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)

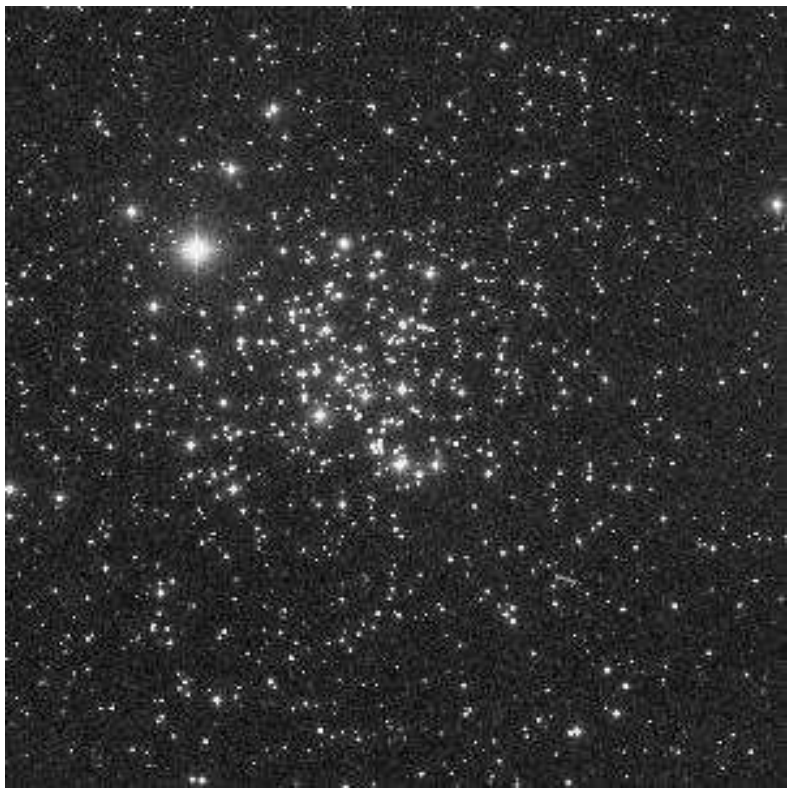
1. The *N*-body problem and its astrophysical settings. Initial conditions. Units. Introduction to a simple *N*-body code (NBODY1). Example - cold collapse. Virialisation and virial equilibrium.
2. Plummer's model. Movie of orbital motions. Crossing time scale. Core collapse - the movie. The relaxation 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).

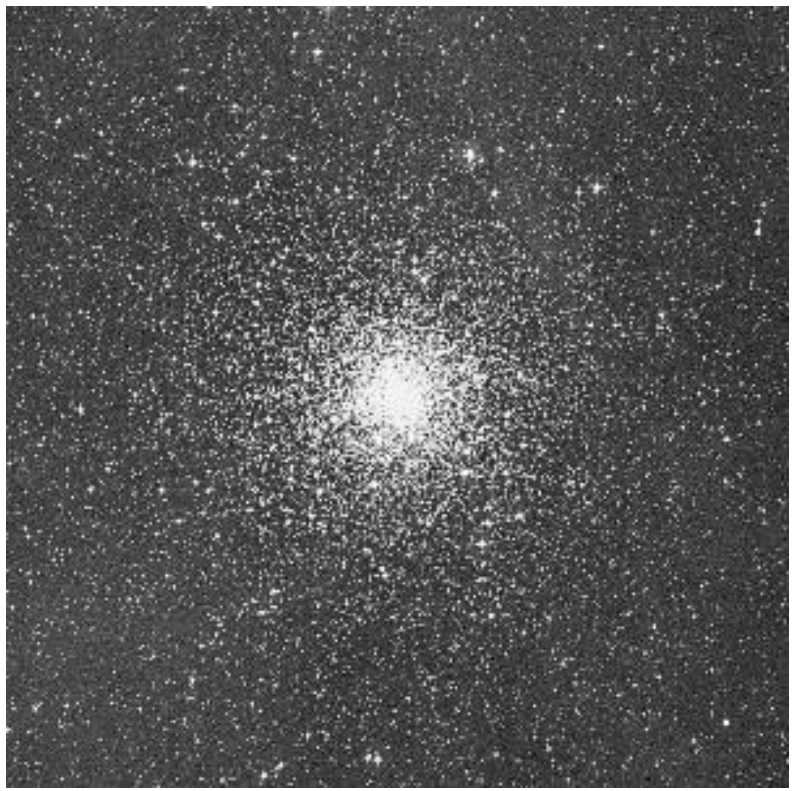
References

- 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

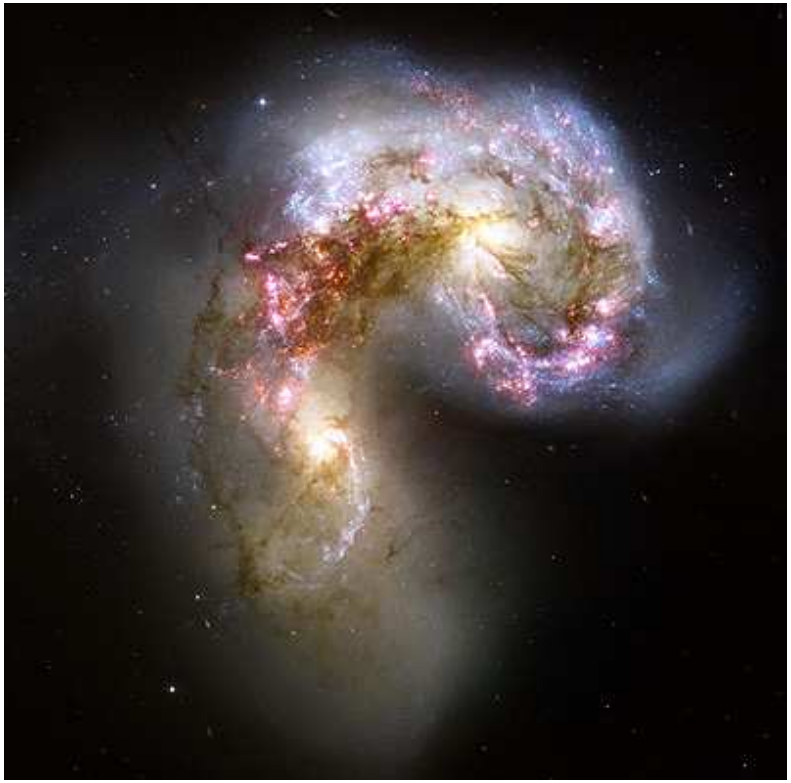
- the Galactic Centre (see http://www.astro.ucla.edu/~ghezgroup/gc/pictures/orbitsMovie.shtml)
- open clusters (M67 here)
- globular clusters (M4 here)





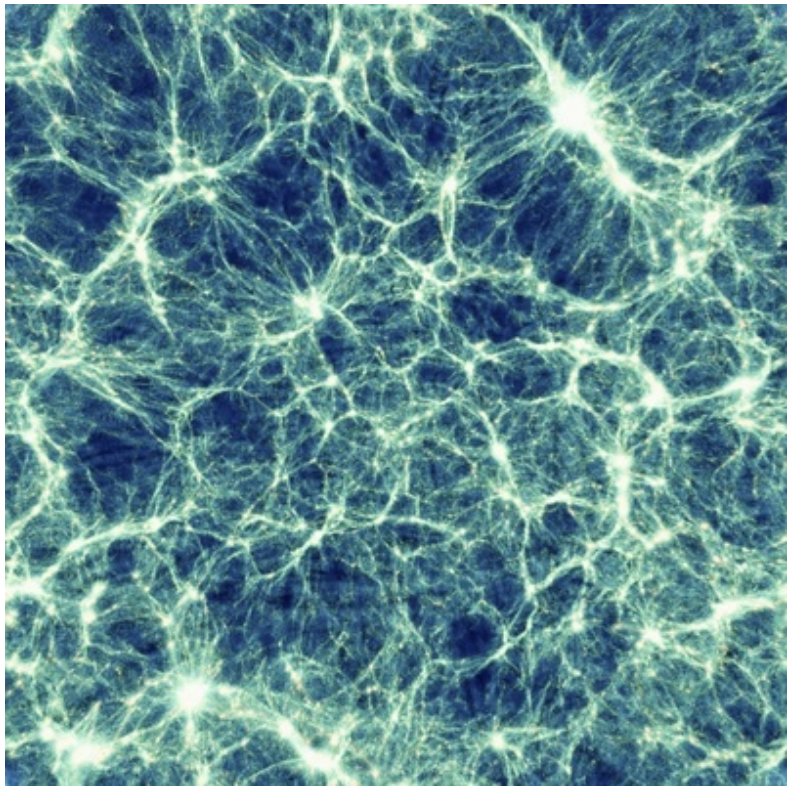
Applications of *N*-Body Schemes in Astrophysics (continued)

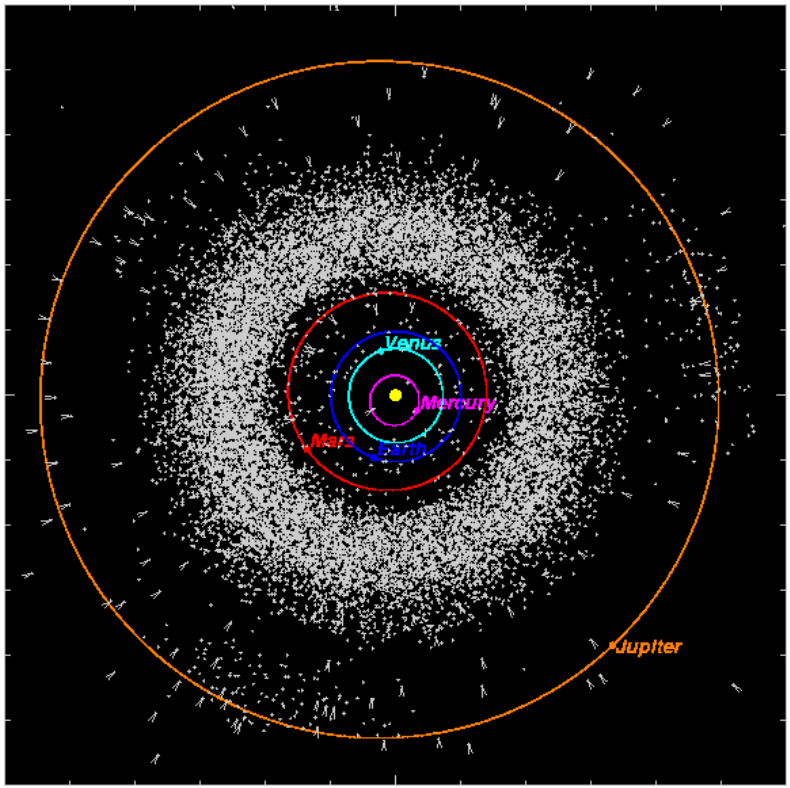
- young dense clusters - Magellanic Cloud clusters
- galaxy dynamics - the Antennae (requires specialised software because of large *N*)



Applications of *N*-Body Schemes in Astrophysics (continued)

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





The *N*-body Problem

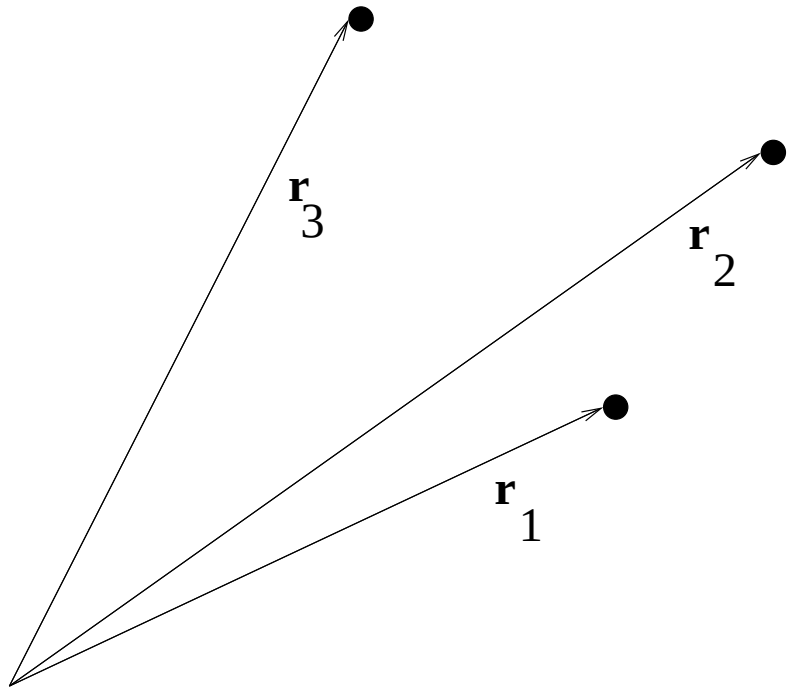
The *N*-body Problem

- *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 $\dfrac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}$, and has magnitude $\dfrac{Gm_1 m_2}{|\mathbf{r}_2 - \mathbf{r}_1|^2}$.

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

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

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 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 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  =  0.00  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* = 3.20E-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 overal 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 equilbrium”, 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

$$\begin{aligned} T &= \frac{1}{2}\sum_{i=1}^{N} m_i \mathbf{v}_i^2 \quad \text{(Kinetic Energy)} \\ V &= -\frac{G}{2}\sum_{i=1}^{N}\sum_{j=1,\neq i}^{N} \frac{m_j m_i}{|\mathbf{r}_i - \mathbf{r}_j|} \quad \text{(Potential Energy)} \\ E &= T + V \quad \text{(Total Energy)} \\ I &= \sum_{i=1}^{N} m_i |\mathbf{r}_i|^2 \quad \text{("Moment of Inertia").} \end{aligned}$$

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

$$\begin{aligned} T + E \simeq 0 \Rightarrow T &\simeq -E \\ \text{and similarly} \quad V &\simeq 2E. \end{aligned}$$

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

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

Mass, length and time scales

- Total mass

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

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

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

- 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

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

- Characterise speeds by (mass weighted) mean square speed

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

- Define time scale

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

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

$$\begin{aligned} G &= 1 \\ M &= 1 \\ R &= 1 \end{aligned}$$

These are often used in *N*-body simulations.

Example

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

Suppose a star cluster has $M = 10^5 M_\odot$, $R = 5$pc. To convert a velocity from the *N*-body code to km/s, multiply by $\sqrt{\dfrac{GM}{R}}$, 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

$$\begin{aligned} G &= 1 \\ M &= 1 \\ R &= 1 \end{aligned}$$

In these units:

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

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 (MIN(5**KZ9,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*RTIDE; RTIDE = 10*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**

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

```
oo.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* - 3.20E-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 <M> = 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. <M> 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

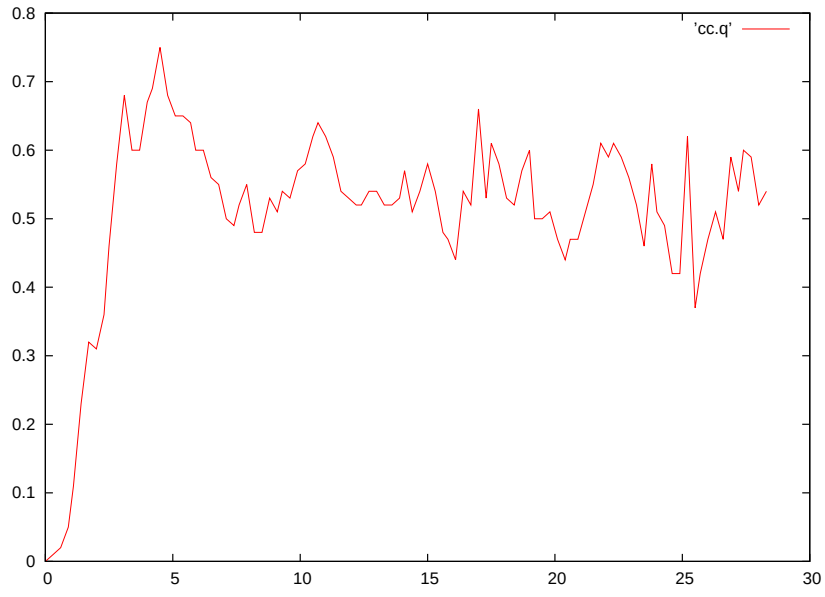
awk -f q.awk cc.out > cc.q

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"

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

Hence

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

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

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

(swapping i and j).

Proof of the Virial Theorem (Contd)

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

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

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.