Continuum Mechanics

Object deformations cause forces that try to restore an undeformed object state. In this context, continuum mechanics establishes concepts how to measure or define deformations and how to derive the respective forces. E.g., if two end points of an elastic spring move relative to each other, this corresponds to a strain of the spring which induces stress which finally corresponds to forces at the end points to minimize the spring deformation. Another example would be a small fluid volume. If this volume gets compressed, this is interpreted as strain (typically some ratio between uncompressed and compressed volume) which causes stress (known as pressure). From stress, a force is derived that tries to restore an uncompressed state of the fluid volume. In a fluid, this force is proportional to the negative gradient of the pressure. Continuum mechanics allows to derive forces for a variety of materials with various deformation measures. E.g., elastic solids cause forces in case of compression, while snow is partially plastic and does not necessarily cause a force when compressed. Elastic solids react to shear deformation, while fluids do not. Fluids, however, react to shear rates which accounts for viscosity.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - Continuum Mechanics - SIGGRAPH Course - 2012.pdf (Chapters 2, 3)

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - MPM survey - 2016.pdf (Chapters 5, 6)

https://cg.informatik.uni-freiburg.de/course_notes/sim_02_elasticSolids.pdf

Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics SPH is used to approximate function values and their derivatives at arbitrary positions from known function values at unorganized samples. A popular application for SPH is particle fluid simulation, where particles / samples at arbitrary positions carry the fluid properties. Here, SPH is used to approximate the density / volume at a sample in order to detect compression which is encoded as pressure at a particle. SPH is then used to approximate the gradient of the pressure field at each particle / sample which results in a particle force. SPH is also used, e.g., to approximate the viscous stress at a particle. The SPH concept is not limited to fluids, but is applied to a variety of materials that are represented as a set of small volumes / particles, e.g. elastic solids, highly viscous fluids, snow and even rigid bodies.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH survey - 2019.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_03_particleFluids.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH survey - 2005.pdf

SPH for Fluids

SPH is increasingly popular in the simulation of fluids. In order to get a fully fledged fluid simulation, the SPH concept has to be embedded into an algorithm with various steps. First, each particle has to find its neighbor particles within a given distance. These neighbors are required to compute all SPH approximations for, e.g. density and pressure gradient. Then, the density deviation, i.e. strain, is computed. Pressure, i.e. stress, is computed from the density deviation. Finally, pressure forces are computed and the particle velocities and positions are updated with a numerical integration scheme. The interaction with solid boundaries is typically realized with pressure forces. I.e., if a particle moves closer to a boundary, it gets compressed which results in a pressure force that accelerates the particle away from the boundary.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH survey - 2019.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_03_particleFluids.pdf

Neighbor Search in SPH Fluids

The neighbor search in SPH simulations is an expensive task. That's why, spatial data structures are investigated to accelerate the search. While the typically employed concept of a uniform grid is simple, its implementation offers some degrees of freedom with significant performance differences. In this context, hashing and sorting are two concepts that are often used in implementations of uniform grids. Using recent variants of the sorting concept, billions of particles and their neighbors can be processed in real-world simulation scenarios.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH dataStructures - 2019.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH survey - 2019.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_03_particleFluids.pdf

Boundary Handling in SPH Fluids

Fluid particles should not move through, e.g., the wall of a solid container. Solving this problem is not only important for the realism of a simulation, but a robust handling of the fluid-boundary interaction also influences the efficiency of a simulation. In particle simulations, solid boundaries are often represented with particles. If fluid particles approach boundary particles, the fluid particles compress and the resulting pressure force accelerates fluid particles away from the boundary. While the general concept is simple, an efficient and versatile implementation of the concept can be involved. E.g., the SPH concept requires various layers of boundary particles and the question arises, whether it would be possible to just use one layer. It would also be beneficial to use boundary particles of varying size.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH boundaries - 2012.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH survey - 2019.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_03_particleFluids.pdf

Pressure Computation in SPH Fluids

Pressure computation in SPH fluids can be simple. E.g., when using a state equation, the estimated density deviation / strain is just multiplied with a stiffness constant to get pressure / stress. This local computation is fast, simple to implement, but very small time steps are required for a stable simulation. In contrast, so-called incompressible SPH variants solve a linear system to compute pressure. This global pressure computation sounds expensive, but recent matrix-free implementations show that the overhead is acceptable. On the positive side, global pressure solvers work with much larger time steps than state-equation solvers, in particular in challenging simulation scenarios.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - SPH survey - 2019.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_03_particleFluids.pdf

Material Point Method

The Material Point Method MPM is a very popular SPH alternative in the graphics community. In contrast to SPH, it uses two sample sets: particles and grid points. Fluid properties are consistently updated on the particles and on the grid. For performance and accuracy reasons, only some approximations are computed on the particles, while others are computed at grid points. This particularly avoids the expensive neighbor search per simulation step. Similar to SPH, MPM is used in the simulation of a variety of materials and their interactions.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - MPM survey - 2016.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - MPM survey - 2019.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - MPM overview - 2020.pdf

Grid Fluids

SPH performs its computation on particles. MPM uses particles and grid cells. Now, grid approaches do most computations on grid cells (although they also often employ some sort of particles that are advected with the fluid flow). Grid approaches can be conceptually simple, when, e.g., fluid properties and its derivatives are computed with finite differences on a regular grid. On the other hand, the realization of grid solvers gets more involved in case of free surfaces or when the fluid interacts with complex geometries.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - Grid Fluids - GPU - 2005.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - Grid Fluids - SIGGRAPH Course - 2007.pdf

Numerical Time Integration

The computation of particle positions over time, i.e. the approximation of particle trajectories, is a central task in animation. Therefore, Newton’s Second Law is considered as governing equation. An ordinary differential equation describes the behavior of the particle positions in terms of their derivatives with respect to time. Numerical integration is employed to approximately solve the governing equation, i.e. to approximate the unknown particle positions over time. There exist a variety of methods that vary in computation time and accuracy. Expensive methods are typically more accurate and more robust for large time steps.

Sources:

https://cg.informatik.uni-freiburg.de/course_notes/sim_01_particleMotion.pdf

Position Based Dynamics

While many simulation concepts compute forces due to some deformation, Position Based Dynamics PBD computes position changes, i.e. particle displacements. The approach makes use of concepts from continuum mechanics and also employs the SPH concept to approximate derivatives. Working with displacements instead of forces is motivated by robustness and efficiency of the resulting simulations. As in other simulation concepts, a large variety of materials can be handled.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - PBD survey - 2017.pdf

Rigid Bodies

Although a rigid body can also be interpreted as a set of particles, the computation of its dynamics differs from other materials. In contrast to a particle, a rigid body does not just have a scalar mass, but is characterized by a distribution of mass within its volume. Also in contrast to a particle, a rigid body has an orientation and this orientation changes over time. Thus, forces do not only influence the linear velocity of a rigid body, but also its angular velocity.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - Rigids - Survey - 2014.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_05_rigidBodies.pdf

Bounding Volume Hierarchies

Spatial data structures are important for many queries in animation. While space subdivision concepts, e.g. uniform grids, are popular in the neighbor search of SPH fluids, bounding volume hierarchies BVH are generally popular in all kinds of collision queries, in particular when dynamic rigid bodies are involved. The idea of a BVH is to enclose a complex geometry with a simple bounding shape and then to recursively subdivide the geometry into smaller parts which are enclosed by smaller bounding shapes. In a collision test, the resulting hierarchy is queried which is significantly more efficient than testing all primitives of the geometries.

Sources:

https://cg.informatik.uni-freiburg.de/intern/seminar/animation - Bounding volumes - 2000.pdf