

Computation and astrophysics of the N-body problem

Douglas Heggie
School of Mathematics
University of Edinburgh
d.c.heggie@ed.ac.uk

Course outline (tentative)

1. The N -body problem and its astrophysical settings. Initial conditions. Units. Introduction to a simple N -body code (NBODY1). Example - cold collapse. Virialisation and virial equilibrium.
2. Plummer's model. Movie of orbital motions. Crossing time scale. Core collapse - the movie. The relaxation time scale. The structure of the simplest direct-summation N -body code: constant, shared, time steps, Euler & Hermite integrators.
3. Quality control & error growth. Complexity. Acceleration of force computations with software & hardware. GPUs. Regularisation.
4. Refinements: external effects (tides), internal effects (stellar evolution, binary stellar evolution, collisions). The NBODY series of integrators. starlab. Example - cluster in a tidal field (NBODY6[tt]).

References

► Stellar Dynamics

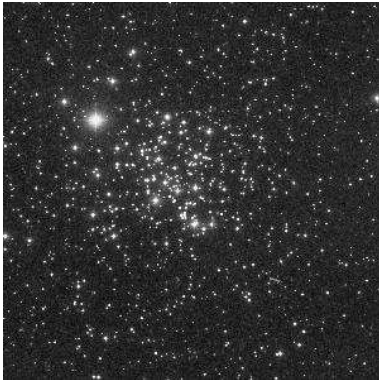
1. *Galactic Dynamics, 2e* James Binney and Scott Tremaine, Princeton University Press, 2008, 885 pp.
2. *Dynamical Evolution of Globular Clusters* Lyman J. Spitzer Jr, Princeton UP, 1988, 196 pp.
3. *The Gravitational Million Body Problem*, Douglas Heggie, Piet Hut; Cambridge UP, 2003

► N-body codes

1. *Gravitational N-body simulations*, Sverre J. Aarseth Cambridge: CUP, 2003, 413pp
2. *The Cambridge N-body Lectures*, S.J. Aarseth, C.A. Tout, R.A. Mardling, eds. Springer, LNP760, 2008, 402pp
3. *Numerical Methods in Astrophysics: An Introduction*, P. Bodenheimer et al, CRC Press, 2006, 344pp
4. <http://www.ast.cam.ac.uk/~sverre/web/pages/nbody.htm>
5. <https://github.com/nbodyx/Nbody6ppGPU>
6. <http://www.ids.ias.edu/~starlab/> (starlab)
7. <http://amusecode.org/> (AMUSE)

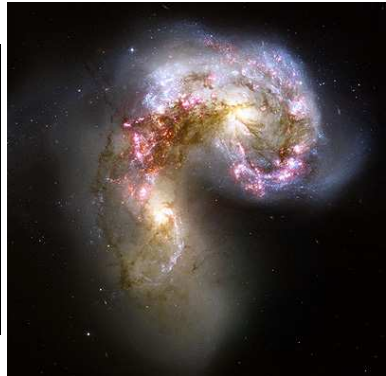
Applications of N -Body Schemes in Astrophysics

- ▶ the Galactic Centre (see <http://www.astro.ucla.edu/~ghezgroup/gc/pictures/orbitsMovie.shtml>)
- ▶ open clusters (M67 here)
- ▶ globular clusters (M4 here)



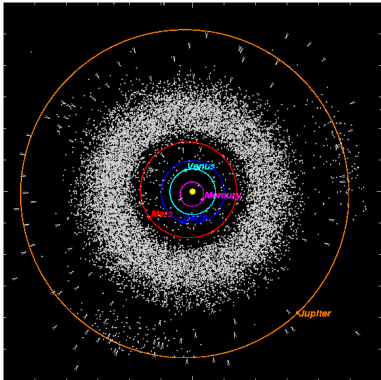
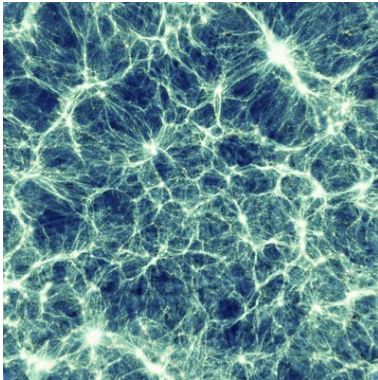
Applications of N -Body Schemes in Astrophysics (continued)

- ▶ young dense clusters - Magellanic Cloud clusters
- ▶ galaxy dynamics - the Antennae (requires specialised software because of large N)



Applications of N -Body Schemes in Astrophysics (continued)

- ▶ cosmic structure (requires specialised software)
- ▶ planetary systems (requires specialised software)



The N -body Problem

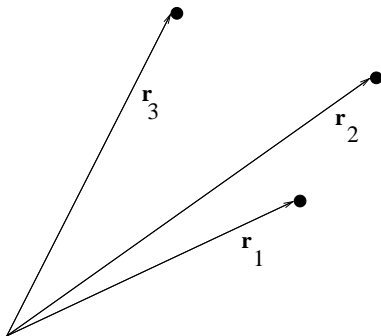
The N -body Problem

- ▶ N point masses (this approximation is good while separation of two stars is much greater than the sum of their radii)

The N -body Problem

- ▶ N point masses (this approximation is good while separation of two stars is much greater than the sum of their radii)
- ▶ classical gravitation and equations of motion (this approximation is good except close to horizon of black hole, or for close binaries emitting gravitational waves)

Equations of motion



Force on 1 due to 2 is in the direction of $\mathbf{r}_2 - \mathbf{r}_1$, i.e. the unit vector $\frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}$, and has magnitude $\frac{Gm_1 m_2}{|\mathbf{r}_2 - \mathbf{r}_1|^2}$.

Equations of motion (continued)

Therefore force on 1 due to 2 is $\frac{Gm_1m_2(\mathbf{r}_2 - \mathbf{r}_1)}{|\mathbf{r}_2 - \mathbf{r}_1|^3}$. Therefore total force on 1 is

$$m_1\ddot{\mathbf{r}}_1 = \sum_{j=1, j \neq 1}^{j=3} \frac{Gm_1m_j(\mathbf{r}_j - \mathbf{r}_1)}{|\mathbf{r}_j - \mathbf{r}_1|^3}$$

In the N -body problem the equation of motion for body i is

$$\ddot{\mathbf{r}}_i = -G \sum_{j=1, j \neq i}^N m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

Softening

The equation of motion is

$$\ddot{\mathbf{r}}_i = -G \sum_{j=1, \neq i}^N m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

There is a singularity if $|\mathbf{r}_i - \mathbf{r}_j| = 0$. To avoid this, the denominator is sometimes replaced by $(|\mathbf{r}_i - \mathbf{r}_j|^2 + \varepsilon^2)^{3/2}$, where ε is a small constant, the *softening parameter*. This approximation may be justifiable if close encounters between particles are unimportant – for example, in galaxy dynamics (with scaling by N). Not necessarily good for modelling star clusters.

Initial Conditions

These are $3N$ second-order ordinary differential equations, and hence require $6N$ initial conditions. Usually one uses the three cartesian components of position \mathbf{r}_i and the three components of velocity $\dot{\mathbf{r}}_i$ of the N particles.

Example: **cold collapse**

The initial velocities are zero, and the initial positions are chosen randomly, with (in this example) a uniform spatial distribution in a sphere of some radius a .

How to Simulate Cold Collapse

1. Go to the web page
<http://www.ast.cam.ac.uk/~sverre/web/pages/nbody.htm>
2. Download nbody1.tar.Z
3. Uncompress it ("gunzip nbody1.tar.Z")
4. Untar it ("tar xvf nbody1.tar")
5. Go to the source subdirectory ("cd Real8")
6. Edit the Makefile as follows
 - ▶ line 1: add " -freal-4-real-8" after "-g"
 - ▶ line 2: replace "f77" by your fortran compiler (e.g. "gfortran")
 - ▶ after line 2, add the line "FC = gfortran" (or your compiler)
7. Make the code ("make")
8. Go to the test subdirectory ("cd ../test")
9. Make a copy of the file intest and edit to the following:


```
1 1.0/25 1 200 1/0.01 0.1 10.0 2.0E-05 0.0/1 0 0 0 0 1 0 0 0 0
0 0 0 0 0/2.0 1.0 1.0/0.0 0.0 0.0 1.0 1.0
```

How to Simulate Cold Collapse (continued)

(It is best to give the file a new name, such as cc.in, and keep the old file intest. The changes are in boldface, and lines are here delimited by “/” to save space, but you should keep the format of intest.)

1. Run the code (“../Real8/nbody1 < cc.in”)
2. Watch the numbers fly past until the job completes or you have to kill it (“Ctrl-c”)¹
3. To save the output, rerun it with redirection of output (“../Real8/nbody1 < cc.in > cc.out”)

¹ If the run fails or fails to complete in a few minutes, kill it. Then edit cc.in to change the third number in line 2 of cc.in to some other positive integer, and try again.

Cold collapse: the output - p1

(Corresponds to slightly different input file.)

```

cc.out_p1      Fri Sep 14 14:14:10 2007      1

      N  NRAND   ETA  DELTAT  TCRIT  QE      EPS
25     200   0.010    0.1    10.0  2.0E-05  0.0E+00

      OPTIONS      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
                   1  0  1  0  0  1  0  0  0  0  0  0  0  0  0

      SCALING:  SX - 0.44 E - -5.72E-01  M(1) - 4.00E-02  M(N) - 4.00E-02  <M> - 4
,00E-02

      SCALING PARAMETERS:  R* - 1.00E+00  M* - 2.50E+01  V* - 3.28E-01  T* - 2.99E+00

```


Cold collapse: the output - p2

```

oo_out_p2      Fri Sep 14 14:14:29 2007      1

T - 0.0 Q - 0.00 STEPS - 0 DE - 0.000000 E - -0.250000 TC - 0.0
<R> - 2.00 RCM - 0.0000 VCM - 0.0000 AZ - 0.00000 T6 - 0 NRUN - 1

T - 0.3 Q - 0.01 STEPS - 35 DE - 0.000000 E - -0.250000 TC - 0.1
<R> - 1.99 RCM - 0.0000 VCM - 0.0000 AZ - 0.00000 T6 - 0 NRUN - 1

T - 0.6 Q - 0.02 STEPS - 87 DE - 0.000000 E - -0.250000 TC - 0.2
<R> - 1.95 RCM - 0.0000 VCM - 0.0000 AZ - 0.00000 T6 - 1 NRUN - 1

T - 0.9 Q - 0.05 STEPS - 154 DE - 0.000000 E - -0.250000 TC - 0.3
<R> - 1.89 RCM - 0.0000 VCM - 0.0000 AZ - 0.00000 T6 - 2 NRUN - 1

  BINARY  1 10 0.040 0.040 -0.1 0.3652 1.3 0.5156 1.54 0.970 0
  BINARY  6 15 0.040 0.040 -0.2 0.2525 2.2 0.3785 1.47 0.973 0
  BINARY  8 9 0.040 0.040 -0.2 0.2602 2.1 0.4063 1.94 0.989 0

T - 1.1 Q - 0.11 STEPS - 258 DE - 0.000000 E - -0.250000 TC - 0.4
<R> - 1.79 RCM - 0.0000 VCM - 0.0000 AZ - 0.00000 T6 - 3 NRUN - 1

  BINARY  6 15 0.040 0.040 -0.2 0.2509 2.2 0.2665 1.47 0.937 0
  BINARY  8 9 0.040 0.040 -0.2 0.2584 2.2 0.3032 1.89 0.983 0
  BINARY 14 22 0.040 0.040 -0.1 0.3191 1.6 0.4527 0.54 0.997 0

```

Cold collapse: the Movie

Initial conditions:

- ▶ All velocities are zero
- ▶ Particles are distributed uniformly in a sphere

Dynamic Equilibrium

Dynamic Equilibrium

- ▶ The system “quickly” reaches a “steady state”

Dynamic Equilibrium

- ▶ The system “quickly” reaches a “steady state”
- ▶ The steady state is in “dynamic equilibrium”, i.e. there is no overall expansion or contraction of the system, or other bulk motion, even though all particles are in motion.
- ▶ A dynamic equilibrium is also a state of “virial equilibrium”, which can be analysed in terms of the Virial Theorem.

The Virial Theorem

Define the total kinetic energy T , the total potential energy V , the total energy E and the total “moment of inertia” I by

$$T = \frac{1}{2} \sum_{i=1}^N m_i \mathbf{v}_i^2 \quad (\text{Kinetic Energy})$$

$$V = -\frac{G}{2} \sum_{i=1}^N \sum_{j=1, \neq i}^N \frac{m_j m_i}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (\text{Potential Energy})$$

$$E = T + V \quad (\text{Total Energy})$$

$$I = \sum_{i=1}^N m_i |\mathbf{r}_i|^2 \quad (\text{“Moment of Inertia”}).$$

Then from the equations of motion we deduce

$$\ddot{I} = 4T + 2V \quad (\text{Virial Theorem})$$

$$E = \text{constant} \quad (\text{Energy Conservation})$$

Proofs: Binney and Tremaine, Sec 7.2.1; this lecture, last pages

Virial Equilibrium

In dynamic equilibrium, I is approximately constant, and so $\ddot{I} \simeq 0$.
 From the Virial Theorem $\ddot{I} = 4T + 2V$ we deduce

$$2T + V \simeq 0.$$

Using total energy $E = T + V$ we deduce

$$T + E \simeq 0 \Rightarrow T \simeq -E$$

and similarly $V \simeq 2E$.

Mass, length and time scales

Mass, length and time scales

- ▶ Total mass

$$M = \sum_{i=1}^N m_i$$

Mass, length and time scales

- ▶ Total mass

$$M = \sum_{i=1}^N m_i$$

- ▶ Characterise system size by “virial radius” R defined by

$$V = -\frac{GM^2}{2R}, \text{ where } M \text{ is total mass}$$

Mass, length and time scales

- ▶ Total mass

$$M = \sum_{i=1}^N m_i$$

- ▶ Characterise system size by “virial radius” R defined by

$$V = -\frac{GM^2}{2R}, \text{ where } M \text{ is total mass}$$

- ▶ Characterise speeds by (mass weighted) mean square speed

$$v^2 = \frac{2T}{M}$$

Mass, length and time scales

- ▶ Total mass

$$M = \sum_{i=1}^N m_i$$

- ▶ Characterise system size by “virial radius” R defined by

$$V = -\frac{GM^2}{2R}, \text{ where } M \text{ is total mass}$$

- ▶ Characterise speeds by (mass weighted) mean square speed

$$v^2 = \frac{2T}{M}$$

- ▶ Define time scale

$$t_{cr} = \frac{2R}{v} \text{ (“Crossing time”)}$$

Other useful expressions and definitions

In virial equilibrium $V = -2T$, and so $v^2 = \frac{GM}{2R}$.

The *virial ratio* is defined to be $Q = \frac{T}{|V|}$, and is 0.5 in virial equilibrium.

Another measure of the size of the system is the *half-mass radius*, the radius of a sphere containing the innermost half of the mass, measured with respect to the “centre” of the system.

More generally, a “lagrangian radius” is the radius of the sphere containing a given fixed fraction of the mass.

Observational astronomers may prefer the *half-light radius*, the radius of a disk containing the innermost half of the light, since it can be “easily” measured.

Hénon Units (*aka N-body units*)

This is a conventional system of units in which

$$G = 1$$

$$M = 1$$

$$R = 1$$

These are often used in N -body simulations.

Example

We have $v^2 = \frac{GM}{2R}$.

Suppose a star cluster has $M = 10^5 M_\odot$, $R = 5\text{pc}$. To convert a velocity from the N -body code to km/s, multiply by $\sqrt{\frac{GM}{R}}$, where G is expressed in the same units (i.e. km/s, M_\odot , pc), i.e. $G \simeq 0.0043$.

Hénon Units (continued)

We have

$$G = 1$$

$$M = 1$$

$$R = 1$$

In these units:

- ▶ The characteristic speed $v^2 = \frac{GM}{2R} = \frac{1}{2}$
- ▶ The crossing time $t_{cr} = \frac{2R}{v} = 2\sqrt{2}$
- ▶ The total energy $E = -\frac{1}{2}Mv^2 = -\frac{1}{4}$

Significance of the crossing time

Significance of the crossing time

- ▶ Time scale of cold collapse

Significance of the crossing time

- ▶ Time scale of cold collapse
- ▶ Time scale of approach to virial equilibrium

Significance of the crossing time

- ▶ Time scale of cold collapse
- ▶ Time scale of approach to virial equilibrium
- ▶ Time scale of orbital motions in virial equilibrium (Lecture 2)

The code NBODY1: input

The meaning of each input parameter is defined in the file define.f in the source subdirectory. For illustration we use the file for the cold collapse simulation:

```
1 0.5
25 1 200 1
0.01 0.1 10.0 2.0E-05 0.0
1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
2.0 1.0 1.0
0.0 0.0 0.0 1.0 1.0
```

The code NBODY1: input

The meaning of each input parameter is defined in the file define.f in the source subdirectory. For illustration we use the file for the cold collapse simulation:

```
1 0.5
25 1 200 1
0.01 0.1 10.0 2.0E-05 0.0
1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
2.0 1.0 1.0
0.0 0.0 0.0 1.0 1.0
```

Line 1: 1 0.5

1. KSTART Control index (1: new run; >1: restart; 3: new params).

Comment: runs can be restarted following a crash or other accident

2. TCOMP Maximum computing time in minutes

The code NBODY1: input line 2

Line 2: 25 1 200 1

1. N Total particle number.
2. NFIX Output frequency of data save or binaries (option 3 & 6 – see below).

Comment: There is a basic output interval. You need not output the data or information on binaries at all such output times.

3. NRAND Random number sequence skip.
Comment: initialises the random number generator
4. NRUN Run identification index.

The code NBODY1: input line 3

Line 3: 0.01 0.1 10.0 2.0E-05 0.0

1. ETA Time-step parameter for total force polynomial.
Comment: this controls the accuracy of the numerical solution of the equations of motion (see later)
2. DELTAT Output time interval in units of the crossing time.
Comment: In N -body units the crossing time is $2\sqrt{2} \simeq 3$.
3. TCRIT Termination time in units of the crossing time.
4. QE Energy tolerance (stop if $DE/E > 5*QE$ & $KZ(2) \leq 1$).
Comment: the program stops if the relative change in energy exceeds the stated value and the appropriate option is chosen (see below)
5. EPS Softening parameter (square saved in EPS2).

The code NBODY1: input line 4

Line 4: 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0

Note: This is the line of *options*. In general, a zero value indicates that the option is not selected

1. 1 COMMON save on unit 1 if TCOMP > CPU or if TIME > TCRIT.

Comment: COMMON is a block of variables which are enough to restart the run, if desired.

2. 2 COMMON save on unit 2 at output (=1); restart if DE/E > 5*QE (=2).

Comment: if this value is 1, the COMMON variables are saved every output time; if 2, and the accuracy of the run has deteriorated, the code attempts to repeat the most recent part of the run with higher accuracy (and carry on)

The code NBODY1: input line 4 (continued)

Line 4: 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0

3 Basic data written to unit 3 at output time (frequency NFIX).
Comment: if this is positive, complete information on the particles is saved in a binary file OUT3 at each output time.

4 Initial conditions on unit 4 (=1: output; =2: input).
Comment: if this value is 1 the code dumps the initial conditions on a file called fort.4; if 2, the initial conditions are read in from this file (usually called fort.4)

5 Initial conditions (=0: uniform & isotropic; =1: Plummer).
Comment: we used 0. The Plummer model is explained in Lecture 2

6 Output of significant binaries.

7 Output of movie frames on unit 7.

Comment: you can experiment with this!

8 Generation of two subsystems (merger experiment).

The code NBODY1: input line 4 (continued)

Line 4: 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0

9 Individual bodies printed at output time ($\text{MIN}(5^{**}\text{KZ9}, \text{N})$).

Comment: the value is referred to as KZ9, and can be used to control how many particles are listed

10 No scaling of initial conditions.

Comment: i.e. use units of the input file fort.4, and do not scale to N -body units

11 Modification of ETA by tolerance QE.

Comment: ETA controls the accuracy of the integration (see above), and this option lets the code attempt to adjust this by monitoring the relative change in energy.

12 Initial parameters for binary orbit.

Comment: if non-zero, the code reads the semi-major axis and eccentricity of a binary formed by the first two particles

13 Escaper removal ($R > 2 * \text{RTIDE}$; $\text{RTIDE} = 10 * \text{RSCALE}$).

Comment: here RSCALE is the virial radius.

The code NBODY1: input line 4 (continued)

Line 4: 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0

14 Adjustment of coordinates & velocities to c.m. condition.

Comment: uses a barycentric coordinate system

15 Ignored

The code NBODY1: input line 5

Line 5: 2.0 1.0 1.0

1. ALPHAS Power-law index for initial mass function (routine DATA).

Comment: the initial mass function of the stars is a power law, i.e. $f(m) \propto m^{-\alpha}$ in the range $BODYN < m < BODY1$ (see below)

2. BODY1 Maximum particle mass before scaling.
3. BODYN Minimum particle mass before scaling.

Comment: choosing $BODY1 = BODYN$ gives equal masses. The code interprets these as being in solar masses, and then scales to N -body units internally.

The code NBODY1: input line 6

Line 6: 0.0 0.0 0.0 1.0 1.0

1. Q Virial ratio (routine SCALE; $Q = 0.5$ for equilibrium).
2. VXROT XY-velocity scaling factor (> 0 for solid-body rotation).
Comment: adds rotation about the z-axis
3. VZROT Z-velocity scaling factor (not used if $VXROT = 0$).
Comment: if less than 1 the z-components of velocity are reduced and the systems tends to flatten
4. RBAR Virial radius in pc (for scaling to physical units).
Comment: specifies the unit of length in parsecs
5. ZMBAR Mean mass in solar units.

Cold collapse: the output - p1

The first four lines repeat most of the input parameters

```

cc.out_p1      Fri Sep 14 14:14:10 2007      1

      N  NRAND   ETA  DELTAT  TCRIT  QE      EPS
25     200   0.010    0.1    10.0   2.0E-05   0.0E+00

      OPTIONS      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
                   1  0  1  0  0  1  0  0  0  0  0  0  0  0  0

SCALING:  SX  =  0.44  E  = -5.72E-01  M(1) = 4.00E-02  M(N) = 4.00E-02  <M> = 4
,00E-02

SCALING PARAMETERS:  R* = 1.00E+00  M* = 2.50E+01  V* = 3.28E-01  T* = 2.99E+00

```

Cold collapse: the output (continued)

line 5: SCALING: $SX = 0.44$ $E = -5.72E-01$ $M(1) = 4.00E-02$ $M(N) = 4.00E-02$ $\langle M \rangle = 4.00E-02$

Comments:

1. E is the energy before scaling to N -body units
2. $M(1), M(N)$ are the same as $BODY1, BODYN$ (see above), but now in N -body units.
3. $\langle M \rangle$ is the mean mass (N -body units)

line 6: SCALING PARAMETERS: $R^* = 1.00E+00$ $M^* = 2.50E+01$
 $V^* = 3.28E-01$ $T^* = 2.99E+00$

Comment: scaling factors for length (to parsecs), mass (to solar masses), velocities (to km/s), time (Myr, approximately)

Cold collapse: the output (continued)

The following gives an example of the output produced at each output time:

T = 0.9 Q = 0.05 STEPS = 154 DE = 0.000000 E = -0.250000 TC = 0.3

<R> = 1.89 RCM = 0.0000 VCM = 0.0000 AZ = 0.00000 T6 = 2
NRUN = 1

BINARY 1 10 0.040 0.040 -0.1 0.3652 1.3 0.5156 1.54 0.970 0

BINARY 6 15 0.040 0.040 -0.2 0.2525 2.2 0.3785 1.47 0.973 0

BINARY 8 9 0.040 0.040 -0.2 0.2602 2.1 0.4063 1.94 0.989 0

Line 1

1. Time in N -body units
2. Virial ratio
3. Number of integration steps taken (see below)
4. Change in energy (N -body units)
5. Total energy
6. Time in units of the crossing time

Cold collapse: the output (continued)

T = 0.9 Q = 0.05 STEPS = 154 DE = 0.000000 E = -0.250000 TC = 0.3

**<R> = 1.89 RCM = 0.0000 VCM = 0.0000 AZ = 0.00000 T6 = 2
NRUN = 1**

BINARY 1 10 0.040 0.040 -0.1 0.3652 1.3 0.5156 1.54 0.970 0

BINARY 6 15 0.040 0.040 -0.2 0.2525 2.2 0.3785 1.47 0.973 0

BINARY 8 9 0.040 0.040 -0.2 0.2602 2.1 0.4063 1.94 0.989 0

Line 2

1. Virial radius
2. Distance of centre of mass from origin
3. Velocity of centre of mass
4. Angular momentum about the z-axis
5. Time in Myr (approximately)
6. Run number (as in input)

Cold collapse: the output (continued)

T = 0.9 Q = 0.05 STEPS = 154 DE = 0.000000 E = -0.250000 TC = 0.3

<R> = 1.89 RCM = 0.0000 VCM = 0.0000 AZ = 0.00000 T6 = 2
NRUN = 1

BINARY 1 10 0.040 0.040 -0.1 0.365 1.3 0.5156 1.54 0.970 0

Line 3 (one line for each binary)

1. (2 numbers) Names of the two components
2. (2 numbers) Masses of the components (*N*-body units)
3. Internal energy of the binary (per unit reduced mass); this excludes the energy of the centre of mass of the binary
4. Semi-major axis of the binary
5. Angular velocity (mean motion) of the binary
6. Separation of the components
7. Distance of the binary from the “centre” of the system
8. Eccentricity of the binary
9. Number of binary periods since $t = 0$

Extracting useful information with awk

The typical output:

T = 0.9 Q = 0.05 STEPS = 154 DE = 0.000000 E = -0.250000 TC = 0.3

<R> = 1.89 RCM = 0.0000 VCM = 0.0000 AZ = 0.00000 T6 = 2
NRUN = 1

BINARY 1 10 0.040 0.040 -0.1 0.3652 1.3 0.5156 1.54 0.970 0

(Note that there are only three lines in the output file, but these may be wrapped in the pdf you are reading.)

We can extract data with the linux command awk.

First, create a text file called (say) q.awk, containing the following line:

```
{if ($1=="T") print $3,$6}
```

This means that we pick out only those lines of output beginning with the string "T", and from such lines print out the third and sixth fields (fields being separated by spaces), i.e. the values of T , Q .

Extracting useful information with awk (continue)

First rerun the code but collect the output in a text file cc.out. (Again the run may have to be killed after a short time). The command is

```
../Real8/nbody1 < cc.in > cc.out
```

Then awk is run with

```
awk -f q.awk cc.out > cc.q
```

which means that awk is to pick up its instructions from the file q.awk, act on the file cc.out, and direct its output to a new file cc.q. Here is an example of cc.q (a table of T , Q values)

```
0.0 0.00  
0.3 0.01  
0.6 0.02  
0.9 0.05  
1.1 0.11  
1.4 0.23
```

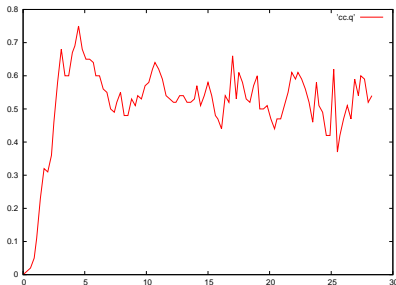
Plotting useful information with gnuplot

Start gnuplot (“gnuplot”), and then plot the file just produced:

```
gnuplot> plot 'cc.q' w l
```

Note: “w l” means “with lines”; otherwise you just get points.

This produces a window with the following graph:



Plotting useful information with gnuplot (continued)

The plot can be improved and made permanent with a sequence of commands like this:

```
gnuplot> set xlab 'Time'
gnuplot> set ylab 'Virial ratio'
gnuplot> set ter post
gnuplot> set out 'q.ps'
gnuplot> replot
```

Then you can print or view the postscript file q.ps

Proof of the Virial Theorem

“Moment of inertia”

$$I = \sum_{i=1}^N m_i (\mathbf{r}_i \cdot \mathbf{r}_i)^2.$$

Hence

$$\begin{aligned}\dot{I} &= \sum_{i=1}^N 2m_i \mathbf{r}_i \cdot \mathbf{v}_i \\ \ddot{I} &= \sum_{i=1}^N 2m_i (\mathbf{v}_i \cdot \mathbf{v}_i + \mathbf{r}_i \cdot \dot{\mathbf{v}}_i).\end{aligned}$$

The first term gives 4 times the kinetic energy, i.e. $4T$. From the equations of motion (above), the second term gives

$$-G \sum_{i=1}^N 2m_i \mathbf{r}_i \cdot \sum_{j=1, \neq i}^N m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} = -G \sum_{j=1}^N 2m_j \mathbf{r}_j \cdot \sum_{i=1, \neq j}^N m_i \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^3} \quad (1)$$

(swapping i and j).

Proof of the Virial Theorem (Contd)

Therefore the second term gives half their sum, i.e.

$$-G \sum_{i=1}^N \sum_{j=1, \neq i}^N m_i m_j (\mathbf{r}_i - \mathbf{r}_j) \cdot \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} = -G \sum \sum \frac{m_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|},$$

the sum being over all distinct i, j between 1 and N . This is *twice* the potential energy, because each pair is counted twice.

Hence $\ddot{I} = 4T + 2V$.

Exercises from Lecture 1

1. Carry out the procedure of downloading NBODY1, running the cold collapse simulation, and plotting the time-dependence of the virial ratio.

Computation and astrophysics of the N-body problem
Lecture 2

Outline of Lecture 1

1. Applications to star clusters
2. Equations of the N -body problem
3. Simulating cold collapse
4. The Virial Theorem; the crossing time; units and scaling
5. NBODY1: input, output, plotting

Plummer's model

- ▶ In the context of star clusters, a “model” is a prescription for the distribution of the stars in phase space, i.e. the joint distribution of position and velocity

Plummer's model

- ▶ In the context of star clusters, a “model” is a prescription for the distribution of the stars in phase space, i.e. the joint distribution of position and velocity
- ▶ Plummer's model is a particular model of a system which is in virial (and dynamic) equilibrium
- ▶ Convenient analytical distributions

- ▶ Density $\rho(r) = \frac{3M}{4\pi a^3 (1 + r^2/a^2)^{5/2}}$

- ▶ Potential $\phi(r) = -\frac{GM}{(1 + r^2/a^2)^{1/2}}$

- ▶ Velocity distribution at radius r : $f(\mathbf{v}) \propto \left(-\phi(r) - \frac{1}{2}v^2\right)^{7/2}$,
 $v^2 < -2\phi$

In N -body units, $G = M = 1$ and $a = \frac{3\pi}{16}$.

Plummer's model in pictures

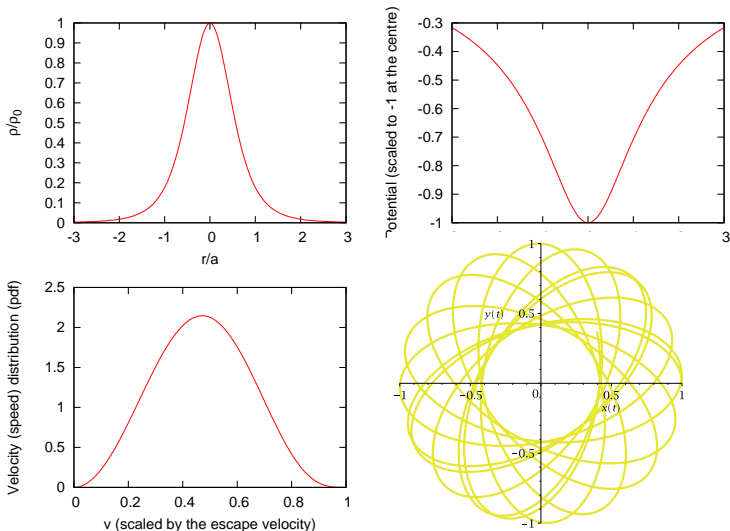


Figure: Density, potential, velocity distribution, an orbit

Models of stellar systems

- ▶ In the language of statistical physics, a system is specified by the distribution of particles in phase space, which has coordinates (\mathbf{r}, \mathbf{v}) .
- ▶ The distribution function is $f(\mathbf{r}, \mathbf{v}, t)$, i.e. the number of stars in a hypercube with sides $d\mathbf{r}, d\mathbf{v}$ is $f(\mathbf{r}, \mathbf{v})d\mathbf{r}d\mathbf{v}$.
- ▶ Because particles are conserved, they obey a conservation equation

$$\begin{aligned}
 \frac{\partial f}{\partial t} + \nabla_{\mathbf{r}} \cdot (f \dot{\mathbf{r}}) + \nabla_{\mathbf{v}} \cdot (f \dot{\mathbf{v}}) &= 0 \\
 \Rightarrow \frac{\partial f}{\partial t} + \nabla_{\mathbf{r}} \cdot (f \mathbf{v}) - \nabla_{\mathbf{v}} \cdot (f \nabla_{\mathbf{r}} \phi(\mathbf{r}, t)) &= 0 \\
 \Rightarrow \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} f - \nabla_{\mathbf{r}} \phi(\mathbf{r}, t) \cdot \nabla_{\mathbf{v}} f &= 0
 \end{aligned}$$

This is the *collisionless Boltzmann equation*

Solving the CBE $\partial f / \partial t + \mathbf{v} \cdot \nabla_{\mathbf{r}} f - \nabla_{\mathbf{r}} \phi(\mathbf{r}, t) \cdot \nabla_{\mathbf{v}} f = 0$

- ▶ Example: suppose that the potential $\phi(\mathbf{r})$ is time-independent, and that f is some function of the particle energy $E = v^2/2 + \phi(\mathbf{r})$, i.e. $f(\mathbf{r}, \mathbf{v}) = F(E)$.
- ▶ Then

$$\partial f / \partial t = 0$$

$$\mathbf{v} \cdot \nabla_{\mathbf{r}} f = F'(E) \mathbf{v} \cdot \nabla_{\mathbf{r}} \phi(\mathbf{r})$$

$$\nabla_{\mathbf{r}} \phi(\mathbf{r}, t) \cdot \nabla_{\mathbf{v}} f = F'(E) \nabla_{\mathbf{r}} \phi(\mathbf{r}) \cdot \mathbf{v}$$

- ▶ Hence this f satisfies the CBE with $\partial f / \partial t = 0$, i.e. it is an equilibrium (stationary) solution.
- ▶ In general, f is a solution if it is expressed as a function of constants of the motion (e.g. E if the potential is independent of t , and angular momentum if the potential is spherically symmetric.) This roughly is *Jeans' Theorem*.

Self-consistent models

- ▶ Plummer's model is given by $F(E) = C|E|^{7/2}$ when $E < 0$, 0 otherwise, where C is constant.
- ▶ In terms of $f(\mathbf{r}, \mathbf{v}) = F(E)$, the space density is given by $\rho(\mathbf{r}) = m \int F(E) d^3\mathbf{v}$, where m is the mass of one star.
- ▶ For Plummer's model

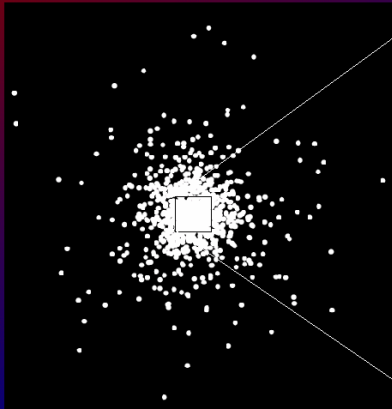
$$\rho(\mathbf{r}) = 4\pi m C \int_0^{\sqrt{-2\phi}} (-\phi - v^2/2)^{7/2} v^2 dv, = C'(-\phi)^5, \quad (2)$$

where C' is another constant.

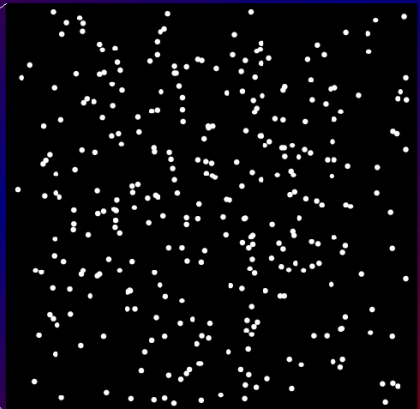
- ▶ ρ and ϕ are also related by Poisson's equation $\nabla^2\phi(\mathbf{r}) = 4\pi G\rho(\mathbf{r})$
- ▶ If we assume that ϕ is spherically symmetric (depending only on r), for Plummer's model this is $\phi''(r) + 2\phi'(r)/r = 4\pi G C'(-\phi(r))^5$.
- ▶ Solution: $\phi(r) = C'' \left(1 + \frac{r^2}{a^2}\right)^{-1/2}$ for suitable constants C'' , a .

YITP

Core collapse: Two simulations



Entire system
13 seconds to $t = 3.3$



Central area
13 seconds to $t = 330$

The initial conditions (Plummer's model)

Lessons from the simulations

Evolution on two time scales:

Lessons from the simulations

Evolution on two time scales:

- ▶ orbital motions (crossing time scale; see Lecture 1)
- ▶ much slower evolution of the statistical distribution: evolution of the central density
 - ▶ This is the statistical result of numerous “close” encounters between pairs of particles
 - ▶ This process is called “two-body relaxation”, or “collisional relaxation”
 - ▶ It acts on a time scale called the “relaxation time”

Other observations:

- ▶ Plummer’s model is in dynamical equilibrium
- ▶ The slow increase in the central density is called “core collapse”
- ▶ *Note: the core is the name given to the region in the centre where the density is nearly uniform*

(Two-body) relaxation time

- ▶ Local definition $t_r = \frac{0.065v^3}{\rho m G^2 \ln \gamma N}$ where
 - ▶ v is the velocity dispersion (root mean square velocity)
 - ▶ ρ is the space (mass-)density
 - ▶ m is the particle mass
 - ▶ γ is a constant (about 0.11 for equal masses)
 - ▶ N is number of particles

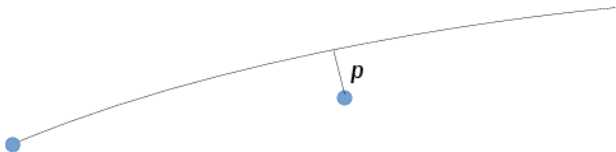
- ▶ Global definition: half-mass relaxation time

$$t_{rh} = 0.138 \frac{N^{1/2} r_h^{3/2}}{m^{1/2} G^{1/2} \ln(\gamma N)}, \text{ where}$$

- ▶ r_h is the *half-mass radius* (containing the innermost half of the system; see Lecture 1); comparable with the virial radius
- ▶ Recall (lecture 1) $\langle v^2 \rangle = \frac{GNm}{2R}$
- ▶ Mean density inside r_h is $0.5Nm/(4\pi r_h^3/3)$

“Derivation” of the relaxation time

- ▶ Consider a distant encounter between two stars at distance p (impact parameter, which is approximately the distance of closest approach in a distant encounter). Let v be relative speed.



- ▶ Acceleration at closest approach is Gm/p^2 , duration $\sim p/v$
- ▶ Hence change of velocity has magnitude $\Delta v \sim Gm/(pv)$.
- ▶ After many encounters, total *transverse* change of velocity is nearly zero. Therefore we measure the strength of two-body encounters by summing $(\Delta v)^2$.

“Derivation” of the relaxation time (contd)

- ▶ $\Delta v \sim Gm/(pv)$.
- ▶ Let n be the number-density of stars.
- ▶ In time t the number of encounters within distance p is of order $n.\pi p^2.vt$



- ▶ The number between p and $p + dp$ is of order $n.\pi 2p.dp.vt$
- ▶ Hence the sum of $(\Delta v)^2$ over this time is

$$\begin{aligned} \sum (\Delta v)^2 &\sim \int nvt \left(\frac{Gm}{pv} \right)^2 p dp \\ &= \frac{ntG^2 m^2}{v} \int \frac{dp}{p} \end{aligned}$$

“Derivation” of the relaxation time (contd)



$$\begin{aligned}\sum (\Delta v)^2 &\sim \frac{ntG^2m^2}{v} \int \frac{dp}{p} \\ &= \frac{ntG^2m^2}{v} [\ln p]_{p_{min}}^{p_{max}}\end{aligned}$$

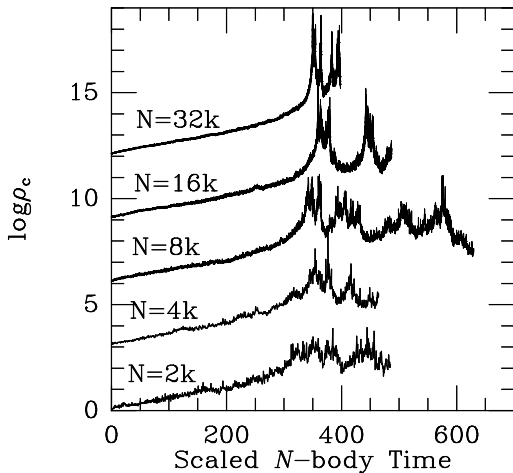
- ▶ For p_{max} we take the radius of the cluster, say the virial radius R
- ▶ For p_{min} we take the value at which the assumption of small deflection breaks down, i.e. $\Delta v \sim v$, or $p \sim Gm/v^2$.
- ▶ Then $[\ln p]_{p_{min}}^{p_{max}} \sim \ln(Rv^2/(Gm))$
- ▶ By the virial theorem, $2T + V \simeq 0$. Therefore $Nm\langle v^2 \rangle - GN^2m^2/(2R) \simeq 0$, and so $R\langle v^2 \rangle/(Gm) \simeq N/2$
- ▶ The total effect of encounters becomes important when $\sum (\Delta v)^2 \simeq v^2$, i.e. at a time $t \sim v^3/(G^2m^2n \ln N)$.

Dynamical friction

- ▶ The assumption that the average of $\Delta \mathbf{v}$ is negligible is only true for the component orthogonal to the space velocity of the star.
- ▶ The longitudinal component (parallel to the velocity \mathbf{v} of the star) is of the form $\sum \Delta v_{\parallel} \sim -vt/t_r$.
- ▶ Because the effect is in the opposite direction to \mathbf{v} it is called “dynamical *friction*”
- ▶ The statistical effect of encounters is a balance between the “dissipative” effect of dynamical friction and the “heating” effect described by $\sum (\Delta v)^2$.
- ▶ These effects drive the velocities towards a Maxwellian distribution
- ▶ Because high-speed stars escape, this distribution is never achieved.

Significance of the relaxation time

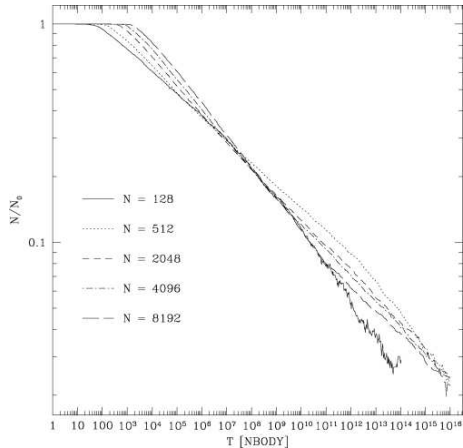
- ▶ Time scale of core collapse - here is central density v. time



Source: Makino 1996

Significance of the relaxation time (cont)

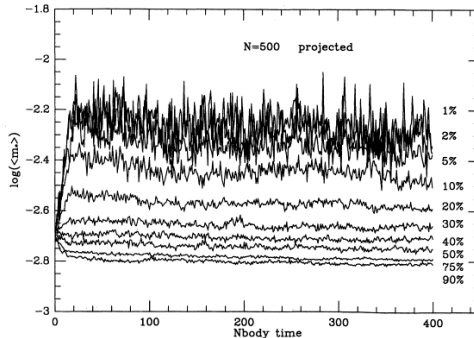
- ▶ Time scale of escape (actually several/many t_r)



Source: Baumgardt+ 2002

Significance of the relaxation time (cont)

- Time scale of *mass segregation* (if there is a distribution of masses, the heavier particles sink to the centre on a time scale which is a fraction of t_r)



Mean mass in Lagrangian shells against time (*Source: Giersz+ 1996*)

Why does core collapse happen?

- ▶ Relaxation acts like collisions in a gas: it causes “conduction” of energy from warmer regions (where the typical stellar speeds are large, i.e. in the core) to cooler (where the speeds are smaller, i.e. in the outer “halo”)
- ▶ Stars in the core are “cooled”, which makes them fall nearer to the centre, and speed up; stars in the halo are “heated”, which makes them move further from the centre, and slow down.
- ▶ This new “temperature” profile accentuates the temperature gradient, and the collapse of the core. There is an accelerating “gravothermal runaway” (or “gravothermal catastrophe”).
- ▶ The time scale of core collapse is the relaxation time scale.

The negative specific heat of self-gravitating systems

- ▶ Virial theorem $\dot{I} = 4T + 2V$
- ▶ Virial equilibrium: $4T + 2V \simeq 0$
- ▶ Energy $E = T + V$, and so $2T + 2E \simeq 0$
- ▶ Initial state $T_0, E_0 \simeq -T_0$
- ▶ Suppose system suddenly loses a little kinetic energy.
 - ▶ New state has $E_1 = T_1 + V_1 \simeq T_1 + V_0 < T_0 + V_0 = E_0$.
 - ▶ New state virialises with kinetic energy $T'_1 = -E_1 > -E_0 = T_0$
 - ▶ Thus, although kinetic energy was removed, the new kinetic energy is larger than the old kinetic energy
 - ▶ Strictly, the virial theorem does not apply (without modification) when applied to *part* of a system
- ▶ Example: an artificial satellite, subject to air drag, speeds up as it falls onto a lower orbit.

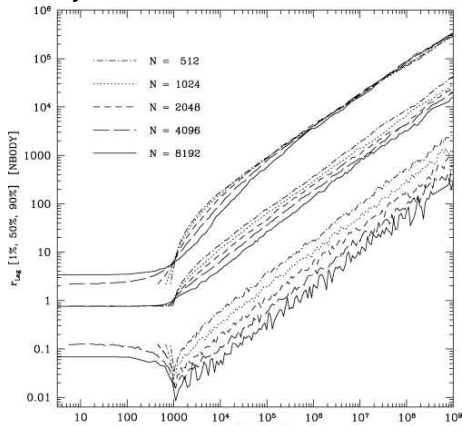
The Two Types of Stellar Dynamics

- ▶ *Collisional* stellar dynamics deals with phenomena on time scales of a few t_{rh} (open and globular star clusters; some galactic nuclei; core collapse; mass segregation)
- ▶ *Collisionless* stellar dynamics deals with phenomena on time scales much less than t_{rh} (spiral structure, galaxy collisions (!), virialisation)
- ▶ In the context of the phrase “Collisional stellar dynamics” the “collisions” are gravitational two-body encounters.

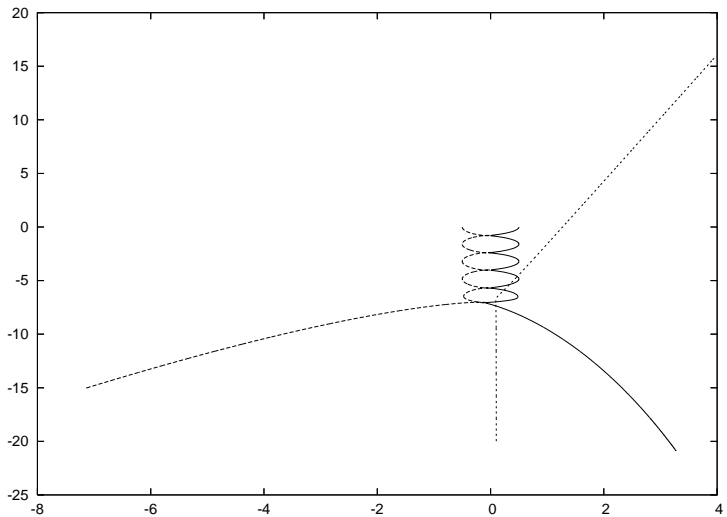
Post-collapse evolution

Depends on boundary conditions:

- ▶ “isolated” system: binaries form in the core, liberating energy, which expands the system on the time scale t_{rh} (by a feedback mechanism). As r_h expands, t_{rh} increases. The system very slowly loses mass



Formation of Binaries - an example



The dynamics of binary stars

- ▶ The dynamics of a binary star is a *two-body problem*, whose solution is well known (“Kepler’s problem”)
- ▶ A binary has two kinds of dynamical degrees of freedom
 - ▶ *internal*, i.e. the relative motion of the two stars (relative speed v), and their gravitational interaction (separation r)
 - ▶ *external*, i.e. the motion of the barycentre (centre of mass) of the two stars (speed V)
- ▶ The energy of a binary consisting of masses m_1, m_2 is

$$\begin{aligned}
 E &= \frac{1}{2}(m_1 + m_2)V^2 + \frac{1}{2} \frac{m_1 m_2}{m_1 + m_2} v^2 - \frac{Gm_1 m_2}{r} \\
 &= \frac{1}{2}(m_1 + m_2)V^2 - \frac{Gm_1 m_2}{2a},
 \end{aligned}$$

where a is the semi-major axis of the Kepler orbit

- ▶ An encounter of a binary with a single star is a *three-body problem*, and has no known general solution (example)

Three-body interactions



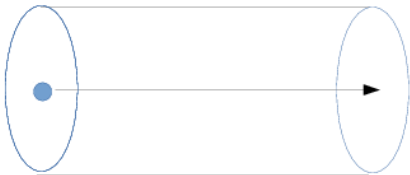
- ▶ If the three stars come to comparable distances, they exchange energy
- ▶ On average, the exchange of energy can be understood as an approach to equipartition
 - ▶ If $v \gg V$ then the internal degrees of freedom tend to lose kinetic energy in the encounter. Therefore the energy of the binary becomes more negative, and a decreases.
 - ▶ If $v \ll V$ then the internal degrees of freedom tend to gain kinetic energy, and a increases.

Hard and soft binaries

- ▶ Reminder: v is the relative speed of the components of a binary, V is its speed relative to an incoming third star.
- ▶ Now $v^2 \sim Gm/a$ (e.g. for a circular binary with equal masses m), and $V^2 \sim \sigma^2$, the rms speed of stars in the cluster which the binary encounters.
- ▶ The condition $v^2 \sim V^2$ is thus roughly the condition $Gm/a \sim \sigma^2$, or $a \sim Gm/\sigma^2$.
- ▶ Thus if $a \ll Gm/\sigma^2$ then $v^2 \gg V^2$, and a tends to decrease; and if $a \gg Gm/\sigma^2$, a tends to increase.
- ▶ Binaries with $a \ll Gm/\sigma^2$ are called (very) hard, and tend to become harder; binaries with $a \gg Gm/\sigma^2$ are called (very) soft, and tend to become softer.

The formation rate of binaries

- ▶ Only hard binaries are of interest, i.e. $a < Gm/\sigma^2$.
- ▶ Consider a system of density n stars per unit volume
- ▶ In time t , one star moves typical distance σt
- ▶ It encounters another star within a distance a if $n \cdot \pi a^2 \cdot \sigma t \sim 1$,
i.e. $t \sim 1/(\pi n \sigma a^2)$



The formation rate of binaries (contd)

- ▶ The probability that there is a third star within a distance $\sim a$ at this time is of order na^3 , and so the rate of formation of a hard binary is

$$\begin{aligned}\dot{n}_b &\sim (n/t)na^3 \text{ per unit volume} \\ &\sim n^3\sigma a^5 \\ &\sim \frac{G^5 m^5 n^3}{\sigma^9}\end{aligned}$$

where we have set $a \sim Gm/\sigma^2$ (threshold between hard and soft binaries)

- ▶ Because of the n -dependence, binary formation is concentrated in the core
- ▶ In core collapse $n, \sigma \rightarrow \infty$ and the volume tends to zero. But n wins.

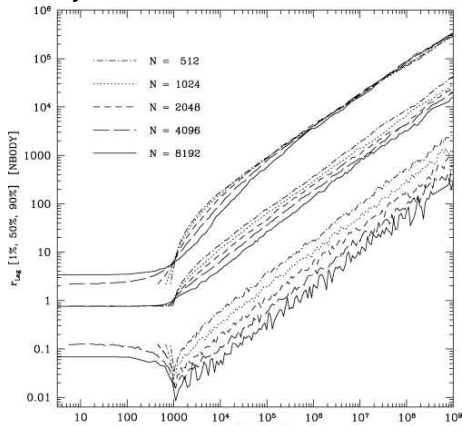
Energy generation by binaries

- ▶ When a binary forms in a three-body encounter, the energy of the three stars long before the encounter is their kinetic energy T_0 . (Their potential energy is small when they are far apart.)
- ▶ After the encounter, there is a binary with negative energy (in its barycentric frame).
- ▶ By energy conservation, the kinetic energy of the two products of the encounter (the barycentre of the binary, and the single star) must exceed T_0 .
- ▶ Encounters with an existing hard binary tend to have the same effect, i.e. an increase in the kinetic energy of the barycentre of the binary and the single star.
- ▶ Binary formation and evolution are an energy-generating mechanism, like nuclear reactions in a star.
- ▶ In this way, binaries can halt core collapse.

Post-collapse evolution (again)

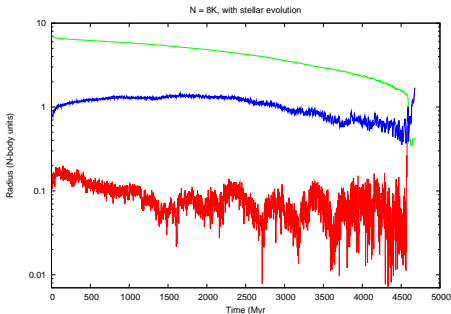
Depends on boundary conditions:

- ▶ “isolated” system: binaries form in the core, liberating energy, which expands the system on the time scale t_{rh} (by a feedback mechanism). As r_h expands, t_{rh} increases. The system very slowly loses mass



Post-collapse evolution (continued)

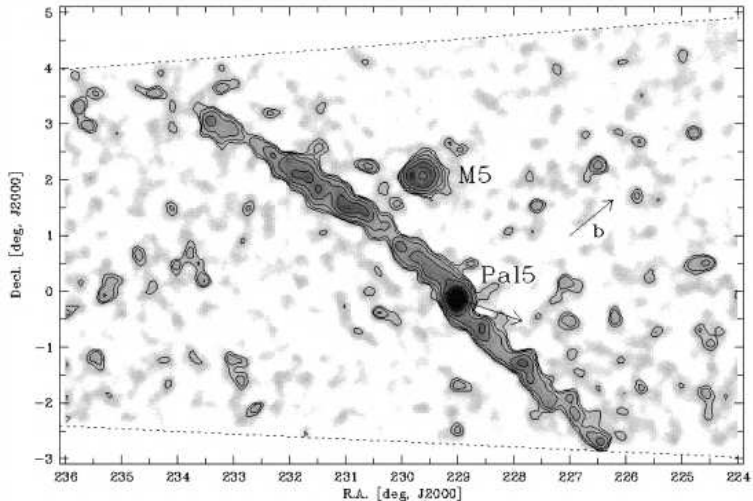
- ▶ “tidally limited” systems: stars escape (roughly speaking) at a tidal radius r_t , where external forces become dominant (see Lecture 4). Mass is lost on time scale t_{rh} ; r_t contracts, r_h contracts (eventually). System dissolves in few t_{rh} .



Plotted (from the top): r_t , r_h and the 1% Lagrangian radius (see Lecture 1)

Post-collapse evolution (continued)

In fact stars escape along “tidal tails”. Here is the example of Pal 5:



Computation and astrophysics of the N-body problem

Lecture 3

Outline of Lecture 2

1. Plummer's model
2. Two-body relaxation, dynamical friction
3. Core collapse, long-term evolution
4. Dynamics of binaries

***N*-body codes**

The equations of motion to be integrated are

$$\ddot{\mathbf{r}}_i = -G \sum_{j=1, \neq i}^N m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

(Lecture 1).

These can be written in the equivalent form

$$\begin{aligned}\dot{\mathbf{r}}_i &= \mathbf{v}_i \\ \dot{\mathbf{v}}_i &= \mathbf{a}_i = - \sum_{j=1, j \neq i}^N Gm_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}\end{aligned}$$

where $\mathbf{r}_i, \mathbf{v}_i$ are the position and velocity of the i th particle.

The Integration Algorithm

Example: Euler method Algorithm:

$$\begin{aligned}\mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \Delta t \mathbf{a}_i(t)\end{aligned}$$

where Δt is the “time step”. In this approximation, the velocity and acceleration of the particle are held constant for the duration of the time step.

Accuracy of the Euler Method

- ▶ A more accurate integrator for the position is Taylor's Theorem:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}(\Delta t)^2 \mathbf{a}_i(t) + \frac{1}{6}(\Delta t)^3 \mathbf{j}_i(t) + \dots,$$

where $\mathbf{j}_i = \dot{\mathbf{a}}_i$, i.e. the derivative of the acceleration, called the “jerk”.

- ▶ In Euler's algorithm we ignore all except the first two terms. This may be a satisfactory approximation if the series converges sufficiently rapidly.
- ▶ The radius of convergence of a Taylor Series is governed by the “distance” to the nearest singularity of the function being expanded. Here, this means the maximum value of Δt for which the function $\mathbf{r}_i(t + \Delta t)$ is well behaved.
- ▶ Thus $\Delta t \ll r/v$, where v is the speed, and r is the distance to the nearest neighbour

Time Step Control: Practice

The Taylor series for \mathbf{r}_i is

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}(\Delta t)^2 \mathbf{a}_i(t) + \frac{1}{6}(\Delta t)^3 \mathbf{j}_i(t) + \dots$$

This converges rapidly if $\frac{1}{6}(\Delta t)^3 |\mathbf{j}_i(t)| \ll \frac{1}{2}(\Delta t)^2 |\mathbf{a}_i(t)|$ (and similar relations between successive pairs of terms), i.e. if

$$\Delta t \ll \frac{|\mathbf{a}_i|}{|\mathbf{j}_i|},$$

This criterion would work well except if, by chance, the denominator were small. Therefore the criterion of choice incorporates several such ratios:

$$\Delta t = \left(\eta \frac{|\mathbf{a}_i|/|\mathbf{j}_i| + |\mathbf{j}_i|/|d\mathbf{j}_i/dt|}{|d^2\mathbf{j}_i/dt^2|/|d\mathbf{j}_i/dt| + |d\mathbf{j}_i/dt|/|\mathbf{j}_i|} \right)^{1/2}.$$

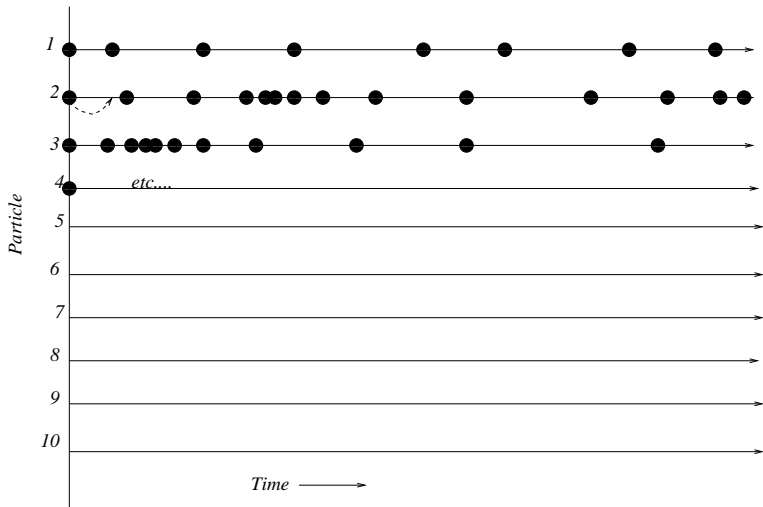
Choices of Time Step

$$\Delta t = \left(\eta \frac{|\mathbf{a}_i|/|\mathbf{j}_i| + |\mathbf{j}_i|/|d\mathbf{j}_i/dt|}{|d^2\mathbf{j}_i/dt^2|/|d\mathbf{j}_i/dt| + |d\mathbf{j}_i/dt|/|\mathbf{j}_i|} \right)^{1/2}.$$

This expression varies with time, and is different for different particles. The following implementations are possible.

- ▶ *Fixed* time step (the minimum over all particles for the entire simulation): too short, can't be predicted - impractical
- ▶ Variable *shared* time step (the minimum over all particles at the current time) - forces all particles to take same Δt - inefficient.
- ▶ Variable *individual* time step - near-optimal, but requires extrapolation.
- ▶ Block time steps ($\Delta t = 2^{-k}$, $k = 0, 1, 2, \dots$) - shares extrapolation

Individual Variable Time Steps



Extrapolation step: $\mathbf{r}_j(t + \delta t) = \mathbf{r}_j(t) + \delta t \mathbf{v}_j(t)$

The Hermite Integrator

A generalisation of Euler.

First stage:

$$\text{Euler: } \mathbf{r}_i := \mathbf{r}_i + \mathbf{v}_i \Delta t$$

$$\mathbf{v}_i := \mathbf{v}_i + \mathbf{a}_i \Delta t$$

$$\text{Hermite: } \mathbf{r}_i := \mathbf{r}_i + \mathbf{v}_i \Delta t + \frac{1}{2} \mathbf{a}_i \Delta t^2 + \frac{1}{6} \dot{\mathbf{a}}_i \Delta t^3$$

$$\mathbf{v}_i := \mathbf{v}_i + \mathbf{a}_i \Delta t + \frac{1}{2} \dot{\mathbf{a}}_i \Delta t^2$$

followed by a second stage (corrector) involving values of $\mathbf{a}_i, \dot{\mathbf{a}}_i$ at the *end* of the time step (Hermite only).

The Hermite Integrator: Correction Step

Let $\mathbf{v}, \mathbf{a}, \mathbf{j}$ be velocity, acceleration and jerk at the start of the step, and $\mathbf{v}', \mathbf{a}', \mathbf{j}'$ be the values at the end (computed after the first stage). Then the corrected formulae are

$$\mathbf{r}_i := \mathbf{r}_i + \frac{1}{2}(\mathbf{v}_i + \mathbf{v}'_i)\Delta t - \frac{1}{10}(\mathbf{a}'_i - \mathbf{a}_i)\Delta t^2 + \frac{1}{120}(\mathbf{j}_i + \mathbf{j}'_i)\Delta t^3$$

$$\mathbf{v}_i := \mathbf{v}_i + \frac{1}{2}(\mathbf{a}_i + \mathbf{a}'_i)\Delta t - \frac{1}{12}(\mathbf{j}'_i - \mathbf{j}_i)\Delta t^2.$$

Computation of the Jerk

The acceleration is

$$\mathbf{a}_i = -G \sum_{j=1, \neq i}^N m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3},$$

and so the jerk is

$$\dot{\mathbf{a}}_i = -G \sum_{j=1, \neq i}^N m_j \left(\frac{\mathbf{v}_i - \mathbf{v}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} - 3 \frac{(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^5} (\mathbf{r}_i - \mathbf{r}_j) \right).$$

Basic structure of an N -body code

1. Initialisation of $\mathbf{r}_i, \mathbf{v}_i, t_{next_i}$ (update time = $t_i + \Delta t_i$), $\mathbf{a}_i, \dot{\mathbf{a}}_i$, all i .
2. Choose i minimising t_{next_i}
3. Extrapolate all $\mathbf{r}_j, \mathbf{v}_j$ to t_{next_i}
4. Compute new $\mathbf{a}_i, \dot{\mathbf{a}}_i$
5. Correct new $\mathbf{r}_i, \mathbf{v}_i$ (Hermite integrator)
6. Compute new t_{next_i}
7. Repeat from step 2

Notes

- ▶ this does not include block time steps
- ▶ this is the basic structure of NBODY1, except for the choice of integrator. (The integrator is described in Bodenheimer et al).

Some Simple N -Body Codes with Variable Individual Time Steps

1. Binney & Tremaine 1e, 1987, PUP, Appendix 4.B (fortran)
2. Heggie & Hut, *The Gravitational Million-Body Problem*, 2004, CUP Appendix A, (matlab, C)
3. Hut & Makino, *The Art of Computational Science* (ruby)
<http://www.artcompsci.org/>
4. <http://www.nbabel.org/codes>
(C++,CUDA,F95,IDL,Java,PerlIDL,PyC++,Python,R,etc.)

Note: none of these codes would reach beyond core collapse in a reasonable time, because of the appearance of binary stars at that point.

Quality control

How do you know that the results of a simulation are correct?

- ▶ There are no useful exact solutions
- ▶ Conserved quantities:
 1. Momentum
 2. Angular Momentum
 3. Energy.

The most sensitive is energy.

Output from NBODY1:

T = 1.4 Q = 0.23 STEPS = 486 **DE = 0.000000 E = -0.250000**

TC = 0.5

<R> = 1.53 RCM = 0.0000 VCM = 0.0000 AZ = 0.00000 T6 =

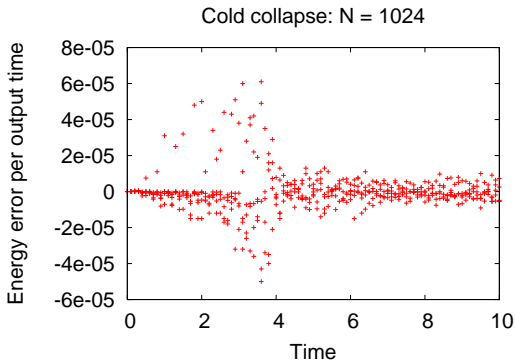
4 NRUN = 1

BINARY 6 15 0.040 0.040 -0.2 0.1840 3.6 0.0741 1.50 0.861 0

BINARY 8 9 0.040 0.040 -0.2 0.2517 2.2 0.1231 1.84 0.979 0

Quality control (continued)

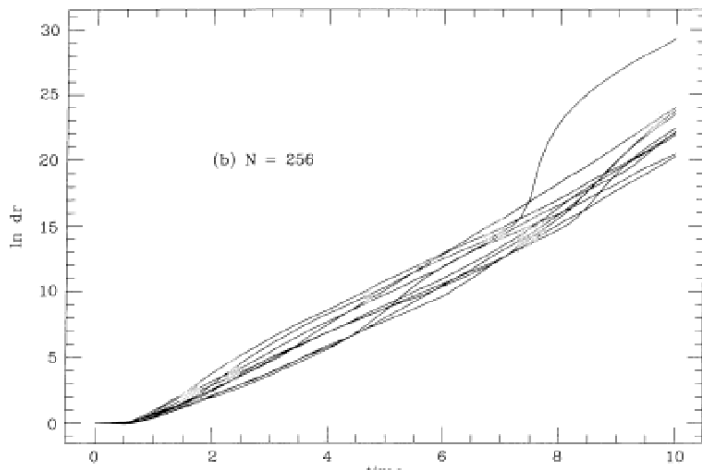
- ▶ (continued) Change in energy per output time



- ▶ Are your answers reasonable?

Growth of Errors in N -body simulations

Experiment: start two N -body simulations with slightly different initial conditions (say, one coordinate of one particle differing by 10^{-14}). How fast do the differences grow?



Growth of Errors in N -body simulations (continued)

- ▶ The growth of errors is exponential
- ▶ The time scale (in units of the crossing time) varies very slowly with N
- ▶ For a Plummer model, with equal masses, it is of order $0.1 t_{cr}$.
- ▶ Growth of error approximately 10^{18} when $\exp(t/(0.1 t_{cr})) = 10^{18}$, i.e. $t \simeq 4 t_{cr} \simeq 12 N$ -body units
- ▶ After a short time the positions and velocities of particles are wrong
- ▶ We *assume* that the statistical properties of the simulation are correct

“Complexity”

Recall structure:

1. Initialisation of $\mathbf{r}_i, \mathbf{v}_i, t_{next_i}$ (update time = $t_i + \Delta t_i$), $\mathbf{a}_i, \dot{\mathbf{a}}_i$, all i .
Done once
2. Choose i minimising t_{next_i} **Proportional to N**
3. Extrapolate all \mathbf{r}_j to t_{next_i} **Proportional to N**
4. Compute new $\mathbf{a}_i, \dot{\mathbf{a}}_i$ **Proportional to N**
5. Correct new \mathbf{r}_i , compute new \mathbf{v}_i (Hermite integrator)
6. Compute new t_{next_i}
7. Repeat from step 2

For each time step Δt , the computational effort is approximately proportional to N .

“Complexity” (continued)

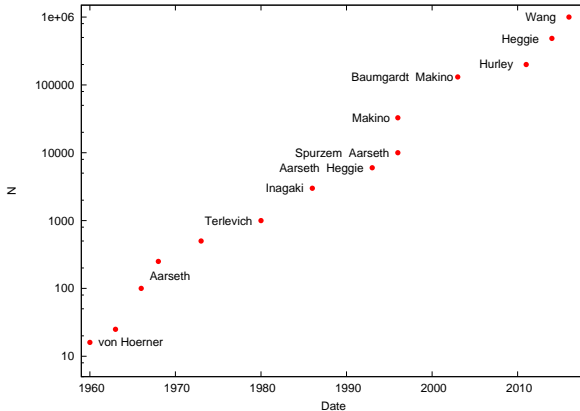
Typical time step is $\Delta t \sim \frac{r}{v}$, where v is typical speed and r is typical distance to nearest neighbour, i.e. $R/N^{1/3}$, where R is the virial radius. In Hénon (N -body) units (Lecture 1) $v \sim 1$, $R = 1$, and so $\Delta t \sim N^{-1/3}$.

Each time step takes of order N operations, and there are N particles. Hence the computational effort per N -body time unit is of order $N.N.N^{1/3} = N^{7/3}$.

Core collapse takes a few relaxation times, and $\frac{t_{rh}}{t_{cr}} \propto N / \ln \gamma N$ (Binney & Tremaine, Sec.1.2.1), where $t_{cr} = 2\sqrt{2}$. Hence the computational effort to core collapse varies as $N^{10/3} / \ln \gamma N$, or roughly N^3 .

The Development of N -body Simulations in History

Though computers have doubled in speed every 18 months until recently (Moore's Law), they have doubled in N much more slowly: about a factor 10 every 10 years.



The Challenge of Milky Way Globular Clusters for N -body

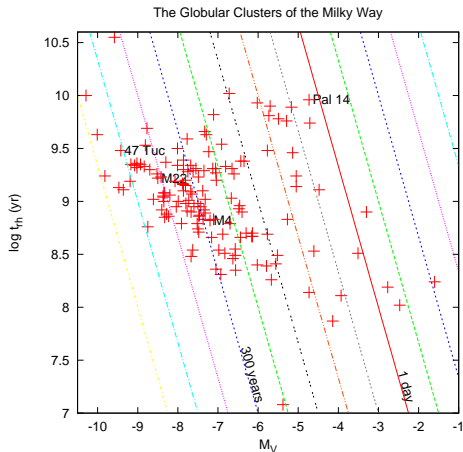


Figure: Contour lines of computational effort at intervals of 1dex

Three Bottlenecks in Each Time Step

The following stages are proportional to N in the basic structure of an N -body code:

- ▶ Choose i minimising t_{next_i} **Can be made proportional to $N^{1/2}$ (see NBODY1)**
- ▶ Extrapolate all \mathbf{r}_j to t_{next_i} **Can be reduced by use of “block time steps”** (mentioned earlier in this lecture)
 - ▶ Group together particles which have very similar update times. The extrapolation is shared among them. Still proportional to N , but much smaller importance.
- ▶ Compute new $\mathbf{a}_i, \dot{\mathbf{a}}_i$ **This is the main residual bottleneck**
- ▶ Software solutions:
 - ▶ Neighbour scheme; incorporated in NBODY2
 - ▶ Tree codes (hierarchical schemes): see Bodenheimer et al
 - ▶ Grid-based methods, e.g. finite differences

Accelerating force calculation: hardware

1. Parallel computation. Example (code fragment of NBODY1, showing calculation of force on particle I)

```
DO 10 J = 1,N
  IF (J.EQ.I) GO TO 10
  A1 = X(1,J) - XI
  A2 = X(2,J) - YI
  A3 = X(3,J) - ZI
  RIJ2 = A1*A1 + A2*A2 + A3*A3 + EPS2
  A5 = BODY(J)/(RIJ2*SQRT(RIJ2))
  FIRR(1) = FIRR(1) + A1*A5
  FIRR(2) = FIRR(2) + A2*A5
  FIRR(3) = FIRR(3) + A3*A5
10 CONTINUE
```

The calculation of $A1*A5$, $A2*A5$, $A3*A5$ can be done simultaneously for different particles J, because there is no data dependency: the result of one calculation does not affect any other.

Accelerating force calculation: hardware (continued)

Parallel computers (continued)

- ▶ Calculations for different J-particles can be farmed out to different processors on the one parallel computer, and/or different cores on a single processor.
- ▶ For supercomputers or large clusters, best code is NBODY6++
(<https://github.com/lwang-astro/betanb6pp/branches>)
- ▶ For low-budget users, best option is video cards: Graphics Processing Unit (GPU)



- ▶ £200 (5000 CZK) each, speedup x100
- ▶ Code: NBODY6 + GPU2
(<http://www.ast.cam.ac.uk/~sverre/web/pages/nbody.htm>)

Software refinements

Close encounters and few-body subsystems

Two problems when particles come very close together:

1. Subtraction of positions of two close neighbours causes an increase in relative rounding error
2. Reduced time step, and the simulation may slow down dramatically.

Close encounters and binaries. I. Offset “regularisation”

Suppose particles i, j form a bound pair, or experience a close encounter. Use offset variables (Jacobi coordinates) \mathbf{r}, \mathbf{R} defined as

$$\begin{aligned}\mathbf{R} &= \frac{m_i \mathbf{r}_i + m_j \mathbf{r}_j}{m_i + m_j} \\ \mathbf{r} &= \mathbf{r}_i - \mathbf{r}_j,\end{aligned}$$

and write equations of motion in terms of \mathbf{r}, \mathbf{R} : e.g.

$$\ddot{\mathbf{r}} = -G(m_i + m_j) \frac{\mathbf{r}}{|\mathbf{r}|^3} + \mathbf{a}'_i - \mathbf{a}'_j,$$

where \prime means we omit force due to i, j .

Advantage: avoids rounding error in repeated calculation of $\mathbf{r}_i - \mathbf{r}_j$.

Note: \mathbf{R} is centre of mass position vector, \mathbf{r} is position of particle j relative to particle i

Close encounters and binaries. II. KS regularisation

- ▶ Singularity in

$$\ddot{\mathbf{r}} = -G(m_i + m_j) \frac{\mathbf{r}}{|\mathbf{r}|^3} + \mathbf{a}'_i - \mathbf{a}'_j$$

requires small time steps for close and/or eccentric binaries.

- ▶ KS regularisation is subtle change of variables which removes the singularity.

Example: one-dimensional regularisation

Unperturbed binary motion is

$$\ddot{x} = -\frac{1}{x^2}$$

where we have scaled to units such that $G(m_i + m_j) = 1$.
Introduce new variables z, τ (transformed coordinate and time)
such that

$$\begin{aligned}x &= z^2 \\ \frac{dt}{d\tau} &= x\end{aligned}$$

One-dimensional regularisation (continued)

Since $x = z^2$, $\dot{x} = 2z\dot{z}$.

Since $dt/d\tau = x$,

$$\begin{aligned}\dot{x} &= 2z\dot{z}' \frac{d\tau}{dt} \text{ where } ' \text{ means } d/d\tau \\ &= \frac{2z\dot{z}'}{x} \text{ since } dt/d\tau = x \\ &= \frac{2\dot{z}'}{z}.\end{aligned}$$

Differentiating again with respect to time gives similarly

$$\begin{aligned}\ddot{x} &= \frac{2\ddot{z}'}{z^3} - 2\frac{\dot{z}'^2}{z^4} \\ &= -\frac{1}{x^2} \text{ (equation of motion)} \\ &= -\frac{1}{z^4}\end{aligned}$$

One-dimensional regularisation (continued)

We have

$$\frac{2z''}{z^3} - 2\frac{z'^2}{z^4} = -\frac{1}{z^4}$$

Hence

$$\begin{aligned} z'' &= \frac{z'^2}{z} - \frac{1}{2z} \\ &= \frac{1}{2}z \left(\frac{2z'^2}{z^2} - \frac{1}{z^2} \right) \\ &= \frac{1}{2}z \left(\frac{1}{2}\dot{x}^2 - \frac{1}{x} \right) \\ &= \frac{1}{2}hz, \end{aligned}$$

where

$$h = \frac{1}{2}\dot{x}^2 - \frac{1}{x}.$$

One-dimensional regularisation (continued)

Now

$$h = \frac{1}{2}\dot{x}^2 - \frac{1}{x}$$

is just the energy of the binary (per unit [reduced] mass); the kinetic energy is $\dot{x}^2/2$, the potential energy is $-1/x$. Thus in the transformed equation of motion

$$z'' = \frac{1}{2}hz,$$

the coefficient of z is constant. This is the simple harmonic oscillator equation.

Bottom line: we have transformed the one-dimensional Kepler problem

$$\ddot{x} = -\frac{1}{x^2}$$

(which is singular at $x = 0$) into the simple harmonic oscillator equation, which is regular everywhere.

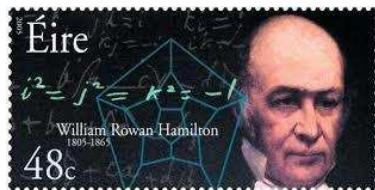
Two-body regularisation: practicalities

Recall: regularisation is the process which transforms the Kepler equation of motion into the simple harmonic oscillator equation. It was established in the one-dimensional case.

- ▶ Need a three-dimensional version. This exists and is called KS (or quaternion) regularisation.
- ▶ Candidates for regularisation recognised by short time steps
- ▶ Deregularisation if the binary is too strongly perturbed
- ▶ Still requires short time step for close binary
- ▶ Freeze unperturbed binaries
- ▶ Code: NBODY3

KS regularisation

- ▶ quaternions are a generalisation of complex numbers
- ▶ A quaternion is written as $a + ib + cj + dk$, where the units i, j, k satisfy $i^2 = j^2 = k^2 = -1$ and $ijk = -1$. Hence $-i = i^2jk = -jk$ and, more generally, $ki = j, ij = k$. Their arithmetic is non-commutative.
- ▶ Every quaternion has a *conjugate* defined to be $a - ib - jc - kd$.
- ▶ Three-vectors $\mathbf{r} = (x, y, z)$ can be mapped to the pure imaginary quaternions $ix + jy + kz$
- ▶ Any three-vector (written as a quaternion) can be expressed as $ix + jy + kz = qi\bar{q}$ for some quaternion q .
- ▶ This is the generalisation of the one-dimensional transformation $x = z^2$.

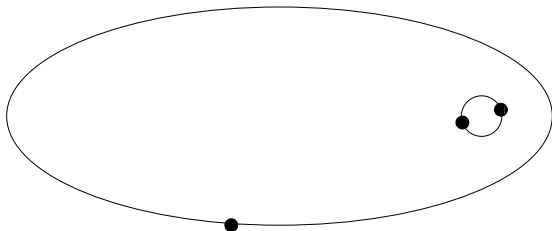


KS regularisation (contd)

- ▶ The same time-transformation $\frac{dt}{d\tau} = r$ is used
- ▶ A further assumption that $\dot{q}i\bar{q} = qi\dot{\bar{q}}$ is made
- ▶ Then the equation of Kepler motion $\frac{d^2\mathbf{r}}{dt^2} = -\frac{Gm\mathbf{r}}{r^3}$ transforms again to $\frac{d^2q}{d\tau^2} = \frac{h}{2}q$.
- ▶ See Heggie & Hut, ch.15
- ▶ In practice, the transformations are written in terms of spinors (2×2 complex matrices) or 4×4 real matrices

Higher-order subsystems: triples, quadruples, etc

- ▶ hierarchical triples are binaries constantly perturbed by a third body: there is a procedure called “slow-down” which follows secular perturbations with (much) larger time step



- ▶ non-hierarchical triples, quadruples: *chain regularisation*, a generalisation of offset and KS regularisation; there are specialisations to triples and quadruples; codes TRIPLE, CHAIN on Aarseth's download page

Exercise for Lecture 3

1. Examine the output of a run of NBODY1 which completed successfully. The last line gives “CPUTOT”, which is the execution time in minutes. (The data may also be obtained using the unix command “time”.) The last main output will also give the number of integration steps completed (“STEPS”). Repeat for larger values of N . (If you find that a run terminates quickly with “FATAL ERROR! BAD INPUT $N = 250$ ” (or something like that) go to the Real8 subdirectory, edit the parameter NMAX in params.h, and recompile.) Can you understand the dependence of the computing time on N , assuming that the computational effort is dominated by the force calculation?

Computation and astrophysics of the N-body problem

Lecture 4

Summary of Lecture 3

- ▶ *N*-body codes: integrators and step size control
- ▶ Quality control
- ▶ Complexity
- ▶ Application to Globular Star Clusters
- ▶ Accelerating the force computation in software and hardware
- ▶ Theory of regularisation; subsystems

Lecture 4: Essential Astrophysics

1. External forces
2. Initial structure
3. The mass function
4. Primordial binaries
5. Stellar evolution
6. Collisions

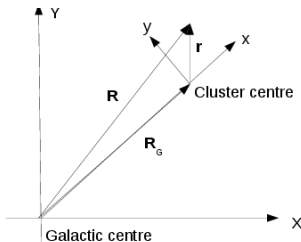
Lecture 4: Essential Astrophysics

1. External forces
2. Initial structure
3. The mass function
4. Primordial binaries
5. Stellar evolution
6. Collisions

And finally: the kitchen sink!

I. External Forces - the Galactic Tide

- ▶ Consider a cluster moving on a circular orbit of radius R_G about the Galactic Centre, with angular speed $\omega(R_G)$
- ▶ Use a rotating, accelerating frame of reference with origin at the centre of the cluster, x -axis pointing away from the Galactic Centre, y in the direction of cluster motion about the Galactic Centre



- ▶ This introduces centrifugal and Coriolis accelerations.

Equations of motion

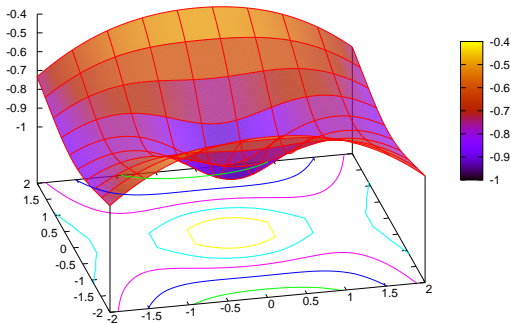
- ▶ $\ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \mathbf{v} + \omega^2 z \mathbf{e}_z + 2R_G \omega \omega' x \mathbf{e}_x = -\nabla \Phi_c(\mathbf{r})$, where
 - ▶ Φ_c is the potential due to cluster stars
 - ▶ $\mathbf{e}_x, \mathbf{e}_z$ are unit x - and z - vectors
 - ▶ $\omega = |\boldsymbol{\omega}|$ is the angular speed of the cluster about the Galaxy, and $\boldsymbol{\omega}$ is its angular velocity vector (orthogonal to the plane of motion of the cluster)
- ▶ Note Coriolis term, those due to a combination of centrifugal and “tidal” Galactic acceleration, and that due to the cluster
- ▶ Here the Galactic acceleration is given in a linear approximation (the “tidal” approximation), which is justified when the radius of the cluster is much smaller than R_G
- ▶ $\ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \mathbf{v} = -\nabla \Phi_{eff}(\mathbf{r})$, where the “effective” potential is

$$\Phi_{eff}(\mathbf{r}) = \Phi_c(\mathbf{r}) + R_G \omega \omega' x^2 + \frac{1}{2} \omega^2 z^2$$

- ▶ “Energy” $E = v^2/2 + \Phi_{eff}(\mathbf{r})$ is conserved
- ▶ $\Phi_{eff}(\mathbf{r}) \leq E$: a star may be confined by equipotentials of Φ_{eff}

External Forces - the Galactic Tide (continued)

- ▶ (Effective) potential and equipotentials in the plane of motion of the cluster ($z = 0$)



- ▶ The cluster sits in the central potential well
- ▶ For a star to escape, its energy E must exceed the value of Φ_{eff} at the two saddle points

Three dimensions

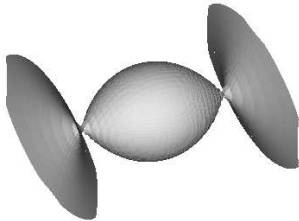


Figure: The equipotential surface through the saddles

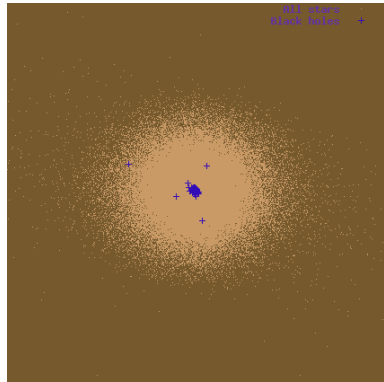
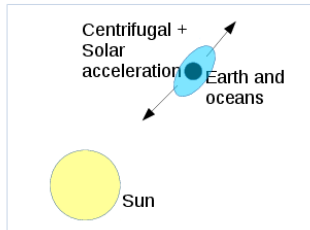
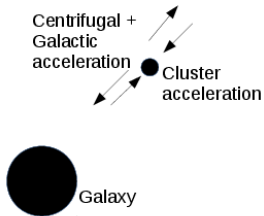
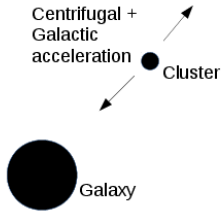
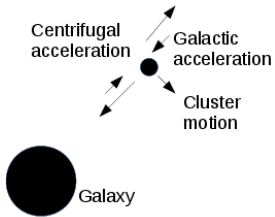


Figure: The equipotential surface is not spherical, and it shapes the star cluster

Lagrange points

- ▶ The saddle points are called *Lagrange points*
- ▶ They are equilibria, because $\nabla\Phi_{eff} = \mathbf{0}$ there
- ▶ Equation of motion is
$$\ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \mathbf{v} + \omega^2 z \mathbf{e}_z + 2R_G \omega \omega' x \mathbf{e}_x = -\nabla\Phi_c(\mathbf{r})$$
- ▶ x-component is $\ddot{x} - 2\omega\dot{y} + 2R_G\omega\omega'x = -\partial\Phi_c(\mathbf{r})/\partial x$
- ▶ For a stationary star at a Lagrange point
$$2R_G\omega\omega'x = -\partial\Phi_c(\mathbf{r})/\partial x$$
- ▶ Assume...
 - ▶ ...Lagrange point is on x-axis. Then $y = z = 0$ and $r = |x|$
 - ▶ ...most of the mass of the cluster is close to the origin. Then $\Phi_c \simeq -GM/r = -GM/|x|$, where M is the cluster mass.
 - ▶ ...Galaxy has a flat rotation curve, i.e. $\omega(R_G) = V/R_G$, where V is constant.
- ▶ Then $-2\omega^2x = -GMx/|x|^3$, and so $|x| = \left(\frac{GM}{2\omega^2}\right)^{1/3}$
- ▶ This is the “tidal radius” (or “Jacobi radius”) r_t of the cluster
- ▶ Along with the core radius r_c and the half-mass (or instead the half-light) radius r_h , these three radii characterise the spatial structure of a star cluster.

Basic tidal dynamics



Essential astrophysics II. Initial structure

- ▶ For convenience use Plummer's model. Specify M_c and a (the “scale radius” of the model)
- ▶ Very common: use King's models, which have finite radius

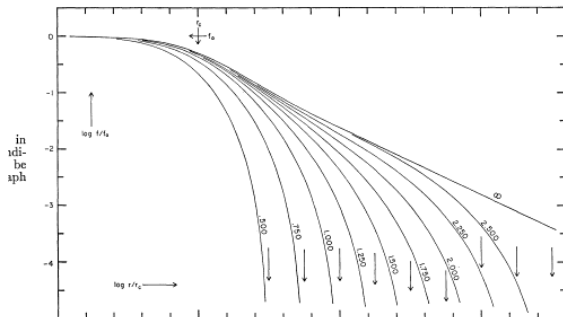


Figure: Surface density profile of King's models with concentration $c = 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, \infty$

Specify M_c , the edge radius (often taken to be the tidal radius), and the concentration c (or, the “scaled central potential” W_0)

Essential astrophysics III. Initial mass function

- ▶ Most common: Kroupa mass function (see Aarseth, Tout & Mardling 2003)
- ▶ Simplest form has probability density

$$f(m) = \begin{cases} A(m/m_b)^a, & m_a < m < m_b \\ A(m/m_b)^b, & m_b < m < m_c \end{cases}$$

where

- ▶ A is constant
- ▶ $a \simeq -1.3, b \simeq -2.3$
- ▶ $m_a = 0.08M_\odot, m_b = 0.5M_\odot, m_c = 100M_\odot$
- ▶ There are several variants, e.g. including brown dwarfs
- ▶ There are several other mass functions, e.g. Miller-Scalo

Essential astrophysics IV. Primordial binaries

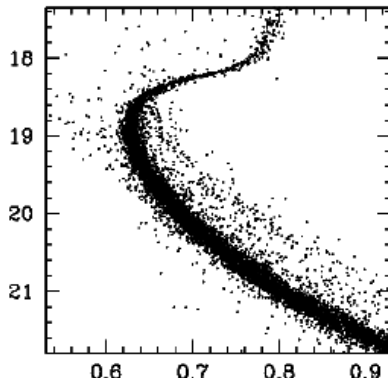


Figure: Photometric offset binaries in NGC288 (Milone et al 2012)

Many choices to be made

- ▶ Masses of components
- ▶ Distribution of semi-major axis and eccentricity

Essential astrophysics V. Stellar evolution

Three levels of sophistication

1. Give each star a (main sequence) lifetime and, at that time, immediately change its mass to that of the stellar remnant (black hole, neutron star, white dwarf). Quick and effective method for approximate modelling of dynamical effects of stellar evolution.
2. Use formulae which have been fitted to the results of stellar evolution models. This gives the evolution of the mass, radius, temperature, from which magnitudes and colours can be derived. This is the most common method.
3. Use a “live stellar evolution” code. Only attempted in recent years. Requires robust code. Slow.

Similar tools are also needed to deal with stellar evolution of binary stars.

VI. Stellar Collisions

Alternatives:

- ▶ Sticky spheres: assume stars coalesce to give a single more massive star
- ▶ Catalogues of precomputed collision simulations (not available yet in any collisional N -body code)
- ▶ Live computation of collisions as they happen.
 - ▶ Example: J. Lombardi

Notes:

- ▶ For techniques for computing stellar collisions, see Bodenheimer et al
- ▶ For a code which integrates many types of stellar dynamics, stellar evolution and collision hydrodynamics, see <http://amusecode.org/>

Flow control

Each integration step may involve any of the following possibilities

1. Standard integration (may be KS)
Only this and 4 (below) were present in NBODY1
2. New KS regularisation
3. KS termination
4. Output
5. 3-body regularisation (See last slide of Lecture 3)
6. 4-body regularisation (“)
7. New hierarchical system (“)
8. Termination of hierarchical system (“)
9. Chain regularisation (“)
10. Physical collisions
11. Stellar evolution

Flow control

Each integration step may involve any of the following possibilities

1. Standard integration (may be KS)
Only this and 4 (below) were present in NBODY1
2. New KS regularisation
3. KS termination
4. Output
5. 3-body regularisation (See last slide of Lecture 3)
6. 4-body regularisation (“)
7. New hierarchical system (“)
8. Termination of hierarchical system (“)
9. Chain regularisation (“)
10. Physical collisions
11. Stellar evolution

Professional *N*-body codes

1. NBODY6 (Aarseth): general purpose hardware [and optional GPU]
2. NBODY6tt (Renaud): NBODY6 with more versatile tidal effects
3. NBODY6++ (Spurzem): parallel computers and/or GPU
4. AMUSE (everyone): Python, FORTRAN, C++, C; includes a Hermite code, NBODY1h, a Barnes-Hut tree code, etc.
Available from <http://amusecode.org/>
5. starlab (McMillan, Hut, Makino, Portegies Zwart): general purpose/GPU-enabled hardware (*no KS regularisation*)
 - ▶ Available from <http://www.ids.ias.edu/~starlab/install/>

How to make a movie

- ▶ Go to <http://www.sns.ias.edu/~starlab/install/> and follow the instructions to download, compile and install Starlab.
- ▶ At the command line, enter

```
makeplummer -n 1024 -i \  
    | makemass -l 0.5 -u 5 \  
    | scale -s \  
    | kira -t 10 -D -5 \  
    | snap_to_image -s 400 -m -p -2 -g -a -d -f plummer-run
```
- ▶ This pipeline makes a Plummer model with $N = 1024$, distributes masses in default distribution between 0.5 and 5 units, scales to standard units, integrates to $t = 10$ with snapshots at intervals of 2^{-5} , and converts to image files `plummer-run.gif` . Takes about 5 min.
- ▶ Make an animated gif with

```
convert -delay 0 plummer*.gif -loop 0 playplummer.gif
```

`playplummer.gif` can be opened in a browser
- ▶ Example: a small star cluster in a tidal field

How to Simulate a Star Cluster in a Tidal Field with NBODY6tt

1. Go to the web page <https://github.com/florentrenaud/nbody6tt>
2. Press the green button “Clone or download”
3. In the pop-up window select “Download ZIP”
4. In the new pop-up window select “Save File” (on Firefox)
5. Go to your downloads folder (“cd ~/Downloads”)
6. Unzip the code (“unzip nbody6tt-master.zip”), which creates a subdirectory nbody6tt-master
7. Go to the source subdirectory (“cd nbody6tt-master/Ncode”)
8. Make the code (“make”)
9. Go to the parent directory (“cd ..”)
10. Make a directory for your runs (“mkdir test”)
11. Go there (“cd test”)
12. Create an input file tt.in by copying a provided sample input file (“cp ../Docs/input tt.in”)
13. Run the code (“../Ncode/nbody6 < tt.in > tt.out”, ~ 30 sec)

Termination of Run on Energy Error

- ▶ For me the run ended with the following output lines (in tt.out):
ADJUST: TIME = 50.00 Q = 0.51 **DE = 5.6E-03** E = -0.260140
RMIN = 4.6E-03 DTMIN = 2.4E-04 TC = 7 DELTA = 8.6E-04
E(3) = -0.129606 DETOT = 0.000884

CALCULATIONS HALTED * * *

- ▶ The energy error in the output step (DE) has exceeded the allowed tolerance (the 7th entry in the third line of the input file, which is
0.02 0.02 0.35 2.0 10.0 1000.0 **4.0E-05** 2.0 0.5)

Recovering from an excessive energy error

- ▶ Two solutions:
 1. Rerun with new accuracy parameters (analogous to η in time step criterion): ETAI, ETAR, ETAU (see define.f in directory Ncode). Disadvantage: gives an essentially different run, because of rapid growth of errors (Lecture 3)
 2. Rerun with greater tolerance of error: parameter QE in input. Disadvantage: how big an energy error can be tolerated?
- ▶ NBODY6tt has an option for recovering by (i) automatically changing the three accuracy parameters and (ii) reading a checkpoint file from the output time before the energy error occurred (an efficient variant of solution 1 above)
 - ▶ To activate this, change the 4th line of tt.in from
1 1 0 0 1 0 1 0 0 0
to
1 2 0 0 1 0 1 0 0 0
 - ▶ Delete the file ESC and HIARCH: "rm ESC HIARCH"
 - ▶ Rerun with "../Ncode/nbody6 < tt.in > tt.out" (about 90 sec)
 - ▶ Ends normally with
END RUN TIME = 318.0 CPUTOT = 0.0 ERRTOT = 0.000190
DETOT = 0.000008

How to Simulate a Star Cluster in a Tidal Field (continued)

14. Typical output extract:

```
T = 10. N = 500 <NB> = 9 KS = 0 NM = 0 MM = 0 NS = 500
NSTEP S = 756717 203 201945 448 DE = -1.7E-05 E =
-0.250673 M = 1.0000
NRUN = 1 M# = 0 CPU = 0.1 TRC = 0.6 DMIN = 3.6E-05
3.6E-05 1.0E+02 1 .0E+02 AMIN = 1.0E+02 RMAX =
0.0E+00 RSMIN = 0.090 NEFF = 259
<R> RTIDE RDENS RC NC MC RHOD RHOM CMAX jCnĴ
lr/R UN NP RCM VCM AZ EB/E EM/E TCR T6 NESC VRMS
#1 0.78 5.1 0.10 0.2913 36 0.133 5. 20. 3. 16.3 0.105 0 0
0.000 0.0000 0.0807 -0.000 -0.000 2.82 21 0 0.6
```

15. For illustration we extract and print some numbers from the last line with the awk script
- ```
{if ($1=="#1") print $21,$5,$2,$3}
```

## 15. (continued)

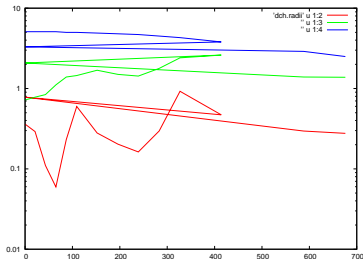
(These are time in millions of years, and the core<sup>2</sup>, half-mass and tidal radii; in NBODY6, with the parameters we are using,  $\langle R \rangle$  really *is* the half-mass radius, whereas in NBODY1  $\langle R \rangle$  is the virial radius.)

## 16. Suppose the awk output is in the file 'tt.radii'.

## 17. Now use gnuplot to plot these radii:

```
set log y; plot 'tt.radii' u 1:2 w l, u 1:3 w l, u 1:4 w l
```

(Note: " means two separate single quotes)




---

<sup>2</sup>The core radius is, roughly speaking, the radius of the inner part of the system, in which the density is at least half the central density.

## How to Simulate a Star Cluster in a Tidal Field (continued)

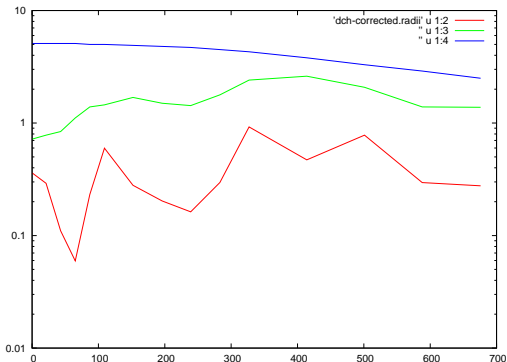
The problem here is a badly formatted line in tt.out (which may occur at different places in different runs, or not at all):

```
#1 2.08 3.3 14.75 0.7801 9 0.040 2. 7. 3. 7.2 0.088 0 0 14.764
0.1791 3.9346 0.000 0.000139.43 501 0 0.4
```

which can be circumvented with a modified awk script:

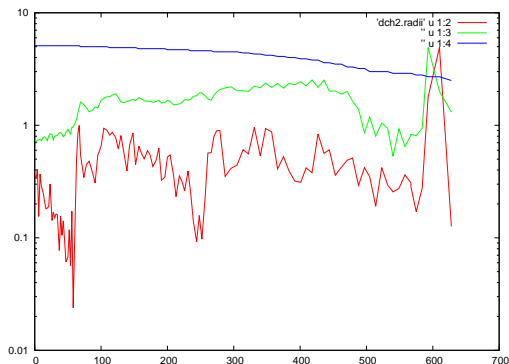
```
{if ($1=="#1") {
 print $(NF-2),$5,$2,$3
}
}
```

(NF is the number of fields in the line, normally 23)

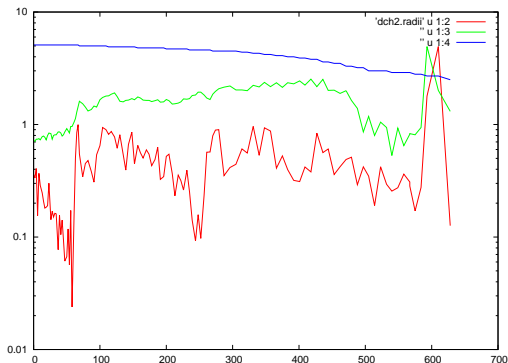


# Changing the output frequency

- Change the third input line from  
0.02 0.02 0.35 2.0 10.0 1000.0 4.0E-05 2.0 0.5  
to  
0.02 0.02 0.35 2.0 1.0 1000.0 4.0E-05 2.0 0.5  
which now gives output every time unit.



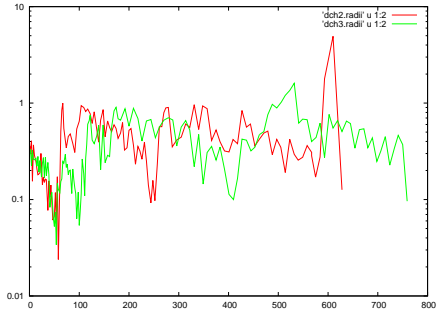
# Interpretation of the results



- ▶ Tidal radius shrinks as stars escape, and the mass of the cluster decreases
- ▶ Core radius shrinks: core collapse at about  $t = 60$ .
- ▶ Reversed by binary formation and hardening, which increases the energy of the cluster. This causes the rise in  $r_h$
- ▶ Large fluctuations near the end of the run, when  $N = 22$  only.
- ▶ Is that a second core collapse at about  $t = 250$ ?

## Statistical Significance of Results

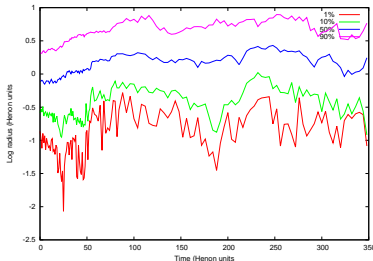
- ▶ Repeat the simulation with a different random number seed.
- ▶ Line 2 of input is  
500 1 25 **10000** 70 1  
Change to  
500 1 25 **10001** 70 1
- ▶ Plot core radii of both runs:
- ▶ Recollapses may occur, but at various times
- ▶ The lifetimes of different runs may differ by tens of percent.





## NBODY6tt: Some optional output

- ▶ fort.14: time and  $\log_{10}$  of Lagrangian radii
  - ▶ A Lagrangian radius is a radius containing a fixed fraction of the mass.
  - ▶ The fractions used are in the line “DATA FLAGR” in lagr.f in the Ncode subdirectory
  - ▶ Change line 4 of input file to  
1 2 0 0 1 0 **5** 0 0 0



- ▶ Increasing output interval can be overridden with option 32

## Other optional output (continued)

- fort.36 gives the mean mass in each Lagrangian shell

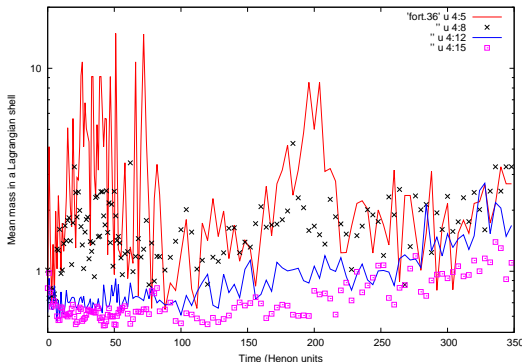


Figure: Mean mass in four Lagrangian shells

- Note mass segregation, especially around core collapse
- General trend of increasing mean mass with time (preferential escape of lighter stars)

## NBODY6: Other Standard Output (continued)

- ▶ Include stellar evolution by changing option 19 to 3 and 12 to 1
- ▶ Remove ESC and HIARCH, and rerun
- ▶ fort.83: information on the stellar properties of the particles

- ▶ Typical output

```
BEGIN 501 0.0
```

```
1 1 5.812 5.656 2.979 0.337 4.276
```

```
2 1 0.705 5.094 2.828 0.310 4.252
```

```
.....
```

- ▶ These are

N TIME (number of “objects”, time)

NAME KW RI M1 ZL1 R1 TE (for each single star)

where

- ▶ NAME identifies the star
- ▶ KW: stellar type (see end of define.f in Ncode)
- ▶ RI: distance from the cluster centre (units of the core radius)
- ▶ M1: mass (solar masses)
- ▶ ZL1: log luminosity (solar luminosities)
- ▶ R1: log radius (solar radii)
- ▶ TE: log effective temperature (K)

## NBODY6: Other Standard Output (continued)

Data for a single time can be obtained with the following awk script (snapshot.awk)

```
{if (NF==2) t = $2}
{if (t==time&&NF==7) print $0}
```

where the time is given in the awk command:

```
awk -f snapshot.awk time=0.0 fort.83 > snap0.0
```

Then the colour-magnitude (actually  $T_{\text{eff}}-L$ ) diagram is given with gnuplot:

```
plot [4.3:3.4] 'snap0.0' u 7:5
```

(The plot range [4.3:3.4] shows that  $T$  should increase to the left.)

## Example: HR Diagram at 0 and 615 Myr

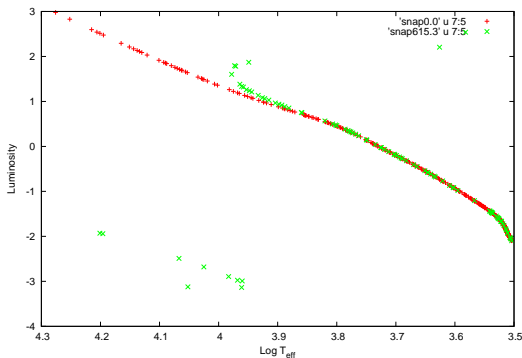


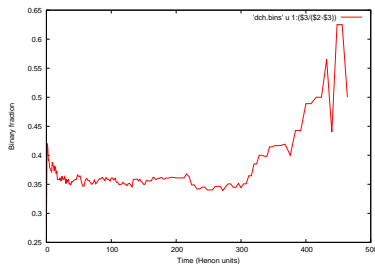
Figure: Note main sequence, giant branch, white dwarfs

# Primordial binaries

- ▶ The file Docs/inbins contains sample initial conditions
- ▶ For better comparison with previous runs we change
  - ▶ Line 2 to **500** 1 20 129000 85 1
  - ▶ Line 3 to 0.02 0.02 0.3 2.0 **1.0** 1000.0 5.0E-04 2.0 0.6
  - ▶ Line 4 to 1 **2** 0 0 1 0 1 4 0 0
  - ▶ Line 10 to 2.3 5.0 0.2 **250** 0 0.02 0.0 100.0
- ▶ The model contains 50% binaries
- ▶ It takes about 5 min

# Evolution of the binary fraction

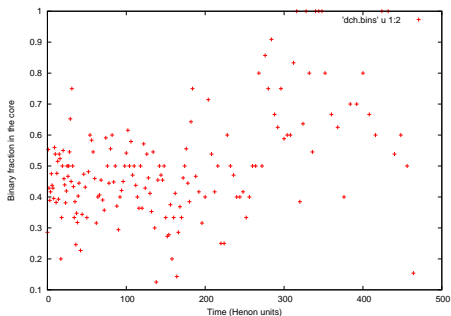
- ▶ Defined as  $\frac{\text{\#binaries}}{\text{\#binaries} + \text{\#number of single stars}}$



- ▶ Calculated by replacing  $\text{\#binaries}$  by the number of KS (regularised) pairs (called KS in output)
- ▶ Soft pairs are not usually regularised, hence the initial value is less than 50%
  - ▶ See Lecture 2 for notions of “soft” and “hard” pairs
- ▶ Early decrease caused by destruction of soft(ish) pairs
- ▶ Late increase caused by mass segregation of hard binaries

# Evolution of the binary fraction in the core

- ▶ Number of objects (single stars + binaries) in the core is field 6 in line beginning “#1”
- ▶ Number of binaries in the core is NC in the line beginning “BINARIES”



- ▶ About 50% for the first half of the evolution
- ▶ Late increase caused by mass segregation of hard binaries



## Tidal tails again

- ▶ Previously, the tails were straight, because of the tidal approximation
- ▶ With NBODY6tt this approximation can be avoided. The tidal acceleration, and the orbit of the cluster inside the Galaxy, are calculated numerically.
- ▶ Example (*Peter Berczik*)

# Outline of Lecture 4

- ▶ Essential astrophysics
  1. Tides
  2. King models
  3. The IMF
  4. Primordial binaries
  5. Stellar evolution
  6. Collisions
- ▶ Kitchen-sink  $N$ -body codes
  1. Starlab example
  2. NBODY6[tt] example