GalactICS - A Galaxy Model Building Package

First do the following:

1. cd src

2. type 'make all' - This builds the programs.

Note that some of the code is in fortran 77, using lines longer than 72 characters in some cases. The -e flag in the makefile allow for this for a Solaris f77 compiler. Other programs are written in C. Again, the linking between these routines works on solaris systems, but may need to be adjusted for other architectures. We have found that linking using f77 instead of ld will often automatically load the appropriate libraries.

The graphics output by some of the programs (dbh, plotforce, diskdf, plothalo) use the PGPLOT library. It can be found at http://astro.caltech.edu/~tjp/pgplot/. Alternatively, remove all calls to routines with names starting with "PG", as well as the -lpgplot flag in the Makefile, and the programs should still run fine.

3. type 'make install' - This copies programs to the directory bin/

Now you're set.

Test that the programs run:

Go into directory Milky_Way/A

Type "make galaxy" - this should build all the files and a small N-body galaxy - look at all of the in.* files to see the various parameters that go into building the models. And read below for more details.

The distribution contains the input files for Milky_Way/A through D, the models that were used in Kuijken & Dubinski 1995.

John & Konrad

This package contains a set of programs and subroutines for building galaxy
models including a disk, a bulge and halo. The details of the inner workings of the code are described in Kuijken and Dubinski 1995.

There are 3 steps in building these models.

A. Calculating the potential.

B. Constructing a disk distribution function which will generate the given potential.

C. And realizing each component with a self-consistent distribution of particle orbits.

These actions are all performed by typing `make galaxy', which runs a succession of programs to end up with a set of N-body particle masses, positions and velocities representing your model.

Descriptions of the individual steps:

A. The Potential

```
Program: dbh
Sample input file: in.dbh
Output: dbh.dat - contains tabulated values of the harmonic
coefficients
                  for the Legendre expansion of the density, potential
and
                  radial force at the specified radii for the entire
model
          h.dat - same as above for halo only
          b.dat - same as above for bulge only
         mr.dat - gives mass and radial extent (or edge) of disk,bulge
and halo
```

```
Parameters in in.dbh:
y                     #yes we want a halo (or no)
-6.0 1.32 1 .1 0.8    #psi0, v0, q, (rc/rk)^2, ra
y                     #yes we want a disk (or no)
.867 1 5 .1 .5        #M_d, R_d, R_outer, z_d, dR_trunc
y                     #yes we want a bulge (or no)
14.45 -3.7 .714       #rho_b, psi_cut, sig_b
.01 5000              #delta_r, nr
10                    #number of harmonics (even number)
dbh.ps/ps             # PGPLOT graphics device for the plots produced.
```

The program asks for parameters describing each of the components. You have the option of including any combination of components (though I think models without a halo won't work).

```
Halo Parameters:
psi0 - central potential - the smaller (the more negative) this
parameter
       the deeper the potential and the more extended the halo
v0   - v0 = sqrt(2.0)*sigma0 where sigma0 is the central velocity
```

dispersion. roughly the velocity where the halo rotation curve peaks
q - an optional flattening parameter for the potential - generally 0.7 < q < 1.05 - q=1.0 will give a nearly spherical halo
(rc/rk)^2 - a core smoothing parameter - ratio of the core radius to the derived King radius for halo only models set this to 1.0. For multicomponent models, this can be a smaller number 0.0 to 0.1. I've found that with this parameter=0.0 the program can crash.
Ra - a scaling radius for the halo -
The halo Ra radius is the radius at which the halo rotation curve, at its initial slope ignoring cutoffs and the other components, reaches v0.

Disk Parameters:
M_d - mass of the exponential disk ignoring cutoffs
R_d - exponential scale length
R_outer - outer radius where we begin to truncate the disk density
z_d - disk scale height assuming a sech^2(z/zd) vertical density law
dR_trunc - truncation width - the disk density smoothly drops to zero in the range R_outer < R < ~R_outer + 2*dR_trunc.

Bulge Parameters:
rho_b - bulge central density
psi_cut - bulge cut-off potential psi0 < psi_cut < 0.0
- energy cut-off for the bulge
sig_b - bulge central potential

Potential parameters:
dr - the width of the radial bins used to calculate the potential
nr - number of radial bins - initially a guess since we don't know the radial extent of the system
lmax - the largest value in the potential harmonic expansion - use lmax=2 to get a quick look at the mass profile and lmax=10 for the final calculation of the model

Creating a galaxy model from these parameters is sort of a black art since the halo and bulge models are not parameterized in terms of their mass profiles but rather properties of their distribution functions. Changes in psi0, v0 etc. have weird but predictable effects on the mass profile.

The halo is a flattened analogue of the King model so the concentration (R_tidal/R_core) is determined by the dimensionless central potential ψ_0/σ_0^2. The more negative the value the greater the concentration. The parameters R_a and v_0, affect the scaling of the halo mass profile.

The effect of different bulge parameters is more predictable. Decreasing the central velocity dispersion will create a more centrally concentrated bulge and decreasing the psi cut off will truncate the bulge and decrease its total mass.

The disk is parameterized directly by its mass profile so its effect on the rotation curve is predictable ahead of time.

Hit and miss seems to be a good strategy for finding a suitable profile. Generate a model to lmax=2 and then view the resulting rotation curve by typing 'make vr.dat'. This uses the program to generate the file vr.dat which tells you the contributions to the total rotation curve. Another useful file is 'mr.dat' which tells you the mass and radial extent of the disk bulge and halo.

The program plotforce will also generate the rotation curves for you directly from the dbh.dat, b.dat and h.dat files.

The potential is determined iteratively: starting from an initial guess at the potential, the density implied by the halo and bulge DFs is calculated, the disk density added, and the potential of that mass distribution is used as starting point for the next iteration. Initially only the monopole (l=0) components are calculated until the model converges, then one more harmonic is added per iteration up to the maximum requested, and once all harmonics are included the iterations are continued until the outer (tidal) radius of the halo is unchanged between iterations. At each iterations plots of the harmonic expansion coefficients are produced.
If the tidal radius reported is "outside grid" for a large number of iterations, increase the number of radial bins or increase their size. Sometimes infinite tidal radii are also reported: this happens when the total mass of the model using the current guess for the potential is insufficient to generate a potential well as deep as requested. If this persists over many iterations, again increase the number or size of the radial bins.

B. Disk distribution function

Program: getfreqs
Input files: dbh.dat h.dat b.dat
Output: freqdbh.dat

getfreqs tabulates various characteristic frequencies (omega, kappa etc.)
in the equatorial plane for use by diskdf below.

Program: diskdf
Input files: freqdbh.dat dbh.dat in.diskdf
Output files: cordbh.dat toomre.dat

The program diskdf iteratively calculates the correction functions for the disk distribution function. These functions are multiplicative corrections to the surface density and vertical velocity dispersion which appear to leading order in the Shu (1969) distribution functions. See KD95 for details. It requires the sample parameters:

```
.47 1.0   #central radial vel. dispersion, exponential scale length of sig_r^2
50        #number of radial intervals for correction functions
10        #number of iterations
diskdf.ps/ps # PGPLOT device for plot of correction functions.
```

It also outputs the Toomre Q as a function of radius in the file toomre.dat.

C. Generating N-body realizations

Programs: gendisk, genbulge, genhalo
Input files: cordbh.dat dbh.dat

gendisk parameters:

```
4000        #number of particles
-1          #negative random integer seed
1           #1=yes we want to center 0=no we don't
dbh.dat     #multipole expansion data file
```

genbulge parameters:

```
0.5       #streaming fraction
1000      #number of particles
-1        #negative integer seed
1         #center the data 1=yes
dbh.dat   #harmonics file
```

genhalo parameters:

```
0.5      #streaming fraction
6000     #number of particles
-1       #negative integer seed for random number generator
1        #1=yes we want to center
dbh.dat  #multipole expansion data file
```

The streaming fraction, f, sets the fraction of orbits with L_z > 0. The remaining fraction, 1-f, have L_z<0. With this parameter you can therefore vary the rotation of the bulge and the halo. f=0.5 refers to the non-rotating case.

The N-body data are written to the stdout so the programs should be run as:

```
gendisk < in.disk > disk
genbulge < in.bulge > bulge
genhalo < in.halo > halo
```

Format is ascii with data arranged as:

```
N_bodies time
m_1 x_1 y_1 z_1 vx_1 vy_1 vz_1
m_2 x_2 y_2 z_2 vx_2 vy_2 vz_2
m_3 x_3 y_3 z_3 vx_3 vy_3 vz_3
.
.
.
etc.
```

There is a shell script 'mergerv' which can merge the disk, bulge and halo
files into a single N-body file.

The program tobinary turns the ascii files into a simple binary format, listing first the number of particles, then all their masses, then the time, and finally the x,y,z,vx,vy,vz coordinates for each particle.