Adaptiving Time Steps for SPH Cloth-Fluid Coupling

Yalan Zhang, Xiaojuan Ban, Xu Liu, Xiaokun Wang
University of Science and Technology Beijing
yalan.zhang920503@gmail.com, banxj@ustb.edu.cn, xuliu213@163.com, wang1xiao2kun3@163.com

***Abstract*—We propose a new cloth-fluid coupling scheme which takes the advantages of the position-based method. With the constraint to distance and angle, deformable sheet could be implemented and coupled with fluid particles. Furthermore, an adaptive time-stepping method is adopted for the cloth-fluid coupling, which increases and decreases the required time step automatically according to the scenario. While comparatively large time steps can be used, the efficiency of the simulation is significantly improved compared to the constant time-stepping.**

Keywords—Adaptiving time steps; Cloth-fluid coupling; SPH

I. INTRODUCTION

Smoothed Particle Hydrodynamics (SPH) is becoming popular for fluid simulation because of its advantages of solving large deformation and discontinuity of free surface flow. Cloth-fluid coupling is a major bottleneck in fluid simulation, since cloth is deformable and thin, the time step has to be chosen small enough to guarantee non-penetration

Recently, the efficient animation of deformable-fluid coupling can be successfully realized with a variety of methods [1][2][3][4]. [5] extrapolated fluid quantities into solid particles to improve solid boundary conditions, avoiding "stacking" and irregular pressure distributions in standing water situations.

To overcome the fluid penetrations in cloth with high velocity, small simulation time steps have to be used, which reduces the efficiency of the simulation systems. [5] suggested an adaptive time step method for the PCISPH method, which adjust the required time step according to the state of the simulation. [6] suggested fluid particles have their own time-steps determined from local conditions.

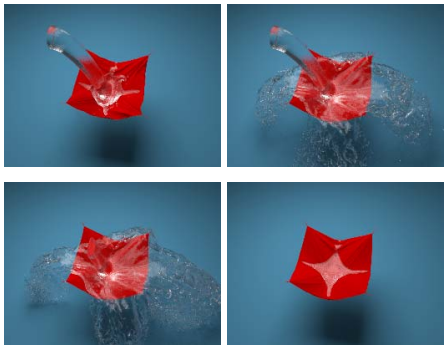


Figure 1 The simulation of deformable-fluid coupling

This paper focuses on the simulation of deformable-fluid coupling, as shown in Figure 1, with large time steps using the SPH method.

Furthermore, an adaptive time-stepping scheme for deformable-fluid coupling is presented, which automatically adapts the time step according to the density fluctuation, the maximum velocity and acceleration.

II. METHOD

As mentioned earlier, the previous methods work well for fluid-rigid coupling. However, homogeneous time step is not sufficient for deformable objects, since large deformations may cause penetration between boundary particles when the time step is not small enough to satisfy CFL condition. In order to avoid this problem, adaptively time steps is necessary. This section starts with a brief introduction of fluid model and a description of deformable structures modeling in Section 2.1, followed by a description of our novel individual time stepping scheme in Section 2.2. Sections 2.3 describes the algorithm.

A. Fluid model and deformable model

We aim to simulate weakly compressible fluids using SPH. In the Lagrangian setting, the density at position x_j is approximated by using the values at a set of neighboring particles x_j:

$$< \rho(x_i) >= \sum_j m_j \frac{\rho_j}{\rho_j} W(x_i - x_j, h) = \sum_j m_j W_{ij} \tag{1}$$

where $W(x_i - x_j, h)$ is the kernel function with the support radius, m_j iterates the mass of particle j.

For weakly compressible SPH, the momentum equation can be written as follow:

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho}\nabla p + \upsilon \nabla \cdot \nabla \mathbf{u} + \mathbf{g} \tag{2}$$

with $\mathbf{u}$ denoting the velocity, p the pressure, $\upsilon \nabla \cdot \nabla \mathbf{u}$ the viscosity and $\mathbf{g}$ the gravity acceleration. Equation (3) represents a simplified version for incompressible fluids and Equation(4) assures conservation of mass.

To relate pressure, we use an EOS [7] as:

$$p_i = \frac{k\rho_0}{\gamma}\left(\left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1\right) \tag{3}$$

where ρ_0 is the rest density of the fluid, usually set as 1000 $kg \cdot m^{-3}$, γ and k control the stiffness, γ is set to 7 and k is chosen.

978-1-5090-2303-5/16 $31.00 © 2016 IEEE
DOI 10.1109/CW.2016.29

Unlike fluid particles, in term of Lagrangian scheme, the topological structure of deformable body vertices will not change. Besides, the relative position and relative angle of deformable body vertices are not constant compared to rigid vertices. As a result, we need to solve a system of non-liner constraints, with one constraint per-particle, to guarantee the invariant of the topological structure when vertices acted by a force and getting back to the original state without external force.

Each constraint is a function of the vertices' position and the positions of its neighbors, which we refer to collectively as $\mathbf{p}_1,...,\mathbf{p}_n$. The vertices of deformable body need to satisfy the distance constraint:

$$C_l(\mathbf{p}_1,\mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2| - d \tag{4}$$

The angle constraint on the vertices of deformable body is given by:

$$C_a(\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4) = \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2) - \phi_0 \tag{5}$$

with $\mathbf{n}_1$, $\mathbf{n}_2$ the included angle of $\mathbf{p}_1,\mathbf{p}_2$, $\mathbf{p}_3,\mathbf{p}_4$:

$$\mathbf{n}_1 = \frac{(\mathbf{p}_2 - \mathbf{p}_1)\times(\mathbf{p}_3 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1)\times(\mathbf{p}_3 - \mathbf{p}_1)\|} \tag{6}$$

$$\mathbf{n}_2 = \frac{(\mathbf{p}_2 - \mathbf{p}_1)\times(\mathbf{p}_4 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1)\times(\mathbf{p}_4 - \mathbf{p}_1)\|} \tag{9}$$

To make the deformable body get back to the original state gradually, we need a vertices position correction $\Delta\mathbf{P}$ that satisfies the constraint:

$$C(\mathbf{P} + \Delta\mathbf{P}) \approx C(\mathbf{P}) + \nabla_p C(\mathbf{P}) \cdot \Delta\mathbf{P} = 0 \tag{7}$$

with $\mathbf{P} = (\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3...\mathbf{p}_n)$, n the number of vertices needed for constraint. In the last-written, we set $\Delta\mathbf{P}$ and $\nabla_p C(\mathbf{P})$ in the same direction, the equation have the following form:

$$\Delta\mathbf{P} = -\frac{C(\mathbf{P})}{|\nabla_{\mathrm{P}} C(\mathbf{P})|^2}\nabla_{\mathrm{P}} C(\mathbf{P}) \tag{8}$$

We determine an individual stiffness coefficient k to control the constraints. After n iterations, the effect of k is $\Delta\mathrm{P}\cdot k^n$ which is non-linear. So we use the following equations to constraint the iterations instead of coefficient k:

$$k' = 1-(1-k)^{1/n} \quad \Delta P(1-k')^n = \Delta P(1-k) \tag{9}$$

B. Adaptively time steps

There are everal time step constraints should be satisfied to ensure numerical stability and convergence. According to the Courant-Friedrich-Levy (CFL) condition:

$$\Delta t_{CFL} \le \lambda_v(\frac{h}{v_{\max}}) \tag{10}$$

where $v_{\max}$ is the maximum particle velocity in the computation. λ_v is a coefficient ensures that the particle moves only a fraction of the smoothing length every time step and is of order 0.1. Another constraint on the time-step comes from considering particles' maximum acceleration. Thus, the time-step is limited by the condition:

$$\Delta t_f \le \lambda_f(\sqrt{\frac{h}{f_{\max}}}) \tag{11}$$

where λ_f is a coefficient depending on the kernel type and the particle arrangement. λ_f is usually in the interval [0, 1]. $f_{max} = \max_i \|d\boldsymbol{v}_i / dt\|$ denotes the magnitude of the maximum force per unit mass for all particles throughout the simulation.

Since SPH uses a stiff EOS which restricts the time step, we take $\lambda_v = 0.1$ and $\lambda_f = 0.05$ for simulation. Instead of using a constant time step, we can now adjust it dynamically by

$$\Delta t_i \le \min_j\left(\lambda_v\frac{h}{\|\boldsymbol{v}_j\|}, \lambda_f\sqrt{\frac{h}{\|d\boldsymbol{v}_j / dt\|}}\right) \tag{12}$$

As stated above, the allowable time-step should be limited by equation (12). However, the maximum force is unknown before the iterative process and should be estimated depending on the initial setting of the scene. For the simulation with fast moving particles and non-moving objects, for example rigid-fluid coupling and deformable-fluid coupling, the global time step should be set as the minimum one which might be too restrictive for the main part of the simulation. So we propose an adaptive time-stepping scheme for deformable-fluid coupling, which smoothly adapts to the locally condition.

For Lagrangian fluids, since particles only interacts with their neighbors, every particle could have its own time step which is determined locally. To save the computing resources, the time step is determined by current minimize individual time step:

$$\Delta t_i = \min\left(\lambda_v\frac{h}{\|\boldsymbol{v}_i\|}, \lambda_f\sqrt{\frac{h}{\|d\boldsymbol{v}_i / dt\|}}\right) \tag{13}$$

Instead of the minimum of global time stepping, the average of individual time stepping is significantly larger, which make the simulation more stable and efficient.

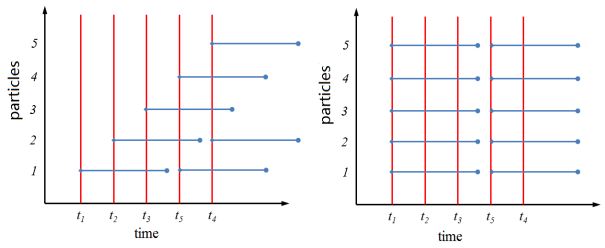
The proposed method varies the time step Δt_i for each particle. Besides, at every time step, only the active particles will be updated. For those inactive particles, their physical quantities are obtained by linear interpolation.

The particle i will be updated when it meets the condition:

$$t_i^{last} + \Delta t_i < t \tag{14}$$

where t_i^{last} iterates the last time particle i updated, t iterates the system time. According to Equation (14), when the system time is larger than the individual time step, the particle i will be set as active particle and be updated. In this setting, for inactive particles, their velocity and position are obtained from the interpolation of the last value directly, while the velocity

and position of active particles are updated as semi-implicit Euler numerical integration.



(a) **(b)**

Figure 2 (a) without synchronization (b) after synchronization

The proposed method adjusts the time step such that the total asynchronous time integration is implemented. In order to synchronize the animation every Δt, [8] restricted the time step Δt_i to $\Delta t / 2^q$ (q is an integer), which decreases the Δt_i. Simulation results show that, the synchronization of particles update will ensure higher numerical stability and resolve the density fluctuation, and then provide larger time step. Therefore, we propose to update particles which time step are approximately close at the same time.

The space-time continuum for time step should be considered, which means particles have close time step when they are adjacent and every particle has nearly the same time step in a while. This leaves us the following equation to synchronize time step:

$$\Delta t_i = \left\lfloor \frac{\Delta t_i}{\Delta t} \right\rfloor \tag{15}$$

where $\lfloor \cdot \rfloor$ is rounded down. In practice, we record the updated step number $K_{\Delta t}$, and set $K_{\Delta t} \bmod \left\lfloor \frac{\Delta t_i}{\Delta t} \right\rfloor = 0$ as update condition. This is shown in Fig.2.

C. The algorithm

Our adaptively time steps method for Lagrangian deformable-fluid coupling is outlined in Algorithm 1. As mentioned above, neighborhood searching (line 6-7) and forces evaluating (line 13-16) are performed only for activate particles.

Algorithm 1: Adaptively time steps for Lagrangian deformable-fluid coupling

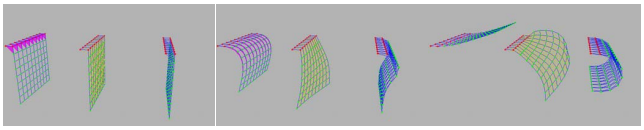
1	**while** *animating* **do**
2	**for all** *vertices a* **do**
3	initialize $\mathrm{x}_a = x_a^0, v_a = v_a^0, w_a = 1/m_a$
4	**end for**
5	select *active*
6	**for all** *active particle i* **do**
7	find neighbors *j*
8	**for all** *particle i* **do**
9	**if** *active* **then**
10	compute ρ_i , p_i
11	**else**
12	interpolate ρ_i , p_i by $\frac{d\rho_i\left(t_i^{last}\right)}{dt}$
13	**for all** *active particle i* **do**
14	compute $\frac{d\boldsymbol{v}_i(t)}{dt}$, $\frac{d\rho_i(t)}{dt}$
15	$\Delta t_i = \min\left(\lambda_v \frac{h}{\|\boldsymbol{v}_i\|}, \lambda_f \sqrt{\frac{h}{\|d\boldsymbol{v}_i / dt\|}}\right)$
16	$t_i^{last} = t$
17	**for all** *particle i* **do**
18	$\Delta t = \min_i(\Delta t_i)$
19	**for all** *vertices a* **do**
20	$v_a \leftarrow v_a + \Delta t w_a f_{ext}(x_a)$
21	$\mathrm{p}_a \leftarrow x_a + \Delta t v_a$
22	**loop** *solver Iteration times*
23	project constraint $(C_1, C_a, \mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3 ..., \mathrm{p}_n)$
24	**end loop**
25	**for all** *vertices a* **do**
26	$v_a \leftarrow (\mathrm{p}_a - x_a)/\Delta t$
27	$x_a \leftarrow \mathrm{p}_a$
28	**end for**
29	update velocity ($v_1, v_2, v_3, ..., v_n$)
30	**for all** *particle i* **do**
31	$\Delta t' = t + \Delta t - t_i^{last}$
32	$\boldsymbol{v}_i\left(t_i^{last} + \Delta t'\right) = \boldsymbol{v}_i\left(t_i^{last}\right) + \Delta t' \frac{d\boldsymbol{v}_i\left(t_i^{last}\right)}{dt}$
33	$\boldsymbol{x}_i\left(t_i^{last} + \Delta t'\right) = \boldsymbol{x}_i\left(t_i^{last}\right) + \Delta t' \boldsymbol{v}_i\left(t_i^{last} + \Delta t'\right)$
34	$t = t + \Delta t$

III. Results and Discussion

In this section, we show the capabilities of our approach. First, we show the effect of our deformable model with different value of k. Then, we discuss our adaptive time-stepping scheme with respect to performance, stability and influence on the simulation. Finally, our method is applied to deformable-fluid coupling scenario. All timings are given for an Intel 3.50 GHz CPU with 4 cores. The simulation software is parallelized with OpenMP. In the given scenarios, the density fluctuation is set to 0.01.

A. Deformable model

We applied this procedure to the case of the deformable sheet described in section 2.2. We set stiffness coefficient as different value to control the constraints. Figures 3 show the results we have obtained, which prove our method effectively.



(a) (b) (c)

Figure 3 Clothing simulation. From left to right, k is set as 0, 0.3 and1. (a) original state (b) the state after 5 frames (c) the state after 15 frames

B. *Adaptively time steps*

Our adaptive time-stepping method requires no manual setting of the time step, since it increases and decreases according to the current global state of the simulation. In order to analyze how the proportion of active particles changed with the process of simulation, we computed a clothing-fluid coupling with 211K particles using adaptive time-stepping.

The numerical result is shown in figure 4, where the red line denotes the global time step, blue line denotes the proportion of active particle. In this scene, when fluid particles in contact with clothing at the beginning of the simulation, the time step is rather short. In the second half of the simulation, as fluid particles verge to smooth and steady, the time step become larger gradually.

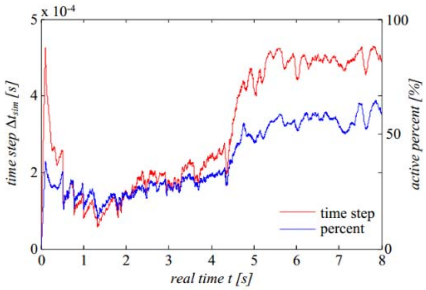


Figure 4 The active percent evolution for clothing-fluid coupling

C. *Performance and application*

We applied the proposed algorithm (see Alg. 1) to a scene where up to 211K particles pour into a clothing (see Fig. 5). In this scene, the fluid interacts realistically with the deformable object while comparatively large time steps could be used.

TABLE I. THE SIMULATION PARAMETER OF DEFORMABLE-FLUID COUPLING

Parameter	***Value***
the scale of simulation domain	$7m \times 7m \times 7m$
the number of fluid particles(average/ maximum)	105.6K / 211.2K
the number of boundary particles	25K
the smooth radii	0.1m
the area of clothing	$16\,m^2$

As shown in the picture, fluid particles wrinkle the clothing, while cloth make fluid particles splashing. After a while, some fluid particles are stored up on the clothing. In our simulation, fluid particles cannot penetrate the deformable sheet.

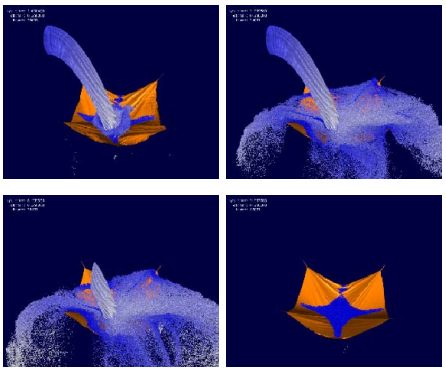


Figure 5 the simulation of deformable-fluid coupling before render

IV. CONCLUSIONS

We proposed a new deformable-fluid coupling scheme which takes the advantages of the position-based method. With the constraint to distance and angle, deformable sheet could be implemented and coupled with fluid particles. Furthermore, we suggested an adaptive time-stepping method which reduces the overall computation time for the simulation.

V. ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (No. 61272357, 61300074, 61572075).

VI. REFERENCES

[1] Carlson M, Mucha P J, Turk G. Rigid fluid: animating the interplay between rigid bodies and fluid[J]. Acm Transactions on Graphics, 2004, 23(3):págs. 375-382.

[2] Müller M, Schirm S, Teschner M, et al. Interaction of fluids with deformable solids[J]. Computer Animation & Virtual Worlds, 2004, 15(3-4):159-171.

[3] Harada T, Koshizuka S, Kawaguchi Y. Real-time Fluid Simulation Coupled with Cloth[J]. Ipsj Sig Notes, 2007, 2007(111):13-18.

[4] Seungtaik O, Younghee K, Byung Seok R. Impulse-based rigid body interaction in SPH[J]. Computer Animation & Virtual Worlds, 2009, 20(2-3):215-224.

[5] Ihmsen M, Akinci N, Gissler M, et al. Boundary Handling and Adaptive Time-stepping for PCISPH[C]// The Workshop on Virtual Reality Interactions and Physical Simulations, Vriphys 2010, Copenhagen, Denmark. 2010:79-88.

[6] Xiaojuan Ban, Xiaokun Wang, Liangliang He, et al. Adaptively stepped SPH for fluid animation based on asynchronous time integration[J]. Neural Computing and Applications, 2016.

[7] Markus, Ihmsen, et al. "Implicit incompressible SPH." IEEE Transactions on Visualization & Computer Graphics 20.3(2014):426-35.

[8] Desbrun M, Cani M P. Space-Time Adaptive Simulation of Highly Deformable Substances[C]// INRIA, BP 105 - 78153 Le Chesnay Cedex. 1999