Development and Astrophysical Applications of a Parallel Smoothed Particle Hydrodynamics Code with MPI

Stefan Kunze, Erik Schnetter, and Roland Speith

Institut für Astronomie und Astrophysik, Abt. Theoretische Astrophysik
Universität Tübingen
Auf der Morgenstelle 10, 72076 Tübingen
E-mail: {kunze,schnetter,speith}@tat.physik.uni-tuebingen.de
Internet: http://www.tat.physik.uni-tuebingen.de/~speith/particle.html

Abstract. Smoothed Particle Hydrodynamics (SPH) is a particle method to simulate compressible fluids. First we present a brief introduction to SPH, where we focus on an approach to treat the physical viscosity. Then we describe in detail the basic principles of our parallel implementation of the SPH method. The efficiency of the code on Cray T3E and IBM SP2 is discussed. In the last part we present a short introduction to accretion disks in interacting binary stars and report on some new results in that field achieved with our code.

1 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a grid-free Lagrangian method for solving the hydrodynamic equations numerically. It was introduced in 1977 by Gingold & Monaghan [2] and by Lucy [3]. The method is especially suited for compressible flows involving free boundaries, a commonplace situation in astrophysics. In SPH the matter distribution is divided into small overlapping mass packets, so-called *particles*, which do not exchange matter. The particles move in space following the motion of the fluid while interacting with their neighbor particles.

1.1 Basic Principles

The system of hydrodynamic equations, i. e., the Navier-Stokes equation, the equation of continuity, and the energy equation together with an equation of state, form a system of coupled partial differential equations. Applying the SPH formalism, these are transformed into a set of ordinary differential equations (ODEs) in two steps. These ODEs are subsequently integrated numerically using standard methods. First the field quantities $f(\boldsymbol{r})$ are *smoothed*, i. e., they are replaced by the convolution

$$f(\boldsymbol{r}) \longrightarrow \int f(\boldsymbol{r}')\, W(|\boldsymbol{r}-\boldsymbol{r}'|, h)\, dV' \tag{1}$$

with a differentiable kernel W which is normalized to unity. The smoothing length h is a measure of the (compact) support of the kernel function, leading to

$$f(\boldsymbol{r}) = \int f(\boldsymbol{r}')\, W(|\boldsymbol{r} - \boldsymbol{r}'|, h)\, dV' + O(h^2) \quad . \tag{2}$$

A common choice for the kernel is a cubic spline (see, e. g., Monaghan [4]).

In the second step the above integrals are evaluated at the particle positions $\boldsymbol{r}_i$, and are approximated by sums

$$f_i = f(\boldsymbol{r}_i) \approx \sum_j V_j f(\boldsymbol{r}_j) W(|\boldsymbol{r}_i - \boldsymbol{r}_j|, h) \quad . \tag{3}$$

The quantity V_j describes the volume that is represented by the particle j. Spatial derivatives of field quantities at the particle positions can be approximated by

$$\nabla_i f(\boldsymbol{r}_i) \approx \sum_j V_j f(\boldsymbol{r}_j) \nabla_i W(|\boldsymbol{r}_i - \boldsymbol{r}_j|, h) \quad . \tag{4}$$

Typical particle quantities are the position $\boldsymbol{r}_i$, the velocity $\boldsymbol{v}_i$, the particle mass $m_i = V_i \varrho_i$, where ϱ_i is the mass density, the internal energy e_i, etc.

1.2 SPH Representation of the Hydrodynamic Equations

The particles follow the motion of the fluid ($d\boldsymbol{r}_i/dt = \boldsymbol{v}_i$) and do not exchange matter ($dm_i/dt = 0$). Therefore, the continuity equation is automatically satisfied. With the abbreviation $W_{ij} = W(|\boldsymbol{r}_i - \boldsymbol{r}_j|, h)$ the particle density ϱ_i can be calculated from

$$\varrho_i = \sum_j m_j W_{ij} \quad , \tag{5}$$

which is motivated by evaluating Equation (3) with $f(\boldsymbol{r}) = \varrho(\boldsymbol{r})$.

For example, using Eq. (4), the Euler equation in Lagrangian formulation yields in the SPH formalism

$$\frac{d\boldsymbol{v}_i}{dt} = -\sum_j m_j \frac{p_j + p_i}{\varrho_j \varrho_i} \nabla_i W_{ij} \quad , \tag{6}$$

where we have added the term proportional to p_i in order to anti-symmetrize the inter-particle forces. Other formulations of the SPH equations are also possible. For a more detailed overview of the standard SPH method see, e. g., Monaghan [4].

The advantage of using a kernel with compact support is the finite interaction range of the particles, i. e., only a particle's neighbors within a distance of the smoothing length h contribute to the above SPH sums.

Because SPH is a grid-free method, complicated geometries are relatively easy to model. The local resolution of SPH can be enhanced considerably by allowing for a variable smoothing length. The disadvantage of the method is that neighborhood relationships change with time and have to be re-established every time the particle positions have changed.

1.3 Physical Viscosity

The viscous interactions are implemented according to the viscous stress tensor in the Navier-Stokes equation (Flebbe et al. [1]). This is important, e. g., for the simulation of accretion disks over large periods of time, because their evolution is governed by viscous processes which cannot be treated properly by an artificial viscosity. The reason is that in the hydrodynamical limit, i. e., for a vanishing smoothing length, the SPH equation of motion converges to the Navier-Stokes equation, and the physical viscosity does not vanish in contrast to the artificial viscosity.

One problem with the physical viscosity is the occurrence of second order derivatives of the velocity, and in the SPH formalism there are different possibilities to approximate these terms. It is generally better to treat them as a sequence of two first order derivatives rather than using second order derivatives of the kernel. This leads to the viscous acceleration

$$\left(\frac{dv_\alpha}{dt}\right)_i^{\text{visc}} = \sum_j m_j \left(\frac{\nu_j}{\varrho_i}\left(\sigma_{\alpha\beta}\right)_j + \frac{\nu_i}{\varrho_j}\left(\sigma_{\alpha\beta}\right)_i\right)\frac{\partial W_{ij}}{\partial x_\beta} \tag{7}$$

with the particle form of the shear $\sigma_{\alpha\beta}$

$$\left(\sigma_{\alpha\beta}\right)_i = \left(V_{\alpha\beta}\right)_i + \left(V_{\beta\alpha}\right)_i - \tfrac{2}{3}\delta_{\alpha\beta}(V_{\gamma\gamma})_i \quad , \tag{8}$$

where $(V_{\alpha\beta})_i$ is the particle representation of the velocity gradient

$$\left(V_{\alpha\beta}\right)_i = \left(\frac{\partial v_\alpha}{\partial x_\beta}\right)_i = \sum_k \frac{m_k}{\varrho_k}\big((v_\alpha)_k - (v_\alpha)_i\big)\frac{\partial W_{ik}}{\partial x_\beta} \quad . \tag{9}$$

Here, the Greek indices label spatial coordinates, whereas particles are again labeled by Latin indices.

It can be easily shown that this formulation conserves momentum exactly (Riffert et al. [5]). In addition, the viscous terms of the energy equation can be formulated in such a way that the total energy is conserved exactly.

2 Implementation Details

Depending on the integration time, simulations with currently up to about one million particles can be performed. In order to simulate three-dimensional flows with more than a million particles while keeping the memory requirements and the turnaround time within reasonable limits, we implemented a distributed version of the SPH algorithm. We distribute the problem using the Message Passing Interface (MPI) by assigning a cuboid region of the simulation domain to every processor. Because the particles are not uniformly distributed, it is important that these cuboid regions can have different sizes which are allowed to vary in time.

2.1 Finding Interactions

The naïve and direct approach to evaluate the SPH sums is clearly not reasonable because it leads to a N^2-algorithm where N is the number of particles. Thus, an efficient way to search for all interacting particles is required. We chose to sort the particles into a uniform spatial grid with a cell size of the order of the smoothing length h which is constant in our applications. All possible interaction partners of a particle must reside within nearby grid cells. Another common approach that is especially suited for a variable smoothing length uses tree structures instead of a grid.

In order to calculate the right hand side of the hydrodynamical equations in the SPH formalism one has to evaluate three SPH sums, the first to determine the density at each particle position, the second and the third to calculate first and second order derivatives of the field quantities. Because finding interacting particles takes up a considerable amount of the total computing time, it is reasonable to re-use the interaction information in all three sums.

Previous implementations stored a list of all interacting particle pairs in memory. This list was then traversed three times. Given that there are of the order of 100 interactions per particles it is clear that this approach is not ideal, because this list is then the largest data structure and determines the total memory requirement. A single Cray T3E node with 128 MB of main memory could thus work on no more than about 20,000 particles.

Our implementation never stores the whole list in memory. Instead the set of particles is swept along a certain dimension, determining interacting particle pairs on the fly. As soon as the result of the first sum is known for a particle, the particle's contribution to the next sum is calculated. Particles for which the right hand sides of the dynamical equations are completely known are dropped from the interaction list, thus decreasing the list length. The state of a particle is determined by the position of the particle in the grid as shown in Figure 1.

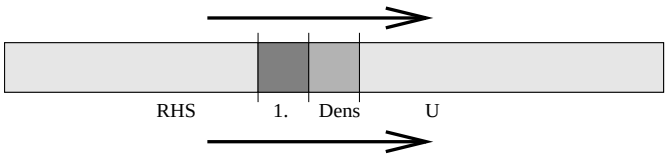

Fig. 1. The particles are swept from left to right. Particles in the region "U" are yet untouched. Only particles from the regions marked "Dens" and "1." are kept in the interaction list, their density and first order derivatives, respectively, are known. Particles in the region marked "RHS" have been completely processed, they have been dropped from the interaction list.

This algorithm drastically reduces the total memory requirement to about one tenth without increasing the overall computing time.

2.2 Boundary Conditions

A very flexible method to formulate boundary conditions in the SPH formalism consists of including *ghost particles*, i. e., particles that are not integrated in time but are merely a kind of mirror image of other particles. Due to the finite smoothing length h it is never necessary to include a boundary regime with a width larger than h. Periodic boundaries are implemented by copying the ghost particles on one side of the domain from the particles on the other side. For reflecting boundaries the ghost particles are a copy of the particles on the same side, and vector and tensor quantities are additionally reflected at the boundary plane. These ghost particles are inserted prior to each right-hand-side evaluation and dropped afterwards.

There are two optimizations that are important when the number of ghost particles is comparable to the number of real particles, as is the case for simulations with only a few real particles. First, one never calculates any quantities for the ghost particles directly, instead those quantities are copied from the corresponding real particles. Secondly, in the case of periodic boundaries, unnecessary duplicate evaluations of SPH sums are avoided by placing ghost particles only on one side of the domain in each spatial direction (see Figure 2). After these optimizations there is no net computing time spent on the ghost particles except for copy operations.

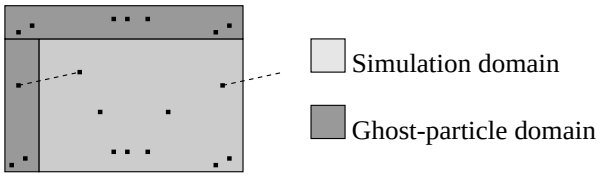

Fig. 2. Periodic boundaries: Ghost particles are placed only on one side of the simulation domain. Interactions that cross the simulation boundary (dotted lines) are accounted for only once, in this case on the left side of the domain.

2.3 Distributed Computing

The MPI library runs a program several times on different processors, there is no uniform view of a single process running distributed. Our aim was to re-use as much as possible from an existing optimized sequential SPH implementation. In our parallel code, each processor is assigned a cuboid

region of the simulation domain. The interface to the other regions is treated as a real, physical boundary rather than an algorithmic one.

The resulting (physical) boundary conditions are realized in a way conceptually similar to periodic boundaries but they include communication to neighbor processors. The ghost particles on one processor are mirror images of particles on other processors. Again, as in the case of periodic boundaries, duplicate evaluations of the sums are avoided. Creating the ghost particles and copying values between them and their real counterparts are the only time critical sections of the code that involve communication.

The sweeping algorithm mentioned above (see again Figure 1) takes place on every processor. Copying values from and to ghost particles, i. e., communication between processors, occurs when a particle changes its state. It is possible to delay changing the state of a particle for some time. This delay can be used to hide the time spent on communication. This delay also increases the size of the interaction list and thus acts as a tuning parameter for a size vs. speed tradeoff.

2.4 Load Balancing

Distributing the problem in the way outlined above is relatively inflexible with respect to load balancing. The boundaries between the processors' regions are treated as physical boundaries, and it is difficult to change them dynamically. It is however possible to change them between individual time steps using statistics gathered during the previous time step. This is also the time to remove or insert particles, if necessary.

In order to equally balance the load, a sufficient number of degrees of freedom is required. If $n \times m$ processor regions are placed on a rectangular $(n \times m)$-grid with unequal spacing then there are only $n + m - 2$ degrees of freedom, which is in general less than the $n \times m - 1$ degrees required. To circumvent this problem, a dimensional tree is used instead rather than a rectangular grid. Thus, an individual region can then have more than one neighbor in one direction (see Figure 3).

The domain decomposition is optimized in order to distribute the workload equally among the regions. Before the first time step, the number of particles is used as a measure for the load. After the right hand side has been evaluated, the number of interactions is used instead. Once an acceptable set of domain boundaries has been found, it is eventually fine tuned according to the time a processor spent waiting on other processors. The time spent re-balancing the load between time steps is only an insignificant fraction of the total computing time.

2.5 Efficiency of the Parallel Code

We have tested our implementation on various computer architectures, among these a cluster of workstations (Beowulf), the IBM SP2, and the Cray T3E.

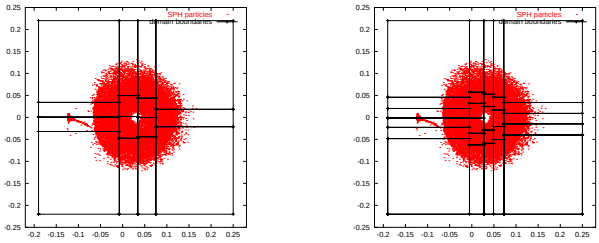

Fig. 3. Sample of our domain decomposition: The simulation domain is cut into slices along the x-axis. These slices are then independently cut along the y-axis, leading to a total of $(n-1)+n\times(m-1)=n\times m-1$ degrees of freedom. The figures show the boundaries of decomposed domains with 15 and with 32 processors, respectively.

Comparison Cray T3E – Pentium2 (300MHz)			
Number of Particles	Number of nodes	runtime Cray	runtime PC
75 000	16	1.77 s	8.08 s
375 000	128	4.16 s	39.8 s

Comparison IBM SP2 – Pentium2 (300MHz)			
Number of Particles	Number of nodes	Runtime IBM SP2	Runtime PC
302 961	16 (wide)	7.902 s	29.7 s
302 961	64 (thin)	1.907 s	29.7 s

Table 1. Comparison of some simulations. The runtime given is the time for one evaluation of the right hand side of the equations on n nodes of the supercomputers vs. 1 CPU on the PC.

A typical moderately sized problem contains 275,000 particles and runs on 64 processors with a total processor usage of more than 92 % on the T3E, as reported by MPI. This we believe to be quite satisfactory.

In Table 1 some comparisons of simulations on Cray TE3 and IBM SP2 vs. standard PC workstations are shown. When comparing CRAY T3E and IBM SP2 one has to keep in mind that a Cray node has only 128 MB memory, whereas the IBM nodes possess 256 MB or 512 MB memory, allowing for larger particle number per node, resulting in less communication costs.

3 Astrophysical Applications: Accretion Disks in Symbiotic Binary Stars

Symbiotic binary stars are so close to each other that their evolution and appearance changes dramatically compared to single stars. Dwarf Novae are

a class of variable symbiotic binary stars where mass transfer from one star to the other occurs. The donor is a light main sequence star, the other a more massive, but much smaller White Dwarf (WD). Due to its intrinsic angular momentum the overflowing gas cannot be accreted by the WD right away, instead a thin gaseous disk around the WD forms and the subsequent accretion is governed by viscous processes in the disk. The physics of these accretion disks is far from being well understood. Existing models of long term outburst behavior are essentially 1D and neglect the tidal influence of the donor star. Observationally, the disks show variability on timescales from minutes to decades, occasionally increasing in brightness up to 5 magnitudes. Numerical simulations are very helpful in exploring and validating the theoretical models. Such simulations, especially in 3D, require enormous amounts of grid points – or particles in our case – to achieve the necessary resolution. Since the problem size is so large, and the integration time so long, parallel programs on supercomputers are the only possible way to go.

3.1 The Stream-Disk Overflow in Dwarf Novae

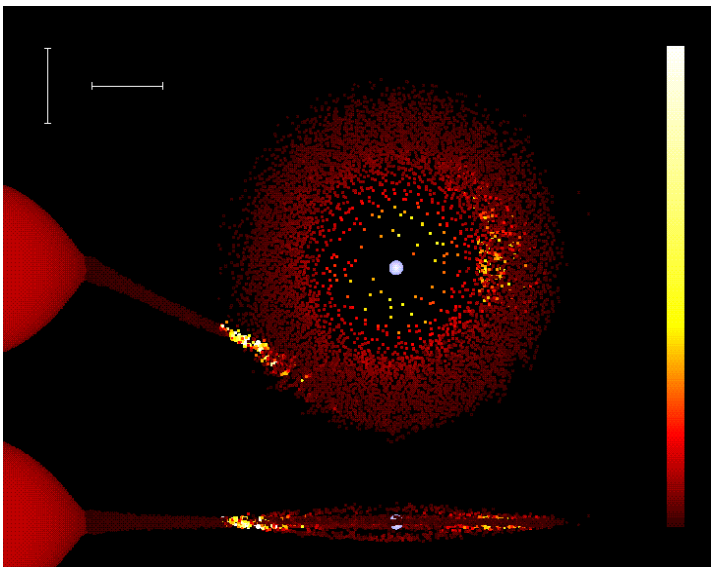

Fig. 4. Pole-on and edge-on view of a simulation of the accretion disk of the Dwarf Nova OY Car. Mass of the White Dwarf: 0.696 $M_\odot$, mass of the donor star: 0.069 $M_\odot$. The scales indicate 0.1 solar radii. The donor star is on the left. Color-coded is the dissipated energy. One can see the bright spot at the impact site of the stream onto the disk, and a secondary bright region right of the White Dwarf, where overflowing material re-enters the disk.

One aspect of Dwarf Nova disks is the impact of the overflowing gas stream onto the rim of the accretion disk. Both flows are highly supersonic and two shock regions form. The shocked gas becomes very hot, a bright spot develops, which sometimes can be brighter than the rest of the disk. The relative heights of the stream and the rim of the disk are unclear. If the stream is thicker than the disk, a substantial portion of the in-falling gas could stream over and under the disk and impact at much smaller radii.

Figure 4 shows a snapshot of the simulation of the accretion disk of the Dwarf Nova OY Carinae. The donor star has 0.069 solar mass, the White Dwarf 0.69 solar mass. Color-coded is the energy release due to viscous dissipation. One can clearly see the bright spot where the streams hits the disk rim. Furthermore, on the far side of the donor star, a secondary bright spot is visible where overflowing stream material finally impacts onto the disk. In this simulation, about 10 to 20% of the stream material can flow over and under the disk.

3.2 Superhumps in AM CVn

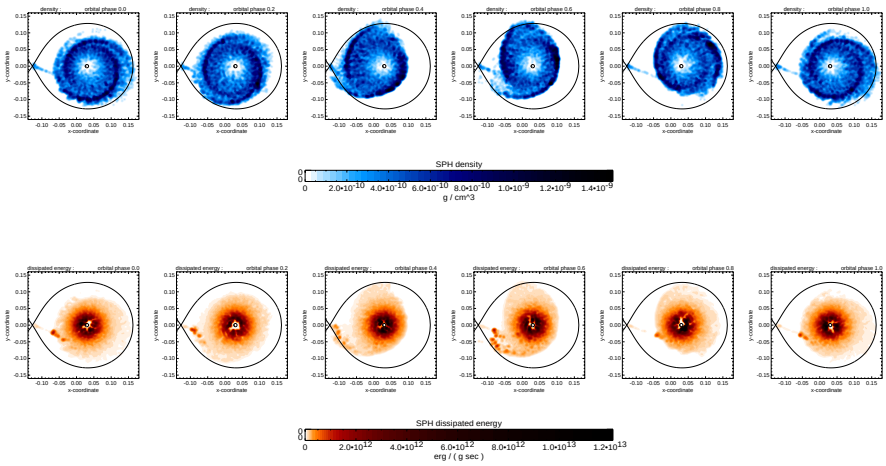

Fig. 5. A series of snapshots of the disk of Am CVn, 0.2 orbital phases apart. Upper panel: density distribution. Lower panel : dissipated energy. The parameters used are: $M_1 = 1$ solar mass, $M_2 = 0.15$ solar masses, mass transfer rate 10^{-10} solar masses per year. A ploytropic equation of state with $\gamma = 1.01$ was used. One can see how the precession of the tidally distorted disk leads periodically to higher dissipation, resulting in superhumps in the light-curve, see Figure 6

AM Canem Venaticorum stars are thought to be the helium counterparts to dwarf novae. So far, there are six systems known. Their common characteristics are hydrogen deficient spectra and photometric variations with very

short periods of 1000 to 3000 s. AM CVn stars are believed to consist of two helium white dwarfs, a rather massive primary and a very light, Roche-lobe filling secondary. Roche-lobe overflow feeds an accretion disk around the primary. Tsugawa & Osaki [6] showed that such helium disks undergo thermal instabilities similar to the hydrogen disks in Dwarf Novae. In three AM CVn stars, Dwarf Nova-like outbursts indeed have been observed.

Among the AM CVn stars, AM CVn itself and its near twin EC 15330-1403 have the shortest photometric periods and show no large amplitude outbursts. This, together with a high mass transfer rate, renders it plausible that the disks are thermally stable in a sort of "permanent outburst" state. The hydrogen counterparts of such systems are the so-called permanent superhumpers.

In order to investigate whether AM CVn exhibits superhumps we performed 3D-SPH simulations of the accretion disk. Particles were inserted at the inner Lagrangian point according to a mass transfer rate of $10^{-10} M_\odot/\mathrm{yr}$. Already after about 30 orbital periods the disk grew to a point where it was subject to the 3:1 inner Lindblad resonance [8]. Subsequently, the disk became more and more tidally distorted and started to precess rapidly in the frame of reference Co-rotating with the stars, which translates to a slow prograde precession in the observers' frame. Every time the bulk of the disk passes the secondary, the tidal stresses and hence the viscous heating are strongest, giving rise to modulations in the photometric light-curve, the superhumps. A Fourier transform of the obtained light-curve reveals a superhump period excess of 4.4 %. This is in good agreement with the periods given by Warner [7], which differ by 3.8 %.

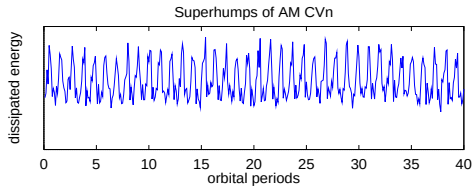

Fig. 6. Shown is the total dissipated energy of the disk over a time span of 40 orbital periods. A Fourier transform of the dissipated energy reveals a superhump period of 4.4 %.

References

1. Flebbe, O., Münzel, S., Herold, H., Riffert, H., Ruder, H. 1994, ApJ **431**, 754
2. Gingold, R. A., Monaghan, J. J. 1977, MNRAS **181**, 375
3. Lucy, L. B. 1977, Astron. J. **82**, 1013
4. Monaghan, J. J., 1992, Ann. Rev. Astron. Astrophys. **30**, 543
5. Riffert, H., Herold, H., Flebbe, O., Ruder, H. 1995, Comp. Phys. Comm. **89**, 1
6. Tsugawa, M., Osaki, Y. 1995, PASJ **49**, 75
7. Warner, B. 1995, Ap&SS **255**, 249
8. Whitehurst, R. 1988, MNRAS **232**, 35