N-Body Methods and Algorithms

Sverre Aarseth

Institute of Astronomy, Cambridge

Equations of Motion

Hermite Method

Basic Hermite Code

Neighbour Scheme

NBODY2H

Programming Exercises

http://www.ast.cam.ac.uk/~sverre

http://www.sverre.com

Newton's Equations

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

Explicit differentiation

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

New solution at $t = \Delta t$

$$\Delta \dot{\mathbf{r}}_i = (\frac{1}{2}\mathbf{F}_i^{(1)} \Delta t + \mathbf{F}_i) \Delta t$$

$$\Delta \mathbf{r}_i = ((\frac{1}{6}\mathbf{F}_i^{(1)} \Delta t + \frac{1}{2}\mathbf{F}_i) \Delta t + \dot{\mathbf{r}}_i) \Delta t$$

Softening

Potential $\Phi(r) = \dfrac{Gm}{(r^2 + \epsilon^2)^{1/2}}$

Energy $E = \frac{1}{2}v^2 - \Phi(r) = \text{const}$

Force $\mathbf{F}_i = -G \sum_{j=1;\, j\neq i}^{N} m_j \dfrac{\mathbf{r}_i - \mathbf{r}_j}{(|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2)^{3/2}}$

Force derivative

$$\mathbf{F}_i^{(1)} = -G \sum_{j=1;\, j\neq i}^{N} m_j \frac{\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_j}{(|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2)^{3/2}} - 3m_j \frac{(\mathbf{r}_i - \mathbf{r}_j)\cdot(\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2} \frac{\mathbf{r}_i - \mathbf{r}_j}{(|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2)^{3/2}}$$

Hermite Integration

Taylor series for $\mathbf{F}$ and $\mathbf{F}^{(1)}$

$$\mathbf{F} = \mathbf{F}_0 + \mathbf{F}_0^{(1)}\, t + \frac{1}{2}\mathbf{F}_0^{(2)}\, t^2 + \frac{1}{6}\mathbf{F}_0^{(3)}\, t^3$$

$$\mathbf{F}^{(1)} = \mathbf{F}_0^{(1)} + \mathbf{F}_0^{(2)}\, t + \frac{1}{2}\mathbf{F}_0^{(3)}\, t^2$$

Prediction

$$\mathbf{r}_j = ((\frac{1}{6}\mathbf{F}_0^{(1)}\, \delta t'_j + \frac{1}{2}\mathbf{F}_0)\, \delta t'_j + \mathbf{v}_0)\, \delta t'_j + \mathbf{r}_0$$

$$\mathbf{v}_j = ((\frac{1}{2}\mathbf{F}_0^{(1)}\, \delta t'_j + \mathbf{F}_0)\, \delta t'_j + \mathbf{v}_0 \,; \qquad \delta t'_j = t - t_0$$

New forces $\qquad \mathbf{F},\ \mathbf{F}^{(1)}$

Higher derivatives

$$\mathbf{F}_0^{(3)} = (2(\mathbf{F}_0 - \mathbf{F}) + (\mathbf{F}_0^{(1)} + \mathbf{F}^{(1)})\, t)\, \frac{6}{t^3}$$

$$\mathbf{F}_0^{(2)} = (-3(\mathbf{F}_0 - \mathbf{F}) - (2\mathbf{F}_0^{(1)} + \mathbf{F}^{(1)})\, t)\, \frac{2}{t^2}$$

Corrector for i

$$\Delta \mathbf{r}_i = \frac{1}{24}\mathbf{F}_0^{(2)}\Delta t^4 + \frac{1}{120}\mathbf{F}_0^{(3)}\,\Delta t^5$$

$$\Delta \mathbf{v}_i = \frac{1}{6}\mathbf{F}_0^{(2)}\Delta t^3 + \frac{1}{24}\mathbf{F}_0^{(3)}\,\Delta t^4$$

Basic Code Structure

Input	Read input parameters
Initial conditions	Generate m, $\mathbf{r}$, $\dot{\mathbf{r}}$
Initialization	$\mathbf{F}$, $\mathbf{F}^{(1)}$ & Δt
Scheduling	Form block-step distribution
Prediction	Neighbours or all N
Neighbour integration	Sequential $\mathbf{F}_n \& \mathbf{F}_n^{(1)}$
Regular force calc	Sequential total force
New block-step	Determine next group

Time-Steps

Basic time-step $\quad \Delta t = \frac{\alpha|\mathbf{r}|}{|\mathbf{v}|}, \quad \Delta t = \frac{\beta|\mathbf{F}|}{|\mathbf{F}^{(1)}|}$

Taylor series $\quad \mathbf{F} = \mathbf{F}_0 + \mathbf{F}_0^{(1)}\Delta t + \frac{1}{2}\mathbf{F}_0^{(2)}\Delta t^2 + \ldots$

Natural time-step $\quad \Delta t = \left(\frac{\eta|\mathbf{F}|}{|\mathbf{F}^{(2)}|}\right)^{1/2}, \quad \eta = 0.02$

General expression $\quad \Delta t = \left(\frac{\eta(|\mathbf{F}||\mathbf{F}^{(2)}| + |\mathbf{F}^{(1)}|^2)}{|\mathbf{F}^{(1)}||\mathbf{F}^{(3)}| + |\mathbf{F}^{(2)}|^2}\right)^{1/2}$

Relative criterion $\quad \Delta t$ independent of mass

Block-steps $\quad \Delta t_n = \frac{\Delta t_1}{2^{n-1}}, \quad \Delta t_1 = 1$

Hierarchical levels $\quad \mathcal{N}_k$ particles with steps Δt_k

Scheduling $\quad i = \min\,(t_j + \Delta t_j)$

Units

(a) Scaling relations

Given length scale $R_{\rm V}$ in pc and total mass $NM_{\rm S}$ in $M_\odot$

Velocity scaling

$\tilde{V}^* = 1 \times 10^{-5} (GM_\odot / L^*)^{1/2}$ km/s, with $L^* = 3 \times 10^{18}$ cm

Velocity unit $\quad V^* = 6.557 \times 10^{-2} (NM_{\rm S}/R_{\rm V})^{1/2}$ km/s

Fiducial time $\quad \tilde{T}^* = (L^{*3}/GM_\odot)^{1/2} = 14.94$ Myr

Time unit $\quad T^* = 14.94\,(R_{\rm V}^3/NM_{\rm S})^{1/2}$ Myr

(b) Conversion from N-body to physical units

$\tilde{r} = R_{\rm V}\, r$ pc, $\quad \tilde{v} = V^*\, v\ km/s, \quad \tilde{t} = T^*\, t$ Myr,

$\tilde{m} = NM_{\rm S}\, m\ M_\odot$

Crossing time $\quad T_{\rm cr} = 2\sqrt{2}\, T^*$ Myr

Scaling of Initial Conditions

Main input	$N,\ N_{\rm b},\ M_{\rm S},\ R_{\rm pc}$
Cluster parameters	optional IMF and Plummer or King model
Initial data	$\tilde{m}_i,\ \tilde{\mathbf{r}}_i,\ \tilde{\mathbf{v}}_i, ...,\ i = 1,\ N$
Total energy	$E = T - U$
Virial theorem	$\mathbf{v}_i = q\,\tilde{\mathbf{v}}_i,\quad q = \left[\frac{Q_{\rm V} U}{T}\right]^{1/2},\quad \mathbf{r}_i = \tilde{\mathbf{r}}_i$
Standard units	$G = 1,\ \ \sum m_i = 1,\ \ E_0 = -0.25$
Standard scaling	$\hat{\mathbf{r}}_i = \frac{\mathbf{r}_i}{S^{1/2}},\ \hat{\mathbf{v}}_i = \mathbf{v}_i\, S^{1/2},\ S = \frac{E_0}{q^2 T - U}$
Astrophysical units	$V^*,\ T^*,\ R^*$ from $M_{\rm tot},\ R_{\rm pc}$
Primordial binaries	split or copy m_i, introduce $a,\ e,\ \Omega$
Force polynomials	$\mathbf{F}_i,\ \dot{\mathbf{F}}_i,\ \Delta t_i, ...,\ i = 1,\ N$
KS regularization	explicit initialization, $R < R_{\rm cl}$

Neighbour Scheme

Total force $\mathbf{F}(t) = \sum_{j=1}^{n} \mathbf{F}_j + \mathbf{F}_d(t)$

Prediction

$$\mathbf{F}(t) = \mathbf{F}_n + \dot{\mathbf{F}}_d(t - t_0) + \mathbf{F}_d(t_0)$$

$$\dot{\mathbf{F}} = \dot{\mathbf{F}}_n + \dot{\mathbf{F}}_d$$

Time-scales

$$\Delta t_n << \Delta t_d, \qquad n << N$$

Neighbour sphere $R_s^{\text{new}} = R_s^{\text{old}} \left(\frac{n_p}{n}\right)^{1/3}$

Neighbour selection $|\mathbf{r}_i - \mathbf{r}_j| < R_s$

Derivative corrections $\mathbf{F}_{ij}^{(2)}, \mathbf{F}_{ij}^{(3)}$

Modification of COMMON

(a) Constant size

Existing dummies	$..., XDUM(10), NDUM(10)$
New variables	$XNEW(2), NEW$
	$..., XNEW(2), XDUM(8), NEW, NDUM(9)$

(b) Enlargement

Increase COMMON	$COMMON/EXTRA/\ \ A(5), B, NEW(6)$
Add to MYDUMP	$REAL*4\ \ XNEW$
New COMMON	$COMMON/EXTRA/\ \ XNEW(18)$
Add READ/WRITE	$..., XNEW$

```
*       common6.
*       -------
*
      INCLUDE 'params.h'
      IMPLICIT REAL*8  (A-H,O-Z)
      REAL*8  MP,MP0,MPDOT
*
      COMMON/NAMES/  NTOT,NPAIRS,NTTOT,N,NNBMAX,NCRIT,NFIX,NMERGE,NSUB,
     &               IPHASE,IFIRST,ICOMP,JCOMP,ICLOSE,JCLOSE,JCMAX,
     &               KSPAIR,NRUN,MODEL,NC,NZERO,NBZERO,NBIN0,NHI0,
     &               NAME0,NCH,NCHAOS,IDUM1,KZ(50),NNBOPT,NEW2(8)

      COMMON/PARAMS/ ETAI,ETAR,DTADJ,DELTAT,TCRIT,QE,RBAR,ZMBAR,
     &               DTMIN,RMIN,ETAU,CMSEP2,ECLOSE,GMIN,GMAX,ETA0,
     &               TWOPI,ONE3,ONE6,ONE9,ONE12,TCR0,TRH,BODYM,BODY1,
     &               SMIN,RMIN2,RMIN22,STEPJ,ALPHA,ZNBMIN,ZNBMAX,EBH,
     &               TIME,TADJ,TNEXT,CPU,CPU0,CPUTOT,ZMASS,RSCALE,TCR,
     &               TRC,BE(3),CMR(4),CMRDOT(4),ZKIN,POT,EBIN,EBIN0,
     &               ESUB,EMERGE,ECOLL,EDISS,ESYNC,E(12),ERROR,ERRTOT,
     &               DETOT,ETCORR,AZ,PCRIT,EBCH0,RTIDE,TSCALE,TIDAL(4),
     &               HT,ETIDE,EGRAV,RSFAC,RSPH2,RC,RC2,RC2IN,VC,ZMC,
     &               RDENS(3),RHOD,RHOM,RSMIN,RMAX,DMIN1,DMIN2,DMIN3,
     &               DMIN4,DMINC,SBCOLL,BBCOLL,CHCOLL,DELTAS,ORBITS(9),
     &               GPRINT(9),TLASTT,TLASTS,TLASTB(9),TDUMP,
     &               SCOEFF(12),TOFF,TTOT,EBESC,EMESC,ESESC,CLIGHT,
     &               RZ,TINY,SMAX,WTOT,WTOT0,VRMS,DUMMY(97)

      COMMON/COUNTS/ NSTEPI,NSTEPR,NSTEPU,NNPRED,NBCORR,NBFULL,NBVOID,
     &               NNTB,NBSMIN,NLSMIN,NBDIS,NBDIS2,NCMDER,NBDER,
     &               NFAST,NBFAST,NBLOCK,NBPRED,NICONV,NCHAIN,NSTEPC,
     &               NKSTRY,NKSREG,NKSHYP,NKSPER,NPRECT,NEWKS,NKSMOD,
     &               NTTRY,NTRIP,NQUAD,NMERG,NSTEPT,NSTEPQ,NDISS,NTIDE,
     &               NCOLL,NSYNC,NSESC,NBESC,NMESC,NTIMER,NSTEPS,NPRINT,
     &               NDUMP,NBPREV,NEWHI,NSTEPB,NBFLUX,NMTRY,NWARN,
     &               NIRECT,NURECT,NBRECT,NRRECT,KSMAG,NOFL(2),NPOP(10),
     &               NBLCKR,NDUMMY(99)

      COMMON/PLPOT/  MP,AP2,VIR,MP0,MPDOT,TDELAY,RTIDE0,QVIR,PLDUM(4)

      COMMON/BLOCKS/ TPREV,TBLOCK,DTK(40),KVEC(2*KMAX)

      COMMON/STARS/  EPOCH0,ZMRG,ZMHE,ZMRS,ZMWD,ZMSN,ZMNH,ZMBH,ZMDOT,
     &               AU,PC,GM,DAYS,YRS,SU,SMU,RAU,TSTAR,VSTAR,STEPX,
     &               TMDOT,TPHYS,TURN,EMDOT,ECDOT,EKICK,TPLOT,DTPLOT,
     &               XHYD,YHEL,ZMET,ZPARS(20),SPNFAC,IQCOLL,NAS,NBH,
     &               NBKICK,NBR,NBRK,NBS,NCHA,NCIRC,NCOAL,NCONT,NDD,
     &               NEMOD,NGB,NGLOB,NGLOB0,NHE,NHG,NHI,NHYP,NKICK,
     &               NMDOT,NMS,NNH,NRG,NRO,NROCHE,NRS,NRSAVE,NSHOCK,
     &               NSLP,NSN,NSP,NSPIR,INSTAB,NTZ,NWD,NCE,NHYPC,NBH0,
     &               ITYPE(5),KSAVE(4),KTYPE(0:14,0:14),NEINT,IBLUE,
     &               ITAIL0,NTAIL,NTDUM,NSTAIL,N1,NGDUM(7),
     &               LISTR(MLR),LISTD(MLD),LISTV(MLV)
```

```
      COMMON/NBODY/  X(3,NMAX),X0(3,NMAX),X0DOT(3,NMAX),F(3,NMAX),
     &               FDOT(3,NMAX),BODY(NMAX),RS(NMAX),XDOT(3,NMAX),
     &               FI(3,NMAX),D1(3,NMAX),D2(3,NMAX),D3(3,NMAX),
     &               FR(3,NMAX),D1R(3,NMAX),D2R(3,NMAX),D3R(3,NMAX),
     &               STEP(NMAX),T0(NMAX),STEPR(NMAX),T0R(NMAX),
     &               TNEW(NMAX),RADIUS(NMAX),TEV(NMAX),TEV0(NMAX),
     &               BODY0(NMAX),EPOCH(NMAX),SPIN(NMAX),XSTAR(NMAX),
     &               ZLMSTY(NMAX),FIDOT(3,NMAX),D0(3,NMAX),
     &               FRDOT(3,NMAX),D0R(3,NMAX),KSTAR(NMAX)

*
      COMMON/PAIRS/  U(4,KMAX),U0(4,KMAX),UDOT(4,KMAX),FU(4,KMAX),
     &               FUDOT(4,KMAX),FUDOT2(4,KMAX),FUDOT3(4,KMAX),
     &               H(KMAX),HDOT(KMAX),HDOT2(KMAX),HDOT3(KMAX),
     &               HDOT4(KMAX),DTAU(KMAX),TDOT2(KMAX),TDOT3(KMAX),
     &               R(KMAX),R0(KMAX),GAMMA(KMAX),SF(7,KMAX),H0(KMAX),
     &               FP0(4,KMAX),FD0(4,KMAX),TBLIST,DTB,KBLIST(KMAX),
     &               KSLOW(KMAX),NAME(NMAX),LIST(LMAX,NMAX)
*
      COMMON/LISTS/  ILIST(NMAX),JLIST(NMAX),JPERT(5*LMAX)
```

```
SOURCE = \
nbody6.f adjust.f assess.f bindat.f binev.f binout.f binpop.f \
block.f bodies.f brake.f brake2.f brake3.f bsetid.f chaos0.f chaos.f \
check.f checkl.f chrect.f clint.f cloud.f cloud0.f cmbody.f \
cmcorr.f cmfirr.f cmfreg.f coal.f comenv.f core.f corerd.f \
cputim.f data.f decide.f define.f deform.f degen.f delay.f \
dgcore.f dtchck.f eccmod.f ecirc.f edot.f efac2.f efac3.f \
expel.f energy.f escape.f events.f evolve.f expand.f fclose.f \
fcloud.f fcorr.f fdisk.f fhalo.f ficorr.f findj.f findm.f \
flyby.f fnuc.f fpcorr.f fpert.f fpoly1.f fpoly2.f freeze.f \
gcinit.f gcint.f giant.f giant3.f gntage.f grrad.f hcorr.f \
hiarch.f hicirc.f hidat.f higrow.f himax.f himax2.f himod.f \
hipop.f hirect.f histab.f hivel.f hmdot.f hmdot2.f hotsys.f \
hrdiag.f hrplot.f hut.f hut2.f iblock.f imf.f imfbd.f imf2.f \
impact.f induce.f inext.f input.f insert.f instar.f intgrt.f \
intide.f jacobi.f kepler.f kick.f kick2.f ksapo.f kscorr.f \
ksin2.f ksinit.f ksint.f kslist.f ksmod.f ksperi.f kspert.f \
kspoly.f kspred.f ksrect.f ksreg.f ksres.f ksres2.f ksterm.f \
kstide.f lagr.f lagr2.f levels.f magbrk.f matrix.f mdot.f merge.f \
merge2.f mix.f mloss.f mlwind.f modify.f mrenv.f mtrace.f mydump.f \
nbhist.f nbint.f nblist.f nbpot.f nbrem.f nbrest.f nbsort.f nbtide.f \
newtev.f nstab.f ntint.f offset.f orbit.f output.f peri.f permit.f \
pfac.f poti.f proto_star.f qtides.f ran2.f reflct.f regint.f \
remove.f rename.f reset.f reset2.f resolv.f rkint.f rl.f roche.f \
rpmax.f rpmax2.f rpmin.f scale.f search.f setup.f setup2.f shrink.f \
sort1.f spiral.f stability.f star.f start.f stepk.f steps.f stumpf.f \
subint.f swap.f sweep.f synch.f tail0.f tcirc.f tides.f tides2.f \
tides3.f touch.f tpert.f trdot.f trdot2.f trflow.f tstab.f tstep.f \
units.f unpert.f update.f verify.f xtrnl0.f xtrnld.f xtrnlf.f xtrnlp.f \
xtrnlt.f xtrnlv.f xvpred.f zare.f zcnsts.f zero.f zfuncs.f \
triple.f derqp3.f difsy3.f erel3.f extend.f qpmod3.f stabl3.f \
stablz.f start3.f subsys.f tperi.f trans3.f \
quad.f derqp4.f difsy4.f endreg.f erel4.f ichain.f newreg.f newsys.f \
qpmod4.f rchain.f rsort.f stabl4.f start4.f status.f trans4.f \
cfuncs.f chain.f chstab.f const.f cstab2.f cstab3.f cstab4.f cstab5.f \
derqp.f difsy1.f erel.f hpsort.f inclin.f invert.f ksphys.f physks.f \
qforce.f qpmod.f r2sort.f recoil.f redraw.f select.f slow.f stablc.f \
swcond.f switch.f transk.f transq.f transx.f vector.f xtf.f xtrnlu.f \
ycopy.f ysave.f \
absorb.f chaos2.f chdata.f chfind.f chfirr.f chinit.f chlist.f chmod.f \
chpert.f chpot.f chterm.f expel2.f fchain.f ghost.f giant2.f kcpert.f \
reduce.f reinit.f renew.f setsys.f tchain.f xcpred.f xtpert.f premsf.f \
circ.f spinup.f
```

Programming

NBODY6	330 routines and 56,000 lines
Languages	FORTRAN, C++ & CUDA
Basic variables	70 size N arrays & size N lists
Parameters	$NMAX$ before compilation
Common blocks	common6.h, arrays & scalars
Subroutines	FORTRAN, C++ & GPU
Coding style	6 letter variables, verbal comments
Options	More than 40, multiple values
Downloads	www.ast.cam.ac.uk/research/nbody

Code Tests

Chaotic system	Exponential error growth
Small time-steps	$\Delta t \propto R^{3/2}$
Two-body problem	Analytical solution
Evolution rate	Core collapse time
Mass segregation	Two mass groups
Systematic errors	Minimize energy drift
Performance	CPU time as f(N)