SIMULATION OF LIQUID-, DEFORMABLE- AND RIGID-BODIES

Niels Boldt
boldt@diku.dk
University of Copenhagen
Department of Computer Science
Denmark

Abstract

Realistic interaction between rigid-, deformable- and liquid-bodies can add an substantial realism in interactive applications such as surgery simulators or computer games. In this paper we propose a particle based method to simulate the three type of bodies and their interactions. Compared to previous methods our method uses the same geometrical representation for all objects which makes collision detection and interaction particular simple. A simulator illustrating the capabilities of the proposed method has been implemented and results are shown.
Keywords: Simulation, Deformable-Bodies, Liquid-Bodies, Rigid-Bodies, Interaction, Particle Based

Nomenclature

$\mathbf{a}$	Acceleration
$\mathbf{a}_{cm}$	Acceleration of center-of-mass(CM)
c	Connection
C	Set of connections connected to a particle
e	Particle
E	A set of particles
$\mathbf{f}$	Force
$\mathbf{g}$	Gravity
$\mathbf{I}$	Inertia tensor
$\mathbf{I}_{body}$	Inertia tensor de£ned in body-space
k_c	Hooke spring constant.
l	Initial length of connection
$\mathbf{L}$	Angular momentum
m	Mass of a particle
M	Mass of a rigid-body
p	Pressure
$\mathbf{p}$	Position of a particle
r	Radius of particle
$\mathbf{r}_{cm}$	Position of CM for a rigid-body
$\mathbf{R}$	Orientation of a rigid-body
$\mathbf{v}$	Velocity of a particle
$\mathbf{v}_{cm}$	Velocity of CM
$\mathbf{W}$	Normalized integration kernel
ρ	Density
ω	Angular velocity
τ	Torque
μ	Viscosity

Introduction

Simulation of liquid-, deformable-, and rigid-bodies have found widespread use in games, surgery-simulators and virtual prototyping. Physical models to simulate the different bodies has been developed and applied with success in these areas. However, not much attention has been given to integration of the different physical models. This poses a problem for collision detection as the geometric representation often is different and interfaces between different physical models must be de£ned.

In this paper we try to remedy this problem by proposing physical models for all three kinds of bodies that can interact through the same interface.

Our work is based on previous work presented in [JV03] where physical models for rigid- and deformable-bodies are presented together with an interface between the physical models. A volumetric mass-spring model is used to simulate deformable-bodies and all interaction are handled between individual particles of the bodies.

We have extended their work with a physical model for liquid-bodies that can be £tted into the existing framework without further work.

The rest of this paper is organized as follows: First we highlight previous work on hybrid simulation in *Previous Work*. In *Physical Models* the physical model is presented and in *Dynamics* dynamics of the bodies are presented. Our implementation and results is presented in *Implementation and Results* and we conclude the paper with a discussion of future work in *Conclusion and Future Work*.

Previous Work

In [BW97] a hybrid simulation method is proposed. A interleaved simulation paradigm is described where simulation results from the previous time step are feed to the other model in the current time step to produce synchronized motion. Rigid-bodies, cloth, and particle systems can be handled. In [JV03] physical models for rigid- and deformable-bodies are presented that allows for interaction between different bodies. Interactions in the form of forces are computed between individual particles and then applied internally in the physical model to produce synchronized motion. Wagenaar [Wag01] have developed a particle based method to simulate rigid- and deformable-bodies. Interactions are handled by creating constraints between the particles which in turn satis£ed by relaxation. Force- and distance-based constraints are discussed. In [MM04] a method is presented to model interactions between liquid-particles and soft-tissue. Pseudo particles are placed onto the surface representing the tissue and the pseudo particles interact with the liquid particles via a Lennard-Jones potential. In [MCT04] a method is presented to model interactions between rigid-bodies and ¤uids. This is done by constraining the ¤uid inside the rigid-body to rigid-body motion.

Physical Models

In this section we present the physical model used to model the bodies. For simplicity we use forces to model interactions but other possibilities exists.

De£nition 0.1. Particle: *A particle* e *is a set of parameters* $\{\mathbf{p}, \mathbf{v}, m, r\}$ *where*

$\mathbf{p}$	*Position*
$\mathbf{v}$	*Velocity*
m	*Mass*
r	*Radius*

Particles part of a liquid-body are also assigned a density ρ and a pressure p which are used in the equations governing liquid motion.

De£nition 0.2. Connection: *A connection* c *is a set of parameters* $\{e_1, e_2, k, l\}$ *where*

e_1, e_2	*Particles comprising the connection*
k	*Hooke spring constant*
l	*Nominal distance*

Connections are used to model both internal and external interactions. A connection can either be static or dynamic. Static connections are used to model interaction between particles in deformable-bodies. Dynamic connections are used to handle interactions between bodies or self-intersections in deformable-bodies. A static connection exist during a whole simulation while a dynamic connection are created when a pair of particles are within a certain distance threshold and destroyed again when the distance between the pair of particles are larger than the threshold.
The set of connections for a body will only consist of connections that can be created with particles from that body. All connections between different bodies will belong to an interface that handles interactions between the bodies.

De£nition 0.3. Deformable-body: *A deformable-body is a set of parameters* $\{E, C\}$

E	*Set of particles comprising the body*
C	*Set of connections*

A deformable-body can be seen as a set of particles connected by springs. The set C consist of static connections and dynamic connection. The static connections decides the elastic properties while the dynamic connections handles self-intersections inside the body. This model is comparable to other particle models [Pro95] except that a volume is assigned to each particle.

De£nition 0.4. Rigid-body: *A rigid-body is a set of parameters* $\{\mathbf{r}_{cm}, \mathbf{R}, \mathbf{v}_{cm}, \omega, M, \mathbf{I}_{body}, E\}$

$\mathbf{r}_{cm}$	*Position of center of mass(CM)*
$\mathbf{R}$	*Orientation of the body*
$\mathbf{v}_{cm}$	*Velocity of* $\mathbf{r}_{cm}$
ω	*Angular velocity*
M	*Mass of the body*
$\mathbf{I}_{body}$	*Inertia tensor*
E	*Set of particles comprising the body*

A rigid-body is a deformable-body, except that the distances of the connections are £xed. A rigid body is treated as a single entity when determining its motion and therefore additional parameters are introduced to ease this treatment. However, these parameters exists implicitly in the particles, and we could control the motion of a rigid-body by satisfying the £xed-length constraints in each time step. No connections should be created inside the body, as distances between rigid-body particles are £xed.

De£nition 0.5. Liquid-Body: *A liquid-body is a set of parameters* $\{E\ ,\ C\}$ *where*
E *Set of particles comprising the liquid-body*

A liquid-body can be viewed as a deformable-body, however, without static connections. Instead interactions between particles in the body will be handled by smoothed particle hydrodynamics [MCG03]. This will be treated in detail in the next section where we discuss dynamics of the bodies.

De£nition 0.6. Interface: *An interface is a set of parameters* $\{C\}$ *where*
C *Set of connections.*

An interface is used to model interactions between bodies. Particles are connected when they are within a certain distance of each other. This resembles penalty-methods [MW88], where springs are inserted between colliding bodies, hence some of the weaknesses of the penalty method is transfered to our approach. Stiff differential equations and penetrations can occur if the parameters of a connection are not chosen with care. An advantage, is that insertion and removal of springs becomes easier, as it is based purely on a distance threshold between two particles.

Dynamics

In this section we will describe the mechanics governing motion of the bodies. For deformable- and liquid-bodies the motion is decided by the motion of an individual particle while rigid-bodies are treated as a single entity.

Deformable-Body Mechanics

The mechanics of a deformable-body is determined by the mechanics of each of its particles $e_i \in E$. For an individual particle we have according to Newtons second law

$$\mathbf{f} = m\ddot{\mathbf{p}} \tag{1}$$

The force is the sum of all forces acting on the particle. The force model from [JV02] is adapted in a simpli£ed version as follows:

$$\mathbf{f} = \mathbf{f}_C + \mathbf{f}_G \tag{2}$$

where $\mathbf{f}_C$ is the force from the connections of the particle and $\mathbf{f}_G$ is a simpli£ed gravitational vector. Then we can write

$$\mathbf{f}_C = \sum_{i\in C_e} -k_c(\|\mathbf{p}_e - \mathbf{p}_i\| - l_c)\frac{\mathbf{p}_e - \mathbf{p}_i}{\|\mathbf{p}_e - \mathbf{p}_i\|} \tag{3}$$

and

$$\mathbf{f}_G = m\mathbf{g} \tag{4}$$

where k_c is a Hooke spring constant, l_c is initial length of the spring and m is the mass of the particle.

Rigid-Body Mechanics

For a rigid body we just describe mechanics of center of mass (CM) and orientation. We will use $\mathbf{f}_e$ to denote the resulting force on each particle constituting the rigid-body. The resulting force on a rigid-body is computed by summing all contributions from the individual particles. The motion of CM is governed by Newtons second law

$$\mathbf{f}_{res} = \sum_{e\in E} \mathbf{f}_e = M\mathbf{a}_{cm} \tag{5}$$

The orientation is governed by the following equations

$$\tau = \frac{d\mathbf{L}}{dt} = \sum_{e\in E} (\mathbf{p}_e - \mathbf{r}_{cm}) \times \mathbf{f}_e \tag{6}$$

and

$$\omega = \mathbf{I}^{-1}L \tag{7}$$

where $\mathbf{I}$ is the inertia tensor in world space computed by $\mathbf{R}\mathbf{I}_{body}\mathbf{R}^T$, $\mathbf{L}$ is the angular momentum, and ω is the angular velocity. Finally the change in orientation can be written

$$\frac{d\mathbf{R}}{dt} = \tilde{\omega}\mathbf{R} \tag{8}$$

Liquid-Body Mechanics

To model the dynamics of a liquid-body we will use a interpolation method denoted *Smoothed Particle Dynamics* (SPH). SPH was invented to simulate nonaxisymmetric phenomena in astrophysics and presented by Monaghan in [Mon92]. Since then, SPH has evolved to other £elds including liquid simulation. Interactive simulation is obtained in [MCG03] which will form the basis for the dynamics we present in the following.

SPH is an interpolation method for particle systems. With SPH, £eld quantities that are only de£ned at discrete locations can be evaluated anywhere in space.

According to SPH, a scalar quantity A is interpolated at location $\mathbf{p}$ by a weighted sum of contributions from all particles:

$$A_s(\mathbf{p}) = \sum_j m_j \frac{A_j}{\rho_j} \mathbf{W}(\mathbf{p} - \mathbf{p}_j, h) \tag{9}$$

where j iterates over all particles, A_j is the £eld quantity of $\mathbf{p}_j$ and ρ_j is the density. We will refer to this equation as the *SPH rule*. The function $\mathbf{W}(\mathbf{p}, h)$ is called the smoothing kernel with smoothing distance h.

The particle mass and density appear in Equation 9 because each particle represents a certain volume $V_j = m_j/\rho_j$. The mass is constant, but the density ρ_j varies and needs to be evaluated at each time step. Through substitution into Equation 9 we get for the density at location $\mathbf{p}$:

$$\begin{aligned} \rho_s(\mathbf{p}) &= \textstyle\sum_j m_j \frac{\rho_j}{\rho_j} \mathbf{W}(\mathbf{p} - \mathbf{p}_j, h) \\ &= \textstyle\sum_j m_j \mathbf{W}(\mathbf{p} - \mathbf{p}_j, h) \end{aligned} \tag{10}$$

Derivatives of £eld quantities can be obtained by ordinary differentiation.

When using SPH to derive ¤uid equations for particles care must be taken to ensure symmetric equations such that conservation of momentum and symmetry of forces are obtained.

In an Eulerian formulation, isothermal ¤uids are described by a velocity £eld $\mathbf{v}$, a density £eld ρ and a pressure £eld p. The evolution of these quantities over time are governed by two equations. The equation of continuity assures conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{11}$$

and the equation of motion assures conservation of momentum

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v} \tag{12}$$

With particles, the conservation of mass is guaranteed and the equation of continuity can be omitted in favour of Equation 10.

In Equation 12 the term $\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}$ can be replaced by the substantial derivative $D\mathbf{v}/Dt$. Thus the acceleration of a particle i can be written

$$\mathbf{a}_i = \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{f}_i}{\rho_i} \tag{13}$$

See [MCG03] for further details.

Application of the SPH rule to the pressure term in the equation of motion and symmetrizing yields

$$\mathbf{f}_i^{pressure} = \sum_b m_j \frac{\rho_j + \rho_i}{2\rho_j} \nabla \mathbf{W}(\mathbf{p}_i - \mathbf{p}_j, h)$$

Application of the SPH rule to the viscosity term in the equation of motion and symmetrizing yields

$$\mathbf{f}_i^{viscosity} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 \mathbf{W}(\mathbf{p}_i - \mathbf{p}_j, h)$$

See [MCG03] for further details. In [MCG03] surface tension is also modelled using a method proposed in [Mor00].

To relate density to pressure a equation of state is needed. We adopt a variant of the ideal gas law

$$p = k(\rho - \rho_0) \tag{14}$$

as suggested by Desbrun [DC96]

Kernels

Stability, accuracy and speed of the SPH method highly depend on the choice of smoothing kernels. In [MCG03] the following kernels was designed with the above mentioned criterion's in mind.

The following kernel

$$\mathbf{W}(\mathbf{p}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - |\mathbf{p}|^2)^3 & 0 \le |\mathbf{p}| \le h \\ 0 & \end{cases}$$

is used in all but two cases. An important feature of this kernel is that $\mathbf{p}$ only appears squared which means that it can be evaluated without the use of square roots in distance computations. However, if this kernel is used for computation of pressure forces particles tend to build clusters under high pressure. This happens because the gradient approaches zero near the origin. Desbrun [DC96] solves this problem by using a kernel with non-vanishing gradient near the origin. Desbruns kernel is given by

$$\mathbf{W}(\mathbf{r},h) = \frac{15}{\pi h^6}\begin{cases} (h-|\mathbf{p}|)^3 & 0 \leq \mathbf{p} \leq h \\ 0 & \end{cases}$$

For computation of viscosity forces a third kernel

$$\mathbf{W}(\mathbf{p},h) = k\begin{cases} -\frac{|\mathbf{p}|^3}{2h^3} + \frac{|\mathbf{p}|^2}{h^2} + \frac{2h}{|\mathbf{p}|} - 1 & 0 \leq |\mathbf{p}| \leq h \\ 0 & \end{cases}$$

was designed where $k = \frac{15}{2\pi h^3}$. This kernel has the property that it always will decrease the ¤uids kinetic energy. See [MCG03] for further details. It is noted, that the use of this kernel for viscosity computations signi£cantly increased the stability of the liquid simulation.

Implementation and Results

A interactive simulator implementing the described method has been constructed.

Since the smoothing kernels used in SPH have £nite support h, the computational complexity has been reduced by using a grid with cells of size h. Thereby potentially interacting partners for a particle i only need to be searched in i's own cell and the neighbours. This technique reduces the computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$ where m is the average number of particles per grid cell.

This grid has also been used to accelerate collision queries between the bodies, as all particles of the bodies are mapped into the grid.

Integration has been tested with a fourth order Runge-Kutta method, Heun predictor-corrector and the simple Euler step [Hea02]. The Heun predictor-corrector was found to be best with respect to computation time and stability.

In Figure 1 a series of pictures from a simulation of rigid- and deformable-bodies are shown. This simulation runs at approximately 9 frames pr second on a 2.8 Gigahertz Pentium IV. In Figure 2 a series of pictures from a simulation of liquid-, rigid- and deformable-bodies are shown. This simulation runs at approximately 6 frames pr second on a 2.8 Gigahertz Pentium IV. The liquid-body consist of approximately 1000 particles. Both movies can be obtained from *www.jonas.dk/movies*.

The liquid is visualized by drawing the particles as spheres. This is suf£cient to illustrate the results but for proper use better techniques for visualization should be applied.

Conclusion and Future Work

We have presented a particle-based method to unify simulation of rigid-, deformable and liquid-bodies. Currently the implemented simulator can be termed interactive.

Collision queries with rigid- and deformable-bodies should be handled with a bounding volume hierarchy as proposed by Larsson [LAM01] and Gotschalk [GLM96]. Faster integration schemes could be implemented. A global adaptive scheme as proposed in [DC96] based on the Courant-Friedrichs-Lewy (CFL) criterion or a local adaptive scheme as proposed in [HK89] also based on the CFL criterion. With these optimizations simulation where the computational time is less than simulated time appears feasible. This is also known as real-time simulation.

Methods to visualize the liquids must be investigated. Currently no interactive method matching results of of¤ine rendering exists. In [MCG03] point splattering and marching cubes are proposed. It is however concluded that more research must be done both to improve quality and performance.

In the future an GPU based approach would be interesting to investigate. The grid accelerating the collision queries could be replaced by a 3D-texture and pixel shaders should be used to compute interactions between particles. This should give a signi£cant performance boost and release the CPU for other tasks.

An interface based on projections is being researched and promising results have been obtained. This however demands a change of state functions for the bodies. An interface where interaction between bodies are handled by SPH equations could be a possible direction for further research. The idea is presented in [SK98].

Acknowledgements

The author would like to thank Kenny Erleben and Jon Sporring for fruitful discussions and thorough proofreading.

References

[BW97] David Baraff and Andrew Witkin. Partitioned dynamics. Technical report, The Robotics Institute, Carnegie Mellon University, march 1997.

[DC96] Mathieu Desbrun and Marie-Paule Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In R. Boulic and G. Hegron, editors, *Computer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation)*, pages 61–76. Springer-Verlag, Aug 1996. Published under the name Marie-Paule Gascuel.

[GLM96] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Computer Graphics*, 30(Annual Conference Series):171–180, 1996.

[Hea02] Michael T. Heath. *Scienti£c Computing - An Introductory Survey*. McGraw-Hill, second edition, 2002.

[HK89] L. Hernquist and N. Katz. TREESPH - A uni£cation of SPH with the hierarchical tree method. *The Astrophysical Journal Supplement series*, 70, june 1989.

[JV02] J. Jansson and J. S. M. Vergeest. A discrete mechanics model for deformable bodies. *Computer-Aided Design*, 34, 2002.

[JV03] Johan Jansson and Joris S.M. Vergeest. Combining deformable- and rigid-body mechanics simulation. *The Visual Computer*, 2003.

[LAM01] Thomas Larsson and Tomas Akenine-Möller. Collision detection for continuously deforming bodies. Eurographics, pages 325–333, 2001.

[MCG03] Matthias Müller, David Charypar, and Markus Gross. Particle-based ¤uid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 154–159. Eurographics Association, 2003.

[MCT04] Peter J. Mucha Mark Carlson and Greg Turk. Rigid ¤uid: Animating the interplay between rigid bodies and ¤uid. 2004.

[MM04] M. Teschner M. Mueller, S. Schirm. Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Journal of Technology and Health Care*, 12(1), 2004.

[Mon92] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.

[Mor00] Joseph P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33, 2000.

[MW88] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animationr3. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298. ACM Press, 1988.

[Pro95] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In Wayne A. Davis and Przemyslaw Prusinkiewicz, editors, *Graphics Interface '95*, pages 147–154. Canadian Human-Computer Communications Society, 1995.

[SK98] Yoshiaki Oka Seiichi Koshizuka, Atsushi Nobe. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, 26:751 – 769, 1998.

[Wag01] Jeroen Wagenaar. *Physically based simulation and visualization - A particle based approach*. PhD thesis, The Mærsk Mc-Kinney Møller Institute for Production Technology, 2001.

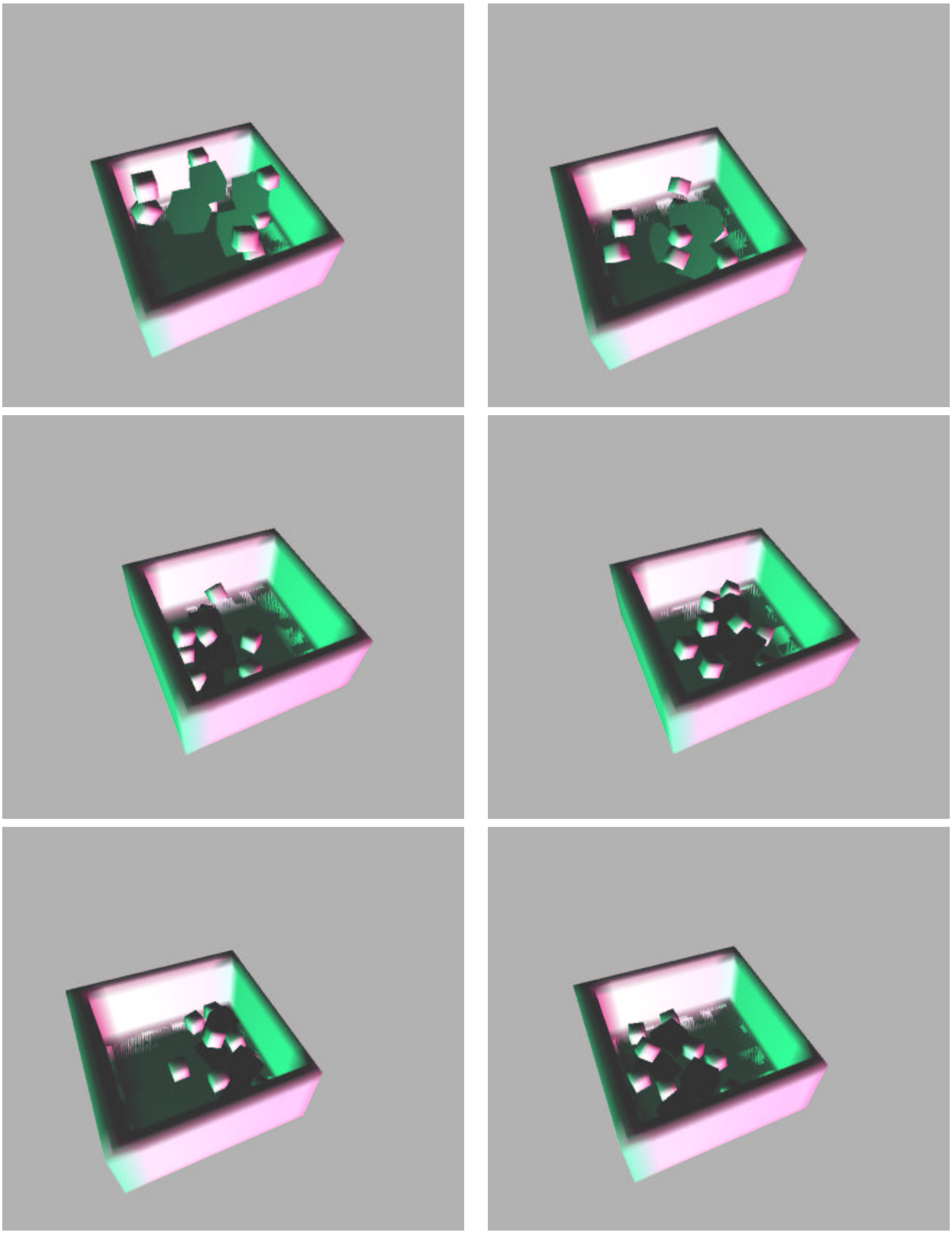

Figure 1: A sequence of frames from a simulation where deformable- and rigid-bodies are tumbling around in a sliding vat. There is 0.2 seconds between each frame. The big green objects are deformable.

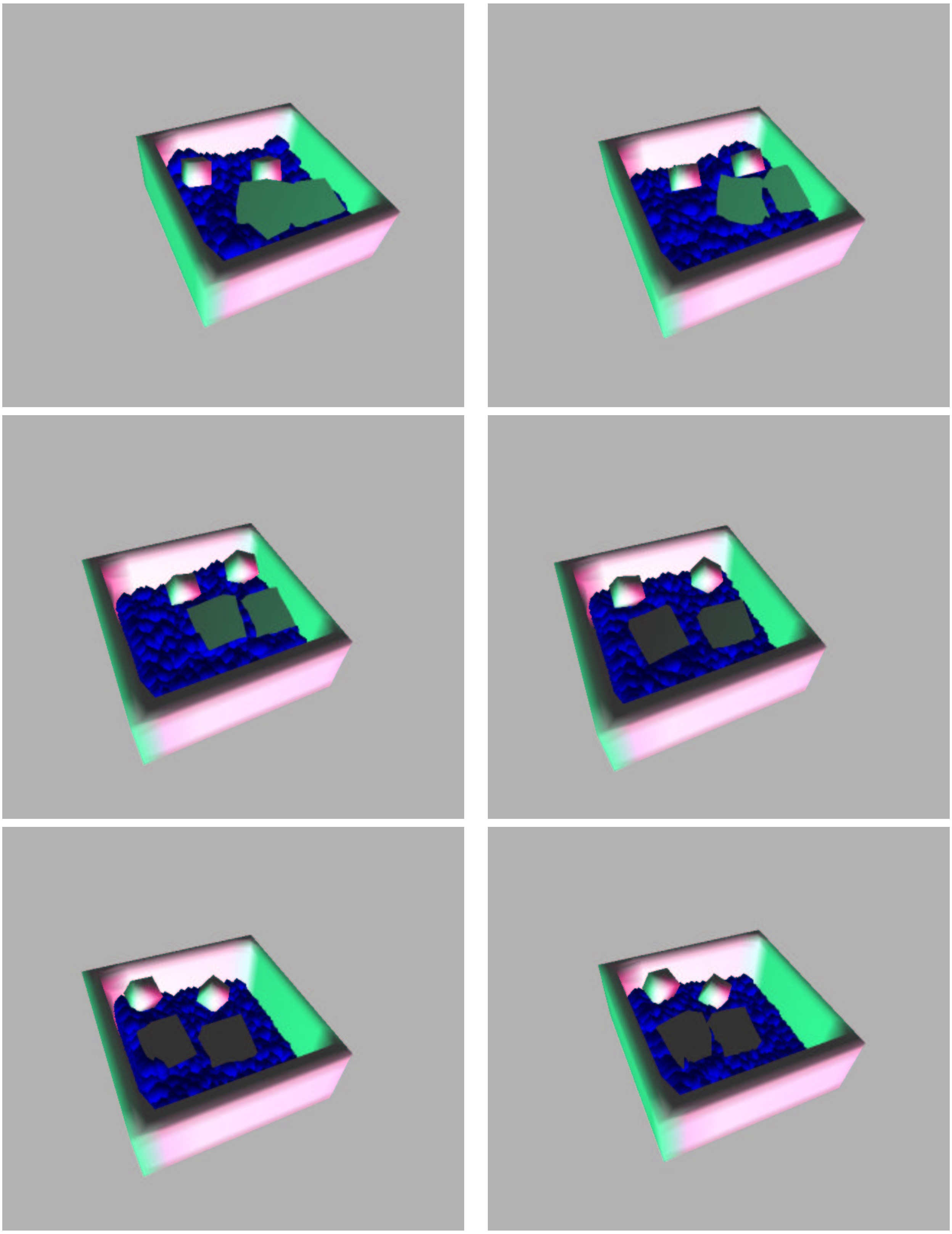

Figure 2: A sequence of frames from a simulation where deformable- and rigid-bodies are ¤oating in a liquid. The vat is sliding. There is 0.2 seconds between each frame. The big green objects are deformable.