Made-to-measure $N$-body systems

D. Syer$^{1,2}$ and S. Tremaine$^1$

$^1$Canadian Institute for Theoretical Astrophysics, McLennan Labs, 60 St. George Street, Toronto M5S 1A7, Ontario, Canada
$^2$Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Straße 1, 85748 Garching-bei-München, Germany

Accepted 1996 May 8. Received 1996 April 24; in original form 1995 November 6

ABSTRACT

We describe an algorithm for constructing $N$-body realizations of equilibrium stellar systems. The algorithm complements existing orbit-based modelling techniques using linear programming or other optimization algorithms. The equilibria are constructed by integrating an $N$-body system, while slowly adjusting the masses of the particles until the time-averaged density field and other observables converge to a prescribed value. The procedure can be arranged to maximize a linear combination of the entropy of the system and the $\chi^2$ statistic for the observables. The equilibria so produced may be useful as initial conditions for $N$-body simulations or for modelling observations of individual galaxies.

Key words: methods: numerical – galaxies: kinematics and dynamics.

1 INTRODUCTION

One of the central problems in stellar dynamics is to construct made-to-measure stellar systems. For example, (i) when modelling observations of an elliptical galaxy, we wish to find a phase-space distribution function $f(r, v)$ (hereafter DF) that solves Poisson’s equation and the collisionless Boltzmann equation and reproduces the observed surface-brightness distribution, rotation curve, velocity-dispersion profile, etc. (in the sense of minimizing $\chi^2$, the mean-square deviation between the observations and the model); (ii) when conducting simulations, we wish to construct initial states that are $N$-body realizations of equilibrium stellar systems with given density profile, rotation curve, bulge/disc ratio, etc.

Existing methods for constructing made-to-measure stellar systems can be classified as follows.

(1) DF-based methods, which solve directly for the DF $f(r, v)$. These generally require that all the integrals of motion are known explicitly (e.g., if the potential is spherical or has Stäckel form) or that the DF is assumed to depend only on known analytic integrals. These include methods that fit observations to few-parameter models with analytic DFs such as spherical Michie or King models; a difficulty with such methods is that the dependence of the derived properties of the galaxy on the choice of the model can be large and uncertain (e.g., Merritt & Tremblay 1994). A more flexible DF-based method is described by Dejonghe (1989), who expands the DF for a spherical system in a truncated series of basis functions and minimizes $\chi^2$ subject to the constraint that the DF is positive, using quadratic programming. DF-based methods can be applied to axisymmetric galaxies, and are completely general so long as the DF does not depend on a third integral (e.g. Kuijken 1995; Qian et al. 1995).

(2) Moment-based methods, which find solutions of the Jeans equations (or higher-order velocity moments of the collisionless Boltzmann equation) that minimize $\chi^2$. This method was used by Binney & Mamon (1982) to construct spherical models of M87, and has been applied to axisymmetric models by Binney, Davies & Illingworth (1990), Magorrian & Binney (1994) and van der Marel et al. (1994). A drawback is that this procedure does not guarantee that there is a positive-definite DF with the required velocity moments.

(3) Orbit-based methods (Schwarzschild 1979, 1993) compute the density distribution of a large library of orbits in a fixed potential, and then determine the weight each orbit must have in order to reproduce the desired final state. In this method the DF or other integrals of motion are not explicitly required, although the DF can be regarded as a sum of delta-functions on the phase-space surfaces covered by the orbits. Orbit-based methods are generally ill-conditioned. The ill-conditioning can be removed by iterating from a smooth initial guess for the orbit weights using the Richardson-Lucy method (Newton & Binney 1984), but the final weights will then depend in a complicated way on the initial guess. A better procedure is to minimize $\chi^2$ minus a profit function that measures the smoothness of the distr-
bution of orbit weights. The profit function may be an entropy (Richstone & Tremaine 1988) or any other function that is large when the DF is smooth (Merritt 1993).

The most flexible of these are orbit-based methods, since they do not require that the integrals of motion are known and the approximation they provide to the DF is known to be positive if the orbit weights are positive.

The goal of this paper is to introduce a novel class of methods for constructing made-to-measure N-body realizations of stellar systems. In the classification above, these methods might be called ‘particle-based’; they work by sculpting an initial N-body system until it matches the prescribed density field and other observables.

2 THE ALGORITHM

In most of our discussion we restrict ourselves to constructing stellar systems in a fixed potential \( \Phi(r) \). We suppose that \( \Phi \) admits a collisionless equilibrium configuration specified by a distribution function in phase space \( f(r, v) \). Thus \( f \) satisfies the time-independent collisionless Boltzmann equation:

\[
\frac{\partial f}{\partial t} - \nabla \Phi \cdot \frac{\partial f}{\partial v} = 0. \tag{1}
\]

An ‘observable’ of the stellar system is a quantity of the form

\[
Y_j = \int K_j(z) f(z) \, dz,
\]

where \( z = (r, v) \), and \( K_j \) is a known kernel. Suitable observables include the surface or volume density at a given point, the surface density times the mean line-of-sight velocity, the surface density times any moment of the line-of-sight velocity, etc.

Now consider a system of \( N \) particles having weights \( w_i \) and phase-space positions \( z_i(t) \) \((i \leq N)\). The observables of this system are

\[
y_j(t) = \sum_{i=1}^{N} w_i K_j(z_i(t)). \tag{3}
\]

Our goal is the following: given a set of distinct observables \( Y_j, j = 1, \ldots, J \), construct a system of \( N \) particles orbiting in the potential \( \Phi(r) \) whose time-averaged observables \( \langle y_j(t) \rangle \) are equal to \( Y_j \).

We hope that this paper will stimulate interest in seeking better particle-based algorithms than the one we describe.

2.1 The force of change

The heart of the algorithm is a prescription for changing the weights \( \{w_i(t)\} \) as the particles proceed along their fixed orbits in the potential \( \Phi(r) \). The prescription is similar to that employed by Syer & Tremaine (1995) in a different context. It consists of applying gentle pressure on \( w_i \) according to the value of \( \Delta_j = y_j(t)/Y_j - 1 \): if \( \Delta_j < 0 \) then increase \( w_i \), and if \( \Delta_j > 0 \) then decrease \( w_i \). More precisely, we let

\[
\frac{dw_i(t)}{dt} = -\epsilon w_i(t) \sum_{j=1}^{J} \frac{K_j(z_i(t))}{Z_j} \Delta_j(t). \tag{4}
\]

where \( \epsilon \) is small and positive, and \( Z \) is so far arbitrary. The factor \( w_i \) on the right-hand side ensures that \( dw_i/dt \to 0 \) as \( w_i \to 0 \), so that \( w_i \) cannot become negative. The factor \( K_j/Z_j \) ensures that the difference \( \Delta_j \) changes the weight \( w_i \) only if particle \( i \) is contributing to observable \( j \). Equation (4) is closely related to Lucy’s (1974) method for solving integral equations with noisy data.

Some insight into the solutions of equation (4) is offered by the following argument. Since \( \epsilon \) is small, the weights \( w_i \) change only over many orbits; thus we may orbit-average:

\[
\frac{dw_i(t)}{dt} = -\epsilon w_i(t) \sum_{j=1}^{J} \frac{K_j(z_i(t))}{Z_j} \langle \Delta_j \rangle. \tag{5}
\]

Here \( \langle \cdot \rangle \) denotes the time-average over an interval that is much longer than a typical orbital period but much less than \( \epsilon^{-1} \) periods, and \( \langle K_j \rangle \) is shorthand for the time-independent quantity \( \langle K_j(z(t)) \rangle \). We have also assumed that the fluctuations in \( K_j(z(t)) \) and \( \Delta_j(t) \) are not correlated, which is plausible if many particles contribute to \( \Delta_j \). We now define

\[
\Theta_k(t) \equiv \left( \frac{Y_k}{Z_k} \right)^{1/2} \langle \Delta_k \rangle, \tag{6}
\]

which obeys the differential equation

\[
\frac{d\Theta_k}{dr} = -\epsilon \sum_i w_i(t) \frac{\langle K_k \rangle \langle K_i \rangle}{(Y_k Z_k Y_i Z_i)^{1/2}} \Theta_i. \tag{7}
\]

When we are close to convergence \((|\Theta| \ll 1)\), the behaviour of the right-hand side is dominated by changes in \( \Theta_k \) rather than changes in \( w_i \), so we may replace \( w_i \) by a constant, \( w_i^0 \). Then the vector \( \Theta \) satisfies the matrix equation

\[
\frac{d\Theta}{dr} = -\epsilon A \cdot \Theta, \tag{8}
\]

where the matrix \( A \) has components

\[
A_{ij} = \sum_{k=1}^{N} w_i^0 \frac{\langle K_k \rangle \langle K_i \rangle}{(Y_k Z_k Y_i Z_i)^{1/2}}. \tag{9}
\]

The solutions to equation (8) have the form

\[
\Theta_k = \sum_{m=1}^{\infty} \lambda_m \exp \left(-\lambda_m r\right), \tag{10}
\]

where the eigenvalues \( \{\lambda_m\} \) are solutions of the equation

\[
\det (\epsilon A - \lambda I) = 0. \tag{11}
\]

Since \( A \) is positive-definite by construction (i.e., \( x^\top A x > 0 \) for all \( x \)), all of its eigenvalues are positive and so \( \lambda_m > 0 \). This argument suggests that all observables converge to the desired values \((|\langle \Delta \rangle| \to 0)\) on \( O(\epsilon^{-1}) \) orbital periods, if \( \epsilon \) is sufficiently small and we start close to the correct final state.

For comparison, orbit-based methods evaluate and store the entire matrix \( \{K_i\} \) (by following the orbits for a fixed time that is much longer than the orbital period; the matrix
\[ \langle K \rangle \] is often called the 'orbit library'), then solve the matrix equation

\[ Y_j = \sum_{i=1}^{N} \langle K_i \rangle w_i; \tag{12} \]

if \( N > J \), the matrix equation is ill-conditioned and must be solved subject to a constraint that maximizes some profit function such as entropy. The storage needed by orbit-based methods is \( O(NJ) \), whereas the storage needed by particle-based methods is only \( O(N) \).

### 2.2 The kernel

Suppose that the observable \( y \) is the density at \( r \). What is the appropriate kernel \( K_y(r', v') \)? One approach would be to use a smooth kernel (à la SPH) with

\[ K_y(r', v') = W(r-r', h), \tag{13} \]

where \( W \) is some smooth function, and \( h \) is the resolution length (possibly a function of \( r \)); \( W \) should be normalized so that \( \int W(r, h) \, d\mathbf{r} = 1 \). If \( h \) is too large, resolution is lost, while if \( h \) is too small, the observables \( y \) fluctuate strongly, because too few particles contribute to a given observable.

The smooth-kernel approach is expensive, because nearest neighbours have to be found at each time-step, and we have therefore adopted a different method. We first divide the coordinate space into bins. Then we set \( K_y(r', v') \) to zero if \( r \) and \( r' \) are not in the same bin, and equal to the inverse of the volume of the bin otherwise. Obviously \( K_y \) and \( K_i \) are the same if \( r \) and \( r_i \) are in the same bin, so there can be at most one density observable per bin. We are still free to choose the parameter \( Z_i \) in equation (4); a simple choice is to set \( Z_i = \) equal to the inverse of the volume of the bin. Thus \( K_i(Z_i)/Z_i \) is unity if \( r \) is in bin \( i \), and zero otherwise.

An improvement to this simple scheme, which we employ here, is to borrow from the smooth-kernel approach and to smear each particle into neighbouring bins. Each particle is replaced by a Gaussian distribution at the bin centre closest to the particle position, with dispersion equal to half the bin width. Neighbouring bins are assigned a weight corresponding to the integral of the Gaussian over those bins (normalized so that the total contribution over all bins is \( w_i \)). The bin closest to the particle position thus contains about two-thirds of its weight. This procedure corresponds nicely to the phenomenon of seeing in the case of a projected observable.

### 2.3 Resolution and smoothing

Suppose that we divide the \( d \)-dimensional coordinate space into \( M^d \) bins, and that each density observable corresponds to the mass per unit volume in a single bin. Then at any given time there are on average \( N/M^d \) particles contributing to each observable, and the rms statistical fluctuations in \( \Delta \) will be of order \( \delta \sim (M^d/N)^{1/2} \). These fluctuations can be kept small if \( d = 1 \) (spherical or one-dimensional systems); for example, if \( M = 30 \) and \( N = 1000 \), we have \( \delta \sim 0.2 \). However, for triaxial systems \( (d=3) \) the fluctuations will be much larger for reasonable values of \( M \) and \( N \).

To improve this situation, we employ a form of temporal smoothing which effectively boosts \( N \) without any need for extra storage or computation per time-step. This is implemented by replacing \( \Delta_i(t) \) in equation (4) with \( \Delta_i(t) \), where

\[ \Delta_i(t) = a \int_{t-	au}^{t} \Delta_i(t-	au) e^{-\eta \tau} \, d\tau, \tag{14} \]

and \( a \) is small and positive. This quantity is most easily calculated using the equivalent differential equation

\[ \frac{d\Delta}{dt} = a(\Delta - \bar{\Delta}). \tag{15} \]

Each particle is smeared backwards along its trajectory, and represents a set of virtual or ghost particles strung out along the orbit with ever decreasing weights. In effect, temporal smoothing increases the effective number of particles from \( N \) to

\[ N_{\text{eff}} = N \sqrt[1/2]{t_{1/2}/\Delta t}, \tag{16} \]

where \( \Delta t \) is the time-step, and \( t_{1/2} = (\ln 2)/a \) is the half-life of the ghost particles.

Some insights into the effect of this smoothing procedure are given by the following argument. As in Section 2.1, let \( \langle \cdot \rangle \) denote the time-average over an interval that is much longer than a typical orbital period but much less than \( a^{-1} \). The time-averaged version of equation (15) is

\[ \frac{d\langle \Delta \rangle}{dt} = a(\langle \Delta \rangle - \langle \bar{\Delta} \rangle) \quad \text{or} \quad \frac{d\bar{\Theta}}{dt} = a(\Theta - \bar{\Theta}), \tag{17} \]

where \( \Theta \) is defined by replacing \( \Delta \) by \( \bar{\Delta} \) in (6). The evolution of \( \Theta \) is described by equation (8), with \( \Theta \) replaced by \( \bar{\Theta} \) on the right-hand side. The solution to equations (8) and (17) has the form (10), with eigenvalues \( \lambda \) that satisfy

\[ \det (\lambda I - \lambda^2 1 + \lambda^2 \lambda^{-1}) = 0. \tag{18} \]

In the simple case of a single observable, we have

\[ \lambda = \frac{1}{2} \alpha + \frac{1}{2} \left( \alpha^2 - 4 \alpha eA \right)^{1/2}, \]

where \( \alpha > 0 \). For \( \alpha \gg eA \), we find \( \lambda \approx eA \), which is the same convergence rate that would obtain without temporal smoothing. However, for \( \alpha < 4\alpha eA \), \( \lambda \) is complex, so the observables execute damped oscillations rather than converging smoothly, and the convergence rate is \( \lambda = 1/2 \text{Re}(\alpha) \), which is slower than the convergence rate without smoothing whenever \( \alpha < 2eA \).

We conclude that excessive temporal smoothing is undesirable and that the maximum smoothing time \( a^{-1} \) should satisfy

\[ \varepsilon < a, \tag{19} \]

assuming \( |\lambda| = O(1) \).

### 2.4 Maximum entropy

If the number of particles exceeds the number of observables, the differential equations (4) are ill-conditioned. In practice, this means that the observables \{\( y_i(t) \)\} will converge fairly rapidly to \{\( y_i \)\}, but that the individual particles...
weights \( \{w_i\} \) will continue to drift long after the observables
have converged. Such behaviour is undesirable, as we would
like to use the particle weights to predict other properties of
the stationary stellar system.

To remove the ill-conditioning, we can maximize some
form of profit function, such as the entropy

\[ S = -\sum_i w_i \log (w_i/m_i), \]

(20)

where \( \{m_i\} \) is a pre-determined set of weights (the 'prior').

Thus we maximize the function

\[ F = \mu S - \frac{1}{2} \chi^2, \]

(21)

where

\[ \chi^2 = \frac{1}{N} \sum_j \Lambda_j^2. \]

(22)

Equation (4) is replaced by

\[ \frac{dw_i(t)}{dt} = \omega w_i(t) \left[ \frac{\partial S}{\partial w_i} - \sum_{j=1}^{J} \frac{K[z_j(t)]}{Z_j} \Lambda_j(t) \right]. \]

(23)

The factor \( \mu \) is a measure of the relative contribution of \( \chi^2 \)
and \( S \) to the final state: if \( \mu \) is large, we get a smooth solution
(in the sense that the \( \{w_i\} \) are close to the \( \{m_i\} \)) but \( \chi^2 \) is
large, while if \( \mu \) is small, the solution is not smooth but \( \chi^2 \) is
likely to be smaller. The parameter \( \mu \) can be specified at the
start of the calculation, or adjusted as the calculation pro­
ceeds using a prescription such as

\[ \frac{d\mu(t)}{dt} = \mu \eta [D^2 - \chi^2(t)], \]

with \( 0 < \eta \ll 1 \); in this case, \( \chi^2 \) will converge to the specified
value \( D^2 \). In the simulations described here, we have kept \( \mu \)
constant.

The condition \( N > J \) (number of particles exceeds the
number of bins) is neither a strict criterion for ill-condition­
ing nor a necessary condition for a sensible result, since
different observables are not independent (both because of
the spatial smoothing described in Section 2.2 and because
a single orbit contributes to many observables).

3 RESULTS

3.1 One-dimensional results

We first present the results of experiments in one dimen­
sion, to illustrate the effects of the various parameters in the
algorithm. We use for the background potential

\[ \Phi(x) = \frac{1}{(b^2 + x^2)^{1/2}} \]

(25)

with \( b = 1/3 \), and for the observables we use the density
distribution

\[ \nu(x) = \frac{1}{(b^2 + x^2)^{5/2}}. \]

(26)

The period of a low-energy orbit is \( 2\pi b^{3/2} = 1.209 \). We use a
fourth-order leapfrog with time-step \( \Delta t = 0.1 \) to integrate
the particle orbits, and adjust the weights \( \{w_i\} \) after every
time-step according to equation (23). To measure the
density we use a uniform grid with 16 bins in the range
\( x \in [-1, 1] \). With \( b = 1/3 \) there is a density contrast of about
200 between the inner and outermost bins. We evaluate the
success of the algorithm by examining the time evolution of
the \( \{w_i\} \) and of \( \chi^2 \). The parameter values we use are summa­
rized in Table 1. In the initial state the particles were uni­
formly distributed in \( x \in [-1, 1] \) with equal weights. The
velocities were uniformly distributed in the range allowed
by the condition that their orbits are restricted to
\( x \in [-1, 1] \).

Fig. 1 shows the evolution of the weights \( \{w_i\} \) for simula­
tion A, in which \( z = 0.0524, \epsilon = 0.025 \) and \( \mu = 0.01 \). All the
\( \{w_i\} \) evolve smoothly, and converge in a time of order \( \epsilon^{-1} \).
We note that the final equilibrium has a wide range of masses, which suggests that some gain in efficiency might be possible if the particles with smaller $w$ could be replaced by a smaller number of particles with a larger $w$ (for example, we could discard particles whose weights fall below a threshold and replace them with new particles on randomly chosen orbits, or combine low-weight particles with nearby orbits).

Fig. 2 shows the evolution of $\chi^2$ in simulation A and compares with simulation B in which the orbit averaging is removed ($\tau = \infty$). Both the final value of $\chi^2$ and its fractional fluctuation are smaller in A, illustrating the benefits of orbit-averaging. Fig. 3 shows $\chi^2$ in simulations A, C and D. Simulations C and A differ only in the parameter $\varepsilon$: C has larger $\varepsilon$ and hence converges more rapidly, although the final states in A and C are very similar.

Simulations D and C differ only in the parameter $\mu$: D has $\mu = 0$ (no entropy constraint). The final value of $\chi^2$ is lower in simulation D – and the fluctuations are larger – since the simulation does not have to trade higher $\chi^2$ for lower entropy.

Fig. 4 shows the evolution of the same six particles from simulations A, B, C and D. Comparing A ($\varepsilon = 0.48 \sigma$) with C and D ($\varepsilon = 0.95 \sigma$), we see the effect of excessive temporal smoothing (equation 19): the trajectories $w_t$ are noisy and do not converge smoothly to their final values. Comparing simulations C and D, we see that this noise is worse for small $w$ when the entropy constraint is removed. Note that the larger $w$ are not much different in C and D. The smaller $w$ do not converge in D, but they do not contribute much to $\chi^2$. In C the entropy constraint has tied down the smaller values of $w$.

![Figure 2](https://example.com/figure2.png)

**Figure 2.** The rms fractional density error $\chi$ (equation 22) for simulation A (bottom curve), and simulation B (top curve). Simulation B has no orbit averaging. The initial state has small $\chi^2$, but is not in equilibrium, so $\chi^2$ immediately grows as soon as the particles start to move.

![Figure 3](https://example.com/figure3.png)

**Figure 3.** The density error $\chi$ for simulations A (top curve), C (middle curve) and D (bottom curve). The initial state has small $\chi^2$, but is not in equilibrium, so $\chi^2$ immediately grows as soon as the particles start to move.
Figure 4. A representative sample of six of the \( \psi \) for each of the simulations A (top left), B (top right), C (bottom left) and D (bottom right).
3.2 Three-dimensional results

We have performed a number of three-dimensional simulations. The parameters for the algorithm are in each case the same, as given in the last row of Table 1.

| Table 1. The parameter values used in the simulations. |
|-----------------------------------------------|
|          | $N$  | $10^2\alpha$ | $10^2\epsilon$ | $10^2\mu$ | $D$   |
| A        | 100  | 5.24         | 2.5             | 1          | 0.05  |
| B        | 100  | $\infty$    | 2.5             | 1          | 0.05  |
| C        | 100  | 5.24         | 500             | 1          | 0.05  |
| D        | 100  | 5.24         | 500             | 0          |       |
| E        | 1000 | 51.9         | 25              | 1          | 0.05  |
| 3D       | 4000 | 5.24         | 2.5             | 0.1        | 0.05  |

3.2.1 Mass models

The mass models were of three types:

- **PS** Plummer sphere, with density law given by equation (26), with $x$ denoting the radius and $4\pi G = 3b^2$. We denote this model by ‘PS’.

Fig. 5 compares the effects of orbit averaging and particle number. Simulation E has 10 times the number of particles as A, but the smoothing time $\tau^{-1}$ is 10 times smaller; as a consequence, the noise in the two simulations is about the same. The shorter smoothing time in E allows a larger $\epsilon$, so that E converges faster than A; note that the time coordinate in Fig. 5 has been shrunk by a factor of 10 for simulation A. Thus there is a trade-off between number of particles and convergence time; simulations A and E each take about 2 min of CPU on a DEC Alpha 3000/300. Fig. 6 shows the same six particle weights as in Fig. 4.
PT Triaxial Plummer model with density law given by equation (26), with $x$ denoting the triaxial radius, $s$, defined by
\[ s^2 = \frac{x^2}{A^2} + \frac{y^2}{B^2} + \frac{z^2}{C^2}, \]
and $(A, B, C) = (1.41, 1.12, 1.00)$.

ST Schwarzschild's (1979) model with axis ratios given by $(A, B, C) = (2.00, 1.25, 1.00)$.

3.2.2 Force calculation

The force calculation is carried out in one of two ways:

A Analytically. In the case of PT the analytic potential is not that which would be self-consistently generated by the mass model, but rather the Plummer potential with $s$ replacing $x$ in equation (25).

F Numerically using the Fourier convolution theorem and FFT on a $16^3$ grid. Mass is assigned to the grid from the desired mass model, and the forces and potential are calculated once only at the beginning of the simulation.

In the ST models the numerical and analytic potentials are significantly different (40 per cent on average), because the model has infinite mass and has been truncated at finite radius. In the PS models the forces agree to within a few per cent.

3.2.3 Initial condition

We also use three different types of initial condition:

B Particles uniformly distributed in $r < 1$, with velocities uniform in the range allowed by the condition that their orbits remain in $r < 1$. In the Schwarzschild potential this initial condition produces a preponderance of box orbits; hence we refer to it as 'box-dominated'.

T Particles uniformly distributed in $r < 1$. Additionally, 20 per cent have their velocities chosen as for B, and 80 per cent are given velocities perpendicular to their radius vector with magnitude equal to that of a circular orbit in a spherical potential with the same total force. In the Schwarzschild potential this initial condition produces more tube orbits than B; hence we refer to it as 'tube-dominated'.

E Particles distributed in $r < 1$ according to the required mass model. Thus there are extra particles in the inner regions as compared with B or T. Velocities chosen as for T.

To summarize the notation by example, simulation PSAB is a Plummer sphere with analytic forces, and box-dominated initial condition; simulation STFE is a Schwarzschild triaxial model with Fourier numerical forces, and extra sampling of the inner regions.

In each case we measure the density using a uniform cubical grid with $M = 16$ elements on a side for a total of $2^{12}$ observables. The core radius of all the models $b = 1/3$, and we restrict particles to orbits with $r < 1$. Simulation PSAB took about 200 min of CPU on the same machine as simulation A, a factor of ~100 longer; most of this factor reflects the factor of 40 increase in the number of particles, while the rest reflects the increased complexity of the differential equations (which must follow six phase-space coordinates instead of two) and increased access time to the much larger array which represents the observables.

Orbit classification is carried out by recording the maximum and minimum values of the angular momentum $L$ about the $z$ axis (the shortest principal axis of the triaxial models). Box orbits tend to reverse direction and 'retrace' their paths, so $L_{\text{max}} \approx -L_{\text{min}}$. Fig. 7 shows the distribution of $L_{\text{max}}$ and $L_{\text{min}}$ for simulation PTFT. The box orbits essentially lie in the upper left quadrant. Tube orbits are here defined as those with $L_{\text{max}}L_{\text{min}} > 0$. Table 2 gives the percentage of tube orbits in each of the three-dimensional simulations.

Figure 7. A scatter plot of $L_{\text{max}}$ versus $L_{\text{min}}$ for simulation PTFT (only 10 per cent of the total number of particles is shown). The values of $L$ plotted are relative to the average magnitude of the angular momentum of each particle.
4 DISCUSSION

4.1 Choosing the parameters

How should the parameters, \((N, M, \alpha, \varepsilon, \mu)\) be chosen to optimize the investment in CPU? The arguments given in Section 2 lead to the following guidelines.

First, choose \(D\), the value of \(\chi\) which is as large as can be tolerated – in the simulations above we had \(\chi\) of a few times \(10^{-2}\). Then choose the resolution required, via the number of resolution elements \(M^d\). This informs our choice of \(N\) and \(\alpha\) through (cf. equation 16)

\[
\chi^2 \approx \frac{M^d}{N} \sim \frac{M^d}{\varepsilon \Delta t}.
\]

Thus, if \(N\) is limited by storage requirements, equation (28) determines the desired value of \(\alpha\).

Once we have \(\alpha\), then we must choose \(\varepsilon\) smaller than \(\alpha\) (equation 19), bearing in mind that the convergence time will be of order \(\varepsilon^{-1}\) orbital periods. In the simulations we found that \(\varepsilon \sim 0.5\alpha\) is about right.

The final step is to choose \(\mu\). This is largely a matter of taste, since \(\mu\) determines the balance between smoothness and accuracy (Merritt & Tremblay 1994 discuss this issue in another context). The natural choice for \(\mu\) is the one for which \(\chi^2\) reflects the observational errors, and this is the value which will be obtained if \(\mu\) is allowed to vary according to equation (24). We do not recommend equation (24) for general use, however, because convergence of \(\mu\) can be rather slow. One method, which has the advantage of being at least semiquantitative, is to do a run including equation (24), with \(D\) equal to the preferred value of \(\chi\). Far from having to converge, it only has to get \(\mu\) into the right ballpark, and then we go back to constant \(\mu\) for an extended run.

4.2 The initial condition and prior

The initial condition of the system should be chosen to sample phase space as well as possible. Considerable effort is sometimes devoted to this choice in orbit-based calculations. In the simulations presented here we have chosen simply to sample phase-space uniformly (Section 3). With detailed knowledge of the potential and its orbit families, more informed choices could be made.

We can think of the prior \(\{m_i\}\), together with the particle positions and velocities, as a random realization of some known DF \(f_\mu(r, v)\). In the simulations presented here, we use \(\{m_i\}\) which are all equal. In those where the initial particle positions and velocities sample phase space uniformly, for example, we are effectively using \(f_\mu\), which is initially equal to a constant. If the initial condition is well mixed, as we may reasonably hope, then \(f_\mu\) is independent of time. The smoothness constraint has the effect of driving the DF of the system towards \(f_\mu\).

With a suitable choice of prior \(\{m_i\}\), the final equilibrium should not depend on the initial condition. It should merely reflect the choice of \(\mu\), which determines the balance between smoothness and accuracy. The resolution of the final equilibrium in phase space, however, may be affected by the choice of initial condition. The wide range of weights in Fig. 1 is a reflection of this fact. However, only perfect knowledge of the target DF would allow one to set up an initial condition which led to an equilibrium with all the \(\{w_i\}\) equal. This problem could be alleviated on the fly by methods that kill particles with low weights and split particles with high weights into several daughter particles with similar orbits.

4.3 Comparison with orbit-based methods

The particle-based method we have described here has several advantages over orbit-based methods.

1. Particle-based methods use less storage: if there are \(N\) particles (or orbits) and \(J < N\) observables, orbit-based methods must store \(O(N)\) variables (the contribution of each orbit to each observable), while a particle-based method stores only \(O(N)\) variables (the particle weights at a
given time). This advantage is particularly important in systems with a large number of observables (triaxial systems, or systems in which the entire line-of-sight velocity distribution is observed).

(2) Although we have only discussed the construction of stellar systems in a fixed potential, it should be possible to generalize particle-based methods so that the potential is determined self-consistently by the particles. Perhaps the best approach would be to expand the potential as a linear combination of a set of basis functions (which can be chosen to preserve any desired symmetries, e.g., spherical symmetry). The coefficient of each basis function, determined from the evolving weights of the particles, could be orbit-averaged (cf. equation 14) to reduce the effects of relaxation.

(3) Model construction with orbit-based methods is a multistep process: first compute the luminosity density from the surface brightness; then solve Poisson's equation assuming (say) constant mass-to-light ratio; then integrate orbits in this potential to construct an orbit library (the matrix containing the contribution of each orbit to each observable); then use some inversion/optimization method such as maximum entropy, Lucy's method, or linear or quadratic programming to determine the orbit weights. In a self-consistent particle-based method all of these steps could be done at the same time.

Particle-based methods also have disadvantages compared to orbit-based methods: they use more computing cycles per orbit because the orbits must be followed for a longer time; as in all Monte Carlo methods, accuracy only scales as \(N^{1/2}\); poorly chosen values of parameters such as \(\alpha\) and \(\varepsilon\) can cripple the method.

The simple experiments we have described show that particle-based methods might be able to compete with orbit-based methods. Possible improvements include (i) non-uniform spatial grids so that the resolution is highest near the centre of the galaxy, (ii) methods that kill particles with low weights and split particles with high weights into several daughter particles with similar orbits, (iii) gridless
methods based on smooth kernels (Section 2.2), and (iv) determining the potential self-consistently from the gravitational field of the particles.

REFERENCES

Binney J., Mamon G. A., 1982, MNRAS, 200, 361
Binney J. J., Davies R. L., Illingworth G. D., 1990, ApJ, 361, 78
Dejonghe H., 1989, ApJ, 343, 113
Kuijken K., 1995, ApJ, 446, 194
Lucy L. B., 1974, AJ, 79, 745

Magorrian J., Binney J., MNRAS, 271, 949
Merritt D., 1993, ApJ, 413, 79
Merritt D., Tremblay B., 1994, AJ, 108, 514
Newton A. J., Binney J., 1984, MNRAS, 210, 711
Qian E. E., de Zeeuw P. T., van der Marel R. P., Hunter C., 1995, MNRAS, 274, 602
Richstone D. O., Tremaine S., 1988, ApJ, 327, 82
Schwarzschild M., 1979, ApJ, 232, 236
Schwarzschild M., 1993, ApJ, 409, 563
Syer D., Tremaine S., 1995, MNRAS, 276, 467
van der Marel R. P., Evans N. W., Rix H. - W., White S. D. M., de Zeeuw T., 1994, MNRAS, 271, 99