Technical documentation of BHINT v0.9

Ulf Löckmann

March 20, 2009

This is the technical documentation of BHINT, a novel post-Newtonian integrator for stellar systems surrounding a SMBH (Löckmann & Baumgardt 2008).

The latest version of the software as well as this documentation is available at `http://www.astro.uni-bonn.de/english/downloads.php`.When using BHINT, please cite Löckmann, U., & Baumgardt, H. 2008, MNRAS, 384, 323.

Contents

1 Installation

To install BHINT, extract the archive `bhint.tgz` to create a directory `bhint` of the following content:

```
linux:~$ ls bhint
bhi_config.h  bhi_io.c      bhi_sse.c       bhi_vector.c
bhi.h         bhi_kepler.c  bhi_timestep.c  makefile
bhi_int.c     bhi_main.c    bhi_util.c      sse
```

To create the executable `bhint`, change into the directory and run `make`:

```
linux:~/bhint$ make
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_int.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_io.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_kepler.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_sse.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_timestep.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_util.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_vector.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_main.c
gcc -o bhint bhi_int.o  bhi_io.o  bhi_kepler.o bhi_sse.o
  bhi_timestep.o  bhi_util.o  bhi_vector.o bhi_main.o
  -L . -L /opt/SUNWhpc/lib -lm
```

1.1 Speeding up integrations with *GRAPE*

To benefit from the capabilities of the Hermite integration scheme, we utilised GRAPE6 boards (Makino et al. 2003) for hardware acceleration. The following *GRAPE* routines as specified under `http://www.artcompsci.org/~makino/softwares/GRAPE6/grape6user/node6.html` are used in module `bhi_int.c`:

```
g6_set_j_particle()
g6_open()
g6_npipes()
g6_set_tunit()
g6_set_xunit()
g6_set_neighbour_list_sort_mode()
g6_set_ti()
g6calc_firsthalf()
g6calc_lasthalf2()
g6_read_neighbour_list()
g6_get_neighbour_list()
g6_close()
```

Any library implementing this interface can be employed by adding it to the `LIBS` entry in the `makefile`:

```
LIBS=-lgrape
```

and uncommenting the `USE_GRAPE` entry in the header file `bhi_config.h`

```
#define USE_GRAPE
```

It is necessary to `make` the project after any such changes.

1.2 Stellar evolution

We modelled mass loss due to stellar evolution using the SSE package by Hurley et al. (2000), which can be obtained at `http://astronomy.swin.edu.au/~jhurley`. SSE is meant for single stars only; however, as an approximation, we also use it for stars marked as binaries. In this case, either of the two stars is evolved as if isolated, with an initial mass of half the binary mass. Within BHINT, SSE can be restricted to a subset of the particles (e.g. only disc stars as opposed to background SBHs). For details on the configuration of SSE see Section 4.5.

The sources of SSE are included in the subdirectory `sse`. Two modifications have been made to `evolv1.f` to make it usable by BHINT: First, the following lines in bold face were added after the declarations:

```
      ...
      REAL scm(50000,14),spp(20,3)
      COMMON /SINGLE/ scm,spp
*
      INTEGER bhflag,ceflag,tflag,ifflag,nsflag,wdflag
      COMMON /FLAGS/ ceflag,tflag,ifflag,nsflag,wdflag
      real*8 sigma,dtmnew
      COMMON /VALUE4/ sigma,bhflag

      neta = 0.5d0
      bwind = 0.0d0
      hewind = 0.5d0
      sigma = 190.d0
      ifflag = 0
      wdflag = 1
      bhflag = 0
      nsflag = 1
      mxns = 3.0d0
      pts1 = 0.05d0
      pts2 = 0.01d0
      pts3 = 0.02d0

      dtm = 0.d0
      r = 0.d0
      ...
```

This sets the configuration parameters for SSE, which are usually read from an input file in `sse.f` (see there for details on the various parameters). In addition, to give BHINT control over the SSE time step, a parameter `dtmnew` was introduced and assigned the time step determined:

```
      SUBROUTINE evolv1(kw,mass,mt,r,lum,mc,rc,menv,renv,ospin,
     &                  epoch,tm,tphys,tphysf,dtp,z,zpars,dtmnew)

          ...
          dtm = MAX(dtm,1.0d-07*aj)
          dtmnew = dtm
          dtm = MIN(dtm,tsave-tphys)
          ...
```

To use SSE, add the `sse` library to the `LIBS` entry in the `makefile`, along with any Fortran libraries needed (depending on the system architecture, this may be `libgfortran` or `libg2c`):

```
LIBS=-lsse -lgfortran
```

uncomment the `USE_SSE` entry in the header file `bhi_config.h`:

```
#define USE_SSE
```

and `make` the project:

```
linux:~/bhint$ make
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_int.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_io.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_kepler.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_sse.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_timestep.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_util.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_vector.c
gcc -Wall -O3 -I /opt/SUNWhpc/include -c bhi_main.c
make[1]: Entering directory `~/bhint/sse'
gcc -c  sse.f
gcc -c  deltat.f
gcc -c  evolv1.f
gcc -c  hrdiag.f
gcc -c  kick.f
gcc -c  mlwind.f
gcc -c  mrenv.f
gcc -c  ran3.f
gcc -c  star.f
gcc -c  zcnsts.f
gcc -c  zfuncs.f
ar -r ../libsse.a *.o
ar: creating ../libsse.a
make[1]: Leaving directory `~/bhint/sse'
gcc -o bhint bhi_int.o  bhi_io.o  bhi_kepler.o bhi_sse.o
  bhi_timestep.o  bhi_util.o  bhi_vector.o bhi_main.o
  -L . -L /opt/SUNWhpc/lib  -lsse -lgfortran -lm
```

2 Units

BHINT uses different unit systems for external communication and internal calculations.

The external system of units describes the units in all input files and output. Here we use physical units, which are convenient for use in systems like the Galactic Centre:

- All masses are given in $M_\odot$,
- all distances in pc,
- all times in yr and hence
- all velocities in $\mathrm{pc\,yr^{-1}} \approx 9.78 \times 10^6\,\mathrm{km\,s^{-1}}$.

The units are defined in `bhi.h`. The internal system of units is chosen such as to increase the computational speed by setting the gravitational constant $G = 1$. It is based on AU and day, and includes scaling factors to meet the requirements of the *GRAPE* hardware:

Dimension	**Value**			
mass	$100^2/G$	$\mathrm{AU^3\,day^{-2}}$	$\approx 3.38 \times 10^7$	$M_\odot$
length	32^2	AU	$\approx 4.96 \times 10^{-3}$	pc
time	$32^3/100$	day	$\approx 8.97 \times 10^{-1}$	a
velocity	$100/32$	$\mathrm{AU\,day^{-1}}$	$\approx 5.41 \times 10^3$	$\mathrm{km\,s^{-1}}$

The scaling factors 32 and 100 for length and mass, which are used for time and velocity as well, can be changed in `bhi.h`:

```
#define CONVINT_X_AU_SQRT       32.
#define CONVINT_M_AU3DAY2_SQRT 100.
```

3 Usage

3.1 Input data

The input file of a BHINT integration contains masses m_i, positions $\boldsymbol{x}_i = (x_i, y_i, z_i)$, and velocities $\boldsymbol{v}_i = (u_i, v_i, w_i)$ of all N particles $n_0 \ldots n_{N-1}$ (seven columns, one particle per line). Additional comment lines start with `#`. The first particle n_0 is the massive central particle, which has to be at rest at the origin, i.e. $x_0 = y_0 = z_0 = u_0 = v_0 = w_0 = 0$. The subsequent particles are preferentially sorted in groups of decreasing "importance" to the model: For example, stars which need to be evolved using SSE or for which detailed output is desired should be provided before background SBHs. An IMBH included in a model will usually be the second particle n_1, for which orbital parameters are also provided in regular output (see Section 3.3).

All values are given in external units ($\mathrm{M}_\odot$, pc, $\mathrm{pc\,yr^{-1}}$). All masses have to be non-zero; negative masses $m_i < 0$ mark a particle as a binary of two stars each of mass $-m_i/2$ (only allowed for $i > 0$).

The following example `kozai.dat` of a Kozai configuration contains a $3.5 \times 10^6\,\mathrm{M}_\odot$ SMBH at the centre, a binary of two $10\,\mathrm{M}_\odot$ stars orbiting at semi-major axis $a = 0.04\,\mathrm{pc}$ and eccentricity $e = 0.9$, and an IMBH of $15\,000\,\mathrm{M}_\odot$ on a circular orbit with semi-major axis $a \approx 0.16\,\mathrm{pc}$.

```
# SMBH at centre
3.5e6   .0       .0       .0       .0       .0       .0
# binary of two 10 M_sun stars, a=0.04 pc, e=0.9
-20.   -1.74e-2  6.45e-3  4.95e-2 -3.25e-4  2.98e-5  3.10e-4
# IMBH of 15,000 M_sun, a=0.16 pc, e=0
15000.  5.58e-3  1.60e-1  .0       .0       .0       3.14e-4
```

3.1.1 Creating a model with BHINT

BHINT has the capability of creating an input model for an integration using the command `m`. The model is written to standard output and may be diverted into a file. The syntax is

`bhint m` N m_0 γ $a_{\min}$ $a_{\max}$ $e_{\min}$ $e_{\max}$ $m_{\min}$ $m_{\max}$ $[\alpha\ [\sigma\ [\delta\ \{m^i\ \alpha^i\}^*]]]$

where the parameters are as follows:

Parameter	Description
N	Number of particles (including central massive body). Requires $N > 2$.
m_0	Mass of central particle (in $\mathrm{M}_\odot$). Requires $m_0 > 0$.
γ	Power-law index of particle distribution. The particles' semi-major axes a_i will be distributed as $f(a) \propto a^\gamma$. Note that γ does not necessarily reflect the three-dimensional density profile $\rho(r)$, since particles moving on eccentric orbits have varying radial distance, which changes ρ near $a_{\min}$ and $a_{\max}$. For a disc with surface density profile $\propto a^\beta$, $\gamma = \beta - 1$.
$a_{\min}$	Minimum semi-major axis. Requires $a_{\min} \geq 0$.
$a_{\max}$	Maximum semi-major axis. Requires $a_{\min} \leq a_{\max}$ and $a_{\max} > 0$.
$e_{\min}$	Minimum eccentricity. Requires $e_{\min} \geq 0$.
$e_{\max}$	Maximum eccentricity. The particles' eccentricities are distributed as $f(e) \propto 2e$, which corresponds to an isotropic configuration for $e_{\min} = 0$, $e_{\max} = 1$. Requires $e_{\min} \leq e_{\max}$.
$m_{\min}$	Minimum particle mass. Requires $m_{\min} > 0$.
$m_{\max}$	Maximum particle mass. Requires $m_{\min} \leq m_{\max}$.
α	Optional parameter for a power-law IMF. The particle masses m_i will be distributed as $f(m) \propto m^{-\alpha}$. Defaults to 0.

Parameter	Description
σ	Optional parameter for creating a disc. If specified and not equal to "_", opening angle (standard deviation) of stellar disc; requires $\sigma \geq 0$. Otherwise, a spherically symmetric system is created (default).
δ	Optional parameter for inclined discs (useful for creating multiple disc systems). Inclination of the system (disc) to the xy-plane in degrees. Choose 180 for a counter-rotating disc. Defaults to 0.
m^i	Optional intermittent mass limits for multi-part power-law IMF. Requires $m_{\min} < m^1 < \cdots < m^P < m_{\max}$.
α^i	Optional successive IMF slopes for multi-part power-law IMF. For a number P of additional (m^i, α^i) pairs $(m^1, \alpha^1) \ldots (m^P, \alpha^P)$ provided, the IMF slope α^i is valid in the interval $\left[m^i \ldots m^{i+1}\right]$ $(0 \leq i \leq P)$, with $m^0 = m_{\min}$, $\alpha^0 = \alpha$, and $m^{P+1} = m_{\max}$.

Example: The following command creates an isotropic, non-rotating, spherical cusp of 10 000 stars of 10 $M_\odot$ each around a SMBH of 4×10^6 $M_\odot$:

```
bhint m 10000 4.e6 -1.2 0 .1 0 1 10 10 > dat/model1.dat
```

Example: The same as above, but with stars following the canonical IMF (Kroupa 2001):

```
bhint m 10000 4.e6 -1.2 0 .1 0 1 .01 120. .3 _ 0 .08 1.3 .5 2.3
```

Example: A (almost) flat and circular disc of stars following a Salpeter (1955) IMF, with surface density $\rho_{2D}(r) \propto r^{-2}$, 0.1 pc $< r <$ 0.5 pc, and opening angle $\sigma = 1.4°$:

```
bhint m 5000 4.e6 -3 .1 .5 0 .04 .4 10. 2.3 1.4
```

Example: A similiar disc at an inclination of 40°.

```
bhint m 5000 4.e6 -3 .1 .5 0 .04 .4 10. 2.3 1.4 40
```

To create a two-disc system, simply concatenate the results of the previous two examples, and delete the second SMBH entry.

3.2 Starting a BHINT integration

The syntax of starting a BHINT integration (command `i`) is

`bhint i` *infile* N_{orbits} N_{steps} *prefix*

where the parameters are as follows:

Parameter	Description
infile	Path and name of input file (model).
N_{orbits}	Number of orbits of first particle until integration ends (assuming a constant orbital period). If $N_{\text{orbits}} < 0$, integration ends after time $t_{\text{end}} = -N_{\text{orbits}}$ yr.
N_{steps}	Number of regular output lines. If $N_{\text{steps}} = 0$, output is performed every time all particles have been moved. If $N_{\text{steps}} < 0$, output is performed after every $-N_{\text{steps}}$ completed revolutions of first particle.
prefix	Path and prefix for output file names.

Example: The following command starts an integration of model `kozai.dat` for 10^7 yr, creating ten lines of output (i.e. every Myr):

```
bhint i dat/kozai.dat -1.e7 10 out/kozai_run1_
```

Example: The following command runs approximately 25 000 orbits of the first particle, creating output after every 1000 revolutions:

```
bhint i dat/kozai.dat 2.5e4 -1000 out/kozai_run2_
```

3.3 Output files

The names of all output files are

*prefix infile*_η_N_{step}_N_{orbits}`.bhint.`*ext*,

where *prefix*, *infile*, and N_{orbits} are as in Section 3.2, η and N_{step} are precision parameters explained in Section 4.1, and *ext* one of the following extensions.

3.3.1 `out`: Regular output

The `out` file lists some of the configuration constants used in the integration as explained in Section 4. After that, output is performed according to the output interval as specified in Section 3.2, and only at synchronised times (i.e. when all particles have moved and thus have the same particle time).

The output includes the following information:

1. Current time
2. Number of particles
3. Current relative energy error per time unit
4. Total relative energy error
5. Fraction of energy removed from system (sticky-particle mergers, escapers, gravitational wave emission)
6. Emitted energy (in units of $M_\odot c^2$)
7. Time since last output
8. Average number of steps since last output
9. Average time step
10. Average block step
11. Progress in time units per wall clock second
12. Maximum eccentricity among all particles
13. Minimum semi-major axis among all particles
14. Inclination between first two orbits in list
15. Angle between first two particles' position vectors
16. Eccentricity, of the *output particle*
17. Semi-major axis of the *output particle*
18. Central distance of the *output particle*
19. Energy of the *output particle*
20. Time step of the *output particle*

The *output particle* is the particle with the name stored in `P_OUTPUT`, which is defined in `bhi_io.c`, and defaults to `1`.

3.3.2 `detail`: Detailed particle output

The `detail` file contains periodic output of particle properties. The output can be restricted to particles with names $i \leq$ `N_MAX_DETAIL`, which is defined in `bhi_config.h`. If set to -2, details are printed for all particles.

The output consists of one line per particle and output time and includes the following information:

1. Current time
2. Particle name
3. Mass
4. x coordinate of position vector
5. y coordinate of position vector
6. z coordinate of position vector
7. x coordinate of velocity vector
8. y coordinate of velocity vector
9. z coordinate of velocity vector
10. Distance
11. Semi-major axis
12. $1-$eccentricity
13. Minimum central distance since last output
14. Maximum central distance since last output
15. Last apocentre distance
16. Last pericentre distance
17. Particle's share of total energy[1]
18. Particle energy
19. Multiplicity (1 for single star, 2 for binary)

3.3.3 `warn`: Integrator warnings

The `warn` file contains information on the integration, such as close encounters, removal of stars (escapers and collisions), (de-)activation of post-Newtonian terms, and particles switching between elliptical and hyperbolic orbits.

Furthermore, it may contain errors or warnings caused by external libraries (e.g. *GRAPE* libraries).

3.3.4 `debug`: Debug information

The `debug` file includes some information on the status of the program, such as program initialisation, or other noteworthy events.

[1] The particle's share of total energy consists of its kinetic energy, potential energy with respect to the central object and background potential, and *half* of the potential energy with respect to the other particles, such that the system energy is the sum of the energy shares of all particles (excluding the central object).

3.3.5 `other`

The `other` file is reserved for various other output.

3.4 Dump files

BHINT creates dump files with the extension *_n*`.dump` of the simulation status on a regular basis, where *n* is a sequential number. After a system crash or hardware failure, the integration can be restarted as

`bhint r` *dumpfile*

where *dumpfile* is the name of the last dump file. All output files of the restarted integration will have `.0001` before the extension, and this number is increased for any subsequent restart.

The particle information is stored in the dump file in binary code; however, the preamble is plain text, and some values may be changed with due care before restart. Most likely, the maximum integration time (line 3) needs to be increased.

A dump file can also be used to analyse the status of an integration: Calling

`bhint d` *dumpfile*

writes the particle information for all particles in the format of the `detail` output file (Section 3.3). This may also be used to create an input file; however, when using stellar evolution, all evolution data is lost, and all particles will be evolved with their current mass assumed to be the zero-age main sequence mass.

3.5 Other BHINT tools

The following table provides an overview of the capabilities of BHINT. The **command** denotes the first command line parameter when calling `bhint`.

Command	Description and usage
`m`	Create a **M**odel (see Section 3.1.1).
`i`	**I**ntegrate the specified model (see Section 3.2).
`r`	**R**estart from specified dump file (see Section 3.4).
`d`	Print particle **D**etails from specified dump file (see Section 3.4).
`c`	**C**omment model file. A model file is read from standard input, and written to standard output. For every particle other than the central, orbital parameters are calculated and added as a comment.
`n`	Convert specified model file to **N**-body units (Hénon 1971), where the sum of all masses is 1, $G = 1$, total linear momentum and centre of mass are 0, and the potential energy is $-1/4$. The syntax is `bhint n` *infile*, the output is written to standard output.
`o`	Calculate **O**rbital parameters for a particle. The syntax is `bhint o` m_0 x y z u v w, where m_0 is the central mass, and (x, y, z) and (u, v, w) are the position and velocity vectors, respectively (all in external units). Any mass profile will be considered (see Section 4.3).
`u`	Print internal **U**nits (conversion factors as specified in Section 2).

Example: The following command calculates the orbital parameters for a particle orbiting a $3 \times 10^6\,\mathrm{M}_\odot$ SMBH in the xy-plane at a distance of 0.1 pc with a velocity of $300\,\mathrm{km\,s^{-1}}$ ($= 3.068 \times 10^{-4}\,\mathrm{pc\,yr^{-1}}$):

```
linux:~/bhint$ bhint o 3e6 .1 .0 .0 .0 3.068e-4 .0
x=(1.000000e-01, 0.000000e+00, 0.000000e+00), |x|=1.000000e-01
v=(0.000000e+00, 3.068000e-04, 0.000000e+00), |v|=3.068000e-04
e=(-3.025066e-01, 0.000000e+00, 0.000000e+00), |e|=3.025066e-01
peri=(-5.355008e-02, 0.000000e+00, 0.000000e+00)
apo=(1.000000e-01, -0.000000e+00, -0.000000e+00)
T=1.150601e+03
a=0.0767750421
```

The output includes position, velocity, Runge-Lenz-vector and eccentricity, pericentre, apocentre, orbital period, and semi-major axis.

Example: Print BHINT internal units:

```
linux:~/bhint$ bhint u
1 mass unit   = 3.379381e+07 M_sun
1 length unit = 4.964492e-03 pc
1 time unit   = 8.971581e-01 yr
1 vel. unit   = 5.533575e-03 pc/yr        = 5.410807e+03 km/s
1 energy unit = 2.300379e+17 G*M_sun^2/pc
              = 1.034782e+03 M_sun*pc^2/yr^2
              = 1.967906e+58 erg
```

4 Configuring BHINT

Configuration of BHINT is done in `bhi_config.h`. In the following, we describe the configuration parameters used.

4.1 Precision parameters

The precision of BHINT integration is controlled by two parameters defining the size of an integration step:

`MIN_EVALS` defines the minimum number of steps per orbit (corresponds to $N_{\rm step}$ from Löckmann & Baumgardt 2008, Eq. 6). For elliptical or perturbed orbits, the actual number of steps may be higher.

`ETA` defines the precision parameter η in the Aarseth (1985) time step criterion for perturbations (see Löckmann & Baumgardt 2008, Eq. 7).

4.2 Analytic spherical potential

BHINT can include a spherical potential to mimic the influence of e.g. a stellar cusp surrounding a SMBH (see also discussion in Löckmann et al. 2009) by uncommenting the `EXT_POT` switch. The potential is included in the calculation of perturbations and requires the specification of the mass density profile $\rho(r)$, the enclosed mass $M(< r)$, and potential $\Phi(r)$. These have to be defined as `EP_RHO(r)`, `EP_M(r)`, and `EP_PHI(r)`, respectively, all in internal units.

For convenience, an implementation based on a two-part broken power-law mass density profile of the form

$$\begin{aligned} \rho(r) &= \rho_0 \, (r/r_0)^{\gamma_1}, \quad r_{\rm min} \leq r \leq r_0 \\ \rho(r) &= \rho_0 \, (r/r_0)^{\gamma_2}, \quad r_0 \leq r \leq r_{\rm max} \end{aligned} \tag{1}$$

is provided in external units (see comments in `bhi_config.h` for details).

While the analytic potential is used as a perturbation during the integration, it is not automatically considered when creating a model, or calculating orbital parameters for output. To correctly calculate orbits for model setup (e.g. when creating a circular disc in an analytic cusp) and output, the enclosed mass profile has to be defined (see below).

4.3 Mass profile

In addition to the mass of the central object, an extended mass (e.g. stellar cusp) resulting from the model stars or an analytic potential may affect a particle's orbit. While this is considered as a perturbation when integrating the orbit, it is not automatically considered for model creation or output. To account for an enclosed mass profile, uncomment the `USE_M_ENCL` switch and define the mass profile `M_ENCL(r)` in internal units.

Ideally, the defined mass should describe the sum of the masses of model (cusp stars) and analytic potential (see above). However, the enclosed mass is assumed to remain constant over the simulation time, which may not always be a valid approximation. In this case, it is suggested to use the *initial* mass distribution, especially when the examined problem requires a particular initial distribution of semi-major axes and eccentricities to be produced at model creation. Analyses of the integration results may be based on the particles' pericentre and apocentre distances r_p and r_a (which are determined at turning points of minimum and maximum central distance, respectively, and thus do not rely on specified mass profiles) instead of semi-major axes a and eccentricities e. The actual values of a and e are then given as $a = \left(r_a + r_p\right)/2$ and $e = \left(r_a - r_p\right) / \left(r_a + r_p\right)$.

For convenience, an implementation based on a two-part broken power-law mass density profile is provided in external units (see comments in `bhi_config.h` for details).

4.4 *GRAPE* usage

BHINT facilitates hardware support by *GRAPE* or graphics processing unit (GPU) boards. Any suitable library has to be included as described in Section 1.1. The hardware is released (e.g. for use of concurrent processes) after every `T_GRAPEFREE_INTERVAL` seconds of wall clock time.

When determining the time step for a particle, BHINT checks all particles in the neighbour sphere (see Löckmann & Baumgardt 2008, Section 2.2). The neighbour list can be provided by the *GRAPE* board, which may cause a problem if the list is too long. Experience shows that a maximum number of 100 neighbours is just manageable by the boards we used; using GPUs or other hardware may require different values, which can be configured by `C_MAX_NEIGHB`. To predict the number of particles in the neighbour sphere, BHINT evaluates the (maximum) local number density. To reduce the computational overhead, we use the number density of the form $\rho_N(r) = \rho_{N,0}\rho_{N,3}(r)^{-3}$, specified as `RHO_N_0` and `RHO_N_3(r)` in internal units. Again, the commented code gives an example of a two-part power-law profile.

4.5 Stellar evolution

To account for mass loss due to stellar evolution, add a suitable library as described in Section 1.2. BHINT evolves all particles (other than the central particle) up to a particle name of `N_MAX_SSE`. For example, a number of young stars moving through a background of stellar black holes (SBHs) may be provided before the SBHs in the model file, since the SBHs need not be evolved.

All particles above a mass of `MAX_SSE_MASS` are assumed to be black holes and are thus not evolved.

All stars are assumed to be zero-age main sequence stars with an initial stellar metallicity of `SSE_Z`.

4.6 Post-Newtonian treatment

The effects of general relativity can be simulated by BHINT using post-Newtonian (PN) approximations, which are activated by uncommenting the `PN` switch. For every step a particle is moved along its Keplerian orbit, the PN terms are evaluated N_{PN} times (see Löckmann & Baumgardt 2008, Section 2.4). This precision is specified in `PN_KEPLER_FACT` $= 1/N_{\mathrm{PN}}$.

Since this evaluation of PN terms is relatively expensive, it can be switched on and off for every particle according to specific requirements. PN evaluation is switched on for a particle once `SWITCHON_PN` is true, and switched off once `SWITCHOFF_PN` is true. To avoid large secular energy errors, PN evaluation should not constantly be switched on and off.

The switch criteria may be any logical expressions and may include particle properties such as name (stored as `p->name`), orbital eccentricity (`p->curr_e`) and semi-major axis (`p->curr_a`). Reasonable criteria may include the particle name (e.g. only use PN terms for certain particles), eccentricity (for highly eccentric orbits only), estimated merging time or combinations of these. The relativistic merging time of two particles of mass m_1 and m_2 as a function of initial semi-major axis a_0 and eccentricity e_0 can be estimated as (Peters 1964, Eq. 5.14)

$$T(a_0, e_0) = \frac{a_0^4}{\beta} \left(1 - e_0^2\right)^{7/2} g(e_0) \tag{2}$$

where

$$\beta = \frac{64}{5} \frac{G^3 m_1 m_2 (m_1 + m_2)}{c^5},$$

G is the gravitational constant, c is the speed of light, and

$$g(e_0) = \frac{12}{19} \frac{\sqrt{1 - e_0^2}}{e_0^{48/19} \left[1 + (121/304) e_0^2\right]^{3480/2299}} \int_0^{e_0} \frac{e^{29/19} \left[1 + (121/304) e^2\right]^{1181/2299}}{(1 - e^2)^{3/2}} \, \mathrm{d}e$$

takes values between 0.24 and 0.46. This merging time as a function of a_0 and e_0 is defined as `T_INSPIRAL` (with the approximation $g(e_0) \approx 0.3$).

Examples for switch criteria are provided in `bhi_config.h`.

4.7 Output

If `PRINT_DETAIL` is set, BHINT prints particle details for every nth regular output (see Section 3.3), where the interval n is defined in `PRINT_DETAIL_INTERVAL`. This detailed output is only provided for particles with names $\leq$ `N_MAX_DETAIL`, or all particles if `N_MAX_DETAIL` $= -2$.

Details for the particle with the name specified by `P_OUTPUT` are also provided in the regular output (see Section 3.3).

4.8 Other parameters

- Regular dump files are created after `DUMP_INTERVAL` seconds of wall clock time (see Section 3.4).
- The maximum number of particles in a model is `N_MAX`.
- The maximum central distance for a particle (in external units) is `MAX_X`. Particles which move beyond this distance will be removed from the simulation.
- The fraction of particles marked as binaries during model creation is specified as `BINARY_FRACTION` (see Section 3.1).

References

Aarseth, S. J. 1985, in J. U. Brackbill, & B. I. Cohen, eds., Multiple Time Scales (Academic Press, New York), 377

Hénon, M. H. 1971, Ap&SS, 14, 151

Hurley, J. R., Pols, O. R., & Tout, C. A. 2000, MNRAS, 315, 543

Kroupa, P. 2001, MNRAS, 322, 231

Löckmann, U., & Baumgardt, H. 2008, MNRAS, 384, 323

Löckmann, U., Baumgardt, H., & Kroupa, P. 2009, MNRAS, submitted

Makino, J., Fukushige, T., Koga, M., & Namura, K. 2003, PASJ, 55, 1163

Peters, P. C. 1964, Physical Review, 136, 1224

Salpeter, E. E. 1955, ApJ, 121, 161