2016 International Conference on Cyberworlds

Individual time-stepping for rigid-fluid coupling of particle based fluids

Ruixiang Li
Minzu University Of China
Beijing,China
ruixiang0822@163.com

XiaoKun Wang
University Science and Technology Beijing
Beijing,China
wang1xiao2kun3@163.com

***Abstract*—We propose an efficient rigid-fluid coupling method using individual time-stepping for particle-based fluid simulation. It updates neighbors and forces only for active particles. It allocates computing resources to complex regions and has an apparent speedup. We realize our method with several scenarios for rigid-fluid interaction. The experimental results demonstrate that our method is capable to implement interaction of rigid body and fluids while it also improves the efficiency.**

Keywords—Physically-basedsimulation;individual time stepping;rigid-fluid coupling

I. INTRODUCTION

Physically-based fluid simulation is a popular issue in computer animation, graphics while has a huge research and application demand in virtual reality, intelligent wearable and smart city. Two major schemes are employed for animating fluids, the grid-based Eulerian approach and particle-based Lagrangian approach. Eulerian method is particularly suited to simulate large volumes fluid, while is restricted by time step and computing time for small scale features. In contrast, Lagrangian method are suitable for capturing small scale effects such as spindrift, droplet. Among various particle-based approaches, Smoothed Particle Hydrodynamics (SPH) is the most commonly used method to simulate.

In reality, rigid-fluid interaction widely exists in many scenarios. As a result, the interesting phenomenon appears when rigid objects are added to fluid simulation. While the interaction between fluids and rigid objects seems to be straightforward, there are still several issues not well resolved. The computational expenses of rigid-fluid coupling are considerable. To handle the increasing needs for more detailed and high efficiency, we present individual time-stepping for rigid-fluid coupling.

II. RELATED WORK

Monaghan's simulating free surface flows with SPH [1] serves as a basis for SPH fluid simulation. Muller et al. [2]proposed using gas state equation for interactive applications, which also bring compressibility issue. Becker and Teschner[3] reduce compressibility with Tait Equation which is known as WCSPH. It significantly increased realistic effects while the efficiency is limited by time step. As incompressibility expenses computation time, many improved algorithms were addressed to enhance the efficiency. Solenthaler and Pajarola presented PCISPH [4] using a prediction-correction scheme to determine the particle pressures and large time steps which significantly improved efficiency.Afterwards, Ihmsen et al. addressed a more efficient approach IISPH [5]. Recently, a promising approach for impressible SPH has been proposed by Bender and Koschier[6].

Besides, adaptive method, either the spatial resolution or temporal sampling, is another way to promote efficiency. They allot computing resources to regions of complicated calculations. Space adaptive methods [7][8][9] adaptively sample particle and employ fewer particles to construct similar details. However, difficulties exist in reproducing quantity, while neighbor searching is usually the bottleneck. As an alternative to adaptive approaches, the time domain can be adaptively sampled as well. Globally adaptive methods [10][11][12] employ only one time step to adjust each step [10] for all particles. Although the particles have the current minimumtime step, it is not the most effective pattern. Locally adaptive methods [7][13][14] use different time steps for particles. He et al. [15] adopt the idea of [7] to make the simulation of stiff fluids stable and efficient. In this paper, we extend it to rigid-fluid coupling to reduce the computing time.

For boundary handling and rigid-fluid coupling, penalty methods have been commonly used [16][17]. However, these methods require large penalty forces which limit the time step, while particles stick to the boundary due to lack of fluid neighbors. The sticking is avoided with frozen and ghost particles based models [18]. In order to ensure non-penetration, the positions of penetrating particles are corrected [19]. However, dealing with two-way interaction is questionable because the estimated density on one side of the boundary influences the fluid particles on the other side. For this reason, Ghost SPH scheme [20] resolves this with a narrow layer of ghost particles and Akinci et al. employed boundary particles to correct the calculation of fluid density [21].Since Ghost SPH is more time consuming, we apply Akinci's boundary handling method to individual time-stepping that receive an efficient rigid-fluid coupling pattern.

III. PARTICLE-BASED FLUID SIMULATION FRAMEWORK

In the Lagrangian description, Navier-Stokes equations for fluids can be expressed as

$$\frac{d\rho_i}{dt} = -\rho_i \nabla \cdot \boldsymbol{v}_i \tag{1}$$

$$\rho_i \frac{D\mathbf{v}_i}{Dt} = -\nabla p_i + \rho_i g + \mu \nabla^2 \mathbf{v}_i \tag{2}$$

Where $\mathbf{v}_i$ is velocity, ρ_i is density, p_i is pressure, μ is viscosity coefficient, g is external force.

SPH is a numerical interpolation method [2], for particle i at location x_i ,

978-1-5090-2303-5/16 $31.00 © 2016 IEEE
DOI 10.1109/CW.2016.48

$$\langle A(\boldsymbol{x}_i)\rangle = \sum_j m_j \frac{A_j}{\rho_j} W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) \tag{3}$$

Where m_j is mass and ρ_j is density, $W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right)$ is smoothing kernel and h is smoothing radius.

Substituting (3) to the density of particle i

$$\rho_i = \sum_j m_j W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) \tag{4}$$

Forces between particles including pressure $\mathbf{f}_i^P$ and viscous force $\mathbf{f}_i^{\mathrm{v}}$, they can be expressed as

$$\mathbf{f}_i^P = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) \nabla W_{ij} \tag{5}$$

$$\mathbf{f}_i^{\mathrm{v}} = \mu \sum_j m_j \frac{\mathbf{v}_{ji}}{\rho_j} \nabla^2 W_{ij} \tag{6}$$

we use Tait equation [3] to calculate the pressure and use the method in [21] to compute viscous force.

IV. Boundary Handling for Particle-Based Fluids

For boundary handling, we implement our simulation based on the work of [21]. The following briefly introducesthis work.

Considering influence of boundary particles, density formula in (4) need to introduce weighted summation influence of boundary particle [21], that is

$$\rho_{f_i} = \sum_j m_{f_j} W_{ij} + \sum_k m_{b_k} W_{ik} \tag{7}$$

Where f_j, b_k denotes fluid particle j and boundary particle k respectively. This formula can overcome the problem of boundary defects in SPH fluid simulation to some extent.

The density of fluid particles is incorrect and instability when the setting of boundary particle mass is unreasonable or distribution of boundary particles is uneven. Thus, using the contribution of boundary particles' volume to a fluid particle as

$$\Psi_{b_i}(\rho_0) = \rho_0 V_{b_i} \tag{8}$$

with ρ_0 is the rest density of fluid, V_{b_i} is the estimation volume of boundary particles. Applying $\Psi_{b_i}(\rho_0)$ replace the boundary particle mass can guarantee the stability.

Therefore, equation (7) can be written as

$$\rho_{f_i} = \sum_j m_{f_j} W_{ij} + \sum_k \Psi_{b_k}(\rho_{0i}) W_{ik} \tag{9}$$

The pressure acceleration generated by boundary particles to fluid particles can be computed as

$$\frac{d\boldsymbol{v}_{f_i}}{dt} = -\frac{kp_{f_i}}{\rho_{f_i}^2} \sum_k \Psi_{b_k}(\rho_{0i}) \nabla W_{ik} \tag{10}$$

where $p_{f_i} > 0$ takes $k = 2$. When $p_{f_i} < 0$, boundary particles and fluid particles attract each other, then we can adjust parameter k ($0 \le k \le 2$) to realize different adsorption effects.

The calculation for friction consults from artificial viscosity, that is

$$\frac{d\boldsymbol{v}_{f_i}}{dt} = -\sum_k \Psi_{b_k}(\rho_{0i}) \Pi_{ik} \nabla W_{ik} \tag{11}$$

Where $\Pi_{ik} = -\nu \left(\dfrac{\boldsymbol{v}_{ik}^T \boldsymbol{x}_{ik}}{\boldsymbol{x}_{ik}^2 + \varepsilon h^2} \right)$, $\nu = \dfrac{2\alpha h c_s}{\rho_k + \rho_j}$.

Then we can get the forces of boundary particles using Newton's third law. The forces generated by fluid particles to boundary particles is

$$\boldsymbol{F}_{b_k} = \sum_i \left(\frac{kp_{f_i}}{\rho_{f_i}^2} + \Pi_{ik} \right) m_{f_i} \Psi_{b_k}(\rho_{0i}) \nabla W_{ik} \tag{12}$$

where i denotes the fluid neighbors of boundary particle k. It is the counter-acting force of Eq.10 and Eq.11.

For a rigid body, the total force and torque need to be calculated. It can be separately written as

$$\boldsymbol{F}_{rigid} = \sum_k \boldsymbol{F}_{b_k} \tag{13}$$

$$\tau_{rigid} = \sum_k \left(\boldsymbol{x}_k - \boldsymbol{x}_{rigid}^{cm} \right) \times \boldsymbol{F}_{b_k} \tag{14}$$

Where $\boldsymbol{x}_k$ denotes the location of boundary particle k, $\boldsymbol{x}_{rigid}^{cm}$ is the mass center of a rigid body. The total force and torque will be transmitted to the physics engine to handle the motion of rigid bodies.

V. Individual time-stepping for rigid-fluid coupling

In this section, we propose a rigid-fluid coupling method employing individual time stepping. Firstly, the time step of simulation must satisfies numerical stability and convergence constraint. The Courant-Friedrich-Levy (CFL) condition is

$$\Delta t_{CFL} \le \lambda_v \left(\frac{h}{v_{max}} \right) \tag{17}$$

where $v_{max} = \max_i \|\boldsymbol{v}_i\|$ is the maximum velocities of particle, coefficient $\lambda_v < 1$. Besides, it also has to consider particles' maximum acceleration. Thus, it must also meet the condition

$$\Delta t_f \le \lambda_f \left(\frac{h}{f_{max}} \right) \tag{18}$$

Where $f_{max} = \max_i \left\| \dfrac{d\boldsymbol{v}_i}{dt} \right\|$ is the maximum force of particles, $\lambda_f < 1$. Rather than using constant time step, we adjust time step dynamically as

$$\Delta t \le \min\left(\Delta t_{CFL}, \Delta t_f \right) \tag{19}$$

Thus, we computes time step for each particle i as

$$\Delta t_i = \min_j \left(\lambda_v \frac{h}{\|\mathbf{v}_j\|}, \lambda_f \sqrt{\frac{h}{\|d\mathbf{v}_j / dt\|}} \right) \tag{20}$$

Compared to globally adaptive method, we consider the neighbors, which makes the algorithm stable for stiff fluids. However, the method needs small parameters for the asynchrony mechanism, so we choose $\lambda_v = 0.05$,$\lambda_f = 0.025$.

Since each particle has its own time step, we need to implement asynchronous time integration for the algorithm which is

$$\Delta t = \min_i(\Delta t_i) \tag{21}$$

where Δt_i is computed by Eq.(20).

Particle i will be updated if it satisfies the condition

$$t_i^{last} + \Delta t_i < t_{sim} \tag{22}$$

where t_i^{last} is the last updated time of particle i , t_{sim} iterates the system time. If system time is larger than individual time step, particle i will be set to active particle and updated.

For asynchronous update, the semi-implicit Euler numerical integrations have the following form:

$$\boldsymbol{v}_i\left(t_i^{last} + \Delta t'\right) = \boldsymbol{v}_i\left(t_i^{last}\right) + \Delta t'\, \boldsymbol{a}_i\left(t_i^{last}\right) \tag{23}$$

$$\boldsymbol{x}_i\left(t_i^{last} + \Delta t'\right) = \boldsymbol{x}_i\left(t_i^{last}\right) + \Delta t'\, \boldsymbol{v}_i\left(t_i^{last} + \Delta t'\right) \tag{24}$$

where $\Delta t'$ is an independent integral time step. For non-active particles, the operation is interpolating, while for active particles, it is a semi-implicit Euler numerical integration.

The individual time stepping for rigid-fluid coupling algorithm is illustrated below. In this algorithm, each particle i has a few additional variables, where $d\rho_i(t)/dt$ is density derivative, Δt_i is time step, $\Delta t_i'$ is individual condition time step, t_i^{last} is last updated time. Besides, t is system time, Δt is system update time step. Particle i is active if $t_i^{last} + \Delta t_i < t$.

Algorithm 1 Individual time stepping for rigid-fluid coupling

```
while animating do
    select active
    for each active fluid particle $i$ do
        find fluid and boundary neighbors
    for each fluid particle $i$ do
        if active then
            compute $\rho_i(t)$ , $p_i(t)$
        else
            interpolate $\rho_i(t)$ , $p_i(t)$ using $d\rho_i(t_i^{last})/dt$
    for each active fluid particle $i$ do
        compute $d\boldsymbol{v}_i(t)/dt$ , $d\rho_i(t)/dt$
        compute time step condition $\Delta t_i'$ (Eq.18)
        $t_i^{last} = t$
    for each boundary particle $k$ do
        compute forces (Eq.12)
    for each fluid particle $i$ do
        compute time step $\Delta t_i = \min_j(\Delta t_i')$
    $\Delta t = \min_i(\Delta t_i)$
    for each rigid body do
        compute total forces, torques (Eq.13, Eq14)
        update rigid body
        update boundary particles of rigid body
    for each fluid particle $i$ do
        $\Delta t' = t + \Delta t - t_i^{last}$
        $\boldsymbol{v}_i(t_i^{last} + \Delta t') = \boldsymbol{v}_i(t_i^{last}) + \Delta t'\, d\boldsymbol{v}_i(t_i^{last})/dt$
        $\boldsymbol{x}_i(t_i^{last} + \Delta t') = \boldsymbol{x}_i(t_i^{last}) + \Delta t'\, \boldsymbol{v}_i(t_i^{last} + \Delta t')$
    $t = t + \Delta t$
```

VI. IMPLEMENTATION AND RESULTS

The simulation is performed on an Intel 3.50GHz CPU with 4 cores. The bullet is used to simulate the rigid objects and OpenMP is used for parallel computation. We reconstruct fluid surface using anisotropic kernels [22]. Images were rendered with Blender.

In order to analysis the proposed algorithm, we compare individual stepping method to globally adaptive stepping and constant stepping method in breaking dam with obstacles scene which used 153K fluid particles respectively. The rending results are showed in fig.5 and the time statistics is listed in TABLE I. From fig.5, the fluid simulation results are almost no differences using three methods. While in TABLE I. , we can find that our method gains 1.5 and 6.4 times speedup comparing to globally adaptive stepping method and constant stepping method respectively. In addition, the average active particles percent of individual stepping method is 31%.

TABLE I. COMPARISON RESULTS OF BREAKING DAM WITH OBSTACLES

method	Total comp. time	avg. Δt (avg. active pct)	speed up
constant steps	175min	0.11ms	-
globally adaptive	41min	0.46ms	-
individual stepping	27min	0.23ms(31%)	1.5(6.4)

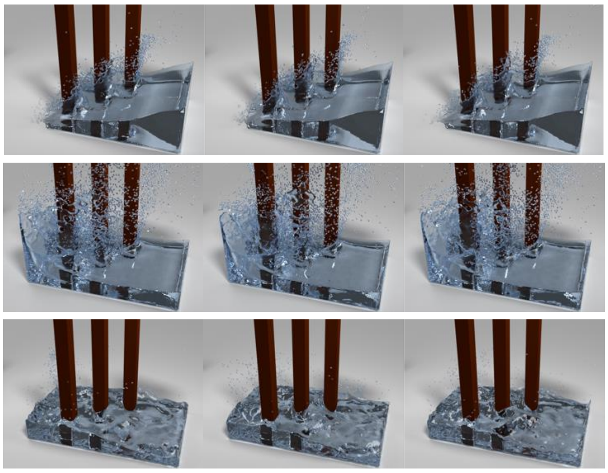

Figure 1. Rendering results of breaking dam with obstacles
Left: Individual time stepping method; Middle: Globally adaptive stepping method; Right: Constant stepping method

In order to demonstrate the validity of our method, we designed a scene of dropping 16 small squares into water using 320k fluid particles. The experimental results are displayed in Fig. 7. It can be seen fromthe diagram that small squares fall into water and splash water while they are rotated and inclined by water. Finally, small squares are force balance and floating on the water.This experiment proved our method can implement vivid fluid-rigid coupling animation simulation with high realistic effects.

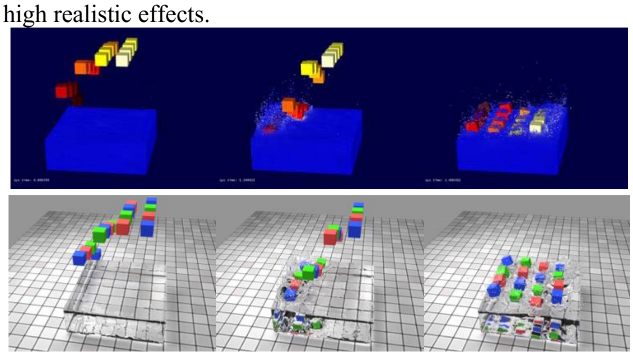

Figure 2. 16 cubes fall into water.
Top:simulation in particle view;Bottom: rendering results.

VII. Conclusion

We proposed an efficient and simple rigid-fluid coupling scheme for particle-based fluid simulation. It updates neighbors and forces of particles only needed and it naturally allocates computing resources to complex regions while obtains an obviously speedup compared to previous method. The scheme was integrated to rigid-body coupling simulation which has a good sense of visual reality. Overall, our method is efficient to compute while the coupling algorithm can be applied to other particle-based simulation or relevant approaches. Future work would be extending the proposed method to IISPH[5] or DFSPH[6] as well as large-scale scenarios.

References

[1] Monaghan J J. Simulating free surface flows with SPH[J]. Journal of computational physics, 1994, 110(2): 399-406.

[2] Muller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications[C]//Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2003: 154-159.

[3] Becker M, Teschner M. Weakly compressible SPH for free surface flows[C]//Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2007: 209-217.

[4] Solenthaler B, Pajarola R. Predictive-corrective incompressible SPH[C]//ACM transactions on graphics (TOG). ACM, 2009, 28(3): 40.

[5] Ihmsen M, Cornelis J, Solenthaler B, et al. Implicit Incompressible SPH[J]. Visualization and Computer Graphics, IEEE Transactions on, 2014,20(3):426-435.

[6] Bender J, Koschier D. Divergence-free smoothed particle hydrodynamics[C]// ACM SIGGRAPH / Eurographics Symposium on Computer Animation. ACM, 2015:147-155.

[7] Desbrun M, Cani M-P (1999) Space-time adaptive simulation of highly deformable substances. Technical Report 3829, INRIA

[8] Adams B, Pauly M, Keiser R, Guibas LJ (2007) Adaptively sampled particle fluids. ACM Trans Graph Proc SIGGRAPH 26(3):48

[9] Solenthaler B, Gross M (2011) Two-scale particle simulation. ACM Trans Graph Proc SIGGRAPH 30(4):81:1–81:8

[10] Monaghan JJ (1992) Smoothed particle hydrodynamics. Ann Rev Astron Astrophys 30:543–574

[11] Desbrun M, Cani M-P (1996) Smoothed particles: a new paradigm for animating highly deformable bodies. In: Eurographics workshop on computer animation and simulation (EGCAS), pp 61–67. Springer, Berlin

[12] Ihmsen M, Akinci N, Gissler M, Teschner M (2010) Boundary handling and adaptive time-stepping for PCISPH. In: Workshop on virtual reality interaction and physical simulation, pp 79–88. The Eurographics Association

[13] Goswami P, Batty C (2014) Regional time stepping for SPH. In: Eurographics, pp 45–48. The Eurographics Association

[14] Goswami P, Pajarola R (2011) Time adaptive approximate SPH. In: Proceedings of the 8th workshop on virtual reality interaction and physical simulation, pp 19–28. VRIPHYS

[15] Liangliang He, Xiaojuan Ban, Xu Liu, Xiaokun Wang. Individual Time Stepping for SPH Fluids. In: Eurographics2015. The Eurographics Association

[16] Harada T, Koshizuka S and Kawaguchi Y. 2007. Smoothed particle hydrodynamics on GPUs. In Proc. of Computer Graphics International, 63–70.

[17] Monaghan J J, and Kajtar J. 2009. SPH particle boundary forces for arbitrary boundaries. Computer Physics Communications 180, 10, 1811–1820.

[18] Hu X and Adams N. 2006. A multi-phase SPH method for macroscopic and mesoscopic flows. Journal of Computational Physics 213, 2, 844–861.

[19] Ihmsen M, Akinci N, Gissler M and Teschner M. 2010. Boundary handling and adaptive time-stepping for PCISPH. In Proc. of VRIPHYS, 79–88.

[20] Schechter H, Bridson R. Ghost SPH for Animating Water[J]. ACM Transactions on Graphics, 2012, 31(4): 61:1–61:8.

[21] Akinci N, Ihmsen M, Akinci G, et al. Versatile rigid-fluid coupling for incompressible SPH[J]. ACM Transactions on Graphics (TOG), 2012, 31(4): 62.

[22] Wang X K, Ban X J, Liu X, et al. Efficient extracting surfaces approach employing anisotropic kernels for SPH fluids. Journal of Visualization, 2016, 19(2): 301-317