Interactive Blood Simulation for Virtual Surgery Based on Smoothed Particle Hydrodynamics

Matthias Müller Simon Schirm Matthias Teschner
Computer Graphics Laboratory - ETH Zurich

Abstract. In this paper, we propose an interactive method based on Smoothed Particle Hydrodynamics (SPH) to simulate blood as a fluid with free surfaces. While SPH was originally designed to simulate astronomical objects, we gear the method towards fluid simulation by deriving the force density fields directly from the Navier-Stokes equation and by adding a term to model surface tension effects. In contrast to Eulerian grid-based approaches, the particle-based approach makes mass conservation equations and convection terms dispensable which reduces the complexity of the simulation. In addition, the particles can directly be used to render the surface of the fluid. Our method can be used in interactive surgical training systems with models of up to 3000 particles.

1 Introduction

Interactive modeling of fluids, such as blood or water, is an essential component in computational surgery. It can add substantial realism to a variety of application scenarios. Further, it can be employed to improve treatment planning, e. g. in aneurysm surgery.

Today's surgical training systems commonly provide a powerful simulation of deformable anatomical models and surgical instruments. However, the integration of interactive fluid models, which can improve the realistic behavior of surgical training systems, still poses a very challenging problem. In endoscopic training systems, the simulation of blood flow due to injured arterial vessels could significantly improve the visual feedback. Further, the suction of blood and irrigation with water could be simulated very realistically.

A second important aspect is treatment planning. In aneurysm surgery, fluid simulation can provide useful information on aneurysm hemodynamics and pathology. This information can be utilized as a design criterion for stents which are used in the treatment of aneurysms. Thus, fluid simulation can be used to improve the quality of the treatment planning and decrease the surgical risk.

In this paper, we propose an interactive fluid model that can be integrated as an additional component into surgical training systems to realistically animate blood or water. Further, the model can support treatment planning in aneurysm surgery.

2 Related Work

Computational Fluid Dynamics (CFD) has a long history. In 1822 Claude Navier and in 1845 George Stokes formulated the famous Navier-Stokes equations that describe the dynamics of

fluids. Besides the Navier-Stokes equation which describes conservation of momentum, two additional equations are needed to simulate fluids namely a continuity equation describing mass conservation and a state equation describing energy conservation. In the CFD literature a large number of methods have been proposed to simulate fluids.

In the fields of computer graphics and virtual reality special purpose fluid simulation techniques have been developed. In 1983 T. Reeves[14] introduced particle systems as a technique for modeling a class of fuzzy objects. Since then both, the particle-based Lagrangian approach and the grid-based Eulerian approach have been used to simulate fluids. Desbrun and Cani[2] and Tonnesen[18] use particles to animate soft objects. Particles have also been used to animate surfaces[6], to control implicit surfaces[19] and to animate lava flows[16]. In recent years the Eulerian approach has been more popular as for the simulation of fluids in general[15], water [4, 3, 17], soft objects[11] and melting effects[1].

So far only a few techniques optimized for the use in interactive systems are available. Stam's grid based method[15] is certainly an important step towards real-time simulation of fluids. Here we propose a particle-based approach based on Smoothed Particle Hydrodynamics (SPH) to animate arbitrary fluid motion.

3 Methods

The most commonly used techniques to simulate fluids are Eulerian or grid-based methods where quantities such as densities, pressures or velocities are computed at fixed locations in space. In contrast, particle-based approaches compute these quantities on particles that follow the fluid making mass conservation equations and convection terms dispensable and, thus, reducing the complexity of the simulation. Since the aim of this project is the real-time simulation of blood for interactive surgery simulators we decided to use a particle method (SPH) to model blood flow. Although SPH was developed for the simulation of astrophysical problems [8, 5], the method is general enough to be used in any kind of fluid simulation.

Section 3.1 describes the principles of SPH, sections 3.2 - 3.5 outline the adaption of the Navier-Stokes equation to SPH, section 3.6 explains how we model interactions with soft tissue and section 3.7 shows how the resulting equations are integrated to animate the particle system.

3.1 Smoothed Particle Hydrodynamics

SPH is an interpolation method for particle systems. With SPH, field quantities that are only defined at discrete particle locations can be evaluated anywhere in space. For this purpose, SPH distributes quantities in a local neighborhood of each particle using radial symmetrical smoothing kernels. According to SPH, a scalar quantity A is interpolated at location $\mathbf{r}$ by a weighted sum of contributions from all particles:

$$A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \tag{1}$$

where j iterates over all particles, m_j is the mass of particle j, $\mathbf{r}_j$ its position, ρ_j the density and A_j the field quantity at $\mathbf{r}_j$. The subscript 'S' indicates a smoothed quantity.

The scalar function $W(\mathbf{r}, h)$ is called the smoothing kernel with core radius h. Since we only use kernels with finite support, we use h as the radius of support in our formulation. If

W is even (i.e. $W(\mathbf{r}, h) = W(-\mathbf{r}, h)$) and normalized, the interpolation is of second order accuracy. The kernel is normalized if $\int W(\mathbf{r})d\mathbf{r} = 1$.

The particle mass and density appear in Eqn. (1) because each particle i represents a certain volume $V_i = m_i/\rho_i$. While the mass m_i is constant throughout the simulation and, in our case, the same for all the particles, the density ρ_i varies and needs to be evaluated at every time step. Through substitution into Eqn. (1) we get for the density at location $\mathbf{r}$:

$$\rho_S(\mathbf{r}) = \sum_j m_j W(\mathbf{r} - \mathbf{r}_j, h). \tag{2}$$

In most fluid equations, derivatives of field quantities need to be evaluated. With the SPH approach, such derivatives only affect the smoothing kernel. The gradient of A is simply

$$\nabla A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \tag{3}$$

while the Laplacian of A evaluates to

$$\nabla^2 A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h). \tag{4}$$

3.2 The Euler Form of the Navier-Stokes Equation

In the Eulerian formulation, isothermal fluids are described by a velocity field $\mathbf{v}$, a density field ρ and a pressure field p. The evolution of these quantities over time is given by two equations. The first equation assures conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{5}$$

while the Navier-Stokes equation[12] formulates conservation of momentum

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}, \tag{6}$$

where $\mathbf{g}$ is an external force density field and μ the viscosity of the fluid. Many forms of the Navier-Stokes equation appear in the literature. Eqn. (6) represents a simplified version for Newtonian incompressible fluids. Although blood is known to be a non-Newtonian fluid Eqn. (6) is appropriate in the context of a real-time environment for surgery simulation and yields visually convincing results.

3.3 Modeling Fluids with Particles

The use of particles instead of a stationary grid simplifies these two equations substantially. First, because the number of particles is constant and each particle has a constant mass, mass conservation is guaranteed and Eqn. (5) can be omitted completely. Second, in the particle formulation, Eqn. (6) takes the simple form:

$$\rho_i \mathbf{a}_i = \mathbf{f}_i, \tag{7}$$

where ρ_i is the density at particle i which can be computed via Eqn. (2), $\mathbf{a}_i$ is the acceleration of particle i and $\mathbf{f}_i$ the force density field – the right hand side of Eqn. (6) – evaluated at the location of particle i. This equation yields the acceleration of every particle at a given time. The positions of the particles at the next time step can be computed by integrating these accelerations over time.

SPH is now used to compute the force density field at the locations of the particles. It is important to realize that forces derived by SPH are not guaranteed to be symmetric. We, thus, symmetrize the forces. For the three force densities on the right hand side of Eqn. (6) we then get:

$$\mathbf{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{8}$$

$$\mathbf{f}_i^{\text{gravity}} = \rho_i \mathbf{g} \tag{9}$$

$$\mathbf{f}_i^{\text{viscosity}} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h). \tag{10}$$

Since particles only carry the three quantities mass, position and velocity, the pressure at particle locations has to be evaluated first. This is done in two steps. Eqn. (2) yields the density at the location of the particle. Then, we compute the pressure via the ideal gas state equation [2] $p = k(\rho - \rho_0)$, where k is a stiffness constant ($10^2 - 10^3\ m^2/s^2$ in our simulations) and ρ_0 the rest density (1000 kg/m^3 for water).

3.4 Surface Tension

We model surface tension forces not present in Eqn. (6) explicitly based on ideas of Morris[9]. The surface of the fluid can be found by using the so called *color field*:

$$c_S(\mathbf{r}) = \sum_j m_j \frac{1}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h). \tag{11}$$

The gradient field of the smoothed color field $\mathbf{n} = \nabla c_s$ yields the surface normal field pointing into the fluid. Surface tension depends on the curvature of the surface κ which is proportional to the divergence of $\mathbf{n}$ and can be computed as $\kappa = \frac{-\nabla^2 c_s}{|\mathbf{n}|}$. For the surface traction we get $\mathbf{t}^{\text{surface}} = \sigma\kappa\frac{\mathbf{n}}{|\mathbf{n}|}$, where σ is the surface tension coefficient. To distribute the surface traction among particles near the surface and to get a force density we multiply by a normalized scalar field $\delta_s = |\mathbf{n}|$ which is non-zero only near the surface. For the force density acting near the surface we finally get

$$\mathbf{f}^{\text{surface}} = \sigma\kappa\mathbf{n} = -\sigma\nabla^2 c_S \frac{\mathbf{n}}{|\mathbf{n}|} \tag{12}$$

3.5 Smoothing Kernels

Stability, accuracy and speed of the SPH method highly depend on the choice of the smoothing kernels. We use even and normalized kernels because they yield interpolations of second order accuracy. We designed the following kernel

$$W_{\text{poly6}}(\mathbf{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \le r \le h \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

and use it in all but two cases. An important feature of this simple kernel is that r only appears squared which means that is can be evaluated without computing square roots in distance computations. However, if this kernel is used for the computation of the pressure forces, particles tend to build clusters under high pressure. As particles get very close to each other, the repulsion force vanishes because the gradient of the kernel approaches zero at the center. Desbrun[2] solves this problem by using a spiky kernel with a non vanishing gradient near the center. For pressure computations we use Debrun's spiky kernel

$$W_{\text{spiky}}(\mathbf{r}, h) = \frac{15}{\pi h^6} \begin{cases} (h-r)^3 & 0 \leq r \leq h \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

that generates the necessary repulsion forces.

To compute the viscosity force, we designed a third kernel:

$$W_{\text{viscosity}}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & 0 \leq r \leq h \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

whose Laplacian

$$\nabla^2 W(\mathbf{r}, h) = \frac{45}{\pi h^6}(h-r) \tag{16}$$

is positive everywhere which is essential for the stability of the simulation. The use of this kernel for viscosity computations increased the stability of the simulation significantly allowing to omit any kind of additional damping.

3.6 Interaction with Soft Tissue

To model the interaction of the fluid particles with deformable objects (e.g. a blood vessel) represented by tetrahedral meshes, we place pseudo particles onto the surface of the mesh. These pseudo particles interact with the fluid particles via a Lennard-Jones-like potential. The potential, both, attracts the fluid particles to the surface (adhesion) and prevents them from penetrating the vessel tissue. The resulting repulsion forces also act symmetrically on the deformable object. This way, the fluid is able to deform the surrounding tissue.

3.7 Simulation

For the integration of the Eqn. (7) we use the Leap-Frog scheme[13]. As a second order scheme for which the forces need to be evaluated only once, it best fits our purposes and in our examples allows time steps up to 10 milliseconds. For the examples we used constant time steps. We expect even better performance if adaptive time steps are used based on the Courant-Friedrichs-Lewy condition[2].

4 Results

The simulations described in this section have been performed on an AMD Athlon 1.8 GHz PC with 512 MB RAM and a GeForce Ti 4400 graphics card with 128 MB RAM.

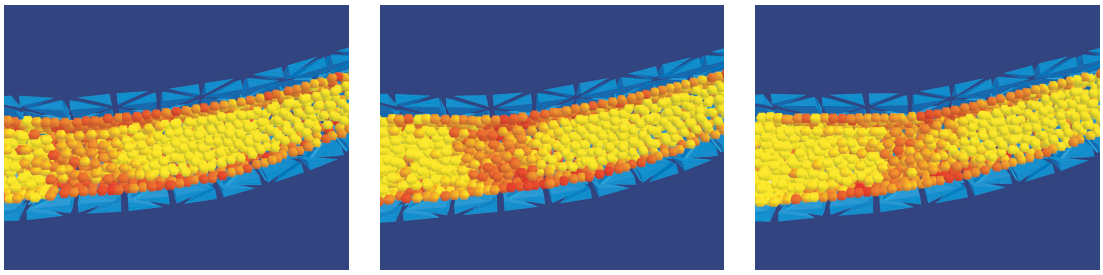

Figure 1: Blood flow through vessel. Fast particles are shown in yellow (bright), slow particles in red (dark).

Fig. 1 illustrates the simulated blood flow through a vessel. The blood stream is modeled with 3000 smoothed particles while the vessel is represented by a tetrahedral mesh consisting of 560 elements. We use the Finite Element based Warped Stiffness approach [10] to compute the dynamic behavior of the deformable tetrahedral mesh. The simulation runs at interactive rates using 66 ms for simulation and 12 ms for rendering per integration time step. The velocities of the particles are color coded. Red (dark) particles are slow while yellow (bright) particles are fast. This way, friction of the fluid with the surrounding tissue as well as pulsation waves inside the fluid are made visible.

Figure 2: Blood flow out of an injured arterial vessel with reconstructed free surface.

The second example (Fig. 2) shows an injured arterial vessel. The free fluid surface is reconstructed as an iso-surface of the color field defined by Eqn. (11) using the marching cubes algorithm [7]. The simulation of the 3000 smoothed particles takes 57 ms per time step while surface reconstruction and surface rendering together take 50 up to 500 ms per time step depending on the resolution of the reconstructed surface mesh.

5 Conclusions

We intend to integrate the presented interactive fluid model into a hysteroscopy simulator that is being developed in a collaboration within our research network NCCR “Co–Me”. Further, we will investigate the potential of the fluid model for improving the treatment planning in aneurysm surgery by employing the fluid simulation as a criterion for stent design and placement.

Currently, our main focus is performance and physically plausible behavior of the simulation, as it is required for surgical training systems. Therefore, the number of particles is limited with respect to this constraint. However, in order to verify the accuracy of our model, we intend to perform simulations with a larger number of particles. Only with a number of particles, significantly higher than the one used in our real-time simulations, experiments can be compared to measured data. Although ongoing work focuses on this comparison, a thorough description of these efforts would go beyond the scope of this paper.

6 Acknowledgements

This project is funded by the National Center of Competence in Research "Computer Aided and Image Guided Medical Interventions" (NCCR "Co–Me") and by the Swiss National "Kommission für Technologie und Innovation" (KTI) project no. 6310.1 KTS-ET.

References

[1] Mark Carlson, Peter J. Mucha, III R. Brooks Van Horn, and Greg Turk. Melting and flowing. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation*, pages 167–174. ACM Press, 2002.

[2] M. Desbrun and M. P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation)*, pages 61–76. Springer-Verlag, Aug 1996.

[3] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 736–744. ACM Press, 2002.

[4] N. Foster and R. Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30. ACM Press, 2001.

[5] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–398, 1977.

[6] Jean-Christophe Lombardo and Claude Puech. Oriented particles: A tool for shape memory objects modelling. In *Graphics Interface'95*, pages 255–262, mai 1995. Quebec city, Canada.

[7] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM Press, 1987.

[8] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, 1977.

[9] J. P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33(3):333–353, 2000.

[10] M. Muller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. *Proceedings of 2002 ACM SIGGRAPH Symposium on Computer Animation*, pages 49–54, 2002.

[11] D. Nixon and R. Lobb. A fluid-based soft-object model. *IEEE Computer Graphics and Applications*, pages 68–75, July/August 2002.

[12] D. Pnueli and C. Gutfinger. *Fluid Mechanics*. Cambridge Univ. Press, NY, 1992.

[13] C. Pozrikidis. *Numerical Computation in Science and Engineering*. Oxford Univ. Press, NY, 1998.

[14] W. T. Reeves. Particle systems — a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics 2(2)*, pages 91–108, 1983.

[15] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.

[16] Dan Stora, Pierre-Olivier Agliati, Marie-Paule Cani, Fabrice Neyret, and Jean-Dominique Gascuel. Animating lava flows. In *Graphics Interface*, pages 203–210, 1999.

[17] T. Takahashi, U. Heihachi, A. Kunimatsu, and H. Fujii. The simulation of fluid-rigid body interaction. *ACM Siggraph Sketches & Applications*, July 2002.

[18] D. Tonnesen. *Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction, and Animation*. PhD thesis, University of Toronto, November 1998.

[19] Andrew Witkin and Paul Heckbert. Using particles to sample and control implicit surfaces. In *Computer Graphics (Proc. SIGGRAPH '94)*, volume 28, 1994.