Adaptive Physically Based Models in Computer Graphics

Pierre-Luc Manteaux, Chris Wojtan, Rahul Narain, Stéphane Redon, François Faure, Marie-Paule Cani

Adaptive Physically-Based Models in Computer Graphics

P.-L. Manteaux[1] C. Wojtan[2], R. Narain[3], S. Redon[1], F. Faure[1], M.-P. Cani[1]

[1]LJK, Univ. Grenoble-Alpes, CNRS & Inria †
[2]IST Austria ‡
[3]University of Minnesota §

Abstract

One of the major challenges in physically-based modeling is making simulations efficient. Adaptive models provide an essential solution to these efficiency goals. These models are able to self-adapt in space and time, attempting to provide the best possible compromise between accuracy and speed. This survey reviews the adaptive solutions proposed so far in computer graphics. Models are classified according to the strategy they use for adaptation, from time-stepping and freezing techniques to geometric adaptivity in the form of structured grids, meshes, and particles. Applications range from fluids, through deformable bodies, to articulated solids.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Computer Graphics—Three-Dimensional Graphics and Realism-Animation

Keywords: adaptivity, physically-based animation

1. Introduction

The ability of physically-based animation to generate highly complex and realistic motion, as well as react to unpredictable input, has made it increasingly indispensable in movies, video games and simulators since its introduction in 1987 by Terzopoulos et al. [TPBF87]. Complex real-world behaviors exhibit multiscale phenomena in space and time. Sophisticated physical models involving a tremendous number of degrees of freedom are required to accurately reproduce these phenomena. This comes at a high memory and computational price, thus practitioners and researchers have been striving to achieve good trade-offs between accuracy and computational resources. This is especially relevant in computer graphics, where visual plausibility is the main quality criterion, and high-resolution results with rich details are sought. Trade-offs are typically obtained by carefully choosing the constitutive laws of the modeled objects, while sampling time and space appropriately. However, a model suitable for a given viewpoint may become overly coarse when the object comes closer to the camera, or waste computational resources when it is no longer visible. Variations in external loading, large deformations, or topology changes may also require model updates during the simulation. Carrying on a simulation across a wide range of viewpoints and changes to the object, while simultaneously optimizing efficiency and visual realism, requires the simulation model to adapt to the changing circumstances of the simulation. These adaptive simulation techniques are the focus of this report.

We call a model or simulation method *adaptive* if it automatically adapts the underlying mathematical representation, data structure and/or algorithm at run time, based on the evolving state of the simulated system. The adaptation is designed for meeting a given criteria which depends on the application. Examples of frequently used criteria include reducing the overall computational complexity without loss of quality, improving the quality of real-time simulation, or simulating more precisely the parts of the scene with which the user is currently interacting.

Computer animation has come a long way since Carlson and Hodgins proposed to switch between dynamic, kinematic and hybrid models for jumping creatures in 1997 [CH97], and the variety of approaches and sub-

† { pierre-luc.manteaux|stephane.redon|françois.faure|marie-paule.cani } @inria.fr
‡ wojtan@ist.ac.at
§ narain@umn.edu

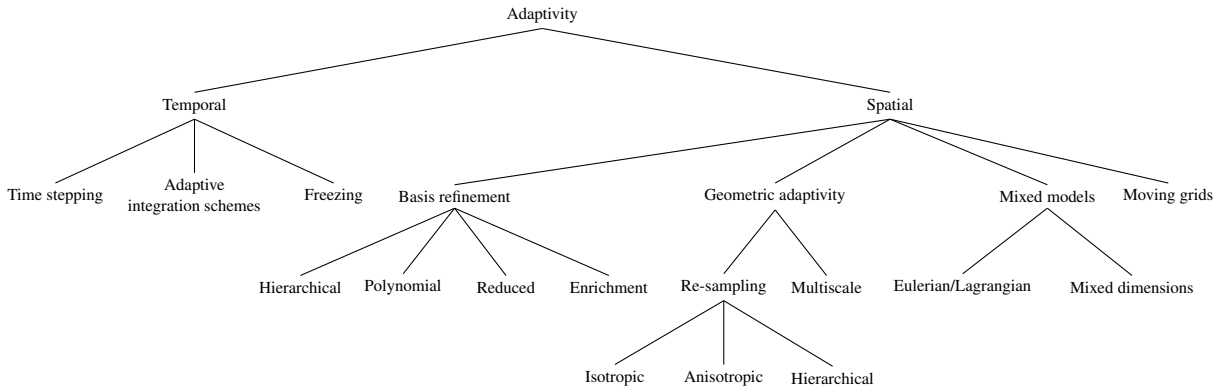


Figure 1: Taxonomy of adaptive physically-based models in Computer Graphics. In this report, we discuss in detail the two most widely used adaptive techniques: Temporal adaptivity in Section 2 and geometric adaptivity in Section 3. We also discuss less common spatial adaptive techniques that proved to be very promising: Basis refinement in Section 4.1, moving grids in Section 4.2 and mixed models in Section 4.3.

domains makes it difficult to sort. Rather than iterating on the different subdomains of physically based computer animation (rigid bodies, deformable solids, liquids, smoke, *etc.*) to present their adaptive approaches, we have chosen to review the different families of adaptive methods, and present the way they have been applied in the different domains. We hope that this organization will give the reader a broader picture of adaptivity, by showing how variations of each approach have been developed in different domains.

In Section 2, we present time adaptive techniques such as dynamic time stepping and freezing techniques. Then, we focus on spatially adaptive techniques. Section 3 discusses the most popular approach of spatial adaptivity: geometric adaptivity(h-adaptivity where h classically refers to the size of elements in the finite element method), which refers to varying the discretization resolution via refinement and coarsening strategies. This section is subdivided according to the types of adaptive spatial discretization. Section 4 covers other spatial adaptivity approaches: basis refinement which adapts the number of bases, their order (p-adaptivity where p stands for polynomial), the basis functions themselves using enrichment, or, in subspace simulation, the deformation modes of the basis; moving grids methods (r-adaptivity where r stands for relocation) which relocate nodes without changing their connectivity; and mixed models which selectively apply a combination of different computational models. This organization reflects the taxonomy we propose in Figure 1. We finally conclude and sketch future research avenues in adaptive simulation.

2. Temporal adaptivity

Animating any simulated object requires integrating its equations of motion over time. There are many reasons why this integration procedure may need to adapt to the circumstances of the simulation, whether for accuracy, consistency, or stability. For instance, a system entering a highly nonlinear regime, such as during fracture, typically requires smaller time steps to maintain the desired degree of accuracy. Alternatively, adaptive time steps may be necessary to prevent the system from entering an invalid state: for example, many collision resolution schemes maintain the invariant that the system is never allowed to enter an interpenetrating configuration, which can require adjusting the time step according to the frequency of collisions. Finally, the stability of continuum-mechanical simulations is closely tied to the relationship between the time step length, the spatial resolution, and the speed of propagation of information in the system, so if either of the latter change (in the presence of *spatial* adaptivity or fast-moving flow) the time step must be adapted as well.

Techniques for temporal adaptivity fall into two main categories. One approach focuses on time resolution, that is, how to choose an appropriate step length for time integration. The second focuses on integration techniques themselves, seeking to switch between different integration schemes depending on the local context. This section explores both of these possibilities for temporal adaptivity.

2.1. Adaptive time step selection

Time step criteria First, let us focus on the simulation of continuous media, like fluids and elastic solids, whose models are governed largely by hyperbolic partial differential equations. Here the most prominent criterion for time step selection is the Courant-Friedrichs-Lewy (CFL) condition [CFL28]. To understand this condition, we first note that the solution of a partial differential equation at some point depends on a particular subset of initial or boundary data; we call this subset of data the *domain of dependence*. The CFL condition states simply that for any numerical scheme to converge to the true solution, the domain of dependence of

the numerical scheme must, in the limit, contain the true domain of dependence of the underlying differential equation. (Otherwise, one could perturb the initial data in the region outside the numerical domain of dependence and change the true solution without affecting the computed one.) The CFL condition can also serve as a stability criterion thanks to the Lax-Richtmeyer equivalence theorem [LR56, Str04], which states that a consistent finite difference method for a well-posed initial value problem is convergent if and only if it is stable.

For example, for the classical wave equation $\partial_t^2 f = c^2 \nabla^2 f$, the domain of dependence of any point $(\mathbf{x},t)$ includes only points $(\mathbf{x}_0,t_0)$ with $\|\mathbf{x}-\mathbf{x}_0\| \leq c(t-t_0)$, because information propagates only at the wave speed c. Consequently, the time step Δt must be small enough to prevent information from propagating outside the spatial stencil over a single time step. In particular, for the first-order explicit finite difference scheme applied to the wave equation, where over each time step a grid cell is only affected by adjacent grid cells (separated by Δx), the CFL condition requires that $c\Delta t \leq \Delta x$.

In general, the CFL condition typically takes the form

$$C \equiv \frac{c\Delta t}{\Delta x} \leq C_{\max}, \tag{1}$$

where c is the speed of information propagation, and $C_{\max}$ is a method-dependent constant which depends on the size of the finite-difference stencils. The dimensionless ratio C is known as the CFL number or the Courant number. Note that implicit methods are not restricted by the CFL condition because the solution at any point at the end of the time step depends on the values at *all* the points at the beginning of the time step, so the numerical domain of dependence is effectively infinite.

The CFL condition can be a useful heuristic even in situations where it does not directly apply. In computer graphics, semi-Lagrangian advection [Sta99] has been a popular scheme for solving the advection equation, as it is unconditionally stable and not restricted by the CFL condition [Bri08]. Nevertheless, excessively large time steps can lead to undesirable artifacts such as numerical dissipation and volume loss. Foster and Fedkiw [FF01] advocate limiting the time step size using (1) with c being the maximum of the flow speed $\|\mathbf{u}\|_\infty$ over the domain, and $C_{\max} = 5$.

In smoothed particle hydrodynamics (SPH), each particle is affected by other particles within its smoothing radius h, analogous to the grid separation Δx in finite-difference methods. Consequently, time step selection criteria based on the CFL condition can be applied. For pressure waves in compressible SPH, we continue to have

$$\Delta t \leq C_{\max}\frac{h}{c}, \tag{2}$$

and values of $C_{\max}$ between 0.25 and 0.4 have been used [Mon92, DC99]. Fast relative motion of particles, which occurs especially during fluid-fluid or fluid-solid collisions, can also produce artifacts due to interactions not considered in (2). Therefore, one can introduce additional time step constraints based on the instantaneous acceleration $\mathbf{a}$ and divergence of velocity $\nabla \cdot \mathbf{v}$ [Mon92, DC99]:

$$\Delta t \leq \Lambda\sqrt{\frac{h}{\|\mathbf{a}\|}}, \tag{3}$$

$$\Delta t \leq \frac{\Gamma}{|\nabla \cdot \mathbf{v}|}. \tag{4}$$

Λ and Γ are dimensionless constants which Desbrun et al. [DC99] set to 0.5 and 0.005 respectively. Equation (3) can in fact be interpreted as a CFL-type condition comparing the size of the spatial neighborhood, h, to a particle's relative displacement $\frac{1}{2}\|\mathbf{a}\|\Delta t^2$ due to acceleration over time Δt. Finally, we also point out that these criteria (2)–(4) may either be applied globally by taking the minimum of all particles' allowed time steps, or locally on a per-particle basis (as we discuss below).

When adapting the time step based on higher-order derivatives such as acceleration (3), care must be taken because such quantities can be much noisier than the system variables themselves, causing large fluctuations in Δt. Ihmsen et al. [IAGT10] have observed that in incompressible SPH simulations, these time step fluctuations can lead to spurious density shocks that destroy the convergence of the simulation. A solution is to change Δt gradually rather than instantaneously: if the system violates any of the desired time step criteria, Δt is decreased by a small amount ([IAGT10] use 0.2%), otherwise, if it is well within all the criteria, Δt is increased by the same amount. This strategy causes the time step to change smoothly towards the ideal step length. On the other hand, it may also prevent the time step from changing quickly enough to resolve sudden shocks, such as those from high-velocity impacts. Therefore, Ihmsen et al. introduce an additional procedure that detects shocks if the density error increases suddenly; if so, the simulation is rewound two time steps and resumed with a sufficiently small Δt.

Shock detection can be considered an example of an *a posteriori* time step adaptation strategy, where undesirably large time steps are detected and rewound. This kind of approach is useful whenever it is costly or difficult to estimate the right time step length in advance. Instead, we simply perform a time step with the current estimate of Δt, and then test whether it adequately resolves the motion of the system. If not, we reject the step, decrease the time step by setting $\Delta t \leftarrow \Delta t/\alpha$ for some factor $\alpha > 1$, and try again. When several time steps in a row are successful, we increase the time step, $\Delta t \leftarrow \alpha\Delta t$.

Bridson et al. [BFA02] use this approach for efficient collision handling in cloth simulation, using a combination of inelastic repulsion forces and a robust collision resolution algorithm. Inelastic repulsions are much more inexpensive

than full collision resolution, but as they only check proximity at discrete points in time, they can easily fail to prevent interpenetrations if the cloth moves too far in a single time step. If this happens, the time step is rejected and Δt is reduced; collision resolution is only triggered if after multiple failures the time step falls to a specified minimum size. The collision resolution step incorporates rigid impact zones [Pro97], which can be seen as a freezing technique and is discussed in Section 2.2. Bargteil et al. [BWHT07] simulate plastic flow using a finite element mesh, where excessive deformation of elements can be problematic. They reject a time step if any edge changes significantly in length, or a sudden acceleration takes place. Both methods discussed here use $\alpha = 2$, that is, time steps are halved or doubled as needed.

The main drawback of this *a posteriori* strategy is that the whole system must be globally rolled back to its previous safe state, even if it was caused by a localized event. In rigid bodies simulation, this challenge was adressed by Mirtich [Mir00] to efficiently handle collisions. Inspired by the work of Jefferson et al. [Jef85], he proposed a *time warp* algorithm to asynchronously handle collision events. Here, the integration of a rigid body is interupted only when resolving an event that concerns it. This technique can be seen as a local time stepping technique, and inspired works on *asynchronous variational integrators* (AVIs) which we discuss later in this section.

Global time stepping The simplest way to perform adaptive time stepping is to choose the time step that is safe for the entire simulation domain, and perform integration for the entire system using that time step. That is to say, given a time step criterion (or criteria) such as (2)–(4) that can be evaluated locally, one evaluates the permissible time step Δt_i at all simulation points i, and steps the entire system forward by a time step of length $\Delta t = \min_i \Delta t_i$. For methods that use implicit integration or other globally coupled schemes, such as grid-based fluids with a global pressure solve, this is typically the only possible approach. For this reason as well as for its conceptual and practical simplicity, global time stepping is probably the most widely used form of temporal adaptivity in practice.

It is worth pointing out here that adaptive time stepping is not a free lunch for all time integration schemes. The Verlet, or leapfrog, scheme is second-order accurate, and has excellent energy conservation properties thanks to its symplectic nature, but both these features rely on the time step being fixed. To maintain second-order accuracy with a variable time step, Bridson et al. [BMF03] proposed a time integration scheme that combines a leapfrog scheme for position with an implicit trapezoidal rule for velocity:

$$
\begin{aligned}
&1:\ \tilde{v}^{n+1/2} = v^n + \tfrac{\Delta t}{2} a(t^n, x^n, \tilde{v}^{n+1/2}) && \triangleright \text{ implicit}\\
&2:\ x^{n+1} = x^n + \Delta t \tilde{v}^{n+1/2} && \triangleright \text{ explicit}\\
&3:\ v^{n+1/2} = v^n + \tfrac{\Delta t}{2} a(t^n, x^n, v^n) && \triangleright \text{ explicit}\\
&4:\ v^{n+1} = v^{n+1/2} + \tfrac{\Delta t}{2} a(t^{n+1}, x^{n+1}, v^{n+1}) && \triangleright \text{ implicit}
\end{aligned}
$$

Maintaining symplecticity is much more challenging, as naively varying the time step can lead to instabilities and inconsistent energy behavior [HVS*09, Sec. 3]. Much more elaborate time stepping schemes are needed to recover energy preservation, such as the asynchronous variational integrators discussed below.

Local time stepping In complex simulation scenarios, different regions of the simulation domain may have very different time step requirements. For example, resolving challenging collision and contact scenarios requires careful time stepping, but only for the parts of the system that are affected by the contact. Similarly, in simulations with adaptive spatial resolution, the CFL condition requires finer-resolution regions to take smaller time steps. It can become intractable to simulate the whole model in lockstep using the most conservative time step. Instead, it is desirable to perform local time stepping, integrating each element of the simulation at its own pace.

Explicit integration schemes can readily incorporate local time stepping. We illustrate this with a generic two-dimensional system,

$$q_1'(t) = f_1\big(t, q_1(t), q_2(t)\big), \tag{5}$$

$$q_2'(t) = f_2\big(t, q_1(t), q_2(t)\big). \tag{6}$$

If at time t, q_2 requires a very small time step Δt_2, one can still integrate q_1 with its own time step Δt_1, giving

$$q_1(t + \Delta t_1) = q_1(t) + \Delta t_1 f_1\big(t, q_1(t), q_2(t)\big). \tag{7}$$

Meanwhile, as q_2 takes multiple steps to cover the same time interval, it will require values of q_1 at intermediate times $t + \Delta t_2$, $t + 2\Delta t_2$, and so on; these can be linearly interpolated from $q_1(t)$ and $q_1(t + \Delta t_1)$. Equivalently, to simplify bookkeeping, one can take the same small time steps Δt_2 for both q_1 and q_2, but only evaluate q_1' at the first step and hold it fixed until a time Δt_1 has been covered. This approach has the same computational advantage because evaluation of f is typically the most expensive part in explicit methods.

Early work on SPH [DC96, DC99] recommended this approach for local time stepping, using the CFL condition as the time step criterion. The same technique was also used to simulate elastic bodies using spatially adaptive finite element meshes [DDCB01], discussed in more detail in Section 3.2. For a linear elastic material with density ρ and Lamé coefficients λ and μ, the upper bound on the time step for an element can be approximated by

$$\Delta t \leq h \sqrt{\frac{\rho}{\lambda + 2\mu}} \tag{8}$$

where h is a measure of the size of the element, such as its inradius: skinnier elements or stiffer materials require smaller time steps. To improve the parallelizability of local time stepping for SPH fluids, Goswami and Batty [GB14] divide the computational domain into blocks and choose time steps independently per block.

Asynchronous variational integrators (AVIs) studied by Lew et al. [LMOW04] are a family of time integration schemes that provide excellent energy conservation behavior while allowing different elements of the system to use different time steps. Unlike the local time stepping model described above, AVIs associate a time step with each *force* rather than each variable. Thus each force term applies a series of impulses to its associated nodes. The time step for a particular term must remain constant throughout the simulation, but different forces may have very different time steps. This approach is typically implemented as an event-driven simulation loop, using a priority queue to schedule the updates for all the forces in order. Thomaszewski et al. [TPS08] applied this approach to cloth simulation with a finite element triangle mesh, choosing time steps independently for each element using the CFL criterion (8). AVIs are known to exhibit "resonance instabilities" that can potentially cause the energy to increase without bound; however, these instabilities tend to be extremely weak in solid mechanics problems [FDL07] and have not been observed to cause difficulties in computer graphics [HVS*09].

The energy conservation properties of AVIs hinge on the regular spacing of the impulses applied by each force term, which makes collisions challenging to incorporate: naively applying contact forces at the moment of collision breaks the periodicity and destroys energy conservation, while applying contact forces at regular intervals risks missing collisions. Recent work [HVS*09, AVGT12] addresses this problem by replacing each contact force with a sum of stiffer and stiffer penalty layers with smaller and smaller time steps, which together are guaranteed to prevent interpenetration. While the number of penalty layers is conceptually infinite, any given collision can only activate a finite number of penalty layers, so the actual amount of computation is finite. Initial work by Harmon et al. [HVS*09] used kinetic data structures to detect all collision events in advance. Ainsley et al. [AVGT12] instead adopt a speculative approach based on the time warp algorithm [Jef85, Mir00], analogous to the *a posteriori* time step adaptation techniques discussed previously. An interval of time is first simulated without considering any new collisions; then, collision detection over the simulated interval is performed, and if new collisions are found, those force terms are activated and the interval is resimulated. Speculative simulation greatly reduces the computation time spent in bookkeeping and collision detection, and also allows for easy parallelization.

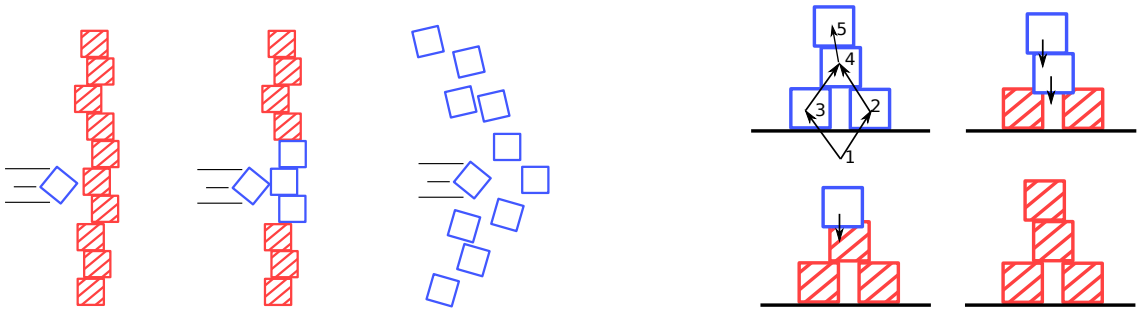
2.2. Adaptive integration

Adaptive choice of time integration scheme In some cases, such as when the system involves highly stiff modes or poorly conditioned elements, an adaptive choice of time step is no longer the most efficient strategy. The time step requirements for explicit integration may become extremely restrictive. Instead, one may locally change the integration scheme to deal with the problematic components, for example by switching to an implicit method or a nonlinear one. By doing so adaptively, one can continue to use an inexpensive explicit integration scheme for the remainder of the system.

In the finite element method, ill-shaped elements impose severe restrictions on the allowed time step for explicit integration (8). When the object undergoes topological changes such as cutting or fracture, it can be difficult to avoid introducing such elements. As an alternative to local time stepping, where ill-shaped elements would have to be simulated with extremely small time steps, Fierz et al. [FSH11] propose an element-wise implicit-explicit (IMEX) scheme. Here the same time step Δt is used for the entire system, but ill-shaped elements that would be unstable if explicitly integrated over Δt are instead simulated with an unconditionally stable implicit scheme. Nodes that are not adjacent to any ill-shaped element are integrated explicitly, then held fixed as boundary conditions for implicit integration of the remaining nodes. As long as the number of ill-shaped elements is low, this relaxes the time step restriction faced by explicit integration, while minimizing the computational cost and numerical dissipation associated with implicit integration.

Thin materials such as hair exhibit a high stiffness in their stretching modes, but pose the additional challenge that their collision response is highly nonlinear due to the presence of rotation. Therefore, depending on the amount of bending, an implicit first-order model for collisions may fail to capture the correct response and lead to instabilities. Kaufman et al. [KTS*14] propose an algorithm that adaptively chooses the *degree* of nonlinearity in each contact resolution step to safely resolve the collision. Specifically, they adapt the number of constrained Newton iterations used to solve the nonlinear contact model, terminating when the stretch over all affected edges is sufficiently reduced. This allows for large simulation time steps in the face of many energetic collisions, while efficiency is maintained because most collisions require only a single iteration (equivalent to a linearly implicit step).

Freezing techniques Freezing techniques, also called sleeping techniques, lie in between temporal and spatial adaptability: the degrees of freedom that are considered unimportant in the simulation are kept constant in time for a specified duration. This can be seen as animating them at a much larger time scale, or as temporarily deactivating them. No memory is saved while doing so, but computation time is reduced. These techniques are useful in situations where the spatial domain is large and filled with many quiescent objects. Therefore, game environments and surgical simulations are perfect candidates for these methods. Conversely, freezing techniques may not be useful in highly dynamic situations where most of the degrees of freedom are active most of the time. In addition to defining good freezing criteria, the main challenge of freezing techniques is to design a reacti-



(a) (b)

Figure 2: Freezing techniques applied to rigid bodies stacking. At the stage of contact solving, those methods can save computational time while ensuring a plausible motion. In (a), quiescent rigid bodies are frozen (in hatched red) in a stacking configuration. A collision event then awakes one of the rigid bodies, and the active rigid body (in blue) propagates the information to its neighbors. In (b), a contact graph stores stack ordering. During shock propagation, a bottom-up traversal is performed and objects at each level are frozen by assigning an infinite mass.

vation process of the frozen degrees of freedom that ensures plausible subsequent motion.

In rigid objects simulation (see the survey of Bender et al. [BETC12]), important computational resources are dedicated to solving contacts between the different objects of a scene. This process becomes unnecessarily expensive when stacking occurs and nothing moves. In these cases, freezing techniques prove to be useful for saving computational time without compromising the plausibility of the simulation.

Schmidl [Sch02] uses a heuristic based on kinetic energy to determine whether to freeze a body or not:

$$\frac{1}{2}m\mathbf{v}^2 + \frac{1}{2}\boldsymbol{\omega}^T\mathbf{I}\boldsymbol{\omega} < \frac{\mathbf{p}_g^2}{2m} \tag{9}$$

where $\mathbf{v}$ and $\boldsymbol{\omega}$ respectively denote the linear and angular velocity of the rigid body of mass m and inertia tensor $\mathbf{I}$, and $\mathbf{p}_g = m\mathbf{g}\Delta t$ is the momentum that the body accumulates during a time step Δt from gravity. If condition (9) is fullfilled by the body during a user-defined number of consecutive time steps, then it is frozen. Note that a single frozen body by itself does not save computation time. Time is saved when multiple neighboring objects become frozen, as their contact forces no longer need to be computed. The only way for a frozen rigid body to be reactivated is when it receives a large impulse during a collision. Once it is reactivated, the information is propagated to all its direct neighbors and a given number of indirect neighbors, potentially awakening them (see Figure 2a).

Freezing techniques have been used as a failsafe procedure for resolving collisions and contact between rigid or deformable bodies. When many interacting contacts are present, iterative processes for collision response can require an excessively large number of iterations to converge, and early termination leaves the system in an interpenetrating state. Freezing procedures are attractive in this context as they can guarantee elimination of interpenetrations. However, they also introduce loss of kinetic energy by treating contacts as fully inelastic, and are therefore only invoked as a last resort after multiple iterations of a physically correct solver have failed to resolve the collisions. For rigid body contact, Guendelman et al. [GBF03] propose a shock propagation strategy which freezes bodies progressively from the ground up. They start by building a directed acyclic graph consisting of "levels" of objects that are resting on objects of lower levels (cyclic dependencies are grouped into the same level). A single bottom-up traversal of the graph resolves contacts level by level, assigning infinite mass to lower-level objects for which contact is solved (see Figure 2b). Rigid impact zones, described by Provot [Pro97] and Bridson et al. [BFA02], are extensively used to resolve collisions in cloth. Nodes involved in multiple interfering collisions are collected into disjoint sets called "impact zones", constructed by merging zones if their nodes are involved in the same node-face or edge-edge collision. Each impact zone is rigidified by replacing the motion of its nodes with a rigid body motion while preserving the total linear and angular momentum. This process eliminates collisions within the zone, but may introduce new collisions with nearby elements outside the zone. Therefore, one must perform collision detection again and grow the impact zones if new collisions are detected, iterating this process until no new collisions occur.

Freezing techniques were also used for articulated rigid bodies, both for dynamics [RGL05] and quasi-statics [RL06]: joints are activated or deactivated based on user-defined error metrics, leading to a simplification of the whole model and a sub-linear computational complexity.

Denoting by $\ddot{\mathbf{q}}^C = (\ddot{\mathbf{q}}_1, \ldots, \ddot{\mathbf{q}}_{N_C})^T$ the composite acceleration of an articulated body C, where N_C is the number of joints in C, the *acceleration metric value* of C is defined as

$$\mathcal{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i,$$

where $\mathbf{A}_i$, $i \in C$, are symmetric, positive definite $d_{J_i} \times d_{J_i}$

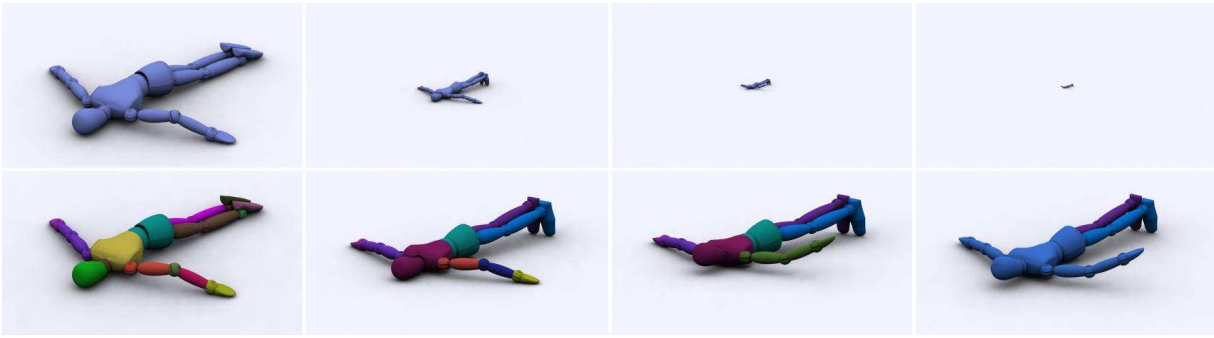


Figure 3: View-dependent dynamics of articulated bodies. Top: the algorithm proposed by Kim et al. [KRK08a] automatically simplifies the dynamics of a falling character as its distance to the viewer increases. Bottom: corresponding rigidification at this time step (one color per rigid group).

weight matrices, and d_{J_i} is the number of degrees of freedom of joint i in C. The weight matrices $\mathbf{A}_i$ are required to depend at most on joint positions, and the simplest choice is the identity matrix. The key to the sub-linear complexity is the demonstration that the acceleration metric of each sub-articulated body is a quadratic function of the forces applied to its handles (i.e. the free rigid bodies used to assemble sub-articulated bodies in Featherstone's methodology [RGL05]). As a result, the acceleration metric may be computed *before computing the joint accelerations themselves*. This is used during the top-down pass of Featherstone's divide-and-conquer algorithm to determine which joint accelerations should be computed because they are sufficiently large [RGL05]. For example, if the acceleration metric of the complete articulated body is small, then all joint accelerations are small and joint velocities are constant for the current time step. A similar metric is built for joint velocities to determine which joint positions should remain constant, and thus which inter-body forces and inertial terms should be updated. Once all position-dependent terms are up to date, the motion metrics are available for the next time step, and a new set of active joints can be determined.

It was shown that this approach could also help to speed up continuous collision detection for articulated bodies [KRK08a] and haptics rendering [MR07], as well as enable view-dependent articulated-body dynamics by combining motion metrics with visibility estimates [KRK08b] (see Figure 3). Gayle et al. [GLM06] demonstrate how to perform contact handling with such an adaptive articulated-body method, which is used as the core of a physics-based sampling algorithm for highly articulated chains [GRS*07] and cable route planning [KGL07].

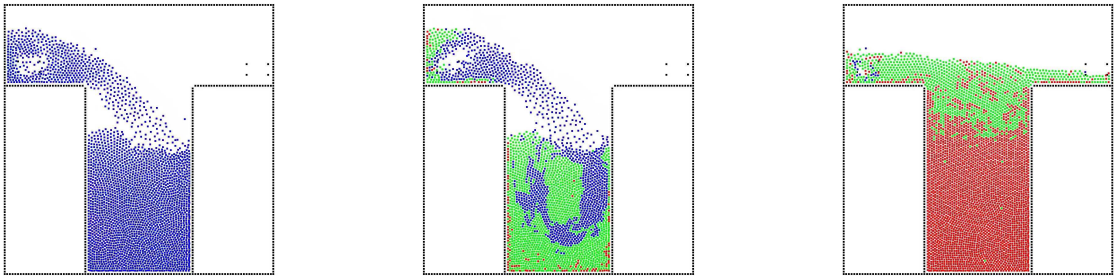
Recent work has sought to apply freezing techniques to accelerate SPH fluid simulation. In these methods, particles are divided into two sets: a set of *active* particles following a classical simulation step, and a set of *inactive* particles that are skipped in the simulation. As physical quantities in SPH are interpolated from neighbors, inactive neighbors of active particles also need to be updated to ensure that active particles compute correct physical quantities. Computation time is saved for inactive particles that only have inactive neighbors. The main challenge in these methods is the definition of the process to transform a particle from the inactive set to the active one and vice versa. Indeed, as SPH is sensitive to particle distribution, an inadequate transition of state can directly lead to instabilities. Additionally, a judicious choice of criterion to decide whether a particle should be active or not is essential.

Goswami and Pajarola [GP11] propose a simple method that evaluates the active status of particles at each time step. Particles are marked as active if they are close to the boundary or if their velocity exceeds a threshold. Inactive neighbors of such particles are also added to the active set, and continue with their last active velocity. All other particles are considered inactive. Unfortunately, the resulting method does not obey Newton's third law, resulting in some loss of momentum.

In the context of molecular simulation, Artemova and Redon [AR12] propose a fundamentally different approach which ensures momentum conservation. This approach, called *Adaptively Restrained Particle Simulation (ARPS)*, builds a modified equation of motion which adapts itself depending on the state of the particles. It continuously interpolates the behavior of a particle switching from one state to another. Here is the modified equation of motion for a particle with $\mathbf{p}$ as momentum, $\mathbf{q}$ as position, $V(\mathbf{q})$ as potential energy, m as mass and ρ a *restraining function*:

$$\begin{aligned} \dot{\mathbf{p}} &= -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}} \\ \dot{\mathbf{q}} &= m^{-1}[1-\rho(\mathbf{p})]\mathbf{p} - \frac{1}{2}\mathbf{p}^T m^{-1}\frac{\partial \rho(\mathbf{p})}{\partial \mathbf{p}}\mathbf{p} \end{aligned} \tag{10}$$

The *restraining function* $\rho(\mathbf{p}) \in [0,1]$ is a twice differentiable function that can be customized to incorporate different freezing criteria such as kinetic energy. We can observe that if $\rho = 0$ the dynamics of the particle will follow the clas-



(a) (b) (c)

Figure 4: (a) shows a classic SPH simulation of a 2D scenario with inflow and outflow boundary conditions at the walls above a chasm. (b) shows at the same time the effect of the ARPS method from Manteaux et al. [MFRC13]. Blue particles are active, red particles are inactive, and green particles are transitive. Large regions quickly become inactive (c), allowing to save significant computational time as numerous particles' neighborhoods do not need to be updated anymore.

sical equation of motion:

$$\begin{aligned} \dot{\mathbf{p}} &= -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}} \\ \dot{\mathbf{q}} &= m^{-1}\mathbf{p} \end{aligned} \tag{11}$$

Whereas if $\rho = 1$ then the particle will still accumulate momentum but will remain still:

$$\begin{aligned} \dot{\mathbf{p}} &= -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}} \\ \dot{\mathbf{q}} &= 0 \end{aligned} \tag{12}$$

In this case, even if particles' momentum are still correctly computed, forces do not need to be recomputed as positions are held fixed. Therefore substantial computational time can be saved. Moreover, by using an incremental algorithm to update interaction forces, inactive particles with active neighbors can be efficiently updated while staying aware of their neighborhood. Finally, if $0 < \rho < 1$ then the particle follows the interpolated dynamics of equation (10). The particle is then called *transitive*. Manteaux et al. [MFRC13] extend their work to SPH (see Figure 4) and to deformable model simulation, in order to better match computer graphics needs.

3. Geometric adaptivity

Geometric adaptivity describes various techniques that adapt the spatial resolution of a model by refining and coarsening its discretization. These techniques are also refered in the literature as *adaptive spatial refinement*. However, as they include coarsening as well, we adopted a more general term.

Geometric adaptive techniques have two major components: a refinement criterion that determines where higher resolution is needed, and a refinement/coarsening scheme that modifies the discretization to match the desired resolution. Both physical and visual criteria have been employed in existing work, and we discuss them in more detail below. The refinement scheme itself essentially depends on the type of spatial discretization. In the following subsections, we will deal with the three major kinds of discretization separately: structured meshes and grids, unstructured meshes, and meshless models.

Refinement criteria The choice of refinement criteria plays a major role in the quality of the resulting simulation. Many techniques use simple heuristics such as the distance to boundaries, surface curvature, and the presence of contacts. However, some authors have shown that employing criteria that are more closely tied to the dynamics of the system can be important in many contexts.

In elastic and plastic solids, the stress and strain in an element characterize the amount of local deformation. Therefore, the values and gradients of these quantities are often used to control refinement. Wu et al. [WDGT01] describe several different error estimators of this type and discuss their relative advantages, drawbacks, and performance. Going beyond estimating a scalar error, Wicke et al. [WRK*10] define a metric tensor $\mathbf{M}$ that approximates the spatial variation of strain by comparing deformation gradients of adjacent elements. The matrix $\mathbf{M}$ is defined in such a way that it has large eigenvalues in directions in which the deformation changes most rapidly, and can thus be used to control anisotropic refinement (see Figure 5).

If a multigrid algorithm is used, the difference between the solution at the current level, $\mathbf{x}^j$, and the one prolonged from the next coarser level, $\mathbf{P}^j\mathbf{x}^{j+1}$, provides a natural measure of the quality of $\mathbf{x}^{j+1}$. Otaduy et al. [OGRG07] weigh the error by the local system matrix $\mathbf{A}$ to reduce possible popping in stiff scenarios, leading to the error metric

$$e = \|A^j(\mathbf{x}^j - \mathbf{P}^j\mathbf{x}^{j+1})\|, \tag{13}$$

and perform refinement if e exceeds a predefined threshold.

Lower-dimensional bodies, like wires, strands, cloth, and thin sheets, undergo not just stretching but also bending (and torsion, in the case of one-dimensional strands). While the

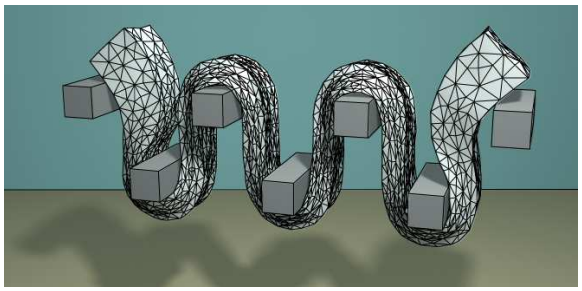


Figure 5: Elastoplastic simulation with dynamic local remeshing [WRK*10]. By using the strain gradient as the refinement criterion, regions undergoing severe deformation are refined locally.

stretching forces within the material are much stiffer than the bending forces, the bending deformation can often be more visually important, and also introduces geometric nonlinearities that must be carefully resolved. In this context, geometrical curvature and the presence of contacts (which are likely to induce bending) are the most commonly used refinement criteria. However, the interaction between stretching and bending leads to additional considerations.

First, in the context of cloth simulation, Simnett et al. [SLD09] and Narain et al. [NSO12] point out that elements under compression are likely to buckle and should therefore also be refined, otherwise the wrinkles that would arise in subsequent time steps will fail to be represented on the coarse mesh. By considering the trade-off between bending and stretching energy, Narain et al. estimate that a sheet under compressive strain ε is likely to form wrinkles of width proportional to $\sqrt{k_b/(k_s\varepsilon)}$ where k_b and k_s are the bending and stretching stiffnesses respectively, providing an estimate of the extent of refinement necessary. In the physics literature, Cerda and Mahadevan [CM03] have provided a detailed semi-analytical analysis of wrinkle geometry that is valid far from the small-deformation limit, and may be useful for future work in graphics.

Second, finer meshes allow higher-frequency modes of transverse oscillation, leading to time step constraints and stability problems that can be fatal for interactive applications. To ensure stability, Servin et al. [SLN08] propose a *coarsening* criterion, reducing the resolution of the mesh so that only those oscillations whose frequency is lower than the time stepping rate can be represented. For simulation of systems with stiff wires, they estimate the maximum frequency of oscillations in a wire discretized with n nodes as

$$\omega_{\max} \approx 2(n+1)\sqrt{\frac{f}{mL}}, \tag{14}$$

where f is the tension in the wire and m and L are its mass and length. Requiring this frequency to be lower than $1/\Delta t$ gives an upper bound on n for each wire.

In fracture simulation, the stress forms a singularity at the crack tip, which must be resolved accurately with a fine resolution mesh for realistic crack paths to be obtained. If the crack origin is known *a priori*, one can track the crack path as the fracture proceeds and refine the mesh based on the distance from the crack [BDW13]. However, if fracture is allowed to originate anywhere, it is necessary to refine the mesh wherever stress is sufficiently high, i.e. close to the material's strength τ, because such regions may generate new cracks. Koschier et al. [KLB14] refine elements whose tensile stress σ exceeds a specified fraction of τ. Pfaff et al. [PNdJO14] choose the desired resolution of the mesh to be proportional to tensile stress by requiring the length of each edge $\mathbf{e}$ to satisfy

$$\|\mathbf{e}\| \leq \max\left(\frac{\tau}{2\sigma}, 1\right)\ell_{\min} \tag{15}$$

where $\ell_{\min}$ is a user-specified refinement limit. This approach allows the mesh to be coarsened again after the crack tip has passed.

It is also possible to perform view-dependent adaptivity by modulating the refinement criteria based on visibility and distance from the camera. Such techniques have been applied to the simulation of wires [SLN08] and cloth [KNO14]. New issues arise in such contexts, such as preventing artifacts when coarser regions come into the field of view and must be refined: Koh et al. [KNO14] achieve this by smoothly increasing the resolution in advance based on the known camera path.

Many adaptive methods for simulation of liquids have relied on the distance to the free surface as the sole refinement criterion. However, much greater adaptivity is possible by observing that in regions where the surface is flat and does not exhibit detailed motion, it can also be coarsened without significantly affecting the flow. Adams et al. [APKG07] propose a purely geometrical criterion based on an "extended local feature size" which measures the distance to the surface and to the medial axis of the fluid volume. This criterion assigns coarse resolution both far from the surface and near thick, flat surfaces. However, it does not take the motion of the fluid into account. To detect regions with significant flow detail, Hong et al. [HHK08] use a "deformation factor" that locally estimates the Reynolds number,

$$\mathrm{Df} = \frac{(\mathbf{u}\cdot\nabla)\mathbf{u}}{\nu\nabla^2\mathbf{u}}, \tag{16}$$

where $\mathbf{u}$ is the fluid's velocity field and ν is its viscosity. Ando et al. [ATW13] use a flexible sizing function that combine multiple criteria to set the desired resolution. Among them the depth of the liquid, the camera viewpoint, the fluid surface curvature and the norm of the strain rate, $e = \|\nabla\mathbf{u}\|_F$, in order to preserve detailed motions.

3.1. Structured meshes and grids

Spatially adaptive simulation techniques can often benefit from symmetry or structure, at the expense of flexibility. Techniques using hexahedral elements or finite differences can use quadtrees or octrees to add spatial adaptivity. Alternatively, techniques that use a volumetric tetrahedral mesh, like some finite element methods or mass-spring systems, can easily take advantage of structured meshes based on lattices. Special techniques can also combine grids with "tall cells" or far-field grid structures, as discussed at the end of this section.

The techniques described in this section are useful when one wishes to speed up simulations by using local spatial adaptivity without completely committing to an unstructured mesh technique. Structured meshes often allow more code to be re-used when converting from a regular grid to an adaptive one, and they are often more cache-coherent than fully unstructured meshes. However, structured meshes do not have as much flexibility as unstructured ones.

Quadtrees and Octrees The spatially adaptive schemes of Hutchinson et al. [HPH96] and Villard and Borouchaki [VB05] use quadtrees to directly connect masses and springs, introducing T-junctions in the process. Ganovelli et al. [GCS99] propose an octree-based multi-resolution method for determining connectivity in a mass-spring system. Debunne et al. [DDBC99] simulate elastic models using control nodes based on approximate finite differences operators, and they use an octree-based refinement to increase detail based on a Laplacian deformation metric. As discussed in several of the aforementioned works, the main difficulty with using mass-spring models instead of approaches based on continuum mechanics is the notion of *convergence*. Convergence is well-studied in the finite element method, and it is trivial to show that increasing spatial resolution will lead to a more accurate solution. Mass-spring models can also converge under refinement if spring stiffness parameters are chosen appropriately, but these stiffness values are not as straightforward to derive compared to the stiffness matrix in a finite element method.

The works of Dick et al. [DGW11] and Seiler et al. [SSSH11] use an octree to define a set of hexahedral finite elements in an elasticity simulation, which is specifically used for simulating cuts in a deformable body. Dick et al. also leverage the hierarchy provided by the octree in a geometric multigrid method for solving the elastodynamics. The work was subsequently extended to interactively animate high resolution boundary surfaces [WDW11], to improve collision handling [WDW13], and to simulate at haptic rates [WWD14b]. For more details on methods for cutting deformable bodies, please see the state of the art report by Wu et al. [WWD14a].

Researchers have also developed octree-based discretizations of the Navier-Stokes equations, which lead to efficient animations of smoke and liquid [SY04, LGF04, LFO05]. These approaches also inspired spatially adaptive works that handle discontinuities across free surfaces [HK05], resolve extremely thin surfaces in bubbles and foam without losing volume [KLL*07], and animate multi-phase fluids using regional level-sets [Kim10]. More recent work [FWD14] combines an adaptive hexahedral finite element method based on octrees with a multigrid Poisson solver to animate highly detailed liquids. Bargteil et al. [BGOS06] use octree-based spatial adaptivity to track detailed deforming liquid surfaces using semi-Lagrangian contouring.

It is worth noting that quadtree and octree grid refinement can have subtle side-effects when discretizing partial differential equations. In particular, the regular staggered-grid discretization of the Poisson equation (which is used for enforcing incompressibility in fluid flows [Bri08]) happens to satisfy Stokes' theorem and exactly integrates fluid fluxes — it doubles as a "finite volume method" and can be alternatively derived using discrete exterior calculus [CDGDS13]. When one replaces the regular grid with an octree, however, it is unsafe to assume that such useful properties will still hold in the presence of T-junctions. The previously-mentioned octree-based fluid simulation methods counteracted this particular problem by carefully designing a new divergence operator, and some researchers observed the emergence of spurious rotational flows when refining an octree near liquid surfaces.

These subtle problems help explain the large number of adaptive BCC-mesh-based liquid solvers discussed below, which do not exhibit T-junctions.

Adaptive BCC lattices A standard way to efficiently generate a tetrahedral mesh with spatial adaptivity is to combine a spatial hierarchy with predefined lattice-based stencils. One particularly popular strategy combines an octree with the body-centered cubic (BCC) lattice tetrahedralization. To give some context, a *regular* BCC lattice is defined by first inserting vertices at the corner of a regular cubic grid, inserting additional vertices at the center of every grid cell, and then creating a Delaunay tetrahedralization of these vertices. A *spatially adaptive* tetrahedral mesh is created similarly by first creating a weakly-balanced octree (instead of a regular grid), inserting vertices at the corners and centers of the octree cells, and then tetrahedralizing the set of vertices. Please see the work of Molino et al. [MBTF03] and Labelle and Shewchuk [LS07] for some examples of how to create such an octree-based adaptive BCC mesh. This particular meshing strategy has several benefits: the average tetrahedral element has a nearly optimal shape, the quality of the worst element is bounded and completely acceptable in practice, and the computation time required to build the mesh is orders of magnitude faster than unstructured meshes, because it makes use of trees and precomputed stencils.

Employing these ideas, Wojtan and Turk [WT08] use an adaptive non-conforming BCC tetrahedralization to simulate

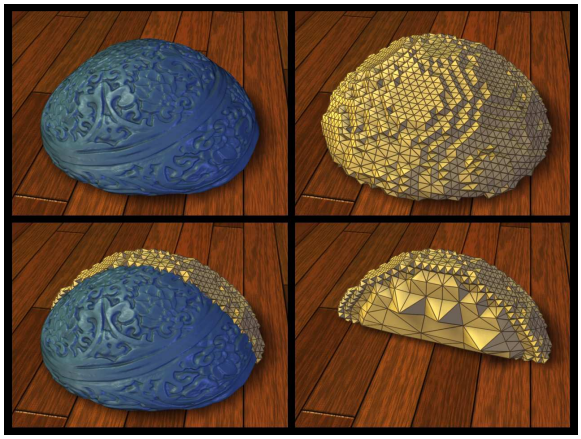


Figure 6: A high resolution surface mesh (blue) embedded into a deforming low resolution spatially-adaptive tetrahedral mesh based on a BCC lattice (gold). The bottom row shows the cross section of the tetrahedral mesh, illustrating the BCC structure. Image from Wojtan and Turk [WT08].

highly plastic materials while efficiently remeshing whenever element quality degrades (See Figure 6). Batty et al. [BXH10] use a similar meshing strategy to simulate inviscid fluids using a finite-volume discretization. The method also handles embedded solid and free-surface boundary conditions, even though the tetrahedral mesh does not conform to the domain boundaries. Batty and Houston [BH11] extend this work by adding an implicit viscosity model. As explained later in the Section 3.3, Ando et al. [ATW13] also use an adaptive BCC mesh for simulating liquids. Sifakis et al. [SSIF07] introduce the concept of "hard bindings" to create an adaptive BCC lattice with T-junctions for the purposes of elastic solid simulation.

In the authors' experience, an adaptively-refined BCC lattice is exceptionally useful in simulations requiring tetrahedral meshes, because the structured lattice makes the typically expensive operations of remeshing and resampling simulation data relatively insignificant. Although the structure of the mesh removes some control over the shapes and nature of the spatial adaptivity, the structure also eliminates typical issues, such as degenerate tetrahedra.

Tall cell grids Simulations of deep water also benefit from adaptive "tall cell" techniques [IGLF06, CM11], which use regular grid cells near the water surface (where detail and accuracy is important), and tall rectangular fluid cells farther below the surface. This strategy effectively assumes that the behavior in regions located deep underwater is simpler, in that the simulation variables cannot change arbitrarily with depth. While we specifically discuss tall cell grids here because of their use of geometric adaptivity, we will also address some related height-field methods in Section 4.3. These methods seem to work very well when the assumptions of the methods hold. However, there is reason to believe that artifacts (like spurious reflecting waves or dissipating vortices) can occur when the tall cells fail to properly resolve important dynamics.

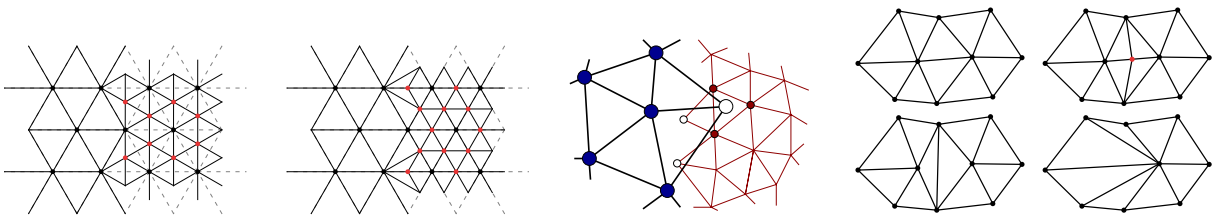
Far-field grid structures Zhu et al. [ZLC*13] propose a method for fluid simulation that maintains an efficient Cartesian grid structure but allows non-uniform spacing between the nodes. This approach makes it easy to concentrate smaller cells where the domain is more interesting and retain larger cells farther away from areas of interest. In addition to concentrating detail in important regions, this method also approximates non-reflecting boundary conditions by greatly extending the boundaries of the simulation domain.

3.2. Unstructured meshes

Adaptivity on unstructured meshes is closely related to the problem of remeshing. Considered purely as a geometrical problem, remeshing has been studied extensively in computational geometry [CDS12] and in a modeling context in computer graphics [AUGA08]. Indeed, the techniques adopted for adaptive simulation often build on the framework of geometrical remeshing methods, and extend them to a simulation context. A number of challenges arise when performing remeshing during simulation, though. First, the mesh elements must remain well-conditioned to avoid degrading the stability of the simulation. Second, modifying the discretization on the fly risks introducing errors in transferring energy and momentum to the new mesh, such as numerical diffusion due to resampling, and discontinuous "popping" artifacts in thin strands or sheets. Third, removing degrees of freedom must be done with care, as a mesh with fewer DOFs cannot represent the previous system state exactly. Depending on the characteristics of the simulated material—whether it is elastic, plastic, or fluid; whether volumetric or lower-dimensional—different remeshing methods are found to be appropriate.

Hierarchical schemes The simplest remeshing strategy is to use a fixed hierarchical scheme for refinement, which provides guaranteed bounds on element quality. Two such schemes are illustrated in Figure 7a, 7b. In cloth simulation, triangle subdivision schemes such as 1-to-4 splits, $\sqrt{3}$ refinement, and edge bisection have been employed [LV05, SLD09, BD12]. A similar scheme for subdivision of tetrahedra was used by Koschier et al. [KLB14] for volumetric fracture. Wu et al. [WDGT01] build on the concept of progressive meshes [Hop96] to precompute FEM parameters, allowing adaptive simulation of deformable bodies at interactive rates.

However, subdivision schemes still degrade element quality by a moderate extent compared to an optimized mesh. When this is undesirable, an alternative is to use a hierarchy of "non-nested meshes" at different resolutions [DCDB00, DDCB01]; see Figure 7c. Each level of the hierarchy is a



(a) (b) (c) (d)

Figure 7: Overview of refinement schemes for unstructured meshes. In (a) and (b), the right half of a mesh is refined using (a) $\sqrt{3}$ refinement and (b) hierarchical edge bisection, with inserted nodes highlighted in red. In (c), we illustrate two levels of non-nested meshes. In (d), the three primitive operations of local remeshing are applied to the middle edge: in reading order, we show the original mesh, after an edge split, after an edge flip, and after one of two possible edge collapses.

complete mesh that does not necessarily share any nodes with meshes at other levels, and can be independently optimized *a priori*. At run time, regions at different levels of detail use subsets of different meshes. Coupling is achieved by allowing the regions to overlap slightly; nodes in the overlap region in each mesh are treated as inactive "ghost" nodes that are embedded in the containing element of the other mesh. The work of Otaduy et al. [OGRG07] seamlessly integrates such adaptive non-nested meshes with a multigrid algorithm and an adaptivity-aware collision detection technique.

Hair simulation can benefit from adaptivity, as the contact interactions between hair strands lead to the formation of emergent clusters. Bertails et al. [BKCN03] introduce a hierarchical structure called an adaptive wisp tree (AWT), which represents hair clusters that can progressively split into smaller clusters from the base to the tip. Refinement is performed by splitting a node if its size and acceleration are large, while coarsening is performed by merging sibling nodes if they have similar positions and velocities.

Nearly regular meshes Some techniques for liquid simulation use a structured mesh in the interior of the volume, but allow irregularities at boundaries to better capture the dynamics of the free surface. These techniques benefit from many of the advantages of structured meshes discussed in Section 3.1, while still retaining much of the flexibility of unstructured meshes, such as the ability to accurately match boundary conditions.

In such methods, one typically maintains a high-resolution representation of the surface as a triangle mesh that is updated at each time step. The surface is superimposed on a regular mesh structure such as an octree or a uniform grid, and elements near the surface are modified, or new elements inserted, to better conform to the surface. In particular, Chentanez et al. [CFL*07] use the isosurface stuffing algorithm [LS07] that generates an adaptive BCC lattice whose surface tetrahedra are warped and possibly subdivided to conform to the surface geometry. Using an octree to construct the lattice allows for coarser resolution away from the free surface. Brochu et al. [BBB10] use a uniform background lattice and introduce additional pressure samples along both sides of the free surface to ensure that all surface features are resolved. The pressure projection is then performed on a mesh consisting of the Voronoi cells of these sample points.

Local remeshing In some contexts, it is desirable to allow the connectivity structure of the mesh to be modified freely during the course of the simulation. Such a requirement arises when anisotropic elements are needed to resolve strongly directional features, or when the material exhibits both elastic properties and unbounded deformation, such as in plastic flow.

While it is possible to perform global remeshing—that is, simply creating a new simulation mesh from scratch whenever needed [KFCO06, BWHT07]—this approach can lead to undesirable diffusion of stored physical quantities such as plasticity information. An increasingly popular alternative is to remesh locally using a set of local operations that refine, coarsen, and reshape existing elements. In local remeshing techniques, any elements in the mesh that do not satisfy the desired size and shape criteria are improved by careful application of these operations. This process is repeated until all mesh elements are satisfactory.

When performing remeshing, it is important to ensure that the mesh remains well-conditioned for simulation, through the use of various element quality measures [She02]. In 2D, the Delaunay triangulation optimizes several important notions of mesh quality, including the minimum angle and the maximum circumradius of any triangle, but the Delaunay tetrahedralization in 3D provides few such guarantees, and more sophisticated mesh improvement strategies may be required [WRK*10]. If anisotropic remeshing is desired, the remeshing criterion is often expressed through a spatially varying metric tensor $\mathbf{M}$, with the goal being that each edge $\mathbf{e}$ should have $\mathbf{e}^T\mathbf{M}\mathbf{e} \approx 1$. Equivalently, we desire $\|\mathbf{M}^{1/2}\mathbf{e}\| \approx 1$; that is, we want edges to be of unit length and elements to be equilateral in the transformed space of $\mathbf{M}^{1/2}$. This viewpoint allows element quality metrics and Delaunay properties defined for isotropic meshes to be carried over to the anisotropic setting.

Figure 8: The anisotropic remeshing algorithm of Narain et al. [NSO12] allows detailed wrinkles in cloth to be resolved accurately with fine elements (red), while much coarser elements (blue/green) are used in flat regions. Long, narrow folds are best represented using anisotropic elements (yellow) aligned with the curvature direction.

For manifold triangle meshes, high-quality remeshing can be accomplished using only the three simple operations of edge split, edge collapse, and edge flip (see Figure 7d). Narain et al. [NSO12] use these operations in cloth simulation, generating an adaptive anisotropic mesh that resolves detailed wrinkles and folds (see Figure 8). First, all edges that are unacceptably long according to the refinement criterion are split, then edge collapses are attempted as long as they do not create new unacceptable edges. During both steps, edge flips are performed to maintain an approximately Delaunay mesh relative to the anisotropic metric.

Local remeshing of tetrahedral meshes is significantly more involved, requiring several different local operations and a complex schedule for the order in which to apply them [KS07]. This technique was first applied to simulation by Wicke et al. [WRK*10], who used it to minimize artificial diffusion in elastoplastic flow. Subsequent work has applied such remeshing techniques to simulation of incompressible liquids [MB12, MEB*12, CWSO13].

Misztal et al. [MB12, MEB*12] build a mesh over the entire simulation domain, with some elements belonging to the fluid and the rest to the exterior, while Clausen et al. [CWSO13] mesh only the fluid volume. The former approach allows topological changes like collisions to be handled automatically without special treatment, although at the cost of maintaining a mesh over a potentially much larger domain. Clausen et al. also describe techniques for guaranteeing incompressibility and momentum conservation that are applicable to both approaches. Two key advantages offered by these methods are that (i) the advection step causes no numerical diffusion, because physical quantities move with the mesh nodes, and that (ii) surface tension can be modeled accurately thanks to an explicit surface representation tied directly to the simulation mesh.

Apart from simply adding or removing vertices, surface tracking algorithms in fluid dynamics may also move vertices along the surface in order to optimize mesh shapes. A process called "null-space smoothing", which slides vertices within the tangent space of a meshed surface, is used in several works [Jia07, BB09, BBB10, WRK*10, CWSO13]. This strategy improves the quality of simulation elements without changing the shape of the tracked surface.

Additional challenges and techniques When refinement is performed, the position of the newly inserted node has to be chosen carefully. Simply placing it at the midpoint of the original element can cause physical quantities such as bending to change discontinuously, injecting artificial energy into the system and leading to instabilities. Instead, it is better to adjust the mesh locally to bring it into an energy-minimizing configuration. Spillmann and Teschner [ST08] consider the positions of the new node $\mathbf{x}_i$ and its neighbors as variables and perform an optimization to minimize the total energy,

$$U(\mathbf{x}_1,\ldots,\mathbf{x}_n) - \sum_{j=1}^{n} \mathbf{f}_j^T \mathbf{x}_j, \tag{17}$$

where U is the internal energy due to elastic forces, and $\mathbf{f}_j$ is the external force acting on node j. A similar approach has been used for simulating the behavior of stiff two-dimensional sheets such as paper and metal [NPO13, PNdJO14], but with $\mathbf{f}_j$ replaced with an acceleration-corrected term $\mathbf{f}_j - m_j\mathbf{a}_j$ to preserve the instantaneous acceleration of each node.

Liquids with surface tension may freely transition between volumes, thin films, filaments, and droplets; representing these transitions is a challenge for most mesh-based techniques. Zhu et al. [ZQC*14] address this problem using non-manifold meshes of mixed dimensionality, composed of tetrahedra, triangles, segments, and points. Beyond the traditional remeshing operations that work within a single dimensionality, they also provide operations for dimensionality transitions via element collapse (e.g. transforming a thin triangle to a segment) and merging (e.g. generating a tetrahedron to connect two adjacent triangles with small dihedral angle).

Finally, we point out the recent "power particles" technique of de Goes et al. [dGWH*15], which builds an unstructured mesh at each time step using a Voronoi-style power diagram. This can be viewed as a global remeshing approach like the ones mentioned above [KFCO06, BWHT07], but this method stores physical quantities on Lagrangian particles without maintaining an explicit connectivity, and thus avoids numerical diffusion due to resampling. While this work is not specifically an adaptive strategy, it is a form of spatial discretization that makes adaptivity very easy to implement, and could be a fruitful basis for future work in adaptive simulation.

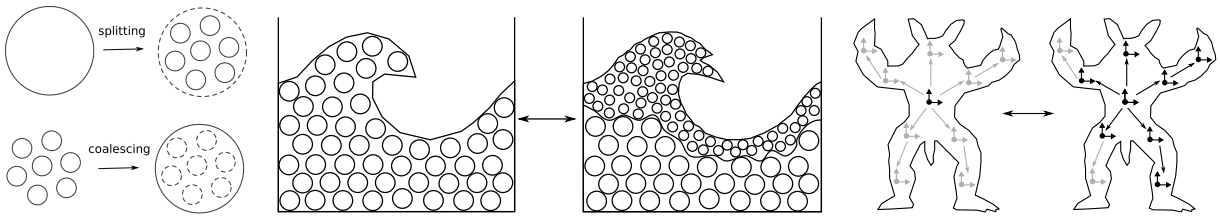


Figure 9: Overview of adaptive meshless techniques. (a) Dynamic local resampling applies splitting and coalescing operators to degrees of freedom in order to locally refine and coarsen regions of interests (b) Multiscale methods couple several simulations with different resolutions. Coarser simulations are used as boundary conditions for the finer resolutions. Feedbacks from the finer resolutions are used to avoid divergences between two different resolutions. (c) Hierarchical refinement dynamically activates or deactivates levels of a precomputed hierarchy between degrees of freedom.

3.3. Meshless models

In the last two decades, numerous meshless models have been extended to perform adaptive physically-based animation. They include smoothed particle hydrodynamics (SPH) (see the survey of Ihmsen et al. [IOS*14]), fluid-implicit particle (FLIP) (see the seminal work of Zhu and Bridson [ZB05]), moving least squares (MLS) (see Muller et al. [MKN*04]) and frame-based models (see Gilles et al. [GBFP11]). These models were used to describe a wide range of phenomena, from fluids (SPH, FLIP) to solids (MLS, frame-based).

Due to the absence of fixed connectivity, meshless models are among the most flexible models for spatial adaptivity. This flexilibity, combined with the variety of models, leads to an impressive number of resampling strategies, developed to resolve details near splashes, large deformations, viscoplastic flows and fractures. We classify these strategies into three categories: (1) dynamic local resampling, (2) multiscale methods, and (3) hierarchical refinement (see Figure 9).

These different strategies make meshless models particularly useful for material that undergo large irreversible deformations such as the one cited above. However, it is important to keep in mind that the flexibility of meshless models come with expensive nearest-neighbor search algorithms to determine the connectivity between material samples at run time.

We inform the reader that complementary information about SPH adaptive techniques can be found in the state-of-the-art report by Ihmsen et al. [IOS*14].

Dynamic local resampling The idea is to dynamically subdivide or merge particles to fit a desired resolution (see Figure 9a). The success of this strategy mainly relies on the resampling scheme's ability to ensure stability, to accurately represent boundaries, and to prevent popping artefacts. Depending on the underlying model (SPH, FLIP, MLS) and its sensitivity to intense resampling, different strategies have been proposed.

As a full particle-based method, the SPH model is a perfect candidate to dynamic resampling. Yet, its sensitivity to particle distribution makes resampling strategies challenging. First, the interaction between particles with different sizes increases the error in the pressure term, leading to instabilities. Therefore smooth grading of resolution is required to minimize this error. Second, the change of positions during resampling can create a sudden change in density which will result in violent pressure forces, which again lead to instabilities (see Orthmann and Kolb [OK12]). Several methods that can be combined were proposed to avoid these local change in density. First, instead of computing the density based on positions, one can use the continuity equation as done by Desbrun and Cani [DC99]. In order to avoid integration error to be accumulated along the simulation, the density still needs to be re-computed based on positions at a user-defined interval. Then, one can perform position optimization to minimize errors during resampling [APKG07] and use quantity blending over time to smooth out inevitable sampling error [OK12]. Also, it is important to keep in mind that another challenge is to efficiently retain the parallel nature of SPH in the adaptive scheme. Zhang et al. [ZSP08] and Yan et al. [YWH*09] propose two different methods to make splitting and merging operators parallel.

More recently, dynamic resampling has been applied to FLIP. As FLIP is a combination of grid and particles, two levels of adaptivity are possible: one on the grid resolution, and the other on the particle sampling. Also, only advection operations are performed on the particles, which results in a less position-sensitive simulation and allows much more flexibility than SPH. More precisely, FLIP does not apply density-based forces to the particles. Consequently, sudden density changes due to particle splitting or merging do not have the same catastrophic consequences as in SPH. However, damping is introduced once particles are merged. This can be taken into account by changing blending parameters of the FLIP simulator as suggested by Ando et al. [ATT12]. Early works on adaptive FLIP perform adaptivity only on particles based on a deformability criterion and the distance to surface [HHK08, ATT12]. Ando et al. use the flexibility of FLIP regarding the particles' positions in order to preserve

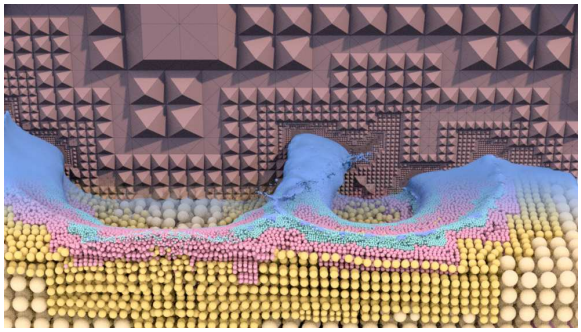


Figure 10: Ando et al. [ATW13] simulate liquid by combining adaptively-sized FLIP particles (bottom, front) with an adaptive tetrahedral mesh for the pressure solve (top, back).

fluid sheets by creating additional particles. In both methods, the largest particle size is bounded by the cell size of the underlying grid, which precludes aggressive adaptive sampling and the use of a fully adaptive FLIP simulator. Ando et al. [ATW13] combine an adaptive BCC mesh (see Section 3.1) with adaptive particle sampling to handle highly different resolutions. (see Figure 10).

When simulating solids, local re-sampling is essential in describing phenomena such as large deformations and fracture. First, like mesh-based methods, large deformations create poorly sampled regions leading to ill-conditioned deformation gradients and instabilities. Second, as explained in Section 3, a challenge in fracture simulation is to reach a sufficiently fine discretization near the tip of the crack in order to obtain a realistic crack path. In these cases, local resampling can greatly improve accuracy and stability while retaining efficiency. However, there are two main challenges that require special care.

The first one consists in accurately describing material discontinuities. Most of the time, shape functions are spherical and they must be modified so that sharp boundaries can be represented. In their work on large viscoplastic deformation and fracture, Pauly et al. [PKA*05] model discontinuities using extended shape functions with transparency criteria, and locally modify a sparse neighborhood graph to update connectivity. Another strategy was proposed by Steinemann et al. [SOG09] to address the high cost of shape functions update. It consists in using a visibility graph to efficiently handle connectivity and approximate material distances used in computing shape functions.

The second one comes from rendering artifacts that can occur at the surface due to splitting. In this context, Jones et al. [JWJ*14] propose a strategy to resample elastoplastic simulation while alleviating popping artifacts. They base their method on the evaluation of each particle neighbor's density. This evaluation is performed using a weighted covariance matrix computed in rest space for each particle i, where $\mathbf{u}_{ij}$ denotes the vector between particle i and particle j, its neighbor.

$$\mathbf{B}_i = \sum_j \frac{\mathbf{u}_{ij}\mathbf{u}_{ij}^T}{\|\mathbf{u}_{ij}\|^4} \tag{18}$$

If the maximum eigenvalue of $\mathbf{B}_i$ is too small, then there are too few particles in the neighborhood and the particle is split in two. New particles are positioned along each side of the eigenvector with the minimum eigenvalue. In order to prevent from rendering artifacts, splittings which are not tangent to the surface are rejected, and particles near the surface are split along the middle eigenvector whose direction is tangent to the surface. Conversely, if the minimum eigenvalue of $\mathbf{B}_i$ is too large, then there are too many particles in the neighborhood and the particle is merged with its closest neighbor. The new particle is positioned halfway between the two merged particles.

Multiscale methods In multiscale methods, several simulations with different resolutions are coupled in a hierarchical way (see Figure 9b). At the coarsest simulation level L_0, the whole domain is discretized. Then, each finer simulation level L_r discretizes a subset of L_{r-1} with a finer scale. This finer subset is defined to match regions of interests that can be physically or visually motivated. Transitions between two scales are bilateral: the coarsest simulation level L_r is used to build boundary conditions for the finer simulation level L_{r+1} and feedbacks from a finer simulation level L_{r+1} to a coarser simulation level Lr are applied, in order to prevent dynamics of two different levels from diverging. Solenthaler and Gross [SG11] and Horvath and Solenthaler [HS13] apply this idea to SPH fluid simulation. Compared to merging and splitting particles, the main advantage is that interactions between different resolutions are not direct anymore. Thus, stability can be more easily ensured and large differences in resolution can be handled. In Solenthaler and Gross's approach, a two-scale simulation is performed. High-resolution regions are defined based on the distance to the surface and the view frustum. In these regions, low-resolution particles emit finer particles according to a cubic pattern which ensures a uniform space sampling. Relaxation steps are performed when particles enter the high-resolution region in order to avoid large pressure forces. Horvath and Solenthaler extend the two-scale simulation to multiscale simulation, and avoid previous artifacts such as mass loss due to particle removal and instabilities due to oversampling near boundaries.

Hierarchical refinement Dynamic resampling techniques were also used in frame-based methods to simulate elastic deformations, see [GBFP11] for a full description of the method. Tournier et al. [TNFG14] use a hierarchical approach to achieve simplifications during deformation without popping artifacts (see Figure 9c). The material is deformed using physically-based control frames organized in a generalized hierarchy. The model can be simplified by attaching frames to their parents at any time in their current rel-

ative positions. Activation and deactivation of nodes is performed based on relative velocity and user-specified metrics, while integration points are updated according to the hierarchy. These hierarchical techniques take advantage of their structure to improve efficiency, but may lack of flexibility, especially regarding topological changes.

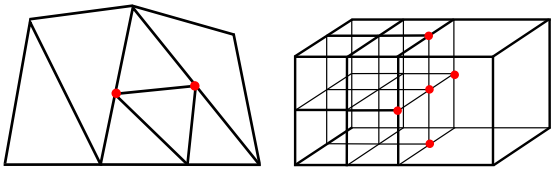
4. Miscellaneous techniques for spatial adaptivity

By far the most popular approach to spatial adaptivity in computer graphics is to add more computational elements where more accuracy or detail is desired, as surveyed in Section 3. This type of spatial adaptivity is often called *h*-refinement, because the length of an edge in a mesh is typically indicated by the letter *h*. In addition to this tried-and-true strategy, there are other fundamentally different approaches for achieving spatial adaptivity. In the next sections, we will discuss strategies that refine the basis within a single element (a superset of *p*-refinement, which refers specifically to the order of a polynomial basis) and during a subspace simulation, strategies that use multiple grids that move and overlap to track locations where more detail is desired, and strategies that mix different reference frames in order to use the computational degrees of freedom optimally.

4.1. Basis refinement

If we wish to achieve spatial adaptivity without explicitly re-meshing (perhaps because it is difficult to control element quality when re-meshing, or because a particular application requires that we preserve the original mesh), then we can perform basis refinement instead. The concept of basis refinement can be a difficult one to grasp for newcomers to the field. One of the best ways to understand basis refinement is in the context of finite element methods (FEM). In generic terms, FEM attempts to approximate a function (typically the solution to a partial differential equation) with a very limited, very specific subset of all possible functions. Most methods in computer graphics use linear interpolation within each element, which essentially restricts the solution to a piecewise linear function. For the purposes of this discussion, we would say that the elements are using linear basis functions, and that the overall solution is expressed in a piecewise-linear basis. However, we can actually represent the solution more accurately (in the sense that the solution converges more quickly under refinement) by using more elaborate bases, like piecewise quadratic functions instead of piecewise linear ones.

This section discusses four types of basis refinement: hierarchical basis refinement, polynomial basis refinement, basis enrichment, and adaptive reduced basis functions. The first three topics discuss adaptivity at the level of basis functions, whereas the fourth is about the adaptive creation of reduced basis in reduced model simulations.



(a) (b)

Figure 11: Refinement by element subdivision is attractive by its simplicity for 2D and 3D meshes, however it introduces T-junctions (in red) at interface between resolutions with different sizes. Bookkeeping or extra-remeshing operations are required to take care of these non-independant degrees of freedom.

Hierarchical basis refinement In computer graphics, hierarchical basis refinement has mainly been applied to finite element simulations (FEM) of solids and shells [CGC*02, GKS02]. The idea is to refine computational basis functions instead of elements. From a theoretical point of view, there are no differences between hierarchical basis refinement and hierarchical element refinement. Both adaptively add more degrees of freedom with increasingly local support in order to improve accuracy where needed. Both use hierarchical schemes in order to efficiently sample the simulation domain. The main differences are practical. By refining basis functions instead of elements, compatibility between regions with different resolutions are implicitly handled. This makes adaptivity much easier and general.

For instance, hierarchical basis refinement allows a simple handling of *T-junctions*. In FEM, each element's node carries a basis and builds a local stiffness matrix from its node's stiffness which are then assemble into a global stiffness matrix. During this process, only independent degrees of freedom should add their contribution to the global matrix. However, when using hierarchical element refinement, non-independent degrees of freedom are added during the subdivision of the simulation mesh. They are called *T-junctions* or *T-nodes* (see Figure 11) and require specific handling. Suddenly, a simple subdivision scheme becomes dependent on the dimensionality, the element type and the basis order, thus requiring important implementation work. In contrast, hierarchical basis refinement handles *T-nodes* at the basis level by making sure that no bases are redundant. The hierarchical structure makes this process simple and thus offers a more general framework for adaptivity which can handle arbitrary resolution differencies.

Capell et al. [CGC*02] embed a high-resolution mesh in a hexahedral complex and precompute a hierarchy of bases up to a given level of refinement. During the simulation, depending on the amount of deformation, each level of the hierarchy will refine or coarsen, thus updating the current set of active bases. Basically, it is the same idea of [DDCB01] but from a basis point of view.

The Conforming Hierarchical Adaptive Refinement Meth-

(a) (b)

Figure 12: Illustrations of p-adaptive techniques for (a) fluids and (b) deformable solids. In (a) Each fluid cell uses a different approximation space depending on its distance to surface (blue line). Surface cells (in red) use fourth order polynomial to precisely approximate pressure. A smooth grading of the approximation is performed inside the fluid (lighter cells) that allows to save computational time. In (b) the canonical tetrahedron with quadratic control points and next to it the quadratic deformation induced by the deformed control mesh. Such a deformation would require many linear elements.

ods (CHARMS) framework of Grinspun et al. [GKS02] generalizes the idea of spatially refining the bases instead of the elements. They provide an in-depth explanation of the concept and describe numerous results of basis refinement applied to shells, solids and electrocardiography simulations. In fact, it is quite surprising that this method was not more studied or extended in the last decade. A possible explanation is the fact that, in the last few years, adaptivity proved to be essential for large and complex deformations such as visco-elastic, visco-plastic flows. In those cases, hierarchical refinement is not sufficient anymore to ensure well-conditioning of the system matrices.

Polynomial basis refinement Polynomial basis refinement methods, also called *p-adaptivity*, increase or decrease the order of the basis functions. For a given spatial resolution, this allows to improve the quality of the deformation without remeshing. In computer graphics, using high order approximation to resolve fine details is not new. However mixing different orders of approximation to adaptively resolve details was only recently applied in the context of fluid simulation and elastic deformations.

For fluid simulation, the smoothness of velocity and pressure makes *p-adaptivity* potentially much more efficient than geometric adaptivity, because the error per degree of freedom decreases exponentially with the approximation order but only geometrically with the spatial resolution. In this favorable context, Edwards and Bridson [EB12, EB14] use polynomial basis refinement in a Discontinuous Galerkin FEM framework in order to simulate detailed water with coarse grids. They use low-order bases deep inside the liquid and increase the basis order closer to the liquid surface, where more visual detail is desired (see Figure 12a). By using basis refinement instead of element refinement, they can keep the simple structure of a low resolution Cartesian grid while pushing back the limit on the scale of details in one cell. In terms of cost, their method is approximately as expensive as a classical high spatial resolution simulation but it provides much more details such as extremely thin sheets.

Bargteil and Cohen [BC14] animate deformable bodies by combining linear and quadratic Bézier elements. The main advantage of their method is the ability to locally increase degrees of freedom and to simulate nonlinear geometry without remeshing (see Figure 12b). To decide whether an element should be linear or quadratic, they compare the linear and quadratic predicted positions of the midpoint of each edge of the element. If the difference between the two positions is larger than a threshold then the edge becomes quadratic. If the difference is less than another threshold then it becomes linear. Thus, some elements can have linear and quadratic edges, usually in transition regions. Bargteil and Cohen observe that, as the number of degrees of freedom increases, visual differences between linear and quadratic elements become difficult to discern. Moreover, the additional cost remains important and local deformations on the surface of a quadratic element due to collisions still cannot be resolved without element refinement. Therefore, their method is particularly efficient on low-resolution models, where it provides smoother geometry and better dynamics quality.

In both cases, the differences of resolution between two regions that can be achieved only using *p-adaptivity* are limited. An important avenue for research would be to combine those methods with geometric adaptive techniques. Such methods have been well studied in engineering fields and are called *hp-adaptive* methods.

Basis enrichment Another way to add spatial detail to a physical model without re-meshing is by using "basis enrichment." The main idea behind basis enrichment is to adaptively add carefully-chosen basis functions that are specifically designed for the phenomena being modeled. (This is in contrast to p-refinement, which is restricted to polynomial functions, regardless of the phenomena being simulated.) For an example of basis refinement, consider an object is being fractured; some material that used to be connected together will have to be split in two. A simple linear basis function could be split into two functions that are linear on one side of the fracture and zero on the other, as illustrated in Figure 13. This type of basis replacement effectively avoids

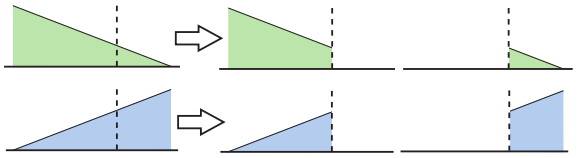


Figure 13: A one-dimensional element with linear basis functions can simulate a fracture by enriching its basis. Here, the new basis functions drop off to zero on the other side of the fracture site indicated by the dashed line, directly encoding the severed connection.

re-meshing by inserting a fracture directly into the basis itself. Note that in this case, we opted to enrich the basis with these particular "step" functions which drop to zero after a certain point, instead of trying to fit some polynomial that might overshoot or otherwise imperfectly capture the desired physics. Such basis enrichment techniques have been termed the "general finite element method" (GFEM) or "extended finite element method" (XFEM) [BGV09].

Adaptive basis enrichment methods have been recently applied in many computer graphics contexts. The virtual node algorithm uses basis enrichment to stabilize fracture and cutting simulations [MBF05, HJST13]. Instead of re-meshing and potentially inserting poorly-conditioned elements, the virtual node algorithm essentially copies the entire element and adapts its basis functions to the fracture site. The virtual node algorithm has recently been improved to robustly allow multiple cuts in a single element [SDF07, WJST14]. One drawback to adaptive basis enrichment is that complicated cuts can require arbitrarily complicated basis functions that may not be simple to compute. In the case of cutting thin shells, Kaufmann and colleagues [KMB*09] note that the most physically-appropriate enrichment functions require the solution of a Laplace equation with time-varying boundary conditions. Similar basis enrichment techniques may not be computationally feasible for the adaptive simulation of more complicated volumetric phenomena.

Adaptive reduced bases As pointed out by Bargteil and Cohen [BC14], volumetric elastic effects are usually low-resolution in space, because we often care about longer time scales in computer graphics. Model reduction methods, also called subspace simulation, exploit this fact to turn extremely costly nonlinear finite element simulations into interactive ones. The idea is to solve equations of motion in a reduced basis that is usually computed using modal analysis or from a database using principal component analysis (PCA). Thus, instead of having a complexity dependent on the simulation mesh resolution, it only depends on the size of the reduced basis (the number of deformation modes) which is typically much smaller. A cubature scheme is used to perform integration on only a reduced sets of well-chosen samples. In the end, this allows simulation of nonlinear deformable models with orders of magnitude speed-up.

The drawbacks are that the creation of the basis requires heavy precomputation, and that the motion of the model is constrained to lie in this basis. Furthermore, while the full finite element simulations often have local compact basis functions and sparse system matrices (because each degree of freedom is a vertex that only influences its neighbors locally), reduced simulations tend to have global basis functions and dense system matrices (because each degree of freedom impacts all points in space simultaneously). Small dense systems are often more efficient than large sparse ones, but the increased storage requirement limits how many modes can be feasibly included in model-reduced simulations, and it limits the use of model reduction in settings where large numbers of modes are essential for visually-plausible behavior (like cloth and fluids). Therefore model reduction is only truly efficient for smooth global deformations and predictable scenarios where it is possible to build a basis that remains constant over time. It is also only useful in situations where a limited number of modes can adequately describe the system.

Several strategies have been proposed to overcome some of these drawbacks by adaptively improving the reduced basis. Common components to those strategies are the different processes to update the basis and the criteria used to decide when the basis should be adapted. Moreover, a common challenge is to ensure temporal coherence while adapting the basis. Kim and James [KJ09] combine a full nonlinear simulation with subspace simulation in order to exploit coherence in the global motion of the model. They incrementally build a reduced-order nonlinear deformable model as the full nonlinear simulation progresses. When possible, full nonlinear steps are skipped with subspace steps resulting in significantly cheaper steps. The challenge of this strategy is to provide efficient operators to update the reduced-basis and to robustly choose which steps can be reduced. We briefly describe the two main operations that compose the incremental construction of the basis: the updating operation and the downdating operation. The updating operation adds a new vector to the basis only if it is significant. To do so, a displacement vector received from a full simulation step is orthogonalized against the existing basis. If its norm is above a given threshold then it is concatenated to the basis. The downdating operator is applied when the basis reached its maximal size r and a new significant vector needs to be added. The basis is then modified so that the $r/2$ most significant directions are preserved. Finally, as the cubature (reduced integration) is dependent on the basis, it also needs to be updated. The updating operation is triggered through different criteria depending on the application. Kim and James describe criteria for quasistatic and dynamic simulations. The dynamic case is particularly challenging as the error is history dependent, which means that taking full steps after reduced steps do not correct errors from the subspace simulation. This method presents impressive speed up for

nonlinear simulations. However, the performance is highly dependent of the rate of expansion of the basis.

As subspace simulation tends to reproduce the global motion of an object, it is particularly challenging to produce very local deformations in this framework. A direct consequence is a simplification of the dynamic behavior. Harmon and Zorin [HZ13] tackle this problem by including *a priori* knowledge in the building of the basis. More precisely, they augment a standard precomputed basis with a dynamic basis. This dynamic basis is built with custom functions derived from analytic solutions to static load. Unpredictable local deformations that arise due to collisions and contacts can then be handled. Moreover, as the change in the basis is very local, they can ensure temporal coherence by projecting the current subspace coordinate vector into the new basis whenever it changes. However, a limitation of the addition of local modes is a restriction on the time step size in order to properly represent the dynamics. Furthermore, the size of local displacements is limited.

Pushing further the idea of using *a priori* knowledge in the building of a reduced basis, Hahn et al. [HTC*14] perform adaptive subspace simulation of cloth. They start with a large amount of high-resolution simulation data from multiple training animations, and convert it into a database of low-dimensional bases associated with poses. Then at each time step, they adaptively choose a subset of low-dimensional bases in the data base depending on the pose of a clothed character. Highly nonlinear folds and wrinkles, which typically are very hard problems for subspace simulation, can then be reproduced. Dynamics is damped near tightly constrained regions such as sleeves, but this may be acceptable in practice for animating tight clothing.

Recently, Teng et al. [TMDK15] propose the use of subspace condensation to locally switch between subspace and fullspace simulation at run time. When dealing with localized deformations, this allows the behavior not to be limited by *a priori* knowledge such as in [HZ13] and [HTC*14].

4.2. Moving grids

As mentioned in Section (3.1), grids are an efficient and simple data structure compared to unstructured meshes. However, this simplicity is counterbalanced by a severe lack of flexibility. Firstly, simulating fluids on very large domains requires a prohibitive amount of memory. Secondly, focusing computational resources on regions of interest remains a challenge. While octrees and other adaptive structured meshes discussed in Section 3.1 address these challenges, they lose the cache-coherent structure that makes uniform grids so efficient. Moving grids methods, also called Chimera grids, allow more flexibility while keeping the advantages of Cartesian grids. The main idea is to use one or more computational grids and allow them to move at each time step to follow the region(s) of interest.

Shah et al. [SCP*04] propose a simple approach in which a single grid is used whose location and size changes to enclose the region containing significant flow. This strategy is useful when there is only a single region of interest in the fluid, such as when simulating explosions.

A more versatile approach is to use multiple independently moving grids, typically centered at each moving object in the scene. The grids may undergo pure translation [CTG10], or may also rotate with the object [DMYN08, EQYF13]; in the latter case, centrifugal and Coriolis forces may also need to be taken into account. A coarser background grid can also be used to represent the global flow in the remainder of the domain not covered by any local grid. The major question that arises in these approaches is how to couple the degrees of freedom in regions where two or more grids overlap.

For interactive smoke simulation, Cohen et al. [CTG10] omit the coupling step entirely. Instead, smoke particles that lie within multiple overlapping grids are simply advected with a weighted average of the flow velocities indicated by different grids. The resulting motion is not physically valid, but works well for interactive applications.

To perform correct coupling for globally incompressible flow, there are two possible strategies: solve for incompressibility as usual in each grid and transfer data between them in an outer loop, or build a single discretization that couples all the grids together. Dobashi et al. [DMYN08] use the former approach to efficiently simulate interaction between smoke and rigid objects. Pressure is solved using a modified Gauss-Seidel solver where each iteration follows three steps. First, data from coarser grids is copied to the boundary cells of finer grids for use as boundary conditions. Then, pressure is computed independently on each grid. Finally, data from the interior cells of finer grids is copied to overlapping coarser grids. This process is repeated until it converges; unfortunately, convergence can be slow. More recently, English et al. [EQYF13] developed a full moving Cartesian grids model. Instead of solving for pressure on each grid separately, they combine all grids in a single discretization. Coarse grid cells that contain the cell center of a finer cell are removed, and a Voronoi mesh is built using the remaining cell centers. An monolithic Poisson solver then computes the pressure over the entire mesh.

In summary, moving grids are a good solution for accelerating Eulerian fluid simulation. In the applied math vocabulary, they belong to dynamic domain decomposition methods. These methods tackle with success the challenge of increasing the local accuracy of Cartesian grids methods while keeping their natural efficiency. However, while such methods are extremely efficient for environments where regions of interests are known or easy to compute, they are not well suited for interactive scenarios where any number of new regions of interest may pop up at any location and time, and where adaptive particle simulation remain more appropriate.

4.3. Mixed models

Another important strategy for adaptively focusing computation in computer animation is to selectively apply a mixture of different computational models. The motivation is that every model has its own strengths and weaknesses, and some are better suited to some situations than others. In particular, many methods combine Eulerian reference frames (which describe motions relative to a fixed point in space) with Lagrangian reference frames (which describe motions relative to a physically important moving trajectory). The driving goal is to judiciously combine techniques in a way that leverages the strengths of each model and suffers none of the drawbacks.

While most of these mixed models represent a clever combination of techniques whose whole is greater than the sum of its parts, the mixed models which could not be classified as "adaptive" have been omitted from this paper. This section only discusses mixed models that adaptively change from one model to another when the situation calls for it. This section is separated into methods used to simulate solid objects and methods used to simulate fluids.

We note that numerous other techniques use two-way coupling between different phenomena ([CMT04, RMSG*08, SSF08, RK13], to name a few). However, while these approaches are adaptive in the sense that they modify computation depending on the phenomena being simulated, we feel these two-way coupling methods are outside of the scope of this report. Instead of surveying all possible combinations of different phenomena-specific discretizations, we only discuss here methods that combine different discretizations of the *same phenomena* in order to gain a computational speedup.

4.3.1. Solids

While most mixed models for simulating solid dynamics do not quite adapt their models to their environment, both Sueda et al. [SJLP11] and Servin et al. [SLNB11] successfully address the challenging problem of simulating stiff elastic strands in a collision-heavy scenario. They accomplished this by introducing Eulerian nodes into a largely Lagrangian strand simulation. The Eulerian nodes sit still at important contact points, while the standard Lagrangian nodes sample the strands as normal. These models are "adaptive" under our definition, because the Eulerian nodes add local detail and their location is decided during run-time.

4.3.2. Fluids

The large memory and computation requirements of 3D fluid discretizations are undesirable, so 2D simplifications are often preferred when applicable. In addition, an Eulerian reference frame is popular for guaranteeing a uniform mesh-spacing, maintaining cache-coherence, avoiding remeshing, and describing swirling flows without explicitly sampling complicated trajectories. However, Eulerian methods are often inferior to their Lagrangian counterparts when sampling fine individual features like droplets and bubbles, or for explicitly tracking many individual vortices.

This section describes many techniques that adaptively combine 2D/3D and Eulerian/Lagrangian techniques in order to get the most out of a fluid simulation. We first survey various methods that combine Eulerian techniques with Lagrangian particles (droplets, bubbles, vortices, etc.). Next, we discuss how some Eulerian models also use Lagrangian particles to couple directly with SPH solvers. After that, we review some methods which combine 3D solvers with 2D techniques or with surface physics.

Eulerian simulation & Lagrangian particles Several early methods combined Eulerian fluid simulations with Lagrangian particles to animate splashing droplets. O'Brien and Hodgins [OH95] combine a 2D pipe-based fluid model with particle-based droplets. Holmberg and Wünsche [HW04] create an Eulerian waterfall model and used Lagrangian particles to animate spray, and Kim et al. [KCC*06] used particles from a 3D surface tracker to fill in missing splash details. Chentanez and Müller [CM10] combine an Eulerian discretization of the shallow water equations with a Lagrangian simulation of spray, splash, and foam particles. The particles add important missing details to the simulations and are allocated dynamically at run-time.

Researchers also use Lagrangian particles to capture bubble behavior in Eulerian simulations. Mould and Yang [MY97] augment a height-field model with Lagrangian particles for droplets and bubbles. Many simulation methods [GH04,HLYK08,PAKF13] compute a 3D Eulerian fluid simulation, and they represent bubbles that are too small to be resolved on the grid with Lagrangian particles. The differences in these methods lie in the varying bubble dynamics and the subtleties of how to transition between the Eulerian grid bubbles and the Lagrangian particle bubbles.

The main concept for these methods is to use an Eulerian representation for the bulk of the flow, but to adaptively turn to a Lagrangian particle representation whenever the Eulerian model is insufficient. This switching point is often easy to detect, because it occurs exactly when the diameter of a water droplet or bubble falls below the Eulerian grid resolution. Not only does this strategy conserve mass and momentum better than simply deleting small features, but it fills in visually important information by animating sprays as a collection of small Lagrangian droplets and foams as a collection of Lagrangian bubbles. Lagrangian droplets and bubbles are practically indispensable in a production workflow, because the small expense of adding additional point geometry with simple physics pays off with enhanced visual realism.

Eulerian simulation & SPH Eulerian methods can also be combined with Lagrangian particles in other ways beyond droplets and bubbles. Losasso et al. [LTKF08] combine SPH

with a FLIP simulation, Wang et al. [WZKQ13] combine SPH with a Lattice-Boltzmann simulation and Chentanez et al. [CMK14] combine SPH with 3D Eulerian grid. This idea of adaptively switching between Eulerian simulations and SPH is still an active research topic. Using SPH instead of simple passive or ballistic particles is clearly more realistic, but it comes with the expense of additional neighborhood operations and more delicate numerical calculations in general. It is not clear yet whether the realism gained by augmenting an Eulerian simulation with SPH particles is worth the computational expense.

Combining 2D & 3D Several techniques like discretizing the shallow water equations [LvdP02,HHL*05] or linearized wave equations [KM90, Tes04, KB14] are useful for reducing computational degrees of freedom, but we do not believe they are inherently *adaptive* by themselves, and we do not discuss them in detail in this document. However, several techniques utilize these 2D discretizations in ways that we would classify as an adaptive "mixed-model" approach.

The work of Thürey et al. [TRS06] combines a 3D Lattice-Boltzmann simulation with a 2D simulation of the shallow water equations, allowing a local region that to adapt to 3D phenomena while distant motion remains a simple height field. Chentanez et al. [CMK14] combine a 3D Eulerian solver with both a 2D shallow water solver and a particle-based fluid simulation. Mixing three models allows them to simulate extremely detailed water interactions at efficient frame rates. These methods fit our definition of adaptivity, because they both locally increase the computational degrees of freedom in interesting regions, and these decisions of where to place the new degrees of freedom are decided at run-time as the simulation progresses. As these papers suggest, adaptively switching simulation dimensions will clearly make animations more efficient, because the computational complexity plummets as the simulation transitions from 3D to 2D. The only reservations here are that there is a significant implementation expense to maintaining two or more solvers (one for each dimension), and the seamless coupling between dimensions is a sensitive process that is still being actively researched.

5. Discussion

Adaptive physically-based models are becoming ubiquitous in computer graphics. In the last decade, various techniques for almost all types of deformable models have been proposed and extended. In this survey, we have classified adaptive techniques into five different categories: (1) temporal adaptivity, (2) geoemtric adaptivity, (3) basis refinement, (4) moving grids and (5) mixed models. For each category, we have described the variants that were developed for different applications and have discussed their strengths and weaknesses.

Among those different categories, geometric adaptive techniques are the most studied, and are perhaps the most intuitive due to their geometrical nature. The many variations in application contexts such as dimensionality and discretization have led to a proliferation of different innovative techniques. Yet, as we have tried to show, there are also many commonalities between aspects of disparate techniques, for example in terms of refinement criteria, and there is potential for consolidation of the many approaches in this area.

In the opposite direction, mixed models represent an important area of work. Unfortunately, as they rely on the specific characteristics of each model, it is more difficult to extract general patterns and strategies. Even so, they perfectly represent the idea and the versatility of adaptive techniques by combining the strengths of different approaches as the simulation evolves.

For now, polynomial basis refinement represents only a small fraction of the methods studied. In computer graphics, this is a very recent topic, but which can build on strong foundations from engineering and applied mathematics where it has been extensively studied. Results in solid and fluid simulation show that polynomial basis refinement can indeed produce impressive animations. One of the most exciting avenues of future research is the combination of this technique with geometric adaptivity.

In subspace simulation, adaptive reduced bases can greatly extend the range of deformation that can be achieved by introducing local and non-linear deformations such as wrinkles. Nevertheless, there is still a large room for improvement and innovation, especially regarding the possibility to handle topological changes and couple different subspace simulations such as deformable solids and fluids.

Even if there are only a few works that focus on temporal adaptivity, these methods are widely used and play a crucial role in ensuring stability and efficiency. Their importance is due to two main reasons. First, spatial and temporal resolution are often strongly related. Secondly, the necessary temporal resolution is inherently dependent on the events occurring during a simulation and cannot always be predicted in advance, necessitating adaptive techniques. As we seek to resolve details at increasingly finer time scales, further research will be required to capture them without paying an exorbitant computational cost.

Adaptive methods are not without limitations, and we briefly summarize the major ones. At present, setting up a new adaptive method is quite difficult: adaptive methods have often been application-specific so far, which makes the study of existing solutions quite intricate. Adaptivity usually makes the implementation of a model much more complex and may ruin the regularity of computations, causing incompatibilities with GPU implementations. Evaluating the future overhead due to the online adaptation process is often difficult, which may make such techniques unusable in performance-constrained contexts such as interactive applications. There are also potential concerns relating to simula-

tion fidelity. If not tackled with care, popping between different spatial and temporal resolutions may cause instabilities and visual artifacts. Furthermore, the energy diffusion that necessarily occurs when permanently adapting a model may be an issue when an accurate simulation is required.

Nevertheless, the space of adaptive simulation techniques is vast and fruitful, and many compelling benefits have been uncovered so far. There is much room for future work in developing new adaptive methods which are both easier to implement and still generic enough to be used in different applications. This challenge requires methods which, for example, minimize the overhead due to additional structures while making it possible to integrate different adaptation criteria. Many other avenues of future research remain, including combinations of different forms of adaptivity and techniques that adapt between different dimensionalities, different formulations, and other characteristics that have traditionally remained separate.

6. Acknowledgements

This work was partly supported by the starting grants ADAPT and BigSplash, as well as the advanced grant EXPRESSIVE from the European Research Council (ERC-2012-StG_20111012, ERC-2014-StG_638176 , and ERC-2011-ADG_20110209).

References

[APKG07] ADAMS B., PAULY M., KEISER R., GUIBAS L. J.: Adaptively sampled particle fluids. *ACM Trans. Graph. 26*, 3 (July 2007). 9, 14

[AR12] ARTEMOVA S., REDON S.: Adaptively restrained particle simulations. *Phys. Rev. Lett. 109* (Nov 2012), 190201. 7

[ATT12] ANDO R., THUREY N., TSURUNO R.: Preserving fluid sheets with adaptively sampled anisotropic particles. *IEEE Transactions on Visualization and Computer Graphics 18*, 8 (Aug. 2012), 1202–1214. 14

[ATW13] ANDO R., THÜREY N., WOJTAN C.: Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. 32*, 4 (July 2013), 103:1–103:10. 9, 11, 15

[AUGA08] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring*, De Floriani L., Spagnuolo M., (Eds.), Mathematics and Visualization. Springer Berlin Heidelberg, 2008, pp. 53–82. 11

[AVGT12] AINSLEY S., VOUGA E., GRINSPUN E., TAMSTORF R.: Speculative parallel asynchronous contact mechanics. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 151:1–151:8. 5

[BB09] BROCHU T., BRIDSON R.: Robust topological operations for dynamic explicit surfaces. *SIAM J. Sci. Comput. 31*, 4 (June 2009), 2472–2493. 13

[BBB10] BROCHU T., BATTY C., BRIDSON R.: Matching fluid simulation elements to surface geometry and topology. *ACM Trans. Graph. 29*, 4 (July 2010), 47:1–47:9. 12, 13

[BC14] BARGTEIL A. W., COHEN E.: Animation of deformable bodies with quadratic bézier finite elements. *ACM Transactions on Graphics (TOG) 33*, 3 (2014), 27. 17, 18

[BD12] BENDER J., DEUL C.: Efficient cloth simulation using an adaptive finite element method. In *VRIPHYS* (2012), Eurographics Association. 11

[BDW13] BUSARYEV O., DEY T. K., WANG H.: Adaptive fracture simulation of multi-layered thin plates. *ACM Trans. Graph. 32*, 4 (July 2013), 52:1–52:6. 9

[BETC12] BENDER J., ERLEBEN K., TRINKLE J., COUMANS E.: Interactive simulation of rigid body dynamics in computer graphics. In *EUROGRAPHICS 2012 State of the Art Reports* (2012), Eurographics Association. 6

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. 21*, 3 (July 2002), 594–603. 3, 6

[BGOS06] BARGTEIL A. W., GOKTEKIN T. G., O'BRIEN J. F., STRAIN J. A.: A semi-lagrangian contouring method for fluid simulation. *ACM Trans. Graph. 25*, 1 (Jan. 2006), 19–38. 10

[BGV09] BELYTSCHKO T., GRACIE R., VENTURA G.: A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering 17*, 4 (2009), 043001. 18

[BH11] BATTY C., HOUSTON B.: A simple finite volume method for adaptive viscous liquids. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 111–118. 11

[BKCN03] BERTAILS F., KIM T.-Y., CANI M.-P., NEUMANN U.: Adaptive wisp tree: A multiresolution control structure for simulating dynamic clustering in hair motion. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), SCA '03, Eurographics Association, pp. 207–213. 12

[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, pp. 28–36. 4

[Bri08] BRIDSON R.: *Fluid Simulation for Computer Graphics*. Ak Peters Series. Taylor & Francis, 2008. 3, 10

[BWHT07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. 26*, 3 (July 2007). 4, 12, 13

[BXH10] BATTY C., XENOS S., HOUSTON B.: Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. In *Proceedings of Eurographics* (2010). 11

[CDGDS13] CRANE K., DE GOES F., DESBRUN M., SCHRÖDER P.: Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 courses* (New York, NY, USA, 2013), SIGGRAPH '13, ACM. 10

[CDS12] CHENG S.-W., DEY T. K., SHEWCHUK J. R.: *Delaunay Mesh Generation*. CRC Press, December 2012. 11

[CFL28] COURANT R., FRIEDRICHS K., LEWY H.: Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen 100*, 1 (1928), 32–74. English translation: Courant et al., "On the partial difference equations of mathematical physics", IBM J. Res. Dev. (1967). 2

[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, pp. 219–228. 12

[CGC*02] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: A multiresolution framework for dynamic deformations. In *Proceedings of the 2002 ACM*

SIGGRAPH/Eurographics Symposium on Computer Animation (2002), SCA '02, ACM, pp. 41–47. 16

[CH97] CARLSON D. A., HODGINS J. K.: Simulation levels of detail for real-time animation. In *Proceedings of the Conference on Graphics Interface '97* (1997), Canadian Information Processing Society, pp. 1–8. 1

[CM03] CERDA E., MAHADEVAN L.: Geometry and physics of wrinkling. *Phys. Rev. Lett. 90* (Feb 2003), 074302. 9

[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. 20

[CM11] CHENTANEZ N., MÜLLER M.: Real-time eulerian water simulation using a restricted tall cell grid. *ACM Trans. Graph. 30*, 4 (July 2011), 82:1–82:10. 11

[CMK14] CHENTANEZ N., MÜLLER M., KIM T.-Y.: Coupling 3D Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (2014), Koltun V., Sifakis E., (Eds.), Eurographics Association, pp. 1–10. 21

[CMT04] CARLSON M., MUCHA P. J., TURK G.: Rigid fluid: animating the interplay between rigid bodies and fluid. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 377–384. 20

[CTG10] COHEN J. M., TARIQ S., GREEN S.: Interactive fluid-particle simulation using translating eulerian grids. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2010), I3D '10, ACM, pp. 15–22. 19

[CWSO13] CLAUSEN P., WICKE M., SHEWCHUK J. R., O'BRIEN J. F.: Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Trans. Graph. 32*, 2 (Apr. 2013), 17:1–17:15. 13

[DC96] DESBRUN M., CANI M.-P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96* (1996), Springer-Verlag New York, Inc., pp. 61–76. 4

[DC99] DESBRUN M., CANI M.-P.: *Space-Time Adaptive Simulation of Highly Deformable Substances*. Rapport de recherche RR-3829, INRIA, 1999. 3, 4, 14

[DCDB00] DEBUNNE G., CANI M.-P., DESBRUN M., BARR A.: Adaptive simulation of soft bodies in real-time. In *Proceedings of the Computer Animation* (2000), CA '00, IEEE Computer Society, pp. 15–. 11

[DDBC99] DEBUNNE G., DESBRUN M., BARR A. H., CANI M.-P.: Interactive multiresolution animation of deformable models. In *Eurographics Workshop on Computer Animation and Simulation'99, September, 1999* (sep 1999), Magnenat-Thalmann N., Thalmann D., (Eds.), Computer Science, Springer, pp. 133–144. 10

[DDCB01] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, ACM, pp. 31–36. 4, 11, 16

[DGW11] DICK C., GEORGII J., WESTERMANN R.: A hexahedral multigrid approach for simulating cuts in deformable objects. *Visualization and Computer Graphics, IEEE Transactions on 17*, 11 (Nov 2011), 1663–1675. 10

[dGWH*15] DE GOES F., WALLEZ C., HUANG J., PAVLOV D., DESBRUN M.: Power particles: An incompressible fluid solver based on power diagrams. *ACM Trans. Graph. 34*, 4 (July 2015), 50:1–50:11. 13

[DMYN08] DOBASHI Y., MATSUDA Y., YAMAMOTO T., NISHITA T.: A fast simulation method using overlapping grids for interactions between smoke and rigid objects. *Computer Graphics Forum 27*, 2 (2008), 477–486. 19

[EB12] EDWARDS E., BRIDSON R.: A high-order accurate particle-in-cell method. *International Journal for Numerical Methods in Engineering 90*, 9 (2012), 1073–1088. 17

[EB14] EDWARDS E., BRIDSON R.: Detailed water with coarse grids: Combining surface meshes and adaptive discontinuous galerkin. *ACM Trans. Graph. 33*, 4 (July 2014), 136:1–136:9. 17

[EQYF13] ENGLISH R. E., QIU L., YU Y., FEDKIW R.: An adaptive discretization of incompressible flow using a multitude of moving cartesian grids. *J. Comput. Phys. 254* (Dec. 2013), 107–154. 19

[FDL07] FONG W., DARVE E., LEW A.: Stability of asynchronous variational integrators. In *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation* (Washington, DC, USA, 2007), PADS '07, IEEE Computer Society, pp. 38–44. 5

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, ACM, pp. 23–30. 3

[FSH11] FIERZ B., SPILLMANN J., HARDERS M.: Element-wise mixed implicit-explicit integration for stable dynamic simulation of deformable objects. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), SCA '11, ACM, pp. 257–266. 5

[FWD14] FERSTL F., WESTERMANN R., DICK C.: Large-scale liquid simulation on adaptive hexahedral grids. *Visualization and Computer Graphics, IEEE Transactions on PP*, 99 (2014), 1–1. 10

[GB14] GOSWAMI P., BATTY C.: Regional Time Stepping for SPH. In *Eurographics 2014* (Apr. 2014), Galin E., Wand M., (Eds.), Eurographics Association, pp. 45–48. 4

[GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM Trans. Graph. 22*, 3 (July 2003), 871–878. 6

[GBFP11] GILLES B., BOUSQUET G., FAURE F., PAI D. K.: Frame-based elastic models. *ACM Trans. Graph. 30*, 2 (Apr. 2011), 15:1–15:12. 14, 15

[GCS99] GANOVELLI F., CIGNONI P., SCOPIGNO R.: Introducing multiresolution representation in deformable object modeling. In *Proceedings of SCCG99* (1999), pp. 149–158. 10

[GH04] GREENWOOD S. T., HOUSE D. H.: Better with bubbles: Enhancing the visual realism of simulated fluid. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), SCA '04, Eurographics Association, pp. 287–296. 20

[GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: Charms: A simple framework for adaptive simulation. *ACM Trans. Graph. 21*, 3 (July 2002), 281–290. 16, 17

[GLM06] GAYLE R., LIN M. C., MANOCHA D.: Adaptive dynamics with efficient contact handling for articulated robots. In *Robotics: Science and Systems* (2006), The MIT Press, pp. 231–238. 7

[GP11] GOSWAMI P., PAJAROLA R.: Time adaptive approximate SPH. In *Proceedings Eurographics Workshop on Virtual Reality Interaction and Physical Simulation* (2011). 7

[GRS*07] GAYLE R., REDON S., SUD A., LIN M. C., MANOCHA D.: Efficient motion planning of highly articulated chains using physics-based sampling. In *Robotics and Automation, 2007 IEEE International Conference on* (2007), IEEE, pp. 3319–3326. 7

[HHK08] HONG W., HOUSE D. H., KEYSER J.: Adaptive particles for incompressible fluid simulation. *Vis. Comput. 24*, 7 (July 2008), 535–543. 9, 14

[HHL*05] HAGEN T. R., HJELMERVIK J. M., LIE K.-A., NATVIG J. R., OFSTAD HENRIKSEN M.: Visual simulation of shallow-water waves. *Simulation Modelling Practice and Theory 13*, 8 (2005), 716–726. 21

[HJST13] HEGEMANN J., JIANG C., SCHROEDER C., TERAN J. M.: A level set method for ductile fracture. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2013), ACM, pp. 193–201. 18

[HK05] HONG J.-M., KIM C.-H.: Discontinuous fluids. *ACM Trans. Graph. 24*, 3 (July 2005), 915–920. 10

[HLYK08] HONG J.-M., LEE H.-Y., YOON J.-C., KIM C.-H.: Bubbles alive. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 48:1–48:4. 20

[Hop96] HOPPE H.: Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 99–108. 11

[HPH96] HUTCHINSON D., PRESTON M., HEWITT T.: Adaptive refinement for mass/spring simulations. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96* (1996), Springer-Verlag New York, Inc., pp. 31–45. 10

[HS13] HORVATH C., SOLENTHALER B.: *Mass Preserving Multi-Scale SPH*. Pixar technical memo 13-04, Pixar Animation Studios, 2013. 15

[HTC*14] HAHN F., THOMASZEWSKI B., COROS S., SUMNER R. W., COLE F., MEYER M., DEROSE T., GROSS M.: Subspace clothing simulation using adaptive bases. *ACM Trans. Graph. 33*, 4 (July 2014), 105:1–105:9. 19

[HVS*09] HARMON D., VOUGA E., SMITH B., TAMSTORF R., GRINSPUN E.: Asynchronous contact mechanics. *ACM Trans. Graph. 28*, 3 (July 2009), 87:1–87:12. 4, 5

[HW04] HOLMBERG N., WÜNSCHE B. C.: Efficient modeling and rendering of turbulent water over natural terrain. In *Proceedings of the 2Nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia* (2004), GRAPHITE '04, ACM, pp. 15–22. 20

[HZ13] HARMON D., ZORIN D.: Subspace integration with local deformations. *ACM Trans. Graph. 32*, 4 (July 2013), 107:1–107:10. 19

[IAGT10] IHMSEN M., AKINCI N., GISSLER M., TESCHNER M.: Boundary handling and adaptive time-stepping for pcisph. In *Proceedings of the Seventh Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2010, Copenhagen, Denmark, 2010* (2010), Erleben K., Bender J., Teschner M., (Eds.), Eurographics Association, pp. 79–88. 3

[IGLF06] IRVING G., GUENDELMAN E., LOSASSO F., FEDKIW R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graph. 25*, 3 (July 2006), 805–811. 11

[IOS*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M.: SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports* (2014), Lefebvre S., Spagnuolo M., (Eds.), The Eurographics Association. 14

[Jef85] JEFFERSON D. R.: Virtual time. *ACM Trans. Program. Lang. Syst. 7*, 3 (July 1985), 404–425. 4, 5

[Jia07] JIAO X.: Face offsetting: A unified approach for explicit moving interfaces. *J. Comput. Phys. 220*, 2 (Jan. 2007), 612–625. 13

[JWJ*14] JONES B., WARD S., JALLEPALLI A., PERENIA J., BARGTEIL A.: Deformation embedding for point-based elastoplastic simulation. *ACM Trans. Graph. 33*, 2 (March 2014). 15

[KB14] KEELER T., BRIDSON R.: Ocean waves animation using boundary integral equations and explicit mesh tracking. In *ACM SIGGRAPH 2014 Posters* (2014), ACM, p. 11. 21

[KCC*06] KIM J., CHA D., CHANG B., KOO B., IHM I.: Practical animation of turbulent splashing water. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, Eurographics Association, pp. 335–344. 20

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Trans. Graph. 25*, 3 (July 2006), 820–825. 12, 13

[KGL07] KABUL I., GAYLE R., LIN M. C.: Cable route planning in complex environments using constrained sampling. In *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling* (2007), SPM '07, ACM, pp. 395–402. 7

[Kim10] KIM B.: Multi-phase fluid simulations using regional level sets. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 175:1–175:8. 10

[KJ09] KIM T., JAMES D. L.: Skipping steps in deformable simulation with online model reduction. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 123:1–123:9. 18

[KLB14] KOSCHIER D., LIPPONER S., BENDER J.: Adaptive tetrahedral meshes for brittle fracture simulation. In *Proceedings of the 2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2014), Eurographics Association. 9, 11

[KLL*07] KIM B., LIU Y., LLAMAS I., JIAO X., ROSSIGNAC J.: Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph. 26*, 3 (July 2007). 10

[KM90] KASS M., MILLER G.: Rapid, stable fluid dynamics for computer graphics. In *ACM SIGGRAPH Computer Graphics* (1990), vol. 24, ACM, pp. 49–57. 21

[KMB*09] KAUFMANN P., MARTIN S., BOTSCH M., GRINSPUN E., GROSS M.: Enrichment textures for detailed cutting of shells. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 50. 18

[KNO14] KOH W., NARAIN R., O'BRIEN J. F.: View-dependent adaptive cloth simulation. In *The Eurographics / ACM SIGGRAPH Symposium on Computer Animation, SCA '14, Copenhagen, Denmark, 2014.* (2014), Eurographics Association, pp. 159–166. 9

[KRK08a] KIM S., REDON S., KIM YOUNG J.: Continuous Collision Detection for Adaptive Simulation of Articulated Bodies. *Visual Computer 24*, 4 (Apr 2008), 261–269. 7

[KRK08b] KIM S., REDON S., KIM Y. J.: View-dependent dynamics of articulated bodies. *Computer Animation and Virtual Worlds 19*, 3-4 (2008), 223–233. 7

[KS07] KLINGNER B. M., SHEWCHUK J. R.: Aggressive tetrahedral mesh improvement. In *Proceedings of the 16th International Meshing Roundtable* (Oct. 2007), pp. 3–23. 13

[KTS*14] KAUFMAN D. M., TAMSTORF R., SMITH B., AUBRY J.-M., GRINSPUN E.: Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Trans. Graph. 33*, 4 (July 2014), 123:1–123:12. 5

[LFO05] LOSASSO F., FEDKIW R., OSHER S.: Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids 35* (2005). 10

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 457–462. 10

[LMOW04] LEW A., MARSDEN J. E., ORTIZ M., WEST M.: Variational time integrators. *Int. J. Numer. Methods Eng 60* (2004), 153–212. 5

[LR56] LAX P. D., RICHTMYER R. D.: Survey of the stability of linear finite difference equations. *Communications on Pure and Applied Mathematics 9*, 2 (1956), 267–293. 3

[LS07] LABELLE F., SHEWCHUK J. R.: Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph. 26*, 3 (July 2007). 10, 12

[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 (July 2008), 797–804. 20

[LV05] LI L., VOLKOV V.: Cloth animation with adaptively refined meshes. In *Proceedings of the Twenty-eighth Australasian Conference on Computer Science - Volume 38* (Darlinghurst, Australia, Australia, 2005), ACSC '05, Australian Computer Society, Inc., pp. 107–113. 11

[LvdP02] LAYTON A. T., VAN DE PANNE M.: A numerically efficient and stable algorithm for animating water waves. *The Visual Computer 18*, 1 (2002), 41–53. 21

[MB12] MISZTAL M. K., BÆRENTZEN J. A.: Topology-adaptive interface tracking using the deformable simplicial complex. *ACM Trans. Graph. 31*, 3 (June 2012), 24:1–24:12. 13

[MBF05] MOLINO N., BAO Z., FEDKIW R.: A virtual node algorithm for changing mesh topology during simulation. In *ACM SIGGRAPH 2005 Courses* (2005), ACM, p. 4. 18

[MBTF03] MOLINO N., BRIDSON R., TERAN J., FEDKIW R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *IMR* (2003), pp. 103–114. 10

[MEB*12] MISZTAL M. K., ERLEBEN K., BARGTEIL A., FURSUND J., CHRISTENSEN B. B., BÆRENTZEN J. A., BRIDSON R.: Multiphase flow of immiscible fluids on unstructured moving meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), SCA '12, Eurographics Association, pp. 97–106. 13

[MFRC13] MANTEAUX P.-L., FAURE F., REDON S., CANI M.-P.: Exploring the Use of Adaptively Restrained Particles for Graphics Simulations. In *VRIPHYS 2013 - 10th Workshop on Virtual Reality Interaction and Physical Simulation* (Sept. 2013), Eurographics Association, pp. 17–24. 8

[Mir00] MIRTICH B.: Timewarp rigid body simulation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 193–200. 4, 5

[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), SCA '04, Eurographics Association, pp. 141–151. 14

[Mon92] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics 30* (1992), 543–574. 3

[MR07] MORIN S., REDON S.: A force-feedback algorithm for adaptive articulated-body dynamics simulation. In *Robotics and Automation, 2007 IEEE International Conference on* (April 2007), pp. 3245–3250. 7

[MY97] MOULD D., YANG Y.: Modeling water for computer graphics. *Computers & Graphics 21*, 6 (1997), 801–814. 20

[NPO13] NARAIN R., PFAFF T., O'BRIEN J. F.: Folding and crumpling adaptive sheets. *ACM Trans. Graph. 32*, 4 (July 2013), 51:1–51:8. 13

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 152:1–152:10. 9, 13

[OGRG07] OTADUY M. A., GERMANN D., REDON S., GROSS M.: Adaptive deformations with fast tight bounds. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, Eurographics Association, pp. 181–190. 8, 12

[OH95] O'BRIEN J. F., HODGINS J. K.: Dynamic simulation of splashing fluids. In *Proceedings of the Computer Animation* (1995), CA '95, IEEE Computer Society, pp. 198–. 20

[OK12] ORTHMANN J., KOLB A.: Temporal blending for adaptive sph. *Comp. Graph. Forum 31*, 8 (Dec. 2012), 2436–2449. 14

[PAKF13] PATKAR S., AANJANEYA M., KARPMAN D., FEDKIW R.: A hybrid lagrangian-eulerian formulation for bubble generation and dynamics. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2013), SCA '13, ACM, pp. 105–114. 20

[PKA*05] PAULY M., KEISER R., ADAMS B., DUTRÉ P., GROSS M., GUIBAS L. J.: Meshless animation of fracturing solids. *ACM Trans. Graph. 24*, 3 (July 2005), 957–964. 15

[PNdJO14] PFAFF T., NARAIN R., DE JOYA J. M., O'BRIEN J. F.: Adaptive tearing and cracking of thin sheets. *ACM Trans. Graph. 33*, 4 (July 2014), 110:1–110:9. 9, 13

[Pro97] PROVOT X.: Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation '97*, Thalmann D., van de Panne M., (Eds.), Eurographics. Springer Vienna, 1997, pp. 177–189. 4, 6

[RGL05] REDON S., GALOPPO N., LIN M. C.: Adaptive dynamics of articulated bodies. *ACM Trans. Graph. 24*, 3 (July 2005), 936–945. 6, 7

[RK13] RÉMILLARD O., KRY P. G.: Embedded thin shells for wrinkle simulation. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 50. 20

[RL06] REDON S., LIN MING C.: An Efficient, Error-Bounded Approximation Algorithm for Simulating Quasi-Statics of Complex Linkages. *Computer-Aided Design* (2006). 6

[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. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 46. 20

[Sch02] SCHMIDL H.: *Optimization-based animation*. PhD thesis, The University of Miami, 2002. 6

[SCP*04] SHAH M., COHEN J. M., PATEL S., LEE P., PIGHIN F.: Extended galilean invariance for adaptive fluid simulation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), SCA '04, Eurographics Association, pp. 213–221. 19

[SDF07] SIFAKIS E., DER K. G., FEDKIW R.: Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), Eurographics Association, pp. 73–80. 18

[SG11] SOLENTHALER B., GROSS M.: Two-scale particle simulation. *ACM Trans. Graph. 30*, 4 (July 2011), 81:1–81:8. 15

[She02] SHEWCHUK J.: What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley 73* (2002). 12

[SJLP11] SUEDA S., JONES G. L., LEVIN D. I. W., PAI D. K.: Large-scale dynamic simulation of highly constrained strands. *ACM Trans. Graph. 30*, 4 (July 2011), 39:1–39:10. 20

[SLD09] SIMNETT T. J. R., LAYCOCK S. D., DAY A. M.: An edge-based approach to adaptively refining a mesh for cloth deformation. In *TPCG* (2009), pp. 77–84. 9, 11

[SLN08] SERVIN M., LACOURSIERE C., NORDFELTH F.: Adaptive resolution in physics based virtual environments. *SIGRAD 2008* (2008). 9

[SLNB11] SERVIN M., LACOURSIÈRE C., NORDFELTH F., BODIN K.: Hybrid, multiresolution wires with massless frictional contacts. *Visualization and Computer Graphics, IEEE Transactions on 17*, 7 (July 2011), 970–982. 20

[SOG09] STEINEMANN D., OTADUY M. A., GROSS M.: Splitting meshless deforming objects with explicit surface tracking. *Graphical Models 71*, 6 (2009), 209 – 220. 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2006). 15

[SSF08] SHINAR T., SCHROEDER C., FEDKIW R.: Two-way coupling of rigid and deformable bodies. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), Eurographics Association, pp. 95–103. 20

[SSIF07] SIFAKIS E., SHINAR T., IRVING G., FEDKIW R.: Hybrid simulation of deformable solids. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, Eurographics Association, pp. 81–90. 11

[SSSH11] SEILER M., STEINEMANN D., SPILLMANN J., HARDERS M.: Robust interactive cutting based on an adaptive octree simulation mesh. *Vis. Comput. 27*, 6-8 (June 2011), 519–529. 10

[ST08] SPILLMANN J., TESCHNER M.: An adaptive contact model for the robust simulation of knots. *Comput. Graph. Forum 27*, 2 (2008), 497–506. 13

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 121–128. 3

[Str04] STRIKWERDA J.: *Finite Difference Schemes and Partial Differential Equations, Second Edition*. Society for Industrial and Applied Mathematics, 2004. 3

[SY04] SHI L., YU Y.: Visual smoke simulation with adaptive octree refinement. In *Proceedings of the Seventh IASTED International Conference on Computer Graphics and Imaging* (2004), pp. 13–19. 10

[Tes04] TESSENDORF J.: Interactive water surfaces. *Game Programming Gems 4* (2004), 265–274. 21

[TMDK15] TENG Y., MEYER M., DEROSE T., KIM T.: Subspace condensation: Full space adaptivity for subspace deformations. *ACM Trans. Graph. 34*, 4 (July 2015), 76:1–76:9. 19

[TNFG14] TOURNIER M., NESME M., FAURE F., GILLES B.: Seamless Adaptivity of Elastic Models. In *Graphics Interface* (May 2014), Kry P. G., Bunt A., (Eds.), Canadian Information Processing Society Toronto, pp. 17–24. 15

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 205–214. 1

[TPS08] THOMASZEWSKI B., PABST S., STRASSER W.: Asynchronous cloth simulation. *Computer Graphics International* (2008). 5

[TRS06] THÜREY N., RÜDE U., STAMMINGER M.: Animation of open water phenomena with coupled shallow water and free surface simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, Eurographics Association, pp. 157–164. 21

[VB05] VILLARD J., BOROUCHAKI H.: Adaptive meshing for cloth animation. *Eng. with Comput. 20*, 4 (Aug. 2005), 333–341. 10

[WDGT01] WU X., DOWNES M. S., GOKTEKIN T., TENDICK F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Computer Graphics Forum 20*, 3 (2001), 349–358. 8, 11

[WDW11] WU J., DICK C., WESTERMANN R.: Interactive high-resolution boundary surfaces for deformable bodies with changing topology. In *Proceedings of 8th Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS) 2011* (2011), pp. 29–38. 10

[WDW13] WU J., DICK C., WESTERMANN R.: Efficient collision detection for composite finite element simulation of cuts in deformable bodies. *The Visual Computer (Proc. CGI 2013) 29*, 6-8 (2013), 739–749. 10

[WJST14] WANG Y., JIANG C., SCHROEDER C., TERAN J.: An adaptive virtual node algorithm with robust mesh cutting. 18

[WRK*10] WICKE M., RITCHIE D., KLINGNER B. M., BURKE S., SHEWCHUK J. R., O'BRIEN J. F.: Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph. 29*, 4 (July 2010), 49:1–49:11. 8, 9, 12, 13

[WT08] WOJTAN C., TURK G.: Fast viscoelastic behavior with thin features. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 47:1–47:8. 10, 11

[WWD14a] WU J., WESTERMANN R., DICK C.: Physically-based simulation of cuts in deformable bodies: A survey. In *Eurographics 2014 State-of-the-Art Report* (2014), Eurographics Association, pp. 1–19. 10

[WWD14b] WU J., WESTERMANN R., DICK C.: Real-time haptic cutting of high resolution soft tissues. *Studies in Health Technology and Informatics (Proc. Medicine Meets Virtual Reality 2014) 196* (2014), 469–475. Published by IOS Press. 10

[WZKQ13] WANG C.-B., ZHANG Q., KONG F.-L., QIN H.: Hybrid particle-grid fluid animation with enhanced details. *Vis. Comput. 29*, 9 (Sept. 2013), 937–947. 21

[YWH*09] YAN H., WANG Z., HE J., CHEN X., WANG C., PENG Q.: Real-time fluid simulation with adaptive sph. *Comput. Animat. Virtual Worlds 20*, 2-3 (June 2009), 417–426. 14

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph. 24*, 3 (July 2005), 965–972. 14

[ZLC*13] ZHU B., LU W., CONG M., KIM B., FEDKIW R.: A new grid structure for domain extension. *ACM Trans. Graph. 32*, 4 (July 2013), 63:1–63:12. 11

[ZQC*14] ZHU B., QUIGLEY E., CONG M., SOLOMON J., FEDKIW R.: Codimensional surface tension flow on simplicial complexes. *ACM Trans. Graph. 33*, 4 (July 2014), 111:1–111:11. 13

[ZSP08] ZHANG Y., SOLENTHALER B., PAJAROLA R.: Adaptive sampling and rendering of fluids on the gpu. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on*

Point-Based Graphics (2008), SPBG'08, Eurographics Association, pp. 137–146. 14