Technical Report, Department of Informatics, University of Zurich, December 17, 2009

Performance Comparison of Parallel PCISPH and WCSPH

B. Solenthaler † and R. Pajarola ‡

University of Zurich

Abstract

In this report we present performance measurements of our parallel predictive-corrective incompressible SPH (PCISPH) implementation. PCISPH is an incompressible fluid simulation method based on the Lagrangian SPH model. In this method, incompressibility is enforced by using a prediction-correction scheme to determine the particle pressures. The information about density fluctuations is actively propagated through the fluid and pressure values are updated until the targeted density is satisfied. With this approach, the computational expenses of solving a pressure Poisson equation is avoided while still being able to use large time steps in the simulation. The results of our implementation show that the multi-threaded execution of both methods leads to another increase in speed-up of PCISPH over WCSPH of up to 40% compared to the single-threaded versions. We furthermore demonstrate that PCISPH can be applied to simulate compressible fluids, still outperforming a standard SPH solver with equal density fluctuations.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object ModelingPhysically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and RealismAnimation.

1. Introduction and Previous Work

The advantage of particles is the natural ability to handle free fluid surfaces and interfaces, splashes and droplets, as well as the interaction with highly complex boundaries and solid objects [MCG03, MSKG05, KAG*05, SSP07, SP08, BTT09]. However, particle approaches have difficulties to satisfy the incompressibility condition of liquids such as water, since the fluid is typically approximated by a compressible fluid where pressures are determined by an equation of state (EOS) [Mon94, BT07]. This problematic issue, and the difficulties to reconstruct smooth surfaces out of particles, are the main reasons why grid methods are currently favored in graphics. However, particles and point representations have gained increasing attention in recent years, both for physics-based animations and rendering. Their simplicity and flexibility render a fluid solver into a powerful tool to simulate all kinds of phenomena such as water, smoke and fire, as well as complex interaction effects between multiple fluids and solid objects.

In the standard SPH model (e.g. [Mon92, MCG03, MSKG05]) pressures are computed using a soft EOS resulting in undesired compression artifacts. Although incompressibility can be enforced, it represents the most expensive part of the whole simulation process and thus renders particle methods less attractive for high quality and photo-realistic animations of water. In the literature, two different strategies have been pursued to model incompressibility. First, the weakly compressible SPH (WCSPH) method has been used where pressure is modeled using a stiff equation of state (e.g. [Mon05, BT07, BTT09]), and second, incompressibility has been achieved by solving a pressure Poisson equation (ISPH methods, e.g. [CR99, Sha06, HA07]) similar to Eulerian fluid solvers. Although both methods satisfy incompressibility, the computational expenses of simulating high resolution fluid animations are too large for practical use (Figure 1). The drawback of WCSPH is the severe time step restriction since the stiffness of the fluid typically dominates the Courant-Friedrichs-Levy (CFL) condition. Thus the computational cost increases with decreasing compressibility. Although ISPH allows larger time steps, the computational cost per physics step is immensely higher. Furthermore, the complexity to formulate and solve the equation system on unstructured particle configurations represents a major problem and is computationally expensive.

Recently, [SP09] proposed a novel, incompressible SPH model (PCISPH) where incompressibility is enforced by us-

† e-mail: solenthaler@ifi.uzh.ch

‡ e-mail: pajarola@acm.org

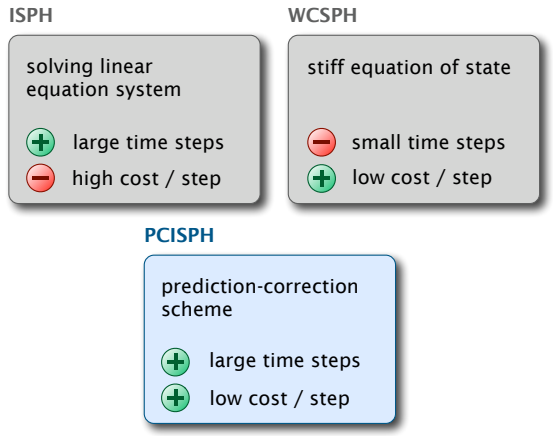

Figure 1: *Comparison of different incompressible SPH methods. The predictive-corrective incompressible SPH (PCISPH) method features the advantages of both ISPH and WCSPH: large time steps and low computational cost per physics update step.*

Algorithm 1 PCISPH

while animating **do**
  for all i **do**
    find neighborhoods $N_i(t)$
  for all i **do**
    compute forces $\mathbf{F}^{v,g,ext}(t)$
    initialize pressure $p(t) = 0.0$
    initialize pressure force $\mathbf{F}^p(t) = 0.0$
  while $(\rho^*_{err}(t+1) > \eta) \parallel (iter < minIterations)$ **do**
    for all i **do**
      predict velocity $\mathbf{v}^*_i(t+1)$
      predict position $\mathbf{x}^*_i(t+1)$
    for all i **do**
      predict density $\rho^*_i(t+1)$
      predict density variation $\rho^*_{err}(t+1)$
      update pressure $p_i(t) += f(\rho^*_{err}(t+1))$
    for all i **do**
      compute pressure force $\mathbf{F}^p(t)$
  for all i **do**
    compute new velocity $\mathbf{v}_i(t+1)$
    compute new position $\mathbf{x}_i(t+1)$

ing a prediction-correction scheme to determine the particle pressures. For this, the information about density fluctuations is actively propagated through the fluid and pressure values are updated until the targeted density is satisfied. With this approach, the computational expenses of solving a pressure Poisson equation are avoided, while still being able to use large time steps in the simulation. The results show that the PCISPH method clearly outperforms the commonly used WCSPH model by more than one order of magnitude while the computations are in good agreement with the WCSPH results.

1.1. Contribution

With PCISPH, one of the major disadvantage of particle systems, which is the efficient solution of the incompressibility condition, has been improved. In this report, we use a multi-threaded PCISPH implementation and show new performance data which highlight again the strength of the PCISPH model.

We show that when executing both PCISPH and WCSPH with multiple threads the speed-up of PCISPH over WCSPH can be increased by up to 40%. This results in an overall speed-up of up to a factor of 77 for the particular simulation example used in [SP09]. Furthermore, the incompressible PCISPH method does not only outperform WCSPH but it is also equally fast as the compressible SPH method which is typically used in the literature. This report shows additionally that PCISPH can also be applied to compressible fluids and still outperforms a standard compressible SPH solver with equal density variations.

2. Parallel PCISPH

Our simulation uses the PCISPH method as described in [SP09]. The algorithm is illustrated in Algorithm 1. In our implementation, we execute PCISPH in parallel with multiple threads by using *Open Multi-Processing* (OpenMP). OpenMP is an implementation of multi-threading where the code that is meant to run in parallel is marked accordingly. Since each *for*-loop in Algorithm 1 can be executed independently for each particle, we can create n threads running on n cores, executing the particular code segments in parallel. We have applied the same multi-threading technique to SPH and WCSPH as well.

As it has been shown in [SP09], the convergence loop represents the computationally most expensive part of PCISPH since time integration, densities and forces have to be computed several times per simulation update step until the predicted maximal density variation η is below a user-defined threshold. Each of these steps depend on each other, however, the individual steps themselves can be computed in parallel for each particle. This results in a faster execution of the convergence loop within each physics update step.

3. Results

We used the same simulation scene setup and identical parameter values as in [SP09] to compare the simulation times of PCISPH and WCSPH (Figure 2). We have implemented single- and multi-threaded versions of both methods to compare their performances. The measurements and simulation data are summarized in Table 1. All timings are given for an Intel Core2 Quad processor 2.66 GHz.

3.1. Scene Setup

We executed different simulation runs with varying particle resolutions (10k and 100k) and varying error threshold η (1%, 0.1%, and 7%) which defines the maximally allowed density fluctuation from the reference density. The 10k and 100k examples have corresponding scene setups but different fluid discretizations. This means that a particle in the 10k example represents a larger fluid volume than one in the 100k example. The setup and the parameter setting is described in detail in [SP09].

3.2. Performance

As shown in [SP09], the single-threaded implementation results in an overall speed-up of PCISPH over WCSPH of a factor of 15 and 16 for $\eta = 1\%$ and 55 for $\eta = 0.1\%$, respectively.

We run the same simulation examples but using our parallel implementations of both methods. In this case, PCISPH reaches a speed-up over WCSPH of a factor of 21 and 20 for $\eta = 1\%$ and 77 for $\eta = 0.1\%$, respectively (Table 1, Figure 3). These timings show that the parallel execution increases the speed-up by up to 40% compared to the single-threaded version. This is mainly due to the faster execution of the convergence loop which represents the most time-consuming part compared to standard SPH.

The PCISPH method can also be used to simulate compressible fluids by allowing larger density variations in the system. This can be easily achieved by setting the maximally allowed density variation η to a higher value. In a standard SPH solver, a stiffness value k of 1000 is typically used. In our particular simulation example, this stiffness results in a density variation of 7%. We can allow the same density fluctuation in PCISPH by setting η to 7%. In this case, PCISPH still outperforms standard SPH by more than a factor of 3. We can conclude that PCISPH always outperforms an SPH / WCSPH solver where an identical compressibility of the fluid is enforced.

Furthermore, the performance measurements show that the simulation cost of PCISPH with 1% density variation equals the cost of the compressible SPH method where a stiffness value of $k = 1000$ is used. Both methods need an overall computation time of approximately 4min.

4. Conclusion

We presented new performance comparisons of PCISPH and WCSPH where both methods are executed in parallel by using OpenMP's multi-threading capability. We have shown that the parallel execution of PCISPH leads to a speed-up over parallel WCSPH of a factor of up to 77 in the particular simulation example used in this report. Even when applying PCISPH to compressible fluids it reaches a speed-up over SPH of more than a factor of 3. In all our test examples, PCISPH outperformed SPH and WCSPH regardless of the simulated compressibility of the fluid. However, the limitations described in [SP09] remain and have to be addressed in subsequent work.

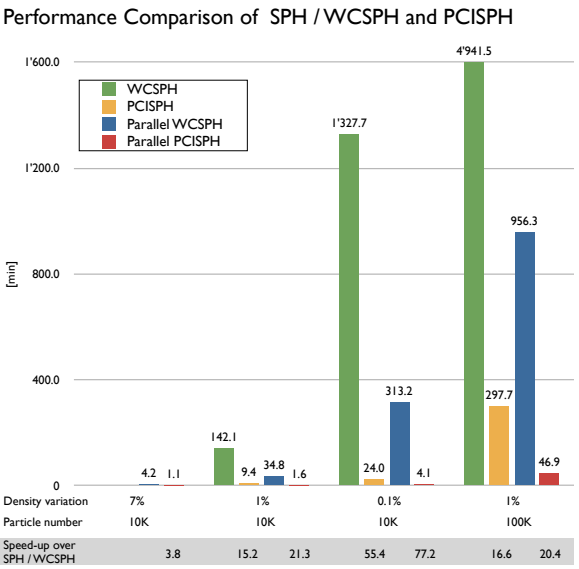

Figure 3: *Graphical illustration of the results presented in Table 1. The parallel execution of both methods with 4 threads increases the speed-up of PCISPH over WCSPH of another 40% approximately. With the parallel PCISPH method, a speed-up over parallel WCSPH of a factor of up to 77 can be reached.*

References

[BT07] BECKER M., TESCHNER M.: Weakly compressible SPH for free surface flows. In *Proceedings of the ACM Siggraph/Eurographics Symposium on Computer Animation* (2007), pp. 209–217.

[BTT09] BECKER M., TESSENDORF H., TESCHNER M.: Direct forcing for Lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (2009), 493–503.

[CR99] CUMMINS S. J., RUDMAN M.: An SPH projection method. *J. Comput. Phys. 152*, 2 (1999), 584–607.

[HA07] HU X. Y., ADAMS N. A.: An incompressible multi-phase SPH method. *J. Comput. Phys. 227*, 1 (2007), 264–278.

[KAG*05] KEISER R., ADAMS B., GASSER D., BAZZI P., DUTRE P., GROSS M.: A unified Lagrangian approach to solid-fluid animation. In *Proceedings of the Eurographics Symposium on Point-Based Graphics* (2005), pp. 125–133.

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the ACM Siggraph/Eurographics Symposium on Computer Animation* (2003), pp. 154–159.

[Mon92] MONAGHAN J. J.: Smoothed Particle Hydrodynamics. *Annu. Rev. Astron. Physics 30* (1992), 543.

[Mon94] MONAGHAN J. J.: Simulating free surface flows with SPH. *J. Comput. Phys. 110* (1994), 399–406.

[Mon05] MONAGHAN J. J.: Smoothed Particle Hydrodynamics. *Rep. Prog. Phys. 68* (2005), 1703–1759.

[MSKG05] MÜLLER M., SOLENTHALER B., KEISER R., GROSS M.: Particle-based fluid-fluid interaction. In *Proceedings of the ACM Siggraph/Eurographics Symposium on Computer Animation* (2005), pp. 237–244.

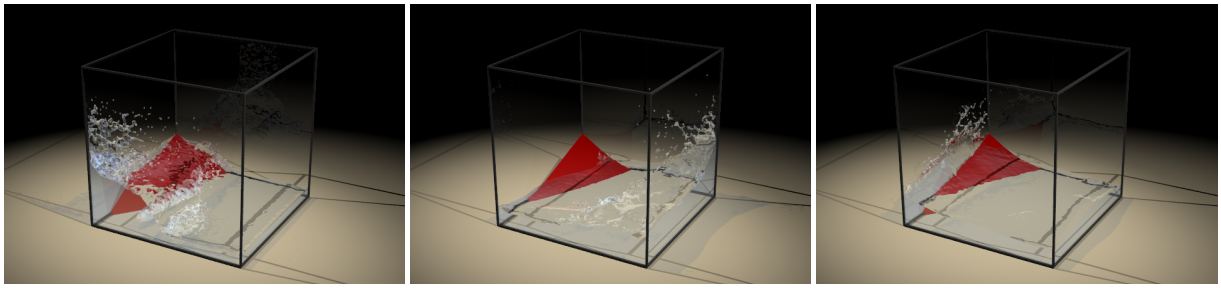

Figure 2: *Simulation scene used for the performance measurements shown in Table 1. These images have been presented in [SP09].*

Model	η [%]	#p	k	Δt [s]	t_{sim} [min]	speed-up	t_{sim} [min] (4 cores)	speed-up (4 cores)
WCSPH	1.0	10k	$7{\cdot}10^4$	3.78e-5	142.1	-	34.8	-
PCISPH	1.0	10k	-	0.0013	9.37	15.2	1.6	21.3
WCSPH	1.0	100k	$7{\cdot}10^4$	1.78e-5	4941.5	-	956.3	-
PCISPH	1.0	100k	-	0.00062	297.7	16.6	46.9	20.4
WCSPH	0.1	10k	$6{\cdot}10^6$	4.08e-6	1327.7	-	313.2	-
PCISPH	0.1	10k	-	0.00062	24	55.4	4.1	77.2
SPH	7.0	10k	10^3	0.0003			4.2	-
PCISPH	7.0	10k	-	0.0017			1.1	3.8

Table 1: *Speed-up comparison of PCISPH over WCSPH. We have run both methods on one single core and 4 cores using OpenMP. We used identical parameter values as in [SP09]. Compared to the single-core implementation, the parallel execution of both methods increases the speed-up of PCISPH over WCSPH of up to 40%. When executing both methods in parallel, PCISPH reaches a speed-up of a factor of 21 and 20 over WCSPH in the case of* η*=1% density variation, and in the case of* η*=0.1% a speed-up factor of 77 is achieved.*

[Sha06] SHAO S.: Incompressible SPH simulation of wave breaking and overtopping with turbulence modelling. *Int. J. Numer. Meth. Fluids 50* (2006), 597–621.

[SP08] SOLENTHALER B., PAJAROLA R.: Density contrast SPH interfaces. In *Proceedings of the ACM Siggraph/Eurographics Symposium on Computer Animation* (2008), pp. 211–218.

[SP09] SOLENTHALER B., PAJAROLA R.: Predictive-corrective incompressible SPH. *ACM Trans. Graph. (Proceedings of SIGGRAPH) 28*, 3 (2009), 40:1–40:6.

[SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid-solid interactions. *Journal of Computer Animation and Virtual Worlds 18*, 1 (2007), 69–82.