

NBODY3 User Manual

Sverre Aarseth

Email: `sverre@ast.cam.ac.uk`

Institute of Astronomy, University of Cambridge

1 Introduction

This is a User Manual for *NBODY3*, a code that has been extensively tested since it was first developed around 1985. This code is mainly intended for laptops and workstations with UNIX operating system.

The code relies on many features of the classical *NBODY5* which dates back to the late 1970s. Briefly stated, single particles and centre-of-mass systems are integrated by a fourth-order method. Binaries and close two-body encounters are studied by the Stumpff version of Kustaanheimo–Stiefel [1964, KS] regularization [Mikkola & Aarseth 1998], while interactions of compact subsystems are calculated by the chain regularization method [Mikkola & Aarseth 1990, 1993, 1996]. Moreover, strong interactions in unperturbed triples and quadruples are treated by three-body [Aarseth & Zare 1974, AZ] and Heggie [1974] global regularization [Mikkola 1985]. Finally, hard triples and higher-order systems satisfying a stability criterion [Valtonen 2014] are reduced to two-body configurations (so-called ‘mergers’ as opposed to collisions). All the relevant equations of motion are derived and discussed extensively in a book [Aarseth 2003], together with algorithms which may be helpful when examining the *FORTRAN* procedures. Hence familiarity with parts of the book is beneficial for understanding the code. An original version of the code *NBODY3* can be downloaded from the web in the form of a compressed tar file.¹

2 Code structure

The whole code consists of some 18 000 lines including comments and layout space. It is written in *FORTRAN* and is f77 compliant but also compiles with g77, Intel or f95 (ignore warning messages of non-standard expressions). There are over 150 routines altogether, mostly with mnemonic names of maximum six characters. Likewise, almost all the *FORTRAN* statements are in upper case while the comments are in lower case. The coding generally conforms to a strict style and layout for clarity.

The main code relies heavily on a general `common` block called `common3.h` which contains a large number of arrays (mostly size `NMAX`) and many useful scalars. This enables a calculation to be split into several parts by saving all the `common` variables after a specified CPU time (or by a ‘touch STOP’ facility at arbitrary times, see below), followed by

¹`ftp.ast.cam.ac.uk/pub/sverre.`

a restart. Except for some special situations, this gives rise to reproducible results which are essential for experimental purposes as well as diagnostic investigation.

The code consists of three main parts: `input`, `output` and `integration`, with the latter split into several large routines employing different methods. Thus a smooth running relies on treating a range of special cases by the appropriate algorithm. However, the decision-making requires very little overheads. A number of optional procedures are included but care is needed to avoid mutual inconsistencies since only some parameter values and no options are checked (routine `verify.f`).

3 Getting started

Once the file `nbody3.tar.gz` is downloaded and uncompressed, the routines are extracted by `'tar xvf nbody3.tar'` and copied to three directories: `Ncode`, `Chain`, `Nchain`. The first contains the main code, while `Chain` holds the basic chain procedures and `Nchain` provides the corresponding *N*-body interface.

The size of most large `common` arrays is given by the parameter file `params.h` and defined in Table 1. Depending on available memory, it is recommended to limit the maximum array sizes somewhat but still leave room for bigger calculations; this will facilitate examining `common` blocks using the same `read` statements for different memberships. Note that the files `params.h` and `common3.h` are also used in the interface directory `Nchain`. Once these parameters have been specified (dir `Ncode`), the correct size of the `common` block is calculated automatically in routine `mydump.f`.

Before compiling, check the *FORTRAN* directives in the `Makefile`, i.e. whether `f77` or `gfortran` and use the highest optimization level. Compile by `'make nbody3'` which should produce the executable `nbody3`. One possible reason for failure could be the CPU timer function `etime` in routine `cputim.f` which is system dependent. Any other complaints and strange run-time behaviour should be reported after making every effort to ascertain the problem. For frequent usage it is recommended to create separate working directories. It is also a good idea to make a back-up of the original source files before making changes.

To start a test run, place the executable and a template input file `input` in one directory and type the command `'nbody3 < input > output &'` (or maybe `'time'` first). Although the results are machine- and compiler-dependent, it is expected that the run will finish normally but there is always a chance that some difficult configuration may occur.

The restart facility can be tested as follows. First specify option `# 1 = 1` as input. This will ensure a `common` dump on `fort.1` if the CPU time is exceeded or `TIME >= TCRIT`, with `TCRIT` given in the input file. The calculation can then be continued from the `common` save by `'nbody3 < rs > output2 &'`, using the restart input file `'KSTART TCOMP'`. Here `KSTART = 1` denotes a new run (followed by the required input) or `= 2` for standard restart, and `TCOMP` is the CPU time in minutes. Also note the possibility of reading some new parameters at restart if `KSTART > 2` (see routine `modify.f` and the restart file `rs`). Before restarting in the same directory, any important output files in capitals (*e.g.* `ESC`, `OUT9`) must be renamed and/or deleted.

The code includes an optional provision for automatic error checking. Thus if option # 2 = 2, any output interval with relative energy error $|\Delta E/E| > 5 Q_E$, with Q_E the tolerance, is restarted from the previous time with reduced value of the basic integration parameter η . If instead the energy error lies in the interval $[Q_E, 5 Q_E]$, the accuracy parameter is reduced by an appropriate amount. Likewise, an increase (up to the initial value) is carried out for relative errors below $Q_E/5$ (see routine `check.f`). Note the use of two options (with `common` saves on `fort.1` and `fort.2`) in order to employ the energy check independently of terminations at arbitrary times. Consequently, in the case of a halted calculation, the previous good `common` save must be copied from `fort.2` to `fort.1` before the restart (possibly with modified accuracy parameters).

4 Input parameters

A standard input file consists essentially of about ten lines defining the membership, accuracy and decision-making parameters, options, KS integration, IMF power-law data and virial theorem scaling. Some of these quantities are given in Table 2 and routine `define.f` contains a complete listing. Many are dimensionless while others have an astrophysical meaning. Likewise, the options KZ(J) can be selected by consulting Table 3 or routine `define.f`. Hence only a few values need to be changed in the input template once a calculation has finished. Uploaded binaries with option 22 can be regularized initially by specifying `NBIN0`. It would also be possible to include a few lines of code here for reading unscaled initial conditions.

For most purposes, and a wide range of particle numbers, the original choice of the accuracy parameter η and η_U may suffice. One good strategy for choosing the output intervals is to adopt $\Delta t_{\text{adj}} = 2.0$ for energy checks and escaper removal and take $\Delta t_{\text{out}} = 10.0$ for main output and data analysis. The choice of the relative energy tolerance Q_E is a matter of taste.

5 Initial conditions

Different choices of initial conditions are available. These include ia sphere of constant density or a Plummer model. Several types of IMF (Salpeter or Kroupa, Tout & Gilmore 1993) may be generated by option # 20. However, it is envisaged that the user will provide original initial conditions for new investigations ($m_i, \mathbf{r}_i, \mathbf{v}_i, i = 1, 2, \dots, N$).

Specially generated initial condition can be read from `fort.10` with option # 22 = 2 before scaling to standard internal units of total energy -0.25 (for bound clusters) and $\sum m_i = 1$. Primordial binaries and single stars can be uploaded using # 22 = 4 instead. Unless special precaution is taken, the same scaling is applied generally. Thus in scaled units and overall virial equilibrium the mean square velocity is $\langle v^2 \rangle = 1/2$ and the crossing time (denoted TCR) is $t_{\text{cr}} = 2\sqrt{2}$, independent of N .

With the initial length unit `RBAR` specified in pc and the mean mass `ZMBAR` (which is not preserved with option # 20 ≥ 0) in M_\odot , any internal quantity may be converted to astrophysical values. A convenient set of conversion factors is given by `RBAR`, `SMU`,

VSTAR, TSTAR for distances (pc), masses (M_\odot), velocities (km sec^{-1}) and times (in Myr). Binary periods in years or days can be obtained from the N -body units using YRS or DAYS in the standard Keplerian expression without 2π . Moreover, length scales may also readily be converted to solar radii (SU) or astronomical units (AU).

6 Decision-making

Individual time-steps are used in the integration of single particles and centre of mass binaries. The main integration cycle comprises a small number of calls to the next level of routines, with a few other important tasks performed either before or at the end. This structure facilitates the investigation of any strange behaviour such as infinite looping since the offending routine can be identified, whereupon the principle of bisection can often be applied to locate a problem.

The strategy for the first step relies on the simple device of determining the smallest value of $t + \Delta t_i$ for the members within a larger time interval instead of searching all N particles each time. The corresponding time-steps are obtained by a relative force criterion of the form

$$\Delta t_i = \left(\frac{\eta |\mathbf{F}_i|}{|\dot{\mathbf{F}}_i|} \right)^{1/2}, \quad (1)$$

where η is a small dimensionless constant ensuring convergence. A more sensitive expression is obtained by including the other force derivatives (see book Eq. 2.13).

As discussed in subsequent sections, the special case of KS and chain regularization merits special treatment. This may necessitate an occasional return to the main routine for the specified task, defined by the flow control indicator (cf. Table 5).

7 Data management

It is the purpose of a code to produce results. However, there is a wide choice of quantities that can be constructed from the basic particle data. Here a dual-purpose strategy has been adopted and if this is not convenient, special data analysis can readily be included either during integration or at output times (i.e. called from routine `adjust.f`).

The data structure itself forms the backbone of the code and will be summarized first (also see book p.117). Given a number of KS solutions N_p , the particle arrays for the components of each pair I_p are placed in locations $2I_p - 1$ and $2I_p$, with the corresponding centre of mass (c.m.) in $N + I_p$. Hence all KS pairs appear contiguously in the order in which they are initialized. This scheme facilitates a sequential treatment of all the particles, with force summations and neighbour lists referring to increasing locations in the range $[2N_p + 1, N + N_p]$. It entails relabelling particle references in perturber lists after each new or terminated KS pair but is a small cost for preserving simplicity. The structure of a given list for particle i is of the form LIST(\mathbf{k}, \mathbf{i}), where $k = 1$ is used for the membership number.

In order to introduce more complex systems, such as stable triples or temporary chain subsystems, the device of ‘ghost’ particles of zero mass is used (see below). This facilitates recovery of the original state without affecting the overall data structure. Pointers to the individual members are saved and any relevant information may be obtained as required, although the identification of multiple hierarchies is non-trivial (see Appendix C of book).

Two types of output are produced. Important results are presented in file names using capitals, while supplementary diagnostics appear in `fort.n` with $n > 2$. Also note the text file `fort.x` is available in the directory `Docs` of `NBODY6`. The main files provide the following information in mathematical notation:

- **ESC** t (Myr), m (M_\odot), $v^2 / \langle v^2 \rangle$, v (km s^{-1}), k^* , \mathcal{N}_i for each escaper
- **OUT9** E_b , e , E_{cm} , r , m_k , m_l , P (days), \mathcal{N}_k , \mathcal{N}_l , k_k^* , k_l^* for each KS binary
- **OUT3** m , x , y , z , \dot{x} , \dot{y} , \dot{z} , \mathcal{N}_i for $i = [1, N + N_p]$ (binary format)

Here \mathcal{N}_i represent original particle names, E_{cm} is the specific c.m. binding energy, P denotes the period (days), k^* is the stellar type, $a_1(1 - e_1)$ is the outer pericentre and R_{pcrit} the corresponding stability boundary. The maximum eccentricity in a Kozai cycle is given by e_{max} , while e_0 and e_1 refer to the respective eccentricities. Note that the file `HIARCH` contains information on the formation and termination of hierarchical systems whereas `HIDAT` provides a summary at each main output. All the above are optional (cf. the multi-valued # 8) and the file `OUT3` is the data bank which may be produced at main output with specified frequency. Note the use of *two* records for each output which enables arrays of variable length to be read.

A variety of results also appear as standard output. This takes the form of an error check at intervals Δt_{adj} , (with $\text{DE} = \Delta E/E$) when the density centre is updated and escapers removed. The accumulated energy error, `DETOT`, gives the systematic drift in total energy. Considerably more information is given at intervals Δt_{out} . In this connection, note the facility (# 32) to increase the output intervals if the energy binding the cluster changes by factors of 2.

We summarize some of the most important quantities given at main output. This information is organized in distinct (optional) groups. The first line gives the particle number, (`< NNB >`), KS solutions (`KS`), number of mergers (`NM` and `MM`, standard and higher order) and single stars (`NS`). Among useful quantities describing the cluster evolution are the half-mass radius (`< R >`), tidal radius (`RTIDE`), core radius (`RC`) and membership (`NC`), energy in binaries and mergers (`EB/E` and `EM/E`) and the time in Myr (`T6`). Escapers are removed outside `2*RTIDE` with option # 23.

If # 8 is active, there is a summary of original and exchanged binaries, the average and maximum eccentricity (`< E >` and `EMAX`), as well as distributions of stellar population types and binary binding energies. The energy budget is also summarized. For historical reasons, the energy binding the cluster is saved in `E(3)`, while `E(1)` and `E(2)` give the energy of primordial and dynamically formed binaries. For the definition of the total energy see Eq. (9.29) of the book. A large number of counters are displayed; see Table 4 for a list of the most important.

The question of adding extra variables to the `common` blocks often arises. Here we mention two useful strategies. The simplest case is to introduce a new labelled `common` in the header file `common6.h` so that these variables become available in most routines. At the same time, the size of the `common` save must be increased accordingly (cf. routine `mydump.f`) and the whole code recompiled. A slight drawback is that old `common` saves cannot be read in the usual way. The alternative is to create dummy variables in an existing labelled `common` which can be used for newly created variables without affecting the overall size. However, the latter method needs to anticipate future demands.

8 Two-body regularization

The KS method first appeared in a standard N -body code at the Cambridge IAU Colloquium # 10 in 1970. It was an instant success and has proved a mainstay for treating binaries and close two-body encounters ever since. Like any versatile algorithm, it has progressed through several distinct versions until ending up with the highly accurate Stumpff formulation [Mikkola & Aarseth 1998]. Although the underlying mathematics is very precise, implementations are something of a black art, with a variety of heuristic procedures. This is particularly the case when dealing with interactions in compact subsystems.

An arbitrary number of KS solutions can be treated at the same time. Decision-making is essentially controlled by two input parameters which are modified at each output if option # 16 is active. A search for close encounter candidates is carried out if the time-step becomes suitably small; i.e. $\Delta t_i < \Delta t_{cl}$, subject to the distance test $R < R_{cl}$ in order to ensure dominant two-body motion (see book for definitions). If necessary, the actual regularization is delayed until both the components are advanced to the same time. Note that the relative time-step criterion gives very similar values for different masses during close encounters. Implementations of the KS transformations and polynomial initialization have been described in considerable detail elsewhere [Aarseth 1985, 2001b, 2003]. It is therefore sufficient to discuss some aspects of the decision-making.²

As noted in a previous section, all relevant KS solutions are considered at the start of the integration cycle and not advanced beyond the end of the current step. For this purpose, the regularized time-step is converted to physical units (book Eq. (11.1)). In practice most primordial binaries may be unperturbed, with time intervals often exceeding the typical c.m. step, and an efficient sorting and insert procedure is employed (see book p.143). An algorithm for specifying intervals of unperturbed motion is also given in the book.

Termination is essentially controlled by the relative perturbation, except that soft binaries and hyperbolic flybys are also subject to a distance test in terms of the initial separation. The strategy for terminating a strongly perturbed hard binary depends on the suitability of a switch to chain regularization. The ideal case is that the latter treatment may be adopted but there are many situations when such configurations are not sufficiently compact (see below). Hence we may instead have repeated switching of particle pairs according to dominant two-body solutions, which is less satisfactory.

²The data structure is described in Section 7.

Following a KS termination, new polynomials and time-steps are assigned to the components. All relevant perturber lists must also be updated in order to be consistent with the new particle sequence. However, the latter task deals with integer arithmetic and is quite fast. Since the c.m. time-step is inevitably small for strong perturbations, the new steps may be comparable and hence do not result in significant shrinkage of time-steps. In any case, small values may increase rapidly if the situation permits.

9 Hierarchical systems

The presence of binaries often lead to the formation of long-lived hierarchies. The computational requirements for direct integration of the inner binary may be quite severe; yet the semi-major axis hardly changes even for modest distance ratios. Since this is the most important binary element, it would seem justified to adopt the c.m. approximation and introduce the KS solution for the outer component, thereby increasing the period and replacing one direct integration. This procedure neglects short-term fluctuations and assumes no secular change, in qualitative agreement with first-order perturbation theory.

Although simple stability criteria were already introduced in the mid 1980s, the chaos approach provided some theoretical justification and led to a semi-analytical criterion [Mardling & Aarseth 1999]. More recently, a general three-body stability criterion has been developed. It is valid for different masses and inclinations and was found to be more reliable than older criteria. However, a new stability criterion [Valtonen 2014] has now been extensively tested and appears more robust, particularly for large mass ratios. This algorithm has recently been adopted throughout.

The identification of suitable configurations is carried out for small c.m. time-steps and the first part of the procedure employs the same algorithm as for chain regularization (see below). Subsequently, the division of labour is essentially made by comparing the outer pericentre, $R_p = a_1(1 - e_1)$, with the inner semi-major axis, a_0 . Roughly speaking, the case $R_p > 3a_0$ justifies a search for stable systems. Further conditions, such as the requirement of the outer binary being hard and not too strongly perturbed must also be satisfied in addition to the stability test.

A new hierarchy is initialized in a similar way as for standard KS. However, when examining the data structure we distinguish between the outer component being a single particle or binary. In the case of a triple, the outer body is defined as a ghost particle after combining it with the inner c.m. into a new wider KS pair. However, with a second binary, both the KS solution and associated c.m. are made inactive by prescribing large values of T_0 and setting the relevant masses to zero. Here the latter plays the role of the ghost particle for triples and once the c.m. location $j > N$ has been identified from the ghost name, the pair index is simply given by $I_p = j - N$. For stability purposes, triples may be considered as degenerate quadruples and a small correction term is included for the smallest binary. Although rare, higher-order systems also occur and are treated in an analogous manner (their data structure is discussed in the book appendix).

Further stability checks are made at each apocentre passage because the outer orbit may have changed due to perturbations. Following termination of a triple, the inner

binary is initialized as a KS solution while the outer component already has the standard form (but is not in the correct location). In the case of a quadruple, the second KS pair and ghost c.m. are initialized *in situ* in the usual way after the neighbour list is formed. The force discontinuity arising from the changed configurations is handled in several ways with appropriate differential corrections to the energy budget. Note that the initialization and reconstruction of hierarchies employ the small labelled `common` block `BINARY` which contains original masses and particle names (as well as the basic KS variables for any second pair).

10 Chain regularization

Several methods are available for treating strong interactions involving more than two particles. The original AZ three-body regularization was implemented first, followed by Heggie’s global method for four particles [cf. Aarseth 1985]. For simplicity, these formulations do not include external perturbations. The versatile chain regularization [Mikkola & Aarseth 1990, 1993, 1996] which includes perturbations has proved more effective. Because of the increased complexity, the relevant routines are placed in the separate directories `Chain` and `Nchain`. Since only one configuration at a time is considered at present, the two unperturbed treatments are still maintained but rarely needed.

The data structure of the basic chain algorithm employs quantities expressed in the local c.m. frame. Transformations to global values must therefore be made before including the effect of perturbers, and likewise at termination. Since the device of ghost particles is adopted to preserve the sequential arrangement, the actual masses are saved together with the corresponding names. It is then a simple matter to recover the relevant global locations of ghosts for initialization. In the case of three initial members, the binary components are placed in the first two single particle locations, as for standard KS termination, while the third body is turned into a ghost particle *in situ*. The termination of two binaries, on the other hand, yields the initial members in the first four locations. However, this arrangement cannot be assumed to persist since a new KS may be formed during the intervening interval and in any case, a four-body system may be reduced by escape. Another point worth noting is that the particle assigned as the c.m. may in fact escape, in which case a new reference body is determined. We also mention that the current scheme now caters for up to six members, although more than four is quite rare.

The treatment of the chain c.m. requires special care. Thus the force and its first derivative are first obtained in the usual way, whereupon differential corrections are added. This entails subtracting the standard c.m. contributions from any perturbers and adding the respective individual mass-weighted terms. For consistency, similar corrections are carried out when dealing with the perturbers. Again the overheads of checking the neighbour lists for identification is modest compared with the actual function evaluations. Moreover, note that the accumulated duration of all the chain regularizations only represents a small fraction of the total time.

Some comments on the time management may be helpful. The choice of intervals for the internal integration is based on the principle of convergence for coordinate prediction.

Essentially the maximum interval for advancing the solutions is determined by examining $T_0 + \Delta t_j$ for the relevant perturbers as well as the c.m. itself. Typically, the Bulirsch–Stoer time-steps are somewhat smaller than this interval. An inversion from physical to regularized time usually ensures that the maximum is barely exceeded (see book for the algorithm). Additional features, such as slow-down and quantization of time are also discussed extensively in the book.

11 External fields

The code includes two types of external tidal field which will be described (for details see chapter 8 of the book). Linearized equations are appropriate for nearly circular orbits with small vertical displacements and are therefore suitable for simulating open clusters. Two optional variants are available: (i) `# 14 = 1` for the standard case based on Oort’s constants, and (ii) `# 14 = 2` which applies to a galactic point mass (also linearized).

The relevant tidal terms (converted from the length unit `RBAR` and total mass) are initialized in routine `xtrnl0.f` and the perturbations are added in `xtrnlf.f` and also in `xtrnlp.f` for KS. The additional terms in the equations of motion are simple. The corresponding contributions to the total energy are included (cf. `ETIDE` in `xtrnlv.f`), thereby facilitating conservation checks. However, the linearized form of the equations of motion are not appropriate well outside the tidal radius. Hence a more detailed exploration of ejected stars should be restricted to modest distances.

References

- Aarseth, S.J. [1985], ‘Direct methods for N-body simulations’, in *Multiple Time Scales*, ed. J.U. Brackbill & B.I. Cohen (Academic Press, Orlando)
- Aarseth, S.J. [2001b], ‘Regularization methods for the N -body problem’, in *The Restless Universe*, ed. B.A. Steves & A.J. Maciejewski (Inst. Phys. Publ.), 93–108
- Aarseth, S.J. [2003], *Gravitational N-Body Simulations* (Cambridge University Press)
- Aarseth, S.J. & Zare, K. [1974], ‘A regularization of the three-body problem’, *Celes. Mech.* **10**, 185–205
- Bulirsch, R. & Stoer, J. [1966], ‘Numerical treatment of ordinary differential equations by extrapolation methods’, *Num. Math.* **8**, 1–13
- Heggie, D.C. [1974], ‘A global regularisation of the gravitational N-body problem’, *Celes. Mech.* **10**, 217–41
- Kustaanheimo, P. & Stiefel, E. [1965], ‘Perturbation theory of Kepler motion based on spinor regularization’, *J. Reine Angew. Math.* **218**, 204–19
- Kroupa, P., Tout, C.A. & Gilmore, G. [1993], ‘The distribution of low-mass stars in the Galactic disc’, *Mon. Not. R. Astron. Soc.* **262**, 545–87
- Mardling, R.A. [2008], ‘A general three-body stability criterion’, in *The Cambridge N-Body Lectures*, ed. S.J. Aarseth, C.A. Tout, R.A. Mardling (Springer), 59–96
- Mardling, R.A. & Aarseth, S.J. [1999], ‘Dynamics and stability of three-body systems’, in *The Dynamics of Small Bodies in the Solar System*, ed. B.A. Steves & A. Roy (Kluwer), 385–92
- Mikkola, S. [1985], ‘A practical and regular formulation of the N -body equations’, *Mon. Not. R. Astron. Soc.* **215**, 171–7
- Mikkola, S. and Aarseth, S.J. [1990], ‘A chain regularization method for the few-body problem’, *Celes. Mech. Dyn. Ast.* **47**, 375–90
- Mikkola, S. & Aarseth, S.J. [1993], ‘An implementation of N -body chain regularization’, *Celes. Mech. Dyn. Ast.* **57**, 439–59
- Mikkola, S. & Aarseth, S.J. [1996], ‘A slow-down treatment for close binaries’, *Celes. Mech. Dyn. Ast.* **64**, 197–208
- Mikkola, S. & Aarseth, S.J. [1998], ‘An efficient integration method for binaries in N -body simulations’, *New Astron.* **3**, 309–20
- Nitadori, K. [2012], ‘An implementation of KS regularization with Hermite integrator and block timestep’, unpublished
- Nitadori, K. & Aarseth, S.J. [2012], ‘Accelerating NBODY6 with graphics processing units’, *Mon. Not. R. Astron. Soc.* **424**, 545–52
- Valtonen, M. [2014], ‘Hiararchical stability criterion’, AAS meeting

12 Appendix

In this Appendix we provide some tables of useful information. Table 1 defines the parameters used in the general `common` block, together with representative values for a test calculation with 1000 single particles and 1000 primordial binaries. Note the provision of extra KS solutions in case of additional close encounters in the early stages.

Table 1: *FORTTRAN parameters.*

N_{\max}	<i>NMAX</i>	Total particle number and c.m. bodies	4010
K_{\max}	<i>KMAX</i>	KS solutions	1010
M_{\max}	<i>MMAX</i>	Hierarchical binaries	10
M_{dis}	<i>MLD</i>	Recently disrupted KS components	22
N_{chain}	<i>NCMAX</i>	Chain membership	10

Table 2 contains an example of standard input parameters and typical values for an $N = 1000$ test run. Both the book and *FORTTRAN* notations are given for convenience. Slightly smaller values of the accuracy parameters are recommended.

Table 2: *Integration parameters.*

η	<i>ETA</i>	Time-step parameter for N -body integration	0.02
Δt_{adj}	<i>DTADJ</i>	Time interval for energy check	2.0
Δt_{out}	<i>DELTAT</i>	Time interval for main output	10.0
Q_E	<i>QE</i>	Tolerance for energy check	1.0×10^{-5}
R_V	<i>RBAR</i>	Virial cluster radius in pc	2.0
M_S	<i>ZMBAR</i>	Mean stellar mass in solar units (#20=0)	0.8
Q_{vir}	<i>Q</i>	Virial theorem ratio ($T/ U - 2W $)	0.5
Δt_{cl}	<i>DTMIN</i>	Time-step criterion for close encounters	4.0×10^{-5}
R_{cl}	<i>RMIN</i>	Distance criterion for KS regularization	0.001
η_U	<i>ETAU</i>	Regularized time-step parameter	0.2
h_{hard}	<i>ECLOSE</i>	Energy per unit mass for hard binary	1.0
γ_{\min}	<i>GMIN</i>	Limit for unperturbed KS motion	1.0×10^{-6}
γ_{\max}	<i>GMAX</i>	Termination criterion for soft binaries	0.001

The main options are listed below. For a complete list see routine `define.f`. To find where option # J is used, type `'grep KZ(J) *.f'` in the N -body directories (or just 'KZ' in the directory `Chain. ARchain`).

Table 3: *Optional features.*

1	Common save on unit 1 by <code>touch STOP</code> or <code>TIME > TCRIT</code>
2	Common save on unit 2 at output time or restart
3	Data bank on unit 3 with specified frequency
5	Standard initial conditions (=0: uniform; =1: Plummer)
6	Output of significant & KS binaries (=1, 2, 3 & 4)
7	Output of Lagrangian radii (several types)
8	Primordial binaries (extra input required)
9	Binary output
10	Regularization diagnostics (=2: NEW KS & END KS)
12	Cluster drag force
14	External tidal force; open or globular clusters
15	Multiple regularization or hierarchical systems
16	Updating of regularization parameters R_{cl} , Δt_{cl}
17	Modification of η by tolerance Q_{E}
18	Primordial triples (extra input required)
20	Different types of initial mass functions (=0: standard)
22	Initial conditions $m_i, \mathbf{r}_i, \dot{\mathbf{r}}_i$ on unit #10 (=2, 3, -1)
23	Removal of distant escapers (isolated or tidal)
26	Slow-down of KS and/or chain regularization (=1, 2, 3)
30	Chain regularization (> 1: special diagnostics)

Table 4 defines significant counters, together with an actual example from run with 1800 single particles and 200 primordial binaries.

Table 4: *Characteristic counters.*

Name	Definition	Counts
<i>NSTEPI</i>	Irregular time-steps	8.0×10^7
<i>NKSTRY</i>	Regularization attempts	2.4×10^6
<i>NKSREG</i>	KS regularizations	2.2×10^3
<i>NKSHYP</i>	Hyperbolic regularizations	800
<i>NKSMOD</i>	KS slow-down modifications	6.6×10^4
<i>NKSPER</i>	Unperturbed two-body orbits	4.6×10^{11}
<i>NMERGE</i>	Hierarchical mergers	359
<i>NEWHI</i>	Independent new hierarchies	25
<i>NCHAIN</i>	Chain regularizations	86
<i>NSTEPU</i>	Regularized time-steps	4.2×10^7
<i>NSTEP C</i>	Chain integration steps	1.1×10^5
<i>NCOLL</i>	Stellar collisions	4
<i>NSESC</i>	Single escapers	1833
<i>NBESC</i>	Binary escapers	176
<i>NMESC</i>	Hierarchical escapers	2

Table 5 lists the control indicators used for decision-making.

Table 5: *Indicator for flow control.*

0	Standard value
1	New KS regularization
2	KS termination
3	Output and energy check
4	Three-body regularization
5	Four-body regularization
6	New hierarchical system
7	Termination of hierarchy
8	Chain regularization
9	Physical collision
-1	Exceptional cases