Rainer Spurzem, spurzem@ari.uni-heidelberg.de

NBODY Lecture 2023

Mon-Thu, March 20-23, 2023

This lecture is NOT an official lecture of Univ. of Heidelberg; if you wish, however, you can get a certificate signed by me, after successful homework done. But this is not required.

1. zoom and chat access, general informations

https://eu02web.zoom.us/j/66152540704?pwd=ZlZHVnUwM1FTV0VBQnpHUWs1eGFudz09

Meeting ID: 661 5254 0704
Passcode: 773483

Schedule (changes and adjustments may occur anytime):

Monday/Tuesday/Wednesday: March 20/21/22, 10:00 - 17:00 (10:00 a.m. - 5:00 p.m.) (proposed lunch break 13-14, and short breaks of 15 mins. in morning and afternoon)

Thursday: March 23: 10:00 - 13:00 (10:00 a.m. - 1:00 p.m.) ; an afternoon session in case of need to discuss any open questions can be offered.

2. Getting access to the kepler computer

1. Every participant has to send me a public key (some have done already). You will all get an account on our kepler system with userid lecturenn where nn is an individual number which I will give to you (email to you). Please be patient, this will take some time on Monday afternoon.

2. It is recommended to work in a group of two persons (one also permitted). You can work on one account together if you wish. Or get two accounts, as you prefer.

3. Create an ssh key Linux/Unix Users (also possible for Windows/Mac):

```
ssh-keygen -t rsa       (passphrase should NOT be empty)
```

It produces a private key id_rsa and a public key id_rsa.pub

Send the public key by email to spurzem@ari.uni-heidelberg.de
Please send it as a file attachment, otherwise I cannot process it. The public key should be a one line file starting with ssh-rsa …

After this has been installed on kepler, you can login with

```
ssh lecturenn@kepler2.zah.uni-heidelberg.de
```

(lecturenn is your account on the system, such as lecture01, lecture02, …)

If you give a non-standard name to your private key, you need to do this:

```
ssh -i ‘full_path_to_private_key’ lecturenn@kepler2.zah.uni-heidelberg.de
```

If it is e.g. in *home/data/xyz/.ssh/my_id.rsa* you need to use inside the ‘..’ above.

For Windows Users:

- you can use a terminal window, which works like a Linux command window, compare https://www.wikihow.com/Use-SSH (check especially ‘create encrypted keys’ and the chmod command, that may cause a problem)
- Second Way: Use the putty client program for ssh; https://putty.org/ (here you need to make sure to copy and paste the openssh public key to send by email!).
- **Third Way – open a Linux Shell on Windows or use a WLS installation (ubuntu on Windows) – then you can follow the Linux informations above.**

3. After login to the kepler computer

* after login to kepler2, do another ssh in the terminal window to wn12: ssh wn12

It should work without password. We need this because of some problem on the main node kepler2; for compiling the code kepler2 has a problem, we need to go to one of the worker nodes. Anyone will be ok, let's use wn12.

* You should see a prompt like [yourname@wn12 ~]$; make sure you are in your home directory (use pwd command). Note that the files and directories are the same on kepler2 and wn12, same data. Create an Nbody subdirectory; and go there:

```
mkdir Nbody ; cd Nbody ; pwd
```

* copy the tar package from my this place to your home directory:

```
cp -p -i /work/Tit7/spurzem/Nbody/Nbody6PPGPU.tgz .
```

Notice: https://github.com/nbody6ppgpu is the github repository, where you find the public code. Here I have prepared a tarball for you, a variant a little different from the public version - it experiments with a new input file structure (you are doing a beta test…).

**Tarball can also be downloaded from:
https://wwwstaff.ari.uni-heidelberg.de/spurzem/nbody-lecture2023/Nbody6PPGPU.tgz**

```
ls -lart
```

```
tar xvfz Nbody6PPGPU.tgz
```

* You should see a new subdirectory Nbody6PPGPU ; just go there with

```
cd Nbody6PPGPU
```

now you should see the source code in ./src/Main/ (use

```
ls -lart ./src/Main/
```

to see); and some files in the current directory, like Makefile and configure.

* Before proceeding you need to double check:

Your file /home/Tit4/lecturenn/.bashrc (nn is your account number) should contain at the end two lines:

```
module load cuda/7.5 (this loads the nvcc cuda c compiler for GPU computing)
ulimit -s unlimited (this allows for large memory allocation)
```

This has been done for you already by the system.

* make sure you are in /home/Tit4/lecturenn/Nbody6PPGPU (check with pwd)

Now enter the command:

```
./configure --with-par=b1m --enable-simd=sse --enable-mcmodel=large
```

(This enables simulations with up to one million particles and many binaries, like up to 10%; you can use actually smaller particle numbers, which are selected in the input file, see below).

In order to use HDF5 file output we have to change manually two lines in build/Makefile - because the configure script does not work well for hdf5 on kepler:

```
HDF5_FLAGS = -D H5OUTPUT -I/usr/include/openmpi-x86_64/ -L/usr/lib64/openmpi/lib/ -lhdf5_fortran
FFLAGS = .... ${HDF5_FLAGS}
```

After it is finished:

```
make clean ; make -j
```

* If all of this is successful, you should find in ./build/nbody6++.sse.gpu.mpi an executable file - it is the file to run the code.

See with

```
ls -lart ./build/
```

The Nbody manual can be found here:
https://www.overleaf.com/read/hcmxcyffjkzq

4. Prepare a run

.... and here are the next few steps, everything can be done on kepler2 now, no need to do ssh to wn12. **IMPORTANT: Please be considerate! In course experiment: takes ~1.5 hours – only one run per account! Expect Waiting Time! Homework needs ~8 hours; please do not start during lecture!! Also homework: only one run at a time for each account!**

```
cd Nbody6PPGPU
```

Copy the executable file with new name to the run directory 100k_test:

```
cp -p -i build/nbody6++.sse.gpu.mpi 100k_test/nbody6++.sse.gpu.mpi.b1m
```

Now, if you go to 100k_test you should have three files: an N-body input file (N100k.inp), a batch job script (nb6++gpu.sbatch.KEPLER), and the executable file, which you produced in the last step. I recommend to give it the name nbody6++.sse.gpu.mpi.b1m , because this is the file name used in the batch job script. If you choose another name, do not forget to change it in nb6++gpu.sbatch.KEPLER .

With these data in 100k_test you are able to start a test run, with 100k single particles (no initial binaries); it will run for 100 time units and produce a lot of output files. How to start it? You should be inside 100k_test; then you enter

```
sbatch < nb6++gpu.sbatch.KEPLER
```

With squeue you should see a message that your job is running. With ls -lart you will see many output files, one of them is an output file like N100k.somenumber.out . This gives informations about the progress of the run. It may take a few hours. **Warning: there will be queues and waiting time. Only one run per account per time!**

Transfer data files to your own computer:

```
scp  lecturenn@kepler2.zah.uni-heidelberg.de:[full path]/filename .
```

5. The Input File

You are doing a beta test here, new input file structure! Not yet in github!!
Nbody6PPGPU/100k_test/N100k.inp:

```
&INNBODY6
KSTART=1,TCOMP=1.E8,TCRTP0=1000.0,isernb=40,iserreg=40,iserks=0 /

&ININPUT
N=100000,NFIX=1,NCRIT=10,NRAND=43532,NNBOPT=80,NRUN=1,NCOMM=10,
ETAI=0.01,ETAR=0.01,RS0=0.15,DTADJ=5.0,DELTAT=5.0,TCRIT=10.0,QE=1.0,RBAR=1.0,ZMBAR=
0.7,
KZ(1:10)= 1 1 1 0 1 0 4 0 0 2
KZ(11:20)=0 1 0 2 2 0 0 0 3 6
KZ(21:30)=1 0 2 0 2 2 3 2 0 2
KZ(31:40)=1 0 2 2 1 0 1 1 2 1
KZ(41:50)=0 0 0 0 0 1 3 0 1 0 ,
DTMIN=2.5E-6,RMIN=8.0E-05,ETAU=0.1,ECLOSE=1.0,GMIN=1.0E-06,GMAX=0.01,SMAX=1.0 /

&INDATA
ALPHAS=2.35,BODY1=150.0,BODYN=0.08,NBIN0=0,NHI0=0,ZMET=0.001,EPOCH0=0.0,DTPLOT=5.0
/

&INSCALE
Q=0.5,VXROT=0.0,VZROT=0.0,RTIDE=0.0 /

&INXTRNL0
GMG=1.78E11,RG0=13.3 /
```

```
&INBINPOP
SEMI0=0.0005, ECC0=-1.0, RATIO=1.0, RANGE=5.0, NSKIP=5, IDORM=0 /
```

==

The meaning of input variables is discussed in our lecture, and also in the Nbody manual. This is control input only – no particle data – they are created by the code itself (or they can be read in from other files if requested).

6. Looking at the output

1) Regularizations - whenever two particles come close to each other, such that their timestep becomes too small for the big N-body system, they are regularized. More informations in my lecture later in September. Practically it means that the two bodies disappear from the big N-body system, and only their center-of-mass (c.m.) comes in as a pseudo-particle. The internal motion is followed in a transformed four-dimensional space using quaternions. Perturbations from outside particles are taken into account. The code uses lines like

NEW KSREG.... and END KSREG to signal that a KS pair has been built (KS = Kustaanheimo-Stiefel regularization).

```
 NEW KSREG   TIME[NB] 6.2175781250E+01 NM1,2,S=         8     77471    100008
KW1,2,S=  14   0   0 IPAIR        2 DTAU  2.17E-03 M1,2[NB]  6.13E-04  2.14E-06
R12[NB]  3.21E-07 e,a,eb[NB]=  1.1037E-01  2.8924E-07 -2.27E-03 P[d]=  8.89E-01 H -
1.06E+03 GAMMA  1.65E-22 STEP(ICM)  1.95E-03 NPERT    0 NB(ICM)   81 M1,2[*]
3.59E+01  1.25E-01 RAD1,2,S[*]  1.52E-04  1.44E-01  1.42E+01 RI,VI[NB]=  1.81E-01
7.01E-01
```

TIME: Nbody units; NM1,2,S: Name of 1,2,c.m.; KW1,2,S: stellar type of 1,2,c.m.; DTAU: Step in KS time variable s or tau; M12; R12: masses and separation in N-body units; e,a,eb: eccentricity, semi-major axis, binding energy in N-Body units (note if hyperbolic encounter we have: e>1, a<0, eb>0); P[d]: Period in days; H: energy per mass in N-body units; GAMMA, STEP(ICM), NPERT, NB(ICM): perturbation, step of c.m. in Hermite, M12[*],RAD12*: Masses and Radii in solar units, separation in solar units, RI,VI: position and velocity of system in cluster (N-body units).

Most of these events are hyperbolic fly-by's; sometimes a longer lasting binary is formed. The output lines contain detailed information what kind of particles are in a two-body encounter, and give most useful parameters, I will show and discuss it with you.

If you should some day in the future have a deeper look in the code an important issue of the data structure is related to KS regularization:

Let's say in the beginning we have our particle data in a vector 1...N for N particles (say X(1,I), I = 1,...N; which is the x-coordinate of particle I). If a KS-Pair (binary) is formed, the two members are put to the first two places in the vector; and the coordinates of the c.m. pseudoparticle come at N+1. So, if NPAIRS is the number of KS pairs we have in the vector:

position 1 to 2*NPAIRS: all members of KS binaries;
2*NPAIRS+1 to N: all single particles
N+1 to N+NPAIRS: all c.m. pseudo-particles
We have auxiliary variable IFIRST = 2*NPAIRS+1 and NTOT = N + NPAIRS .

So, you find in the code two kinds of loops:

from 1,N : all particles, no matter whether they are inside a KS binary or not from IFIRST,NTOT: all single particles plus c.m. particles

2)a) Lagrangian Radii

Lagrangian radii are spherical shells, containing a fixed fraction of total mass. The half-mass radius(50% Lagr. radius) is frequently used, but we have currently 18 different valus, defined in lagr.f of mass fraction, ranging from 0.1 % to 100%. The code prints lines like (take these lines out of the big output file, use e.g. unix grep, into a new file for plotting).

Heading Line (fraction of total mass M/MT, last column core radius)

```
   TIME   M/MT:   1.00E-03 3.00E-03 5.00E-03 1.00E-02 3.00E-02 5.00E-02 1.00E-01
2.00E-01 3.00E-01 4.00E-01 5.00E-01 6.00E-01 7.00E-01 8.00E-01 9.00E-01 9.50E-01
9.90E-01 1.00E+00      <RC
```

Lagrangian Radii Lines:

```
0.0000D+00 RLAGR: 6.75E-02 9.38E-02 1.02E-01 1.30E-01 1.90E-01 2.34E-01 3.07E-01
4.29E-01 5.40E-01 6.58E-01 7.82E-01 9.25E-01 1.15E+00 1.46E+00 2.12E+00 2.88E+00
4.72E+00 6.10E+00 3.17E-01
```

at every N-body time (the interval depends on DELTAT input parameter). These lines can be collected in a file (e.g. using linux grep) and used to plot Lagr. Radii as function of time (the first number in the line). Also other quantities of the Lagr. shells are printed, like SIGR, SIGT, VROT and a few more. I want to show it to you in the output file and discuss.

If your N-Body output file is N100k.19491.out you get these lines in a file:

```
grep RLAGR  N100k.19491.out > RLAGR.data
```

RLAGR.data contains all the lines, ready for plotting Lagrangian radii as a function of time (column 1).

2b) Average Mass:

Printed is the average stellar mass in N-body units (use ZMBAR to scale to solar masses) inside the Lagrangian mass shells, as defined in 2a). You find the line in the code:

```
0.0000D+00 AVMASS: …
```

If your N-Body output file is N100k.19491.out you get these lines in a file:

```
grep AVMASS  N100k.19491.out > AVMASS.data
```

AVMASS.data contains all the lines, ready for plotting average masses in Lagrangian mass shells as a function of time (column 1).

3) Stellar Evolution File

The code produces files with names like sev.83_n , where again n is the time (n=0 initial model, n=1 at one N-body time unit and so on). These files contain stellar evolution data for all stars, for example columns 6,7,8,9 are mass in solar masses, and luminosity (in log10 solar units), stellar radius (in log10 solar units), Teff (in log10 Kelvin). These data can be used to plot directly Hertzsprung Russel Diagrams, using x-axis column 9, y-axis column 7. Note: Astronomical habit is to plot the x-axis such that small Teff is left, and large Teff is right.

Note also that column 5 contains the distance of the star from the cluster center, and column 4 the type of star, from which you can read whether we have main sequence star, or a red giant, or black hole or whatever (see our manual).

7. Homework

(if possible by April 23, 2023, if longer time needed let me know)

* Have in your account the N-body run output file, produced by your run, 100k particles, for up to 100 time units (need to change TCRIT from 10 to 100).
* Have a plot of Lagrangian radii as a function of time (it is ok to plot 5 selected Lagrangian radii, not all)
* Have a plot of average masses as a function of time (it is ok to plot 5 selected Lagrangian radii, not all).
* Have two “colour-magnitude” diagrams, one for t=0, another for t=100 (can be in one plot). “Colour-magnitude” is astrophysical observers terminology; for us it means: plot log(luminosity) as a function of log(Teff), with high temperaturs to the left, low temperatures to the right. Plot a dot for every star.
* These runs will take about eight hours, so if every participants does a run it will take a couple of days for all to get through. So, patient waiting is required...