DAWN: SIMULATIONS OF THE FORMATION AND EARLY EVOLUTION OF EMBEDDED CLUSTERS

Y. Bernard[1], E. Moraux[1], D. J. Price[2], F. Motte[1], F. Louvet[1] and I. Joncour[1]

Abstract. This presentation describes the new methods that have been integrated into the SPH code `Phantom`. We implemented specialised stellar dynamics algorithms to avoid any bottleneck from hard systems. We developed a new subgrid model to form groups of individual stars from 'large' sink particles to relax resolution requirements. Finally, we optimised the gas-star interaction that appeared to be the main bottleneck when most of the stars have been created in an embedded cluster. These developments enable highly efficient simulations of embedded cluster formation and evolution up to their emergence in the galactic environment. This efficiency is necessary to accumulate a large number of possible evolutions and determine statistically the preferred evolutionary paths.

Keywords: Star formation, Simulations, Embedded clusters, Stellar dynamics

1 Introduction

Embedded clusters (Lada & Lada 2003) are formed by the concentration of proto-stars and newborn stars during the gravitational collapse of a molecular cloud. During this collapse, the continuous formation of stars feeds these clusters, making them grow during their early life. This formation process imprints the parent collapsing cloud spatial distribution and kinematic properties on them. Hence, young star clusters are tightly linked to their surrounding gaseous environment. Their future evolution is closely related to this gas-star coupling, which is simultaneously driven by the gravitational interactions and stellar feedback processes (Bastian & Goodwin 2006; Krumholz et al. 2014). This early evolution connects these primordial states of stellar systems to more evolved structures such as open clusters or associations. Thus, defining the preferred evolutionary paths could reveal predominant properties of star-forming regions by backtracing older structures to their initial states (Parker et al. 2014; Wright et al. 2014).

However, the modelling of this evolution proved to be a great challenge. The interactions driving this evolution introduce significant stochasticity. Hence, no analytical solutions or simplistic numerical models could correctly describe the early evolution of embedded clusters (Kroupa et al. 2001). Consequently, refined numerical simulations directly modelling the formation and the evolution of a star cluster through the gravitational collapse of a molecular cloud up to the gas removal by the stellar feedback are the preferred option to tackle this challenge(Wall et al. 2019; Grudić et al. 2021). While such simulations are computationally expensive, a large number of realisations are required to statistically determine typical cluster evolution paths. Therefore, the real challenge lies in producing an efficient simulation framework to accumulate numerous different final states, thus providing a better understanding of the early evolution of star clusters and revealing their initial formation conditions.

In this presentation, we introduce our new methods, which optimise the star formation process and gas-star interactions, that have been implemented in the SPH code `Phantom` (Price et al. 2018; Bernard et al. 2025). Here, we make the choice to run rather low-resolution simulations, and we use a subgrid model to form individual stars inside sink particles. Collisional stellar dynamics is also a key factor in cluster evolution. We present several state-of-the-art methods that we implemented in `Phantom`. They are composed of a fourth-order time integration scheme, which is now the default option and regularisation algorithms to handle close

[1] Univ. Grenoble Alpes, CNRS, IPAG, 38000 Grenoble, France
[2] School of Physics and Astronomy, Monash University, Vic. 3800, Australia

encounters, hard binaries, multiples and their secular evolution. Then, we describe our implementation of H II region expansions to model feedback in our simulations. We also present a new option to push sink particles in the tree for faster sink-gas gravitational interactions. Finally, we present the outcomes of this new simulation framework applied to typical giant molecular clouds.

2 Stellar dynamics

In `Phantom`, gravitational forces between point masses (as stars) are computed using a direct method. This direct algorithm is coupled to a Leapfrog symplectic integrator to follow stellar dynamics. This integrator is only second order, and needs short time steps to reach a sufficient accuracy. This combination of methods is inefficient and can be inaccurate during the early evolution of embedded clusters. To resolve this accuracy and performance issue, we implemented three methods commonly used in stellar dynamics code.

First, we switched the Leapfrog second-order integrator for a fourth-order one, called forward symplectic integrator or FSI (Chin & Chen 2005; Dehnen & Hernandez 2017; Rantala et al. 2021). It can integrate star motions accurately and efficiently without using backward substeps in its scheme, which is a common drawback of these high-order integrators. In practice, using this new integrator relaxed the number of timesteps by a factor of four, with still better accuracy compared to the previous one.

However, using only a higher order of integration can not overcome the constraints induced by close encounters, close binaries and multiple systems. In such cases, the timestep can drastically drop to values of multiple orders of magnitude below the mean constraint. It can impact both the performance as the simulation stalls on these fast dynamics, and the accuracy by computing a tremendous number of steps that could induce more and more truncation and round-off errors. In such cases, we implemented a new integrator called the time-transformed leapfrog (TTL)(Mikkola & Tanikawa 1999; Mikkola & Aarseth 2002), which is only used for these stellar subsystems. This method regularises the gravitational singularity of a pair interaction by introducing a new numerical time variable s that could use constant timesteps even during a close encounter. The real time is an integrated variable related to s by :

$$\mathrm{d}s = \Omega(\mathbf{r})\mathrm{d}t, \tag{2.1}$$

where $\Omega(\mathbf{r})$ is a time transformation function. This function is often taken as the gravitational potential of the system, so that the time is slowly integrated as the potential increases during a close encounter. This new integration can efficiently control the accuracy of these close subgroups of stars. However, it can not handle the closest interactions where the timestep constraint is the highest.

To handle this potential major bottleneck, we implemented a last algorithm called slowdown binaries (SD)(Mikkola & Aarseth 1996; Wang et al. 2020) that applies a transformation to the Hamiltonian of weakly perturbed hard binaries.

$$H = \frac{1}{\kappa}H_b + (H - H_b), \tag{2.2}$$

A slowdown factor κ is applied to the close binary Hamiltonian H_b. The result of this transformation is a slowdown of the hard binary motion, reducing the timestep constraint by κ times. While the real orbital phase is lost, the secular evolution is still preserved as the perturbations applied to the binary are still accounted for. The accuracy of the method is then controlled by the choice of κ that codes the number of orbits the binary can do before being modified too much by the perturbations. We used the following equation to control this value accordingly to the tidal perturbation on the binary.

$$\kappa = \frac{m_p m_s}{m_p + m_s(a(1+e))} \sum_{j=1, j\neq p,s}^{N} \frac{r_{bj}^3}{m_j}, \tag{2.3}$$

where m_p and m_s are the masses of the primary and secondary, a is the semi-major axis, and e the eccentricity. The sum part computes the total tidal perturbation applied to the binaries by stars in the environment.

The last two algorithms are applied only to small compact subgroups in the simulation, which are identified in a searching radius of 200 au using a similar method as shown in Rantala et al. (2023).

3 Star formation recipe

We use a subgrid model to form stars inside 'large' sink particles, where the fragmentation and accretion processes are not yet over. Typically, accretion radii are set to 4000 astronomical units in our prescription. In that sense, sinks can be compared to dense cores observed in star-forming regions. This choice of large sinks can significantly lower the resolution requirement for the simulation.

Our star formation procedure from sink particles is illustrated in Fig. 1 and proceeds as follows:

1. First, a sink is formed when a clump of gas reaches formation conditions(Bate et al. 1995; Price et al. 2018);
2. The sink accretes surrounding continuously during $t_{\rm acc}$;
3. At the time $t_{\rm seed}$, the number of star seeds inside the sink is sampled randomly between 1 and 5. Accretion continues;
4. When $t_{\rm acc}$ is reached, accretion is over. The sink is dissolved, and young stars are released within its volume, keeping its momentum. Star masses are set by randomly sharing the parent sink mass reservoir. Positions and velocities are sampled to respect the dynamical stability of the group as described below;
5. If sinks intersect each other by $r_{\rm acc}$ during their lifetime, they merge to form one single sink. In that case, if star seeds have already formed, then their number will be summed up into the merged one. If not, only the mass and the momentum of both sinks are added to the merged sink.

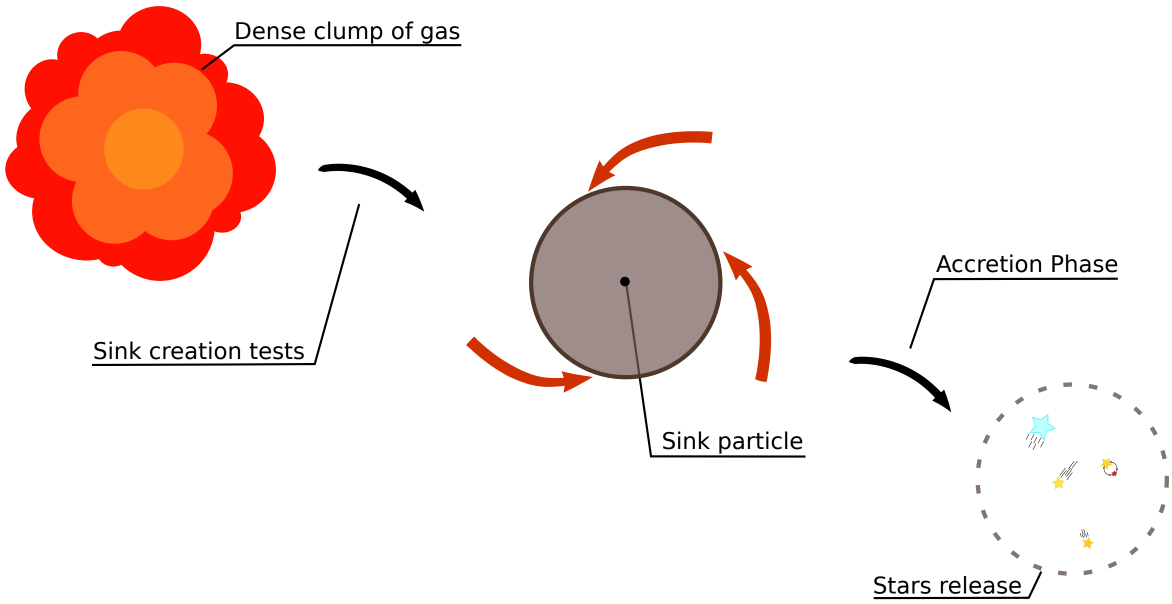

Fig. 1. Star formation process described by our new prescription, from the clump transformation into sink particles to stars release inside the simulation domain. This process is subdivided into three parts. First, a collapsing clump of gas passes tests to be transformed into a sink particle. The sink particle can accrete during $t_{\rm acc}$. After reaching this time, the sink is dissolved and its mass is shared to produce newborn stars at its last location.

As was stated above, released stars after accretion onto their parental sink need to be distributed in their birth vicinity. Each new star will have a mass, position, and velocity that need to conserve the physical properties of the parental sinks. The total mass of the sink is conserved and shared between child stars. The simplest way to share sink mass between stars is a random sharing with a cut at 0.08 $M_\odot$ to ensure the formation of stars and avoid brown dwarf and substellar mass objects. We implemented this by sampling $n_{\rm seed} - 1$ random numbers between zero and one. That way, a unit segment is subdivided into multiple sub-segments of random length, and their sum is always equal to one. Multiplying these segments by the mass of the sinks then produces a set of masses randomly chosen from the available mass in the reservoir. In addition to that, we resampled some points to ensure that no sub-segments are lower than 0.08 $M_\odot$.

Finally, star velocities and positions should produce a bound system inside the sink volume. Many solutions exist for sampling these quantities. We opted for a simplistic approach using the same method that produces Plummer-like clusters in stellar dynamics-oriented simulations (Aarseth et al. 1974)

4 Optimised Star-Gas interactions

The first simulations reported a large performance improvement thanks to the new stellar dynamics and the new star formation recipe that could reach a factor of 80 in our first tests. Nonetheless, the star-gas gravitational interactions began to be the main bottleneck (up to 80% of the compute time) that limited the simulation from running for long enough. These are calculated with a direct method in Phantom and always at the lowest time step of the scheme. In most applications of `Phantom` to date, this part of the algorithm was not the main bottleneck, as the sink number was not large enough for the $\mathcal{O}(N_{\rm sink}N_{\rm gas})$ interactions to dominate the complexity. Such interactions essentially scale linearly with particle numbers in our simulations, as in previous studies. As a result, this bottleneck makes the computation too expensive to go further in time than 6 to 7 Myr for our application.

To untie this bottleneck, we added a new option in `Phantom` that allows the user to compute the sink-gas interaction using the kd-tree of the code. Using this new option, sink particles are pushed inside the tree structure. That way, the long-range star-gas interactions can be approximated using the multipole computed on distant nodes of the tree, reducing the numerical complexity to $\mathcal{O}(N_{\rm sink}\log(N_{\rm gas}) + N_{\rm gas}\log(N_{\rm sink}))$. The short-range star-gas continues to be computed with a direct method. As done for the self-gravity in `Phantom`, the switch between the two types is controlled by an opening angle criterion $\theta = 0.5$. Overall, this optimisation made this bottleneck completely vanish. To give an example, a calculation that would have taken 25 days without the optimisation takes now 2.5 days.

5 Conclusion

To conclude, the methods presented above have relaxed major bottlenecks that prevented the generation of many realisations of embedded cluster formation and evolution simulations. The `Phantom` code is now fully prepared to perform a large sample of simulations, which is necessary for future parametric studies and to discover preferred evolutionary tracks. The upcoming paper of the DAWN project will focus on one of these parametric studies to test the mass distributions outputted from our star formation recipe.

References

Aarseth, S. J., Henon, M., & Wielen, R. 1974, A&A, 37, 183
Bastian, N. & Goodwin, S. P. 2006, MNRAS, 369, L9
Bate, M. R., Bonnell, I. A., & Price, N. M. 1995, MNRAS, 277, 362
Bernard, Y., Moraux, E., Price, D. J., et al. 2025, arXiv e-prints, arXiv:2508.05296
Chin, S. A. & Chen, C. R. 2005, Celestial Mechanics and Dynamical Astronomy, 91, 301
Dehnen, W. & Hernandez, D. M. 2017, MNRAS, 465, 1201
Grudić, M. Y., Guszejnov, D., Hopkins, P. F., Offner, S. S. R., & Faucher-Giguère, C.-A. 2021, MNRAS, 506, 2199
Kroupa, P., Aarseth, S., & Hurley, J. 2001, MNRAS, 321, 699
Krumholz, M. R., Bate, M. R., Arce, H. G., et al. 2014, in Protostars and Planets VI, ed. H. Beuther, R. S. Klessen, C. P. Dullemond, & T. Henning, 243–266
Lada, C. J. & Lada, E. A. 2003, ARA&A, 41, 57
Mikkola, S. & Aarseth, S. 2002, Celestial Mechanics and Dynamical Astronomy, 84, 343
Mikkola, S. & Aarseth, S. J. 1996, Celestial Mechanics and Dynamical Astronomy, 64, 197
Mikkola, S. & Tanikawa, K. 1999, MNRAS, 310, 745
Parker, R. J., Wright, N. J., Goodwin, S. P., & Meyer, M. R. 2014, MNRAS, 438, 620
Price, D. J., Wurster, J., Tricco, T. S., et al. 2018, PASA, 35, e031
Rantala, A., Naab, T., Rizzuto, F. P., et al. 2023, MNRAS, 522, 5180
Rantala, A., Naab, T., & Springel, V. 2021, MNRAS, 502, 5546
Wall, J. E., McMillan, S. L. W., Mac Low, M.-M., Klessen, R. S., & Portegies Zwart, S. 2019, ApJ, 887, 62
Wang, L., Nitadori, K., & Makino, J. 2020, MNRAS, 493, 3398
Wright, N. J., Parker, R. J., Goodwin, S. P., & Drake, J. J. 2014, MNRAS, 438, 639