), and the authors report faster and more accurate results in comparison to XPBD [MMC16]. Concurrently, Chen et al. [CHC*24b] presented a Position-Based Nonlinear Gauss-Seidel method which follows a similar idea of vertex-based Gauss-Seidel iterations and can be applied to quasistatic and dynamic simulations. However, they focus mostly on isotropic hyperelastic materials.

2.4. Projective Dynamics

A different optimization-based framework for the interactive simulation of various materials and phenomena is *Projective Dynamics* (PD) as introduced by Bouaziz et al. [BML*14]. It follows the alternating local/global optimization scheme proposed for static geometry problems by Bouaziz et al. [BDS*12] and for Hookean mass-spring systems by Liu et al. [LBOK13] and generalizes it to a wider range of geometrically motivated constraints for dynamic simulations. Originally, PD supported the simulation of elastic solids, shells and rods with contact and was later extended and generalized to support physical material models, rigid bodies, fluids, penetration-free frictional contact and more. The main advantage of the method in comparison to a full Newton method is that updating a Hessian matrix involving the constraint's second derivatives is not needed and early iterations can be more efficient in reducing the objective. This is achieved by restricting the general optimization-based framework described before to quadratic potentials that pull the degrees of freedom towards target positions obtained through non-linear local constraint projections.

2.4.1. Foundations of Projective Dynamics

Projective Dynamics is based on the Backward Euler incremental potential introduced in Equation (5), but uses constraint functions $\mathbf{C}_i$ to define potentials for individual physical effects. To facilitate a local/global split of the solver, PD introduces auxiliary variables $\mathbf{p}_i$ that can be interpreted as target configurations satisfying each constraint such that $\mathbf{C}_i(\mathbf{p}_i) = 0$. The global objective that has to be minimized in PD is then given by

$$E_{\mathrm{PD}}(\mathbf{x},\mathbf{p}) = \frac{1}{2\Delta t^2}(\mathbf{x}-\tilde{\mathbf{x}})^T\mathbf{M}(\mathbf{x}-\tilde{\mathbf{x}}) + \sum_i d_i(\mathbf{x},\mathbf{p}_i), \tag{68}$$

where d_i are constraint specific functions measuring the distance between the current configuration $\mathbf{x}$ and the target configuration on the constraint manifold $\mathbf{p}_i$. In particular, PD restricts this to

quadratic functions of the form

$$d_i(\mathbf{x}, \mathbf{p}_i) = \frac{w_i}{2} ||\mathbf{A}_i \mathbf{S}_i \mathbf{x} - \mathbf{B}_i \mathbf{p}_i||^2 , \tag{69}$$

where w_i is a nonnegative weight, $\mathbf{A}_i$ and $\mathbf{B}_i$ are constant matrices specific to the respective constraint type and $\mathbf{S}_i$ is a selection matrix mapping the global to the constraint specific degrees of freedom. This ensures that the global potential in Equation (68) remains quadratic for fixed $\mathbf{p}_i$.

Local/Global Solver In each iteration of the solver algorithm, PD first determines the target positions $\mathbf{p}_i$ for every constraint by performing a local *constraint projection* for a fixed current configuration $\mathbf{x}$

$$\arg\min_{\mathbf{p}_i} \frac{w_i}{2} ||\mathbf{A}_i \mathbf{S}_i \mathbf{x} - \mathbf{B}_i \mathbf{p}_i||^2 , \quad \text{s.t.}\, \mathbf{C}_i(\mathbf{p}_i) = 0 , \tag{70}$$

which corresponds to finding the projection of $\mathbf{x}$ to the nearest point on the constraint manifold of $\mathbf{C}_i$ with respect to distance d_i (e.g. finding the closest zero strain configuration of a deformed triangle). These local projections capture the nonlinearity of the constraint forces and closed-form solutions exist for many constraint types. The individual projections are independent and can be performed in parallel. This is followed by a global solve for updated positions $\mathbf{x}$ using the fixed, previously obtained target configuration $\mathbf{p}$

$$\min_{\mathbf{x}} E_{\mathrm{PD}}(\mathbf{x}, \mathbf{p}) , \tag{71}$$

which can be done in a single linear solve as E_{PD} is quadratic in $\mathbf{x}$. This global solve pulls the configuration of the system towards the target positions. The linear solve can be accelerated by pre-factorizing the constant system matrix in case of fixed constraint sets. To handle topology changes for contact, cutting or fluids efficient refactorization schemes [LLKC21, WTB*21] or iterative solvers [Wan15, WKB16, LMY*22] were proposed.

A broad range of geometrically motivated constraints fit in this framework, including deformation (Hookean or ARAP models), bending, joints and contact [BML*14]. However, many non-linear constitutive laws, such as the Neohookean model, as well as hard constraints, cannot be directly considered in this way. This was addressed later in generalizations of PD to a Quasi-Newton [LBK17] or ADMM [OBLN17] method. Although methods for simulating fluids or granular and plastic media based on PD were presented [WKB16, HWW18], a coupled multiphysics simulation together with, e.g., deformables and frictional contact within the PD framework was not thoroughly explored, yet.

Relation to (Extended) Position Based Dynamics Projective Dynamics and Extended Position Based Dynamics (XPBD) (see Section 3.1.1) both directly build on constraint functions and are targeted at interactive applications. However, a fundamental difference is that PD is a *primal* method solving for the physical system's degrees of freedom directly (i.e. positions or velocities), whereas XPBD is a *dual* method that enforces constraints by first solving for Lagrange multipliers. A practical consequence is that PD can easily deal with high mass ratios and may struggle with high stiffness ratios while the converse applies to XPBD [MEM*20]. Similar observations can be made for the original PBD algorithm as it can be interpreted as a restriction of XPBD to constraints with infinite stiffness.

The additional global solve in PD ensures that a global compromise between all constraints and the implicit inertia potential is found, even considering incompatible constraints (corresponding to a least squares fit [BDS*12]). For PBD and XPBD, the solution of problems with incompatible constraints depends on the ordering of the local constraint projections. In general, the original XPBD algorithm also does not converge to a state of force balance, mainly because a residual term is discarded in its derivation from the full equations of motion (see Section 3.1.1). This was recently addressed by Chen et al. [CHC*24a].

XPBD has the advantage that it can naturally handle hard constraints and also supports more general constraint functions whereas PD always requires a sensible split into a quadratic distance measure and a constraint manifold. As noted in the following paragraph, this was also addressed in the literature but requires significant deviations from the original concept of PD [LBK17, OBLN17].

Generalizations and Enhancements Applying the Chebyshev Semi-Iterative Method was proposed by Wang [Wan15] to improve the efficiency of the global solve in Projective Dynamics. In combination with a Jacobi solver, this results in a highly efficient GPU implementation of PD which also works well with dynamically changing constraint connectivity due to the iterative methods. However, Fratarcangeli et al. [FTP16] claim that the Chebyshev approach can introduce instabilities and instead propose a parallel Gauss-Seidel approach for PD and PBD.

Liu et al. [LBK17] show that the original PD algorithm can be interpreted as a Quasi-Newton method. In this context they generalize PD to more hyperelastic materials which also necessitates the introduction of a line search to the method. They improve convergence by enhancing their initial Hessian approximation from PD with L-BFGS updates. Concurrently, Overby et al. [OBLN17] demonstrate that PD can also be interpreted as a special case of ADMM and used this insight to generalize PD to arbitrary conservative forces, hard constraints and to incorporate implicit friction models [BOFN18]. However, they note that the convergence speed of the ADMM approach heavily depends on the choice of weights in the optimization algorithm.

A typical approach to further improve performance of PD is to combine it with subspace or reduced order modeling, such as Hyper-Reduced Projective Dynamics [BEH18], Medial Elastics [LLF*20], or other subspace approaches, e.g., targeted at cloth simulation (see below) [LFL*23, LLL*24].

For parameter estimation, motion planning and inverse problems, Du et al. [DWM*21] introduced DiffPD which speeds up backpropagation to facilitate efficient differentiable simulations based on PD. Li et al. [LDW*22] draw inspiration from this to make cloth simulation with frictional contact differentiable.

2.4.2. Multiphysics in Projective Dynamics

Projective Dynamics is an established approach for interactive simulations especially involving deformables, cloth and multibody systems including skinning, coupled by frictional contact. There are also a few works on multiphysics effects such as plasticity, granu-

lar media and fluids. However, their coupling to deformables within PD remains an open problem.

Multibody Systems, Contact & Damping The original PD method does not directly support rigid bodies and hard constraints. To simulate articulated multibodies coupled with deformables more accurately while maintaining the efficiency of PD, Li et al. [LLK19, LLK22] propose to represent rigid bodies using affine transformations that are locally projected to the rigid transformation manifold SE(3) while taking hard joint constraints into account in an iterative scheme.

For contact, many PD-based works either use dynamically introduced soft unilateral constraints or springs. To avoid refactorizations of the global matrix and to support dry friction, Ly et al. [LJBBD20] reformulate Projective Dynamics in terms of velocity degrees of freedom and introduce semi-implicitly computed local per-vertex contact forces that satisfy the Signorini-Coulomb law at every iteration. To facilitate real-time physically-based skinning of animated skeletons, Komaritzan and Botsch [KB18, KB19] propose Projective Skinning and a corresponding GPU implementation based on iterative solvers which supports global self-collisions. For the simulation of very high resolution meshes with locally restricted collisions in interactive simulators, Wang et al. [WTB*21] present a highly optimized method based on partial Cholesky factorization with sparse updates and penalty-based collision constraints between selected vertices.

Based on the barrier potentials used by IPC [LFS*20], Lan et al. [LMY*22] introduced a PD framework with non-penetration guarantees. They propose a local projection for the barriers and a two-level iteration scheme where barriers are only added or removed by CCD after several PD iterations to prevent sticking and oscillations during iterations. Furthermore, they present an aggregated Jacobi solver specifically tailored for better performance on the GPU.

To improve upon typical naive "ether drag" damping models, more accurate models such as Rayleigh damping [BOFN18] and Laplacian damping [LLK18] were proposed with adaptions for PD. Dinev et al. [DLK18] investigated the numerical damping introduced through the backward Euler integration of PD and propose a blending technique with the midpoint rule. However, this approach is not always successful in conserving energy [DLL*18]. To improve on this, Kee et al. [KUJH21] proposed per vertex constraints to conserve energy and momentum.

Rods and Cloth For the simulation of rods, Soler et al. [SMSH18] introduce Cosserat rods into the PD framework by deriving the local projections and weights for the corresponding stretch, shear bend and twist constraints.

Simulating cloth with widely used geometrically motivated and example-based material models is already possible within the original PD framework. Recent research mostly aims to improve performance and convergence for highly detailed simulations especially on the GPU. Li et al. [LFL*23] propose to accelerate the global solve using B-spline subspace simulation for low-frequency motion as preconditioning for a full-order iterative solver. To handle penetration-free contact, they apply a time-splitting and contact proxy approach inspired by the work of Xie et al. [XLYJ23]. As collision detection and line search filtering remained a large contributor to the cost of previous methods, Lan et al. [LLL*24] propose non-distance barriers that are dependent on the number of iterations they are active instead of exact distances and combine this with a partial CCD scheme. They also use a much smaller subspace based on a small number of eigenvectors that further improves performance of the global solve.

Elastoplasticity and Granular Materials Based on the Projective Dynamics local/global solver approach, He et al. [HWW18] introduced the Projective Peridynamics framework for the meshless simulation of elastoplastic bodies, shells, rods and viscolestic fluid and granular flows.

Fluids To simulate fluids within the PD framework, Weiler et al. [WKB16] formulate a density constraint based on an SPH discretization similar to the Position Based Fluids [MM13] approach (see Equation (79)) and derive the corresponding local constraint projections. To avoid costly refactorizations due to changing particle neighborhoods, they propose to solve the global system with a matrix-free CG solver instead.

For real-time simulation of fluid and elastic solid coupling, Brandt et al. [BSEH19] proposed the Reduced Immersed Method framework, coupling PIC/FLIP fluids [ZB05] and deformables simulated with model reduction. For the coupling they employed the *immersed boundary method* [Pes02] where both fluids and solids are treated as a single incompressible medium. This approach eliminates the need for fluid-solid surface tracking and explicit contact constraints in the PD solver, but also introduces no-slip conditions between deformables themselves and to the fluid, which can lead to sticking. The immersed boundary method can additionally lead to leakage of fluid through the boundary of thin solids.

3. Constraint-Based Multiphysics Modeling

In constraint-based simulation methods, the simulated multiphysics system is discretized using a Lagrangian description. A set of constraint functions is then used to restrict the motion of the system's degrees of freedom and combined with the equations of motion. This principle modeling approach makes this group of methods commonly quite robust under coarser material discretizations. This is specifically beneficial for real-time use cases as it allows reducing the number of degrees of freedom in the system and can thus lead to significant workload reductions. In contrast to most approaches from the category of energy-based models in Section 2, constraint-based methods are also typically designed to handle hard constraints well. Furthermore, as opposed to explicit penalty-based methods which struggle with stability when simulating stiff systems [SLM06], the constraint-based methods presented in this section can simulate such systems robustly even under large time steps.

In the most general terms, both approaches employ different strategies to solve the following non-linear system of equations

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) \\ \mathbf{C}_b(\mathbf{x}) &= \mathbf{0} \\ \mathbf{C}_u(\mathbf{x}) &\geq \mathbf{0} \end{aligned} \tag{72}$$

Figure 3: *Various examples of constraint-based material simulations.* ***Left:*** *A plastic material resembling modeling clay is pressed into strings and piles up on the ground [YLL*24]. **Middle:** Realistic wrinkles form on a position-based elastic cloth being pushed down by a heavy object [BKCW14]. **Right:** Coupled real-time simulation of an articulated excavator, modeled as a stiff multibody system, interacting with position-based cohesive soil [HG18].*

which combines Newton's second law with n_b equality (or *bilateral*) and n_u inequality (or *unilateral*) constraints $\mathbf{C}_k = [C_{k_1}(\mathbf{x}),\dots,C_{k_{n_k}}(\mathbf{x})]^T$, where $k \in \{b,u\}$, for the system's generalized coordinates $\mathbf{x} = [x_1,\dots,x_n]^T$, a system mass matrix $\mathbf{M}$, and a vector of internal and external generalized forces $\mathbf{f}(\mathbf{x})$ acting on the system's degrees of freedom. This formulation provides a flexible modeling framework for multiphysics simulations. Through combination of different constraints tailored to the desired behaviors and interactions in the same system, it enables modeling various materials and physical phenomena in a strongly coupled, monolithic manner. Figure 3 depicts several examples of constraint-based simulations in this framework.

3.1. Simulation Methods

In the following we introduce two popular constraint-based simulation methods, namely Position Based Dynamics (PBD) and eXtended Position Based Dynamics (XPBD).

3.1.1. Position Based Dynamics

Introduced to the computer graphics community in 2006 by Müller et al. [MHHR06] for simulating soft bodies and cloth, Position Based Dynamics has emerged as a unified multiphysics simulation model for interactive and real-time simulation over the last two decades through a number of extensions that allows the approach to be used for the simulation of other materials and physical phenomena.

In its original form, the method employs a local optimization scheme to minimize the errors in a set of constraints that restrict the motion of a collection of particles representing the discretized material [BML*14]. These constraints can take different forms, making it possible to flexibly model different physical phenomena, materials and their interaction. The method operates directly on the position-level. Particle positions can only move to admissible positions, making the scheme very stable and allowing for use of large time steps [MHHR06]. These qualities, paired with its simplicity, efficiency and versatility, have made PBD one of the most popular real-time physics simulation methods for games and VR applications of recent times. Among others, it is used in the real-time game multiphysics framework NVIDIA FleX [NVI24]. PBD also finds application in many popular visual effects tools, such as in Houdini [Sid24], and is also leveraged in other areas, such as medical simulations [FYCZ23]. A position-based modeling scheme similar to PBD was previously proposed by Jakobsen [Jak01]. For a more detailed introduction and overview of PBD, we refer to the survey of Bender et al. [BMM17].

Foundations of Position Based Dynamics PBD discretizes the simulated solids into n particles defined by positions $\mathbf{x}_i$, velocities $\mathbf{v}_i$ and scalar masses m_i and defines a set of l position-based scalar constraints, each restricting the relative positions of a particle subset. These constraints can be both equality and inequality constraints, where equality constraints are used for modeling *bilateral* interactions such as the resistance to stretching between two material points in a cloth, and inequality constraints are used for modeling *unilateral* interactions as for example experienced in a collision.

In a predictor-corrector type fashion the method first produces a prediction of the particle positions at the beginning of each simulation step, denoted by $\mathbf{p}_i$, before subsequently correcting the resultant constraint violations. Assuming the particle motion is unrestricted by any constraints and only affected by external forces, the particles are moved forward in time using symplectic Euler integration, yielding

$$\mathbf{p}_i = \mathbf{x}_i + \Delta t \mathbf{v}_i + \Delta t^2 \frac{\mathbf{F}_{\text{ext}}(\mathbf{x}_i)}{m_i}, \tag{73}$$

where Δt denotes the simulation time step and $\mathbf{F}_{ext}(\mathbf{x}_i)$ corresponds to the external forces applied to particle i. With the predicted particle positions in hand, temporary unilateral collision constraints are generated based on the particles' predicted trajectories, and added to the set of permanent constraints, which are then iteratively corrected via constraint projections in Gauss-Seidel style. Given the vector of concatenated predicted positions $\mathbf{p} := [\mathbf{p}_{i_1}^T, \dots, \mathbf{p}_{i_{n_j}}^T]^T$ for a subset of particles $\{i_1, \dots, i_{n_j}\}$ constrained by a single constraint C, PBD finds a position correction $\mathbf{p} + \Delta\mathbf{p}$ that satisfies the constraint through the following linearization approach.

Without loss of generality, linearizing a bilateral, scalar constraint of the form $C(\mathbf{p}) = 0$ via a first-order Taylor expansion yields the approximation

$$C(\mathbf{p} + \Delta\mathbf{p}) \approx C(\mathbf{p}) + \nabla C(\mathbf{p})^T \Delta\mathbf{p} = 0\,. \tag{74}$$

The above equation corresponds to an under-determined system for the unknown $\Delta\mathbf{p}$. To address this issue, PBD introduces the scalar Lagrange multiplier λ and restricts the position correction to lie in the constraint gradient direction, such that

$$\Delta\mathbf{p} = \lambda \mathbf{M}^{-1} \nabla C(\mathbf{p})\,, \tag{75}$$

with $\mathbf{M}$ denoting a diagonal matrix with dimensions $3n_j \times 3n_j$ containing the scalar particle masses.

The above condition ensures existence of a unique solution and also conveniently prevents any linear or angular momentum gain within the constrained system [BMM17]. By inserting Equation (75) into Equation (74) we can derive the Lagrange multiplier

$$\lambda = \frac{-C(\mathbf{p})}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p})}\,, \tag{76}$$

which is used in Equation (75) to compute the position correction and to update the particle positions. With each position correction becoming immediately visible to other constraints, this iterative scheme is non-linear in nature. The approach can easily be generalized to also handle unilateral constraints robustly by skipping the position update for a unilateral constraint in iterations in which the constraint is not violated.

Finally, after a given number of iterations, the particles' new velocities are recovered from the change in their positions as

$$\mathbf{v}_i = \frac{1}{\Delta t}(\mathbf{p}_i - \mathbf{x}_i), \tag{77}$$

and the particle positions $\mathbf{x}_i$ are set to the final predicted and corrected particle positions $\mathbf{p}_i$ for the next simulation step. The fact that the velocities are obtained from a change of admissible particle positions rather than through extrapolation makes the method very stable.

The position based dynamics approach is outlined in Algorithm 3. Note that there also exist open-source implementations of the method [B*26a, Ben26].

One limitation with this approach is that Gauss-Seidel iterations may not converge in certain situations [BML*14], which can be circumvented by using Jacobi iterations instead [MMCK14]. As an additional benefit, Jacobi iterations allow for easy parallelization of the iterative solver, on both GPUs and CPUs, as demonstrated by Macklin et al. [MMCK14] and Holz [Hol14], respectively.

Algorithm 3 Position-based dynamics

```
for all vertices i do
    initialize x_i, v_i, m_i
end for
loop
    for all vertices i do v_i ← v_i + Δt F_ext(x_i)/m_i
    for all vertices i do p_i ← x_i + Δt v_i
    for all vertices i do genCollConstraints(x_i → p_i)
    loop solverIteration times
        for all constraints do
            compute λ using Eq. (76)
            compute Δp using Eq. (75)
            p ← p + Δp
        end for
    end loop
    for all vertices i do
        v_i ← (p_i − x_i)/Δt
        x_i ← p_i
    end for
end loop
```

This can lead to improved performance compared to the original Gauss-Seidel approach as demonstrated by Frâncu et al. [FM17]. As another alternative to Gauss-Seidel iterations, Goldenthal et al. [GHF*07] compute all Lagrange multipliers combined via a linear solve in each iteration, yielding improved efficiency for inextensible cloth simulations.

Other limitations of the original PBD method are that solutions are dependent on material resolution, time-step and the number of solver iterations [BML*14], and that global angular momentum is not preserved [DB19]. The latter was addressed by Dahl and Bargteil [DB19] through particle velocity adjustments.

3.1.2. XPBD

The time-step and iteration dependence issue was addressed by the *eXtended* Position Based Dynamics (XPBD) method, proposed by Macklin et al. [MMC16]. In XPBD, the non-physical constraint relaxation approach introduced with the original PBD method [MHHR06], is replaced with the physically-plausible constraint relaxation proposed by Servin et al. [SLM06]. In this method, the calculation of the Lagrange multiplier in PBD, given by Equation (76), is replaced by a Lagrange multiplier update, which produces a change $\Delta\lambda$ applied to a total Lagrange multiplier λ in each solver iteration. For a purely elastic constraint $C(\mathbf{p})$, this change is calculated as

$$\Delta\lambda = \frac{-C(\mathbf{p}) - \frac{\alpha}{\Delta t^2}\lambda}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p}) + \frac{\alpha}{\Delta t^2}}\,, \tag{78}$$

where α corresponds to the constraint's compliance (or inverse stiffness). Viscous or visco-elastic constraint behavior can also be modeled [MMC16]. Note that, for XPBD, the position correction from Equation (75) uses $\Delta\lambda$ instead of λ as well. We can see that the original PBD update is recovered from Equation (78) for zero compliance with $\alpha = 0$ (i.e. for hard constraints).

Algorithm 4 outlines the XPBD approach. Note that only the

Algorithm 4 eXtended Position-Based Dynamics (XPBD)

1: **for all** vertices i **do**
2: initialize $\mathbf{x}_i$, $\mathbf{v}_i$, m_i
3: **end for**
4: **loop**
5: **for all** vertices i **do** $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t \mathbf{F}_{\text{ext}}(\mathbf{x}_i)/m_i$
6: **for all** vertices i **do** $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$
7: **for all** vertices i **do** genCollConstraints($\mathbf{x}_i \rightarrow \mathbf{p}_i$)
8: **loop** solverIteration **times**
9: **for all** constraints **do**
10: compute $\Delta\lambda$ using Eq. (78)
11: compute $\Delta\mathbf{p}$ using Eq. (75)
12: $\lambda \leftarrow \lambda + \Delta\lambda$
13: $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$
14: **end for**
15: **end loop**
16: **for all** vertices i **do**
17: $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)/\Delta t$
18: $\mathbf{x}_i \leftarrow \mathbf{p}_i$
19: **end for**
20: **end loop**

lines 10 and 12 differ from the PBD algorithm 3. XPBD is also available as open-source implementation [B*26a, Ben26].

The XPBD solver is derived from a Quasi-Newton method which approximates the system's Hessian with its mass matrix, ignoring second derivatives of the constraint functions [MMC16]. Specifically for interactive real-time applications, this has several advantages, as calculating the Hessian is a time consuming process, and its use requires strategies for dealing with indefinite or singular Hessians [BML*14], creating challenges for robustness. While this approximation in XPBD reduces the convergence rate (compared to Newton's method), it does not significantly impact the end result of the XPBD solver and makes it more competitive for interactive applications. In some cases however, a second simplification employed in the derivation of XPBD, which always assumes the nonlinear residual (corresponding to the right hand side in Newton's method) to be zero, can prevent it from reaching force balance with respect to the equations of motion. For example, this entails that performing more solver iterations does not necessarily minimize the inertia potential of the system. This was recently addressed by Chen et al. [CHC*24a].

The high stability of (X)PBD is related to its derivation from implicit Euler integration [MMC16, FM17], a characteristic also found in Projective Dynamics (PD) (see Section 2.4). As a consequence, both methods suffer from artificial damping introduced by this numerical integration scheme, which causes strictly dissipating energy in the system depending on the time step size. In order to reduce the artificial damping in PBD, higher-order integrators could be used [BMM17]. For inverse problems including parameter, pose and shape optimization involving deformables and contacts, XPBD can also be made differentiable as demonstrated by Stuyck and Chen [SC23].

3.1.3. Performance Improvements

A multigrid approach for improving the convergence rate of the PBD solver was proposed by Müller [Mü08]. With the same goal in mind, Wang [Wan15] proposes the use of the Chebyshev method applied to the Jacobi-style PBD solver [MMCK14, Hol14]. The authors' experiments show that while this approach works well in certain cases, it is unable to significantly improve the rate of convergence in all cases and can even lead to divergence [Wan15]. Mercier-Aubin and Kry [MAK24] employ model reduction techniques to adaptively rigidify portions of elastic solids simulated in XPBD, based on local strain rates within the material, significantly reducing the workload and increasing the convergence rate in XPBD. Macklin et al. [MSL*19] employ sub-stepping with reduced solver iterations, yielding improved convergence and higher accuracy, as well as reduced artifical damping through improved energy conservation. Higher accuracy in XPBD can also be achieved by correcting energy residuals in constraints [Cet19].

Commonly, graph coloring algorithms [FP13] are used for the parallelization of Gauss-Seidel-style solvers as featured in PBD. For dense constraint networks this approach can yield large numbers of independent constraint partitions which reduces the level of solver parallelism. Thon-That et al. [TTKA23] address this issue by employing a block-iterative approach and applying graph coloring on the resultant graph of sub-domains. This sub-structuring leads to faster solver convergence while also reducing the number of independent constraint partitions, consequently improving solver parallelism.

3.2. Materials and Phenomena

In the following we will discuss different materials and phenomena that can be modeled using the constraint-based simulation approach.

3.2.1. Fluids

For Position Based Fluids (PBF), Macklin and Müller [MM13] employ the fluid model proposed by Bodin et al. [BLS12]. Here, an incompressible fluid is modeled using the constraint $C_i = \rho_i/\rho_0 - 1 = 0$, which enforces a uniform reference density ρ_0 at every position $\mathbf{x}_i$ in the medium. The per-particle density ρ_i is computed using SPH-based material interpolation (see Section 4.1.1), yielding the position-level fluid constraint

$$C_i = \frac{1}{\rho_0} \sum_{j=1}^{n} m_j W(\mathbf{x}_i - \mathbf{x}_j, h) - 1 = 0 \tag{79}$$

which dynamically constrains each fluid particle i to its neighboring particles j within the compact support of the SPH kernel W. As in SPH, here we also assume an entirely meshfree Lagrangian view of the medium, which makes the method ideal for modeling the large deformations observed in fluids.

Takahashi et al. [TNF14] extend the above method for the simulation of viscous fluids with elasticity, capturing buckling and coiling effects, and demonstrating phase transitions caused by temperature changes. Xing et al. [XRW*22] introduce a strong surface tension model based on an area-minimization constraint. The constraint is derived from local meshes constructed at the fluid

interface, thereby overcoming limitations in the original model which makes use of artificial pressure terms [MM13]. Abu Rumman et al. [ARNM*20] propose a technique for the two-way coupling of deformable PBD solids and SPH fluids, which can handle challenging scenarios such as leakage-free interactions with thin deformable shells, and overcomes the issue of particle clumping and visible gaps at fluid/solid boundaries in the original PBF [MM13]. In their unified SPH-based multiphysics model, Shao et al. [SLZ17] use position-based non-penetration constraints to improve fluid/solid coupling and enable small-scale turbulences through vorticity constraints that are solved using PBD. Macklin et al. [MMCK14] propose a PBD framework for the unified simulation of position-based fluids, smoke, rigid bodies and deformable solids using a GPU-based Jacobi solver. Barreiro et al. [BGAO17] demonstrate position-based simulations of viscous, viscoelastic, elastoplastic, and inviscid liquids, for which they employ a novel, doubly constrained PBD solver that can handle position-based constraints as well as the velocity-based constraints required by their proposed viscoelasticity model.

3.2.2. Rigid Bodies

Further extending the capabilities of the PBD framework for multiphysics modeling, Deul et al. [DCB16] enable maximal coordinate rigid body dynamics simulation with joints in PBD by strongly coupling position-based rigid bodies with particles in a unified manner. This approach has stability benefits compared to the direct forcing method proposed by Müller et al. [MHHR06] for coupling PBD particles with rigid bodies. The method was later also applied for modeling rigid bodies in XPBD, with further extensions for simulating restitution and friction [FM17, MMC*20].

3.2.3. Continuous Materials

To model continuous materials in PBD, various works derive position-based constraints from constitutive material models for elastic, viscous and plastic deformable solids.

Bender et al. [BKCW14] were the first to demonstrate how to model continuous materials in the PBD framework independent of the resolution in the material discretization – a limitation of the original PBD approach [BML*14]. The method uses an FEM-based material discretization and employs position-level constraints on the strain energy in deformed tetrahedral volume elements, and captures various complex physical phenomena such as lateral contraction, anisotropy, viscoelasticity and elastoplasticity. Müller et al. [MCKM15] propose a strain limiting approach by directly constraining the entries of the Green–Saint-Venant strain tensor via position-based constraints, which was recently adapted for use in XPBD [Cet24]. Macklin et al. [MMC16] and Francu et al. [FM17] employ a constraint regularization method for resolution-independent simulation of elastic, continuous materials in XPBD, initially proposed by Servin et al. [SLM06] for constraint-based modeling at the velocity-level. Rather than constraining strain energies directly as proposed by Bender et al. [BKCW14], here, position-based constraints are derived from elastic energy potentials and included in the XPBD framework. While the method is limited to Hookean materials, a recent extension enables Neo-Hookean materials [MM21]. Ton-That et al. [TTKA23] further improve the approach, demonstrating better convergence, and Chen et al. [CHC*24a] later generalize XPBD to arbitrary hyperelasticity by reinterpreting the Lagrange multipliers as components of the stress tensor. Their method also shows significantly improved accuracy and convergence behavior. Saillant et al. [SZDJ24] introduce a constraint formulation of higher-order finite elements for XPBD, increasing simulation accuracy and performance. They also address that some continuous material constraints were not stable at rest in XPBD.

As an alternative to the physically accurate methods above, efficient simulation of visually plausible deformations can be achieved with shape matching, in which sets of material points are regrouped and their deformed state mapped to the undeformed rest state of a target shape without any need for connectivity information. This geometrically motivated approach was introduced by Müller et al. [MHTG05] and subsequently adapted to the PBD framework for deformation and fracture [Cho14, CMM16]. For more information about this family of techniques we recommend the survey on PBD by Bender et al. [BMM17].

3.2.4. Rods

As a special type of continuous material, rods find application in various areas of computer graphics, in both real-time and offline scenarios. They are particularly well-suited for the representation of hair, fur, vegetation, cables and ropes, and several position-based approaches have been proposed for their simulation.

Müller et al. [MKC12] use a *follow the leader* approach for modeling simple, position-based rods in PBD designed for inextensible hair and fur. While the method guarantees zero stretch in a single iteration, making it very fast, it suffers from energy injection and results in implausible behavior, which is addressed through artificial damping. Umetani et al. [USS15] introduce a novel discretization of Cosserat theory for the position-based modeling of bending and twisting elastic rods by using *ghost points* to define consistent rod material frames. Kugelstadt and Schömer [KS16] extend the approach through use of quaternions as material frames, allowing them to formulate strain measures directly as position and orientation constraints. For the simulation of stiff, elastic rods and trees in XPBD, Deul et al. [DKWB18] replace the standard Gauss-Seidel solver by a direct solver, which guarantees high material stiffness and yields significant speed-ups compared to [USS15] and [KS16]. Angles et al. [ARM*19] add support for volumetric deformations, and demonstrate volume conservation of compressed or extended rods.

3.2.5. Granular Materials

We can identify two popular approaches for modeling granular materials in literature, both of which are compatible with the Position Based Dynamics framework. These are DEM-based approaches and approaches based on plasticity theory.

DEM-based simulation of cohesionless granular materials in PBD was demonstrated by both Holz [Hol14] and Macklin et al. [MMCK14], employing inter-particle Coulomb friction constraints modeled at the position-level for stable piling of granular materials. Holz [Hol14] improved the behavior further by modeling constraints as spring-dampers, enabling viscoelastic material

behavior and making the artificial damping proposed by Müller et al. [MHHR06] unnecessary for this use case. Viscoelastic constraint behavior was also proposed by Macklin et al. [MMC16] in XPBD, and also by Francu et al. [FM17] who demonstrated viscoelastic collisions among others for granular material simulation. Later, Holz and Galarneau [HG18] modeled position-based cohesive granular materials through use of a unilateral adhesion constraint derived from a capillary bridge model, details of which can be found in [KKHS20].

In a variation of XPBD, Yu et al. [YLL*24] employ plasticity theory for the simulation of granular and other inelastic materials. Their proposed eXtended Position-Based Inelasticity (XPBI) framework can represent a variety of visco- and elastoplastic materials, such as snow, sand, metal and foam in a unified, strongly coupled manner. Just like MPM (see Section 5.6), XPBI employs a hybrid grid-particle discretization, which facilitates capturing the large plastic deformations and fracture commonly observed in plastic and granular materials. However, without use of an Eulerian background grid, XPBI does not suffer from the grid artifacts common to MPM simulations.

4. Lagrangian Point-Based Methods

In this section we will look in more detail at current approaches using a point-based discretization of physical objects and phenomena. Specifically, we will go into detail about how to solve problems defined in the continuum on a set of points.

In general, an unordered set of points is a versatile representation for a large variety of physical objects. Particularly volumetric objects with changing topology, such as fluids and granular materials, are well suited to be represented in this way, due to their lack of inherent fixed structure. The representation as points can also be very efficient in terms of computational efficiency and memory usage, since only regions of interest will be discretized. This can be further improved by using spatial adaptivity, similar to grid- or mesh-based methods. Objects where topological changes are less desirable, such as rigid bodies and deformable solids, can be represented by simply "locking" specific particle arrangements. For most particle-based methods, this does not require special handling and fits directly into the respective paradigm. More challenging are co-dimensional structures, such as shells, thin sheets, chains, or rods, which require special treatment in particle-based methods. The difficulty stems from the fact, that the unordered particles do not natively carry any information about co-dimensionality and therefore have to be explicitly classified and transitioned.

For the case of fixed topology, the Finite Element Method (FEM) is also a popular choice in graphics, which differentiates itself from point-based methods by using a mesh. While FEM is particularly useful for the Lagrangian simulation of deformable solids, it typically needs to be explicitly coupled to other physical effects through non-unified interaction terms. In contrast, purely point-based methods more easily permit a unified discretization of multiple physical effects. In our case of multiphysical simulation, FEM has become frequently used for the discretization of energy-based systems, which are outlined in more detail in Section 2.2.

With respect to Lagrangian point-based methods, we introduce the approach of Smoothed Particle Hydrodynamics (SPH) in Section 4.1.1. Although SPH was initially proposed for astrophysics [Luc77, GM77] and is now mainly used for fluid dynamics, recent work has shown its remarkable ability for coupling of many different materials and physical effects (see Figure 4). In Section 4.1.2, we will briefly discuss the conceptual relationship to other purely particle-based methods, such as Moving Least Squares (MLS) [LS81] and the Reproducing Kernel Particle Method (RKPM) [LJZ95], both of which have also shown versatility in modeling multiphysics phenomena. Finally, we will discuss how multi-material and multiphysics effects are typically incorporated into the point-based formulation.

4.1. Simulation Methods

4.1.1. Smoothed Particle Hydrodynamics

The Smoothed Particle Hydrodynamics (SPH) method approximates a continuous spatial integral using unordered and unconnected point clouds with associated physical quantities such as mass m and rest density ρ_0. The core of SPH lies in computing material interactions of these finite points, using a (compactly-supported) kernel function W. The kernel function in itself is an approximation of the δ-distribution identity. In the continuum, this can be written as

$$f(\mathbf{x}) = \int f(\mathbf{y})\delta(\mathbf{x}-\mathbf{y})d\mathbf{y} \approx \int f(\mathbf{y})W(\mathbf{x}-\mathbf{y};h)d\mathbf{y}\,. \tag{80}$$

In SPH, this continuous integral is approximated using a finite number of discrete points $\mathbf{x}_j$ within the support radius $\hbar$ which depends on the smoothing length h in order to compute arbitrary field quantity $f(\mathbf{x})$ as

$$f(\mathbf{x}) \approx \sum_j V_j f(\mathbf{x}_j) W(\mathbf{x}-\mathbf{x}_j;h)\,. \tag{81}$$

This summation approximates the volume integral by assuming that each point has an associated finite volume V_j in which the function value $f(\mathbf{x}_j)$ remains constant. In order to satisfy this approximation, the SPH kernel W has to fulfill the following conditions $\forall\, \mathbf{r} \in \mathbb{R}^d, h \in \mathbb{R}^+$ [KBST19]:

$$\int_{\mathbb{R}^d} W(\mathbf{r}';h)dv' = 1 \qquad \text{(normalization condition)}$$

$$\lim_{h'\to 0} W(\mathbf{r};h') = \delta(\mathbf{r}) \qquad \text{(Dirac-}\delta\text{ condition)}$$

$$W(\mathbf{r};h) \geq 0 \qquad \text{(positivity condition)}$$

$$W(\mathbf{r};h) = W(-\mathbf{r};h) \qquad \text{(symmetry condition)}$$

$$W(\mathbf{r};h) = 0 \text{ for } \|\mathbf{r}\| \geq \hbar. \qquad \text{(compact support condition)}$$

The cubic-spline kernel is one of the most common choices, fulfilling all necessary conditions.

In the following we will use the common abbreviations $A_j = A(\mathbf{x}_j)$ for a field quantity A and $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j;h)$ for the kernel function.

Since we introduced the kernel W, derivatives can be computed

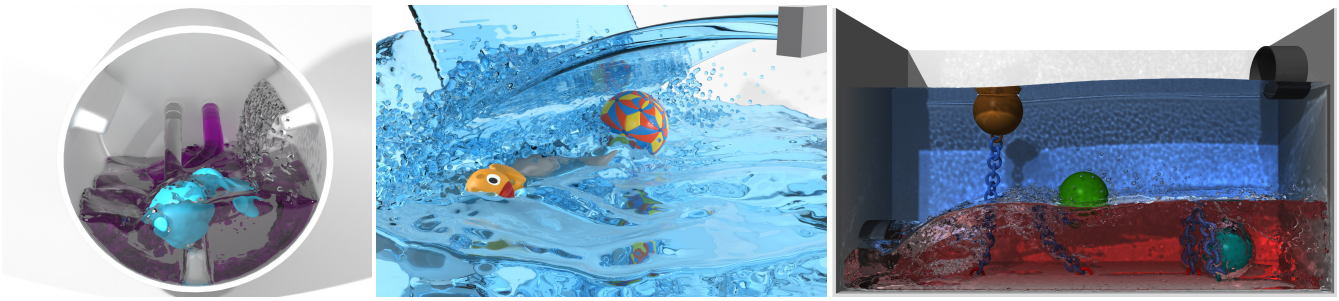

Figure 4: ***Left:*** *SPH simulation of an elastic bunny which is coupled with a fluid, a highly viscous material and a rotating washing machine drum [WJB23].* ***Middle:*** *An SPH fluid interacts with a duck and a ball [LBJB23].* ***Right:*** *A fluid and a highly viscous material are interacting with chains of rigid bodies [GPB*19].*

by simply applying the derivative operator only to the kernel function. Therefore the gradient is determined as

$$\nabla A_i \approx \sum_j V_j A_j \nabla W_{ij}\,. \tag{82}$$

Over the years, multiple variations of computing derivatives using the SPH kernel have been proposed, each with their own properties and implications [KBST19]. The most most widely used formulations for first order derivatives are the difference formula:

$$\nabla A_i \approx \sum_j \frac{m_j}{\rho_j}\left(A_j - A_i\right)\nabla W_{ij}, \tag{83}$$

which provides an improved approximation, and the symmetric formula:

$$\nabla A_i \approx \rho_i \sum_j m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2}\right)\nabla W_{ij}, \tag{84}$$

which is typically used to compute linear and angular momentum conserving forces.

In addition, it has become common to evaluate second derivatives without explicitly taking the second derivative of the kernel function [KBST22], but instead using

$$\nabla^2 A_i \approx 2\sum_j \frac{m_j}{\rho_j}\left(A_i - A_j\right)\frac{(\mathbf{x}_i - \mathbf{x}_j)\cdot\nabla W_{ij}}{||\mathbf{x}_i - \mathbf{x}_j||^2 + 0.01h^2} \tag{85}$$

$$\nabla^2 \mathbf{A}_i \approx 2(d+2)\sum_j \frac{m_j}{\rho_j}\frac{(A_i - A_j)\cdot(\mathbf{x}_i - \mathbf{x}_j)}{||\mathbf{x}_i - \mathbf{x}_j||^2 + 0.01h^2}\nabla W_{ij}, \tag{86}$$

for scalar and vector quantities, respectively, where d is the spatial dimension. The reason for this is that the Laplacian of the kernel function changes sign within its support radius and therefore the computation is prone to noise in the particle distribution [Pri12].

For more details, including details on implementation, the reader is referred to recent state-of-the-art reports [KBST22, IOS*14, LL10].

4.1.2. MLS and RKPM

In graphics, SPH is sometimes combined with two other methods due to their improved guarantees on the order of approximation or stability [WJB23]. These methods are Moving Least Squares (MLS) [LS81] and the Reproducing-Kernel Particle Method (RKPM) [LJZ95] and can be considered variations of a kernel-based method. Rewriting Equation (81) using a generalized shape function $\phi(\mathbf{x})$ and smoothing length h yields

$$f(\mathbf{x}) = \sum_j f(\mathbf{x}_j)\phi(\mathbf{x}, \mathbf{x}_j; h). \tag{87}$$

MLS and RKPM can now be represented by expressing $\phi(\mathbf{x})$ in the following ways [AW09, WJB23]:

$$\phi^{\text{SPH}}(\mathbf{x}, \mathbf{x}_j; \theta) = V_j W(\mathbf{x} - \mathbf{x}_j; h), \tag{88}$$

$$\begin{aligned}\phi^{\text{MLS}}(\mathbf{x}, \mathbf{x}_j; \theta) &= \mathbf{b}(\mathbf{x})^T \mathbf{M}(\mathbf{x})\mathbf{b}(\mathbf{x}_j)W(\mathbf{x} - \mathbf{x}_j; h)\\ \mathbf{M}(\mathbf{x}) &= \left[\sum_j W(\mathbf{x} - \mathbf{x}_j; h)\mathbf{b}(\mathbf{x}_j)\mathbf{b}(\mathbf{x}_j)^T\right]^{-1},\end{aligned} \tag{89}$$

$$\begin{aligned}\phi^{\text{RKPM}}(\mathbf{x}, \mathbf{x}_j; \theta) &= V_j \mathbf{b}(\mathbf{x})^T \mathbf{N}(\mathbf{x})\mathbf{b}(\mathbf{x}_j)W(\mathbf{x} - \mathbf{x}_j; h)\\ \mathbf{N}(\mathbf{x}) &= \left[\sum_j V_j W(\mathbf{x} - \mathbf{x}_j; h)\mathbf{b}(\mathbf{x}_j)\mathbf{b}(\mathbf{x}_j)^T\right]^{-1}.\end{aligned} \tag{90}$$

The vector $\mathbf{b}$ is typically chosen as a specific polynomial basis, such as the linear basis $\mathbf{b}(\mathbf{x}) = [1\ x\ y\ z]$.

In SPH simulations, first-order MLS derivatives can be used to guarantee linear consistency as demonstrated by Band et al. [BGPT18] and Westhofen et al. [WJB23]. MLS was also investigated to simulate deformable solids with elastic and plastic deformations [MKN*04], fracturing [PKA*05] and as discretization for unified simulations of volumetric deformables, shells and rods [MKB*10]. Chen et al. [CLC*20] propose a moving least square reproducing-kernel method for multiphase continua simulation. They also integrate a previous phase-field model for simulating mixtures, and show a multitude of coupled systems.

4.2. Materials and Phenomena

In this section we discuss SPH methods for the simulation of multiple materials and phenomena.

Algorithm 5 Simulation loop for SPH simulation of weakly compressible fluids.

for all *particle i* **do**
  compute density ρ_i using using Eq. (92)
  determine pressure p_i using state equation (Eq. (93))
end for
for all *particle i* **do**
  compute pressure force $\mathbf{f}_i^{\mathrm{p}} = -\nabla p_i$ using Eq. (94)
  compute viscosity force $\mathbf{f}_i^{\mathrm{v}} = \nu\nabla^2\mathbf{v}_i$, e.g, using Eq. (86)
end for
for all *particle i* **do**
  $\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{\rho_i}\left(\mathbf{f}_i^{\text{pressure}} + \mathbf{f}_i^{\text{viscosity}} + \mathbf{f}_i^{\text{ext}}\right)$
  $\mathbf{x}_i(t+\Delta t) = \mathbf{x}_i(t) + \Delta t\,\mathbf{v}_i(t+\Delta t)$
end for

4.2.1. Fluid Dynamics

As described in the last section, SPH was proposed as a general function approximation framework. However, as the name implies, Smoothed Particle Hydrodynamics mainly started out as a method for the simulation of incompressible fluids, commonly described by the incompressible Navier-Stokes equations

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f}, \qquad \frac{D\rho}{Dt} = -\rho(\nabla\cdot\mathbf{v}) = 0, \tag{91}$$

where ρ, $\mathbf{v}$, p, μ and $\mathbf{f}$ denote density, velocity, pressure, dynamic viscosity and external forces, respectively. The first equation describes conservation of momentum, while the second describes conservation of mass. Both of these have to be fulfilled for any continuous physical medium, while the specific form in Equation (91) describes the dynamic motion of incompressible fluids.

Pressure Arguably the most important step in solving this equation is computing the pressure force density $-\nabla p$, which ensures constant density within incompressible fluids. Mathematically speaking, it acts as a Lagrange multiplier enforcing the incompressibility constraint of the fluid. It can also be interpreted as a kind of "contact" or "collision" resolution for point-based methods, as it prevents particles from "intersecting" each other.

Early approaches, such as weakly compressible SPH (WCSPH) [BT07], used an equation of state (EOS) to explicitly relate density to pressure. The density can be determined by applying the standard SPH approach (see Eq. (81)) to the density field:

$$\rho_i = \sum_j m_j W_{ij}. \tag{92}$$

An example for an equation of state is

$$p_i = \frac{\kappa\rho_0}{\gamma}\left(\left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1\right), \tag{93}$$

where γ and κ are stiffness parameters. The equation of state determines the pressure values for each particle which can be used to determine the final pressure forces using the symmetric formula as

$$\mathbf{f}^{\mathrm{p}} = -\nabla p_i \approx \rho_i \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla W_{ij}. \tag{94}$$

A WCSPH simulation step is outlined in Algorithm 5.

WCSPH requires the use of a stiffness constant, which when combined with explicit time integration, could easily result in compression artifacts for low stiffnesses or instabilities for large stiffness values. To mitigate this, implicit methods were proposed, which enforce constant density by solving the pressure Poisson equation (PPE)

$$\Delta t\nabla^2 p = \frac{D\rho}{Dt} \tag{95}$$

in various ways. Solenthaler and Pajarola [SP09] introduced a predictive-corrective incompressible SPH (PCISPH) approach for reducing density deviations. This method showed significant improvement over explicit pressure solvers. Bodin et al. [BLS12] introduced a constant density constraint for modeling incompressible fluids in a constraint-based simulation framework, subsequently adopted by Macklin and Müller [MM13] for use in a position-based fluid (PBF) solver (cf. Section 3.1.1). Later, Weiler et al. [WKB16] solved the same constraint using Projective Dynamics (see Section 2.4). Ihmsen et al. [ICS*14] proposed implicit incompressible SPH (IISPH) to directly solve the PPE by computing the Laplace operator on the left hand side of Equation (95) using two SPH derivatives – the first one to compute the pressure gradient ∇p, and the second one to determine the divergence of this gradient $\nabla\cdot(\nabla p) = \nabla^2 p$. Bender and Koschier [BK17] and later Cornelis et al. [CBG*19] proposed to solve a second PPE $\Delta t\nabla^2 p = \rho\nabla\cdot\mathbf{v}$ to enforce a constant density and a divergence-free velocity field. The former is commonly referred to as divergence-free SPH (DFSPH). Enforcing both improves the stability of the simulation and the performance since it reduces the iteration count of the pressure solver. Note that there exist open-source implementations of most of the aforementioned pressure solvers [B*26b, Ben26].

All aforementioned implicit pressure solvers are equivalent except for small differences [KBST22]. While PCISPH uses a prototype particle to avoid a division by zero when computing the diagonal matrix entries, later works avoid this due to stability issues. PBF updates the particle positions and therefore the matrix in each iteration step of the solver. However, IISPH and DFSPH solve the PPE on the velocity level to avoid the expensive matrix updates. Finally, in contrast to the other solvers, DFSPH also enforces a divergence-free velocity field to improve the stability and performance.

For more details on SPH-based pressure solvers, we refer the reader to the surveys of Ihmsen et al. [IOS*14] and Koschier et al. [KBST19, KBST22].

Viscosity The Laplacian of the velocity field in Equation (91) denotes the viscosity forces. These are traditionally decoupled from the pressure solvers by operator-splitting and computed in isolation. Some approaches simply "smooth" the velocity field to simulate viscous behavior, for example using artificial viscosity [BT07] or XSPH [SB12a]. When using XSPH, the velocity field is updated in each time step by computing a smoothed velocity $\hat{\mathbf{v}}_i$ for each particle i as:

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \alpha\sum_j \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i)W_{ij}, \tag{96}$$

where $0 \leq \alpha < 1$ is a user-defined parameter. Others approximate the viscous stress tensor by either taking two first-order SPH derivatives [TDF*15] or a combination of an SPH derivative with finite differences [Pri12] (see Equations (85) and (86)).

While low viscosity fluids can be simulated using explicit time integration, implicit methods have been investigated in recent years to simulate highly viscous fluids. Some of these methods use the strain rate tensor for an implicit formulation [TDF*15, PT17, BGAO17]. However, Weiler et al. [WKBB18] showed that this leads to visual artifacts at the surface due to a particle deficiency problem. Therefore, they propose an implicit method which directly determines the Laplacian of the velocity field.

These implicit solvers are weakly coupled with the pressure solver and enable the simulation of highly viscous Newtonian fluids. Liu et al. [LHWW22] introduced a method based on the idea of the SIMPLE algorithm to implement a strong coupling of pressure and viscosity solver. The simulation of non-Newtonian viscous material behavior was demonstrated by Zhang et al. [ZLX*24].

Multi-Phase, Mixtures and Porous Flow After determining how to compute the main terms in Equation (91), it becomes interesting how to couple multiple fluids either in a multi-phase simulation, or through mixture models. The most widely-used method of stably coupling multiple fluids in SPH, even with relatively large density ratios, was proposed by Solenthaler and Pajarola [SP08]. In this method, different fluid materials, i.e., phases, are represented as their own distinct set of particles. An alternative approach for multi-phase simulations uses only a single set of particles and represents individual phases by volume fractions of these particles [RLY*14, YCR*15, YJL*16, RXL21, RHLC22, YR23, XWW*23]. This approach can model effects such as mixing, diffusion and dissolution very well. However, conservation of mass and and momentum across all mixture components becomes more challenging, as does reconstruction of individual phase boundaries.

Porous flow is a type of interaction within a mixture of different materials, which describes the flow of a fluid through another porous medium and is governed by Darcy's law. Early approaches relied on shrinking and growing particles [LAD08, LD09, PC13] as they enter and exit porous material. Movement within the porous body is tracked using a saturation term, which can also affect material properties. Another approach for porous flow is adopted by Yan et al. [YJL*16], in which porous flow is computed using their multi-phase fluid framework. This avoids changing particle sizes by adjusting per-particle volume fractions of each phase. This is beneficial, as changing particle size and mass can result in instabilities due to large mass ratios. Newer works propose an alternate way to compute porous flow with SPH particles. The fluid particles enter and overlap the porous body, but become decoupled from the pressure solver [RXL21]. Instead, their motion is entirely governed by Darcy's law when overlapping solid particles. A similar approach is also adopted by Wang et al. [WFM21] for simulating wet sand, where the sand is simulated using the Discrete Element Method (DEM). Böttcher et al. [BWJB25] instead take local porosity into account when computing pressure forces to limit particle overlap and use coupling forces to simulate drag and capillary action.

Further Fluid Phenomena There are a variety of further fluid-based effects that can be captured using SPH, which are typically modeled by an additional interaction force in Equation (91). One such effect, which has garnered increased interest in recent years, is fluid turbulence. SPH fluid simulations can suffer from numerical diffusion which leads to a loss in turbulent details. Therefore, different methods were proposed to maintain these details, generally by reducing the amount of kinetic energy lost through dissipation. Bender et al. [BKKW19] introduced rotational degrees of freedom to the SPH particles. Using the equation for conservation of angular momentum and an additional momentum exchange term, it was possible to obtain better conservation of energy and turbulent details. This was further improved by using vorticity refinement [LWB*21], which uses a stream function to relate vorticity to velocity, instead of a momentum-exchange term. An advantage of this approach is, that it allows for better control of turbulence and even allows for amplification of vortices. Finally, where Liu et al. [LWB*21] used the stream function to avoid explicit computation of the Biot-Savart integral, Ye et al. [YWX*24] proposed an efficient computation of this integral using the Monte Carlo method.

Surface tension is another phenomenon that is important for realistic fluid simulations. It is caused by pressure differences at fluid interfaces, which is formalized by the Young-Laplace equation, relating the force to the local curvature of the interface and the surface normal. Initial attempts at computing surface tension explicitly, by means of the continuum-surface force (CSF) [MCG03], have found the surface particle deficiency to result in unstable and erroneous forces. Using a cohesion-force-based approach instead exhibited more stability [BT07]. This has been further improved on by using a combination of cohesion forces and approximate curvature forces with custom kernel functions [AAT13]. Furthermore, there were attempts to introduce an energy for minimizing the surface area of the fluid [HWZ*15]. It was found by Yang et al. [YML*17] that increasing the support radius of the SPH kernel and a pair-wise force formulation increases the stability and accuracy of interface-dominated phenomena, such as surface tension and adhesion. However, this comes at the cost of significantly increased computational effort. Zorilla et al. [ZRS*20] presented an accurate computation of surface curvature using Monte Carlo integration. Wang et al. [WDK*21] used the SPH discretization method to introduce a method for the simulation of thin films. To do this, they simulate a height field on bubbles consisting of a single layer of SPH particles. Local surface reconstruction was also used to more accurately compute surface normals and coupled with a position-based fluid approach [XRW*22]. Jeske et al. [JWL*23] instead proposed an implicitly integrated model based purely on cohesion forces, in order to simulate very large surface tension coefficients. Finally, Wang et al. [WJL*20] pursued an MLS-based co-dimensional approach, where thin-films and filaments of fluids are treated individually. This allows them to achieve greater accuracy on surface-driven phenomena, such as surface tension.

Finally, solutions have been proposed to use SPH, partially in combination with other methods, to solve the shallow water equations (SWE). Solenthaler et al. [SBC*11] discretize 2D shallow water equations on particles instead of uniform grid cells, offering benefits common to particle discretization like simplification of sparsely filled domains and easy interaction with arbitrary bound-

ary geometry. Chentanez et al. [CM10] develop a shallow-water solver that spawns particles in regions difficult to capture with a height field, such as breaking waves and waterfalls, where these particles represent spray or foam independently before merging back into bulk fluid. By coupling a particle level-set method (FLIP) with an SPH solver, Losasso et al. [LTKF08] show simulations of diffuse phenomena like mixtures of water spray and air. Finally, Chentanez et al. [CMK15] improve on this approach by using a multi-resolution fully-adaptive approach, enabling the simulation of larger domains. They combine an Eulerian solver with PBF or SPH, while allowing for different time step sizes among particles and the grid.

4.2.2. Rigid Bodies and Frictional Contact

Coupling fluids or deformable solids with static or dynamic rigid bodies is an essential topic in order to enable dynamic and expressive multiphysics simulations. A common approach is to sample the surface of a rigid body using particles. These particles can then be used to compute explicit interaction forces [BTT09] or, more commonly, be used as additional sampling points in the density and pressure force computations [AIA*12]. Akinci et al. [ACAT13] expand on this approach by dynamic resampling to enable coupling with thin structures such as cloth. Gissler et al. [GPB*19] extend the particle-based approach to also resolve contacts with friction between rigid bodies using the surface particles. This enables a strong rigid-fluid coupling since fluids and rigid bodies are handled in one global SPH solver. Probst and Teschner [PT23] further improve this strong coupling by introducing a more realistic friction handling.

An alternative to the particle-based approach is to use an implicit boundary representation [FM15, KB17, BKWK20, WAK20]. Fujisawa and Miura [FM15] propose a fast approach to compute the integral of the kernel within the boundary for triangular meshes and integrate this into a PBF solver. Koschier and Bender [KB17] use a fixed grid and precompute the density contribution of the boundary in the support radius of each grid point. In the simulation the boundary contribution for each particle can be determined easily by interpolating the grid values of the cell which contains the particle. This approach was later improved by precomputing the volume intersection of the particle domain and the boundary instead of the density [BKWK20]. Finally, Winchenbach et al. [WAK20] compute boundary kernel contributions by defining a locally representative planar boundary for arbitrary geometries defined by signed distance functions and deriving an analytic solution.

When using additional sampling points, the standard SPH formulation commonly used in pressure solvers requires a pressure value at each sampling point. Different strategies were developed to determine these values: mirroring the pressure values from the corresponding fluid particles [AIA*12], considering the additional sampling points in the linear system solver [BGI*18] or extrapolating the pressure values from the fluid [BGPT18]. However, recently it was demonstrated that an additional boundary pressure computation can be avoided entirely by a reformulation of the pressure solver [BWRJ23].

4.2.3. Elastic and Elastoplastic Materials

Deformable solids are often simulated using mesh-based methods since they can be more efficient for this application than meshless approaches. However, point-based methods like SPH have the advantage that a unified representation for fluids and solids facilitates coupling between different materials, the simulation of state transitions like solidification or melting, as well as topology changes.

Elasticity A common way to simulate a deformable solid using SPH is to first determine the particle displacement $\mathbf{u} = \mathbf{x} - \mathbf{x}^0$ as the difference of its current and its initial position [SSP07]. Then, the SPH formulation is used to compute the gradient of the displacement field $\nabla\mathbf{u}$ and the deformation gradient $\mathbf{F} = \nabla\mathbf{u} + \mathbb{1}$, where $\mathbb{1}$ denotes the identity matrix. This typically uses a fixed particle neighborhood from the rest state. Becker et al. [BIT09] propose a corotational approach to determine the gradients, which ensures a proper handling of rotations. The strain can be calculated using the deformation gradient, and the stress is obtained by applying a constitutive model. Finally, symmetric forces are determined from the stress to simulate elastic behavior.

The methods of Solenthaler et al. [SSP07] and Becker et al. [BIT09] are based on a conditionally stable explicit time integration. To improve the stability, Peer et al. [PGBT18] propose to apply a corotated linear elasticity model in combination with an implicit Euler time integration. Kugelstadt et al. [KBF*21] improved the performance significantly by splitting the model in a stretch and a volume term. The separate consideration of the stretching term leads to a linear system with a constant system matrix which can be solved very efficiently using a precomputed Cholesky factorization. While the aforementioned implicit methods are based on linear elasticity models, Kee et al. [KUKH23] propose an implicit time integration for non-linear models using a quasi-Newton method.

Elastoplasticity Inspired by the work of Zhu and Bridson [ZB05], elastoplastic material models were investigated in the field of SPH to simulate granular materials. The core idea is that elastic material behavior is used to simulate static friction, and if a yield condition is met, the material can start to flow, deforming plastically.

Lenaerts and Dutré [LD09] also use an elastoplastic model to simulate granular materials. In their explicit solver, plasticity is modeled using the Mohr-Coulomb yield criterion. Alduán and Otaduy [AO11] propose a similar approach but use the Drucker-Prager yield criterion and introduce an implicit solver. Ihmsen et al. [IWT13] extend this method by an upsampling technique to increase the detail. Later, Gissler et al. [GHB*20] introduced an SPH snow simulation based on an elastoplastic material model in combination with an implicit pressure solver for compressible fluids.

4.2.4. Multiphysics Systems

Besides the previously presented methods for modeling individual effects, which can readily be coupled in unified particle-based discretizations, there have been recent contributions to modeling multi-physical systems in general. These contributions generally outline improvements to the fundamental particle-based methodology which can benefit the simulation accuracy and stability of multiple physical phenomena in a unified discretization.

Solenthaler et al. [SSP07] early on proposed a system that allowed for coupled simulations of fluids, rigid bodies and deformables. This included a simple model for phase-change including heat-transfer within and across different phases, allowing for effects such as melting, joining and solidification. Interaction between different materials was handled by the pressure solver. An improved system is presented by Yang et al. [YCL*17], in which all of the aforementioned effects can be simulated, but can further be combined with granular materials and plasticity, as well as multiple phases including solubility with continuous interfaces. Instead of using a unified discretization method, Xie et al. [XLYJ23] introduce an approach with contact proxies to couple SPH fluids and deformables (see Section 2.2).

5. Eulerian and Hybrid Methods

Thus far, we have focused on simulations methods with a Lagrangian viewpoint. This class of approaches treats a physical continuum like a particle system, where each bit of matter in a fluid or solid is represented by a separate particle with position and velocity and other physical quantities.

Eulerian simulations, on the other hand, track a set of fixed points in space and how physical quantities at these points evolve over time, e.g., using a uniform grid. The Lagrangian and Eulerian viewpoints are connected through the *material derivative*, which for some physical quantity q can be expressed as

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \mathbf{v} \cdot \nabla q \,. \tag{97}$$

The first term on the right hand side is simply the rate at which q is changing at a specific point in space, whereas the second term is an advective derivative that gives the change due to movement through a velocity flow field $\mathbf{v}$.

We have seen the material derivative before. For instance, in Equation (91) the material derivative of the velocity of a fluid flow field appears in the Navier-Stokes equations. In the Lagrangian view, since physical quantities move with particles in the simulation, the advective derivative is not needed, and the material derivative is simply the time derivative of the velocity, $\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t}$. However, since the Eulerian view measures quantities at fixed points, the advective term of the Navier-Stokes equations $\mathbf{v} \cdot \nabla \mathbf{v}$ must be considered.

In practice, Eulerian formulations are attractive because they turn spatial operators into local stencil computations on a fixed grid. Before discussing specific phenomena, we briefly summarize the geometric representation and discretization pipeline that underlies many such methods.

5.1. Eulerian Discretization Basics

Building on the Eulerian viewpoint above, we now briefly revisit how continuum equations are turned into computable algebraic systems on a fixed grid. A standard Eulerian setup introduces a bounded simulation domain $\Omega \subset \mathbb{R}^d$ and partitions it into cells of size Δx (and analogously $\Delta y, \Delta z$ in higher dimensions). Using regular grids is typical, but irregularly sampled tetrahedral meshes can also be supported. Scalar or vector fields are then sampled at predefined locations (cell centers, faces, or nodes). For fluid simulation, a common choice is a staggered marker-and-cell (MAC) layout, where pressure is stored at cell centers and velocity components are stored at face centers [HW*65, Bri15].

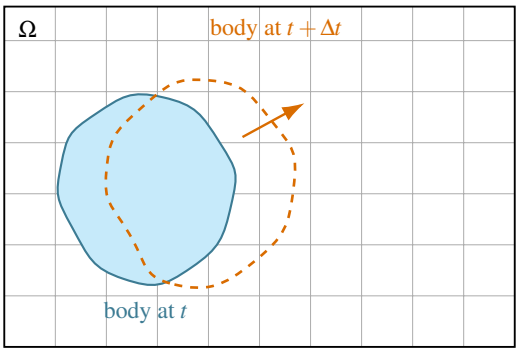

Figure 5: *Conceptual illustration of a moving and deforming body within an Eulerian geometric representation on a fixed grid.*

Geometric representation. In the Eulerian view, a continuum body moving through space is represented implicitly through fields defined on the fixed background grid. For example, one may use an indicator or level-set field

$$\phi(\mathbf{x},t) = \begin{cases} < 0, & \mathbf{x} \text{ inside the body,} \\ = 0, & \mathbf{x} \text{ on the interface,} \\ > 0, & \mathbf{x} \text{ outside the body.} \end{cases} \tag{98}$$

The interface is recovered as the zero isosurface of ϕ, while transport and deformation are captured by the evolution of the field values. This is a common way to implement free-surface flows in the Eulerian viewpoint and is referred to as level-set method (LSM).

Finite differences and relation to FVM/FEM. A direct way to discretize PDEs on structured grids is the finite difference method (FDM), which approximates derivatives by local stencil operations. For a sufficiently smooth scalar field $q(x)$ sampled at $x_i = i\Delta x$, a first derivative can be approximated as

$$\left.\frac{\partial q}{\partial x}\right|_i \approx \frac{q_{i+1} - q_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2). \tag{99}$$

Compared with FDM, finite volume methods (FVM) emphasize conservation through flux balances over control volumes, while finite element methods (FEM) use weak formulations and basis functions on elements [TBHF03, Bat06]. While different in fundamental approach, on regular grids, these viewpoints are closely related and often lead to similar sparse linear systems.

Canonical stencils in 1D, 2D, and 3D. In one dimension, the second derivative is approximated by

$$\left.\frac{\partial^2 q}{\partial x^2}\right|_i \approx \frac{q_{i-1} - 2q_i + q_{i+1}}{\Delta x^2}. \tag{100}$$

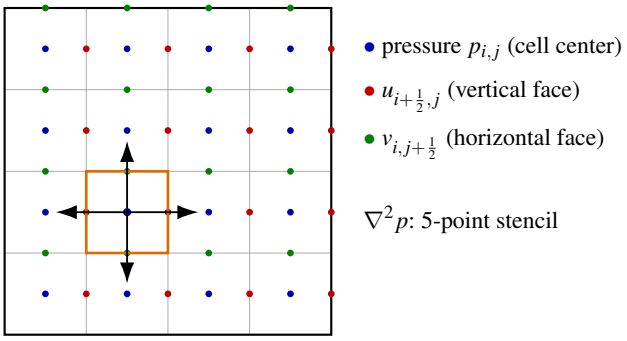

Figure 6: *2D MAC grid with finite-difference stencil evaluation points.*

In two dimensions, the standard 5-point Laplacian stencil is

$$\nabla^2 q\Big|_{i,j} \approx \frac{q_{i+1,j} - 2q_{i,j} + q_{i-1,j}}{\Delta x^2} + \frac{q_{i,j+1} - 2q_{i,j} + q_{i,j-1}}{\Delta y^2}. \tag{101}$$

In three dimensions, this extends to the 7-point stencil

$$\nabla^2 q\Big|_{i,j,k} \approx \frac{q_{i+1,j,k} - 2q_{i,j,k} + q_{i-1,j,k}}{\Delta x^2} \tag{102}$$

$$+ \frac{q_{i,j+1,k} - 2q_{i,j,k} + q_{i,j-1,k}}{\Delta y^2} \tag{103}$$

$$+ \frac{q_{i,j,k+1} - 2q_{i,j,k} + q_{i,j,k-1}}{\Delta z^2}. \tag{104}$$

Most stencils can be derived from Taylor expansion around a specific point, such that a variety of stencils exist for special cases. For example, boundary nodes cannot utilize central finite difference methods. The stencil may also have to be adjusted depending on the imposed boundary condition.

Example PDE. As a representative model problem, consider advection-diffusion of a scalar quantity q:

$$\frac{\partial q}{\partial t} + \mathbf{u} \cdot \nabla q = \kappa \nabla^2 q + s, \tag{105}$$

where $\mathbf{u}$ is a prescribed velocity field, κ is the diffusion coefficient, and s is a source term. This equation contains the two dominant operators that also appear in many fluid simulation substeps: transport and elliptic diffusion/projection.

Semi-discretization via the method of lines. The most common approach to discretize space-time PDEs is to use the so-called method of lines. With the method of lines, one first discretizes only space and keeps time continuous. Using a grid vector $\mathbf{q}(t)$ containing all sampled values of q, Equation (105) becomes

$$\frac{d\mathbf{q}}{dt} = \mathbf{L}(\mathbf{q}) + \mathbf{s}, \tag{106}$$

where $\mathbf{L}(\cdot)$ is the discrete spatial operator assembled from advection and diffusion stencils. For linearized advection and constant coefficients, one obtains

$$\frac{d\mathbf{q}}{dt} = \mathbf{A}\mathbf{q} + \mathbf{s}, \tag{107}$$

with sparse matrix $\mathbf{A}$ induced by local stencil couplings. A time integrator (e.g., forward Euler, Runge-Kutta, backward Euler) then advances this ODE system.

Resulting algebraic systems. An explicit step has the form

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \left(\mathbf{A}\mathbf{q}^n + \mathbf{s}^n\right), \tag{108}$$

while an implicit backward Euler step gives

$$(\mathbf{I} - \Delta t\, \mathbf{A})\, \mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t\, \mathbf{s}^{n+1}. \tag{109}$$

Hence, the simulation loop repeatedly alternates between local stencil evaluations and the solution of sparse linear systems, exactly the structure encountered in pressure projection for incompressible flow [Bri15].

Strengths, limitations, and practical extensions. Structured-grid discretizations are simple, robust, and map incredibly well to parallel hardware because each stencil operation is local and regular. This is one reason Eulerian solvers are highly effective on GPUs and many-core architectures. At the same time, limitations include numerical dissipation (especially in low-order advection), CFL restrictions for explicit time stepping, and the need to predefine the simulated spatial extent [Sta99, FM96]. This problem also causes free-surfaces reconstructed from a LSM to be afflicted by smearing artifacts, where the interface of the fluid becomes blurrier with passing time. Practical extensions to Eulerian methods include adaptive grids, higher-order advection schemes, and hybrid particle-grid methods.

5.2. Fluid Dynamics

Eulerian methods are perhaps most well known in graphics for simulating fluid dynamics. In this setting, a grid stores physical and flow field quantities that are typically updated with an operator splitting approach in a three step process:

1. Update velocities due to external forces.
2. Compute a pressure projection to make the fluid incompressible.
3. Advect the velocity field and any physical quantities.

Velocity updates are guided by the Navier Stokes equations as shown in Equation (91). The second step is arguably the most computationally expensive, especially for higher resolution grids, since it requires solving a pressure Poisson equation (see Equation (95)).

The grid structure used for storing pressure and flow field quantities is an important consideration. Staggered grids, where velocities are stored at the center of cell faces and all other physical quantities are stored at cell centers, are often preferred for their stability when using finite differences to evaluate spatial derivatives. The marker-and-cell (MAC) grid [HW*65] is a popular choice. Furthermore, work on efficient grid data structures and update algorithms is an active area of research [SABS14, Mus13, KLM24, WHS*24, BBT25].

The reader is directed to the excellent reference material by Bridson [Bri15] for further background material on Eulerian fluid simulations. However, since our focus is on the multiphysics setting, we next consider how Eulerian fluid simulations can be coupled with other fluid phases, deformable objects and rigid bodies.

5.3. Solids

Eulerian representations have proven effective for simulating elastic solids undergoing large deformations. The regular grid structure used by many implementations avoids expensive remeshing steps, and further facilitates parallel implementations. Levin et al. [LLJ*11] discretize the momentum equations on a grid and use the finite volume method [TBHF03] to compute the traction of each cell from a Cauchy stress field. Their approach uses an explicit scheme to model interactions between colliding bodies. However, Teng et al. [TLK16] later extended the method with an implicit scheme allowing larger time steps, and further introduced fluid-solid coupling in a purely Eulerian regime.

5.4. Multi-phase Fluids

One of the advantages of using an encapsulating Eulerian grid, is that it becomes possible to capture multi-phase effects, particularly those between a liquid phase and an air phase. This interaction is at the core of many real-world phenomena, such as water with bubbles. Within the Eulerian regime of approaches, the ghost fluid method has been popular for modeling the jump in fluid densities [HK05, BB12]. Other approaches range from using stream functions [ATW15] and explicit bubble constraints [GAB20], to using the lattice Bolzmann method (LBM) [LMLD22].

5.5. Fluid-Solid Coupling

Many works in computer graphics have addressed the subject of fluid-solid coupling. A central problem here is representing the solids and their boundaries in grid data structures. For instance, early work by Foster and Metaxas [FM96] demonstrated grid-aligned one-way coupling using a MAC grid. This approach requires rasterizing the solid onto the grid in order to determine the boundary interface, thus giving a one-way coupling. Takahashi et al. [TUKF02] achieve two-way coupling by assigning zero Neumann boundary conditions for pressures of any grid cell more than half filled with a solid. Cell velocities covered by a solid are then assigned velocities from the solid, although the approach cannot couple forces and torques due to fluid momentum.

The immersed boundary method [Pes02] is another popular technique that gives a continuous forcing function that allows the fluid pressure field to change the solid's velocity by considering the solid to be part of the fluid. Similarly, Carlson et al. [CMT04] fixes the solid density to be the same as the fluid and enforces a rigidity constraint for velocities inside the solid using Lagrange multipliers. Guendelman et al. [GSLF05] proposed a solid-fluid coupling approach specialized for thin shells and cloth. The immersed boundary method has also recently been applied to turbulent flow simulation using the lattice Boltzmann method (LBM) [LCD*20].

Cut-cell methods are another approach for computing interactions between Eulerian fluids and solids. They are popular due to their simplicity and ability to encode fluxes due to geometric details smaller than the grid resolution, but without having to refine or change the grid structure. The area of grid cell faces covered by the solid boundary is used to improve the accuracy of the divergence calculations. These approaches were first used in computer graphics by Roble et al. [RZF05] and are subject of ongoing research [ZB17, TBFL19]. Batty et al. [BBB07] accounts for partial grid cell overlap by approximated volume weights, which are used by a variational formulation of the pressure projection solve that improves accuracy of sub-grid fluid velocities near object boundaries. Cut-cell methods are also used in combination with mesoscale approaches for more accurate coupling between fluids and solids [LLDL21]. Shi et al [SZYA24] proposed a coupling framework that target heterogeneous computing platforms combining a cut-cell method and adaptive grids for interactive simulation of rigid bodies and fluids.

Two-way coupling between fluids and solids often necessitates taking small time steps and using many coupling iterations to achieve stable simulations with correct pressure and momentum exchange between the two models [CMT04, GSLF05]. Achieving strong two-way coupling requires accounting for both the solid and fluid dynamics during the pressure projection step. However, coupling techniques can sometimes result in non-physical momentum transfer, even though fluid-solid boundary conditions are satisfied. Robinson-Mosher et al. [RMSG*08] proposed a modification to the momentum update used by Batty et al. [BBB07] that improves fluid-to-solid momentum transfer for thin solids and shells by a mass-lumping strategy that combines equations for both fluids and solids in mixed cells. Follow-up work resolved problems with this approach, allowing for correct slip conditions of tangential velocities [RMEF09]. Klingner et al. [KFCO06] proposed an implicit approach for rigid-smoke coupling, and later Chentanez et al. [CFL*07] extended the approach to liquids, that discretizes the domain using an adaptive tetrahedral mesh, which allows finer details to be preserved due to more accurate treatment of boundary conditions. However, the approach is computationally costly due to frequent remeshing for dynamic objects. Chentanez et al. [CGFO06] proposed a unified approach for strong coupling of fluids and deformable objects, although it results in a non-symmetric linear system that requires a more costly solver algorithm. Stable coupling can be achieved, however, without monolithic frameworks that solve for pressures and velocities across the fluid and solid domains. For instance, Akbay et al. [ANZS18] showed that a partitioned impulse-based solver using a reduced model interface was effective for stable simulation using independent "black-box" fluid and solid solvers, requiring only Jacobian approximations collected from simulation data.

More recently, Takahashi and Batty [TB20] developed the Monolith solver for fluid-rigid coupling, and later extended the approach for elastic bodies [TB22]. While most work focuses on fluid-solid interactions of individual dynamical bodies, their approach also accounts for interactions between solids, for instance due to frictional contact, as well as viscosity. A hybrid approach, they combine an Eulerian fluid simulation with Lagrangian mesh-based elastic bodies, and rigid bodies using a strong coupling approach formulated as a unified constrained optimization problem. Although their fluid simulator uses a particle-in-cell method, coupling is solved using a grid, and hence can be considered an Eulerian approach. However, hybrid approaches combining grid and particle representations have gained prominence in computer graphics, in particular the Material Point Method, which we consider next.

5.6. Material Point Method

The Material Point Method (MPM) is a hybrid Eulerian/Lagrangian method. Hybrid methods inherit the strengths from both material descriptions: Lagrangian particles make advection trivial, while the underlying Eulerian grid eases the computation of spatial derivatives. As one of the first hybrid methods, the Particle-In-Cell method (PIC) was used to simulate compressible fluid back in the 1960s [Har62, HW*65]. Due to frequent grid-particle transfers, PIC yields excessive dissipation and the resulting simulation appears highly dampened. This issue was later fixed by Fluid-Implicit-Particle (FLIP) [BR86], which only averages the velocity and displacement increments during the grid-particle transfer. Zhu and Bridson [ZB05] mixed PIC and FLIP with a blending weight to control the viscous damping and instability. The Material Point Method further generalizes PIC/FLIP [SZS95] to solid mechanics by allowing particles to carry additional physical quantities such as the deformation gradient. MPM-based approaches have been firstly introduced to the graphics community for snow animation [SSC*13], and quickly generalized for a multitude of materials and phenomena (see Figure 7). After a brief summary of MPM in the following section, we will provide details on several of its applications for multiphysics modeling.

5.6.1. Foundations of the Material Point Method

Analogous to the Finite Element Method (FEM) [Bat06], the Material Point Method is derived from the weak form of the conservation of momentum, making it an accurate modeling framework for the discretization of constitutive laws [JST*16]. Furthermore, its hybrid Lagrangian/Eulerian setting has various advantages, which makes it excel in handling phenomena involving topological changes and very large deformations. The Lagrangian representation is ideally suited for large deformations as the material and its quantities are effectively tracked by persistent particles [SZS95]. The Eulerian representation is only transient and is used for computing spatial derivatives and solving the material's governing equations. The resulting velocity field triggers the movements of particles. Self-collision and fracture can thus be efficiently modeled on the Eulerian background grid, independent of the number of particles.

In order to make use of both representations, MPM needs to constantly transition information between Lagrangian particles and the Eulerian grid nodes. As such, while the exact details in different MPM-based method implementations vary, a basic symplectic Euler MPM algorithm in its simplest manifestation performs the following operations in each time step:

1. *Particle-to-Grid (P2G) Transfer.* Transfer particle quantities to the grid, computing mass and momentum at grid nodes. Different P2G transfer schemes can be used in this step, such as Affine Particle-In-Cell (APIC) [JSS*15].
2. *Grid Velocity Computation.* Compute velocities at grid nodes from the mass and momentum computed above.
3. *Explicit Grid Force Computation.* Compute forces acting on grid nodes due to nearby particles.
4. *Grid Velocity Update.* Update grid node velocities based on the forces computed above and the grid node masses, while accounting for boundary conditions or specific collision handling.
5. *Deformation Gradient Update.* Update the deformation gradient for each particle based on the grid velocity field.
6. *Grid-to-Particle Transfer (G2P).* Transfer velocities from the grid to the particles using a specific transfer scheme. Depending on the scheme employed in the G2P and P2G steps, additional information needs to be transferred here.
7. *Particle Advection:* Advect particles using their new velocities.

For more information on the foundations of MPM we recommend the course by Jiang et al. [JST*16].

5.6.2. Accuracy, Stability and Performance Improvements

Several notable improvements of MPM have made the framework more attractive over the years. With the development of the Affine Particle-In-Cell (APIC) method, proposed by Jiang et al. [JSS*15] and later generalized by Fu et al. [FGG*17] with Polynomial Particle-In-Cell (PolyPIC), grid-particle transfer better conserves affine momentum and kinetic energy while maintaining the stability of the simulation. The Power-Particle-In-Cell further takes particle volume into consideration [QLDGJ22], which leads to better volume preservation. Recently, Sancho et al. [STBA24] proposed the Impulse Particle-In-Cell Method (IPIC) to better conserve circulation and vortical details for fluids using flow maps.

Explicit time integration in MPM requires small time steps adhering to the CFL condition for feasible stability [SSS20] [WLF*20]. For details, we refer to the stability analysis of Bai and Schroeder [BS22]. Stomakhin et al. [SSC*13] propose a time integration scheme to enable semi-implicit and implicit time integration, allowing the use of large time steps and thereby increasing efficiency of MPM. Hyde et al. [HGMRT20] adopt the implicit MPM formulation to simulate surface tension effects. Fang et al. [FLGJ19] use implicit MPM to handle viscoelastic and elastoplastic solids. Implicit MPM also contributes strong coupling between fluids and solids [FLGJ19]. Gast et al. [GSS*15] present an optimization-based formulation of implicit integration in MPM. Combining the Moving Least Squares (MLS) material sampling method (see Section 4) with MPM, Hu et al. [HFG*18] enable two times faster simulation rates, with a simpler implementation and with visually comparable results. Jiang et al. [JGT17] leverage MPM for processing frictional contacts. Gao et al. [GTJS17] design an adaptive grid scheme to reduce the collision gap for MPM deformables. Ding and Schroeder [DS19] show the feasibility of combining MPM with Coulomb frictional contacts for accurate coupling with rigid bodies. For further performance gains, prior works also explore GPU-based MPM implementations [GWW*18, WQS*20, FHG21, QRL*23].

5.6.3. Multiphysics Phenomena

Multiphysics simulation is naturally supported in MPM by assigning different constitutive laws and material models to different particles [JST*16]. Thanks to its hybrid discretization, MPM also exhibits a high versatility in robustly handling various phenomena ranging from snow [SSC*13], phase-changing materials [SSJ*14, TLZ*24, SXH*21], foams [RGJ*15, QLY*23, YSB*15] to granular materials [DBD16, KGP*16, TGK*17], and even magnetized materials [SNZ*21]. The Lagrangian particles in MPM are insensitive to topological changes. Therefore, MPM has also been a popular choice for cutting [HFG*18] and fracture [WFL*19, WDG*19,

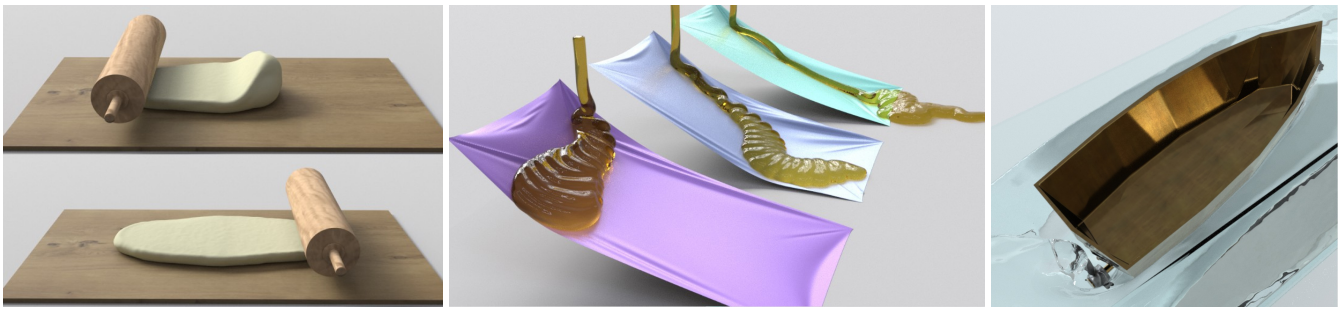

Figure 7: *MPM simulations of various interacting materials [LLH*24]. **Left:** Plastic dough material being flattened by a rigid, rolling pin. **Middle:** Highly viscous honey dropped on a cloth. **Right:** Two-way coupled FEM-MPM simulation of a motor boat interacting with water.*

FCK22, CCL*22]. In the following, we will provide brief descriptions of a representative selection of these methods.

Stomakhin et al. [SSC*13] integrated an elasto-plastic constitutive model with MPM for modeling the dynamics of snow. A particle carries an elastic deformation gradient and a plastic deformation gradient. Out-of-range deformation accumulates in the plastic portion, while elastic deformation yields internal forces. Possible topology changes, caused by snow breaking or merging, are easily captured with the meshfree Lagrangian description, and the Eulerian background grid enables implicit handling of self-collisions and fracture. This feature inspires several follow-up works to use MPM to handle fracture [WFL*19, WDG*19, FCK22].

Shortly after, Stomakhin et al. [SSJ*14] augmented MPM with heat transport on the grid so that particles with different temperatures are assigned different material properties. A deviatoric/dilational splitting of the constitutive model allows phase transitions during the simulation, such as melting and solidifying, which further enriches the MPM ecosystem.

Using MPM to simulate Non-Newtonian fluid has also been explored. Ram et al. [RGJ*15] employed a modified Oldroyd-B model to approximate volume-preserving plastic flows. Yue et al. [YSB*15] demonstrate that Herschel–Bulkley fluid can also be handled within the MPM framework with particle resampling. Su et al. [SXH*21] proposed an extended POM-POM model [ML98, OMTA12, VPB01] that can uniformly handle a range of viscoelastic liquids with phase change.

Early MPM frameworks often chose explicit grid-level integrations for spatial differentiation of internal elasto-plastic forces. This simplification requires a highly conservative time step size for stiff simulation problems, e.g., granular materials with a high Young's modulus. Daviet et al. [DBD16] further enhanced the stability of MPM for such cases by treating both elastic and plastic parts of the dynamics in a semi-implicit manner. Other examples of MPM-based granular material simulation are the methods by Klar et al. [KGP*16] and Tampubolon et al. [TGK*17]. To implicitly handle plasticity in optimization-based MPM frameworks, Li et al. [LLJ22] present strain energies for specific plasticity models (see also Section 2.2).

Hu et al. [HFG*18] employ Moving Least Squares (MLS) interpolation in their MLS-MPM method for the unified simulation of fluids, elastic, plastic and granular materials with robust support for cutting. Any external rigid body dynamics simulator can be combined with their method, providing two-way coupling with rigid bodies. Qu et al. [QLY*23] propose an extension of MLS-MPM which is capable of accurately capturing both macro- and mesoscale material geometries. They demonstrate the versatility of their approach with simulations of bubbles, sands, liquid and foams, and show that in contrast to traditional MPM, their method does not suffer from minimum particles-per-cell restrictions and is able to faithfully conserve material volume, all the while producing more uniform particle distributions.

Tu et al. [TLZ*24] combine MPM with a phase field solver for the unified simulation of elastic, plastic and viscous materials that undergo phase transitions. Their method is able to represent interacting fluids, deformable and rigid solids, and granular materials which can dissolve and melt, all in one unified framework. This is achieved through blending between different constitutive models in a unified particle representation. Collisions between different materials not undergoing phase changes was modeled more realistically by Gao et al. [GPH*18] using MPM, although not in a unified manner, as we will see in the next section. The approach by Tu et al. [TLZ*24] also inherits a common limitation in MPM, which causes inaccuracies in contact behavior at multi-material interfaces – a consequence of stickiness and numerical viscosity, leading to excessive cohesion and friction. This limitation was addressed in the modified Material Point Method introduced by Han et al. [HGG*19], details of which are provided in the next section. Chen et al. [CKMR*21] propose an implicit model for spatially varying surface tension in MPM. The surface tension coefficient can be dependent on local concentrations or temperatures and for the latter it is coupled with an implicit thermomechanical model to accurately simulate melting.

5.6.4. Coupling Techniques

In order to address the aforementioned shortcomings with collision modeling in traditional MPM manifesting in artifacts such as excessive cohesion and friction, Han et al. [HGG*19] introduce a modified Material Point Method which makes use of collision particles that are uniformly distributed over the surface of volumetric solids. This approach prevents information loss during particle-to-grid transfers and enables accurate Coulomb frictional contact

modeling between volumetric solids. The method also effectively resolves the dependency issue found in traditional MPM between Eulerian grid size and Lagrangian particle density. Through use of two separate Eulerian background grids, the method supports coupling with traditional MPM and also provides a simple method for two-way coupling with rigid bodies.

Gao et al. [GPH*18] propose an MPM-based method for simulating the interaction of granular materials with fluids through explicit two-way coupling. The authors use a density criterion to identify individual grains that become separated from the continuum body of the granular material which are treated as separate bodies, not subject to the elastoplastic Drucker-Prager constitutive law used to model still connected granular material clumps. The fluid and granular material domains are advanced in a sequential manner, with sub-stepping used for the granular material. The method captures intricate details in fluid/granular material interactions such as sediment transport in particle-laden fluid flows which can not be reproduced in the unified method proposed by Tu et al. [TLZ*24].

Guo et al. [GHF*18] present a method for simulating elastoplastic thin shells based on an FEM subdivision surface discretization within MPM and demonstrate frictional contact and coupling to other MPM materials. To simulate fracture due to shocks in compressible flow, Cao et al. [CCL*22] developed a monolithic solve for strongly coupled MPM and FEM. This velocity-pressure system is discretized with a mixed-order finite element of B-spline shape functions. More recently, for two-way coupling between MPM and FEM of possible co-dimensional geometry, Li et al. [LLH*24] propose an asynchronous time-splitting scheme. Specifically, an implicit integration scheme is used at the internal region of the FEM domain. The contact force is applied as a constant external Lagrangian force on MPM particles throughout substeps of explicit MPM integration.

6. Multiphysics Frameworks

This section presents a selection of frameworks for multiphysics simulation which make use of the mathematical models, coupling techniques and discretizations discussed in this report. These frameworks offer convenient out-of-the-box simulation capabilities via ready-to-use interfaces which allow easy deployment of simulations or extensions of the frameworks' capabilities. Some frameworks employ a single unified model while others use a modular approach and combine different formulations and discretizations through coupling techniques and expose them in a unified simulation suite. We will briefly discuss several popular frameworks in this section.

Chrono Tasora et al. [TSM*16] propose the *Chrono* framework which enables fully coupled multiphysics simulations of rigid and deformable bodies, fluids and granular materials. It is designed for use in engineering applications, and, depending on the resolution and scale of the simulation, can achieve real-time frame rates, or can be used for accurate offline simulations. The framework is freely available with source code and documentation accessible at `https://projectchrono.org`.

The framework combines a number of simulation models and provides out-of-the-box coupling for their interactions. Constrained rigid bodies are represented using a Nonsmooth Multidomain Dynamics (NMD) formulation. Granular materials are represented using either a nonsmooth DEM approach within NMD or via a penalty-based method. Fluids are modeled using SPH and coupled with solids using boundary particles. Rods, sheets and thin beams are represented using the Absolute Nodal Coordinate Formulation (ANCF), and for volumetric flexible elements a corotational FEM approach is employed. The advantage of using ANCF is that it can be embedded within the mathematical framework of NMD and thus provides strong coupling without need for any explicit coupling technique [TSM*16].

SOFA With *SOFA*, Faure et al. [FDD*12] propose a versatile, extensible framework based on multiple overlayed data models, providing flexible data flow and algorithm control that targets interactive and real-time multiphysics simulation for virtual surgery training. SOFA is an open-source framework with documentation and source code available at `https://www.sofa-framework.org`.

The framework provides a readily available multi-material simulation system. Fluids are modeled using SPH. Deformable bodies are simulated using corotational FEM, which interact with rigid bodies through unilateral contact constraints, solved using a projected Gauss-Seidel method. In addition, the framework offers various linear solvers and constraint solvers supporting both bilateral and unilateral constraints which can be employed to any simulation problem.

Loki Entirely tailored to offline simulations for the film industry, the Loki multiphysics framework, proposed by Lesser et al. [LSD*22], employs a generic coupling approach that uses interleaved Newton iterations of the individual methods at small time steps, which ensures a high level of solution accuracy. This approach also allows for the use of best-in-class methods for simulation of different material types, while avoiding combinatorial explosions from the use of an abundance of specialized coupling methods for the various pair-wise material type interactions. This is made possible through the use of generic and common material descriptions shared by all methods, such as Lagrangian particles and Eulerian grid nodes carrying common physical material quantities. For increased performance and scalability, Loki uses heterogeneous computing and distribution of workloads via MPI. Loki is a commercial framework that is currently not available to the public.

STARK was introduced by Fernández-Fernández et al. [FFLL*24] as an open-source framework with state-of-the-art methods for energy-based simulations as outlined in Section 2. In this original work, the framework is qualitatively validated in challenging scenarios of friction-dominated robot-cloth interactions. STARK features implementations of widely used models for deformable solids, shells, rigid bodies with joints and motors as well as an IPC-based model for frictional contact. For the numerical solution it uses Newton's method with line search. The framework can be used as a black box simulator using high-level C++ and Python APIs. However, it is also well

suited for research and development of new models as it uses the symbolic differentiation engine *SymX* [FFLW*25] to handle the differentiation and assembly components. This makes it possible to add new models by only specifying a scalar energy potential in symbolic form. STARK has already been used in several recent publications that demonstrate the advantages of energy-based strong coupling for multiphysics applications [LFFJB24, WFFJB24, LFFJ*23]. The source code is available at https://github.com/InteractiveComputerGraphics/stark.

SPlisHSPlasH [B*26b] is a research-driven open-source SPH simulation framework. The implementation contains many of the state-of-the-art methods discussed in Section 4.1.1, including most pressure solvers and boundary handling approaches as well as implicit methods for elasticity, viscosity and surface tension. Its architecture supports weak coupling in both a unified and non-unified way through interaction forces. It is used by a multitude of recent works either as a data-generation tool for learning methods, or as a reference implementation for comparison purposes. In recent years, it has also found application in the engineering community, e.g., for welding and thermally-sprayed coating simulations [JSS*22, JBB*22]. It is actively being developed and provides a Python interface for easy access and usability, with source code available at https://github.com/InteractiveComputerGraphics/SPlisHSPlasH.

NVIDIA Flex [NVI24] is a freely available multiphysics framework for the real-time time simulation of various materials including rigid and deformable bodies and fluids on the GPU. The framework is tailored for use in video games and VR. Its implementation is based on the unified position-based dynamics method proposed by Macklin et al. [MMCK14]. Due to its lack of various functionalities required in gameplay mechanics such as collision queries and callbacks for integration with game applications, manual coupling of Flex with CPU-based rigid body dynamics engines commonly used in games might be required. NVIDIA Flex provides access to source code which is available at https://github.com/NVIDIAGameWorks/FleX.

Open-Source MPM Simulators

Various open-source multiphysics frameworks based on MPM (see Sec. 5.6) are available.

Karamelo [dVNNT21] provides a lightweight C++ framework for unified CPU-based multiphysics simulations using MPM, designed to be easy to modify in order to facilitate research conducted on and with MPM. Parallelization is offered through use of MPI. Documentation and source code are available at https://karamelo.org.

With *Taichi*, Hu et al. [HFG*18] make the implementation of their method freely available at https://github.com/yuanming-hu/taichi_mpm. Taichi provides a high performance CPU implementation of MLS-MPM with cutting support and two-way coupling with rigid bodies that offers a convenient Python API.

A GPU implementation of MPM using CUDA is provided by Gao et al. [GWW*18] that offers support for multiple particle-to-grid transfer schemes, FLIP, APIC and MLS, and enables both implicit and explicit time integration. The source code can be found at https://github.com/kuiwuchn/GPUMPM.

7. Emerging Trends

In this section, we briefly discuss trends and breakthroughs in multiphysics simulation that are emerging from the computer graphics community.

Notably, there is increasing interest on developing simulation techniques that leverage recent advances in machine learning, with the motivation being that increased levels of detail and realism can be achieved at reduced computational cost. Much work on this topic focuses on deriving a reduced simulation model by sampling behavior obtained from traditional simulation techniques. Some approaches have focused on end-to-end learning of response to external interactions with learned physical objects [TPNK24,HDDN19], whereas other approaches focus on improving performance by learning reduced latent sub-spaces [CXC*23, CCW*23, SYS*21]. Zong et al. [ZLL*23] used a common latent representation to learn neural fields embedding the deformation, stress and affine momentum for MPM simulations of fracture and elastoplasticity for a wide range of material behaviors. The recently proposed Simplicits [MSP*24] addresses the problem of performing elastodynamic simulation of solids from arbitrary geometric representations, thereby foregoing the need for specific mesh-based or point-based discretizations. Models derived from the rendering community have also proven interesting for their ability to combine visual appearance and physical dynamics [FSL*23, LQC*23, XZQ*24]. Point-based simulation frameworks in particular have recently been demonstrated to pair well with 3D Gaussian Splatting [JYX*24], and results demonstrating fluid-solid interactions have also be shown [FFS*24].

Although many methods in this class of emerging approaches tend to focus on simulation of individual phenomena or a specific material behavior, there is increasing emphasis on integrating neural models in a multiphysics setting. Development of frameworks that facilitate research on this topic is therefore interesting [HNN*21, Gen24].

8. Conclusion

In this tutorial, we presented the mathematical foundations and key algorithms behind the most widely used multiphysics simulation methods in computer graphics. We covered energy-based and constraint-based unified formulations, Lagrangian point-based methods, as well as Eulerian and hybrid approaches such as the Material Point Method.

For each approach, we discussed how different materials and physical phenomena – from rigid and deformable bodies to fluids and granular materials – can be represented within a common framework, and how unified models and coupling techniques each offer distinct trade-offs between simplicity, flexibility, and stability. We also presented a selection of multiphysics software frameworks and discussed emerging trends such as machine learning-

based simulation methods, which will likely have a significant influence on multiphysics simulation in the years to come.

We hope that these course notes serve as a useful reference for researchers and practitioners looking to implement or extend multiphysics simulation systems.

Acknowledgements

We would like to thank Daniel Holz and Yin Yang for kindly providing images from their works for use in this tutorial. This work is partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — 281466253; 310833819.

References

[AAT13] AKINCI N., AKINCI G., TESCHNER M.: Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics 32*, 6 (Nov. 2013), 182:1–182:8. doi:10.1145/2508363.2508395. 25

[ACAT13] AKINCI N., CORNELIS J., AKINCI G., TESCHNER M.: Coupling elastic solids with smoothed particle hydrodynamics fluids. *Computer Animation and Virtual Worlds 24*, 3-4 (May 2013), 195–203. doi:10.1002/cav.1499. 26

[AEF22] ANDREWS S., ERLEBEN K., FERGUSON Z.: Contact and friction simulation for computer graphics. In *ACM SIGGRAPH 2022 Courses* (Aug. 2022), SIGGRAPH '22, Association for Computing Machinery, pp. 1–172. doi:10.1145/3532720.3535640. 10

[AIA*12] AKINCI N., IHMSEN M., AKINCI G., SOLENTHALER B., TESCHNER M.: Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics 31*, 4 (Aug. 2012), 1–8. doi:10.1145/2185520.2185558. 26

[And24] ANDO R.: A cubic barrier with elasticity-inclusive dynamic stiffness. *ACM Transactions on Graphics 43*, 6 (Nov. 2024), 1–13. doi:10.1145/3687908. 10

[ANZS18] AKBAY M., NOBLES N., ZORDAN V., SHINAR T.: An extended partitioned method for conservative solid-fluid coupling. *ACM Transactions on Graphics 37*, 4 (July 2018). URL: https://doi.org/10.1145/3197517.3201345, doi:10.1145/3197517.3201345. 29

[AO11] ALDUÁN I., OTADUY M. A.: SPH granular flow with friction and cohesion. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aug. 2011), SCA '11, Association for Computing Machinery, pp. 25–32. doi:10.1145/2019406.2019410. 26

[ARM*19] ANGLES B., REBAIN D., MACKLIN M., WYVILL B., BARTHE L., LEWIS J., VON DER PAHLEN J., IZADI S., VALENTIN J., BOUAZIZ S., TAGLIASACCHI A.: VIPER: Volume Invariant Position-based Elastic Rods. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (July 2019), 19:1–19:26. doi:10.1145/3340260. 21

[ARNM*20] ABU-RUMMAN N., NAIR P., MÜLLER P., BARTHE L., VANDERHAEGHE D.: ISPH–PBD: coupled simulation of incompressible fluids and deformable bodies. *The Visual Computer 36*, 5 (May 2020), 893–910. doi:10.1007/s00371-019-01700-y. 21

[ATW15] ANDO R., THUEREY N., WOJTAN C.: A stream function solver for liquid simulations. *ACM Trans. Graph. 34*, 4 (July 2015), 53:1–53:9. doi:10.1145/2766935. 29

[AW09] ADAMS B., WICKE M.: Meshless Approximation Methods and Applications in Physics Based Modeling and Animation. In *Proceedings of the Eurographics conference* (2009), The Eurographics Association, pp. 213–239. 23

[B*26a] BENDER J., ET AL.: PositionBasedDynamics, 2026. URL: https://github.com/InteractiveComputerGraphics/PositionBasedDynamics. 19, 20

[B*26b] BENDER J., ET AL.: SPlisHSPlasH Library, 2026. URL: https://github.com/InteractiveComputerGraphics/SPlisHSPlasH. 24, 33

[Bat06] BATHE K.-J.: *Finite element procedures*. Klaus-Jurgen Bathe, 2006. 27, 30

[BAV*10] BERGOU M., AUDOLY B., VOUGA E., WARDETZKY M., GRINSPUN E.: Discrete viscous threads. *ACM Transactions on Graphics 29*, 4 (July 2010), 1–10. doi:10.1145/1778765.1778853. 11

[BB08] BATTY C., BRIDSON R.: Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland, 2008), SCA '08, Eurographics Association, p. 219–228. doi:10.5555/1632592.1632624. 15

[BB12] BOYD L., BRIDSON R.: Multiflip for energetic two-phase fluid simulation. *ACM Trans. Graph. 31*, 2 (Apr. 2012). URL: https://doi.org/10.1145/2159516.2159522, doi:10.1145/2159516.2159522. 29

[BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics 26*, 3 (July 2007), 100–es. doi:10.1145/1276377.1276502. 29

[BBT25] BRAUN B., BENDER J., THUEREY N.: Adaptive phase-field-flip for very large scale two-phase fluid simulation. *ACM Trans. Graph. 44*, 4 (July 2025). URL: https://doi.org/10.1145/3730854, doi:10.1145/3730854. 28

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. *Computer Graphics Forum 31*, 5 (Aug. 2012), 1657–1667. doi:10.1111/j.1467-8659.2012.03171.x. 15, 16

[BEH18] BRANDT C., EISEMANN E., HILDEBRANDT K.: Hyper-reduced projective dynamics. *ACM Transactions on Graphics 37*, 4 (July 2018), 1–13. doi:10.1145/3197517.3201387. 16

[Ben26] BENDER J.: Physics Simulation in Visual Computing, 2026. URL: https://github.com/InteractiveComputerGraphics/physics-simulation. 19, 20, 24

[BET14] BENDER J., ERLEBEN K., TRINKLE J.: Interactive simulation of rigid body dynamics in computer graphics. *Computer Graphics Forum 33*, 1 (2014), 246–270. doi:10.1111/cgf.12272. 13

[BGAO17] BARREIRO H., GARCÍA-FERNÁNDEZ I., ALDUÁN I., OTADUY M. A.: Conformation constraints for efficient viscoelastic fluid simulation. *ACM Transactions on Graphics 36*, 6 (Nov. 2017), 221:1–221:11. doi:10.1145/3130800.3130854. 21, 25

[BGI*18] BAND S., GISSLER C., IHMSEN M., CORNELIS J., PEER A., TESCHNER M.: Pressure boundaries for implicit incompressible sph. *ACM Transactions on Graphics 37*, 2 (Feb. 2018), 14:1–14:11. doi:10.1145/3180486. 26

[BGPT18] BAND S., GISSLER C., PEER A., TESCHNER M.: MLS pressure boundaries for divergence-free and viscous SPH fluids. *Computers & Graphics 76* (Nov. 2018), 37–46. doi:10.1016/j.cag.2018.08.001. 23, 26

[BHSH*22] BUNGE A., HERHOLZ P., SORKINE-HORNUNG O., BOTSCH M., KAZHDAN M.: Variational quadratic shape functions for polygons and polyhedra. *ACM Transactions on Graphics 41*, 4 (July 2022), 1–14. doi:10.1145/3528223.3530137. 9

[BIT09] BECKER M., IHMSEN M., TESCHNER M.: Corotated SPH for deformable solids. In *Proceedings of the Fifth Eurographics Conference on Natural Phenomena* (Apr. 2009), NPH'09, Eurographics Association, pp. 27–34. 26

[BK17] BENDER J., KOSCHIER D.: Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Transactions on Visualization and Computer Graphics 23*, 3 (Mar. 2017), 1193–1206. doi:10.1109/TVCG.2016.2578335. 24

[BKCW14] BENDER J., KOSCHIER D., CHARRIER P., WEBER D.: Position-based simulation of continuous materials. *Computers & Graphics 44* (Nov. 2014), 1–10. doi:10.1016/j.cag.2014.07.004. 18, 21

[BKKW19] BENDER J., KOSCHIER D., KUGELSTADT T., WEILER M.: Turbulent Micropolar SPH Fluids with Foam. *IEEE Transactions on Visualization and Computer Graphics 25*, 6 (June 2019), 2284–2295. doi:10.1109/TVCG.2018.2832080. 25

[BKWK20] BENDER J., KUGELSTADT T., WEILER M., KOSCHIER D.: Implicit Frictional Boundary Handling for SPH. *IEEE Transactions on Visualization and Computer Graphics 26*, 10 (Oct. 2020), 2982–2993. doi:10.1109/TVCG.2020.3004245. 26

[BLS12] BODIN K., LACOURSIERE C., SERVIN M.: Constraint Fluids. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (Mar. 2012), 516–526. doi:10.1109/TVCG.2011.29. 20, 24

[BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), SCA '03, Eurographics Association, p. 28–36. 12

[BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics 33*, 4 (July 2014), 1–11. doi:10.1145/2601097.2601116. 15, 16, 18, 19, 20, 21

[BMM17] BENDER J., MÜLLER M., MACKLIN M.: A Survey on Position Based Dynamics, 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials* (Apr. 2017), EG '17, Eurographics Association, pp. 1–31. doi:10.2312/egt.20171034. 18, 19, 20, 21

[BOFN18] BROWN G. E., OVERBY M., FOROOTANINIA Z., NARAIN R.: Accurate dissipative forces in optimization integrators. *ACM Transactions on Graphics 37*, 6 (dec 2018). doi:10.1145/3272127.3275011. 4, 10, 11, 16, 17

[BR86] BRACKBILL J. U., RUPPEL H. M.: FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics 65*, 2 (Aug. 1986), 314–343. doi:10.1016/0021-9991(86)90211-1. 30

[Bri15] BRIDSON R.: *Fluid simulation for computer graphics*. AK Peters/CRC Press, 2015. 27, 28

[BS22] BAI S., SCHROEDER C.: Stability analysis of explicit mpm. *Computer Graphics Forum 41*, 8 (Dec. 2022), 19–30. doi:10.1111/cgf.14620. 30

[BSEH19] BRANDT C., SCANDOLO L., EISEMANN E., HILDEBRANDT K.: The reduced immersed method for real-time fluid-elastic solid interaction and contact simulation. *ACM Transactions on Graphics 38*, 6 (Dec. 2019), 1–16. doi:10.1145/3355089.3356496. 17

[BT07] BECKER M., TESCHNER M.: Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aug. 2007), SCA '07, Eurographics Association, pp. 209–217. 24, 25

[BTT09] BECKER M., TESSENDORF H., TESCHNER M.: Direct Forcing for Lagrangian Rigid-Fluid Coupling. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (May 2009), 493–503. doi:10.1109/TVCG.2008.107. 26

[BUAG12] BATTY C., URIBE A., AUDOLY B., GRINSPUN E.: Discrete viscous sheets. *ACM Transactions on Graphics 31*, 4 (July 2012), 1–7. doi:10.1145/2185520.2185609. 11

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (July 1998), SIGGRAPH '98, Association for Computing Machinery, pp. 43–54. doi:10.1145/280814.280821. 12

[BWH*06] BERGOU M., WARDETZKY M., HARMON D., ZORIN D., GRINSPUN E.: A quadratic bending model for inextensible surfaces. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (2006), SGP '06, Eurographics Association, p. 227–230. 12

[BWJB25] BÖTTCHER T., WESTHOFEN L., JESKE S. R., BENDER J.: Implicit Incompressible Porous Flow using SPH. *ACM Transactions on Graphics 44*, 6 (Dec. 2025). doi:10.1145/3763325. 25

[BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete elastic rods. In *ACM SIGGRAPH 2008 Papers* (2008), SIGGRAPH '08, Association for Computing Machinery. doi:10.1145/1399504.1360662. 13

[BWRJ23] BENDER J., WESTHOFEN L., RHYS JESKE S.: Consistent SPH Rigid-Fluid Coupling. *Vision, Modeling, and Visualization* (2023), 209–217. doi:10.2312/VMV.20231244. 26

[CBG*19] CORNELIS J., BENDER J., GISSLER C., IHMSEN M., TESCHNER M.: An optimized source term formulation for incompressible SPH. *The Visual Computer 35*, 4 (Apr. 2019), 579–590. doi:10.1007/s00371-018-1488-8. 24

[CCL*22] CAO Y., CHEN Y., LI M., YANG Y., ZHANG X., AANJANEYA M., JIANG C.: An efficient b-spline lagrangian/eulerian method for compressible flow, shock waves, and fracturing solids. *ACM Transactions on Graphics 41*, 5 (May 2022), 1–13. doi:10.1145/3519595. 30, 32

[CCW*23] CHANG Y., CHEN P. Y., WANG Z., CHIARAMONTE M. M., CARLBERG K., GRINSPUN E.: Licrom: Linear-subspace continuous reduced order modeling with neural fields. In *SIGGRAPH Asia 2023 Conference Papers* (2023), SA '23, Association for Computing Machinery. doi:10.1145/3610548.3618158. 33

[CDGB19] CHANG J., DA F., GRINSPUN E., BATTY C.: A unified simplicial model for mixed-dimensional and non-manifold deformable elastic objects. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (July 2019), 1–18. doi:10.1145/3340252. 14

[CDY23] CHEN H., DIAZ E., YUKSEL C.: Shortest path to boundary for self-intersecting meshes. *ACM Transactions on Graphics 42*, 4 (jul 2023). doi:10.1145/3592136. 10

[Cet19] CETINASLAN O.: Position-Based Simulation of Elastic Models on the GPU with Energy Aware Gauss-Seidel Algorithm. *Computer Graphics Forum 38*, 8 (Nov. 2019), 41–52. doi:10.1111/cgf.13759. 20

[Cet24] CETINASLAN O.: ESBD: Exponential Strain-based Dynamics using XPBD algorithm. *Computers & Graphics 116*, C (Mar. 2024), 500–512. doi:10.1016/j.cag.2023.09.014. 21

[CFL*07] CHENTANEZ N., FELDMAN B. E., LABELLE F., O'BRIEN J. F., SHEWCHUK J. R.: Liquid simulation on lattice-based tetrahedral meshes. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, Eurographics Association, p. 219–228. 29

[CGFO06] CHENTANEZ N., GOKTEKIN T. G., FELDMAN B. E., O'BRIEN J. F.: Simultaneous coupling of fluids and deformable bodies. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, Eurographics Association, p. 83–89. 29

[CHC*24a] CHEN Y., HAN Y., CHEN J., MA S., FEDKIW R., TERAN J.: Primal residual reduction with extended position based dynamics and hyperelasticity. *Computers & Graphics 119*, C (July 2024). doi:10.1016/j.cag.2024.103902. 16, 20, 21

[CHC*24b] CHEN Y., HAN Y., CHEN J., ZHANG Z., MCADAMS A., TERAN J.: Position-Based Nonlinear Gauss-Seidel for Quasistatic Hyperelasticity. *ACM Transactions on Graphics 43*, 4 (July 2024), 115:1–115:15. doi:10.1145/3658154. 15

[Cho14] CHOI M. G.: Real-time simulation of ductile fracture with oriented particles. *Computer Animation and Virtual Worlds 25*, 3-4 (May 2014), 457–465. doi:10.1002/cav.1601. 21

[CKMR*21] CHEN J., KALA V., MARQUEZ-RAZON A., GUEIDON E., HYDE D. A. B., TERAN J.: A momentum-conserving implicit material point method for surface tension with contact angles and spatial gradients. *ACM Transactions on Graphics 40*, 4 (July 2021), 1–16. doi:10.1145/3450626.3459874. 31

[CLC*20] CHEN X.-S., LI C.-F., CAO G.-C., JIANG Y.-T., HU S.-M.: A moving least square reproducing kernel particle method for unified multiphase continuum simulation. *ACM Transactions on Graphics 39*, 6 (Nov. 2020), 176:1–176:15. doi:10.1145/3414685.3417809. 23

[CLF25] CHENG Y., LIU L., FU X.: Eigenvalue blending for projected newton. *Computer Graphics Forum* (Apr. 2025). doi:10.1111/cgf.70027. 7

[CLJ*24] CHEN H., LIU H.-T. D., JACOBSON A., LEVIN D. I., ZHENG C.: Trust-region eigenvalue filtering for projected newton. In *SIGGRAPH Asia 2024 Conference Papers* (Dec. 2024), SA '24, Association for Computing Machinery. doi:10.1145/3680528.3687650. 7

[CLL*22] CHEN Y., LI M., LAN L., SU H., YANG Y., JIANG C.: A unified newton barrier method for multibody dynamics. *ACM Transactions on Graphics 41*, 4 (jul 2022). doi:10.1145/3528223.3530076. 3, 4, 14

[CLL*24] CHEN H., LIU H.-T. D., LEVIN D. I., ZHENG C., JACOBSON A.: Stabler neo-hookean simulation: Absolute eigenvalue filtering for projected newton. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24* (July 2024), SIGGRAPH '24, ACM, pp. 1–10. doi:10.1145/3641519.3657433. 7

[CLYY24] CHEN A. H., LIU Z., YANG Y., YUKSEL C.: Vertex Block Descent. *ACM Transactions on Graphics 43*, 4 (July 2024), 116:1–116:16. doi:10.1145/3658179. 15

[CM10] CHENTANEZ N., MÜLLER M.: Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), SCA '10, Eurographics Association, pp. 197–206. 26

[CMK15] CHENTANEZ N., MÜLLER M., KIM T.-Y.: Coupling 3D Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena. *IEEE Transactions on Visualization and Computer Graphics 21*, 10 (2015), 1116–1128. URL: https://ieeexplore.ieee.org/document/7132780, doi:10.1109/TVCG.2015.2449303. 26

[CMM16] CHENTANEZ N., MÜLLER M., MACKLIN M.: Real-time simulation of large elasto-plastic deformation with shape matching. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2016), SCA '16, Eurographics Association, pp. 159–167. 21

[CMT04] CARLSON M., MUCHA P. J., TURK G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics 23*, 3 (aug 2004), 377–384. doi:10.1145/1015706.1015733. 29

[CMTT21] COROS S., MACKLIN M., THOMASZEWSKI B., THÜREY N.: Differentiable simulation. In *SIGGRAPH Asia 2021 Courses* (New York, NY, USA, 2021), SA '21, Association for Computing Machinery. URL: https://doi.org/10.1145/3476117.3483433, doi:10.1145/3476117.3483433. 8

[CNZ*22] CHEN X., NI X., ZHU B., WANG B., CHEN B.: Simulation and optimization of magnetoelastic thin shells. *ACM Transactions on Graphics 41*, 4 (jul 2022). doi:10.1145/3528223.3530142. 14

[CSvRV18] CHEN H.-Y., SASTRY A., VAN REES W. M., VOUGA E.: Physical simulation of environmentally induced thin shell deformation. *ACM Transactions on Graphics 37*, 4 (jul 2018). doi:10.1145/3197517.3201395. 12, 14

[CWSO13] CLAUSEN P., WICKE M., SHEWCHUK J. R., O'BRIEN J. F.: Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics 32*, 2 (Apr. 2013), 1–15. doi:10.1145/2451236.2451243. 15

[CXC*23] CHEN P. Y., XIANG J., CHO D. H., CHANG Y., PERSHING G. A., MAIA H. T., CHIARAMONTE M. M., CARLBERG K. T., GRINSPUN E.: CROM: Continuous reduced-order modeling of PDEs using implicit neural representations. In *The Eleventh International Conference on Learning Representations* (2023). URL: https://openreview.net/forum?id=FUORz1tG8Og. 33

[CXY*23] CHEN Y., XIE T., YUKSEL C., KAUFMAN D., YANG Y., JIANG C., LI M.: Multi-layer thick shells. In *ACM SIGGRAPH 2023 Conference Papers* (2023), SIGGRAPH '23, Association for Computing Machinery. doi:10.1145/3588432.3591489. 13

[DB19] DAHL A., BARGTEIL A.: Global Momentum Preservation for Position-based Dynamics. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Oct. 2019), MIG '19, Association for Computing Machinery, pp. 1–5. doi:10.1145/3359566.3360078. 19

[DBD16] DAVIET G., BERTAILS-DESCOUBES F.: A semi-implicit material point method for the continuum simulation of granular materials. *ACM Transactions on Graphics 35*, 4 (July 2016), 102:1–102:13. doi:10.1145/2897824.2925877. 30, 31

[DCB16] DEUL C., CHARRIER P., BENDER J.: Position-based rigid-body dynamics. *Computer Animation and Virtual Worlds 27*, 2 (2016), 103–112. doi:10.1002/cav.1614. 21

[DKWB18] DEUL C., KUGELSTADT T., WEILER M., BENDER J.: Direct Position-Based Solver for Stiff Rods. *Computer Graphics Forum 37*, 6 (2018), 313–324. doi:10.1111/cgf.13326. 21

[DLCT24] DU Y., LI Y., COROS S., THOMASZEWSKI B.: Robust and artefact-free deformable contact with smooth surface representations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2024), SCA '24, Eurographics Association, p. 1–13. doi:10.1111/cgf.15187. 10

[DLK18] DINEV D., LIU T., KAVAN L.: Stabilizing Integrators for Real-Time Physics. *ACM Transactions on Graphics 37*, 1 (Jan. 2018), 9:1–9:19. doi:10.1145/3153420. 4, 17

[DLL*18] DINEV D., LIU T., LI J., THOMASZEWSKI B., KAVAN L.: FEPR: fast energy projection for real-time simulation of deformable objects. *ACM Transactions on Graphics 37*, 4 (July 2018), 1–12. doi:10.1145/3197517.3201277. 17

[DS19] DING O., SCHROEDER C.: Penalty force for coupling materials with coulomb friction. *IEEE Transactions on Visualization and Computer Graphics 26*, 7 (2019), 2443–2455. 30

[dVNNT21] DE VAUCORBEIL A., NGUYEN V. P., NGUYEN-THANH C.: Karamelo: an open source parallel C++ package for the material point method. *Computational Particle Mechanics 8*, 4 (July 2021), 767–789. doi:10.1007/s40571-020-00369-8. 33

[DWM*21] DU T., WU K., MA P., WAH S., SPIELBERG A., RUS D., MATUSIK W.: Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics 41*, 2 (Nov. 2021), 1–21. doi:10.1145/3490168. 16

[FCK22] FAN L., CHITALU F. M., KOMURA T.: Simulating brittle fracture with material points. *ACM Transactions on Graphics 41*, 5 (2022), 1–20. 30, 31

[FDD*12] FAURE F., DURIEZ C., DELINGETTE H., ALLARD J., GILLES B., MARCHESSEAU S., TALBOT H., COURTECUISSE H., BOUSQUET G., PETERLIK I., COTIN S.: SOFA: A Multi-Model Framework for Interactive Physical Simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, vol. 11. Springer Berlin Heidelberg, 2012, pp. 283–321. doi:10.1007/8415_2012_125. 32

[FFLB26] FERNÁNDEZ-FERNÁNDEZ J. A., LÖSCHNER F., BENDER J.: Progressively projected newton's method. *Computer Graphics Forum (Eurographics) 45*, 2 (2026). doi:10.1111/cgf.70386. 7

[FFLL*24] FERNÁNDEZ-FERNÁNDEZ J. A., LANGE R., LAIBLE S., ARRAS K. O., BENDER J.: Stark: A unified framework for strongly coupled simulation of rigid and deformable bodies with frictional contact. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2024), pp. 16888–16894. doi:10.1109/ICRA57147.2024.10610574. 3, 9, 32

[FFLW*25] FERNÁNDEZ-FERNÁNDEZ J. A., LÖSCHNER F., WESTHOFEN L., LONGVA A., BENDER J.: Symx: Energy-based simulation from symbolic expressions. *ACM Trans. Graph. 45*, 1 (Oct. 2025).

URL: https://doi.org/10.1145/3764928, doi:10.1145/3764928. 1, 8, 33

[FFS*24] FENG Y., FENG X., SHANG Y., JIANG Y., YU C., ZONG Z., SHAO T., WU H., ZHOU K., JIANG C., YANG Y.: Gaussian splashing: Unified particles for versatile motion synthesis and rendering. *arXiv preprint arXiv:2401.15318* (2024). 33

[FGG*17] FU C., GUO Q., GAST T., JIANG C., TERAN J.: A polynomial particle-in-cell method. *ACM Transactions on Graphics 36*, 6 (Nov. 2017), 222:1–222:12. doi:10.1145/3130800.3130878. 30

[FHG21] FEI Y., HUANG Y., GAO M.: Principles towards real-time simulation of material point method on modern gpus. *arXiv preprint arXiv:2111.00699* (2021). 30

[FJZ*23] FERGUSON Z., JAIN P., ZORIN D., SCHNEIDER T., PANOZZO D.: High-order incremental potential contact for elastodynamic simulation on curved meshes. In *ACM SIGGRAPH 2023 Conference Papers* (July 2023), SIGGRAPH '23, ACM, pp. 1–11. doi:10.1145/3588432.3591488. 10

[FLGJ19] FANG Y., LI M., GAO M., JIANG C.: Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Transactions on Graphics 38*, 4 (2019), 1–13. 30

[FLS*21] FERGUSON Z., LI M., SCHNEIDER T., GIL-URETA F., LANGLOIS T., JIANG C., ZORIN D., KAUFMAN D. M., PANOZZO D.: Intersection-free rigid body dynamics. *ACM Transactions on Graphics 40*, 4 (jul 2021). doi:10.1145/3450626.3459802. 3, 13

[FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graphical Models and Image Processing 58*, 5 (1996), 471–483. doi:10.1006/gmip.1996.0039. 28, 29

[FM15] FUJISAWA M., MIURA K. T.: An Efficient Boundary Handling with a Modified Density Calculation for SPH. *Computer Graphics Forum 34*, 7 (2015), 155–162. doi:10.1111/cgf.12754. 26

[FM17] FRÂNCU M., MOLDOVEANU F.: Unified simulation of rigid and flexible bodies using position based dynamics. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations* (Apr. 2017), VRIPHYS '17, Eurographics Association, pp. 49–58. doi:10.2312/vriphys.20171083. 19, 20, 21, 22

[Fo20] FERGUSON Z., OTHERS: IPC Toolkit, 2020. URL: https://github.com/ipc-sim/ipc-toolkit. 9

[FP13] FRATARCANGELI M., PELLACINI F.: A GPU-Based Implementation of Position Based Dynamics for Interactive Deformable Bodies. *Journal of Graphics Tools 17*, 3 (July 2013), 59–66. doi:10.1080/2165347X.2015.1030525. 20

[FSL*23] FENG Y., SHANG Y., LI X., SHAO T., JIANG C., YANG Y.: Pie-nerf: Physics-based interactive elastodynamics with nerf, 2023. arXiv:2311.13099. 33

[FTP16] FRATARCANGELI M., TIBALDO V., PELLACINI F.: Vivace: a practical gauss-seidel method for stable soft body dynamics. *ACM Transactions on Graphics 35*, 6 (Nov. 2016), 1–9. doi:10.1145/2980179.2982437. 16

[FYCZ23] FANG J., YOU L., CHAUDHRY E., ZHANG J.: State-of-the-art improvements and applications of position based dynamics. *Computer Animation and Virtual Worlds 34*, 5 (2023), e2143. doi:10.1002/cav.2143. 18

[GAB20] GOLDADE R., AANJANEYA M., BATTY C.: Constraint bubbles and affine regions: Reduced fluid models for efficient immersed bubbles and flexible spatial coarsening. *ACM Trans. Graph. 39*, 4 (Aug. 2020), 43:43:1–43:43:15. doi:10.1145/3386569.3392455. 29

[Gen24] GENESIS AUTHORS: Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL: https://github.com/Genesis-Embodied-AI/Genesis. 33

[GGWZ07] GARG A., GRINSPUN E., WARDETZKY M., ZORIN D.: Cubic shells. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, Eurographics Association, p. 91–98. 12

[GHB*20] GISSLER C., HENNE A., BAND S., PEER A., TESCHNER M.: An implicit compressible SPH solver for snow simulation. *ACM Transactions on Graphics 39*, 4 (Aug. 2020), 36:36:1–36:36:16. doi:10.1145/3386569.3392431. 26

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells, 2003. doi:10.2312/SCA03/062-067. 12

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3 (July 2007), 49–es. URL: https://dl.acm.org/doi/10.1145/1276377.1276438, doi:10.1145/1276377.1276438. 19

[GHF*18] GUO Q., HAN X., FU C., GAST T., TAMSTORF R., TERAN J.: A material point method for thin shells with frictional contact. *ACM Transactions on Graphics 37*, 4 (July 2018), 1–15. doi:10.1145/3197517.3201346. 32

[GLY*24] GUO D., LI M., YANG Y., WANG G., LI S.: Barrier-augmented Lagrangian for GPU-based elastodynamic contact. *ACM Transactions on Graphics* (June 2024). doi:10.1145/3687988. 10

[GM77] GINGOLD R. A., MONAGHAN J. J.: Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society 181*, 3 (Dec. 1977), 375–389. doi:10.1093/mnras/181.3.375. 22

[GPB*19] GISSLER C., PEER A., BAND S., BENDER J., TESCHNER M.: Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling. *ACM Transactions on Graphics 38*, 1 (Jan. 2019), 5:1–5:13. doi:10.1145/3284980. 1, 23, 26

[GPH*18] GAO M., PRADHANA A., HAN X., GUO Q., KOT G., SIFAKIS E., JIANG C.: Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics 37*, 4 (July 2018), 149:1–149:11. doi:10.1145/3197517.3201309. 31, 32

[GSLF05] GUENDELMAN E., SELLE A., LOSASSO F., FEDKIW R.: Coupling water and smoke to thin deformable and rigid shells. In *ACM SIGGRAPH 2005 Papers* (July 2005), SIGGRAPH '05, Association for Computing Machinery, pp. 973–981. doi:10.1145/1186822.1073299. 29

[GSS*15] GAST T. F., SCHROEDER C., STOMAKHIN A., JIANG C., TERAN J. M.: Optimization integrator for large time steps. *IEEE Transactions on Visualization and Computer Graphics 21*, 10 (Oct. 2015), 1103–1115. doi:10.1109/tvcg.2015.2459687. 3, 11, 30

[GTJS17] GAO M., TAMPUBOLON A. P., JIANG C., SIFAKIS E.: An adaptive generalized interpolation material point method for simulating elastoplastic materials. *ACM Transactions on Graphics 36*, 6 (2017), 1–12. 30

[GWW*18] GAO M., WANG X., WU K., PRADHANA A., SIFAKIS E., YUKSEL C., JIANG C.: GPU optimization of material point methods. *ACM Transactions on Graphics 37*, 6 (Dec. 2018), 254:1–254:12. doi:10.1145/3272127.3275044. 30, 33

[Har62] HARLOW F.: *The particle-in-cell method for numerical solution of problems in fluid dynamics*. Tech. rep., Los Alamos National Laboratory (LANL), Mar. 1962. Report Number: LADC-5288, 4769185. doi:10.2172/4769185. 30

[HB23] HAFNER C., BICKEL B.: The design space of Kirchhoff rods. *ACM Transactions on Graphics 42*, 5 (sep 2023). doi:10.1145/3606033. 13

[HCLK24] HUANG K., CHITALU F. M., LIN H., KOMURA T.: GIPC: Fast and stable gauss-newton optimization of IPC barrier energy. *ACM Transactions on Graphics 43*, 2 (mar 2024). doi:10.1145/3643028. 10

[HDDN19] HOLDEN D., DUONG B. C., DATTA S., NOWROUZEZAHRAI D.: Subspace neural physics: fast data-driven interactive simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2019), SCA '19, Association for Computing Machinery. doi:10.1145/3309486.3340245. 33

[HFG*18] HU Y., FANG Y., GE Z., QU Z., ZHU Y., PRADHANA A., JIANG C.: A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics 37*, 4 (July 2018), 150:1–150:14. doi:10.1145/3197517.3201293. 30, 31, 33

[HG18] HOLZ D., GALARNEAU A.: Real-time mud simulation for virtual environments. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D), Poster Session* (2018). 18, 22

[HGG*19] HAN X., GAST T. F., GUO Q., WANG S., JIANG C., TERAN J.: A Hybrid Material Point Method for Frictional Contact with Diverse Materials. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (July 2019), 17:1–17:24. doi:10.1145/3340258. 31

[HGMRT20] HYDE D. A., GAGNIERE S. W., MARQUEZ-RAZON A., TERAN J.: An implicit updated lagrangian formulation for liquids with large surface energy. *ACM Transactions on Graphics 39*, 6 (2020), 1–13. 30

[HK05] HONG J.-M., KIM C.-H.: Discontinuous fluids. *ACM Trans. Graph. 24*, 3 (July 2005), 915–920. URL: https://doi.org/10.1145/1073204.1073283, doi:10.1145/1073204.1073283. 29

[HLL*25] HUANG K., LU X., LIN H., KOMURA T., LI M.: Stiffgipc: Advancing gpu ipc for stiff affine-deformable simulation. *ACM Transactions on Graphics 44*, 3 (May 2025). doi:10.1145/3735126. 14

[HNN*21] HENNIGH O., NARASIMHAN S., NABIAN M. A., SUBRAMANIAM A., TANGSALI K., FANG Z., RIETMANN M., BYEON W., CHOUDHRY S.: Nvidia simnet™: An ai-accelerated multi-physics simulation framework. In *Computational Science – ICCS 2021: 21st International Conference, Krakow, Poland, June 16–18, 2021, Proceedings, Part V* (2021), Springer-Verlag, p. 447–461. doi:10.1007/978-3-030-77977-1_36. 33

[Hol14] HOLZ D.: Parallel Particles (P2): A Parallel Position Based Approach for Fast and Stable Simulation of Granular Materials. In *VRIPHYS 14: 11th Workshop on Virtual Reality Interactions and Physical Simulations, Bremen, Germany, 2014. Proceedings* (2014), Eurographics Association, pp. 135–144. doi:10.2312/VRIPHYS.20141232. 19, 20, 21

[HPF*25] HUANG Z., PAIK M., FERGUSON Z., PANOZZO D., ZORIN D.: Geometric contact potential. *ACM Transactions on Graphics 44*, 4 (July 2025). URL: https://doi.org/10.1145/3731142, doi:10.1145/3731142. 10

[HSK24] HERHOLZ P., STUYCK T., KAVAN L.: A mesh-based simulation framework using automatic code generation. *ACM Transactions on Graphics 43*, 6 (Nov. 2024), 1–17. doi:10.1145/3687986. 8, 9

[HW*65] HARLOW F. H., WELCH J. E., ET AL.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids 8*, 12 (1965), 2182. 27, 28, 30

[HWW18] HE X., WANG H., WU E.: Projective Peridynamics for Modeling Versatile Elastoplastic Materials. *IEEE Transactions on Visualization and Computer Graphics 24*, 9 (Sept. 2018), 2589–2599. doi:10.1109/TVCG.2017.2755646. 16, 17

[HWZ*15] HE X., WANG H., ZHANG F., WANG H., WANG G., ZHOU K.: Robust Simulation of Sparsely Sampled Thin Features in SPH-Based Free Surface Flows. *ACM Transactions on Graphics 34*, 1 (Dec. 2015), 7:1–7:9. doi:10.1145/2682630. 25

[ICS*14] IHMSEN M., CORNELIS J., SOLENTHALER B., HORVATH C., TESCHNER M.: Implicit Incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (Mar. 2014), 426–435. doi:10.1109/TVCG.2013.105. 24

[IOS*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M.: SPH Fluids in Computer Graphics. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2014), 426–435. doi:10.2312/egst.20141034. 23, 24

[IWT13] IHMSEN M., WAHL A., TESCHNER M.: A Lagrangian framework for simulating granular material with high detail. *Computers & Graphics 37*, 7 (Nov. 2013), 800–808. doi:10.1016/j.cag.2013.04.010. 26

[Jak01] JAKOBSEN T.: Advanced character physics. In *Proceedings Game Developer's Conference 2001* (San José, USA, 2001). 18

[JBB*22] JESKE S. R., BENDER J., BOBZIN K., HEINEMANN H., JASUTYN K., SIMON M., MOKROV O., SHARMA R., REISGEN U.: Application and benchmark of SPH for modeling the impact in thermal spraying. *Computational Particle Mechanics* (jan 2022). doi:10.1007/s40571-022-00459-9. 33

[JGT17] JIANG C., GAST T., TERAN J.: Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics 36*, 4 (2017), 1–14. 30

[JSS*15] JIANG C., SCHROEDER C., SELLE A., TERAN J., STOMAKHIN A.: The affine particle-in-cell method. *ACM Transactions on Graphics 34*, 4 (July 2015), 1–10. doi:10.1145/2766996. 15, 30

[JSS*22] JESKE S. R., SIMON M. S., SEMENOV O., KRUSKA J., MOKROV O., SHARMA R., REISGEN U., BENDER J.: Quantitative evaluation of SPH in TIG spot welding. *Computational Particle Mechanics* (apr 2022). doi:10.1007/s40571-022-00465-x. 33

[JST*16] JIANG C., SCHROEDER C., TERAN J., STOMAKHIN A., SELLE A.: The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses* (July 2016), SIGGRAPH '16, Association for Computing Machinery, pp. 1–52. doi:10.1145/2897826.2927348. 30

[JWL*23] JESKE S. R., WESTHOFEN L., LÖSCHNER F., FERNÁNDEZ-FERNÁNDEZ J. A., BENDER J.: Implicit Surface Tension for SPH Fluid Simulation. *ACM Transactions on Graphics* (Nov. 2023). doi:10.1145/3631936. 25

[JYX*24] JIANG Y., YU C., XIE T., LI X., FENG Y., WANG H., LI M., LAU H., GAO F., YANG Y., JIANG C.: Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers* (2024), SIGGRAPH '24, Association for Computing Machinery. doi:10.1145/3641519.3657448. 33

[KB17] KOSCHIER D., BENDER J.: Density maps for improved SPH boundary handling. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (July 2017), SCA '17, Association for Computing Machinery, pp. 1–10. doi:10.1145/3099564.3099565. 26

[KB18] KOMARITZAN M., BOTSCH M.: Projective skinning. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 1 (July 2018), 1–19. doi:10.1145/3203203. 17

[KB19] KOMARITZAN M., BOTSCH M.: Fast projective skinning. In *Motion, Interaction and Games* (Oct. 2019), vol. 22 of *MIG '19*, ACM, pp. 1–10. doi:10.1145/3359566.3360073. 17

[KBF*21] KUGELSTADT T., BENDER J., FERNÁNDEZ-FERNÁNDEZ J. A., JESKE S. R., LÖSCHNER F., LONGVA A.: Fast Corotated Elastic SPH Solids with Implicit Zero-Energy Mode Control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 4*, 3 (Sept. 2021), 33:1–33:21. doi:10.1145/3480142. 26

[KBST19] KOSCHIER D., BENDER J., SOLENTHALER B., TESCHNER M.: Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. In *Eurographics 2019 - Tutorials* (2019), The Eurographics Association. doi:10.2312/egt.20191035. 22, 23, 24

[KBST22] KOSCHIER D., BENDER J., SOLENTHALER B., TESCHNER M.: A Survey on SPH Methods in Computer Graphics. *Computer Graphics Forum 41*, 2 (2022), 737–760. doi:10.1111/cgf.14508. 23, 24

[KE22] KIM T., EBERLE D.: Dynamic deformables: implementation and production practicalities (now with code!). In *ACM SIGGRAPH 2022 Courses* (2022), SIGGRAPH '22, Association for Computing Machinery. doi:10.1145/3532720.3535628. 9

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Transactions on Graphics 25*, 3 (July 2006), 820–825. doi:10.1145/1141911.1141961. 29

[KGBG09] KAUFMANN, PETER, GISCHIG, SEBASTIAN MARTIN, BOTSCH, MARIO, GROSS, MARKUS: *Implementation of discontinuous Galerkin Kirchhoff-Love shells*. Tech. rep., ETH Zürich, 2009. doi:10.3929/ETHZ-A-006733717. 13

[KGP*16] KLÁR G., GAST T., PRADHANA A., FU C., SCHROEDER C., JIANG C., TERAN J.: Drucker-prager elastoplasticity for sand animation. *ACM Transactions on Graphics 35*, 4 (2016), 1–12. 30, 31

[Kim20] KIM T.: A finite element formulation of Baraff-Witkin cloth. *Computer Graphics Forum 39*, 8 (Nov. 2020), 171–179. doi:10.1111/cgf.14111. 12

[KKB18] KUGELSTADT T., KOSCHIER D., BENDER J.: Fast corotated fem using operator splitting. *Computer Graphics Forum 37*, 8 (Sept. 2018), 149–160. doi:10.1111/cgf.13520. 9

[KKHS20] KARPMAN E., KÖVECSES J., HOLZ D., SKONIECZNY K.: Discrete element modelling for wheel-soil interaction and the analysis of the effect of gravity. *Journal of Terramechanics 91* (Oct. 2020), 139–153. doi:10.1016/j.jterra.2020.06.002. 22

[KLM24] KIM D., LEE M., MUSETH K.: Neuralvdb: High-resolution sparse volume representation using hierarchical neural networks. *ACM Trans. Graph. 43*, 2 (Feb. 2024). URL: https://doi.org/10.1145/3641817, doi:10.1145/3641817. 28

[KMOW00] KANE C., MARSDEN J. E., ORTIZ M., WEST M.: Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering 49*, 10 (2000), 1295–1325. doi:10.1002/1097-0207(20001210)49:10<1295::aid-nme993>3.0.co;2-w. 4

[KS16] KUGELSTADT T., SCHÖMER E.: Position and orientation based Cosserat rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2016), SCA '16, Eurographics Association, pp. 169–178. 13, 21

[KSNG17] KARAMOUZAS I., SOHRE N., NARAIN R., GUY S. J.: Implicit crowds: optimization integrator for robust crowd simulation. *ACM Transactions on Graphics 36*, 4 (July 2017), 1–13. doi:10.1145/3072959.3073705. 11

[KUJH21] KEE M. H., UM K., JEONG W., HAN J.: Constrained projective dynamics: real-time simulation of deformable objects with energy-momentum conservation. *ACM Transactions on Graphics 40*, 4 (July 2021), 1–12. doi:10.1145/3450626.3459878. 17

[KUKH23] KEE M. H., UM K., KANG H., HAN J.: An Optimization-based SPH Solver for Simulation of Hyperelastic Solids. *Computer Graphics Forum 42*, 2 (2023), 225–233. doi:10.1111/cgf.14756. 26

[KYT*06] KHAREVYCH L., YANG W., TONG Y., KANSO E., MARSDEN J. E., SCHRÖDER P., DESBRUN M.: Geometric, variational integrators for computer animation. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, Eurographics Association, p. 43–51. 3

[LAD08] LENAERTS T., ADAMS B., DUTRÉ P.: Porous flow in particle-based fluid simulations. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1–8. doi:10.1145/1360612.1360648. 25

[LBB17] LARIONOV E., BATTY C., BRIDSON R.: Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Transactions on Graphics 36*, 4 (July 2017), 1–11. doi:10.1145/3072959.3073628. 15

[LBJB23] LÖSCHNER F., BÖTTCHER T., JESKE S. R., BENDER J.: Weighted Laplacian Smoothing for Surface Reconstruction of Particle-based Fluids. In *Vision, Modeling, and Visualization* (2023), The Eurographics Association. doi:10.2312/vmv.20231245. 23

[LBK17] LIU T., BOUAZIZ S., KAVAN L.: Quasi-Newton methods for real-time simulation of hyperelastic materials. *ACM Transactions on Graphics 36*, 3 (may 2017). doi:10.1145/2990496. 16

[LBOK13] LIU T., BARGTEIL A. W., O'BRIEN J. F., KAVAN L.: Fast simulation of mass-spring systems. *ACM Transactions on Graphics 32*, 6 (Nov. 2013), 214:1–214:7. doi:10.1145/2508363.2508406. 3, 15

[LCD*20] LI W., CHEN Y., DESBRUN M., ZHENG C., LIU X.: Fast and scalable turbulent flow simulation with two-way coupling. *ACM Trans. Graph. 39*, 4 (Aug. 2020), 47:47:1–47:47:20. doi:10.1145/3386569.3392400. 29

[LCK22] LIN H., CHITALU F. M., KOMURA T.: Isotropic ARAP energy using cauchy-green invariants. *ACM Transactions on Graphics 41*, 6 (nov 2022). doi:10.1145/3550454.3555507. 9

[LD09] LENAERTS T., DUTRÉ P.: Mixing Fluids and Granular Materials. *Computer Graphics Forum 28*, 2 (Apr. 2009), 213–218. doi:10.1111/j.1467-8659.2009.01360.x. 25, 26

[LDW*22] LI Y., DU T., WU K., XU J., MATUSIK W.: Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics 42*, 1 (Oct. 2022), 1–20. doi:10.1145/3527660. 16

[LFFJ*23] LÖSCHNER F., FERNÁNDEZ-FERNÁNDEZ J. A., JESKE S. R., LONGVA A., BENDER J.: Micropolar elasticity in physically-based animation. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 3 (Aug. 2023), 1–24. doi:10.1145/3606922. 9, 33

[LFFJB24] LÖSCHNER F., FERNÁNDEZ-FERNÁNDEZ J. A., JESKE S. R., BENDER J.: Curved three-director Cosserat shells with strong coupling. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2024), SCA '24, Eurographics Association, p. 1–16. doi:10.1111/cgf.15183. 3, 9, 13, 33

[LFL*23] LI X., FANG Y., LAN L., WANG H., YANG Y., LI M., JIANG C.: Subspace-Preconditioned GPU Projective Dynamics with Contact for Cloth Simulation. In *SIGGRAPH Asia 2023 Conference Papers* (Dec. 2023), SA '23, Association for Computing Machinery, pp. 1–12. doi:10.1145/3610548.3618157. 16, 17

[LFS*20] LI M., FERGUSON Z., SCHNEIDER T., LANGLOIS T., ZORIN D., PANOZZO D., JIANG C., KAUFMAN D. M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics 39*, 4 (aug 2020). doi:10.1145/3386569.3392425. 3, 4, 9, 11, 17

[LFS*23] LI M., FERGUSON Z., SCHNEIDER T., LANGLOIS T., ZORIN D., PANOZZO D., JIANG C., KAUFMAN D. M.: Convergent incremental potential contact, July 2023. arXiv Preprint. arXiv:2307.15908. 10

[LHWW22] LIU S., HE X., WANG W., WU E.: Adapted SIMPLE Algorithm for Incompressible SPH Fluids With a Broad Range Viscosity. *IEEE Transactions on Visualization and Computer Graphics 28*, 9 (Sept. 2022), 3168–3179. doi:10.1109/TVCG.2021.3055789. 25

[LJBBD20] LY M., JOUVE J., BOISSIEUX L., BERTAILS-DESCOUBES F.: Projective dynamics with dry frictional contact. *ACM Transactions on Graphics 39*, 4 (Aug. 2020), 57:57:1–57:57:8. doi:10.1145/3386569.3392396. 17

[LJZ95] LIU W. K., JUN S., ZHANG Y. F.: Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids 20*, 8-9 (1995), 1081–1106. doi:10.1002/fld.1650200824. 22, 23

[LKJ21] LI M., KAUFMAN D. M., JIANG C.: Codimensional incremental potential contact. *ACM Transactions on Graphics 40*, 4 (jul 2021). doi:10.1145/3450626.3459767. 3, 12

[LKL*22] LAN L., KAUFMAN D. M., LI M., JIANG C., YANG Y.: Affine body dynamics: fast, stable and intersection-free simulation of stiff materials. *ACM Transactions on Graphics 41*, 4 (jul 2022). doi:10.1145/3528223.3530064. 13

[LL10] LIU M. B., LIU G. R.: Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments. *Archives of Computational Methods in Engineering 17*, 1 (Mar. 2010), 25–76. doi:10.1007/s11831-010-9040-7. 23

[LLA*24] LARIONOV E., LONGVA A., ASCHER U. M., BENDER J., PAI D. K.: Implicit frictional dynamics with soft constraints. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–12. doi:10.1109/TVCG.2024.3437417. 11

[LLDL21] LYU C., LI W., DESBRUN M., LIU X.: Fast and versatile fluid-solid coupling for turbulent flow simulation. *ACM Trans. Graph. 40*, 6 (Dec. 2021), 201:1–201:18. doi:10.1145/3478513.3480493. 29

[LLF*20] LAN L., LUO R., FRATARCANGELI M., XU W., WANG H., GUO X., YAO J., YANG Y.: Medial elastics: Efficient and collision-ready deformation via medial axis transform. *ACM Transactions on Graphics 39*, 3 (Apr. 2020), 1–17. doi:10.1145/3384515. 16

[LLFF*23] LONGVA A., LÖSCHNER F., FERNÁNDEZ-FERNÁNDEZ J. A., LARIONOV E., ASCHER U. M., BENDER J.: Pitfalls of projection: A study of newton-type solvers for incremental potentials, 2023. arXiv Preprint. arXiv:2311.14526. 6, 7

[LLH*24] LI X., LI M., HAN X., WANG H., YANG Y., JIANG C.: A Dynamic Duo of Finite Elements and Material Points. In *ACM SIGGRAPH 2024 Conference Papers* (July 2024), SIGGRAPH '24, Association for Computing Machinery, pp. 1–11. doi:10.1145/3641519.3657449. 1, 15, 31, 32

[LLJ*11] LEVIN D. I. W., LITVEN J., JONES G. L., SUEDA S., PAI D. K.: Eulerian solid simulation with contact. *ACM Transactions on Graphics 30*, 4 (July 2011). URL: https://doi.org/10.1145/2010324.1964931, doi:10.1145/2010324.1964931. 29

[LLJ22] LI X., LI M., JIANG C.: Energetically consistent inelasticity for optimization time integration. *ACM Transactions on Graphics 41*, 4 (jul 2022). doi:10.1145/3528223.3530072. 11, 31

[LLJ*23] LAN L., LI M., JIANG C., WANG H., YANG Y.: Second-order stencil descent for interior-point hyperelasticity. *ACM Transactions on Graphics 42*, 4 (July 2023), 1–16. doi:10.1145/3592104. 10

[LLK18] LI J., LIU T., KAVAN L.: Laplacian damping for projective dynamics. In *Proceedings of the 14th Workshop on Virtual Reality Interactions and Physical Simulations* (Goslar, DEU, 2018), VRIPHYS '18, Eurographics Association, p. 29–36. 17

[LLK19] LI J., LIU T., KAVAN L.: Fast simulation of deformable characters with articulated skeletons in projective dynamics. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2019), SCA '19, Association for Computing Machinery, pp. 1–10. doi:10.1145/3309486.3340249. 17

[LLK*20] LONGVA A., LÖSCHNER F., KUGELSTADT T., FERNÁNDEZ-FERNÁNDEZ J. A., BENDER J.: Higher-order finite elements for embedded simulation. *ACM Transactions on Graphics 39*, 6 (nov 2020). doi:10.1145/3414685.3417853. 9, 13

[LLK22] LI J., LIU T., KAVAN L.: Soft Articulated Characters in Projective Dynamics. *IEEE Transactions on Visualization and Computer Graphics 28*, 2 (Feb. 2022), 1385–1396. doi:10.1109/TVCG.2020.3010236. 17

[LLKC21] LI J., LIU T., KAVAN L., CHEN B.: Interactive cutting and tearing in projective dynamics with progressive cholesky updates. *ACM Transactions on Graphics 40*, 6 (Dec. 2021), 1–12. doi:10.1145/3478513.3480505. 16

[LLL*24] LAN L., LU Z., LONG J., YUAN C., LI X., HE X., WANG H., JIANG C., YANG Y.: Efficient GPU cloth simulation with non-distance barriers and subspace reuse. *ACM Transactions on Graphics 43*, 6 (Nov. 2024). doi:10.1145/3687760. 16, 17

[LMLD22] LI W., MA Y., LIU X., DESBRUN M.: Efficient kinetic simulation of two-phase flows. *ACM Trans. Graph. 41*, 4 (July 2022), 114:1–114:17. doi:10.1145/3528223.3530132. 29

[LMY*22] LAN L., MA G., YANG Y., ZHENG C., LI M., JIANG C.: Penetration-free projective dynamics on the GPU. *ACM Transactions on Graphics 41*, 4 (July 2022), 69:1–69:16. doi:10.1145/3528223.3530069. 10, 16, 17

[LQC*23] LI X., QIAO Y.-L., CHEN P. Y., JATAVALLABHULA K. M., LIN M., JIANG C., GAN C.: Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification, 2023. arXiv:2303.05512. 33

[LS81] LANCASTER P., SALKAUSKAS K.: Surfaces Generated by Moving Least Squares Methods. *Mathematics of Computation 37*, 155 (1981), 141–158. doi:10.2307/2007507. 22, 23

[LSD*22] LESSER S., STOMAKHIN A., DAVIET G., WRETBORN J., EDHOLM J., LEE N.-H., SCHWEICKART E., ZHAI X., FLYNN S., MOFFAT A.: Loki: a unified multiphysics simulation framework for production. *ACM Transactions on Graphics 41*, 4 (July 2022), 50:1–50:20. doi:10.1145/3528223.3530058. 32

[LTKF08] LOSASSO F., TALTON J., KWATRA N., FEDKIW R.: Two-Way Coupled SPH and Particle Level Set Fluid Simulation. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 797–804. URL: https://ieeexplore.ieee.org/document/4459322, doi:10.1109/TVCG.2008.37. 26

[Luc77] LUCY L. B.: A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal 82* (Dec. 1977), 1013–1024. doi:10.1086/112164. 22

[LWB*21] LIU S., WANG X., BAN X., XU Y., ZHOU J., KOSINKA J., TELEA A. C.: Turbulent Details Simulation for SPH Fluids via Vorticity Refinement. *Computer Graphics Forum 40*, 1 (2021), 54–67. doi:10.1111/cgf.14095. 25

[Mac22] MACKLIN M.: Warp: A high-performance Python framework for GPU simulation and graphics. https://github.com/nvidia/warp, March 2022. NVIDIA GPU Technology Conference (GTC). 15

[MAK24] MERCIER-AUBIN A., KRY P. G.: A Multi-Layer Solver for XPBD. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Sept. 2024), SCA '24, Eurographics Association, pp. 1–11. doi:10.1111/cgf.15186. 20

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2003), SCA '03, Eurographics Association, pp. 154–159. 25

[MCKM15] MÜLLER M., CHENTANEZ N., KIM T.-Y., MACKLIN M.: Strain based dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2015), SCA '14, Eurographics Association, pp. 149–157. 21

[MEB*14] MISZTAL M. K., ERLEBEN K., BARGTEIL A., FURSUND J., CHRISTENSEN B. B., ANDREAS BAERENTZEN J., BRIDSON R.: Multiphase flow of immiscible fluids on unstructured moving meshes. *IEEE Transactions on Visualization and Computer Graphics 20*, 1 (Jan. 2014), 4–16. doi:10.1109/tvcg.2013.97. 15

[MEM*20] MACKLIN M., ERLEBEN K., MÜLLER M., CHENTANEZ N., JESCHKE S., KIM T.: Primal/Dual Descent Methods for Dynamics. *Computer Graphics Forum 39*, 8 (2020), 89–100. doi:10.1111/cgf.14104. 10, 11, 13, 16

[MHHR06] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position Based Dynamics. In *Vriphys: 3rd Workshop in Virtual Reality, Interactions, and Physical Simulation* (2006), The Eurographics Association. doi:10.2312/PE/vriphys/vriphys06/071-080. 18, 19, 21, 22

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Transactions on Graphics 24*, 3 (July 2005), 471–478. doi:10.1145/1073204.1073216. 21

[MKB*10] MARTIN S., KAUFMANN P., BOTSCH M., GRINSPUN E., GROSS M.: Unified simulation of elastic rods, shells, and solids. *ACM*

Transactions on Graphics 29, 4 (July 2010), 1–10. doi:10.1145/1778765.1778776. 23

[MKC12] MÜLLER M., KIM T.-Y., CHENTANEZ N.: Fast Simulation of Inextensible Hair and Fur. In *VRIPHYS 12: 9th Workshop on Virtual Reality Interactions and Physical Simulations, Darmstadt, Germany, 2012. Proceedings* (2012), Eurographics Association, pp. 39–44. doi:10.2312/PE/VRIPHYS/VRIPHYS12/039-044. 21

[MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), p. 141. doi:10.1145/1028523.1028542. 23

[ML98] MCLEISH T., LARSON R.: Molecular constitutive equations for a class of branched polymers: The pom-pom polymer. *Journal of rheology 42*, 1 (1998), 81–110. 31

[MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM Transactions on Graphics 32*, 4 (July 2013), 104:1–104:12. doi:10.1145/2461912.2461984. 17, 20, 21, 24

[MM21] MACKLIN M., MÜLLER M.: A Constraint-based Formulation of Stable Neo-Hookean Materials. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Nov. 2021), MIG '21, Association for Computing Machinery, pp. 1–7. doi:10.1145/3487983.3488289. 21

[MMC16] MACKLIN M., MÜLLER M., CHENTANEZ N.: XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games* (Oct. 2016), MIG '16, Association for Computing Machinery, pp. 49–54. doi:10.1145/2994258.2994272. 15, 19, 20, 21, 22

[MMC*20] MÜLLER M., MACKLIN M., CHENTANEZ N., JESCHKE S., KIM T.-Y.: Detailed Rigid Body Simulation with Extended Position Based Dynamics. *Computer Graphics Forum 39*, 8 (2020), 101–112. doi:10.1111/cgf.14105. 21

[MMCK14] MACKLIN M., MÜLLER M., CHENTANEZ N., KIM T.-Y.: Unified particle physics for real-time applications. *ACM Transactions on Graphics 33*, 4 (July 2014), 153:1–153:12. doi:10.1145/2601097.2601152. 19, 20, 21, 33

[MSAO18] M. SÁNCHEZ-BANDERAS R., A. OTADUY M.: Strain rate dissipation for elastic deformations. *Computer Graphics Forum 37*, 8 (Sept. 2018), 161–170. doi:10.1111/cgf.13521. 10

[MSL*19] MACKLIN M., STOREY K., LU M., TERDIMAN P., CHENTANEZ N., JESCHKE S., MÜLLER M.: Small steps in physics simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2019), SCA '19, Association for Computing Machinery, pp. 1–7. doi:10.1145/3309486.3340247. 20

[MSP*24] MODI V., SHARP N., PEREL O., SUEDA S., LEVIN D. I. W.: Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM Transactions on Graphics 43*, 4 (July 2024). doi:10.1145/3658184. 33

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM Transactions on Graphics 30*, 4 (jul 2011). doi:10.1145/2010324.1964967. 3

[Mus13] MUSETH K.: Vdb: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics 32*, 3 (July 2013). URL: https://doi.org/10.1145/2487228.2487235, doi:10.1145/2487228.2487235. 28

[MZS*11] MCADAMS A., ZHU Y., SELLE A., EMPEY M., TAMSTORF R., TERAN J., SIFAKIS E.: Efficient elasticity for character skinning with contact and collisions. *ACM Transactions on Graphics 30*, 4 (July 2011). doi:10.1145/2010324.1964932. 9

[Mü08] MÜLLER M.: Hierarchical Position Based Dynamics. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2008)* (2008), The Eurographics Association. doi:/10.2312/PE/vriphys/vriphys08/001-010. 20

[NVI24] NVIDIA: NVIDIA Flex 1.1.0 Documentation, 2024. URL: https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/flex. 18, 33

[NW06] NOCEDAL J., WRIGHT S.: *Numerical optimization*. Springer Science & Business Media, 2006. doi:10.1007/978-0-387-40065-5. 4, 6

[OBLN17] OVERBY M., BROWN G. E., LI J., NARAIN R.: Admm ⊇ projective dynamics: Fast simulation of hyperelastic models with dynamic constraints. *IEEE Transactions on Visualization and Computer Graphics 23*, 10 (Oct. 2017), 2222–2234. doi:10.1109/tvcg.2017.2730875. 16

[OMTA12] OISHI C. M., MARTINS F. P., TOMÉ M. F., ALVES M. A.: Numerical simulation of drop impact and jet buckling problems using the extended pom–pom model. *Journal of Non-Newtonian Fluid Mechanics 169* (2012), 91–103. 31

[PC13] PATKAR S., CHAUDHURI P.: Wetting of Porous Solids. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (Sept. 2013), 1592–1604. doi:10.1109/TVCG.2013.8. 25

[Pes02] PESKIN C. S.: The immersed boundary method. *Acta numerica 11* (2002), 479–517. doi:10.1017/S0962492902000077. 17, 29

[PGBT18] PEER A., GISSLER C., BAND S., TESCHNER M.: An Implicit SPH Formulation for Incompressible Linearly Elastic Solids. *Computer Graphics Forum 37*, 6 (2018), 135–148. doi:10.1111/cgf.13317. 26

[PKA*05] PAULY M., KEISER R., ADAMS B., DUTRÉ P., GROSS M., GUIBAS L. J.: Meshless Animation of Fracturing Solids. *ACM Transactions on Graphics 24*, 3 (2005), 957–964. doi:10.1145/1186822.1073296. 23

[Pri12] PRICE D. J.: Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics 231*, 3 (Feb. 2012), 759–794. doi:10.1016/j.jcp.2010.12.011. 23, 25

[PT17] PEER A., TESCHNER M.: Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion. *IEEE Transactions on Visualization and Computer Graphics 23*, 12 (Dec. 2017), 2656–2662. doi:10.1109/TVCG.2016.2636144. 25

[PT23] PROBST T., TESCHNER M.: Monolithic Friction and Contact Handling for Rigid Bodies and Fluids Using SPH. *Computer Graphics Forum 42*, 1 (Feb. 2023), 155–179. doi:10.1111/cgf.14727. 26

[QLDGJ22] QU Z., LI M., DE GOES F., JIANG C.: The power particle-in-cell method. *ACM Transactions on Graphics 41*, 4 (2022). 30

[QLY*23] QU Z., LI M., YANG Y., JIANG C., DE GOES F.: Power plastics: A hybrid lagrangian/eulerian solver for mesoscale inelastic flows. *ACM Transactions on Graphics 42*, 6 (2023), 1–11. 30, 31

[QRL*23] QIU Y., REEVE S. T., LI M., YANG Y., SLATTERY S. R., JIANG C.: A sparse distributed gigascale resolution material point method. *ACM Transactions on Graphics 42*, 2 (2023), 1–21. 30

[RGJ*15] RAM D., GAST T., JIANG C., SCHROEDER C., STOMAKHIN A., TERAN J., KAVEHPOUR P.: A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Aug. 2015), SCA '15, Association for Computing Machinery, pp. 157–163. doi:10.1145/2786784.2786798. 30, 31

[RHLC22] REN B., HE W., LI C.-F., CHEN X.: Incompressibility Enforcement for Multiple-Fluid SPH Using Deformation Gradient. *IEEE Transactions on Visualization and Computer Graphics 28*, 10 (Oct. 2022), 3417–3427. doi:10.1109/TVCG.2021.3062643. 25

[RLY*14] REN B., LI C., YAN X., LIN M. C., BONET J., HU S.-M.: Multiple-Fluid SPH Simulation Using a Mixture Model. *ACM Transactions on Graphics 33*, 5 (Sept. 2014), 171:1–171:11. doi:10.1145/2645703. 25

[RMEF09] ROBINSON-MOSHER A., ENGLISH R. E., FEDKIW R.: Accurate tangential velocities for solid fluid coupling. In *Proceedings of*

the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (New York, NY, USA, 2009), SCA '09, Association for Computing Machinery, p. 227–236. URL: https://doi.org/10.1145/1599470.1599500, doi:10.1145/1599470.1599500. 29

[RMSG*08] ROBINSON-MOSHER A., SHINAR T., GRETARSSON J., SU J., FEDKIW R.: Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1–9. URL: https://doi.org/10.1145/1360612.1360645, doi:10.1145/1360612.1360645. 29

[RO99] RADOVITZKY R., ORTIZ M.: Error estimation and adaptive meshing in strongly nonlinear dynamic problems. *Computer Methods in Applied Mechanics and Engineering 172*, 1 (1999), 203–240. doi:10.1016/S0045-7825(98)00230-8. 3

[RSCO25] ROMANYÀ-SERRASOLSAS M., CASAFRANCA J. J., OTADUY M. A.: Painless differentiable rotation dynamics. *ACM Transactions on Graphics 44*, 4 (July 2025), 1–13. doi:10.1145/3730944. 13

[RXL21] REN B., XU B., LI C.: Unified particle system for multiple-fluid flow and porous material. *ACM Transactions on Graphics 40*, 4 (July 2021), 118:1–118:14. doi:10.1145/3450626.3459764. 25

[RZF05] ROBLE D., ZAFAR N. B., FALT H.: Cartesian grid fluid simulation with irregular boundary voxels. In *ACM SIGGRAPH 2005 Sketches* (2005), SIGGRAPH '05, Association for Computing Machinery, p. 138–es. doi:10.1145/1187112.1187279. 29

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing* (2007), SGP '07, Eurographics Association. doi:10.2312/SGP/SGP07/109-116. 9

[SABS14] SETALURI R., AANJANEYA M., BAUER S., SIFAKIS E.: Sp-grid: a sparse paged grid structure applied to adaptive smoke simulation. *ACM Transactions on Graphics 33*, 6 (Nov. 2014). doi:10.1145/2661229.2661269. 28

[SB12a] SCHECHTER H., BRIDSON R.: Ghost SPH for animating water. *ACM Transactions on Graphics 31*, 4 (2012), 61:1–61:8. doi:10.1145/2185520.2335412. 24

[SB12b] SIFAKIS E., BARBIC J.: FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (Aug. 2012), SIGGRAPH '12, ACM. doi:10.1145/2343483.2343501. 9

[SBB*22] SCHMIDT P., BORN J., BOMMES D., CAMPEN M., KOBBELT L.: Tinyad: Automatic differentiation in geometry processing made simple. *Computer Graphics Forum 41*, 5 (2022), 113–124. doi:https://doi.org/10.1111/cgf.14607. 8

[SBC*11] SOLENTHALER B., BUCHER P., CHENTANEZ N., MÜLLER M., GROSS M.: SPH Based Shallow Water Simulation. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2011)* (2011), Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2011), The Eurographics Association, p. 8 pages. URL: http://diglib.eg.org/handle/10.2312/PE.vriphys.vriphys11.039-046, doi:10.2312/PE/VRIPHYS/VRIPHYS11/039-046. 25

[SC23] STUYCK T., CHEN H.-Y.: Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 3 (Aug. 2023), 1–14. doi:10.1145/3606923. 20

[SCBL24] SHEN X., CAI R., BI M., LV T.: Preconditioned nonlinear conjugate gradient method for real-time interior-point hyperelasticity. In *ACM SIGGRAPH 2024 Conference Papers* (2024), SIGGRAPH '24, Association for Computing Machinery. doi:10.1145/3641519.3657490. 10

[SDG*19] SCHNEIDER T., DUMAS J., GAO X., ZORIN D., PANOZZO D.: PolyFEM. https://polyfem.github.io/, 2019. 9

[SGK18] SMITH B., GOES F. D., KIM T.: Stable neo-hookean flesh simulation. *ACM Transactions on Graphics 37*, 2 (mar 2018). doi:10.1145/3180491. 9

[SHD*18] SCHNEIDER T., HU Y., DUMAS J., GAO X., PANOZZO D., ZORIN D.: Decoupling simulation accuracy from mesh quality. *ACM Transactions on Graphics 37*, 6 (dec 2018). doi:10.1145/3272127.3275067. 9

[SHG*22] SCHNEIDER T., HU Y., GAO X., DUMAS J., ZORIN D., PANOZZO D.: A large-scale comparison of tetrahedral and hexahedral elements for solving elliptic PDEs with the finite element method. *ACM Transactions on Graphics 41*, 3 (mar 2022). doi:10.1145/3508372. 9

[SHST12] STOMAKHIN A., HOWES R., SCHROEDER C., TERAN J. M.: Energetically consistent invertible elasticity. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), SCA '12, Eurographics Association, p. 25–32. 9

[Sid24] SIDEFX: Houdini 20.0 Documentation, 2024. URL: https://www.sidefx.com/docs/houdini. 18

[SK23] SHI A., KIM T.: A unified analysis of penalty-based collision energies. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 3 (aug 2023). 10

[SLM06] SERVIN M., LACOURSIERE C., MELIN N.: Interactive simulation of elastic deformable materials. In *Proceedings of SIGRAD Conference* (2006), Linköping University Electronic Press, pp. 22–32. 17, 19, 21

[SLZ17] SHAO X., LIAO E., ZHANG F.: Improving SPH Fluid Simulation Using Position Based Dynamics. *IEEE Access 5* (2017), 13901–13908. doi:10.1109/ACCESS.2017.2729601. 21

[SMSH18] SOLER C., MARTIN T., SORKINE-HORNUNG O.: Cosserat Rods with Projective Dynamics. *Computer Graphics Forum 37*, 8 (2018), 137–147. doi:10.1111/cgf.13519. 17

[SNZ*21] SUN Y., NI X., ZHU B., WANG B., CHEN B.: A material point method for nonlinearly magnetized materials. *ACM Transactions on Graphics 40*, 6 (2021), 1–13. 30

[SP08] SOLENTHALER B., PAJAROLA R.: Density contrast SPH interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2008), SCA '08, Eurographics Association, pp. 211–218. 25

[SP09] SOLENTHALER B., PAJAROLA R.: Predictive-corrective incompressible SPH. In *ACM SIGGRAPH 2009 papers* (July 2009), SIGGRAPH '09, Association for Computing Machinery, pp. 1–6. doi:10.1145/1576246.1531346. 24

[SSC*13] STOMAKHIN A., SCHROEDER C., CHAI L., TERAN J., SELLE A.: A material point method for snow simulation. *ACM Transactions on Graphics 32*, 4 (July 2013), 102:1–102:10. doi:10.1145/2461912.2461948. 30, 31

[SSJ*14] STOMAKHIN A., SCHROEDER C., JIANG C., CHAI L., TERAN J., SELLE A.: Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics 33*, 4 (July 2014), 138:1–138:11. doi:10.1145/2601097.2601176. 30, 31

[SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid–solid interactions. *Computer Animation and Virtual Worlds 18*, 1 (Feb. 2007), 69–82. doi:10.1002/cav.162. 26, 27

[SSS20] SUN Y., SHINAR T., SCHROEDER C.: Effective time step restrictions for explicit mpm simulation. *Computer Graphics Forum 39*, 8 (2020), 55–67. 30

[ST07] SPILLMANN J., TESCHNER M.: CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, Eurographics Association, p. 63–72. 13

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., p. 121–128. doi:10.1145/311535.311548. 28

[STBA24] SANCHO S., TANG J., BATTY C., AZEVEDO V. C.: The impulse particle-in-cell method. *Computer Graphics Forum 43*, 2 (Apr. 2024). doi:10.1111/cgf.15022. 30

[SWP*23] SHI A., WU H., PARR J., DARKE A., KIM T.: Lifted Curls: A Model for Tightly Coiled Hair Simulation. In *Proceedings of the ACM on Computer Graphics and Interactive Techniques* (2023), ACM Association for Computing Machinery. doi:10.1145/3606920. 13

[SXH*21] SU H., XUE T., HAN C., JIANG C., AANJANEYA M.: A unified second-order accurate in time mpm formulation for simulating viscoelastic liquids with phase change. *ACM Transactions on Graphics 40*, 4 (2021), 1–18. 30, 31

[SYS*21] SHEN S., YANG Y., SHAO T., WANG H., JIANG C., LAN L., ZHOU K.: High-order differentiable autoencoder for nonlinear model reduction. *ACM Trans. Graph. 40*, 4 (July 2021). URL: https://doi.org/10.1145/3450626.3459754, doi:10.1145/3450626.3459754. 33

[SZDJ24] SAILLANT B., ZARA F., DAMIAND G., JAILLET F.: High-order elements in position-based dynamics. *The Visual Computer 40*, 7 (June 2024), 4737–4749. doi:10.1007/s00371-024-03467-3. 21

[SZS95] SULSKY D., ZHOU S.-J., SCHREYER H. L.: Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications 87*, 1-2 (May 1995), 236–252. doi:10.1016/0010-4655(94)00170-7. 30

[SZYA24] SHI H., ZORDAN V., YANG Y., ANDREWS S.: Adaptive distributed simulation of fluids and rigid bodies. In *Proceedings of the 17th ACM SIGGRAPH Conference on Motion, Interaction, and Games* (New York, NY, USA, 2024), MIG '24, Association for Computing Machinery. URL: https://doi.org/10.1145/3677388.3696334, doi:10.1145/3677388.3696334. 29

[TB20] TAKAHASHI T., BATTY C.: Monolith: a monolithic pressure-viscosity-contact solver for strong two-way rigid-rigid rigid-fluid coupling. *ACM Transactions on Graphics 39*, 6 (Nov. 2020), 1–16. doi:10.1145/3414685.3417798. 15, 29

[TB21] TAKAHASHI T., BATTY C.: FrictionalMonolith: a monolithic optimization-based approach for granular flow with contact-aware rigid-body coupling. *ACM Transactions on Graphics 40*, 6 (Dec. 2021), 1–20. doi:10.1145/3478513.3480539. 15

[TB22] TAKAHASHI T., BATTY C.: ElastoMonolith: A monolithic optimization-based liquid solver for contact-aware elastic-solid coupling. *ACM Transactions on Graphics 41*, 6 (Nov. 2022), 1–19. doi:10.1145/3550454.3555474. 3, 15, 29

[TBFL19] TAO M., BATTY C., FIUME E., LEVIN D. I. W.: Mandoline: robust cut-cell generation for arbitrary triangle meshes. *ACM Transactions on Graphics 38*, 6 (Nov. 2019), 1–17. doi:10.1145/3355089.3356543. 29

[TBHF03] TERAN J., BLEMKER S., HING V. N. T., FEDKIW R.: Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2003), SCA '03, Eurographics Association, p. 68–74. 27, 29

[TDF*15] TAKAHASHI T., DOBASHI Y., FUJISHIRO I., NISHITA T., LIN M. C.: Implicit Formulation for SPH-based Viscous Fluids. *Computer Graphics Forum 34*, 2 (2015), 493–502. doi:10.1111/cgf.12578. 25

[TG13] TAMSTORF R., GRINSPUN E.: Discrete bending forces and their Jacobians. *Graph. Models 75*, 6 (nov 2013), 362–370. doi:10.1016/j.gmod.2013.07.001. 12

[TGK*17] TAMPUBOLON A. P., GAST T., KLÁR G., FU C., TERAN J., JIANG C., MUSETH K.: Multi-species simulation of porous sand and water mixtures. *ACM Transactions on Graphics 36*, 4 (2017), 1–11. 30, 31

[TLK16] TENG Y., LEVIN D. I. W., KIM T.: Eulerian solid-fluid coupling. *ACM Transactions on Graphics 35*, 6 (Dec. 2016). URL: https://doi.org/10.1145/2980179.2980229, doi:10.1145/2980179.2980229. 29

[TLZ*24] TU Z., LI C., ZHAO Z., LIU L., WANG C., WANG C., QIN H.: A Unified MPM Framework Supporting Phase-field Models and Elastic-viscoplastic Phase Transition. *ACM Transactions on Graphics 43*, 2 (Apr. 2024), 1–19. doi:10.1145/3638047. 2, 30, 31, 32

[TNF14] TAKAHASHI T., NISHITA T., FUJISHIRO I.: Fast simulation of viscous fluids with elasticity and thermal conductivity using position-based dynamics. *Computers & Graphics 43* (Oct. 2014), 21–30. doi:10.1016/j.cag.2014.06.002. 20

[TPNK24] TAO Y., PUHACHOV I., NOWROUZEZAHRAI D., KRY P.: Neural implicit reduced fluid simulation. In *SIGGRAPH Asia 2024 Conference Papers* (New York, NY, USA, 2024), SA '24, Association for Computing Machinery. URL: https://doi.org/10.1145/3680528.3687628, doi:10.1145/3680528.3687628. 33

[TSIF05] TERAN J., SIFAKIS E., IRVING G., FEDKIW R.: Robust quasistatic finite elements and flesh simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), SCA '05, Association for Computing Machinery, p. 181–190. doi:10.2312/SCA/SCA05/181-190. 7

[TSM*16] TASORA A., SERBAN R., MAZHAR H., PAZOUKI A., MELANZ D., FLEISCHMANN J., TAYLOR M., SUGIYAMA H., NEGRUT D.: Chrono: An Open Source Multi-physics Dynamics Engine. In *High Performance Computing in Science and Engineering*, vol. 9611. Springer International Publishing, 2016, pp. 19–49. doi:10.1007/978-3-319-40361-8_2. 32

[TTKA23] TON-THAT Q.-M., KRY P. G., ANDREWS S.: Parallel block Neo-Hookean XPBD using graph clustering. *Computers & Graphics 110*, C (Feb. 2023), 1–10. doi:10.1016/j.cag.2022.10.009. 20, 21

[TUKF02] TAKAHASHI T., UEKI H., KUNIMATSU A., FUJII H.: The simulation of fluid-rigid body interaction. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications* (2002), SIGGRAPH '02, Association for Computing Machinery, p. 266. doi:10.1145/1242073.1242279. 29

[TWS06] THOMASZEWSKI B., WACKER M., STRASSER W.: A Consistent Bending Model for Cloth Simulation with Corotational Subdivision Finite Elements. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2006), The Eurographics Association. doi:/10.2312/SCA/SCA06/107-116. 13

[USS15] UMETANI N., SCHMIDT R., STAM J.: Position-based elastic rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2015), SCA '14, Eurographics Association, pp. 21–30. 21

[VPB01] VERBEETEN W. M., PETERS G. W., BAAIJENS F. P.: Differential constitutive equations for polymer melts: The extended pom–pom model. *Journal of rheology 45*, 4 (2001), 823–843. 31

[WAK20] WINCHENBACH R., AKHUNOV R., KOLB A.: Semi-analytic boundary handling below particle resolution for smoothed particle hydrodynamics. *ACM Transactions on Graphics 39*, 6 (Nov. 2020), 173:1–173:17. doi:10.1145/3414685.3417829. 26

[Wan15] WANG H.: A Chebyshev semi-iterative approach for accelerating projective and position-based dynamics. *ACM Transactions on Graphics 34*, 6 (Nov. 2015), 246:1–246:9. doi:10.1145/2816795.2818063. 16, 20

[WB23] WEN J., BARBIČ J.: Kirchhoff-Love shells with arbitrary hyperelastic materials. *ACM Transactions on Graphics 42*, 6 (dec 2023). doi:10.1145/3618405. 13

[WBG07] WICKE M., BOTSCH M., GROSS M.: A finite element method on convex polyhedra. *Computer Graphics Forum 26*, 3 (Sept. 2007), 355–364. doi:10.1111/j.1467-8659.2007.01058.x. 9

[WDG*19] WANG S., DING M., GAST T. F., ZHU L., GAGNIERE S., JIANG C., TERAN J. M.: Simulation and visualization of ductile fracture with the material point method. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (2019), 1–20. 30, 31

[WDK*21] WANG M., DENG Y., KONG X., PRASAD A. H., XIONG S., ZHU B.: Thin-film smoothed particle hydrodynamics fluid. *ACM Transactions on Graphics 40*, 4 (July 2021), 110:1–110:16. doi:10.1145/3450626.3459864. 25

[Wei12] WEISCHEDEL C.: *A discrete geometric view on shear-deformable shell models*. PhD thesis, Georg-August-Universität Göttingen, 2012. doi:10.53846/goediss-2453. 13

[WFFJB24] WESTHOFEN L., FERNÁNDEZ-FERNÁNDEZ J. A., JESKE S. R., BENDER J.: Strongly Coupled Simulation of Magnetic Rigid Bodies. *Computer Graphics Forum* (2024). doi:10.1111/cgf.15185. 3, 14, 33

[WFL*19] WOLPER J., FANG Y., LI M., LU J., GAO M., JIANG C.: Cd-mpm: continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics 38*, 4 (July 2019). doi:10.1145/3306346.3322949. 30, 31

[WFM21] WANG X., FUJISAWA M., MIKAWA M.: Visual Simulation of Soil-Structure Destruction with Seepage Flows. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 4*, 3 (Sept. 2021), 41:1–41:18. doi:10.1145/3480141. 25

[WHP11] WAWRZINEK A., HILDEBRANDT K., POLTHIER K.: Koiter's Thin Shells on Catmull-Clark Limit Surfaces. In *Vision, Modeling, and Visualization (2011)* (2011), The Eurographics Association. doi:/10.2312/PE/VMV/VMV11/113-120. 13

[WHS*24] WILLIAMS F., HUANG J., SWARTZ J., KLAR G., THAKKAR V., CONG M., REN X., LI R., FUJI-TSANG C., FIDLER S., SIFAKIS E., MUSETH K.: fvdb: A deep-learning framework for sparse, large scale, and high performance spatial intelligence. *ACM Trans. Graph. 43*, 4 (July 2024). URL: https://doi.org/10.1145/3658226, doi:10.1145/3658226. 28

[WJB23] WESTHOFEN L., JESKE S., BENDER J.: A comparison of linear consistent correction methods for first-order SPH derivatives. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 3 (Aug. 2023), 48:1–48:20. doi:10.1145/3606933. 23

[WJL*20] WANG H., JIN Y., LUO A., YANG X., ZHU B.: Codimensional surface tension flow using moving-least-squares particles. *ACM Transactions on Graphics 39*, 4 (Aug. 2020), 42:42:1–42:42:14. doi:10.1145/3386569.3392487. 25

[WK23] WU H., KIM T.: An eigenanalysis of angle-based deformation energies. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 3 (aug 2023). doi:10.1145/3606929. 12

[WKB16] WEILER M., KOSCHIER D., BENDER J.: Projective fluids. In *Proceedings of the 9th International Conference on Motion in Games* (Oct. 2016), ACM, pp. 79–84. doi:10.1145/2994258.2994282. 16, 17, 24

[WKBB18] WEILER M., KOSCHIER D., BRAND M., BENDER J.: A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum 37*, 2 (2018), 145–155. doi:10.1111/cgf.13349. 25

[WLF*20] WANG X., LI M., FANG Y., ZHANG X., GAO M., TANG M., KAUFMAN D. M., JIANG C.: Hierarchical optimization time integration for cfl-rate mpm stepping. *ACM Transactions on Graphics 39*, 3 (2020), 1–16. 30

[WQS*20] WANG X., QIU Y., SLATTERY S. R., FANG Y., LI M., ZHU S.-C., ZHU Y., TANG M., MANOCHA D., JIANG C.: A massively parallel and scalable multi-GPU material point method. *ACM Transactions on Graphics 39*, 4 (2020), 30–1. 30

[Wri15] WRIGHT S. J.: Coordinate descent algorithms. *Mathematical Programming 151*, 1 (Mar. 2015), 3–34. doi:10.1007/s10107-015-0892-3. 15

[WTB*21] WANG Q., TAO Y., BRANDT E., CUTTING C., SIFAKIS E.: Optimized Processing of Localized Collisions in Projective Dynamics. *Computer Graphics Forum 40*, 6 (2021), 382–393. doi:10.1111/cgf.14385. 16, 17

[WYW23] WANG Z., YANG Y., WANG H.: Stable discrete bending by analytic eigensystem and adaptive orthotropic geometric stiffness. *ACM Transactions on Graphics 42*, 6 (Dec. 2023), 1–16. doi:10.1145/3618372. 12

[XLYJ23] XIE T., LI M., YANG Y., JIANG C.: A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling. *ACM Transactions on Graphics 42*, 4 (July 2023), 122:1–122:14. doi:10.1145/3592115. 14, 17, 27

[XRW*22] XING J., RUAN L., WANG B., ZHU B., CHEN B.: Position-Based Surface Tension Flow. *ACM Transactions on Graphics 41*, 6 (Nov. 2022), 244:1–244:12. doi:10.1145/3550454.3555476. 20, 25

[XWW*23] XU Y., WANG X., WANG J., SONG C., WANG T., ZHANG Y., CHANG J., ZHANG J. J., KOSINKA J., TELEA A., BAN X.: An Implicitly Stable Mixture Model for Dynamic Multi-fluid Simulations. In *SIGGRAPH Asia 2023 Conference Papers* (Dec. 2023), SA '23, Association for Computing Machinery, pp. 1–11. doi:10.1145/3610548.3618215. 25

[XZQ*24] XIE T., ZONG Z., QIU Y., LI X., FENG Y., YANG Y., JIANG C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jun 2024), IEEE Computer Society, pp. 4389–4398. doi:10.1109/CVPR52733.2024.00420. 33

[YCL*17] YANG T., CHANG J., LIN M. C., MARTIN R. R., ZHANG J. J., HU S.-M.: A unified particle system framework for multi-phase, multi-material visual simulations. *ACM Transactions on Graphics 36*, 6 (Nov. 2017), 224:1–224:13. doi:10.1145/3130800.3130882. 27

[YCR*15] YANG T., CHANG J., REN B., LIN M. C., ZHANG J. J., HU S.-M.: Fast multiple-fluid simulation using Helmholtz free energy. *ACM Transactions on Graphics 34*, 6 (Nov. 2015), 201:1–201:11. doi:10.1145/2816795.2818117. 25

[YJL*16] YAN X., JIANG Y.-T., LI C.-F., MARTIN R. R., HU S.-M.: Multiphase SPH simulation for interactive fluids and solids. *ACM Transactions on Graphics 35*, 4 (July 2016), 79:1–79:11. doi:10.1145/2897824.2925897. 25

[YLL*24] YU C., LI X., LAN L., YANG Y., JIANG C.: XPBI: Position-Based Dynamics with Smoothing Kernels Handles Continuum Inelasticity, Sept. 2024. arXiv:2405.11694 [cs]. doi:10.1145/3680528.3687577. 1, 18, 22

[YML*17] YANG T., MARTIN R. R., LIN M. C., CHANG J., HU S.-M.: Pairwise Force SPH Model for Real-Time Multi-Interaction Applications. *IEEE Transactions on Visualization and Computer Graphics 23*, 10 (Oct. 2017), 2235–2247. doi:10.1109/TVCG.2017.2706289. 25

[YR23] YAN H., REN B.: High Density Ratio Multi-Fluid Simulation with Peridynamics. *ACM Transactions on Graphics 42*, 6 (Dec. 2023), 191:1–191:14. doi:10.1145/3618347. 25

[YSB*15] YUE Y., SMITH B., BATTY C., ZHENG C., GRINSPUN E.: Continuum Foam: A Material Point Method for Shear-Dependent Flows. *ACM Transactions on Graphics 34*, 5 (Nov. 2015), 160:1–160:20. doi:10.1145/2751541. 30, 31

[YWX*24] YE X., WANG X., XU Y., KOSINKA J., TELEA A. C., YOU L., ZHANG J. J., CHANG J.: Monte Carlo Vortical Smoothed Particle Hydrodynamics for Simulating Turbulent Flows. *Computer Graphics Forum 43*, 2 (2024), e15024. doi:10.1111/cgf.15024. 25

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Transactions on Graphics 24*, 3 (July 2005), 965–972. doi:10.1145/1073204.1073298. 17, 26, 30

[ZB17] ZARIFI O., BATTY C.: A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (July 2017), vol. 2003 of *SCA '17*, ACM, pp. 1–11. doi:10.1145/3099564.3099572. 29

[ZLL*23] ZONG Z., LI X., LI M., CHIARAMONTE M. M., MATUSIK W., GRINSPUN E., CARLBERG K., JIANG C., CHEN P. Y.: Neural stress fields for reduced-order elastoplasticity and fracture. In *SIGGRAPH Asia 2023 Conference Papers* (2023), SA '23, Association for Computing Machinery. doi:10.1145/3610548.3618207. 33

[ZLQF15] ZHU B., LEE M., QUIGLEY E., FEDKIW R.: Codimensional non-Newtonian fluids. *ACM Transactions on Graphics 34*, 4 (July 2015), 115:1–115:9. doi:10.1145/2766981. 15

[ZLX*24] ZHANG Y., LONG S., XU Y., WANG X., YAO C., KOSINKA J., FREY S., TELEA A., BAN X.: Multiphase Viscoelastic Non-Newtonian Fluid Simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Sept. 2024), SCA '24, Eurographics Association, pp. 1–12. doi:10.1111/cgf.15180. 25

[ZQC*14] ZHU B., QUIGLEY E., CONG M., SOLOMON J., FEDKIW R.: Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics 33*, 4 (July 2014), 111:1–111:11. doi:10.1145/2601097.2601201. 15

[ZRS*20] ZORILLA F., RITTER M., SAPPL J., RAUCH W., HARDERS M.: Accelerating Surface Tension Calculation in SPH via Particle Classification and Monte Carlo Integration. *Computers 9*, 2 (June 2020), 23. doi:10.3390/computers9020023. 25