DISTRIBUTED IMPLEMENTATION OF SPH FOR SIMULATIONS OF ACCRETION DISKS

R. SPEITH, E. SCHNETTER, S. KUNZE, H. RIFFERT
Institut für Astronomie und Astrophysik
Auf der Morgenstelle 10, D-72076 Tübingen, Germany
Email: lastname*@tat.physik.uni-tuebingen.de*

We present a brief introduction to Smoothed Particle Hydrodynamics (SPH), where we focus on an approach to treat the physical viscosity. Additionally we describe in detail the basic principles of our parallel implementation of the SPH method. In the second part we summarise some results of test simulations performed to achieve general features of the method, and we end discussing some astrophysical applications, namely the simulation of accretion disks in Cataclysmic Variables.

1 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a grid-free Lagrangian method for solving the Navier-Stokes equations numerically. It was introduced in 1977 by Gingold & Monaghan[1] and by Lucy[2]. 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 neighbour 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 normalised to unity. The smoothing length h is a measure of the (compact) support of the kernel function, and it follows

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

A common choice for the kernel is a cubic spline (see, e. g., Monaghan[3]).

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 Eq. (3) with $f(\boldsymbol{r}) = \varrho(\boldsymbol{r})$.

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

$$\frac{d\boldsymbol{v}}{dt} = -\frac{1}{\rho} \nabla p \tag{6}$$

yields in the SPH formalism

$$\frac{d\boldsymbol{v}_i}{dt} = -\sum_j m_j \frac{p_j + p_i}{\varrho_j \varrho_i} \nabla_i W_{ij} \quad , \tag{7}$$

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[3].

The advantage of using a kernel with compact support is the finite interaction range of the particles, i. e., only a particle's neighbours 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 neighbourhood 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.[4]). 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

$$\left(\frac{dv_\alpha}{dt}\right)_i^{\text{visc}} = \sum_j m_j \left(\frac{\nu_j}{\varrho_i}\left(\sigma_{\alpha\beta}\right)_j + \frac{\nu_i}{\varrho_j}\left(\sigma_{\alpha\beta}\right)_i\right)\frac{\partial W_{ij}}{\partial x_\beta} \tag{8}$$

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

$$\left(\sigma_{\alpha\beta}\right)_i = (V_{\alpha\beta})_i + (V_{\beta\alpha})_i - \tfrac{2}{3}\delta_{\alpha\beta}(V_{\gamma\gamma})_i \quad , \tag{9}$$

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

$$(V_{\alpha\beta})_i = \left(\frac{\partial v_\alpha}{\partial x_\beta}\right)_i = \sum_k \frac{m_k}{\varrho_k}\left((v_\alpha)_k - (v_\alpha)_i\right)\frac{\partial W_{ik}}{\partial x_\beta} \quad . \tag{10}$$

Here, the greek indices label spatial coordinates, whereas particles are again labelled 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 cuboids can have different sizes which 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

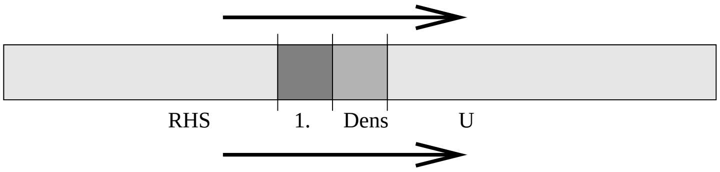

Figure 1. The particles are swept from left to right. Particles in the region "U" are yet untouched. Only particles from the regions marked "Dens" and "1." are kept in the interaction list, their density and first order derivatives, respectively, are known. Particles in the region marked "RHS" have been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

takes up a considerable amount of the total computing time, it is reasonable to reuse 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 was the largest data structure and determined the total memory requirement. A single Cray T3E node with 128 MB of main memory could thus work on no more than about 10,000 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.

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 optimisations 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 optimisations there

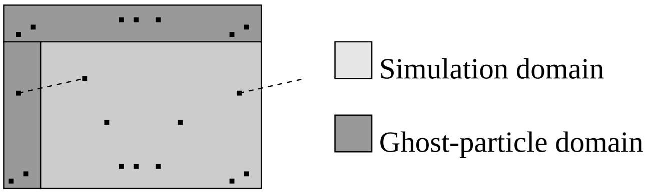

Figure 2. Periodic boundaries: Ghost particles are placed only on one side of the simulation domain. Interactions that cross the simulation boundary (dotted lines) are accounted for only once, in this case on the left side of the domain.

is no net computing time spent on the ghost particles except for copy operations.

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 distributedly. Our aim was to re-use as much as possible from an existing optimised 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 neighbour 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

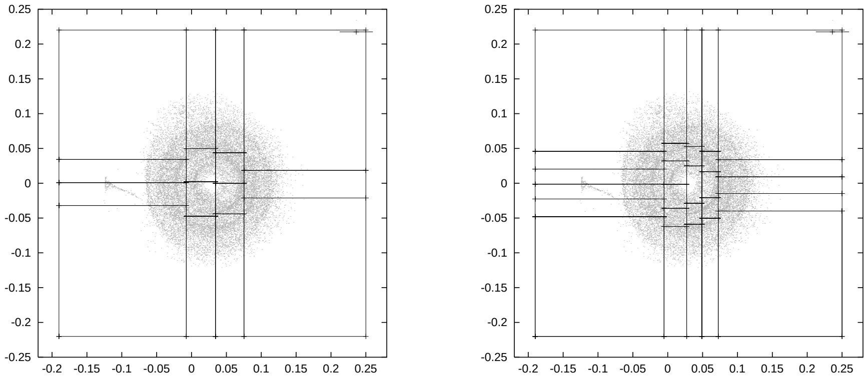

Figure 3. Sample of our domain decomposition: The simulation domain is cut into slices along the x-axis. These slices are then independently cut along the y-axis, leading to a total of $(n - 1) + n \times (m - 1) = n \times m - 1$ degrees of freedom. The figures show the boundaries of decomposed domains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

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 neighbour in one direction (see Figure 3).

The domain decomposition is optimised in order to distribute the work-load 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.

We have tested our implementation on various computer architectures, among these a cluster of workstations (Beowulf), the IBM SP2, and the Cray T3E. 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. This we believe to be quite satisfactory.

3 Testing SPH

As every numerical algorithm, SPH has to be tested in order to verify its stability, accuracy, and robustness. An ideal test is, of course, a comparison with a model problem where an analytic solution is known, and examples for one-dimensional flows include the well-known shock tube, advection problems, or self-similar solutions of the Euler equation for a perfect gas. For many numerical methods error estimates are impossible to obtain, and analytic solutions can be used to determine error bounds empirically. This is even more important when the mathematical foundations of the method are poorly understood, as is currently the case for Smoothed Particle Hydrodynamics.

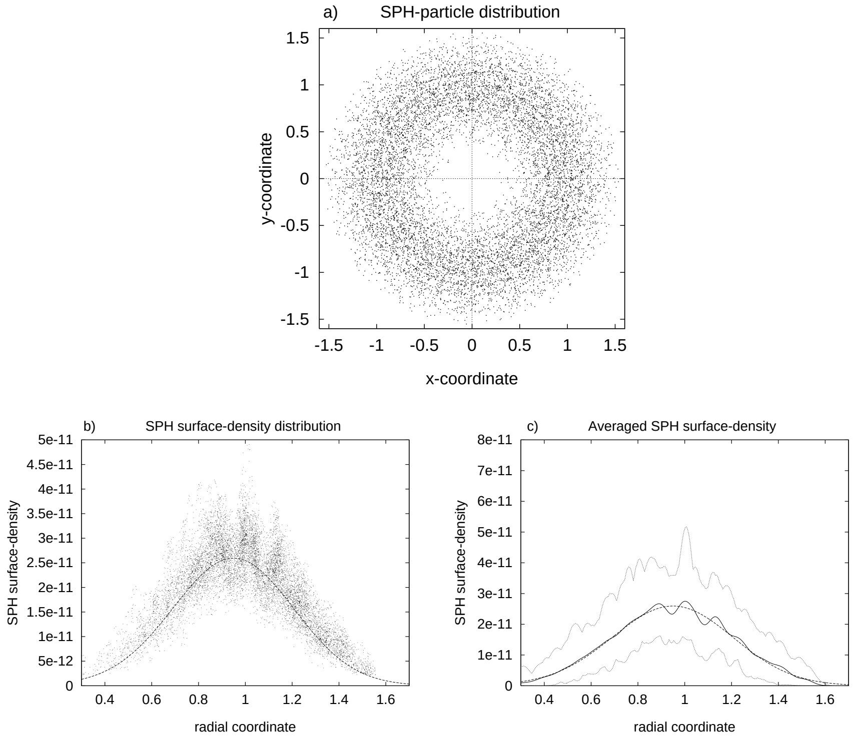

Figure 4. Simulation of the viscous ring: Figure a) on top shows a snapshot of the SPH particle distribution. Figure b) on the left gives the azimuthally projected surface density of each particle. Note the large scattering of the values around the analytic solution (dashed line). In Diagram c) on the right, the straight line represents the azimuthally averaged SPH surface density, while the dashed line again indicates the analytic solution, and the dotted lines give an upper and lower limit of the scattering of the results. Note that the averaged solution matches the analytic solution quite well.

3.1 A Test Problem for the Physical Viscosity

For two- or three-dimensional gas flows analytically solvable problems are rare, but there is one astrophysically interesting example where at least an approximate solution of the hydrodynamic equations can be found. This test problem, the viscous spreading of a thin gas ring around a central point mass, is an idealised example of an astrophysical accretion disk. The gas experiences only the gravitation of the central object and viscous shear forces. Azimuthally the gas moves with Keplerian velocity. The viscous friction transports mass inwards while angular momentum is transported outwards and the ring is spreading in the radial direction. If rotational symmetry is assumed an evolution equation for the density can be derived, and for a constant kinematic viscosity an analytic solution of the initial value problem is obtained (Lüst[6], Pringle[7]).

Figure 4 shows an example of some SPH simulation results. The simulations proved the accuracy of the viscosity description introduced in Sec. 1.3. The global dynamical behaviour of the disk is reproduced quite accurately. But the local accuracy and the spatial resolution highly depend on the number of particles. If it is too small, the mean deviations from the analytical values are very large.

3.2 General Results

Our test calculations have shown some important and universal features of the SPH method. One result is that SPH is able to reproduce global attributes and the dynamical evolution of the considered physical problems. However, the local accuracy and the spatial resolution of SPH simulations are rather poor. Actually, the results obtained from a two-dimensional SPH code using 10^4 particles should be compared to a finite difference simulation on a grid of 100×100 cells. Moreover, the SPH calculations show a large fluctuation of the results, at least if the particle distribution remains in a "stochastic" state, which is always the case when external forces dominate the hydrodynamical forces.

Two fundamental numerical parameters of the method can be identified. The smoothing length h determines the spatial resolution of the simulations, whereas the global accuracy depends on the mean number $\overline{N}_{\rm i}$ of interactions per particle. The higher $\overline{N}_{\rm i}$ and the lower h the better are the simulation results. Because these conditions are partly contradictory, one has to find a compromise by choosing the total particle number according to the computer resources available.

4 Accretion Disks in Close Binary Stars

Dwarf Novae are variable binary star systems where mass transfer from one star to the other occurs. The donor star is a light main sequence or late type star, the other one is a more massive but much smaller White Dwarf (WD). Due to its intrinsic angular momentum the overflowing gas cannot be accreted by the WD directly. Instead, a thin gaseous disk forms around the WD and the subsequent accretion is governed by viscous processes in the disk. Due to a thermal instability in the disk Dwarf Novae occasionally undergo outbursts, i. e., a phase of increased brightness (about 3 mag), typically lasting a couple of days (this feature makes Dwarf Novae preferred objects for amateur astronomers). The physics of the accretion disks is far from being well understood. Existing models of the long term outburst behaviour are essentially one-dimensional and neglect the tidal influence of the donor star. Observationally, the disks show variability on time scales from minutes to decades, occasionally increasing in brightness up to 5 magnitudes. Numerical simulations are very helpful in exploring and validating the theoretical models.

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 hot, and a bright spot develops which sometimes can be brighter than the rest of the disk. The ratio of the heights of the stream and the rim of the disk are unknown. If the stream is thicker than the disk, a substantial portion of the in-falling gas could flow over and under the disk and impact at much smaller radii.

Figure 5 shows a snapshot of the simulation of the accretion disk of the Dwarf Nova OY Carinae. One can clearly see the bright spot where the streams hits the

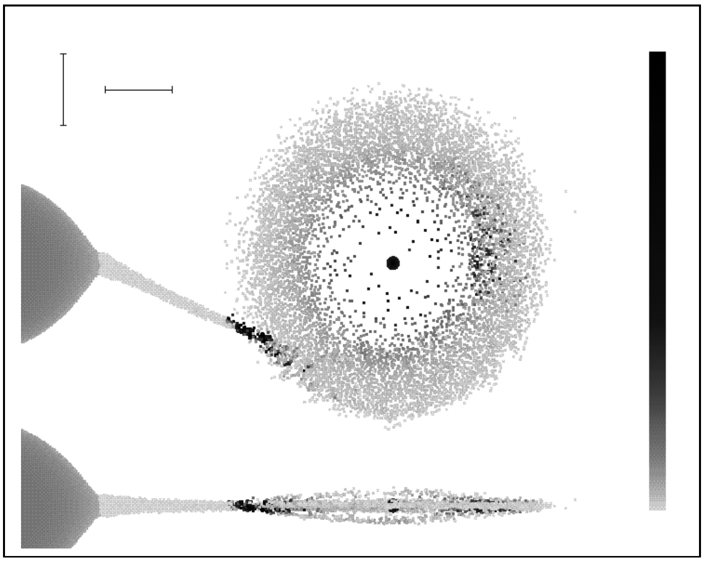

Figure 5. Pole-on and edge-on view of a simulation of the accretion disk of the Dwarf Nova OY Car. Mass of the White Dwarf: 0.696 $M_\odot$, mass of the donor star: 0.069 $M_\odot$. The scales indicate 0.1 solar radii. The donor star is on the left. Grey-coded is the dissipated energy. One can see the 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.

disk rim. Furthermore, on the far side[8] 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 flows over and under the disk.

Another point of interest is the nature of the so-called superoutburst phenomenon in a subclass of Dwarf Novae, the SU UMa stars. In addition to the normal outbursts, these stars occasionally exhibit brighter and longer lasting superoutbursts with an additional modulation of the brightness by approximately 30 % with a period which is a few percent longer than the orbital period. This period excess is specific for each system and ranges from about 0.8 % up to 8 %.

In our simulations we were able to confirm the combined tidal and thermal instability model by Osaki[9]. In systems where the WD is about four times more massive than the secondary star the accretion disk can grow up to the point where a tidal resonance sets in which can transform the disk from a rather round and stable shape into a strongly distorted one that precesses rapidly. A snapshot of such a disk is shown in Figure 6. Every time the bulk of the disk passes the secondary, the tidal friction is at a maximum and the brightness increases. Only SPH simulations have been able to reproduce the distorted precessing disks so far, and only with our prescription of the viscous stresses it has been possible to correctly reproduce the form of the outburst light curve as well as the correct period excesses for individual systems (see Kunze et al.[10]).

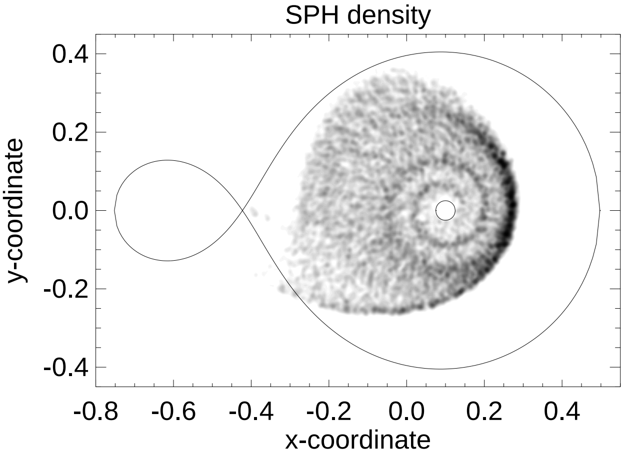

Figure 6. Snapshot of the simulation of the accretion disk of a SU UMa-type star during superoutburst. The eight-shaped line marks the Roche lobes of the stars. The secondary on the left fills its Roche lobe completely and looses mass to the primary via the inner Lagrangian point. The overflowing matter forms a disk with the WD in the centre.

Acknowledgments

The authors want to thank U. Kraus, F. Ott, and S. Siegler for many fruitful discussions and contributions. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) and is to some extend part of the project A1 of the Sonderforschungsbereich SFB 382.

References

1. R.A. Gingold and J.J. Monaghan, MNRAS **181**, 375 (1977).
2. L.B. Lucy, Astron. J. **82**, 1013 (1977).
3. J.J. Monaghan, Ann. Rev. Astron. Astrophys. **30**, 543 (1992).
4. O. Flebbe, S. Münzel, H. Herold, H. Riffert, H. Ruder, ApJ **431**, 754 (1994).
5. H. Riffert, H. Herold, O. Flebbe, and H. Ruder, CPC **89**, 1 (1995).
6. R. Lüst, Z. Naturforschg. **7**, 87 (1952).
7. J.E. Pringle, Ann. Rev. Astron. Astrophys. **19**, 137 (1981).
8. G. Larson, "The Far Side", Andrews McMeel Publishing, Kansas City (1982).
9. Y. Osaki, PASJ **41**, 1005 (1989).
10. S. Kunze, R. Speith, and H. Riffert, MNRAS **289**, 889 (1997).