NEMO: a case study for AMSC 664

Peter Teuben
Astronomy Department
teuben@astro.umd.edu

URL: http://www.astro.umd.edu/nemo

URL: http://www.manybody.org
Collional and Collisionless Dynamics

STARLAB

NEMO
NEMO

• Observational Astronomy has many software packages (AIPS, IRAF, Gipsy, Figaro, ...)
  – Each telescope has specific calibration needs
  – Image processing is nearly always the same
  – A data interchange standard (FITS) emerged
  – The wheel was re-invented many times
  – Group Effort

• Theoretical Astronomy has not! (1986)
  – Individual Effort (still)
  – but: Virtual Observatory (2000 decadal survey)
Design

• *N-body integrator(s) with* many small tools, each performing a small well defined task
  – ? modern approach → python-like scripting ?
  – NEMO vs. tipsy approach

• Easy to use

• Easy to extend
  – Add your own code
  – Add foreign code
Design (cont'd)

- Uniform (command line) user interface
  - Good help facilities
  - Graphics vs. Command Line
- Portable binary (hierarchical) dataformat
  - endianism, floating point accuracy
  - Unix-like use of pipes
- Graphics: YAPP
- Dynamic function use (.so, .dll)
User Interface

- main(argc, argv) → nemo_main(void)
  - nemomain.c defines main()

- User interface:
  char *defv[] = {
    "out=???
 input file",
    "nbody=100
 particles",
    "VERSION=1.0
 9-apr-2004 PJT",
    NULL
  };

- Program vs. System keywords
User Interface (cont'd)

- System keywords
  - help=
    - Internal help vs. external (man pages, html)
  - debug=
    - dprintf(2,"N=%d Level=d Radius=%g\n",n,l,r);
  - error=
    - error("%d too large (MAXFOO=%d)",n,MAXFOO);
  - yapp=
    - Value depends on the library used at installation
User Interface (cont'd)

• Help
  – Internal help
    • (help=) comes with every NEMO executable
  – External help:
    • Standard unix man pages (and html formatted)
      – Man, tkman, xman, gman
    • Users and Programmers Guide
    • FAQ
File Format

• Binary Structured Files
  – Sequence of tagged items
    • Tag: name, type, dimension
  – Hierarchical
  – Always written in native endianism
  – Portable (detect endianism)
  – Transparently detect pipes (fname=-)

• User tools: tsf, rsf, csf, qsf
NEMO file formats
Graphics: YAPP

• Yet Another Plotting Package
  – Define a simple API that can be implemented by a number of popular graphics packages
    • pgplot (Caltech Astronomy)
    • plplot (sourceforge)
    • Mongo ($$$)
    • SM ($$$)
    • PS (nemo)
    • OpenGL
    • Null (nemo)
Dynamics Functions

- Interface to an efficient way to use dynamics functions (now implemented via dlopen(3))
  - **Snapshots**: bodytrans variables (e.g. xvar=x/z or evar='m/sqrt(x*x+y*y)'
  - **Orbits**: potential functions, so tools do not have to be recompiled for new potential. Uniform interface using potname=, potpars=, potfile= (also used in some Nbody integrators now)
  - **Tables**: fitting functions, only used in non-linear least squared fitting program (tabnllsqfit)
Building NEMO

- Autoconf + hierarchical makefile's
  - Single library (libnemo.a)
  - Lots of optional Alien packages in NEMOLIB
    - HDF, cfitsio, pgplot, gsl, vogl,
- Testfile's for regression testing
  - Not hierarchical, a script hunts for them and runs “make -f Testfile all”
  - Output can be compared to archived version
- NEMODAT contains
  - standard datasets
  - Benchmark data
NEMO

- A toolkit of libraries and tools (programs)
- Scripts provide the glue to do simulations and analysis
- Portable structured (binary) files (snapshot, orbit, image, table)
- Initial work by Barnes, Hut & Teuben (1986) [Teuben 1995]
- SRC: source: 193 KLOC, man: 33 KLOC files: 936
- USR: source: 860 KLOC, files: 4141
- Unix makefiles, autoconf, CVS
- Mostly C, and some C++ and Fortran
- Many user contributions
- Wishlist....

http://www.astro.umd.edu/nemo

Info and Download
## NEMO: some public codes

- Nbody* (Aarseth) [usr]
- Ptreecode (Dubinski)
- PMCode (Klypkin)
- Gadget (Springel)
- AP3M/hydra (Couchman)
- Galaxy (Sellwood) [usr]
- Treecode (Hernquist) [usr]
- Treecode1 (Barnes) [usr]
- Tree++ (Makino) [usr]
- Vtc (Kawaii) [usr]
- Scfm (Hernquist) [usr]
- Multicode (Barnes) [usr]
- Flowcode (Teuben) [usr]
- Superbox (Richardson)
- YANC (Dehnen) [usr]
- gyrfalcON
Evolved exponential disk, rotated and inclined velocity field

```
NEMO example

mkexpdisk - 20000 rcut=2 | hackcode1 - disk4.out tstop=4
snaprotate disk4.out - 60,45 xz | \n  snapplot - times=2                                             (left panel)
  snaprotate disk4.out - 60,45 xz | \n  snapgrid - - zvar=vz moment=-1 times=2 | \n  ccdplot - contour=-1:1:0.2 blankval=0              (right panel)
```
Optimal N-body softening:
Seed=1,2,3,4

Dehnen & Teuben, 2004

IAU 208 : 12 July 2001, Tokyo
# ! /bin/csh -f
#

mkexpdisk out=$run.ini nbody=$nbody Qtoomre=$Qtoomre seed=$seed rcut=$rcut tab=t \ 
   headline="$*" time=0 > $run.tab

YancNemo in=$run.ini out=$run.snp \ 
   eps=$eps theta=$theta kernel=$kernel \ 
   tstop=$tstop step=$step hmin=$hmin give_pot=1 give_rhoe=1 > $run.yanc

set times=0:$tstop:$step
set weight="-phi*phi*phi"

# loop over all times requested
rm -f $run.psi
foreach t (`nemoinp $times`)  
   rm -f $run.snp.tmp
# extract time & sort bodies by potential
   snaptrim $run.snp - times=t \ 
      snap_sort -- phi \ 
      snapmask $run.snp.tmp 0:$nfrac
# align & get phase angles
   snaprect $run.snp.tmp . weight="$weight" > $run.tmp1
   set ex=('grep e_x $run.tmp1 | awk -F: '{print $2}"
   if ($#ex != 6) continue
# also obtain axis ration of moments of inertia
   snapinert $run.snp.tmp - weight="$weight" tab=t > $run.tmp2
   set si=('cat $run.tmp2')
# output: time psi Ixx Iyy Izz
end
Optimal N-body softening:
Seed=1,2,3,4

Dehnen & Teuben, 2004

Time Phi(t) b/a Phi(t)-model

Phases and orbital elements of a few selected objects are presented. The evolution of the system is shown in the time series of \( \Phi(t) \) and the ratio \( b/a \). The \( \Phi(t) \)-model shows the evolution of the system over time.
Optimal N-body softening
GRAPE-6 and baby-GRAPE-6

Tflops and Tbytes
Hayden Planetarium
Setup in the Hayden Planetarium
Dark time in the Dome
SpaceOrb motion control
Galaxy Modeling

➔ GIPSY, AIPS/AIPS++, NEMO, karma