NIGO User Manual

Luca J. Rossi *

Centre for Astrophysics and Supercomputing, Swinburne University of Technology, Hawthorn, VIC 3122, Australia

1 Introduction

The present document is the user manual of NIGO, a Numerical Integrator of Galactic Orbits. The main functionality of the code is to integrate the orbits of test particles moving within the gravitational potential generated by a multi–component galaxy.

An overview of the code structure, main algorithms and performance is presented in Rossi (2014). This user manual includes the main steps to install the code on laptops or workstations hosting a UNIX operating system, a description of the input files, initial conditions, outputs and a brief description of the subroutines of NIGO. The code has been written in FORTRAN90.

2 Install NIGO

This section contains the instructions for a basic installation of the code. NIGO is supported by a few external open source libraries, namely the SLATEC, ODE and ASA147 libraries, and they are provided with the NIGO package. The first step to get started is to compile the libraries. We provide a `bash` script (`compile_LIBS.sh`) that compiles the libraries and moves the executables in the directory containing the source codes, but of course the user can choose the best way to do this step according to the particular needs of the case. In the same way, the code can be compiled running the script `compile_NIGO.sh`. Before to proceed with the installation of the code, the user should edit the following scripts

- `compile_LIBS.sh`
- `compile_NIGO.sh`
- `template.txt`

In particular, the user has to specify the path of the directory containing the executables. All the previous scripts contain examples of paths and comments, so the user shouldn't have too many troubles in the editing process. To make the scripts executable simply write on terminal

```
chmod +x /path/to/directory/compile_LIBS.sh
chmod +x /path/to/directory/compile_NIGO.sh
chmod +x /path/to/directory/run_NIGO.sh
```

To compile the libraries, type

```
/path/to/directory/compile_LIBS.sh
```

To compile the code, type

```
/path/to/directory/compile_NIGO.sh
```

At this point the code should be correctly installed and ready to go. To run NIGO make sure the the directory containing the NIGO executable includes the input files `InputN.dat` and `InputS.dat`, edit the file `run_NIGO.sh` to chose the number of threads on which the integration process will be distributed and simply type

*Email: `lucarossi@swin.edu.au`

```
/path/to/directory/run_NIGO.sh
```

3 Input parameters for the integrator and for the mass model

The file `InputN.dat` contains the instructions for the orbit integrator and the definition of the parameters of the mass model. The meaning of the variables to be edited in the file `InputN.dat` is explained in detail in Rossi (2014). A brief description can be found in the subroutine `define.f90`. An example of `InputN.dat` is

```
N_STAR   T_START  T_STOP   DELTA_T  ABSERR
3        0        1000     500      1e-10

R0       V0
8.5      220.0

MBULGE1  BB1
0.0e0    2.7

MBULGE2  BB2
0.0e0    0.42

MSER     RE       NSER
3.0e+09  2.7      1.0

MDISC1   AD1      BD1
7.7e+10  5.8      0.3

MDISC2   AD2      BD2
-6.8e10  17.43    0.3

MDISC3   AD3      BD3
2.6e+10  34.84    0.3

MBAR1    ABAR1    BBAR1    CBAR1    NBAR1    OMEGA_B1 PHI0_B1
9.8e09   7.0      2.8      2.0      2.0e0    55.9     25.0

MBAR2    ABAR2    BBAR2    CBAR2    NBAR2    OMEGA_B2 PHI0_B2
0.0e0    3.5      1.4      1.0      2        55.9     0.0

HCHOICE  VHL      AHL      MHALO    AH       MSERH    REH      NSERH
2        220.0    7.9      3.3e+12  45.02    3.3e+12  45.02    4.0

NSP      RS       ISP      OMEGA_SP PHI0_SP  LSP      ZSP      ASP      DELTA_SP
4        3.5      11.6     55.9     25.0     10       0.3      0.5      1.0e-02
```

In this particular example the code will integrate the o rbit of 4 stars with an accuracy of 10^{-10} for 1000 Myr, printing the ephemeris every 100 Myr. We refer to the subroutine `define.f90` or to Rossi (2014) for a brief or for a comprehensive description of the meaning of the model parameters, respectively.

4 Initial conditions

The input file `InputS.dat` contains the initial state vector of the particles, expressed in a phase–space 6D right–handed galactocentric Cartesian frame of reference. The first three components of the initial state vector are the spatial coordinates (x, y, z) expressed in kpc, while the last three components are the velocities $(\dot{x}, \dot{y}, \dot{z})$ expressed in km s^{-1}. An example of `InputS.dat` file for three stars can be found below.

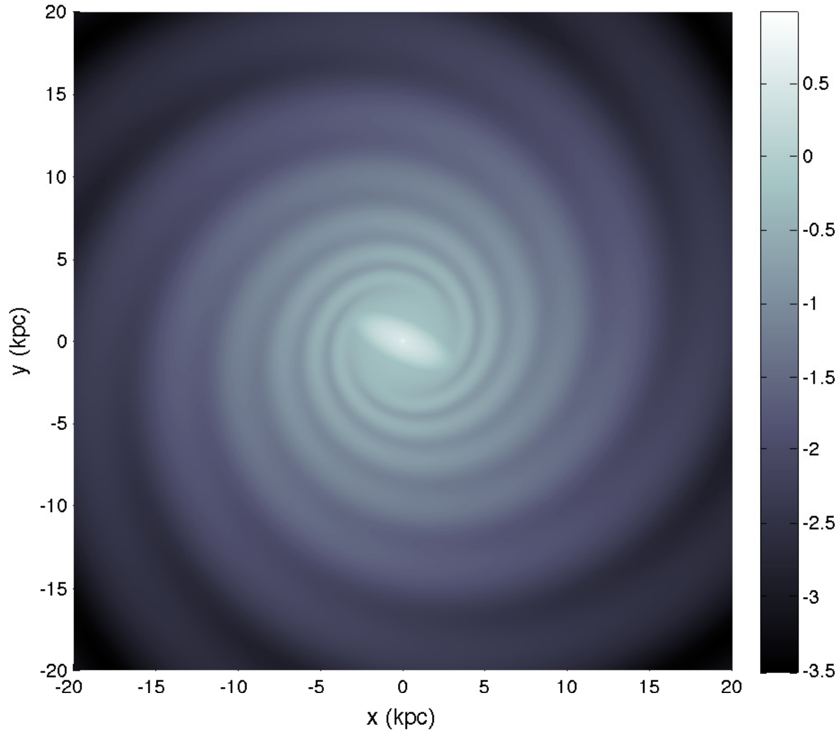

Figure 1: Mass model generated by `Plot_density.m`.

```
-0.92    3.10    1.95    142.97    43.80    -1.49   # Star 1
-1.36    3.08    1.95    133.19    58.00     1.30   # Star 2
-3.62   -0.40    1.94   -10.15     173.91    0.26   # Star 3
```

5 Outputs

The code produces 2 output files. One of them is a log file, containing information on the computation time, number of stars and number of threads used. The second file is called `Orbits.dat`, and contains the ephemeris of the particles. An example of output file for a simulation integrating the orbits of three stars is shown below.

```
0.00       -0.92    3.10    1.95    142.97     43.80     -1.49    # Star 1, t = 0.0 Myr
0.00       -1.36    3.08    1.95    133.19     58.00      1.30    # Star 2; t = 0.0 Myr
0.00       -3.62   -0.40    1.94    -10.15    173.91      0.26    # Star 3, t = 0.0 Myr
500.00     -0.99   -0.73   -1.55   -100.54    109.18      7.69    # Star 1, t = 500.0 Myr
500.00     -0.52   -1.27   -1.49   -143.14      9.30     39.07    # Star 2, t = 500.0 Myr
500.00      1.96   -2.13   -1.65   -154.12   -104.80    -40.55    # Star 3, t = 500.0 Myr
1000.00     3.40   -0.59    1.98    -30.78   -148.66     14.58    # Star 1, t = 1000.0 Myr
1000.00     1.62   -0.92   -0.23    -12.29   -131.55   -180.15    # Star 2, t = 1000.0 Myr
1000.00     3.27    3.21    2.02    113.02   -132.65    -52.87    # Star 3, t = 1000.0 Myr
```

6 Output reduction and visualization

NIGO is provided with a package of MATLAB® subroutines to produce basic plots of the results. The main subroutines that the user can adopt to visualize the results of the simulations are `Plot_density.m`, `Plot_orbit.dat` and `Plot_galaxy.dat`. The script `Plot_density.m` produces the logarithmic color–map of the density field chosen by the user. The units are in $M_{\odot}$ pc^3. Figure 1 shows the mass model generated by the input file `InputN.dat` shown above. The script `Plot_galaxy.m` produce the smoothed 2D plots shown in Figure 2. Finally, the script `Plot_orbit.dat` is useful to visualize the orbital evolution of a single particle. Figure 3 shows an example. The tools presented can be used only for a basic visualization and

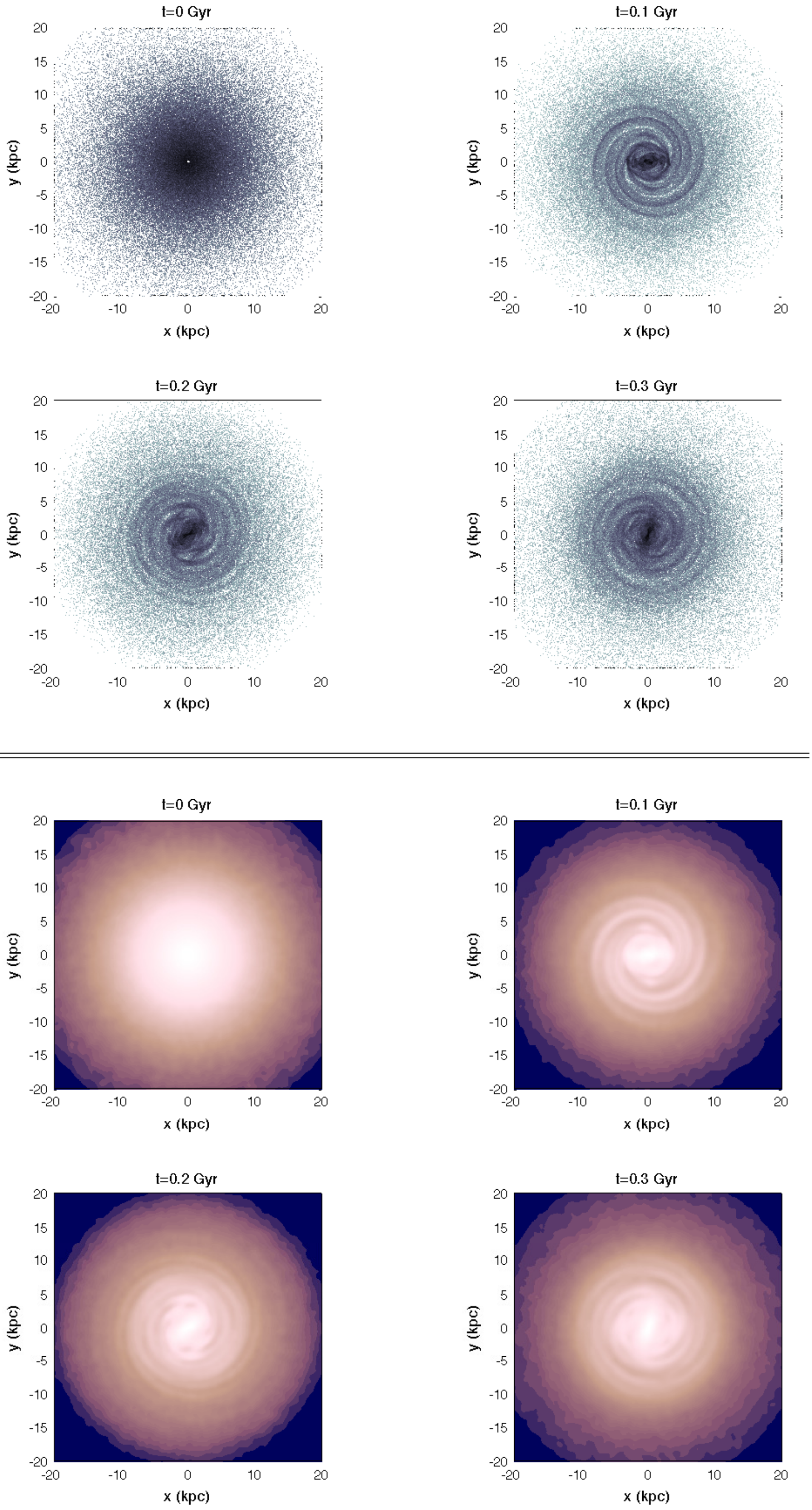

Figure 2: Evolution of the test particles for different values of the smoothing parameter.

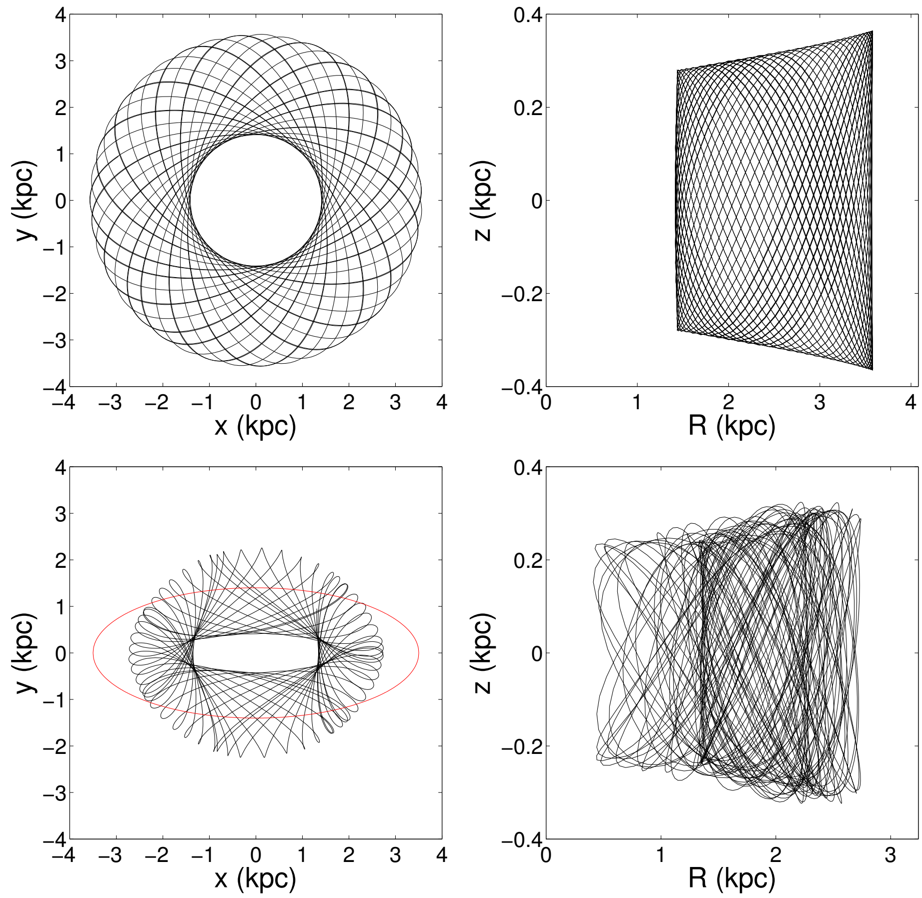

Figure 3: Orbit of a star projected on the galactic plane and meridional plane moving within an axisymmetric mass model (top panels) and within an non–axisymmetric rotating mass model (bottom panels). Note that in the case of the barred potential we plotted the orbit in the non–inertial frame of reference corotating with the bar.

reduction of the outputs, and the users might be interested in developing their own macros to study the particular problem they aim to address.

7 NIGO subroutines

This section contains a brief description of the subroutines of NIGO.

Subroutine	Called by	Description
`bar_coefficients.f90`	`fbar.f90`	Coefficients defining the force due to a Ferrer's ellipsoid
`define.f90`	`main.f90`	Definition of the parameters used by the code
`eom_integration.f90`	`nigo.f90`	Integration of the equations of motion
`eom.f90`	`ode.f90`	Definition of the equations of motion
`fbar.f90`	`eom.f90`	Force due to the Ferrer's bar
`fbulge.f90`	`eom.f90`	Force due to the Miyamoto–Nagai disc
`fhalo.f90`	`eom.f90`	Force due to the dark matter halo
`find_lambda.f90`	`bar_coefficients.f90`	Compute the lambda parameter
`fsersic.f90`	`eom.f90` & `fhalo.f90`	Force due the the Sérsic mass distribution
`fspiral.f90`	`eom.f90`	Force due to the spiral perturbation of the disc
`main.f90`		Main program of NIGO
`nigo.f90`	`main.f90`	Read the initial conditions and call the integrator
`parameters.f90`	`define.f90`	Module with the common variables used by NIGO
`wtime.f90`	`nigo.f90`	Return a reading of the wall clock time

In subdirectory `Test` the user can find the input files and the output files of a test run that can be used to check the correct behaviour of NIGO.