The Adaptive TreePM code

J. S. Bagla

Harish-Chandra Research Institute

Allahabad, India

June 1, 2005

(Conference on Computational Cosmology, ICTP)

Why Adaptive TreePM?

Cosmological N-Body simulations must have a large dynamic range in order to simulate a fair sample of the Universe, and at the same time resolve structures like galaxies.

Each particle in these simulations represents a large number of real particles, therefore we must suppress force at short range.

What about existing methods?

Particle-Mesh (PM) codes can be run with a large number of particles but have poor resolution. Force is anisotropic at grid scale.

P^3M codes have a high resolution but tend to slow down at late times. Anisotropy of the PM force persists. If softening parameter is chosen to be much smaller than the inter-particle separation then two body collisions cannot be avoided.

The GOTPM code replaces the particle-particle sum with a tree, and hence does not slow down at late times. Anisotropy of the PM force persists, as does the two body scattering problem.

What about existing methods?

The TPM code replaces the particle-particle sum with a tree, and hence does not slow down at late times. Resolution is better in regions with high density, two body collisions are not a problem. Anisotropy of the PM force persists.

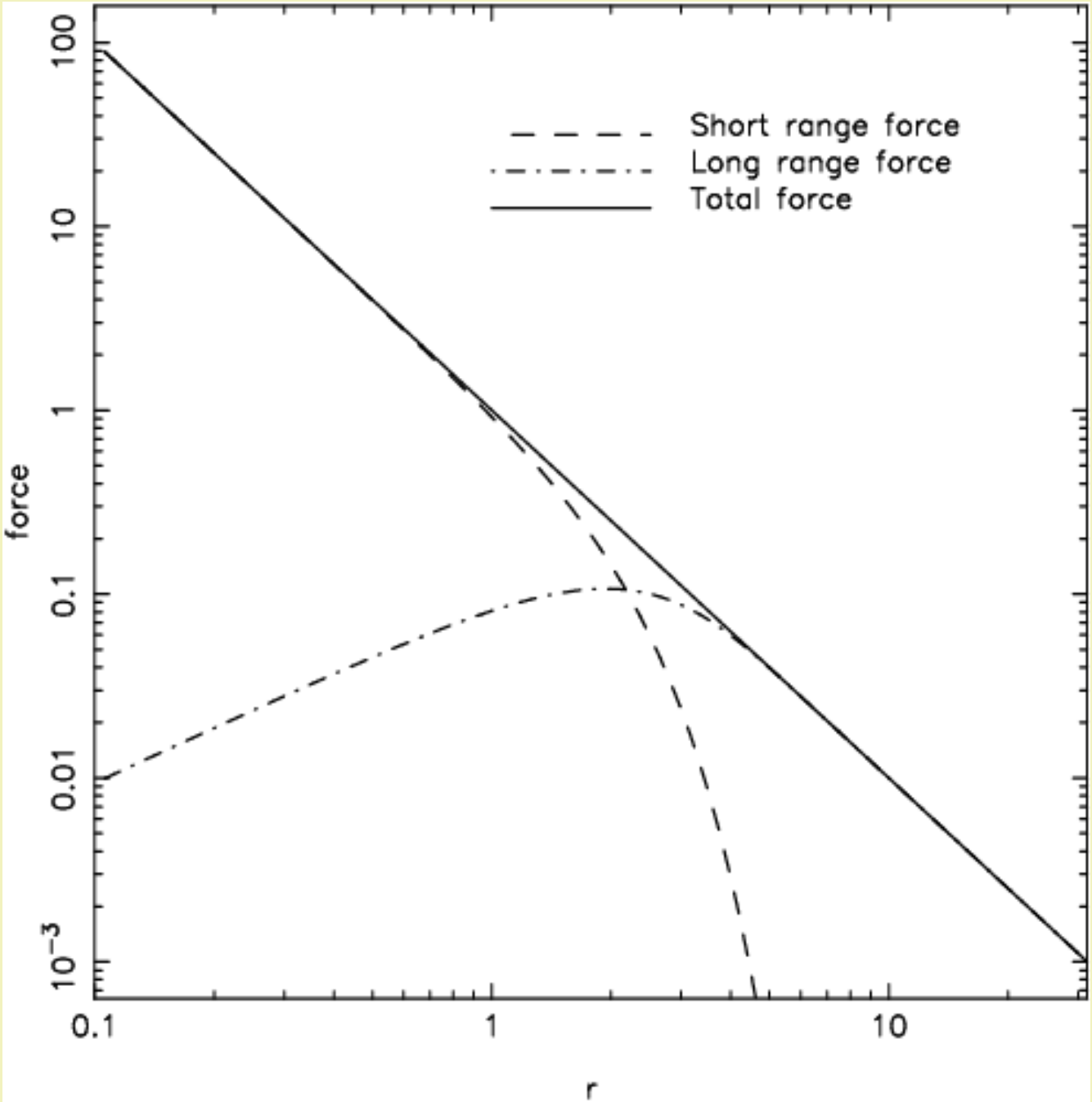
Adaptive Mesh Refinement codes (e.g., ART, MLAPM, RAMSES) have, as the name suggests, an adaptive resolution and the problem of two body scattering is not there. Use of a mesh implies presence of anisotropies in the force at small scales.

The TreePM method

The inverse square force is divided into a long range and a short range part.

$$
\begin{aligned}
\varphi_k &= -\frac{4\pi G \varrho_k}{k^2} \\
&= \varphi_k^l + \varphi_k^s \\
\varphi_k^l &= -\frac{4\pi G \varrho_k}{k^2} \exp\left(-k^2 r_s^2\right) \\
\varphi_k^s &= -\frac{4\pi G \varrho_k}{k^2} \left(1 - \exp\left(-k^2 r_s^2\right)\right) \\
\mathbf{f}^s(\mathbf{r}) &= -\frac{Gm\mathbf{r}}{r^3}\left(\mathrm{erfc}\left(\frac{r}{2r_s}\right) + \frac{r}{r_s\sqrt{\pi}} \exp\left(-\frac{r^2}{4r_s^2}\right)\right) \qquad (1)
\end{aligned}
$$

The choice of Gaussian for splitting force is the optimum one from the point of view of errors in force.


Short range force
Long range force
Total force
force
r
100
10
1
0.1
0.01
10^{-3}
0.1
1
10

The TreePM method

Error in force is minimised for a range of particle distributions by choosing $r_s = L$, where L is the grid length, and by choosing $\theta_c = 0.5$. The short range force is summed up to $5\, r_s$.

For this configuration, error is below 1% for more than 99% of particles for an unclustered distribution and the situation is better for clustered distributions.

The TreePM code replaces the particle-particle sum with a tree, and hence does not slow down at late times. Long range force is modified to suppress anisotropies. Two body collisions are a problem if softening length is chosen to be small.

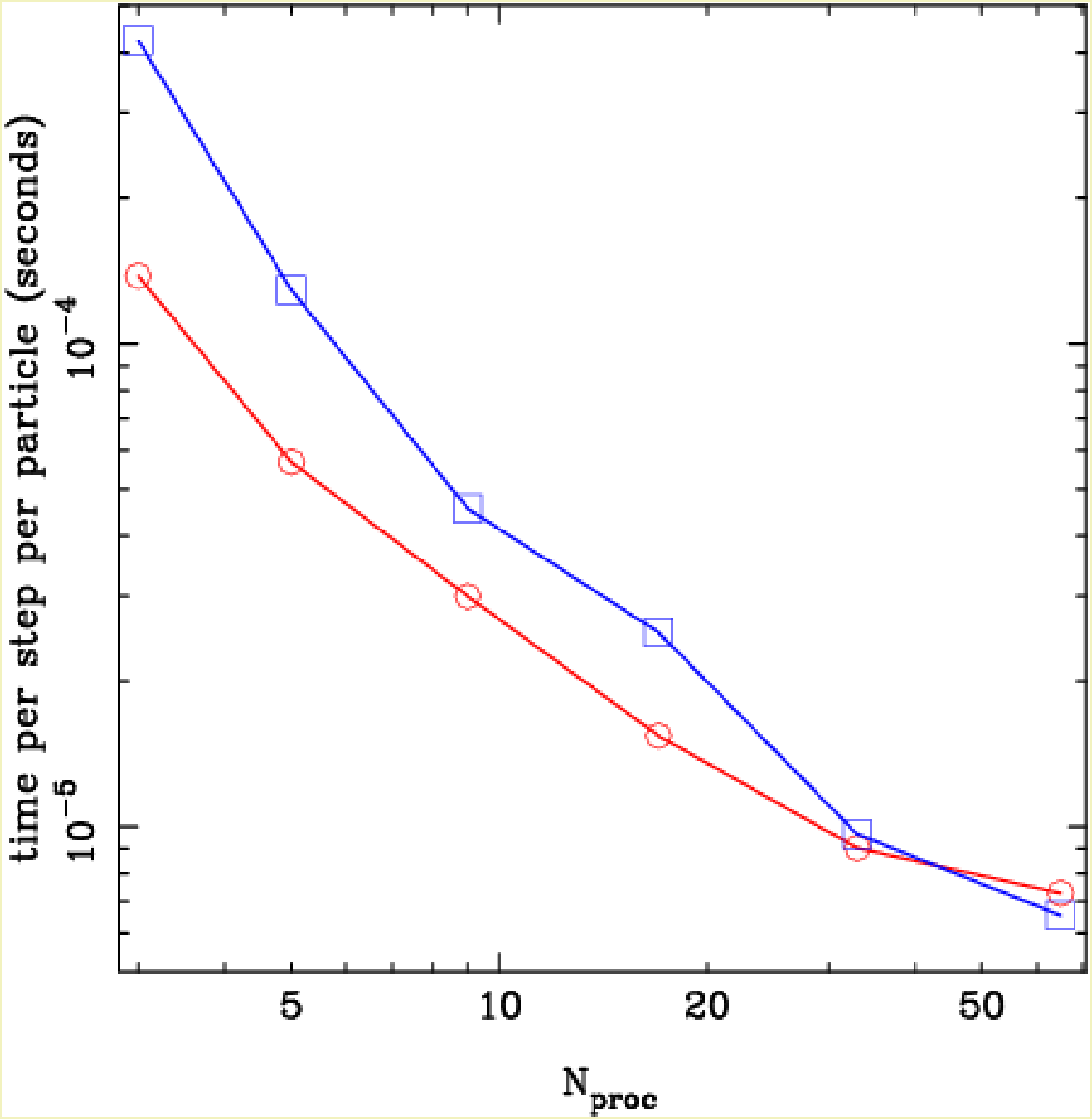
An implementation of TreePM is available as a part of Gadget-2.

The Parallel TreePM method

Uses functional (PM - Tree) decomposition as well as domain decomposition (tree) for a load balanced code.

Scaling has been checked for up to 65 processors.

One time step takes slightly less than two minutes for a 256^3 simulation on 65 processors. (2.4 GHz Xeon processors, connected with SCI interconnect in a 2-Torus topology. See http://cluster.mri.ernet.in/ for more details. The 42 node, 84 processer cluster cost USD 150000.)


time per step per particle (seconds)
10^{-4}
10^{-5}
5
10
20
50
N_{proc}

The Adaptive TreePM method

The TreePM method addresses all the other problems except for two body scattering. We can avoid this problem by making the softening length a function of the local number density of particles.

$$\epsilon = \left(\frac{n_c}{n\left(\mathbf{r}\right)} \right)^{1/3} \qquad ; \qquad n_c \simeq 10 \tag{2}$$

The dynamical system can no longer be described by the usual Hamiltonian.

Neither energy, nor momentum is conserved.

The Adaptive TreePM method

We can recover momentum conservation by using symmetrised force:

$$f_{ij} = \frac{1}{2}\left(f_{ij} + f_{ji}\right) \tag{3}$$

Symmetrisation is needed only if the separation of particles is smaller than the large of the two softening lengths. We use a form of softening such that $f \sim 1/r^2$ for $r \geq \epsilon$.

The largest value of softening length within a cell is also stored along with the usual quantities. The cell opening criteria is modified:

$$r\,\theta_c \;\geq\; L_{cell} \qquad \Rightarrow \qquad (r - \epsilon_{max})\;\theta_c \;\geq\; L_{cell} \tag{4}$$

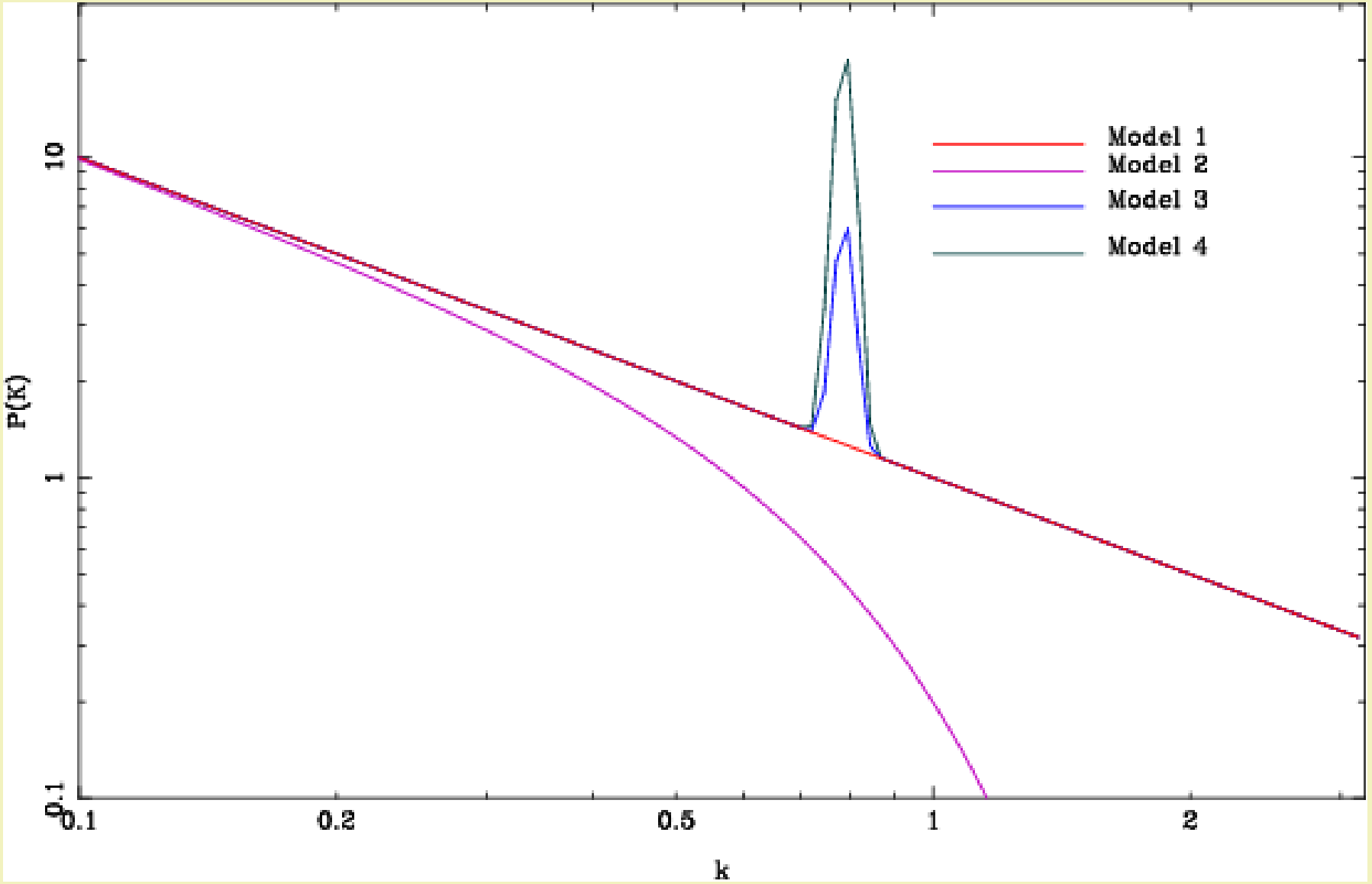
It helps to put an upper bound on the softening length, $\epsilon \leq L$ where L is the average inter-particle separation.

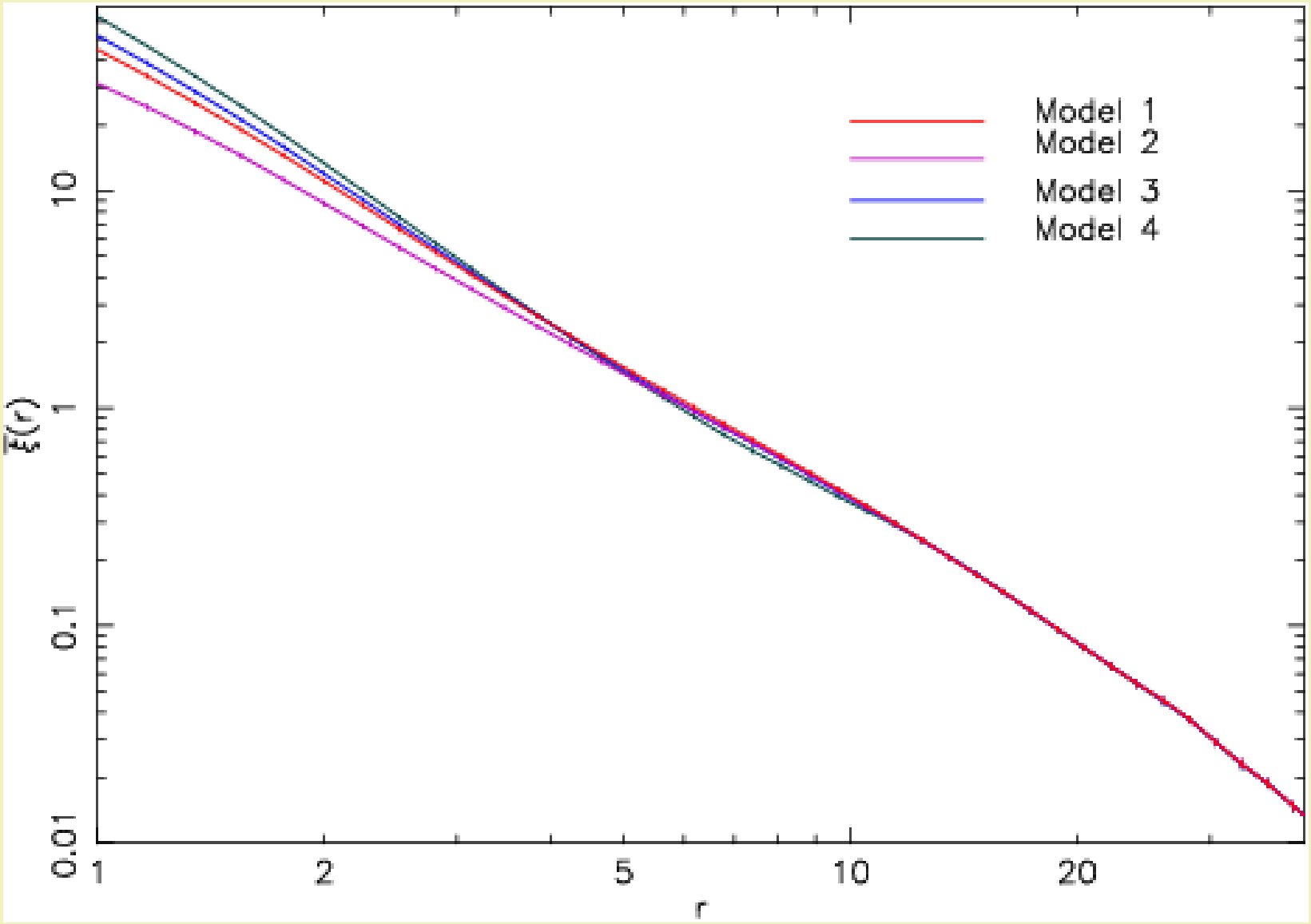
A lower bound to softening length?

It is known that changes in power spectrum at small, non-linear scales do not affect evolution of much larger scales, particularly if the larger scales are also non-linear. Therefore we do not introduce any significant errors by using a length resolution that is smaller than the mass resolution.

Given that we are using a finite number of particles, we cannot claim to have an infinite or an arbitrarily large dynamic range in number density.

It is important to prevent ϵ from taking on arbitrarily small values, as this will slow down the code.


Model 1
Model 2
Model 3
Model 4
P(K)
10
1
0.1
0.1
0.2
0.5
1
2
k


Model 1
Model 2
Model 3
Model 4
$\bar{\xi}(r)$
10
1
0.1
0.01
1
2
5
10
20
r

Computing Force

The long range force is computed as in the TreePM method, using the PM method but with a modified kernel.

Short range force at scales larger than the local value of ϵ is computed as in the TreePM method, using a global tree.

At smaller scales, a pair-wise sum is required for computing the symmetrised force.

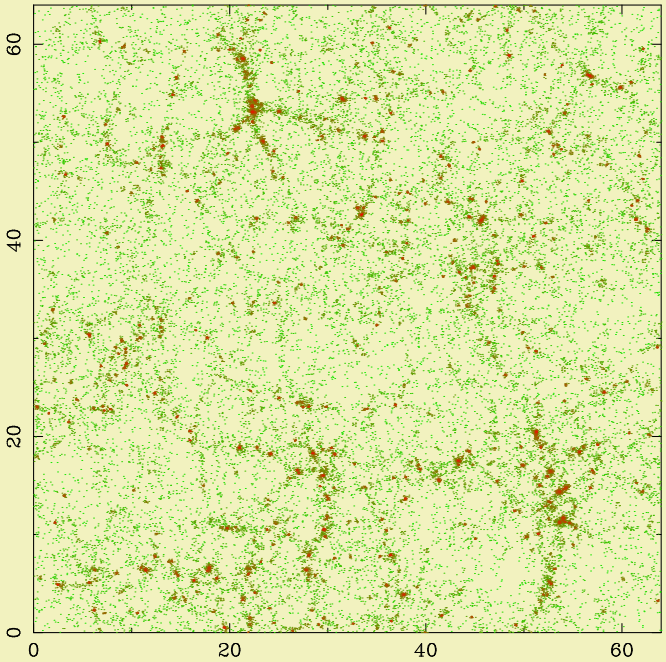
As we use Barnes' optimisation of computing interaction list for groups of particles, this last step does not introduce a large overhead.

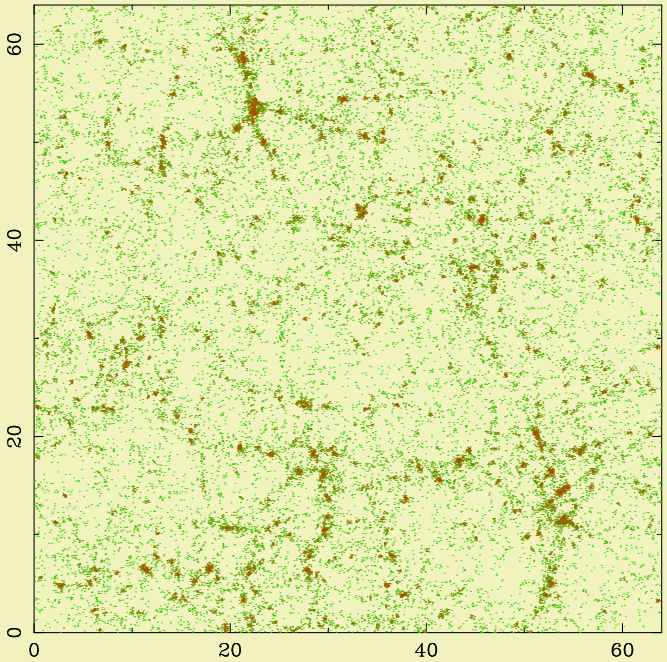
Evolving Trajectories

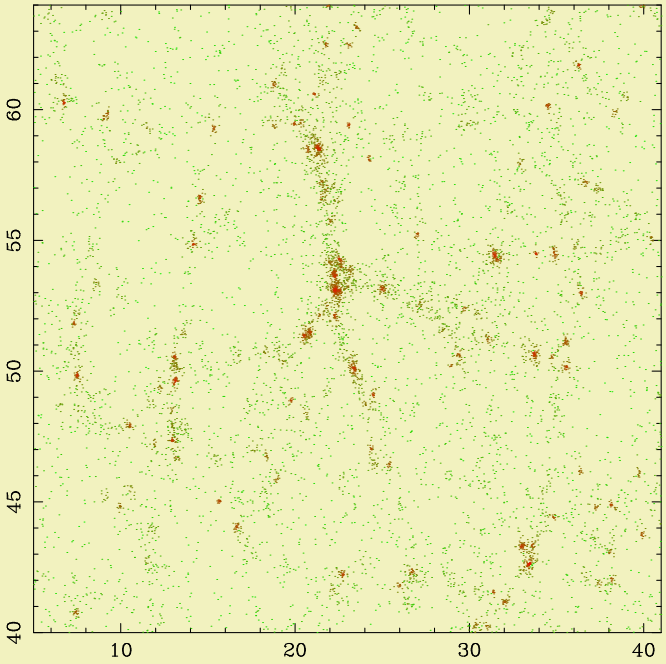
A range of softening lengths in the system leads to very different requirements for the time step.

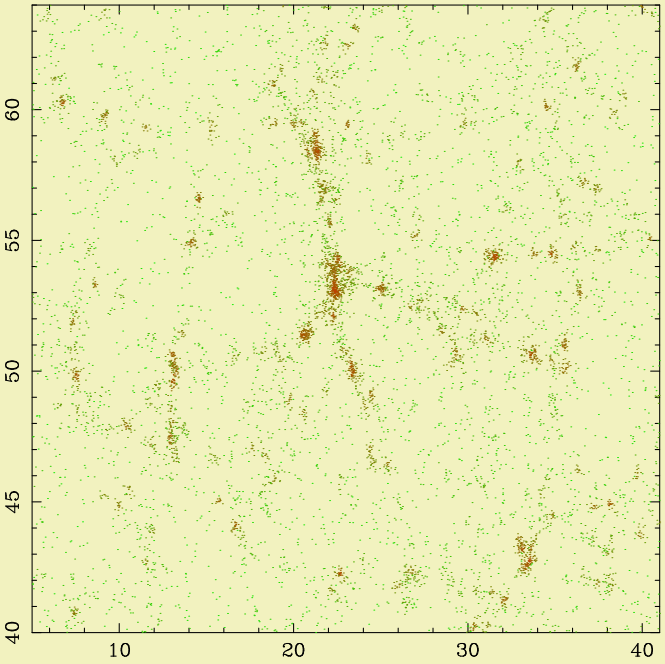
It is essential to use different time steps for different particles, using the smallest required time step for each particle slows down the evolution considerably.

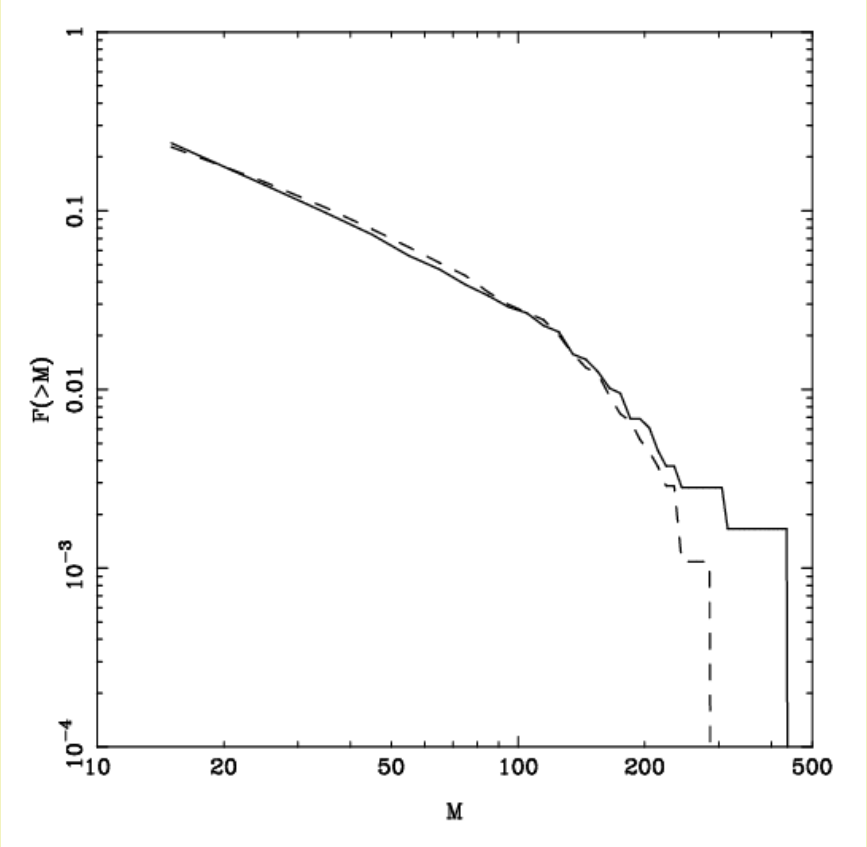
At present we have an unoptimised sequential version in place. We expect to release an optimised, parallel version by end of this year.

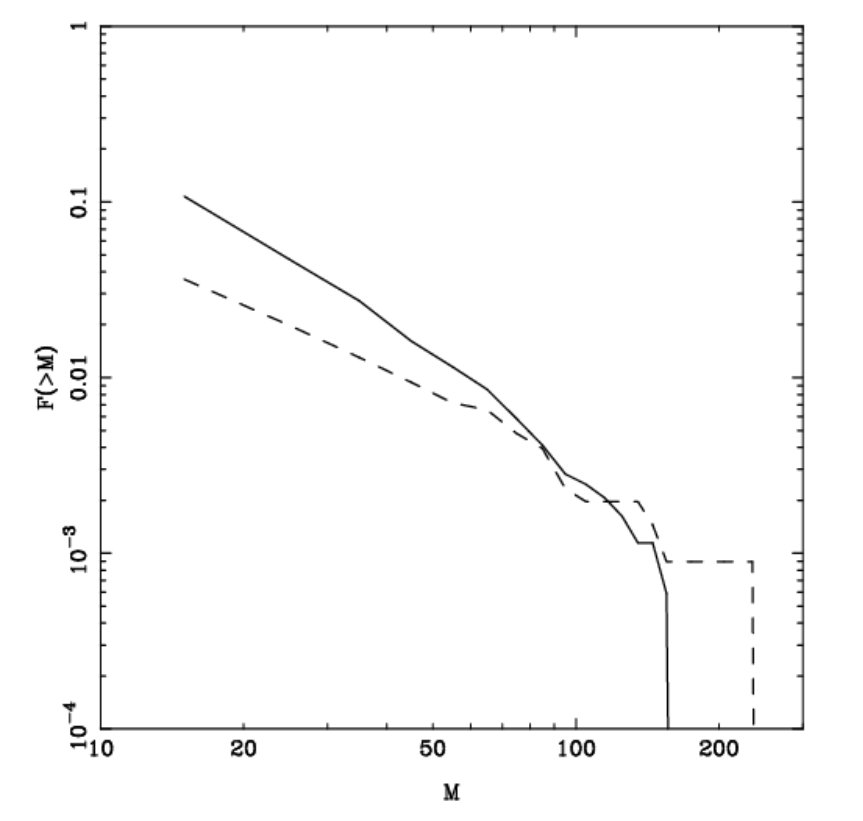

0
20
40
60
0
20
40
60

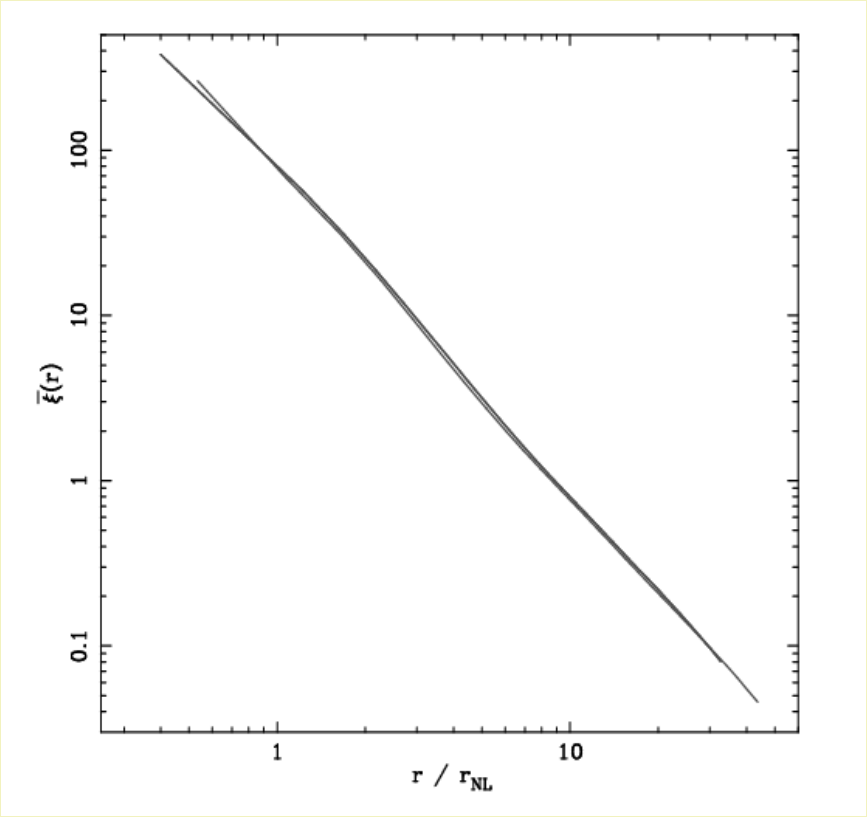

0
20
40
60
0
20
40
60

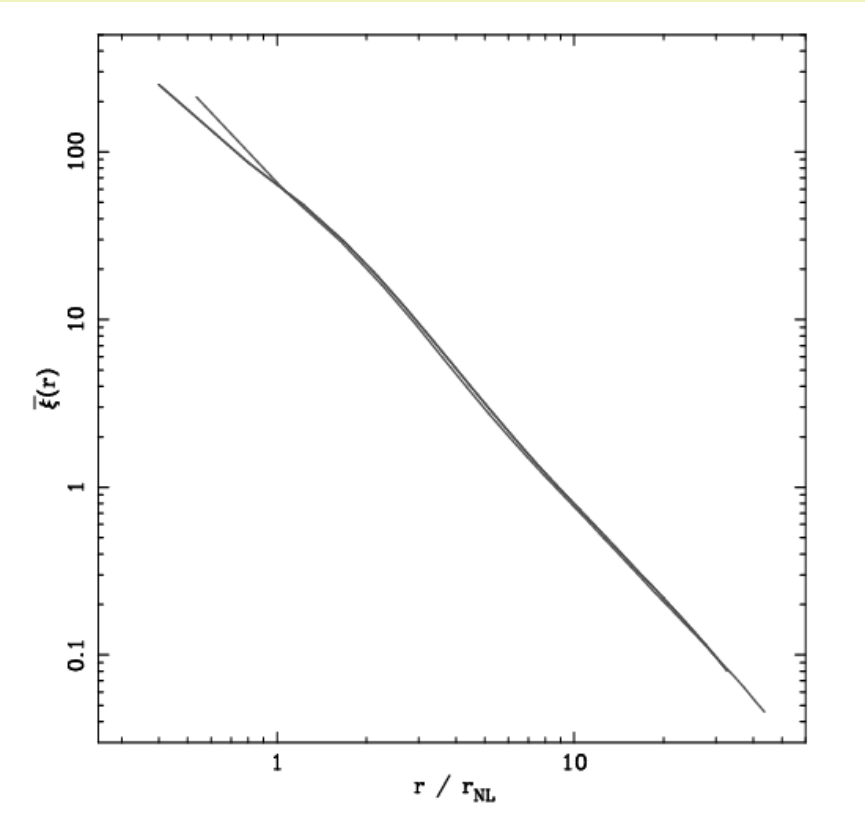

40
45
50
55
60
10
20
30
40

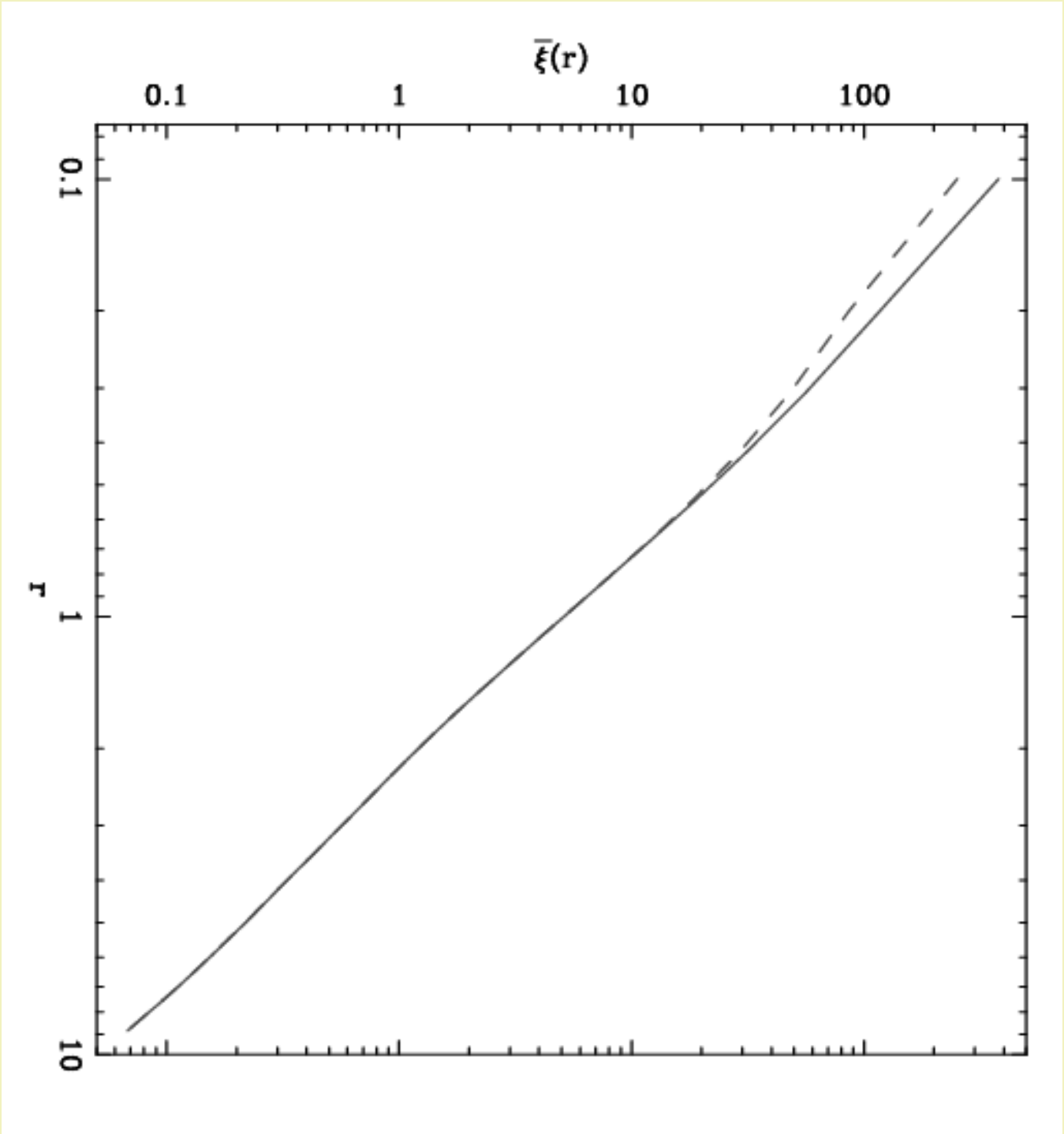

40
45
50
55
60
10
20
30
40


F(>M)
1
0.1
0.01
10^{-3}
10^{-4}
10
20
50
100
200
500
M


F(>M)
1
0.1
0.01
10^{-3}
10^{-4}
10
20
50
100
200
M


$\bar{\xi}(r)$
100
10
1
0.1
1
10
r / r_{NL}


$\bar{\xi}(r)$
100
10
1
0.1
1
10
r / r_{NL}


$\bar{\xi}(r)$
0.1
1
10
100
r
0.1
1
10

Conclusions

Preliminary studies indicate that the Adaptive TreePM is a significant improvement over the standard TreePM method.

It will be very useful for studies of non-linear clustering and dark matter haloes.

Collaborators:

Jayanti Prasad: Effect of small scales fluctuations on large scales.

Suryadeep Ray & Nishikanta Khandai: Adaptive TreePM.