COLA with scale-dependent growth

Hans Winther, ICG Portsmouth
(IPhT CEA Saclay September 2017)

Motivation

High precision test of gravity on cosmological scales is one of the main objectives of many current and future galaxy surveys.

Perturbation theory can only bring us so far. To obtain accurate predictions in the quasi-linear to non-linear regime numerical simulations are required.

For current and future surveys there is also a need to develop large ensembles of galaxy mocks to model the observables and their covariances etc. N-body simulations are computationally expensive. Faster methods are needed.

I will discuss one such method - COmoving Lagrangian Acceleration - for doing such simulations and how this can be extended from LCDM to a large class of non-standard models and also including massive neutrinos.

N-body simulations in LCDM

The CDM density/velocity field in the universe is described by particles which move under Newtonian looking equations

$$\ddot{\vec{x}} = -\vec{\nabla}\Phi$$
$$\nabla^2\Phi = 4\pi G a^4 \delta\rho_m$$

Simple to solve. Challenge (apart from introduction of additional physics) is to make it accurate on a wide range of scales and fast. Requires adaptive resolution (adaptive grids or trees), good load-balancing / scaling, etc.

N-body simulations beyond LCDM

In common MG scenarios (scalar-tensor theories) we simply add an additional force which an associated field-equation:

$$\ddot{\vec{x}} = -\vec{\nabla}\Phi - \vec{\nabla}\varphi$$

$$\nabla^2\Phi = 4\pi G a^4 \delta\rho_m$$

$$\nabla^2\varphi = f(\varphi, \nabla\varphi, \delta\rho_m, a)$$

Not that much more complicated. The main difference is that the field-equation is (has to be) highly non-linear and often has bad convergence properties so often a factor of a few slower (can be improved see e.g. **Barreira et al. 2015** and **Bose et al. 2016**).

COLA method

Idea (**Tassev et al. 2013**) is simple: instead of solving for the full particle trajectories in a simulation we perturb around the path predicted by Lagrangian perturbation theory.

$$\vec{x} = \vec{x}_{\rm LPT} + \vec{\delta x} \qquad\qquad \vec{x}_{\rm LPT} = \vec{q} + \vec{\Psi}(\vec{q}, a)$$

$$\ddot{\vec{\delta x}} = -\vec{\nabla}\Phi - \ddot{\vec{\Psi}}$$

$$\nabla^2\Phi = 4\pi G a^4 \delta\rho_m$$

Otherwise solved the same way as a normal N-body simulation.

COLA method

The displacement field can be expanded in a perturbative series

$$\vec{\Psi} = \vec{\Psi}^{(1)} + \vec{\Psi}^{(2)} + \ldots$$

$$\vec{\nabla} \cdot \vec{\Psi}^{(1)}_{\rm ini} = -\delta_{\rm ini} \qquad \vec{\nabla} \cdot \vec{\Psi}^{(2)}_{\rm ini} = -\frac{1}{2}[\vec{\Psi}^{(1)}_{i,i}\vec{\Psi}^{(1)}_{j,j} - \vec{\Psi}^{(1)}_{i,j}\vec{\Psi}^{(1)}_{i,j}]$$

In LCDM the displacement field factors in time and space

$$\vec{\Psi}^{(i)} = D_i(a)\vec{\Psi}_{\rm ini}$$

This means we only need to compute the displacement field once (which we get this for free when we generate the initial conditions).

COLA method

- The COLA split is exact in the limit of large number of time steps / small grid-size.

- What do we gain doing this? Allows us to take large time-steps, O(10) instead of O(1000) in N-body, and still maintain accuracy on the largest scales (at the expense of sacrificing accuracy on the smallest scales).

- We can work with a fixed mesh which allows for solving everything with FFTs.

- The method is in practice often O(100-1000) times faster than standard N-body.

- However… not a replacement of N-body! Useful for large scale clustering statistics, generation of galaxy mocks etc. Not useful for studying small-scale dynamics.

COLA for scale-dependent growth

In general the growth of linear perturbations are determined by an effective Newtonian constant

$$\frac{G(k,a)}{G} \equiv \mu(k,a)$$

which is often scale-dependent -> scale-dependent growth-factors. To generalise COLA we need to add support for this.

We also need to solve for the additional gravitational degrees of freedom, i.e. solving

$$\nabla^2\varphi = f(\varphi, \nabla\varphi, \delta\rho_m, a)$$

The common used methods are too slow. We instead use a fast approximate method

A fast approximate method

For typical MG models we know how the solution to the field equation behaves in two regimes:

[1] On large cosmological scales the evolution is linear and typically on the form

$$\nabla^2\varphi = m^2(a)\varphi + \beta(a)\cdot 4\pi G a^4 \cdot \delta\rho_m$$

Using this in simulations is often not good enough as it misses the main ingredient of viable modified gravity theories: a screening mechanism to recover General Relativity in certain regimes.

[2] For a spherical symmetric density configuration we can solve (or approximate) the solution analytically. Typically one finds

$$F_\varphi = -\vec{\nabla}\varphi = C(a)\cdot\vec{\nabla}\Phi\cdot\epsilon(\Phi,\vec{\nabla}\Phi,\nabla^2\Phi\propto\rho_m)$$

Fifth-force is some strength times the Newtonian-force modified by some screening factor which depends on both the object in question and the environment.

A fast approximate method

The screening factor tells us how much of the matter-density that contributes to the fifth-force.

Usually depends on quantities that we know (or can easily compute) in a numerical simulation: the newtonian potential and/or it's first derivatives.

Proposed method (**Winther & Ferreira 2014**): we amend the linear-equation with a screening factor derived from spherically symmetric solutions:

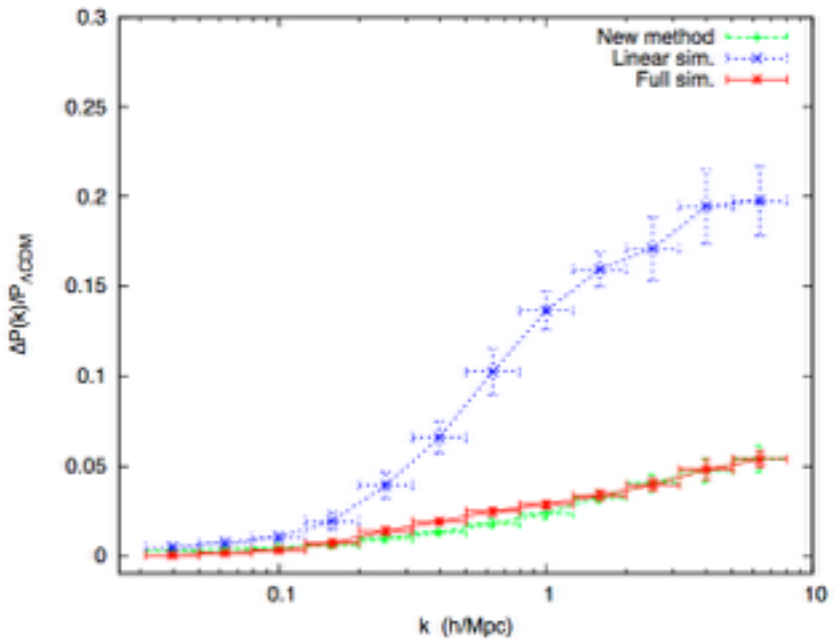
$$\nabla^2\varphi = m^2(a)\varphi + \beta(a)\cdot 4\pi G a^4 \cdot \delta\rho_m \cdot {\color{red}\epsilon(\Phi, \nabla\Phi, \rho_m)}$$

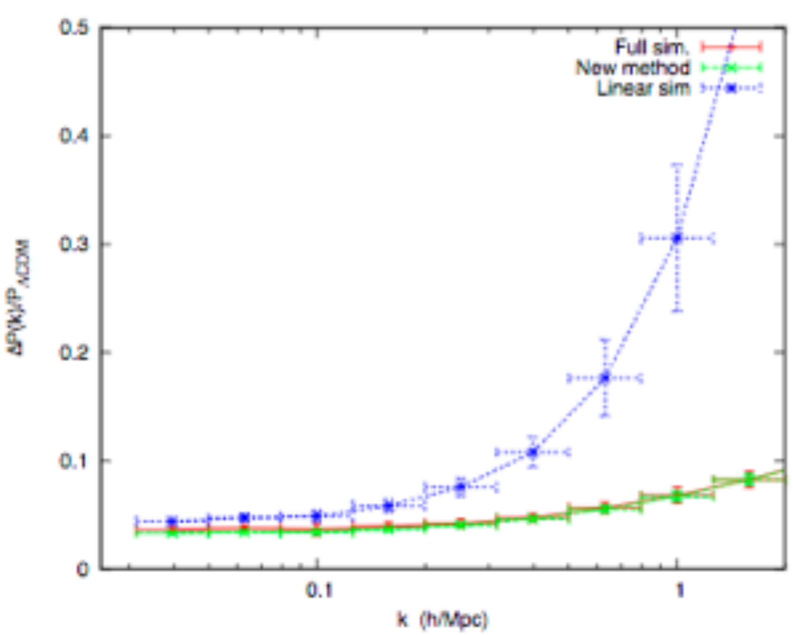
Agrees well with linear theory on the largest scales and models the screening effect on non-linear scales.

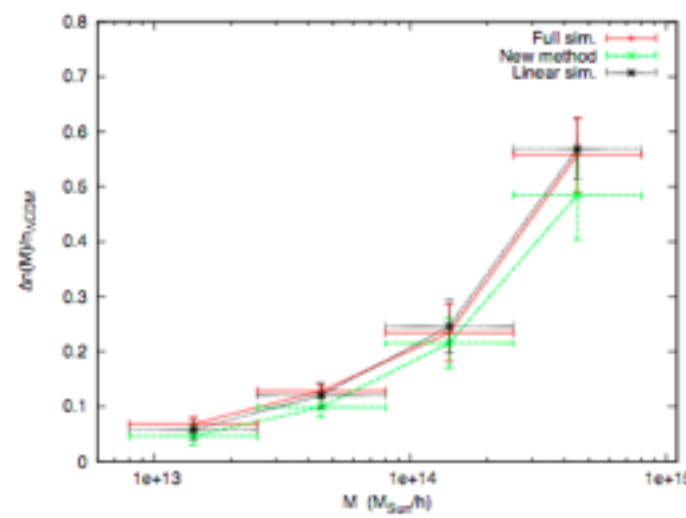
Linear field equation -> much easier to solve (FFTs).

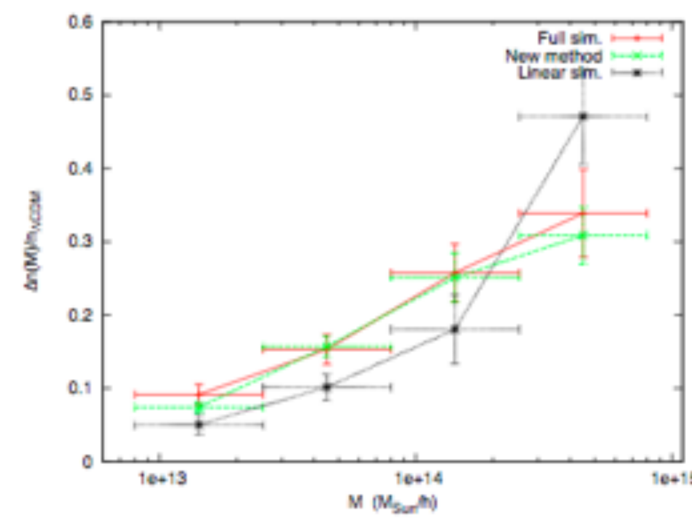
Implemented and tested in traditional N-body for several different models [f(R) gravity, nDGP, cubic galileon, symmetron]

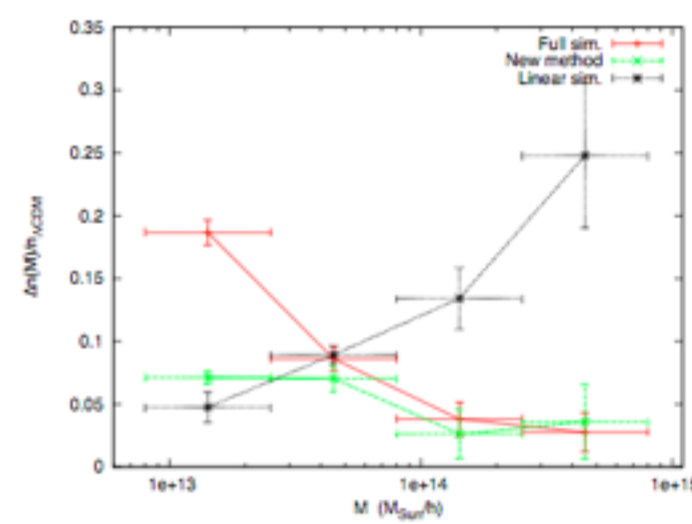
Comparison with exact simulations











For the models tested we find percent level agreement in the matter power spectrum to k of one to a few and in the halo mass function for the largest masses. Often conservative results.

COLA with scale-dependent growth

The displacement-fields does not factor in time and space, but they do factor in Fourier space to first order

$$\vec{\hat{\Psi}}^{(1)}(\vec{k}, a) = D_1(k, a)\hat{\Psi}^{(1)}_{\rm ini}(\vec{k}, a_{\rm ini})$$

Simple growth-factors for each Fourier mode

$$\ddot{D}_1(k, a) = \frac{3}{2}\Omega_m(a)a^4\, {\color{red}\mu(k, a)}\, D_1(k, a)$$

COLA with scale-dependent growth

To second order the expansion will in general depend on two wavenumbers:

$$\hat{\Psi}^{(2)}(k,a) = \int \frac{\mathrm{d}k_1 \mathrm{d}k_2}{(2\pi)^3} \delta^D(k - k_{12})\delta(k_1,a)\delta(k_2,a)D_2(k,k_1,k_2,a)$$

where the growth factor is determined by

$$\ddot{D}_2 \simeq \frac{3}{2}\Omega_m(a)H^2a^4\mu(k,a)\left[D_1(k_1,a)D_1(k_2,a) - D_2\right]\left(1 - \frac{(k_1\cdot k_2)^2}{k_1^2k_2^2} + \gamma_2(k,k_1,k_2,a)\right)$$

where gamma is a model dependent quantity. Computing the integral above is too expensive so we settle on an approximation here by assuming the same k1,k2-dependence as in LCDM:

$$\ddot{\hat{D}}_2(k,a) = \frac{3}{2}\Omega_m(a)H^2a^4\mu(k,a)[D_1(k,a)^2 - \hat{D}_2(k,a) + \gamma_2]$$

COLA with scale-dependent growth

With these approximations the displacement-fields can be computed at every time-step as:

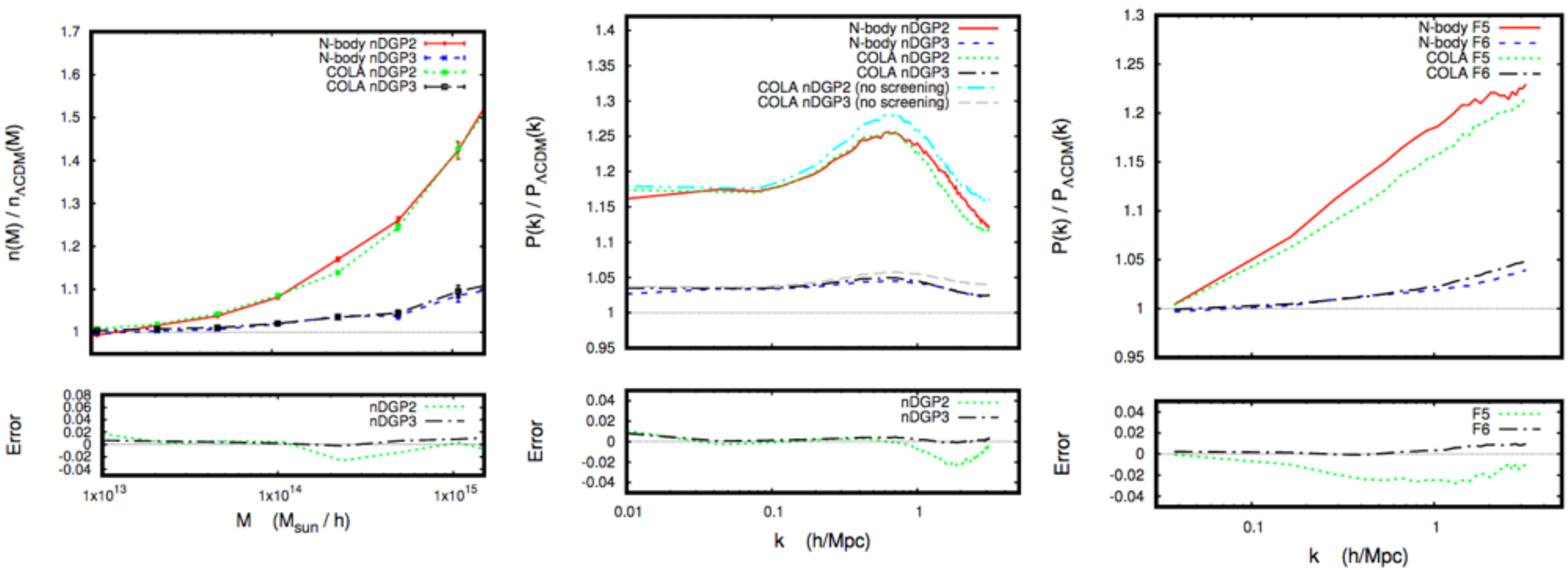
$$\vec{\Psi}^{(i)}(\vec{q}, a) = \mathcal{F}^{-1}\left[\hat{\vec{\Psi}}^{(i)}(\vec{k}, a_{\rm ini})\frac{D_i(k, a)}{D_i(k, a_{\rm ini})}\right]$$

Makes it roughly 3 - 4 - 5 times slower than COLA for LCDM, however still a huge O(100) speedup compared to N-body.

Using only time-dependent growth-factors (small k limit) is one approximation that seems to work very well for many models and makes the time comparable to LCDM.

We implemented this in the public availiable L-PICOLA code (**Howlett, Manera, Percival 2015**) and tested it against full N-body simulations (**Winther, Koyama, Manera, Wright, Zhao 2017**).

Comparison with N-body



Percent level agreement in boost-factors wrt LCDM up to k of a few
(even though the pure P(k) deviates at percent level at much larger scales)

COLA with massive neutrinos

Neutrinos are known to have mass and affect structure formation significantly on quasi-linear to non-linear scales (depending on the mass).

Large velocity dispersion of non-relativistic neutrinos prevent clustering on small scales -> suppresses growth of structures -> scale dependent growth.

Can roughly be though of as a scale-dependent Newton's constant so it fits directly in to the picture we had above

$$\mu(k,a) = 1 \qquad k \ll k_{\rm fs}$$

$$\mu(k,a) = \frac{\Omega - \Omega_\nu}{\Omega} \qquad k \gg k_{\rm fs}$$

Massive neutrinos often highly degenerate with MG so important to take into account.

COLA with massive neutrinos

For the particle mesh part of the code we use the grid-method of (**Brandbyge and Hannestad 2008**). Neutrinos are kept in Fourier space during the simulation and evolved lineary. When needed we simply add it to the source of the Poisson equation in Fourier space

$$-k^2\hat{\Phi}(\vec{k},a) = \frac{3}{2}\Omega_m a \left[\frac{\Omega_{\rm cdm}}{\Omega_m}\delta_{\rm cdm}(\vec{k},a) + \frac{\Omega_\nu}{\Omega_m}\delta_\nu(\vec{k},a)\right]$$

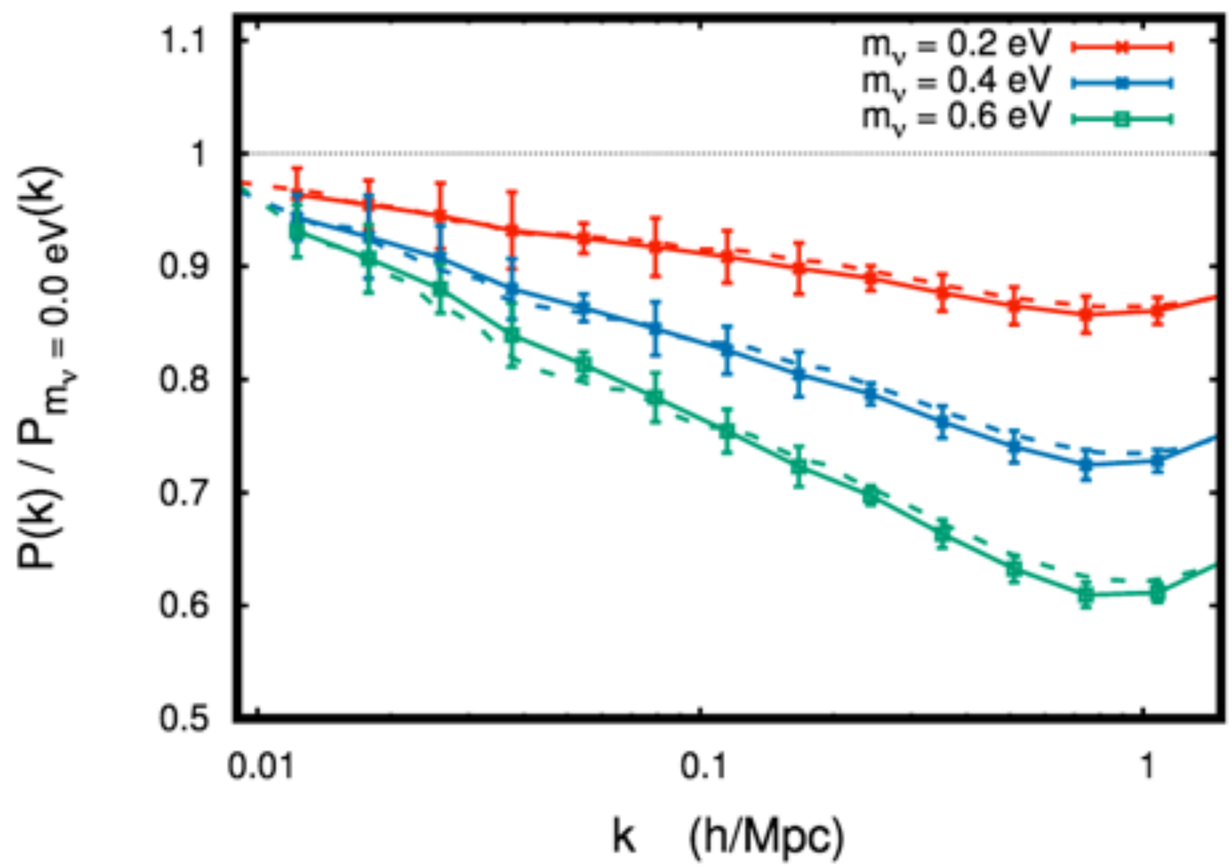
where

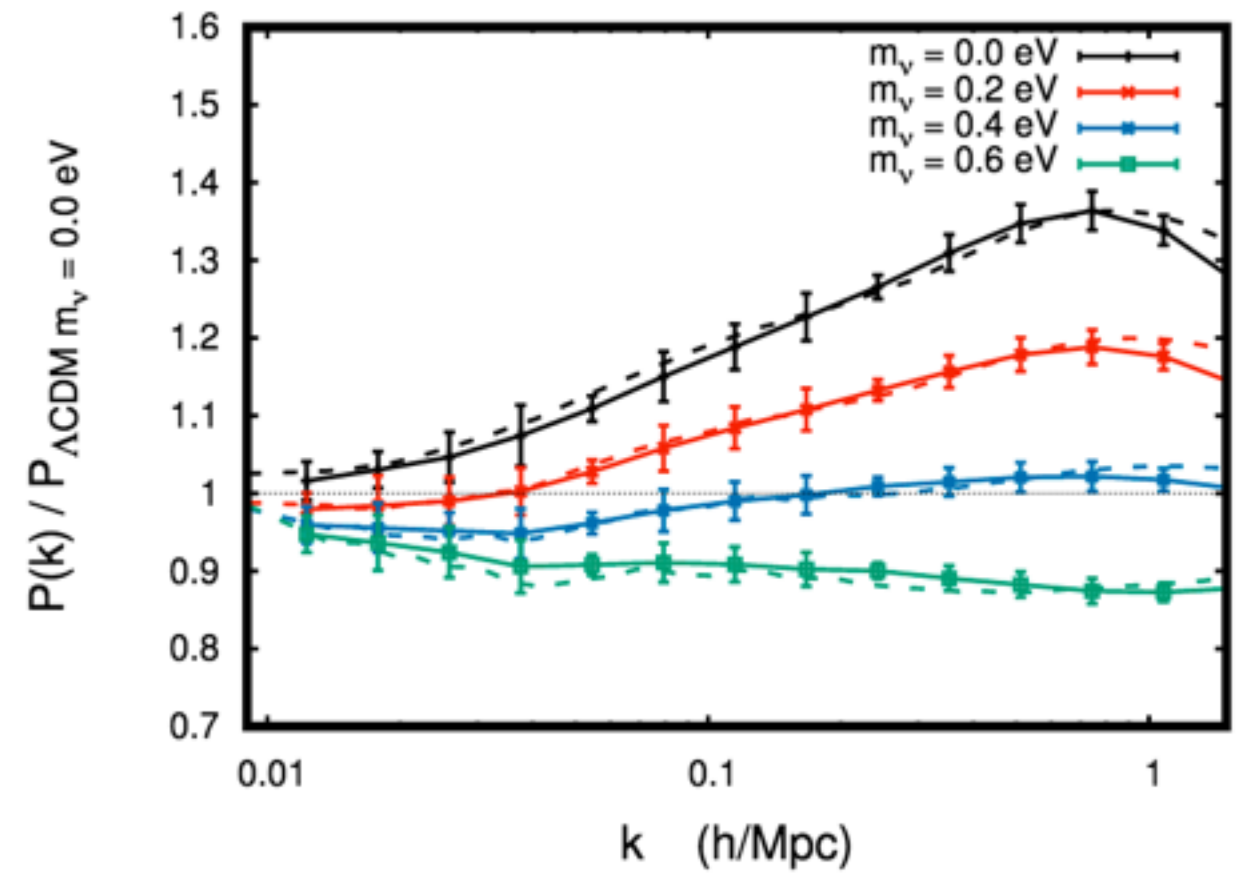
$$\delta_\nu(\vec{k},a) = \delta_\nu(\vec{k},a_{\rm ini})\frac{D_\nu(k,a)}{D_\nu(k,a_{\rm ini})}$$

and where we use the same initial seed as for CDM to generate the initial neutrino perturbations.

For the exact details see **Wright, Winther, Koyama 2017**.

Comparison to N-body with neutrinos





LCDM (left) and f(R) (right) compared to full N-body simulations with massive neutrinos.

The right plot illustrates the degeneracy of massive neutrinos (suppresses growth) and modified gravity (enhances growth).

Summary

We have extended the COLA method to models with scale dependent growth. Depending on the model in question this can be done almost as fast as for LCDM down to a factor of 4-5 slower (but still a huge speedup compared to N-body).

Scale-dependent COLA can also be used for accurate modelling of the effects of massive neutrinos in both LCDM and MG.

Very useful for mock generation, for studying clustering statistics (P(k), RSD etc.) and boost-factors wrt LCDM (even down to fairly non-linear scales).

A public available code can be found at

[https://github.com/HAWinther/MG-PICOLA-PUBLIC]

Comes with models such as f(R), DGP / cubic galileon, symmetron, general (m(a),beta(a)) models, JBD, … and contains support for three common methods of screening. Can do a lot of analysis on the fly (P(k), RSD multipoles, halofinding, …)