

SpherIC

Miguel Rocha

September 28, 2009

Abstract

SpherIC is an initial conditions generator for spherical systems in equilibrium with isotropic velocity dispersion. It can generate one-component or two-component systems by sampling self-consistently the phase space distribution function generated from the corresponding density profiles. SpherIC can generate initial conditions for

1. A one-component dark matter system.
2. A one-component stellar system.
3. A stellar system embedded on a larger dark matter system.
4. A dark matter system in which particles are tagged with a mass-to-light¹ ratio, so that the realized light profile follows a given stellar density profile.

SpherIC can write the final output in ASCII, GADGET2, TYPsy or IFRIT formats (HDF5 will be available soon).

1 Generating a one-component dark matter system

The dark matter systems that SpherIC can generate are those with density profiles from the $\alpha\beta\gamma$ -model family. This part of the code was based on the HALOGEN4MUSE code written by Marcel Zemp², who has already written a good documentation about the theory behind HALOGEN4MUSE and hence SpherIC. Thus I am passing the HALOGEN4MUSE documentation together with this documentation. Here I will focus more about the practical aspects of using the code by showing how to set up an example system. An example command line to generate an NFW halo with SpherIC would be

```
spheric -halo -Nhalo 3000000 -Mhalo 1 -a 1 -b 3 -c 1 -rs 3.6 -rcutoff 55 -name nfw.
```

In the above command the option `-halo` specifies that you want to initialize a system with dark matter particles on it, thus the options `-Nhalo`, `-a(α)`, `-b(β)`, `-c(γ)`, `-rs` and `-rcutoff` must be given, otherwise the code will complain. If the total mass `-Mhalo` is not given then it is set to 1 by default. The option `-rcutoff` is not necessary for systems with $\beta > 3$. The usage style and all the options of HALOGEN4MUSE were kept, however some of command line options changed and new ones were added. You can always do `spheric -h/-help` to get a list of the available command line options (see the Appendix).

¹Here I use the term mass-to-light ratio as the probability for a dark matter particle to be a star.

²<http://www.astro.lsa.umich.edu/~mzemp/>

2 Generating a one-component stellar system

In addition SpherIC can generate systems of star particles that have density profiles described by King, Hernquist or Plummer models (Sersic models may be added in the future). In this case the command would look something like

```
spheric -king -Mstar 0.05 -Nstar 1000000 -rc 0.3 -rt 4 -name king
```

for a king model,

```
spheric -plummer -Mstar 0.05 -Nstar 1000000 -rp 0.3 -name plummer
```

for a plummer model, and

```
spheric -hernquist -Mstar 0.05 -Nstar 1000000 -rhern 0.3 -name hernquist
```

for a hernquist model. In any of these cases `-Mstar`(total stellar mass), `-Nstar`(number of stellar particles) and the scale-radius of the system must be given or the code will complain, additionally the tidal radius `-rt` must be given for the King models.

3 Generating a stellar system embedded on a larger dark matter system

To generate a two-component system in which any of the possible stellar distributions of Section 2 are embedded into any of the dark matter distributions of Section 1, one simply has to combine the command line options of Section 2 with those of Section 1. Thus, an example command for generating an spheroidal galaxy with a NFW halo and a stellar component that follows a King profile would be

```
spheric -halo -Nhalo 3000000 -Mhalo 1 -a 1 -b 3 -c 1 -rs 3.6 -rcutoff 55 -king -Mstar 0.05 -Nstar 1000000 -rc 0.3 -rt 4 -name nfwKing
```

Yes, that is a long command!! In the future SpherIC will be able to read a parameter file. For now the command that was used is written in the “<name>.out” file, in case you want to use it to remember what parameters were used, or for copying and pasting when generating similar systems.

IMPORTANT NOTE: It is possible that certain configurations will result in negative stellar distribution functions close to the center of the system. In the above example the distribution function for the stars will be negative for $r \leq 0.045$. In fact, for NFW profiles and stellar profiles that are $\propto 1 - r^2$ at small radii, you will always encounter this problem because the dark matter profile is too steep to physically allow a stellar profile of that kind. When that happens the code will set the distribution function for the stars to 0, resulting in a somewhat steeper profile at very small radii. I have not found this to be an issue for the stability of the systems.

4 Generating a dark matter system in which particles are tagged with a mass-to-light ratio

SpherIC can also initialize dark matter particle systems in which some particles are tagged with a mass-to-light ratio, so that the light profile of the system follows any of the possible stellar profiles of Section 2. This is useful if you want to neglect the contribution of the stars to the potential of the system and want to do a dark matter-only simulation while still interested on the fate of the luminous material. In order to do this you only need to add the option `-nostarpot` to the command line that you would use for generating a stellar system embedded on a larger dark matter system, that is

```
spheric -nostarpot -halo -Nhalo 3000000 -Mhalo 1 -a 1 -b 3 -c 1 -rs 3.6 -rcutoff 55
-king -Mstar 0.05 -Nstar 1000000 -rc 0.3 -rt 4 -name nfwKing
```

When doing this instead of generating star particles the code will calculate mass-to-light ratios for every dark matter particle. The assigned mass-to-light ratios will depend only on the energy of the particle and calculated by

$$\frac{M}{L}(E) = \frac{dM_{\text{dark}}(E)}{dM_{\text{star}}(E)} = \frac{f_{\text{dark}}(E)g(E)dE}{f_{\text{star}}(E)g(E)dE} = \frac{f_{\text{dark}}(E)}{f_{\text{star}}(E)}, \quad (1)$$

where f_{dark} and f_{star} are the dark matter and stellar distribution functions respectively and $g(E)$ is the density of states of the system. In this way the assigned mass-to-light ratio will be equal to the probability for a dark matter particle to be a star, and the corresponding light profile will match that of the stellar profile used to calculate f_{star} , namely a King, Hernquist or Plummer profile.

When using this functionality the mass-to-light ratios will be written on the ASCII initial conditions file together with the particle id's, so the user would need this file to associate particle id's with their corresponding mass-to-light ratios later on the simulation.

5 Units and fine tuning

5.1 Units

SpherIC performs all calculations by setting $G = 1$ and using the system of units that you pass to it on the command line for mass and length. At the end of the calculation it converts all velocities multiplying them by the conversion factor *velConvert* that is defined in the file “definitions.h”. Thus *velConvert* must be equal to the value of \sqrt{G} in the system of units that you are using. For example $\text{velConvert} = 207.398$ when using units of $10^{10} M_{\odot}$ for mass, kpc for length and km/s for velocity. This is how SpherIC is distributed, if you want to use a different system of units you will have to change the value of *velConvert* in the “definitions.h” file.

5.2 Fine tuning

There is a few other definitions that you may want to change on the “definitions.h” file.

NGRIDR and *NGRIDDF* control the number of grid points used for the grid in radius and the grid in the distribution function respectively. These are used for interpolation in the calculation and could be adjusted for testing performance and accuracy.

SpherIC can calculate the density and velocity dispersion profiles of the realized system and write these to an ASCII file that can be later used to compare the realized system with the theoretical models. If you want to do this you have to set the option **-opfs** on the command line and adjust the values of the *ngrid_profile*, *rin_profile* and *rout_profile* definitions which control the number of grid points, the inner radius and the outer radius when sampling.

6 Appendix: Available command line options

- halo : generate a dark matter halo
- Nhalo <value> : number of halo particles
- Mhalo <value> : total mass of the halo (default Mhalo = 1)
- a <value> : alpha parameter in the halo density profile
- b <value> : beta parameter in the halo density profile
- c <value> : gamma parameter in the halo density profile
- rs <value> : scale radius (default rs = 1)
- rcutoff <value> : cutoff radius for cutoff halo models (i.e. beta \geq 3)
- king : generate a stellar component with a King profile
- hernquist : generate a stellar component with a Hernquist profile
- plummer : generate a stellar component with a Plummer profile
- Nstar <value> : number of star particles
- Mstar <value> : total stellar mass
- rc <value> : king core radius
- rt <value> : king tidal radius
- rhern <value> : scale radius for Hernquist profile
- rp <value> : scale radius for Plummer profile
- MBH <value> : mass of black hole (default MBH = 0)
- name <value> : name of the output file
- dx/dy/dz <value> : position offset for the initial conditions (default All = 0)
- dvx/dvy/dvz <value> : velocity offset for the initial conditions (default All = 0)
- ogr : set this flag for outputting the grid in r in an ASCII file
- ogdf : set this flag for outputting the grid in the distribution function in an ASCII file
- ogb : set this flag for generating a GADGET2 initial conditions binary file
- otb : set this flag for generating a TIPSy initial conditions binary file
- oift : set this flag to write positions for IFRIT (binary)
- opfs : set this flag to write a table of density profiles in an ASCII file
- nostarpot : set this flag for excluding the stellar potential
- randomseed <value> : set this flag for setting a value for a random seed (default: random value)
- dorvirexact : set this flag for calculating rvir exactly via N^2 sum - Warning: time consuming for large N!