



RAGA reference

Eugene Vasiliev

Oxford University

email: eugvas@lpi.ru

Version 2.0

February 1, 2017

Contents

1	Introduction	1
2	Obtaining and compiling the software	2
3	Structure of the algorithm	2
4	INI parameters	4

1 Introduction

RAGA¹ is a Monte Carlo code for dynamical evolution of self-gravitating non-spherical stellar systems. The method is described in [1]; here comes a more technical and practical guide.

This document describes the version 2 of the program, which is a nearly complete rewrite of the first version, although built on the same principles. It is now based on the AGAMA library for galaxy modelling [2] and included in its distribution. The main features are:

- Simulation of stellar systems with a much smaller number of particles N than the number of stars in the actual system N_* ;
- Representation of an arbitrary non-spherical potential via a Multipole expansion, with coefficients computed from particle trajectories;

¹Relaxation in Any Geometry

- Two-body relaxation modelled by local (position-dependent) velocity diffusion coefficients (as in Spitzer’s Monte Carlo formulation); the magnitude of relaxation can be adjusted to the actual number of stars in the target system and is not related to the number of particles in the simulation;
- Particle trajectories are computed independently and in parallel, using a high-accuracy adaptive-timestep integrator; the potential expansion and diffusion coefficients are updated at rather long intervals (possibly comprising many dynamical times, but much shorter than the relaxation time);
- Can model the effect of a central massive black hole (capture of low angular momentum stars) and/or a massive black hole binary (scattering of stars and hardening of the binary).

In the present version, RAGA comes as a standalone program which takes an input N -body snapshot and evolves it forward in time, optionally storing N -body snapshots and other parameters at regular intervals during the evolution. In the future we plan to integrate it into the *AMUSE* framework [3].

The main improvements compared to the first version are a more modular design and new, more efficient and robust implementations of potential and relaxation models.

2 Obtaining and compiling the software

RAGA, as part of AGAMA, is available for download at <https://github.com/GalacticDynamics-Oxford/Agama>. The compilation instructions are in the main documentation for the library ([readme.pdf](#)). To compile, one needs the following additional libraries:

- GSL (C math library).
- optional (but recommended): Eigen library for numerical linear algebra.
- optional: UNSIO library – to enable support for GADGET and NEMO N -body snapshot formats; without it only the text format is available for input and NEMO for output.

Check and correct paths to various libraries and compilation flags in the `Makefile` file, then run `make exe/raga.exe`.

3 Structure of the algorithm

The simulation progresses in so-called episodes; the duration of each episode can be much longer than the dynamical time, but shorter than the relaxation time. The episode length is constant for all particles. Particle trajectories are computed independently from each other (`openmp`-parallelized) during each episode, using a high-accuracy, adaptive-timestep 8th order Runge–Kutta integrator. The smooth gravitational potential in which all particles move is represented by a spherical-harmonic (multipole) expansion with the order and symmetries

chosen by the user. The potential is constructed at the beginning of the simulation and (optionally) updated after each episode using trajectories of particles recording during the episode. During the orbit integration, one or more "tasks" can be attached to each particle, collecting data and/or changing the properties of the orbit. After all particles have been processed, each task is performing its own "finalization" step, possibly changing the global properties of the system, and the entire episode is repeated until the end of simulation time.

The available tasks, and the conditions for them to be used, are described below, in the same order as they are invoked during the simulation (the ordering matters).

- *Loss cone treatment*: invoked when there is a single or a binary black hole at origin with a non-zero capture radius. When the distance of closest approach of a particle to the black hole is less than this loss-cone radius, the particle is eliminated from the subsequent simulation and its mass (or some fraction of it) is added to the black hole mass at the end of episode.
- *Binary black hole evolution*: applies when there is a binary black hole in the center. The orbit of the binary is assumed to be Keplerian, oriented in the $x - y$ plane along the x axis. Time-dependent potential causes the particles that approach the vicinity of the binary to change energy and z -component of angular momentum. These changes are recorded, and at the end of the episode the sum of these changes, weighted by particle masses, is used to adjust the orbital parameters of the binary (semimajor axis and eccentricity), using the conservation laws. Optionally, these parameters may additionally change due to gravitational-wave emission.
- *Potential recomputation*: switched on by the corresponding flag in the INI file (on by default). Collect sampling points from each particle's orbit during the episode and use them to update the total potential at the end of the episode. In the simplest case, only the position at the end of episode is used, but more than one sampling point per particle is possible by setting the `numSamplesPerEpisode` parameter – this reduces the discreteness noise in the potential.
- *Relaxation*: turned on by a non-zero value of the `relaxationRate` parameter. The Spitzer's Monte Carlo approach for simulation two-body relaxation consists of adding perturbations to particle velocities during orbit integration, after each internal timestep of the ODE solver. These perturbations are computed from the local (position-dependent) drift and diffusion coefficients, which in turn depend on the distribution function of scatterers. The latter is identified with the entire population of particles in the simulation, but approximated by a spherically-symmetric isotropic distribution function $f(E)$. The amplitude of these coefficients is scaled to the number of stars in the target stellar system being modelled (not necessarily the number of particles in the simulation), see next section. At the same time, samples of particle energies are recorded at regular intervals during the episode and used to recompute the distribution function (in the same way as for the potential, possibly using more than one sample per particle).
- *Trajectory output*: if the output interval is assigned, store the particle positions, velocities and masses in an N -body snapshot file.

4 INI parameters

All parameters are set in the INI file, which should be passed as the one and only command-line argument. The file should contain a section named `[Raga]` with the following parameters (case-insensitive, default values are given in brackets).

- **fileInput** – the input N -body snapshot (required). It may be in any of the formats supported by UNSIO library (e.g., NEMO or GADGET), or – even without this library – a simple text file with 7 columns: 3 positions, 3 velocities, and mass of each particle (not including the central massive black hole).
- **fileLog** (`fileInput.log`) – the name of a text file where the diagnostic information will be written.
- **timeTotal** – the total simulation time (required).
- **timeInit** (0) – initial time, i.e., an offset added to all internal timestamps (useful if continuing a previous simulation).
- **episodeLength** – duration of one episode; if none provided, this means that the entire simulation is performed in a single go. Typically it should be considerably shorter than the timescale on which the system evolves (either the relaxation time or the binary black hole hardening timescale), but may well be longer than the characteristic dynamical time.
- **Symmetry** (`triaxial`) – the type of potential symmetry that determines the choice of non-trivial coefficients in the Multipole expansion. Possible values: `spherical`, `axisymmetric`, `triaxial`, `reflection`, `none`, or a numerical code (see `coords.h`); only the first letter is important.
- **lmax** (0) – the order of angular expansion (should be an even value, 0 implies spherical symmetry).
- **GridSizeR** (25) – size of the radial grid in potential expansion (rarely needs to be adjusted, grid nodes are assigned automatically).
- **Mbh** (0) – mass of a central black hole, or the combined mass of a binary black hole. Its center of mass is fixed at origin, and it adds a Newtonian contribution to the stellar potential. Warning: the input snapshot should not contain the black hole particle.
- **binary_sma** (0) – semimajor axis of a binary black hole (0 means no binary, while a positive value turns on the task of evolving the binary orbit parameters).
- **binary_q** (0) – mass ratio of a binary black hole (components have masses $M_{\text{bh}}q/(1+q)$ and M_{bh}/q , 0 means no binary).
- **binary_ecc** (0) – eccentricity of a binary black hole; its orbit is assumed to lie in $x-y$ plane oriented along x axis.

- **updatePotential** (`true`) – whether the stellar potential is recomputed after each episode.
- **relaxationRate** (`0`) – the amplitude of velocity perturbations that mimic the effect of two-body relaxation. Its numerical value corresponds to $N_\star^{-1} \ln \Lambda$ of the target stellar system. In other words, if one wants to simulate a nuclear star cluster with $N_\star = 10^8$ stars and a massive black hole of $M_\bullet = 10^6 M_\odot$, then the value of Coulomb logarithm is usually determined by the number of stars within the influence radius of the black hole ($\ln \Lambda \simeq \ln M_\bullet/M_\odot \sim 15$), and one should set **relaxationRate**=`1.5e-7`. The crucial feature of the Monte Carlo algorithm is that the actual number of particles in the simulation N may be far less than N_\star . One should keep in mind that even with the relaxation rate set to zero, the recomputation of potential from particles leads to unavoidable discreteness noise, which is however much lower than the level of numerical relaxation in conventional N -body simulations: both the long interval between updates (episode length) and using more than one sample per particle greatly suppress this noise.
- **numSamplesPerEpisode** (`1`) – number of sample points taken from the orbit of each particle during one episode and used in recomputation of the potential and the distribution function; a value > 1 reduces the discreteness noise (a few dozen is a reasonable value).
- **gridSizeDF** (`25`) – size of the grid in energy space used for representing the distribution function.
- **captureRadius**, **captureRadius2** (`0`) – loss-cone radius (or two separate values in case of a binary black hole); particles passing within the given distance from the black hole are captured.
- **captureMassFraction** (`1`) – fraction of mass of captured particles that is added to the mass of the black hole.
- **speedOfLight** (`0`) – if nonzero and a binary black hole is present, determines the loss of energy and angular momentum due to gravitational-wave emission; this parameter specifies the speed of light in N -body velocity units.
- **outputInterval** (`0`) – interval between writing out N -body snapshots, potential expansion coefficients, and relaxation model (should be an integer multiple of episode length, `0` disables this).
- **fileOutput** – file for writing N -body snapshots (if the format is NEMO, all snapshots are stored in a single file, otherwise each one goes into a separate file with the timestamp appended to the file name).
- **fileOutputFormat** (`text`) – output format for snapshots (`text`/NEMO/GADGET, the latter is available only with the UNSIO library; only the first letter matters).
- **fileOutputPotential** – base filename for writing the coefficients of Multipole expansion (timestamp appended to the name).

- `fileOutputRelaxation` – base filename for writing the table with parameters of the spherical model used to compute diffusion coefficients (timestamp appended to the name).
- `fileOutputLosscone` – file for storing the list and parameters of particles captured by the black hole(s).
- `fileOutputBinary` – file for storing the orbital parameters of the binary black hole (semimajor axis and eccentricity) after each episode.

References

- [1] Vasiliev E., 2015, MNRAS, 446, 3150
- [2] Vasiliev E., in prep.
- [3] Portegies Zwart, S., McMillan, S., van Elteren, E., Pelupessy, I., de Vries, N. 2013, Comp.Phys.Comm., 184, 456