Tutorial

Zhang Fupeng

August 16, 2023

GNC is a Monte-Carlo numerical method to calculate the dynamical evolution of a cluster of particles around a massive black hole using Forkker-Planck methods. The code is written fully in Fortran free format. If you have used the code, please cite the following reference:

Zhang, fupeng, Ameau-senaro Pau, arXiv:2306.03924

1 Installation

The code is found working under the following environment

- gfortran 11.3.0 + open mpi 4.1.2.
- ifort 14.0.2 + mpich 3.1.3

Other versions of fortran compiler or mpi haven't been tested, probably work, however.

Before installation of the code, we recommend strongly you install HDF5 library, and the tools to view HDF5.

Add the following line into your $\sim$ /.bashrc or $\sim$ /.bash_profile to set a larger limit of stack size:

```
ulimit -s 2048000
```

Install execstack

```
bshell > sudo apt-get install execstack
```

Now go to "source" directory

```
bshell > cd source
```

Open makefile, if you are using ifort, change the line

```
COMPILE=gfortran
```

to `COMPILE=ifort`. Otherwise keep it unchanged.

If HDF5 library is installed on your system, make sure in makefile set

```
use_hdf5=1
```

and remember to specify the path to the library and include files of HDF5 by changing the following line:

```
hdf5dirlib=/usr/lib/x86_64-linux-gnu/hdf5/serial/
hdf5dirinc=/usr/include/hdf5/serial/
```

If you are not using HDF5, then simply set `use_hdf5=0`.

If you want the maximum compatible between the results of GNC compiled by gfortran and ifort, then set

```
maximum_compatible=1
```

If you are using ifort only, then you can set it to be zero such that the program may run a little faster.

After these configurations, now you are ready to make files

```
bshell > make
```

You can now change to the the directory "main"

```
bshell > cd ../main
```

open `Makefile`, and specify `COMPILE`, `use_hdf5`, `hdf5dirlib`, `hdf5dirinc`, and `maximum_compatible` to the same value as those of the makefile in the source directory.

Now compile the programs by

```
bshell > make -B
```

2 Generate an auxiliary file

Before running the programs, you need to generate an auxiliary file using "cfuns" for fast computation of the diffusion coefficients. It can be done by running

```
bshell > mpirun -np (num_cores) ./cfuns (num_bins)  (log j_min)  (log j_max)
     (output_file_name)
```

where

- num_cores: The number of coresthreads you want to run cfuns. Usually cfuns will takes about a few minutes. Using multiple cores can make it much faster. Make sure that num_bins/num_cores= an integer number. For example, if num_bins=1024, num_cores can be 8 or 16.
- num_bins: The number of bins, usually 1024 is enough.
- log j_min: The log10 of minimum dimensionless angular momentum parameter, usually -3 is enough. However, if the loss cone you considered is very small (such as those of the compact objects), you should make it smaller, e.g., to be -3.4.
- log j_max: The log10 of maximum dimensionless angular momentum parameter, usually set to 0.
- output_file_name: the name of the file contains the results of the auxiliary information. ".bin" will be attached to the output_file_name automatically.

For example, in most cases you can run cfuns by the following parameter

```
bshell > mpirun -np 8 ./cfuns 1024 -3.5 0 ../common_data/cfuns_34
```

Note that cfuns need to run ONLY ONCE. The file generated by cfuns can be used for any computation of GNC in the future.

3 Some examples of *GNC*

The "examples" directory contains some files to show how to run the program, where

- 1comp is an equal mass star cluster with MBH $M_\bullet = 4 \times 10^6 m_\odot$, without loss cone;
- 1comp_lc is similar to 1comp, but considering loss cone;
- 5comp_lc is a cluster containing 5-component model, with stars, brown dwarfs, white dwarfs neutron stars and black holes.

The components and some settings of the program are shown in the file model.in and mfrac.in:

- The model.in file: contains the setting of parameters of the ini and main program.
- mfrac.in file: contains the setting of parameters of the mass components.

Each of these two files contain detailed comments that explain parameters. The comments are lines start with "#" symbol.

4 Run the main loop and get the results

find the "run.sh" in each example directory, and in the command line run

```
bshell >./run.sh
```

When the computation is done, we can get a number of HDF5 file named "dms_*_*.hdf5" in the directory "output/ecev/dms/", which contains some information in each of the snapshot, including

- fMa: Cumulative mass of component as function of radius in unit of r_h;
- fNa: Cumulative number of component as function of radius in unit of r_h;
- fden: Number density distribution of components as function of radius in unit of r_h, calculated according to the distribution of $g(x)$ and assuming isotropic distribution of angular momentum j;
- fden_simu: similar to fden, but calculated according to the dimensionless energy x and angular momentum j of each of the simulated particles;
- fgx: the dimensionless distribution of particle $\bar{g}(x)$;

- gxj: Two-dimenional dimensionless distribution of $x - j$ of each component.
- dej: contains different terms of diffusion coefficients as function of x and j.

The binary files in each snapshot of the simulation are saved in the directory "output/ecev/bin/single/". If you want to read these files directly, please see the source file "md_particle_sample.f90" for more details.

You can goto the plot directory, and use python3 to plot the results:

```
bshell >python3 ge.py
```

The results of 1comp, 1comp_lc, 5comp_lc are shown in Figure 1.

program pro provide some statistics of the number of particles in each snapshot of the simulation:

```
bshell >./pro
```

It will generate some files in each different kinds of stars in directory "output/pro", 'BD' for brown dwarfs, 'BH' for stellar mass black holes, 'MS' for main sequence stars, 'NS' for Neutron stars, 'WD' for white dwarfs.

In file _event_N.txt

- Tsnap: The time of snapshot ends;
- N_all: The number of all samples calculated during the snapshot (average by tasks, the same for all numbers below);
- N_norm: The number of samples that remained (exited due to end of simulation time) after the snapshot;
- N_norm_bd: Similar to N_norm but only samples within the boundary;
- N_emax: The number of samples passed through the inner boundary during the snapshot;
- N_td: The number of samples pass through loss cone during the snapshot;
- N_tdfull: Simlar to N_td but only for samples pass through full loss cone;
- N_tdempty: Simlar to N_td but only for samples pass through empty cone;

In file _event_N_weight.txt: similar to _event_N.txt but we count the weighted number of particles.

In file _rates.txt:

- Tsnap: The time of snapshot ends;
- R_emax: The rates of samples passed through the inner boundary during the snapshot;
- R_td: The rates of samples pass through loss cone during the snapshot;
- R_tdfull: Similar to R_td but only for samples pass through full loss cone;
- R_tdempty: Similar to R_td but only for samples pass through empty cone;

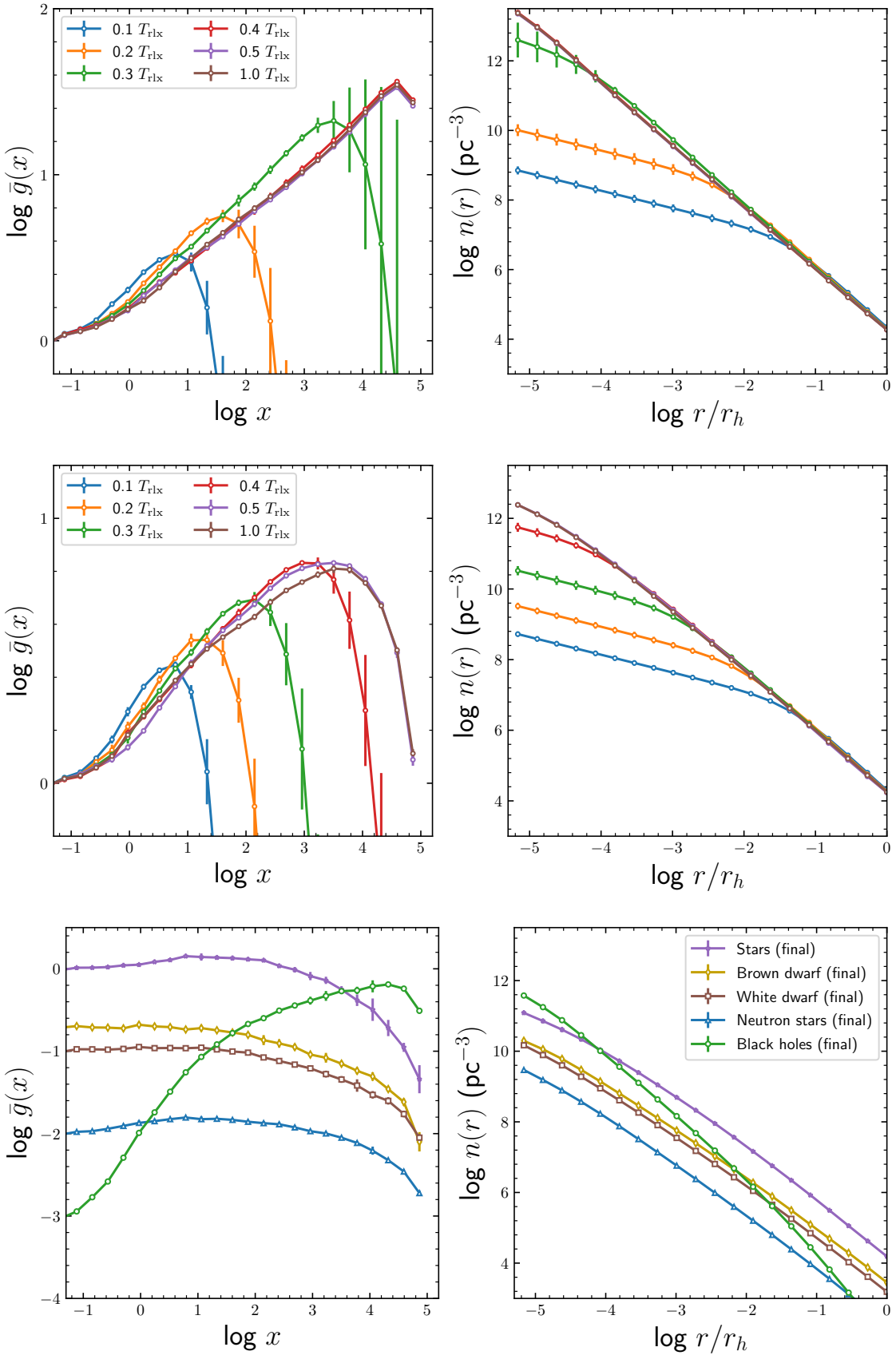

Figure 1: Top panel: Distributions of $\bar{g}(x)$ and number density in example "1comp". Lines with different colors are results at different times of simulation (in unit of two body relaxation time at r_h, i.e., $1.0T_{\rm rlx}$); Middle panel: Similar to top panel but for example "1comp_lc"; Bottom panel: Results of final distribution (at 1.0 $T_{\rm rlx}$) in example "5comp_lc". Lines with different colors are results of different stellar types.