Processing binary populations in star clusters

Observatoire astronomique de Strasbourg

UNIVERSITÉ DE STRASBOURG

Julien Dorval, Christian Boily

Observatoire astronomique de Strasbourg, CNRS, UMR 7550, 11 rue de l'université, F-67000 Strasbourg, France

Contact : julien.dorval@astro.unistra.fr

Introduction

We investigate through N-body simulations the fate of a binary population within a young star cluster ongoing a violent core collapse. Our main interest lies in the potential transition in effect from small open clusters to globular cluster, and the growing influence of the global tidal field with increasing N. We use the Nbody6 solver (Aarseth, 2009) as it can easily handle close binaries through KS regularisation. We present here two aspects of our work :

- We introduce a new way to produce self-consistent fragmented initial clusters through our « Hubble-velocities » method
- As well, a new algorithm to detect binaries in a cluster is presented.

Background

Binary population processing is the action of background stars interacting with binaries : disruption, hardening, etc, which changes the period or eccentricity distribution. The present-day, evolved populations we find in star clusters are windows into their dynamical past. Some advocate for the existence of an universal birth binary population (analog to an universal IMF) which would be processed differently depending on the local density (Marks & Kroupa, 2012). Others investigate the influence of different birth populations (Kaczmarek et al, 2011), or structural parameters, such as the degree of substructure present in the young cluster (Parker et al, 2011).

Fragmentation through *Hubble-like* expansion

Several observations and simulations have shown young star clusters may be born cold and substructured (Parker et al, 2011). To start a simulation with such a fragmented cluster, the first task is to create a self-consistent initial distribution in both space and velocity. This is usually obtained through fractal growth (Goodwin & Whitworth, 2004).

We introduce the « Hubble velocity » method. A spatially uniform sphere is randomly generated, then particles are attributed velocities through the relation :

v = H x **r**

with H a chosen initial Hubble parameter. The system is evolved through an Nbody solver, up to the « turnoff » (when H[t] = 0). At this point, the cluster has spent its initial kinetic energy : it stops expanding, fully fragmented, before evolving to a state of equilibrium.

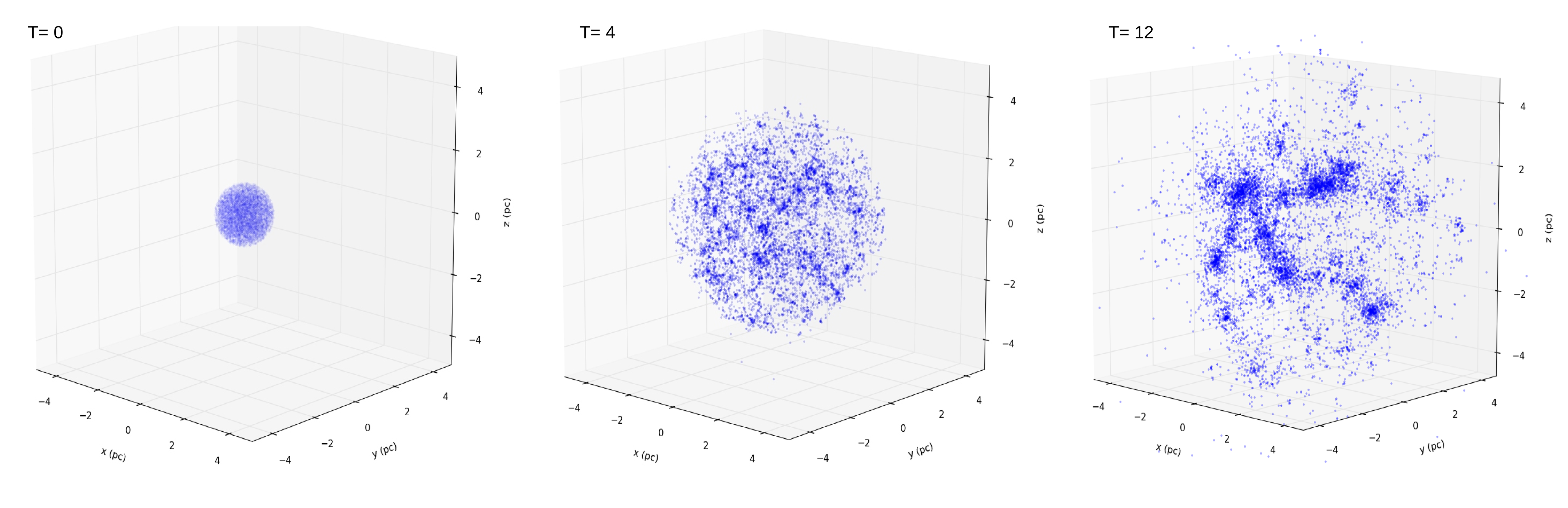


Fig 1 : Evolution of an initially uniform sphere (N= 8000) with Hubble-like velocities (H=1.28) . Time is given in dynamical time.

Velocity distribution

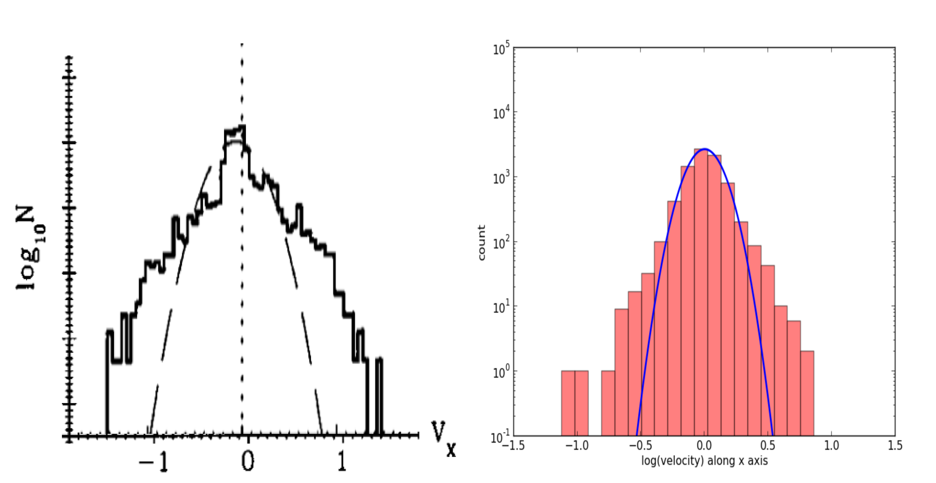


Fig 2 : V_{1d} distribution of a fragmented cluster. Left : Klessen & Burkert 2000, right : present work.

We obtain the same velocity distribution shape Klessen & Burkert (2000) obtained for their pre-stellar cores. While they were using SPH particles to form substructures through gas dynamics, we get similar results through pure Nbody dynamics.

The initial velocity distribution is Gaussian. The denser substructures lead to an increase in velocity dispersion, thus some high velocity tails . Here are displayed the v_{1d} distribution from Klessen & Burkert (2000) and the one we obtain for the above cluster at the turnoff. Gaussian fits are overlayed as lines.

How it works

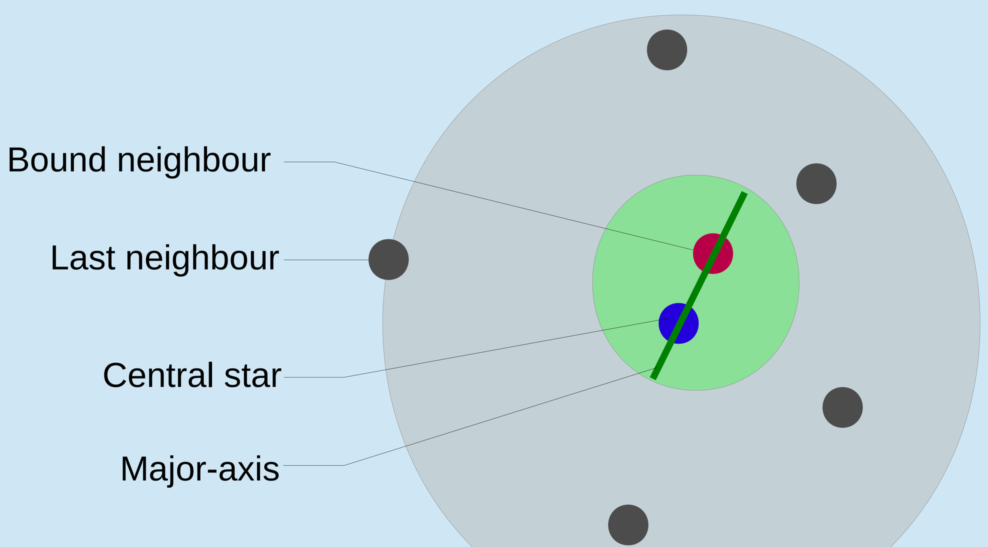
The unavoidable poissonian fluctuations in the initial stellar density grow during the expansion, much like the cosmological density fluctuations have grown over the expansion of the Universe. Stars rush towards denser regions, small clumps form, which either collect more stars or merge with other clumps. A the turnoff point, we obtain a subvirial, fragmented star cluster, with a fully self-consistent velocity distribution.

The degree of substructures can be tuned through the initial Hubble constant. The higher the initial velocities, the longer the expansion lasts, which allows more time for clumps to grow. It should also be noted that the total number of particles will also influence the growth of the clumps : the fragmentation is initiated by the Poissonian noise, which is larger with fewer particles.

An alternative to obtain initially fragmented clusters have been proposed by Goodwin & Whitworth (2004). They grow a « particle tree » with random branching, which ultimately leads to a fragmented, fractal structure.

Binary finding algorithm

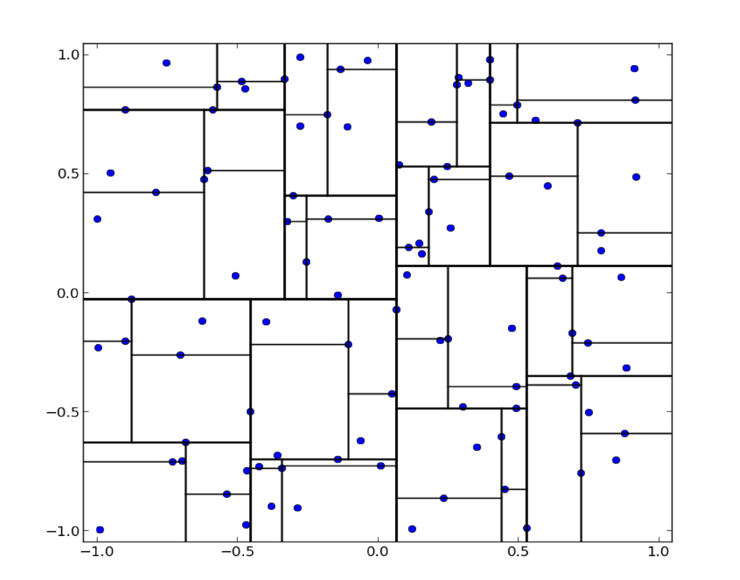
To detect binary in a star cluster is not a trivial task, whether in simulation or observations. Several paths have already been explored. The AMUSE framework provides a built-in method, comparing the « hardness » of bound pairs of stars to the average kinetic energy of the cluster. Kouwenhoven et al (2010) and Parker et al (2009) suggest two more advanced methods : bound pairs of stars are validated as binaries if they are mutual nearest neighbours and satisfy a proximity criterion. This is the approach we chose. Our method takes two input parameters :



- N_{nb} : the number of neighbours we consider.
- R_ρ : the density ratio for validation

1) For each star, the N_{nb} neighbours are found thanks to the KD tree.
2) All the neighbours are checked for negative binding energy with the central star.
3) if a bound neighbour is detected, its semi-major axis a is computed.
4) The mass of the pair spread out on a sphere of radius a gives the « binary density »
5) The total mass of the neighbours and the distance to the last one gives the « local density ».
6) If the density ratio binary/local is higher than R_ρ, the pair is validated as a binary.

Fig 4 : A 2D illustration of a KD tree.



The method we use is based on a KD tree. For N particles, the computational time to build the tree scales as N.log(N).

KD trees are built by sorting particles over one dimension after another, splitting space at the median. They allow computationally cheap neighbour search on any star in the cluster. A brute force nearest neighbor search for all N particle scales as N² while KD tree bring it down to N.Log(N).

Testing the algorithm

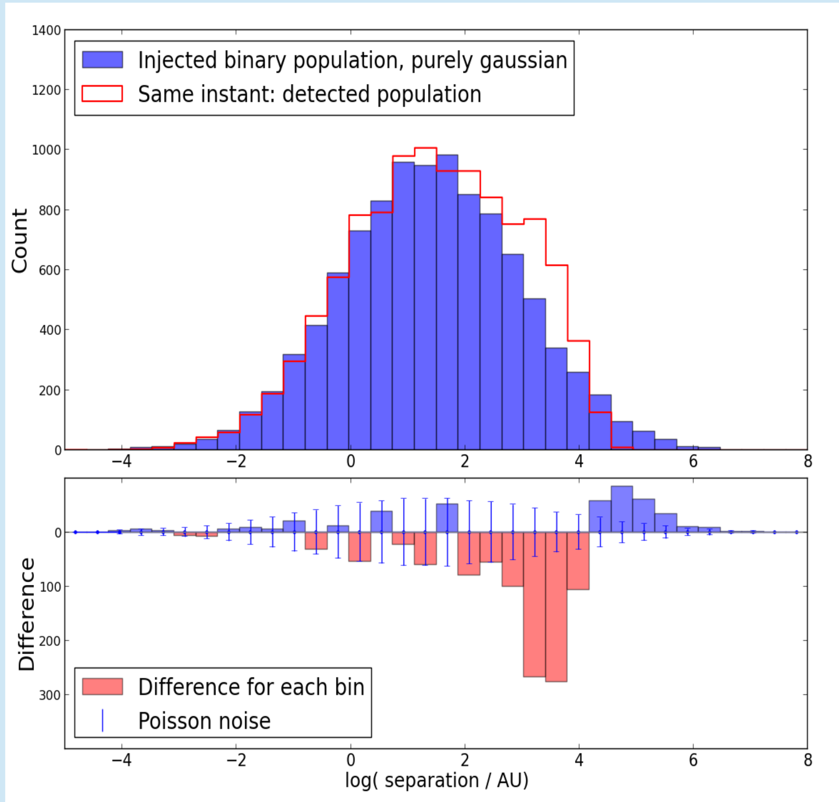


Fig 5 : Separation distribution for 10 000 binaries in a fragmented 30 000 star cluster.

We inject a binary population in a fragmented cluster, then we use our algorithm to detect this population. We get two different semi-major axis distributions, displayed on figure 5.

The overdetection for log(a) ~ 3 is real, these are « spontaneous binaries » dynamically formed, whereas the underdetection for log(a) > 4 means these binaries are in too dense an environment to survive, and the algorithm rejects them.

We can see that the population we want is not always the population we get, as the cluster automatically « pre-process » the binaries : spontaneous binaries appear, and the large separations can be eroded, depending on the local density.

What's next

We are currently working to evaluate the effect of the core-collapse following fragmentation on our binary population. As stated in the introduction, we expect a transition from small, open clusters to large, globular cluster in the effect of the global tidal field. To do so, a consistent way to scale the cluster depending on the number of stars is needed, as binaries separations introduce a real-world value. This is an issue at the injection of the binary population.

In the long run, we wish to introduce gas dynamics to better model young clusters.

References

Aarseth Sverre., 2010, book, Cambridge U. press
Allison Richard J. et al, 2010, MNRAS, 407, 1098A
Goodwin S.P. & Whitworth A.P, 2004,A&A, 413, 929G
Kaczmarek T. et al, 2011, A&A, 528A, 144K
Klessen R. & Burkert A., 2000, ApJS, 128, 287k
Kouwenhoven M. B. N. et al , 2010, MNRAS, 404, 1835K
Kouwenhoven M.B.N. et al, 2007, A&A, 474, 77K
Marks M. & Kroupa P, 2012, A&A, 543A, 8M
Parker Richard J., 2009, MNRAS, 397, 1577P