

Lecture 3

Simulations of Nonstandard Models

Baojiu Li (ICC, Durham)

Mexican Numerical Simulations School (03-06 Oct 2016)

Why use AMR for Nonstandard Models

- We have seen from Lecture 2 that the particle-mesh (PM) method, augmented by the adaptive mesh refinement (AMR) technique, can be a powerful tool to solve partial differential equations in boundary-value problems
- A particular advantage of the relaxation method is that **it applies to linear as well as nonlinear partial differential equations**. The latter are common in nonstandard cosmological models.
- Particle-based methods, in contrast, are more difficult to apply since there is no universal analytical force law like in standard gravity

Why use AMR for Nonstandard Models

- Recall there are two main reasons for developing the AMR technique in N-body simulations, and they are strengthened for nonstandard models:
- AMR allows us to achieve high mesh resolution (and therefore high force resolution) in high density regions, which are where we need high accuracy in solving the nonstandard model equations.
- AMR allows us to improve the efficiency of the code without sacrificing accuracy in low density regions. This is critically important for nonstandard models, as the relaxation method is usually slow when applied to highly nonlinear differential equations.

This Lecture

- How the AMR technique is applied to solve the equations in some most widely studied nonstandard cosmological models. Two cases will be considered:
- The case of a scalar field with self-interaction prescribed by an interacting potential $V(\varphi)$
- The case of a scalar field with a derivative type self-interaction
- Challenges facing simulations of nonstandard models, and some approximation methods that can be employed to tackle these challenges

The Quasi-Static Approximation

- We have seen that the relaxation method is good for solving boundary value problems. But the scalar field equation is not a boundary value problem.

background perturbation

$$\varphi(\mathbf{x}, t) = \bar{\varphi}(t) + \delta\varphi(\mathbf{x}, t)$$

$$\square\varphi + \frac{dV(\varphi)}{d\varphi} = C(\varphi)\rho_m$$

$$\ddot{\varphi} + 3\frac{\dot{a}}{a}\dot{\varphi} + \nabla^2\varphi + \frac{dV(\varphi)}{d\varphi} = C(\varphi)\rho_m$$

$$\ddot{\bar{\varphi}} + 3\frac{\dot{a}}{a}\dot{\bar{\varphi}} + \frac{dV(\bar{\varphi})}{d\varphi} = C(\bar{\varphi})\bar{\rho}_m$$

$$\ddot{\delta\varphi} + 3\frac{\dot{a}}{a}\dot{\delta\varphi} + \nabla^2\delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m$$

the whole equation is a wave equation;
which is an initial value problem

the purple part has no time derivatives,
and is a boundary value problem

The Quasi-Static Approximation

- In practice, we often employ the so-called quasi-static approximation where we neglect the time derivative terms.
- The name ‘quasi-static’ means that the time derivatives are much smaller than spatial derivatives, so that time evolution can be neglected.
- This is a good approximation because typically in simulations each time step is quite small, during which the scalar field is not expected to evolve significantly.

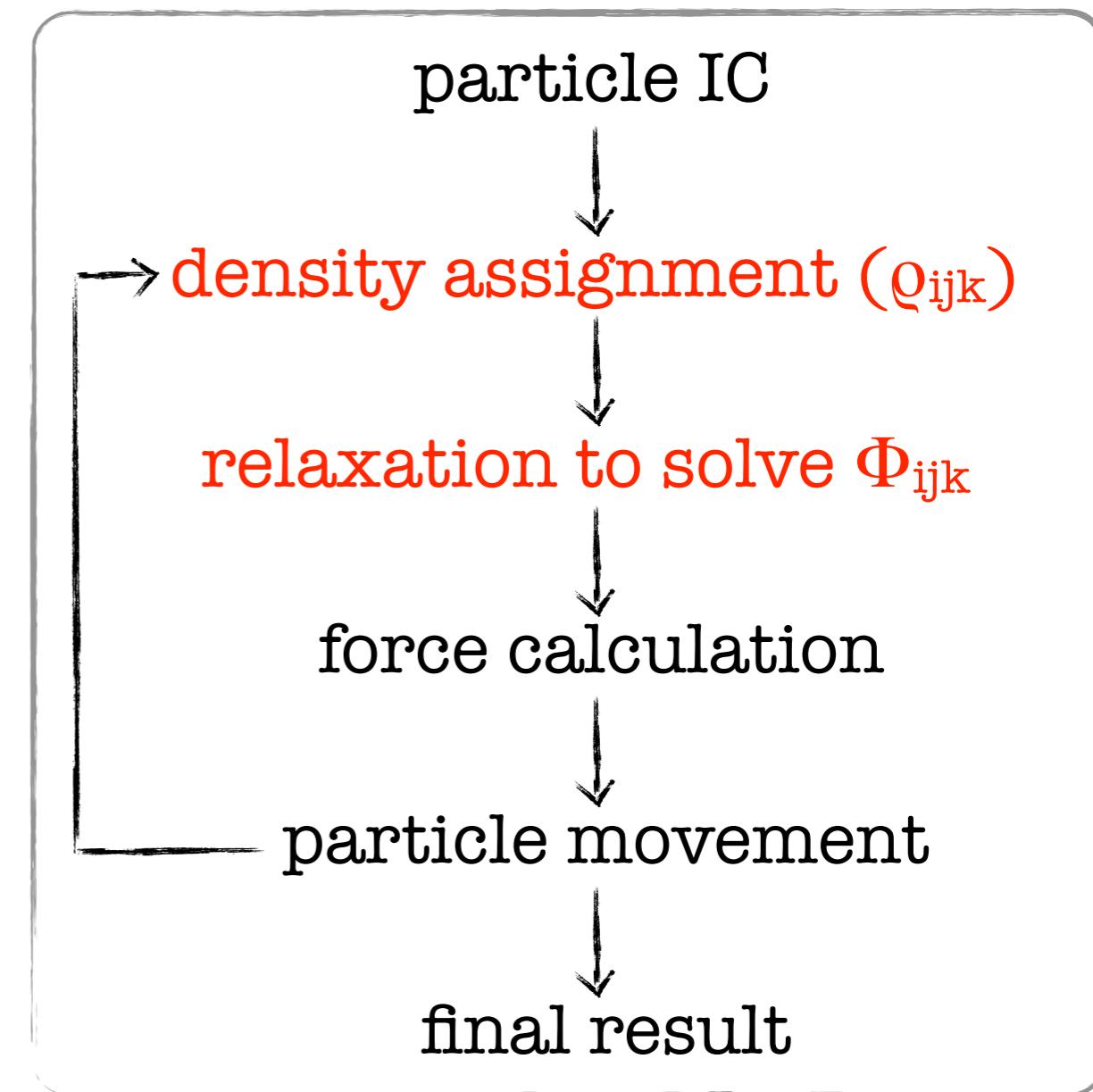
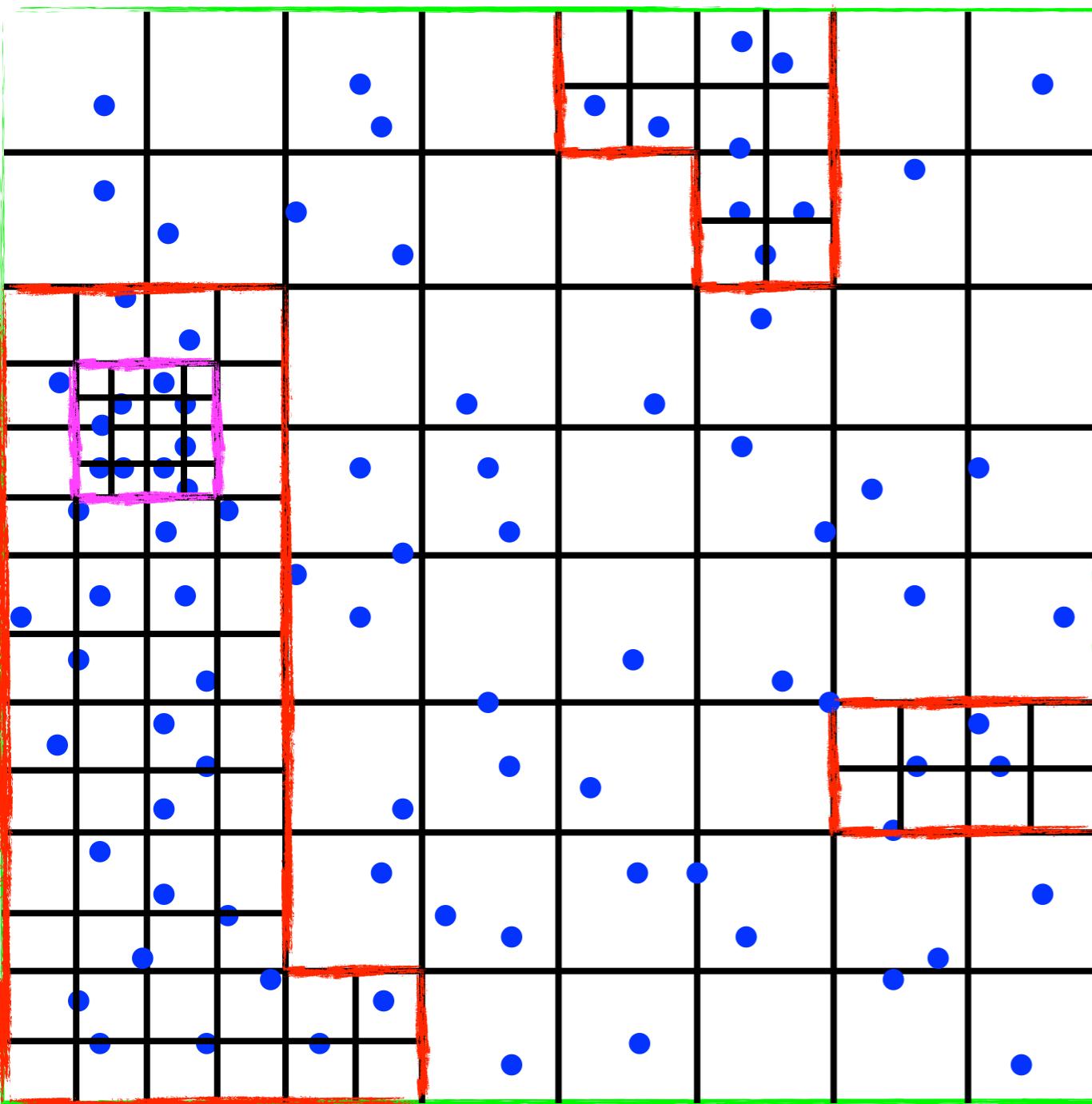
$$\ddot{\delta\varphi} + 3\frac{\dot{a}}{a}\dot{\delta\varphi} + \nabla^2\delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m$$

The Quasi-Static Approximation

- This makes the scalar field equation an elliptical partial differential equation which has only spatial derivative terms.
- The problem becomes a boundary value problem: namely we want solve the scalar field configuration as a function of space, $\varphi(\mathbf{x})$, given certain boundary conditions (BCs).
- The situation is the same as in standard simulations, with periodic BCs for simulation box boundaries, and fixed BCs interpolated from coarse level for refinement boundaries.

$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m$$

A Summary of AMR Algorithm



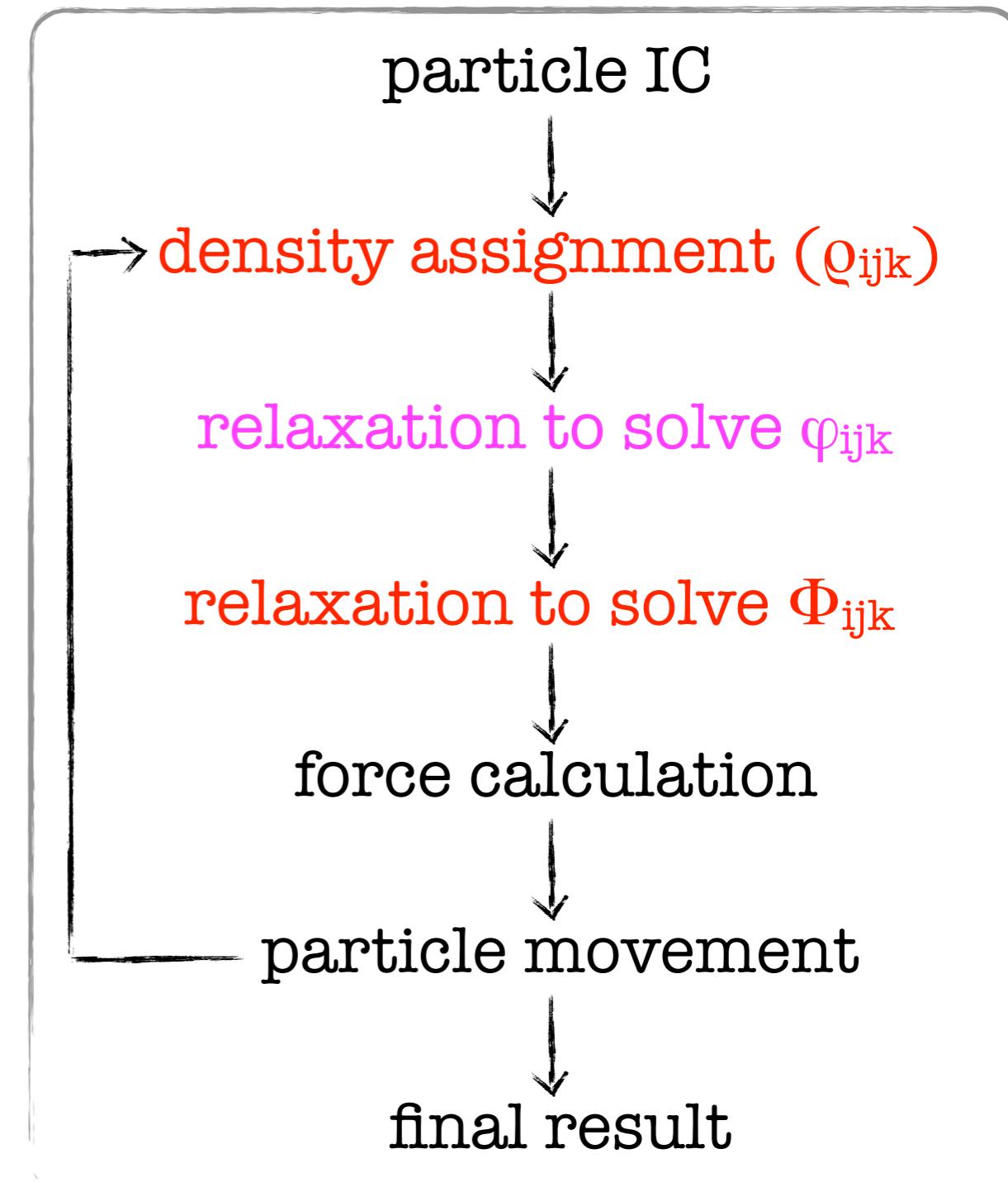
Where to Solve the Scalar Field Equation?

$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = 4\pi G [C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m]$$
$$\nabla^2 \Phi = 4\pi G \delta\rho_m + \beta(\varphi) \nabla^2 \delta\varphi$$

The modified Poisson equation depends on the scalar field, so the latter must be solved first.

The scalar field equation depends on the density field, so it can only be solved after density assignment.

The scalar field equation can be solved on the **same** mesh (AMR) structure as the Poisson equation!



Where to Solve the Scalar Field Equation?

$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = 4\pi G [C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m]$$
$$\nabla^2 \Phi = 4\pi G \delta\rho_m + \beta(\varphi) \nabla^2 \delta\varphi$$

relaxation to solve φ_{ijk}

As we have learned how to solve the Poisson equation using AMR, in this lecture we will only focus on things we need to know in order to solve the scalar field equation using (the same) AMR.

Where to Solve the Scalar Field Equation?

$$\begin{aligned}\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} &= 4\pi G [C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m] \\ \nabla^2 \Phi &= 4\pi G \delta\rho_m + \beta(\varphi) \nabla^2 \delta\varphi\end{aligned}$$

The multigrid method described in the last lecture is an efficient way to speed up relaxation convergence and works for nonstandard models.

Therefore, what we really focus on here is how to properly discretise the equations so that the multigrid method can be applied efficiently.

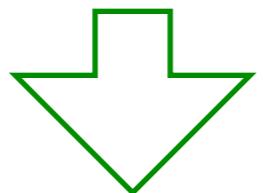
relaxation to solve φ_{ijk}

This is actually a nontrivial task. The experience is that, without doing this properly, the relaxation method often suffers from serious instability and divergence numerical issues.

Scalar Field with Self-interacting Potential

Scalar Field with Self-interacting Potential

$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = 4\pi G [C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m]$$
$$\nabla^2 \Phi = 4\pi G \delta\rho_m + \beta(\varphi) \nabla^2 \delta\varphi$$



get rid of scalar field derivatives in the Poisson equation, just for convenience

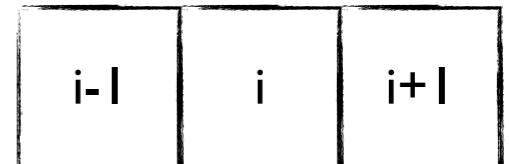
$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} = C(\varphi)\rho_m - C(\bar{\varphi})\bar{\rho}_m$$
$$\nabla^2 \Phi = 4\pi G[1 + \beta(\varphi)C(\varphi)]\delta\rho_m - \beta(\varphi) \left[\frac{dV(\varphi)}{d\varphi} - \frac{dV(\bar{\varphi})}{d\varphi} \right]$$

Usually, these background terms either are unimportant or almost cancel themselves, so that we can neglect them from now on

Discretisation of Equation

$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} = C(\varphi) \delta\rho_m$$

$$\nabla^2 \Phi = 4\pi G [1 + \beta(\varphi)C(\varphi)] \delta\rho_m - \beta(\varphi) \frac{dV(\varphi)}{d\varphi}$$



$$\begin{aligned}
 \frac{\partial \varphi}{\partial x} &\rightarrow \frac{1}{2h} (\varphi_{i+1,j,k} - \varphi_{i-1,j,k}) \quad \text{or} \quad \frac{1}{h} (\varphi_{i+1,j,k} - \varphi_{i,j,k}) \quad \text{or} \quad \frac{1}{h} (\varphi_{i,j,k} - \varphi_{i-1,j,k}) \\
 \frac{\partial^2 \varphi}{\partial x^2} &\rightarrow \frac{1}{h} \left[\left(\frac{\partial \varphi}{\partial x} \right)_{i+1,j,k} - \left(\frac{\partial \varphi}{\partial x} \right)_{i,j,k} \right] \\
 &\rightarrow \frac{1}{h} \left[\frac{1}{h} (\varphi_{i+1,j,k} - \varphi_{i,j,k}) - \frac{1}{h} (\varphi_{i,j,k} - \varphi_{i-1,j,k}) \right] \\
 &= \frac{1}{h^2} (\varphi_{i+1,j,k} - 2\varphi_{i,j,k} + \varphi_{i-1,j,k})
 \end{aligned}$$

Discretisation of Equation

$$\nabla^2 \delta\varphi + \frac{dV(\varphi)}{d\varphi} = C(\varphi) \delta\rho_m$$

$$\nabla^2 \Phi = 4\pi G [1 + \beta(\varphi)C(\varphi)] \delta\rho_m - \beta(\varphi) \frac{dV(\varphi)}{d\varphi}$$

$$\mathcal{L}(\varphi_{i,j,k}) = 0$$

$$\begin{aligned}\mathcal{L}(\varphi_{i,j,k}) &\equiv \frac{1}{h^2} [\varphi_{i+1,j,k} + \varphi_{i-1,j,k} + \varphi_{i,j+1,k} + \varphi_{i,j-1,k} + \varphi_{i,j,k+1} + \varphi_{i,j,k-1} - 6\varphi_{i,j,k}] \\ &\quad + V_{,\varphi}(\varphi_{i,j,k}) - C(\varphi_{i,j,k}) a \Omega_m (\tilde{\rho}_{i,j,k} - 1)\end{aligned}$$

$L(\varphi_{ijk})$ nonlinear function of $\varphi_{ijk}!$

Relaxation

$$\varphi_{i,j,k}^{\text{new}} = \varphi_{i,j,k}^{\text{old}} - \frac{\mathcal{L}(\varphi_{i,j,k}^{\text{old}})}{\partial \mathcal{L}(\varphi_{i,j,k}^{\text{old}}) / \partial \varphi_{i,j,k}}$$

$$\mathcal{L}(\varphi_{i,j,k}) = 0$$

$$\begin{aligned}\mathcal{L}(\varphi_{i,j,k}) &\equiv \frac{1}{h^2} [\varphi_{i+1,j,k} + \varphi_{i-1,j,k} + \varphi_{i,j+1,k} + \varphi_{i,j-1,k} + \varphi_{i,j,k+1} + \varphi_{i,j,k-1} - 6\varphi_{i,j,k}] \\ &\quad + V_{,\varphi}(\varphi_{i,j,k}) - C(\varphi_{i,j,k}) a \Omega_m (\tilde{\rho}_{i,j,k} - 1)\end{aligned}$$

$L(\varphi_{ijk})$ nonlinear function of $\varphi_{ijk}!$

Change of Variables

$$\varphi_{i,j,k}^{\text{new}} = \varphi_{i,j,k}^{\text{old}} - \frac{\mathcal{L}(\varphi_{i,j,k}^{\text{old}})}{\partial \mathcal{L}(\varphi_{i,j,k}^{\text{old}}) / \partial \varphi_{i,j,k}}$$

In some cases, the value of φ_{ijk} is tiny, but it cannot cross zero. For example, for $\varphi_{ijk} < 0$ functions of φ_{ijk} may not be physically defined.

However, the term in red box above can be large numerically, which will mean that, even if the old φ_{ijk} is positive (therefore physical), the new φ_{ijk} can be negative and therefore unphysical.

This is a pure numerical issue as $\varphi_{ijk} < 0$ never happens physically. In numerics this can be avoided by defining a new variable $u = \log(\varphi_{ijk})$.

Change of Variables

$$\frac{\partial \varphi}{\partial x} = \frac{\partial e^u}{\partial x} = e^u \frac{\partial u}{\partial x}$$

$$\frac{\partial^2 \varphi}{\partial x^2} \rightarrow \frac{1}{h} \left[\left(e^u \frac{\partial u}{\partial x} \right)_{i+1,j,k} - \left(e^u \frac{\partial u}{\partial x} \right)_{i,j,k} \right]$$

$$\rightarrow \frac{1}{h} \left[(e^u)_{i+\frac{1}{2},j,k} \left(\frac{\partial u}{\partial x} \right)_{i+1,j,k} - (e^u)_{i-\frac{1}{2},j,k} \left(\frac{\partial u}{\partial x} \right)_{i,j,k} \right]$$

$$= \frac{1}{h^2} \left[(e^u)_{i+\frac{1}{2},j,k} u_{i+1,j,k} - \left((e^u)_{i+\frac{1}{2},j,k} + (e^u)_{i-\frac{1}{2},j,k} \right) u_{i,j,k} + (e^u)_{i-\frac{1}{2},j,k} u_{i-1,j,k} \right]$$

$$(e^u)_{i+\frac{1}{2},j,k} \equiv \frac{1}{2} [e^{u_{i,j,k}} + e^{u_{i+1,j,k}}]$$

$$(e^u)_{i-\frac{1}{2},j,k} \equiv \frac{1}{2} [e^{u_{i,j,k}} + e^{u_{i-1,j,k}}]$$



Change of Variables

$$\begin{aligned}\mathcal{L}(u_{i,j,k}) &= 0 \\ \mathcal{L}(u_{i,j,k}) &\equiv \frac{1}{h^2} \left[(e^u)_{i+\frac{1}{2},j,k} u_{i+1,j,k} - \left((e^u)_{i+\frac{1}{2},j,k} + (e^u)_{i-\frac{1}{2},j,k} \right) u_{i,j,k} + (e^u)_{i-\frac{1}{2},j,k} u_{i-1,j,k} \right] \\ &\quad + \frac{1}{h^2} \left[(e^u)_{i,j+\frac{1}{2},k} u_{i,j+1,k} - \left((e^u)_{i,j+\frac{1}{2},k} + (e^u)_{i,j-\frac{1}{2},k} \right) u_{i,j,k} + (e^u)_{i,j-\frac{1}{2},k} u_{i,j-1,k} \right] \\ &\quad + \frac{1}{h^2} \left[(e^u)_{i,j,k+\frac{1}{2}} u_{i,j,k+1} - \left((e^u)_{i,j,k+\frac{1}{2}} + (e^u)_{i,j,k-\frac{1}{2}} \right) u_{i,j,k} + (e^u)_{i,j,k-\frac{1}{2}} u_{i,j,k-1} \right] \\ &\quad + V_{,\varphi}(e^{u_{i,j,k}}) - C(e^{u_{i,j,k}}) a \Omega_m (\tilde{\rho}_{i,j,k} - 1)\end{aligned}$$

This guarantees that $\varphi_{ijk} = e^u$ is never negative and so any function of φ_{ijk} is always physical and properly defined.

The price is that we have added more nonlinearity to the equation, which will only make the relaxation method even slower.

Scalar Field with Derivative Couplings

Scalar Field with Derivative Couplings

- Field equations usually involve higher order powers of first- and second-order derivatives (note: **not** higher-order derivatives)

$$\nabla^2 \varphi + \begin{aligned} & (\nabla^2 \varphi)^2 \\ & \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \\ & (\nabla^2 \varphi)^3 \\ & \nabla^i \nabla^j \varphi \nabla_j \nabla^k \varphi \nabla_k \nabla_i \varphi \\ & \nabla^2 \varphi \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \\ & |\nabla \varphi|^2 \\ & \dots \end{aligned} = C(\varphi) \rho_m$$

Scalar Field with Derivative Couplings

- As a concrete example, we consider the equation for the well-known Dvali-Gabadze-Poratti (DGP) model. The methods described here have been shown to work for similar models.

$$\nabla^2 \varphi + \frac{r_c^2}{3\beta(a)a^2} \left[(\nabla^2 \varphi)^2 - \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \right] = \frac{1}{3\beta(a)} 8\pi G \delta \rho_m a^2$$

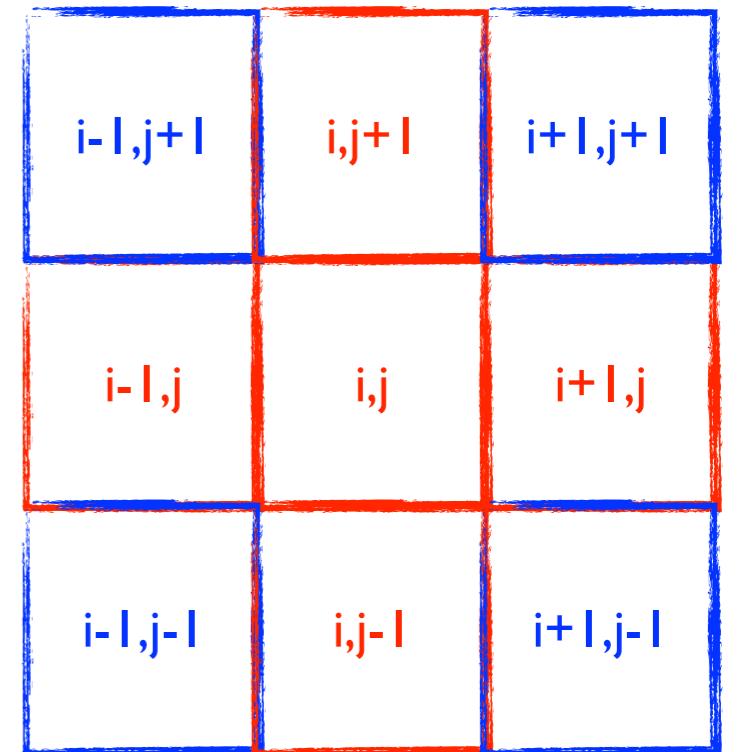
Discretisation of Equation

- The first step is again to discretise the equation, so that it can be solved on a mesh
- This is however more complicated than the previous case...

$$\nabla_i \nabla_i \varphi \rightarrow \frac{1}{h^2} (\varphi_{i+1,j,k} - 2\varphi_{i,j,k} + \varphi_{i-1,j,k})$$

$$\nabla_j \nabla_j \varphi \rightarrow \frac{1}{h^2} (\varphi_{i,j-1,k} - 2\varphi_{i,j,k} + \varphi_{i,j+1,k})$$

$$\nabla_i \nabla_j \varphi \rightarrow \frac{1}{4h^2} (\varphi_{i+1,j+1,k} - \varphi_{i+1,j-1,k} - \varphi_{i-1,j+1,k} + \varphi_{i-1,j-1,k})$$



Discretisation of Equation

$$\varphi_{i,j,k}^{\text{new}} = \varphi_{i,j,k}^{\text{old}} - \frac{\mathcal{L}(\varphi_{i,j,k}^{\text{old}})}{\partial \mathcal{L}(\varphi_{i,j,k}^{\text{old}}) / \partial \varphi_{i,j,k}}$$

$$\nabla^2 \varphi + \frac{r_c^2}{3\beta(a)a^2} \left[(\nabla^2 \varphi)^2 - \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \right] = \frac{1}{3\beta(a)} 8\pi G \delta \rho_m a^2$$

The relaxation algorithm depends on the derivative of the operator \mathcal{L} with respect to φ_{ijk} , so it is important to find which terms in the above equation depend on φ_{ijk} after discretisation.

Discretisation of Equation

$$\nabla^2 \varphi + \frac{r_c^2}{3\beta(a)a^2} \left[(\nabla^2 \varphi)^2 - \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \right] = \frac{1}{3\beta(a)} 8\pi G \delta \rho_m a^2$$

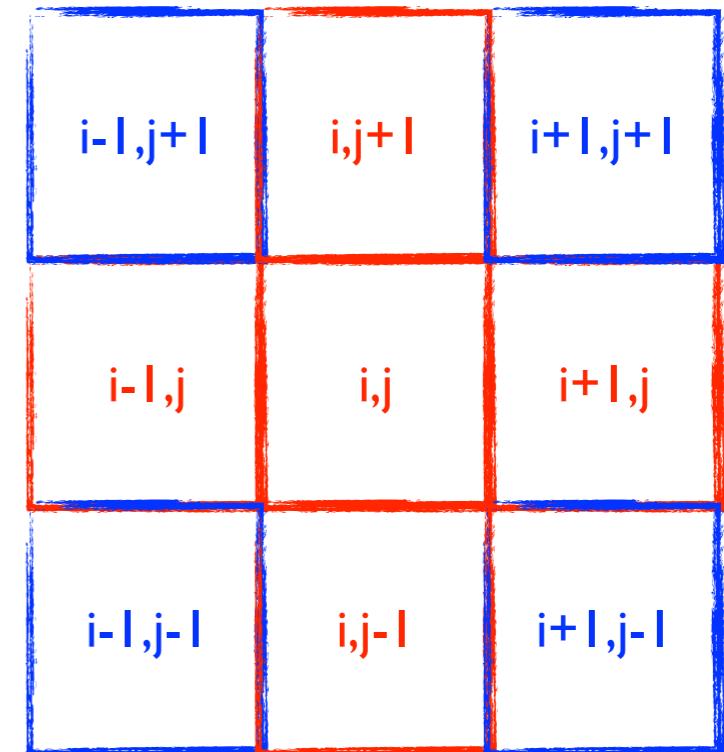
Here is a way to do it:

$$\nabla_i \nabla_j \varphi = \delta_{ij} \nabla^2 \varphi + \bar{\nabla}_i \bar{\nabla}_j \varphi$$

$$\bar{\nabla}_i \bar{\nabla}_j \varphi = \begin{cases} 0 & \text{if } i = j \\ \nabla_i \nabla_j \varphi & \text{if } i \neq j \end{cases}$$



$$(\nabla^2 \varphi)^2 - \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi = \frac{2}{3} (\nabla^2 \varphi)^2 - \bar{\nabla}^i \bar{\nabla}^j \varphi \bar{\nabla}_i \bar{\nabla}_j \varphi$$



Discretisation of Equation

$$\nabla^2 \varphi + \frac{r_c^2}{3\beta(a)a^2} \left[\frac{2}{3} (\nabla^2 \varphi)^2 - \bar{\nabla}^i \bar{\nabla}^j \varphi \bar{\nabla}_i \bar{\nabla}_j \varphi \right] = \frac{1}{3\beta(a)} 8\pi G \delta \rho_m a^2$$

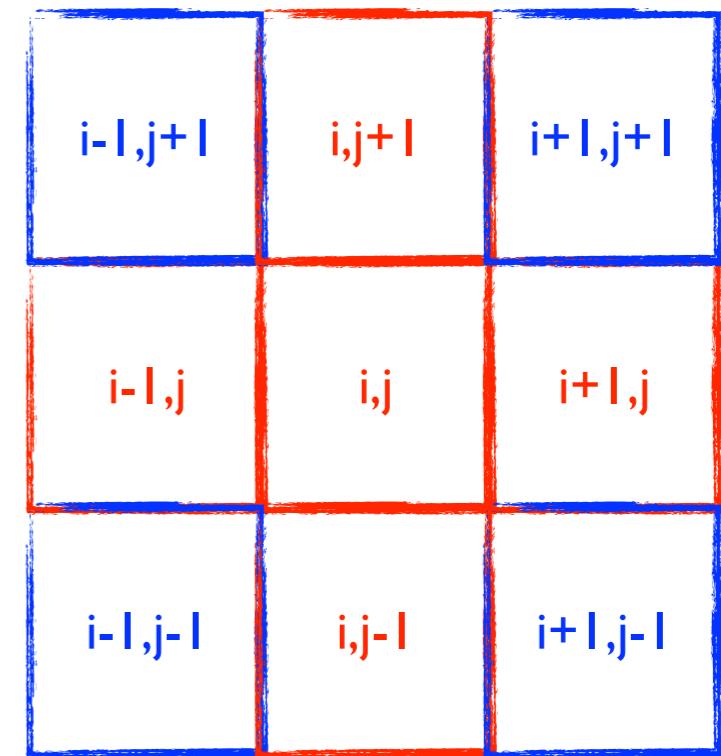
This can be considered as a quadratic (algebraic) equation for $\nabla^2 \varphi$, which can be solved first by hand. This gives

which branch to take?

$$\nabla^2 \varphi = \frac{3}{4} \left[-\alpha \left(\pm \frac{\alpha}{|\alpha|} \right) \left(\alpha^2 + \frac{8}{3} \Sigma \right)^{1/2} \right]$$

$$\alpha \equiv \frac{3\beta(a)a^4}{r_c^2}$$

$$\Sigma \equiv \bar{\nabla}^i \bar{\nabla}^j \varphi \bar{\nabla}_i \bar{\nabla}_j \varphi + \frac{\alpha}{\beta} \Omega_m a \delta \rho_m$$



Discretisation of Equation

$$\nabla^2 \varphi = \frac{3}{4} \left[-\alpha \pm \frac{\alpha}{|\alpha|} \left(\alpha^2 + \frac{8}{3} \Sigma \right)^{1/2} \right]$$

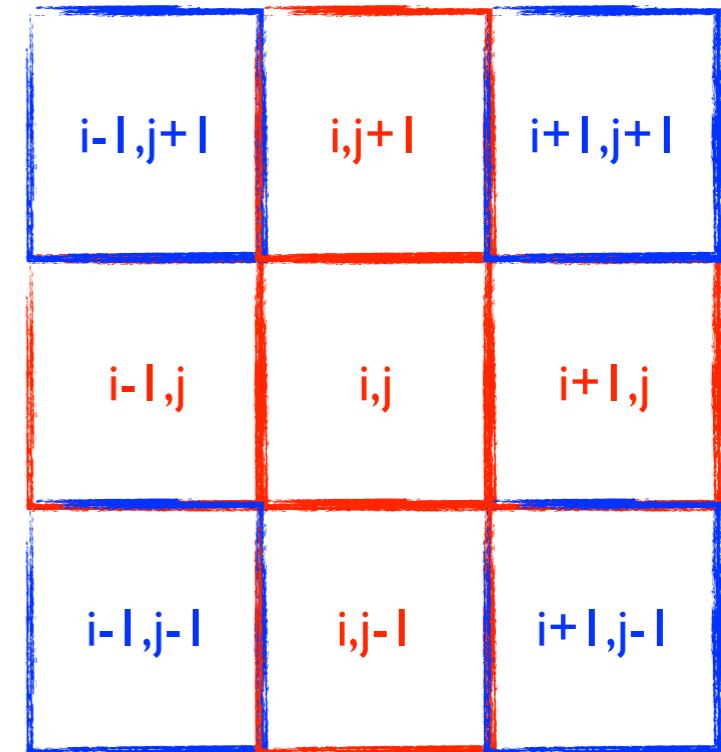
$$\alpha \equiv \frac{3\beta(a)a^4}{r_c^2}$$

$$\Sigma \equiv \bar{\nabla}^i \bar{\nabla}^j \varphi \bar{\nabla}_i \bar{\nabla}_j \varphi + \frac{\alpha}{\beta} \Omega_m a \delta \rho_m$$

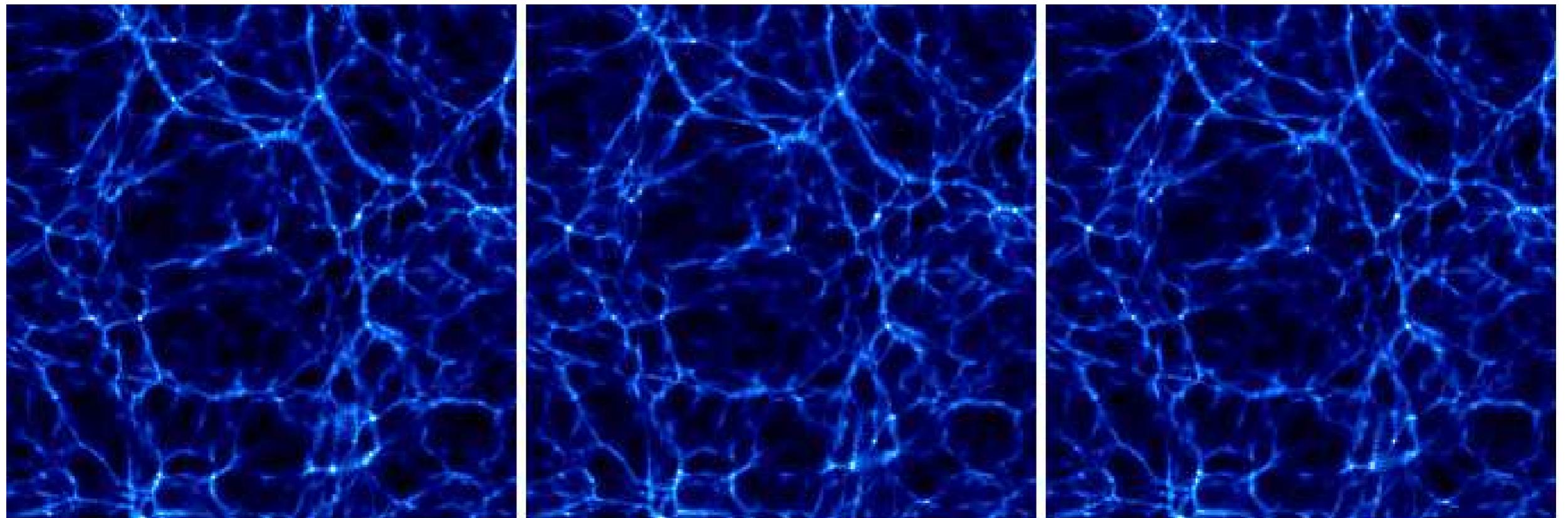
$$\mathcal{L}(\varphi_{i,j,k}) = 0$$

$$\mathcal{L}(\varphi_{i,j,k}) \equiv \frac{1}{h^2} [\varphi_{i+1,j,k} + \varphi_{i-1,j,k} + \varphi_{i,j+1,k} + \varphi_{i,j-1,k} + \varphi_{i,j,k+1} + \varphi_{i,j,k-1} - 6\varphi_{i,j,k}]$$

$$-\frac{3}{4} \left[-\alpha + \frac{\alpha}{|\alpha|} \left(\alpha^2 + \frac{8}{3} \Sigma_{i,j,k} \right)^{1/2} \right] \rightarrow \text{independent of } \varphi_{ijk}$$



Example of Simulation Output



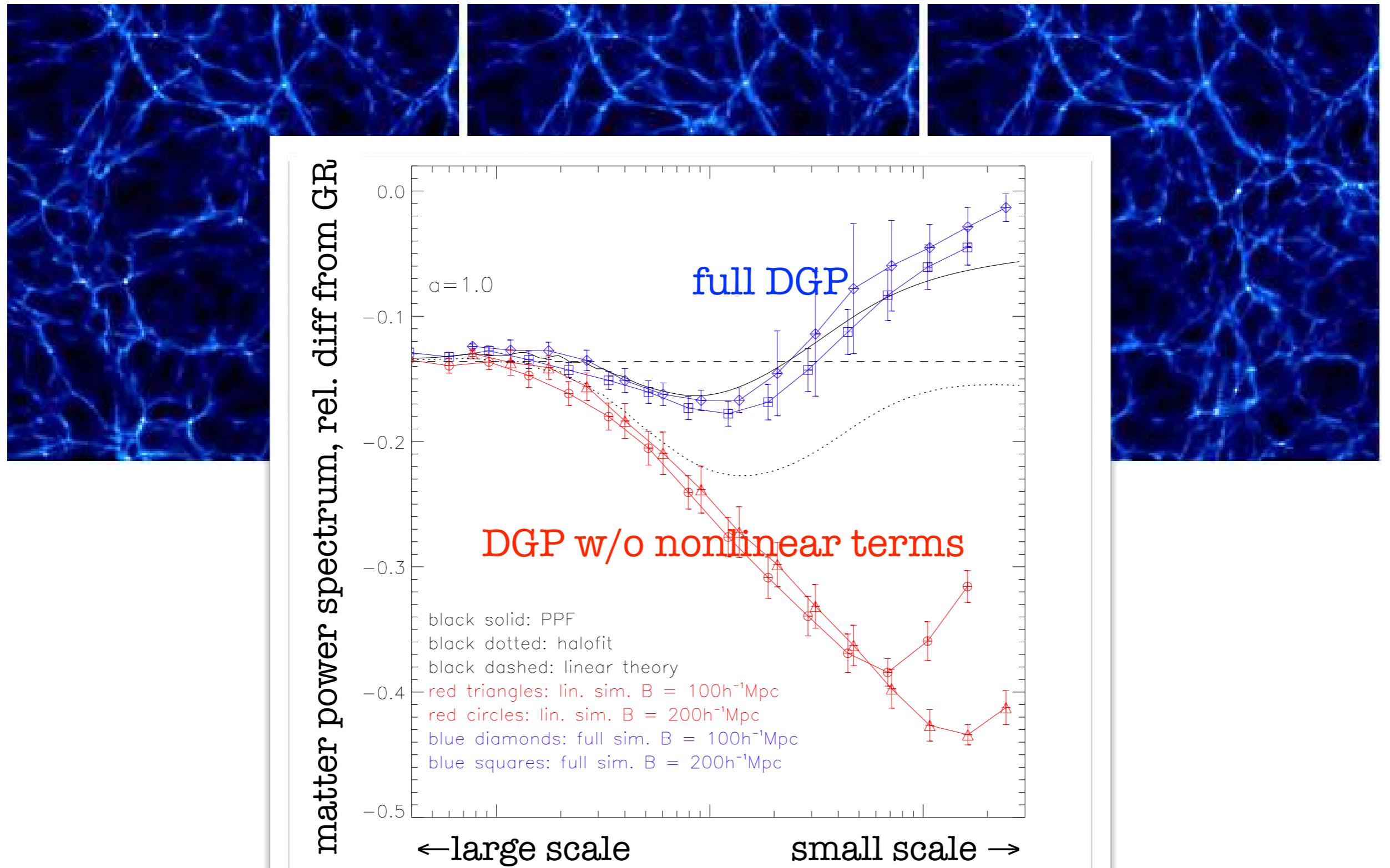
standard gravity

DGP w/o nonlinear terms

full DGP

$$\nabla^2 \varphi + \frac{r_c^2}{3\beta(a)a^2} \left[(\nabla^2 \varphi)^2 - \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \right] = \frac{1}{3\beta(a)} 8\pi G \delta \rho_m a^2$$

Example of Simulation Output



Beyond Quasi-static Approximation

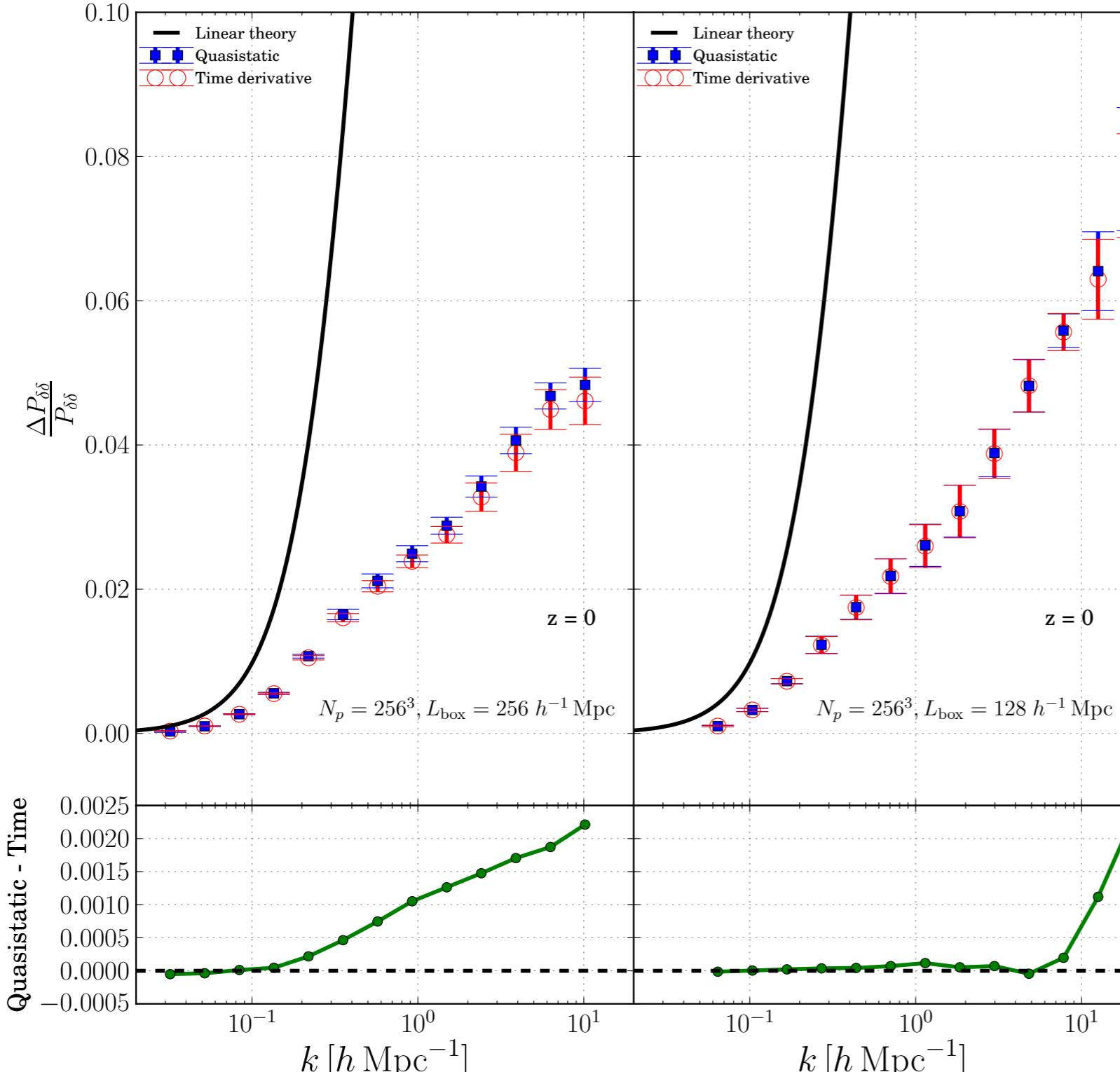
$$\ddot{\mathbf{x}} + 2\frac{\dot{a}}{a}\dot{\mathbf{x}} + \frac{1}{a^2}\nabla[\Phi + \beta(\varphi)\varphi] = 0$$
$$\ddot{\varphi} + 3\frac{\dot{a}}{a}\dot{\varphi} + \frac{1}{a^2}\nabla^2\varphi + \frac{dV(\varphi)}{d\varphi} + C(\varphi)\delta\rho_m = 0$$

Quasi-static approximation: neglecting the terms in the red box, so that the scalar field equation becomes an elliptical equation involving only spatial derivatives. Relaxation method applies.

Is this a good approximation?

For chameleon-type theories, the scalar field is trapped near $\varphi = 0$ throughout the cosmic evolution. So the time derivatives cannot be large. The approximation works well.

Beyond Quasi-static Approximation



model: $f(R)$ gravity

Filled circles: quasi-static approximation

Open circles: no quasi-static approximation

Relative difference
between quasi-static
approximation and no
quasi-static
approximation.

Beyond Quasi-static Approximation

$$\begin{aligned}\ddot{\mathbf{x}} + 2\frac{\dot{a}}{a}\dot{\mathbf{x}} + \frac{1}{a^2}\nabla[\Phi + \beta(\varphi)\varphi] &= 0 \\ \ddot{\varphi} + 3\frac{\dot{a}}{a}\dot{\varphi} + \frac{1}{a^2}\nabla^2\varphi + \frac{dV(\varphi)}{d\varphi} + C(\varphi)\delta\rho_m &= 0\end{aligned}$$

However, for some other types of theories, such as symmetron model, φ can evolve significantly in a shorter time scale, and these time derivatives are not necessarily negligible.

The equation then becomes a wave equation, which involves both time and spatial derivatives.

Relaxation does not apply, and direct integration is needed.

Beyond Quasi-static Approximation

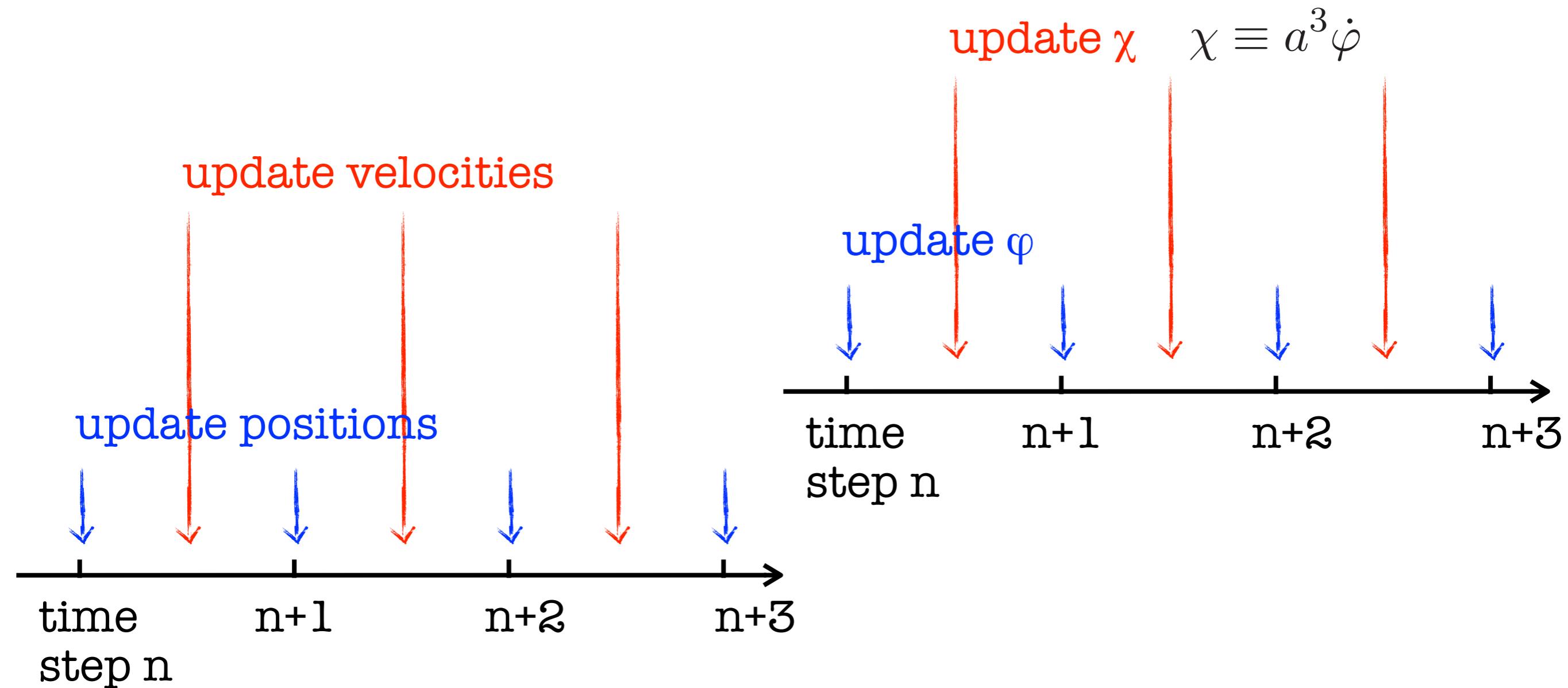
$$\begin{aligned}\dot{\mathbf{x}} &= \frac{1}{a^2} \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{1}{a} \vec{\nabla} [\Phi + \beta(\varphi)\varphi]\end{aligned}$$

$$\ddot{\mathbf{x}} + 2\frac{\dot{a}}{a}\dot{\mathbf{x}} + \frac{1}{a^2} \nabla [\Phi + \beta(\varphi)\varphi] = 0$$

$$\ddot{\varphi} + 3\frac{\dot{a}}{a}\dot{\varphi} + \frac{1}{a^2} \nabla^2 \varphi + \frac{dV(\varphi)}{d\varphi} + C(\varphi)\delta\rho_m = 0$$

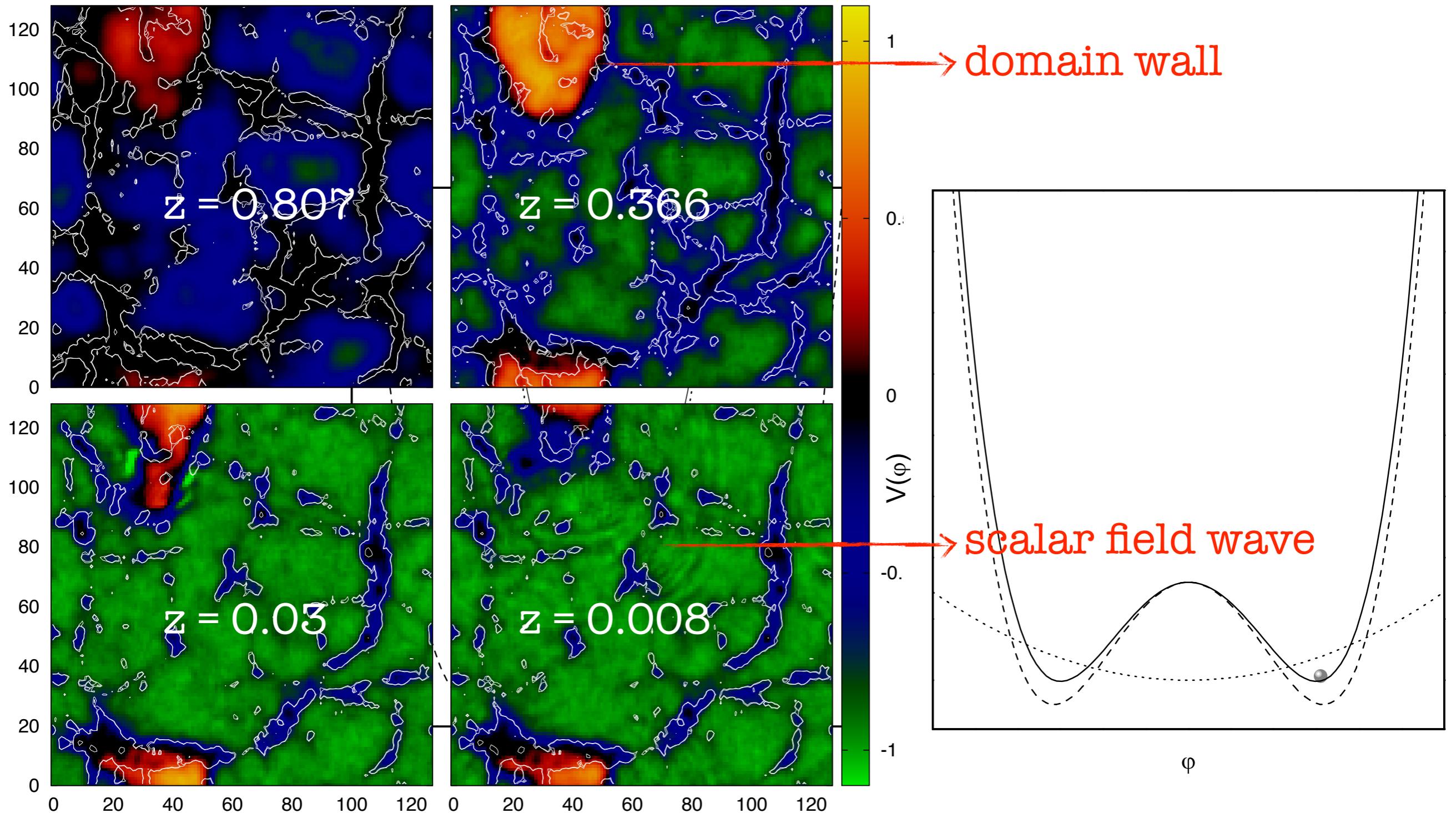
$$\begin{aligned}\dot{\varphi} &= \frac{1}{a^3} \chi \\ \dot{\chi} &= -a^3 \left[\frac{1}{a^2} \nabla^2 \varphi + \frac{dV(\varphi)}{d\varphi} + C(\varphi)\delta\rho_m \right]\end{aligned}$$

Beyond Quasi-static Approximation



Use the same leapfrog scheme to integrate: (1) the velocities and positions of particles, and (2) the scalar field and its first time derivative in cells

Beyond Quasi-static Approximation



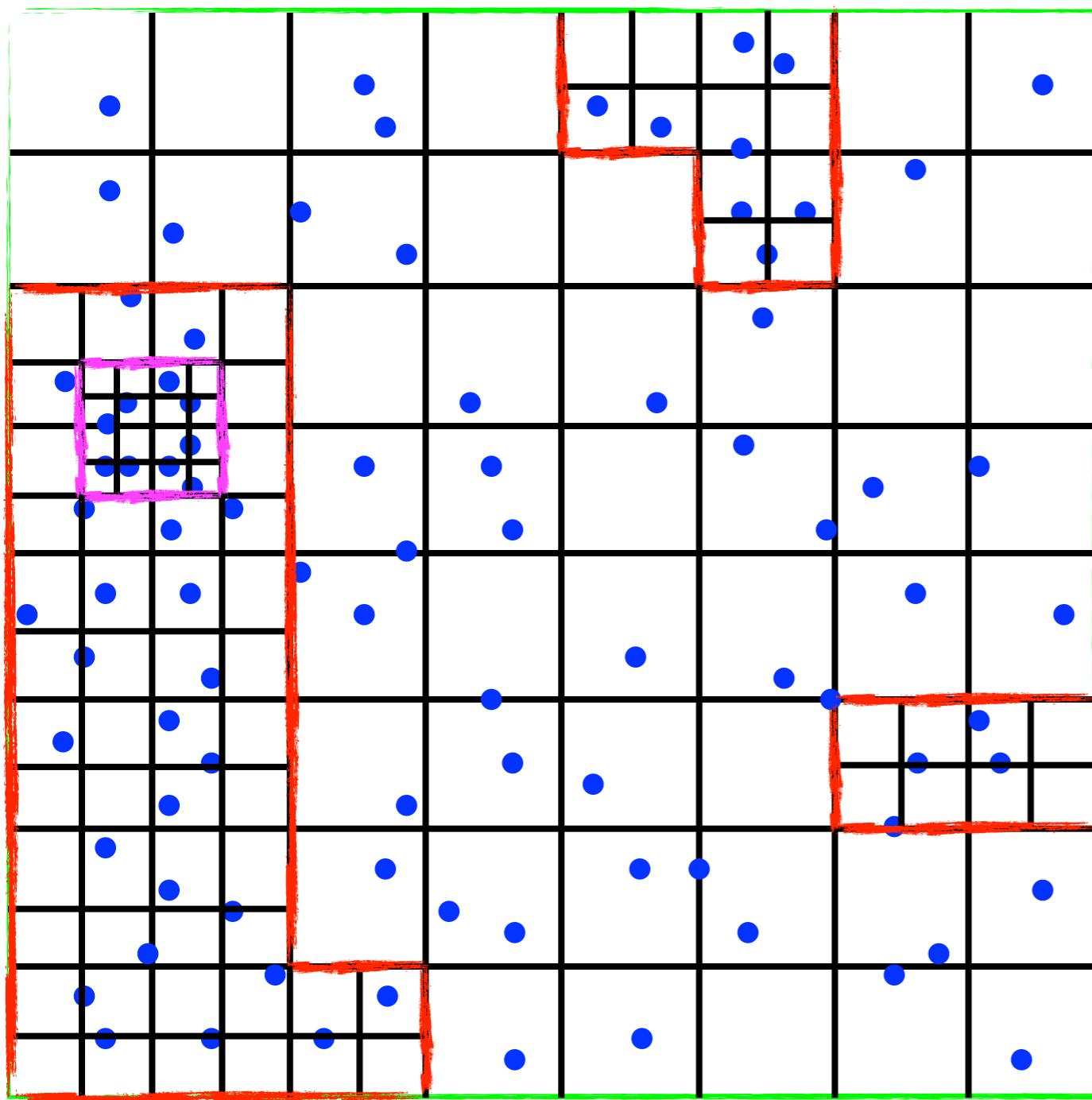
Limitations of Numerical Simulations

- Relaxation method, with appropriate discretisation of the equations to be solved (discussed above), works
- But its efficiency varies wildly depending on the nonlinearity of the equation. The more nonlinear the equation is, the harder it is for the method to converge
- It can take much longer time than a LCDM simulation with the same specifications. Some of the high-resolution simulations carried out so far take 10-20 times longer.
- Not yet realistic to run large number of simulations to, e.g., estimate covariance matrix.

Methods to Improve Performance

- There exist certain methods to speed up numerical simulations, or get around them.
- Can boost performance and allow large number of simulations to be generated in a reasonable timescale with realistic demand on computing resource.
- But may involve approximations so that accuracy is not always great, so has to be used with caution.

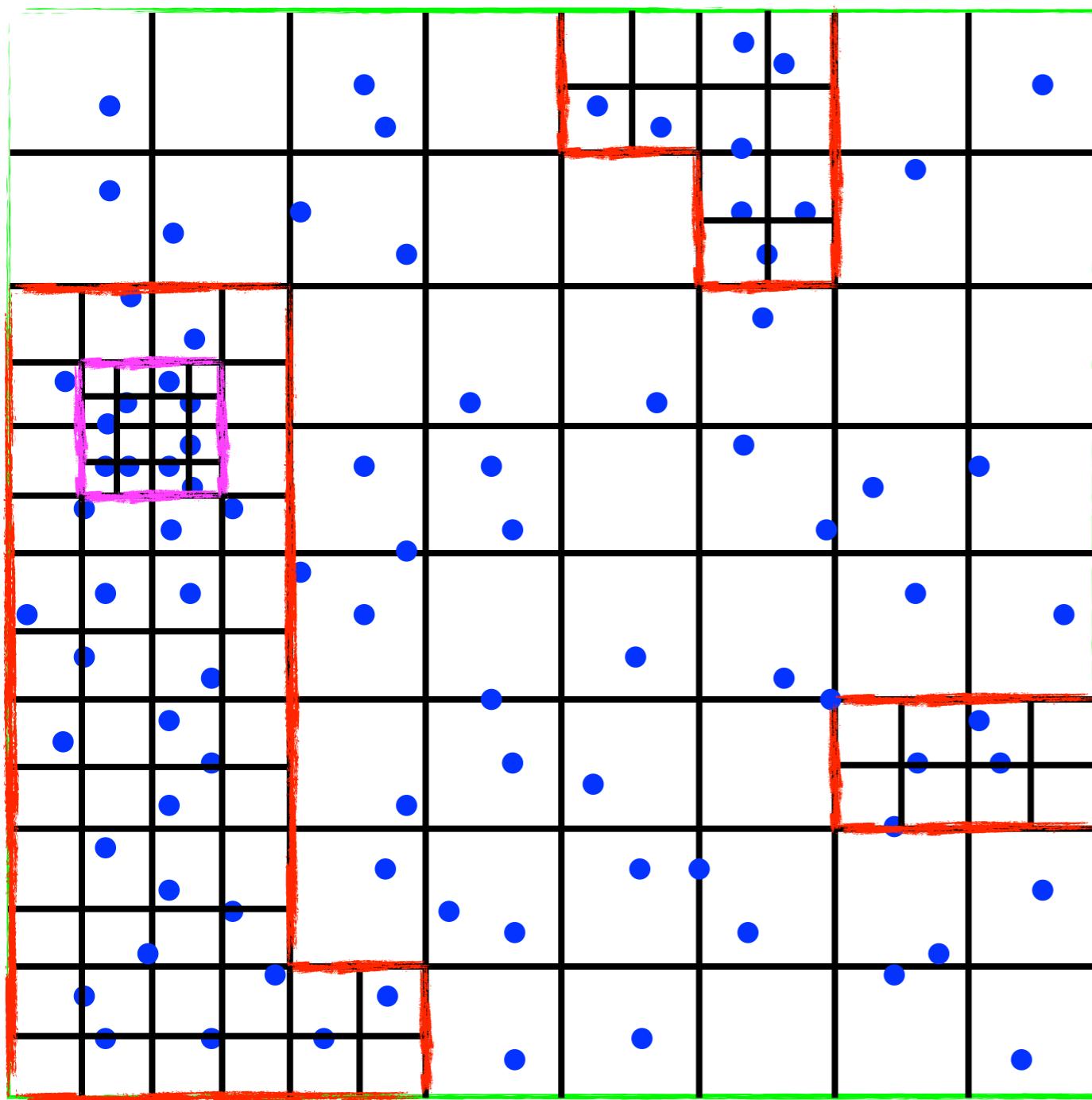
Truncated Simulations



In high-resolution simulations, many regions in the simulation box are refined, and a lot of the time is spent doing relaxations on these refinements. For non standard models this can be extremely slow.

But is it completely necessary to do this?

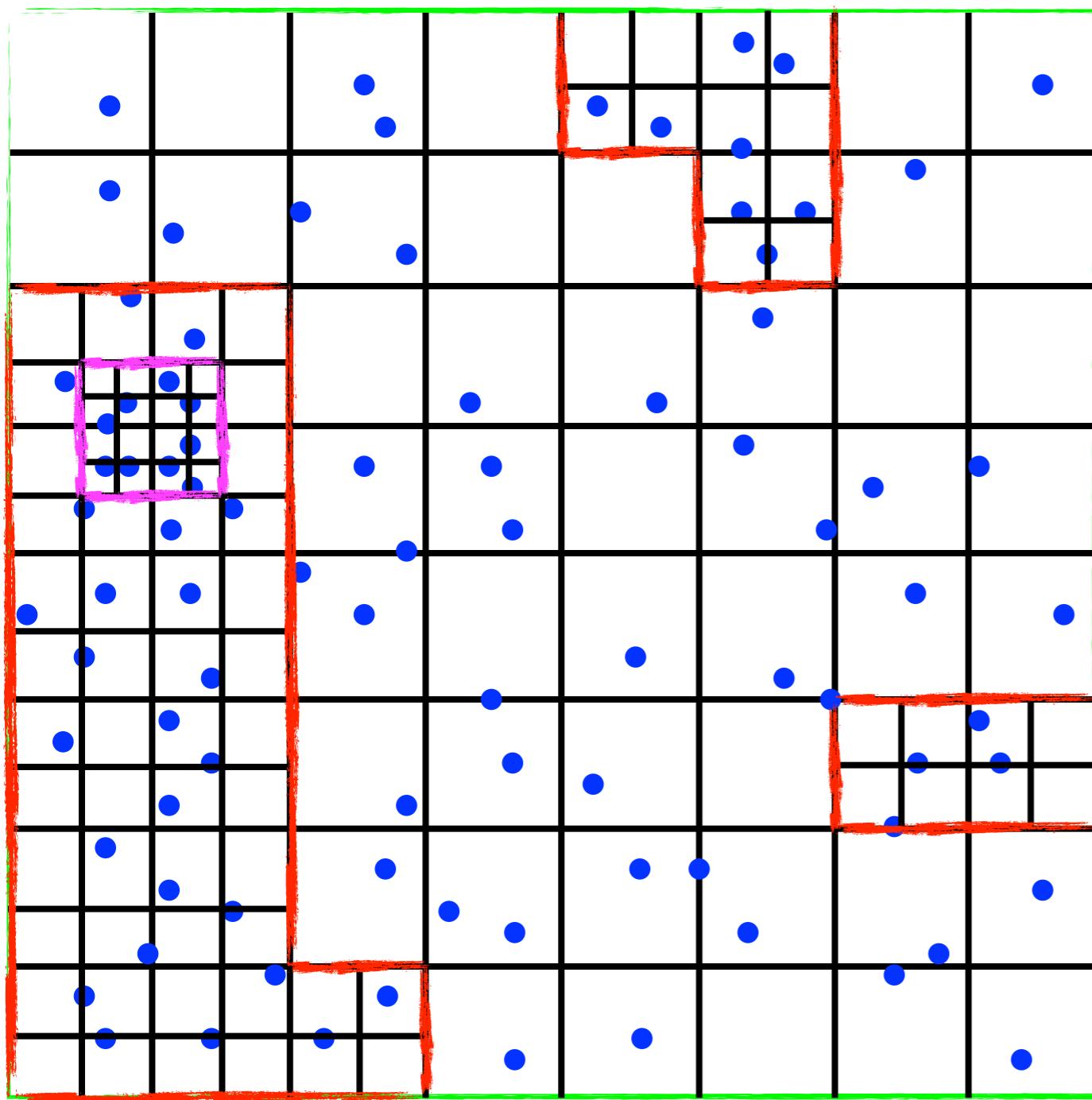
Truncated Simulations



In many nonstandard models, solar system tests place severe constraints on how strong the new force can be in high density regions.

We are spending a lot of time solving a force in regions where it is supposed to be weak! Is it a good idea?

Truncated Simulations

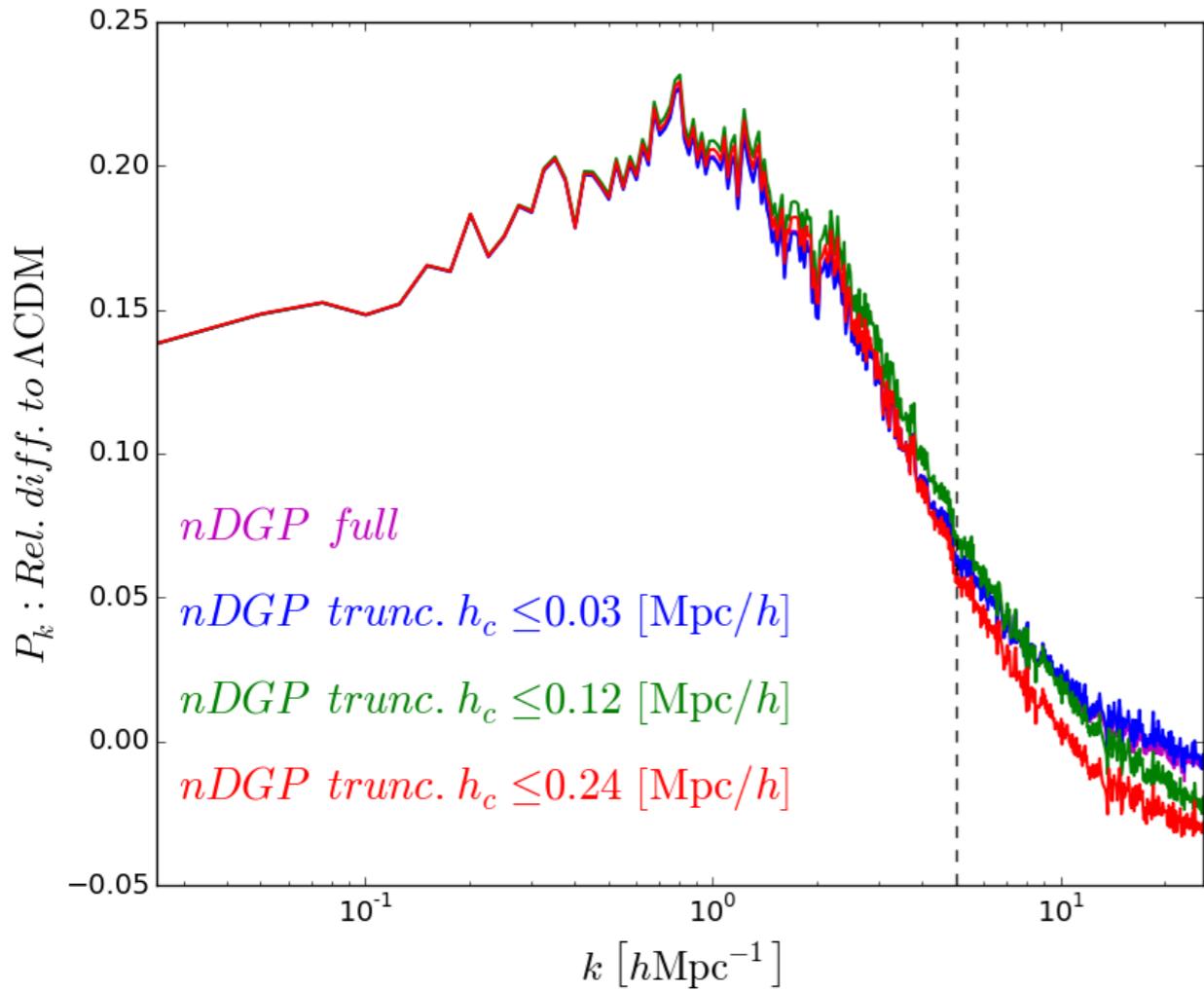


An alternative: calculate the standard gravitational force on all levels of refinements, but only calculate the new force on the lowest levels. Note that this is **not** to neglect the new force in dense regions, but instead to get its values by interpolating from coarse to fine levels.

A small error on a small quantity may be insignificant?

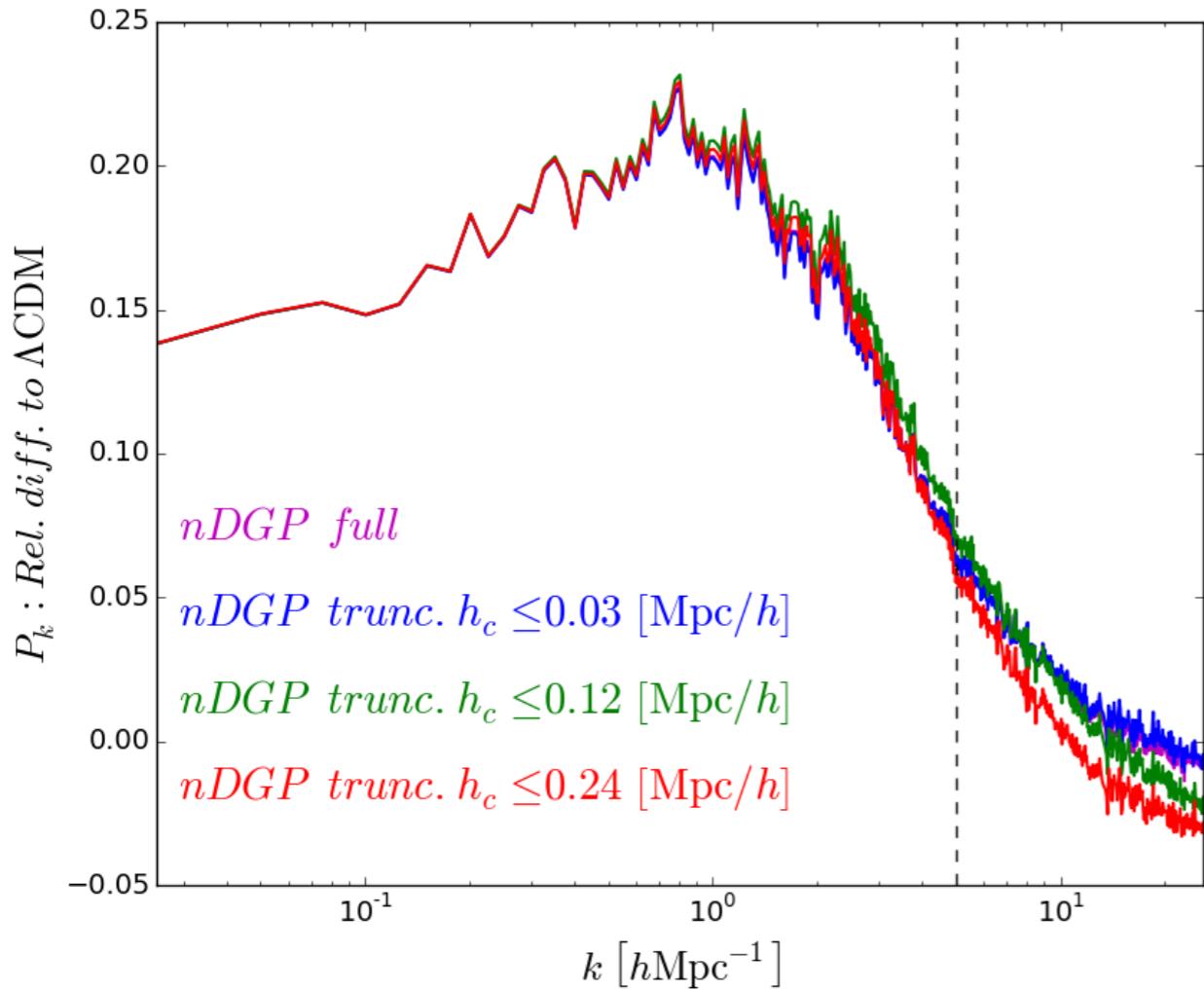
Truncated Simulations

$$\nabla^2 \varphi + \frac{r_c^2}{3\beta(a)a^2} \left[(\nabla^2 \varphi)^2 - \nabla^i \nabla^j \varphi \nabla_i \nabla_j \varphi \right] = \frac{1}{3\beta(a)} 8\pi G \delta \rho_m a^2$$



- 1% accuracy for matter power spectrum at $k < 5 \text{ h/Mpc}$;
- 3% accuracy for halo mass, abundance, profile etc.;
- 0.05% accuracy for halo positions and velocities.

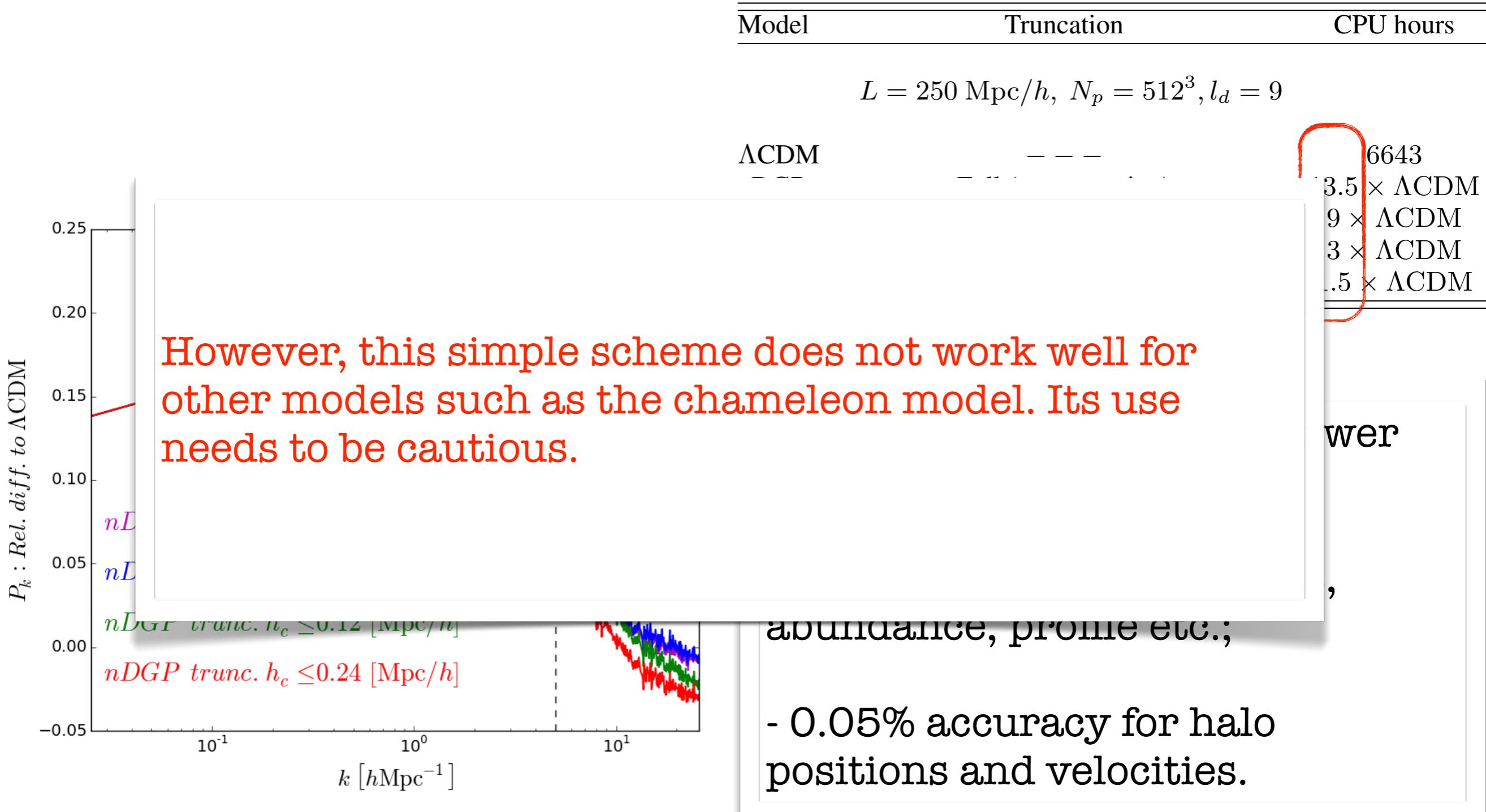
Truncated Simulations



Model	Truncation	CPU hours
$L = 250 \text{ Mpc}/h, N_p = 512^3, l_d = 9$		
ΛCDM	— — —	6643
nDGP	Full (no truncation)	$13.5 \times \Lambda\text{CDM}$
nDGP	$h_c \leq 0.03$ ($l_{trunc} = l_d + 4$)	$9 \times \Lambda\text{CDM}$
nDGP	$h_c \leq 0.12$ ($l_{trunc} = l_d + 2$)	$3 \times \Lambda\text{CDM}$
nDGP	$h_c \leq 0.24$ ($l_{trunc} = l_d + 1$)	$1.5 \times \Lambda\text{CDM}$

- 1% accuracy for matter power spectrum at $k < 5 \text{ h/Mpc}$;
- 3% accuracy for halo mass, abundance, profile etc.;
- 0.05% accuracy for halo positions and velocities.

Truncated Simulations



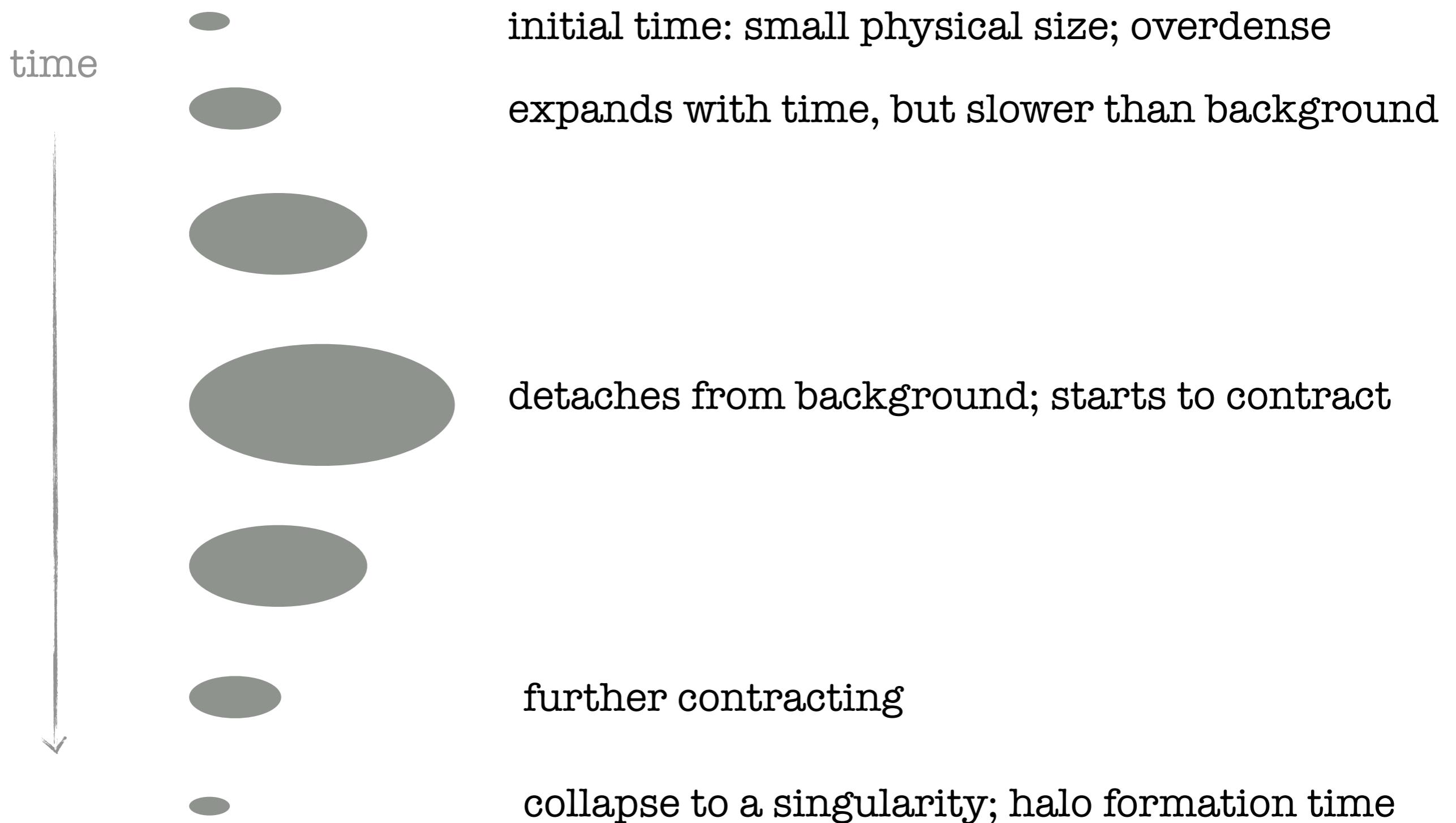
Fitting Formulae

- These are analytical formulae for specific quantities (e.g., halo mass function, matter power spectrum)
- The formulae are motivated by analytical predictions, which usually capture the essential physics so that they are qualitatively correct, but lack quantitative accuracy because the simple analytical predictions are not expected to fully recover the nonlinear physics
- As a result, the analytical predictions are generalised to allow more flexibility. This introduces new parameters which are then calibrated using simulations.
- Allows continuous variation of model parameters.

Fitting Formulae (Halo Mass Function)

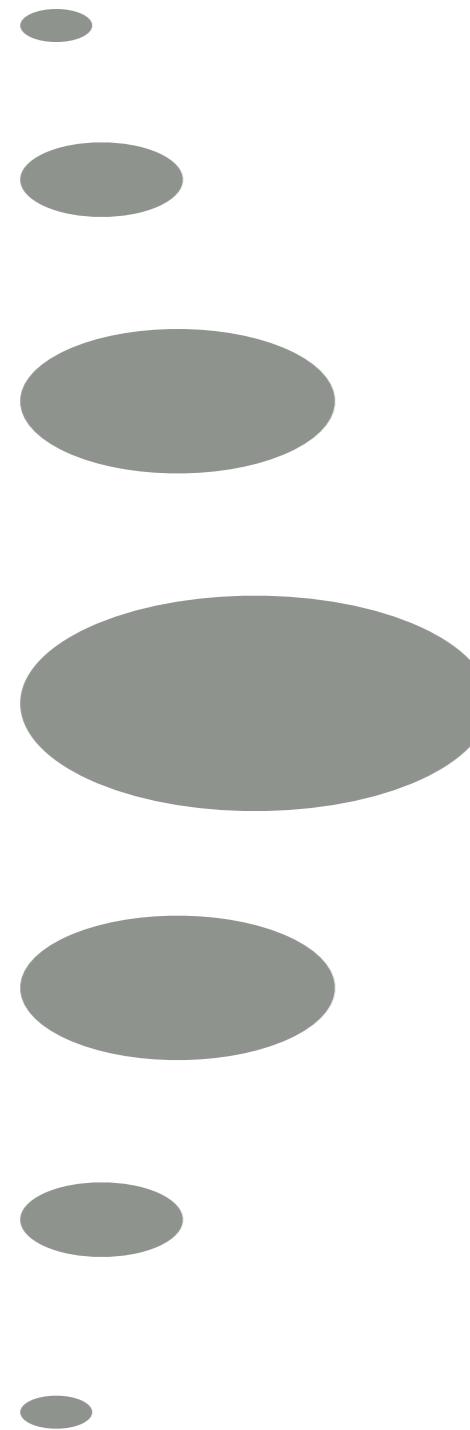
- One example is the halo mass function.
- The current consensus is that dark matter haloes form as a consequence of initially slightly overdense regions attracting more matter towards them gravitationally.
- As a result, we can connect dark matter haloes at late times to peaks in the initial density field (which is nearly Gaussian).

Fitting Formulae (Halo Mass Function)



Fitting Formulae (Halo Mass Function)

time



initial time: small physical size; overdense

This so-called spherical collapse model models the evolution of the spherical initial overdense region.

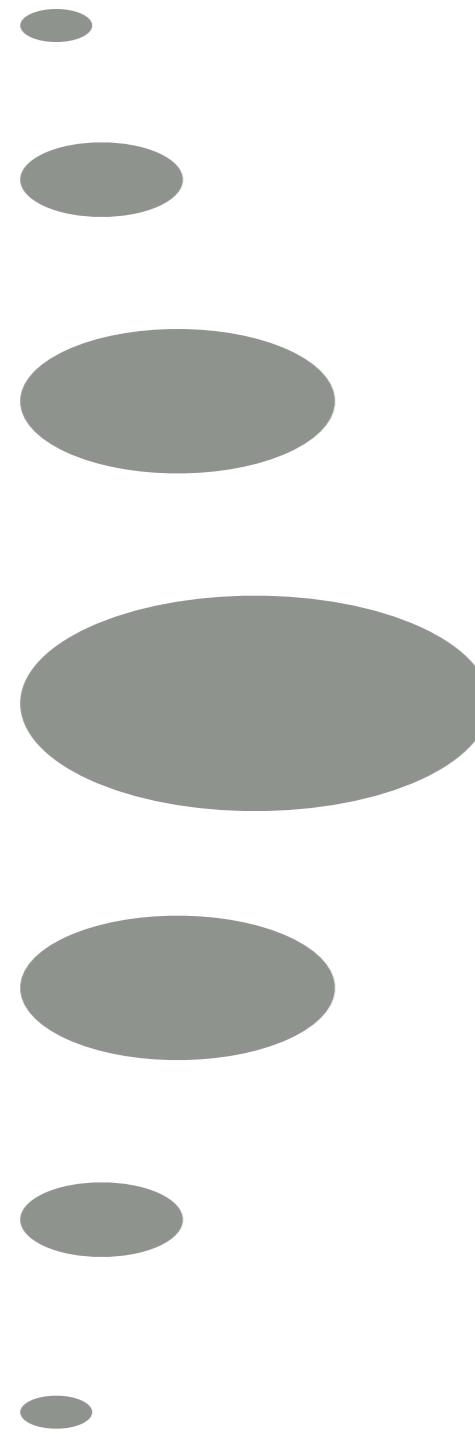
If the initial size of the spherical region is larger, more mass is contained inside it and the final halo is more massive.

If the initial overdensity of the spherical region is larger, the collapse happens earlier so that the halo forms earlier.

collapse to a singularity; halo formation time

Fitting Formulae (Halo Mass Function)

time



initial time: small physical size; overdense

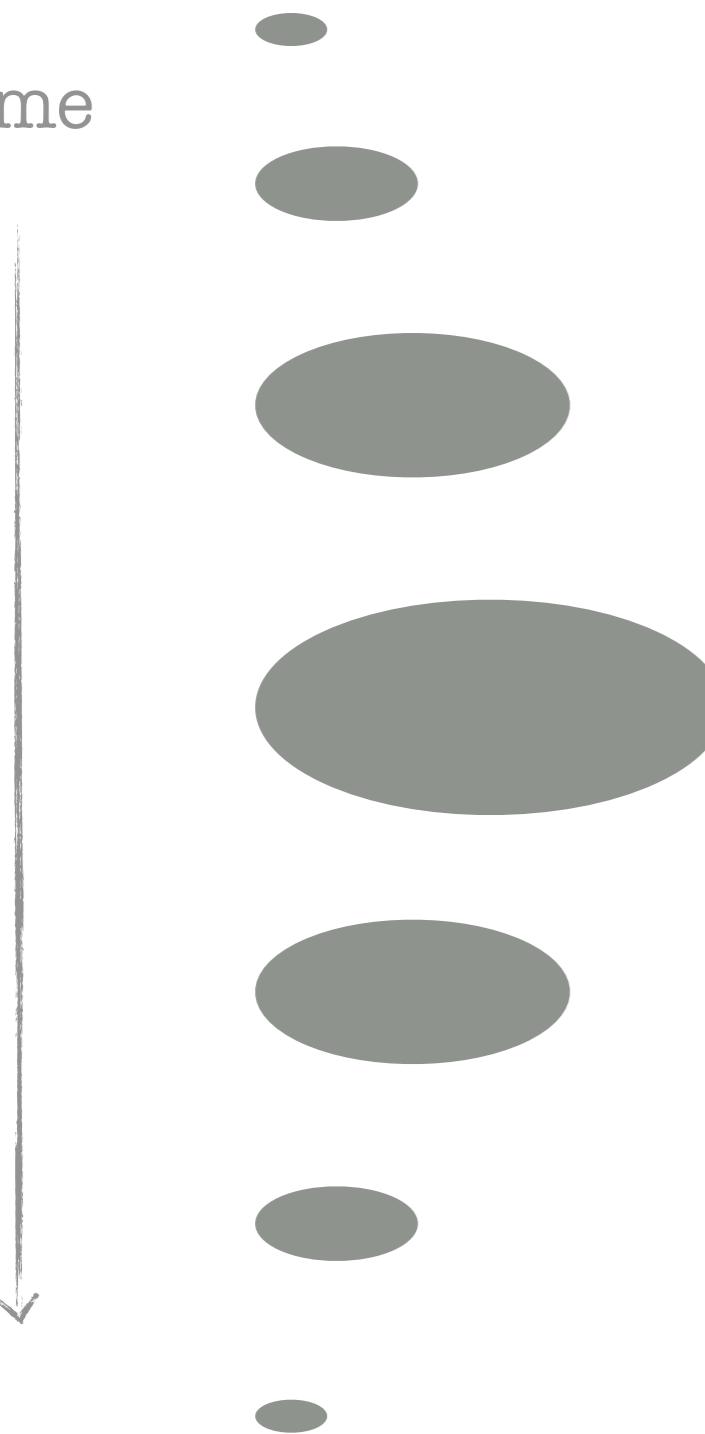
At a result, we will know what initial size and initial overdensity the spherical region must have for a halo of a given mass M to form at a given redshift z .

The abundance of such initial spherical regions can be obtained from the initial Gaussian density field. This gives us an idea of the abundance of haloes with mass M at redshift z .

collapse to a singularity; halo formation time

Fitting Formulae (Halo Mass Function)

time



initial time: small physical size; overdense

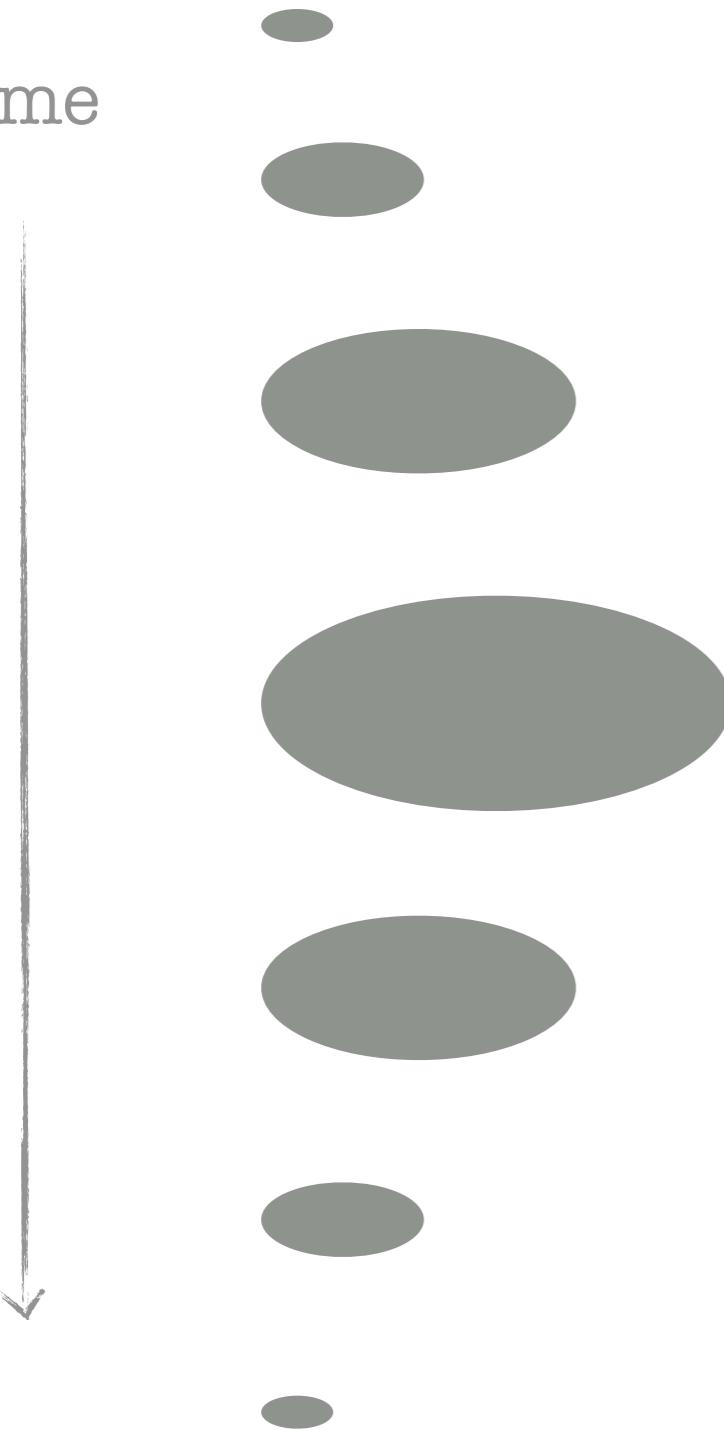
The idea is incredibly simple, and there is no reason why it won't work for nonstandard models.

Indeed, in the nonstandard models, if we agree to start from the same initial Gaussian density field (which we should, as this constrained by CMB), the only thing that needs to be changed is the spherical collapse model, which tells us how an initial spherical overdense patch evolves in time.

collapse to a singularity; halo formation time

Fitting Formulae (Halo Mass Function)

time



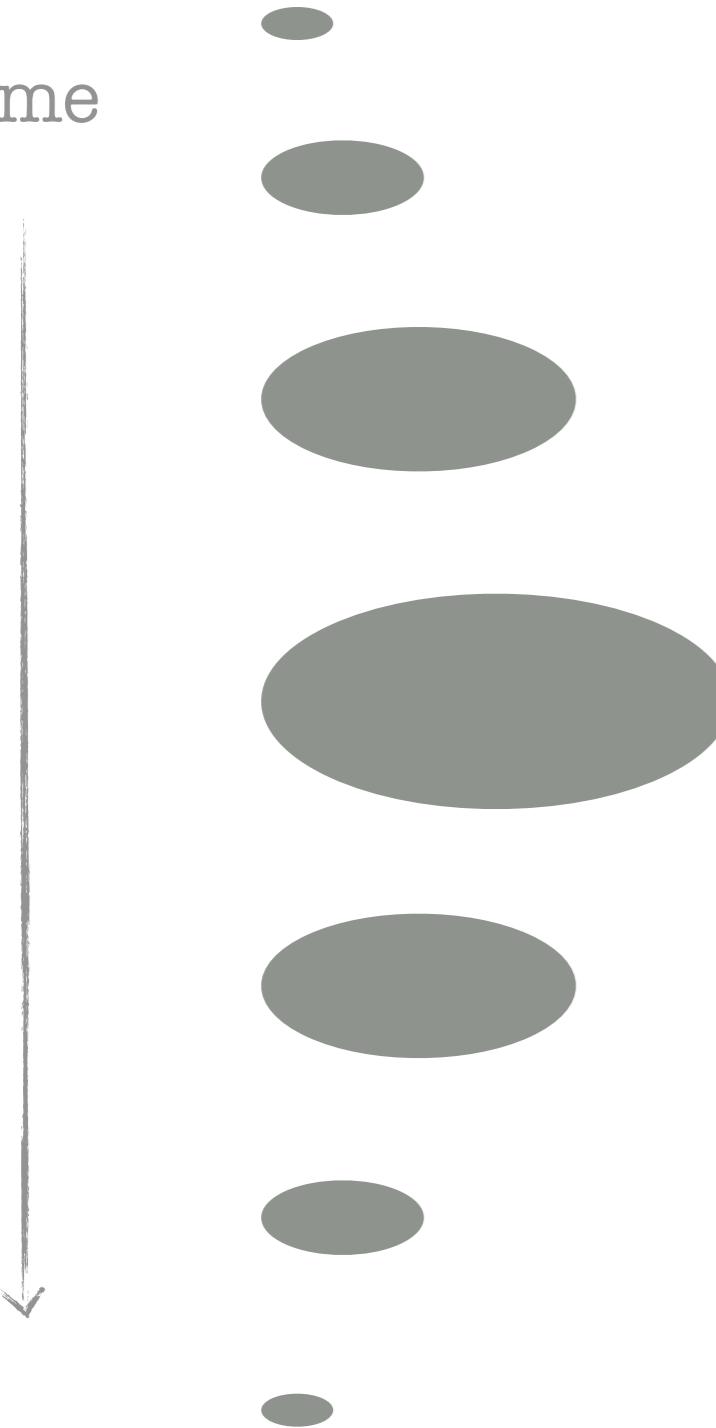
Case 1: Standard gravity

For spherical symmetry the evolution of the patch **does not** depend on what is present outside the patch. The evolution is completely determined by **Newtonian gravitational law**.

Easy.

Fitting Formulae (Halo Mass Function)

time



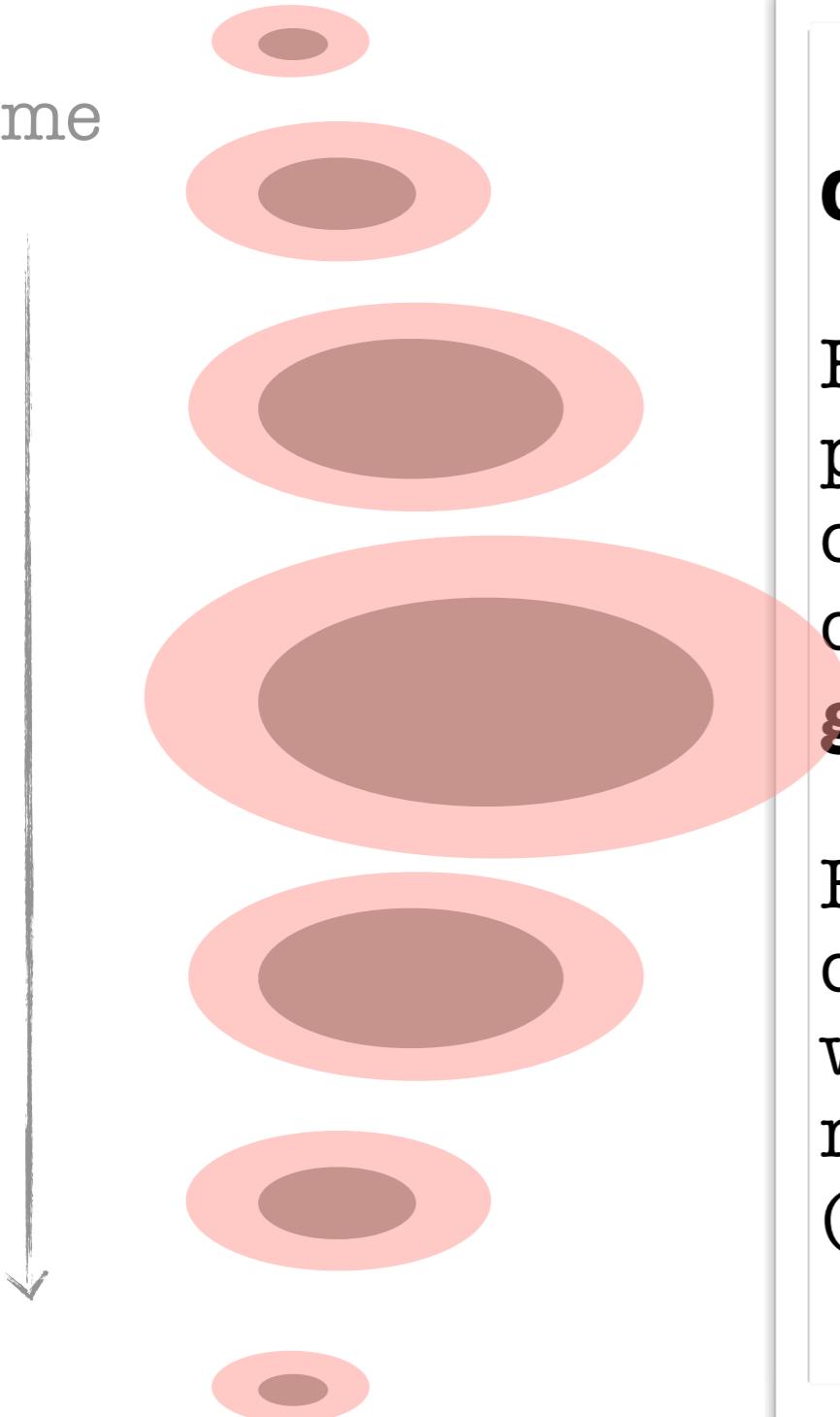
Case 2: Vainshtein-type models

For spherical symmetry the evolution of the patch **does not** depend on what is present outside the patch. The evolution is completely determined by a **modified gravitational gravitational law which is known**.

Less easy, but still straightforward.

Fitting Formulae (Halo Mass Function)

time

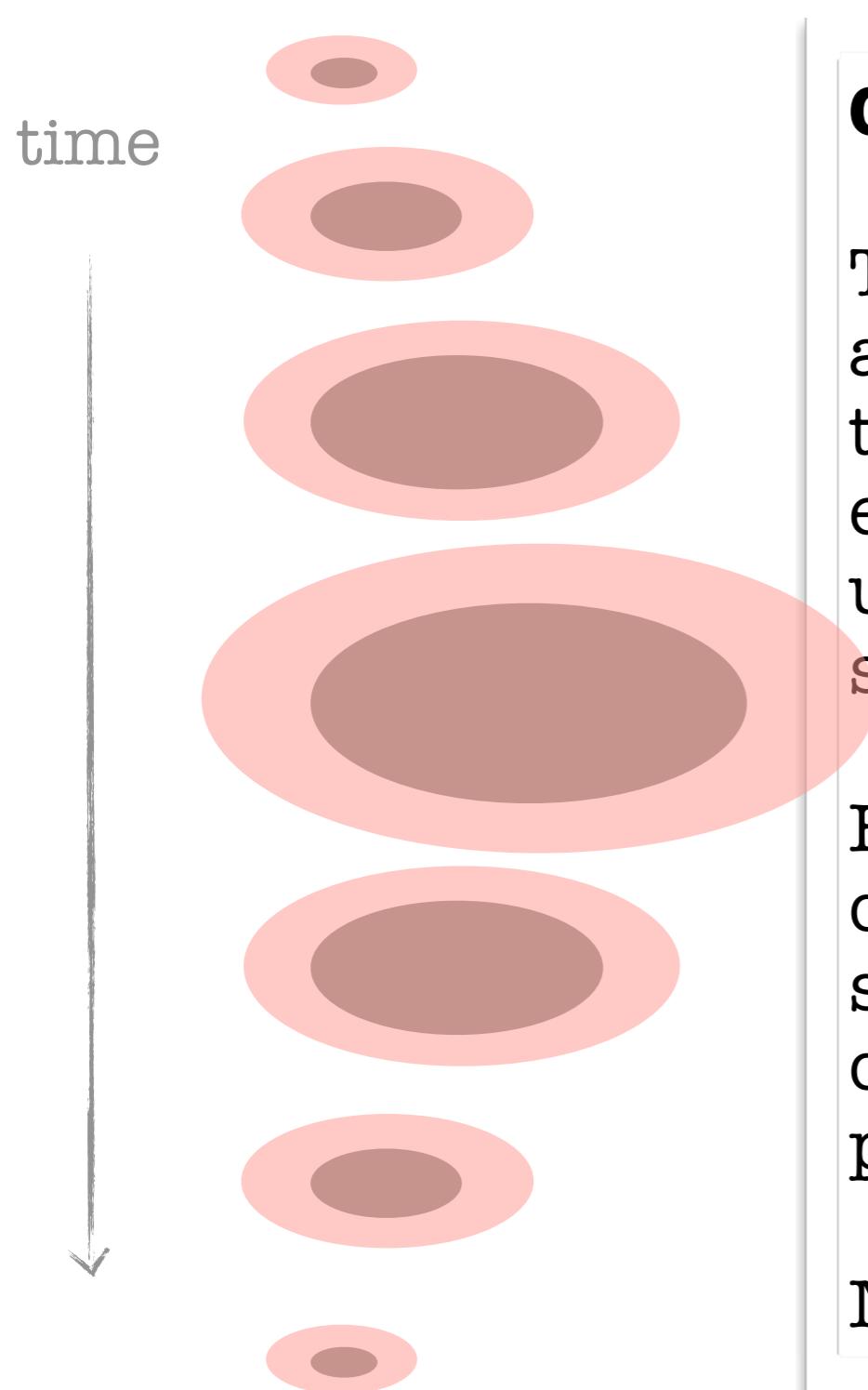


Case 3: Chameleon-type models

For spherical symmetry the evolution of the patch **does** depend on what is present outside the patch. The evolution is determined by a **modified gravitational gravitational law which is known**.

Hard. The evolution of the spherical patch depends on the density in its environment, which evolves itself. Has to simultaneously model the evolution of the spherical patch (grey) and its environment (red).

Fitting Formulae (Halo Mass Function)



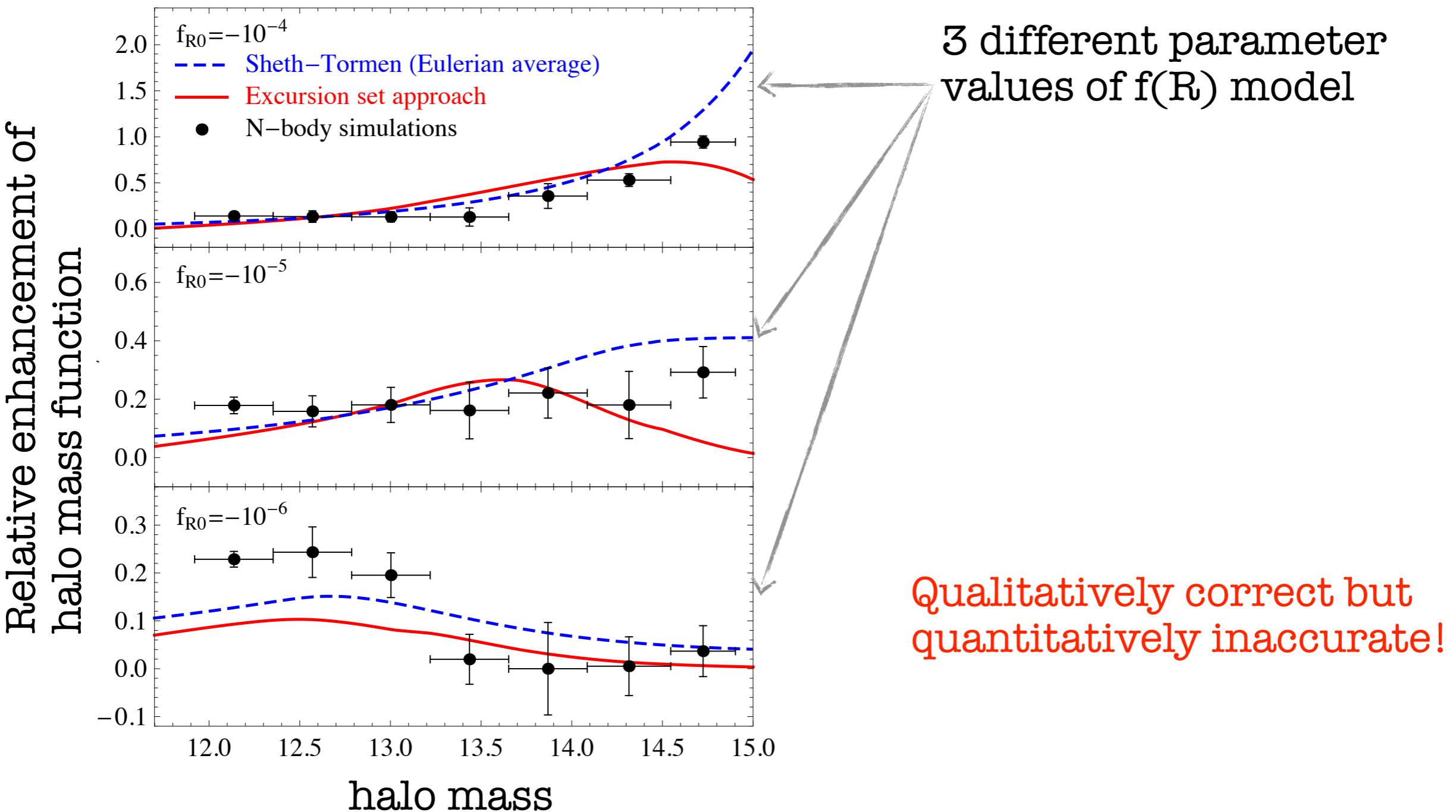
Case 3: Chameleon-type models

The environment (red) is spherical as well, and is considered to be significantly larger than the overdense patch (grey). Its evolution can be approximately done by using the spherical collapse model for standard gravity.

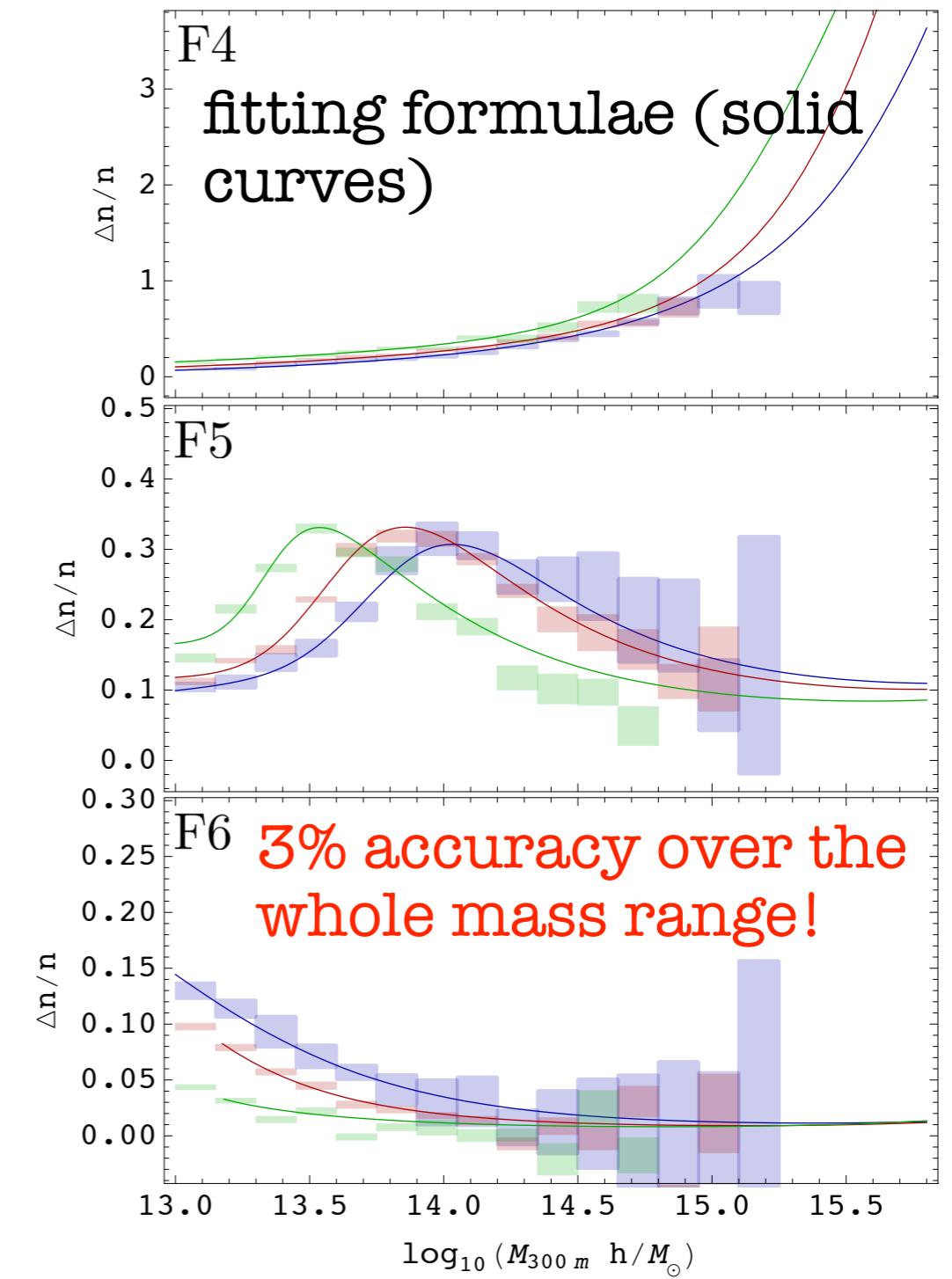
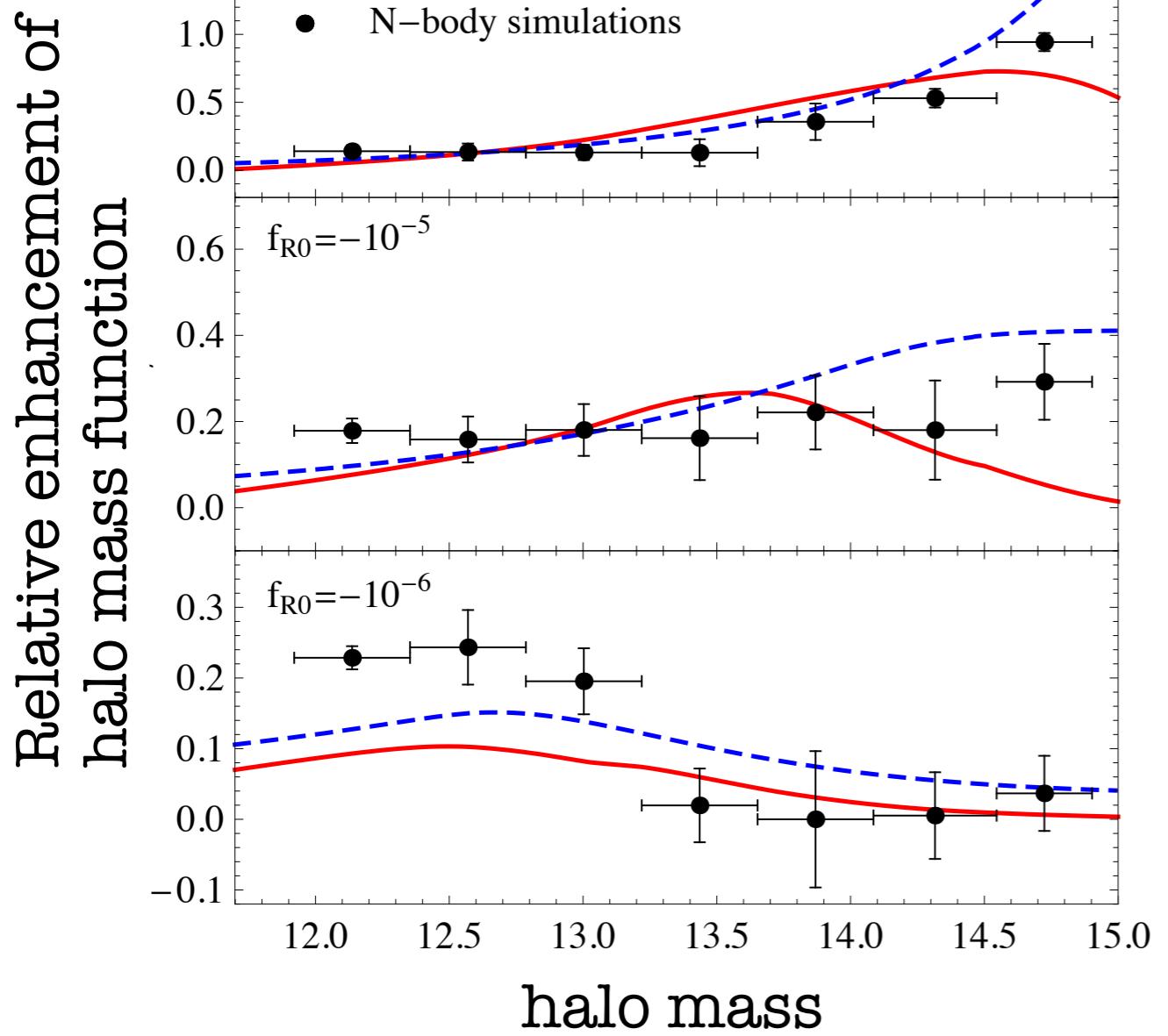
Environments with different initial densities evolve differently. Has to find the spherical collapse in different environment densities, and then integrate over the probability distribution of the latter.

Nontrivial but still doable.

Fitting Formulae (Halo Mass Function)



Fitting Formulae (Halo Mass Function)

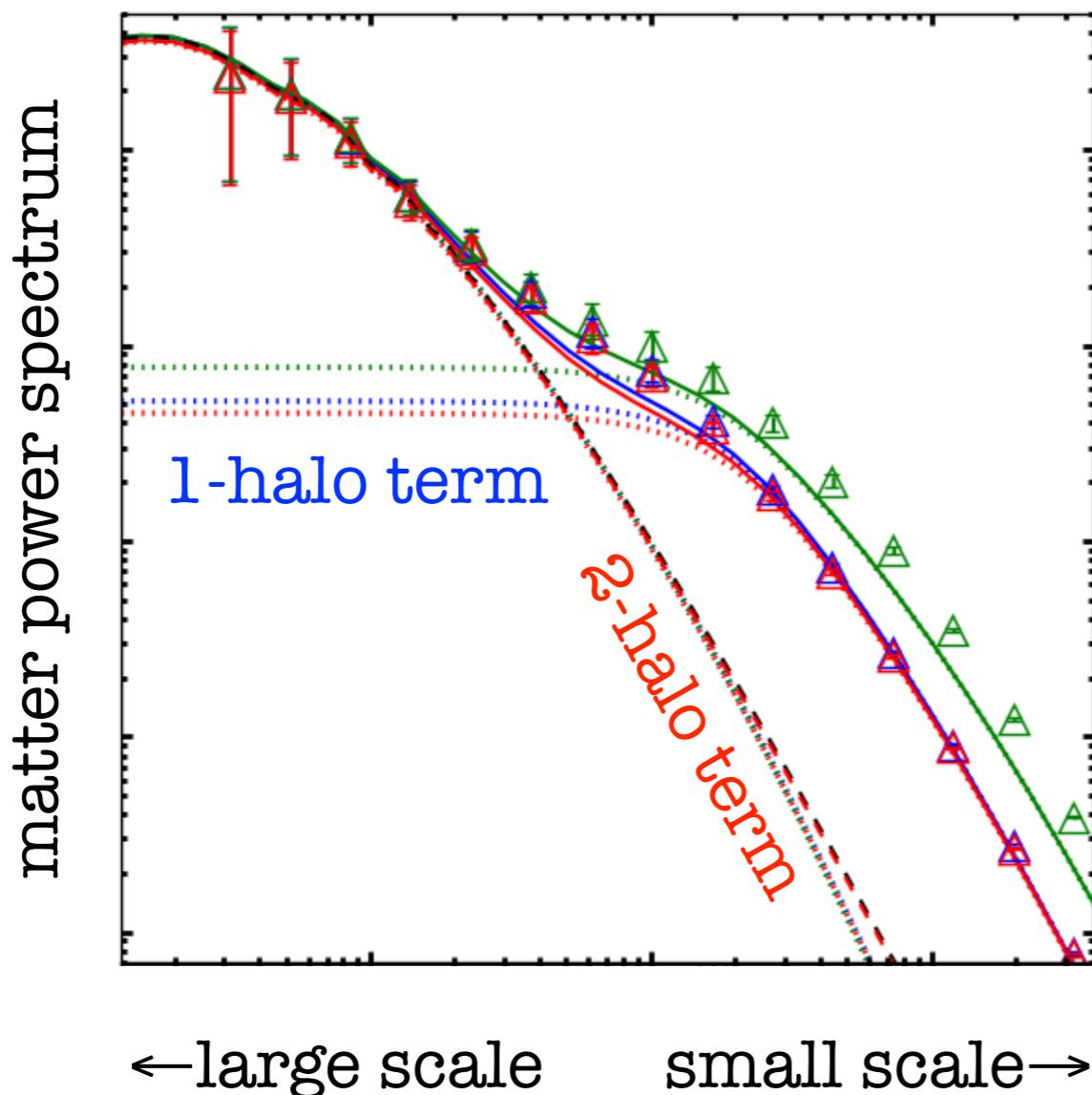


Fitting Formulae (Halofit)

- The original halo model separates the nonlinear matter power spectrum into two parts:
- a 1-halo term which quantifies the correlation of matter within single haloes (assuming NFW profile); dominates on small scales
- a 2-halo term which quantifies the correlation of matter from different haloes; dominate on large scales

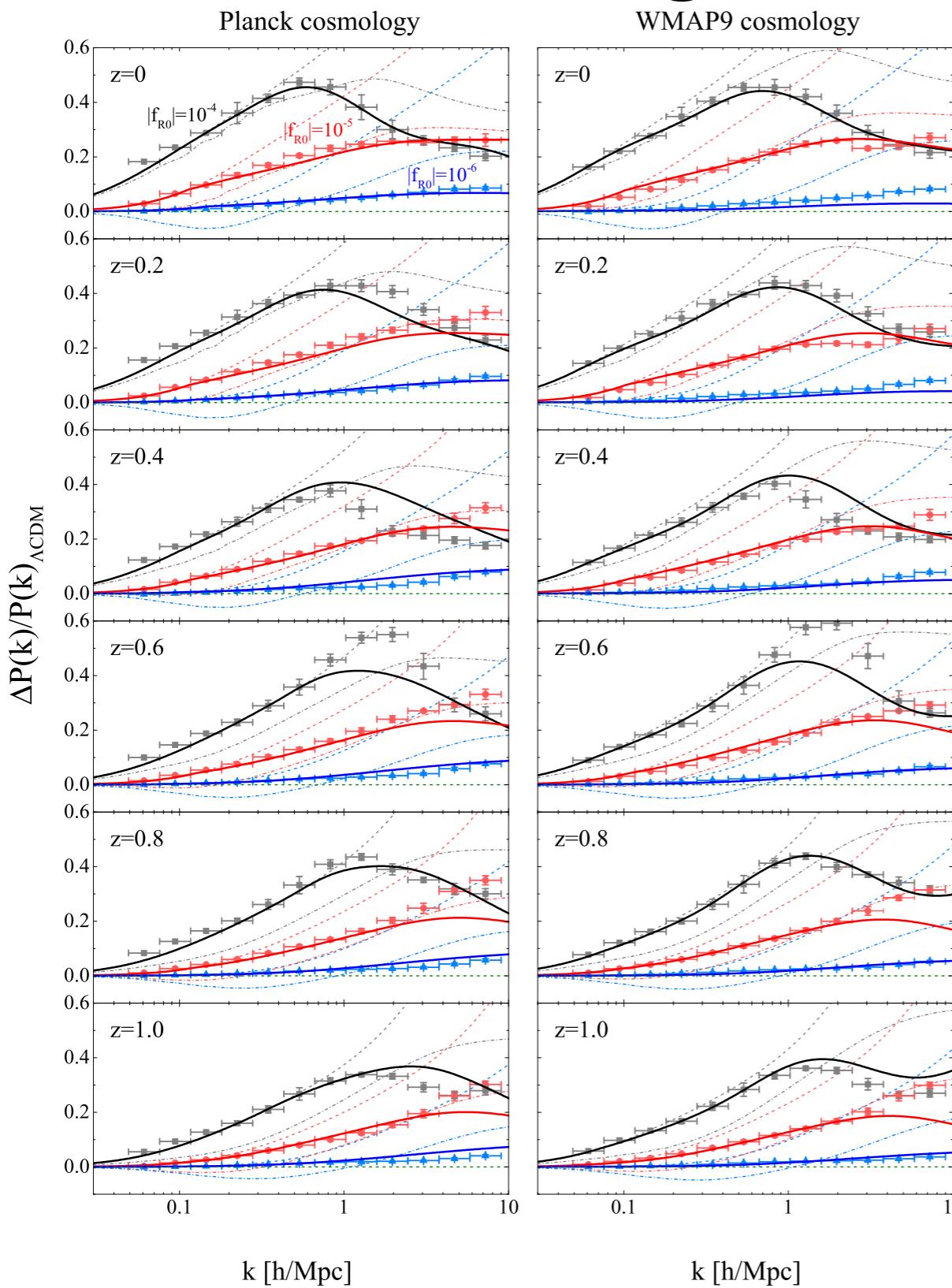
Fitting Formulae (Halofit)

colours: 3 different models
solid lines: halo model prediction
symbols: simulations



Qualitatively correct but
quantitatively inaccurate!

Fitting Formulae (Halofit)



colours: 3 different $f(R)$ models
 panels: different redshifts
 symbols: simulations
 solid lines: fitting formulae
 dashed lines: linear theory

vertical axis: relative
 enhancement of matter power
 spectrum wrt standard gravity

horizontal axis: scale
 (wavenumber)

Other Methods

- There are many other possibilities of replacing accurate simulations with less accurate, but (much) more efficient approximate methods, e.g., remapped simulations, Zel'dovich displacement of particles, 2LPT, etc.
- Some of these have been studied and found to work well for nonstandard models, e.g., remapped simulations (see work of Alex Mead, John Peacock and collaborators)
- Other possibilities are less explored, but can be potentially useful as well.