Higher order Hamiltonian Monte Carlo Sampling for Cosmological Large Scale Structure Analysis

Mónica Hernández Sánchez

Advisor:
Francisco-Shu Kitaura

Collaborators:
Metin Ata
Sergio Rodríguez Torres
Claudio Dalla Vecchia

Universidad de La Laguna
Instituto de Astrofísica de Canarias · IAC

VII MEETING ON FUNDAMENTAL COSMOLOGY
Madrid 9/09/19

CONTENTS

1. Introduction
   1.1. Motivation and Objectives
   1.2. The BIRTH code

2. Bayesian Statistics
   2.1. Hamiltonian Monte Carlo sampling

3. Fourth order Leapfrog algorithm
   3.1. Study of the i parameter and the stepsize
   3.2. Convergence of the method

4. BIRTH Results

5. Conclusions

1. INTRODUCCION

1.1. Motivation and objectives

- Structures have been formed from the gravitational instability of primordial fluctuations, which are assumed to be closely Gaussian distributed; generating a process of collapse and originating the Cosmic Web (non-Gaussian).

- Galaxies are biased discrete tracers of the dark matter field.

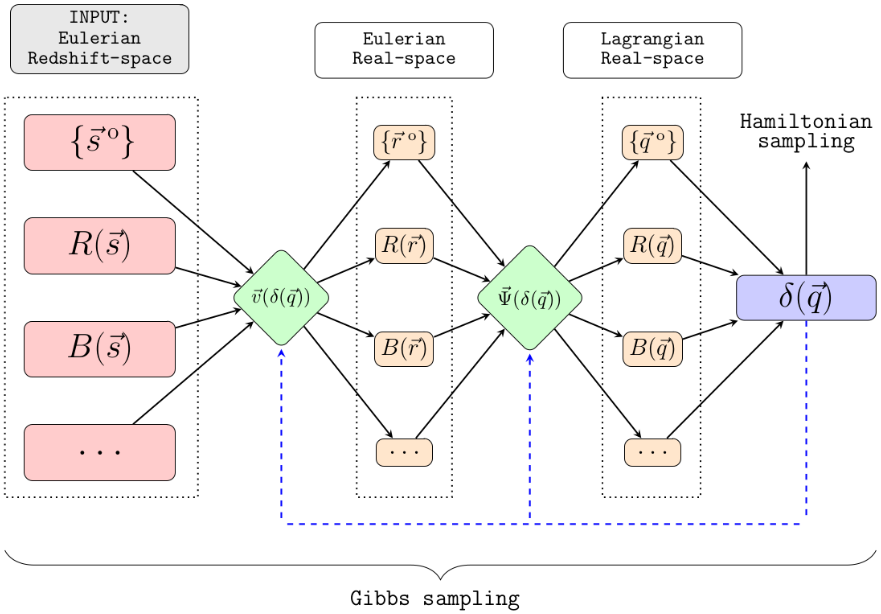
- We aim at reconstructing the initial conditions of the Universe from Galaxy catalogues with a statistical Bayesian framework: Hamiltonian Monte Carlo sampling.

- We aim at improving the efficiency of the method through the implementation of a higher order discretization of the equations of motion: the **fourth order Leapfrog algorithm.**

1. INTRODUCCION
1.2. BIRTH Code

Bayesian **I**nference for **R**eality vs **TH**eory

- Bayesian inference algorithm to reconstruct the primordial evolved cosmic density field from galaxy surveys on the light-cone.
- General to any structure formation model.
- Self consistent treatment of the survey geometry and selection function.
- Non-linear Lagrangian bias.
- Redshift-space distortions modelling.
- Higher order Hamiltonian Monte Carlo sampling.



Kitaura et al. (Mónica Hernández-Sánchez) in preparation

1. INTRODUCCION
1.2. BIRTH Code

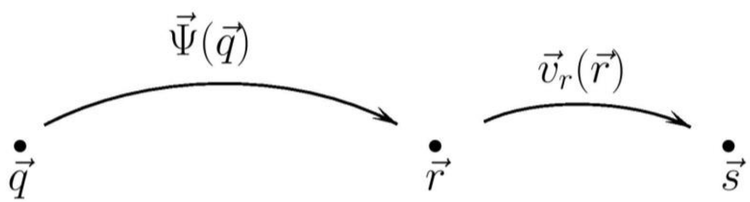
Nested Gibbs-Hamiltonian sampling

- In a first step we assume the data is in Lagrangian real-space at high redshift:
  - Power-law bias
  - Almost Poisson distribution of galaxies: Poissonian likelihood
  - Lognormal prior

$$\delta(q) \curvearrowleft \mathscr{P}\left(\delta(q)|\{q\},\{b\},\{R\}\right)$$

- In a second step: forward modelling to obtain S:

$$\{q\} \curvearrowleft \mathscr{P}\left(\{q\}|\{s^o\},\delta(q),\mathcal{M}\right)$$



  - Likelihood comparison: $s \longleftrightarrow s^o$

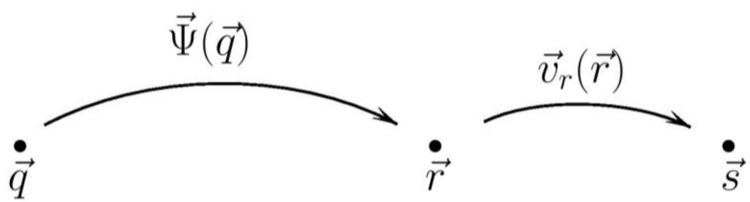
Nested Gibbs-Hamiltonian sampling

- In a first step we assume the data is in Lagrangian real-space at high redshift:
  - Power-law bias
  - Almost Poisson distribution of galaxies: Poissonian likelihood
  - Lognormal prior

$$\delta(q) \curvearrowleft \mathscr{P}\left(\delta(q)|\{q\},\{b\},\{R\}\right)$$

- In a second step: forward modelling to obtain s:

$$\{q\} \curvearrowleft \mathscr{P}\left(\{q\}|\{s^o\},\delta(q),\mathcal{M}\right)$$



  - Likelihood comparison: $s \longleftrightarrow s^o$

2. BAYESIAN STATISTICS

2.1. Hamiltonian Monte Carlo Sampling

$$\mathscr{H}(q,p) = U(q) + K(p) \qquad K(p) = \frac{1}{2}p^T M^{-1} p$$

$q \longrightarrow$ variable we want to sample: primordial fluctuations

$p \longrightarrow$ artificially introduced to evolve the system

We can write: $U(q) = -\ln \mathscr{P}(q)$

As we are interested on evolving the system with the momenta we use:

$$e^{-H} = e^{-K} e^{-U} \quad \begin{cases} e^{-K} = e^{-\frac{1}{2}p^T M^{-1} p} \\ e^{-U} = \mathscr{P}(q) \end{cases}$$

We evolve the system solving the Hamilton's equations:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} = M^{-1} p_i$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} = -\frac{\partial U}{\partial q_i}$$

(Jasche & Kitaura, 2010)

2. BAYESIAN STATISTICS
2.1. Hamiltonian Monte Carlo Sampling

We discretize them with the Leapfrog algorithm:

$$\begin{aligned} p_i\left(\tau+\frac{\epsilon}{2}\right) &= p_i(\tau)-\frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(\tau)) \\ q_i(\tau+\epsilon) &= q_i(\tau)+\epsilon\frac{p_i\left(\tau+\frac{\epsilon}{2}\right)}{m_i} \\ p_i(\tau+\epsilon) &= p_i\left(\tau+\frac{\epsilon}{2}\right)-\frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(\tau+\epsilon)) \end{aligned}$$

We accept or reject the steps with the Metropolis-Hastings criterion

- Second order Leapfrog: $T_\epsilon = T_p(\epsilon/2)T_q(\epsilon)T_p(\epsilon/2)$
- Higher order Leapfrog: $T_{n+2}((2i-s)\epsilon) = T_n(\epsilon)^i T_n(-s\epsilon)T_n(\epsilon)^i$

$$s\epsilon = (2i)^{1/(n+1)}\epsilon$$

We discretize them with the Leapfrog algorithm:

$$p_i\left(\tau+\frac{\epsilon}{2}\right) = p_i(\tau) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(\tau))$$

$$q_i(\tau+\epsilon) = q_i(\tau) + \epsilon\frac{p_i\left(\tau+\frac{\epsilon}{2}\right)}{m_i}$$

$$p_i(\tau+\epsilon) = p_i\left(\tau+\frac{\epsilon}{2}\right) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(\tau+\epsilon))$$

We accept or reject the steps with the Metropolis-Hastings criterion

- Second order Leapfrog: $T_\epsilon = T_p(\epsilon/2)T_q(\epsilon)T_p(\epsilon/2)$
- Higher order Leapfrog: $T_{n+2}((2i-s)\epsilon) = T_n(\epsilon)^i T_n(-s\epsilon)T_n(\epsilon)^i$

Mónica Hernández-Sánchez (Kitaura) et al. in preparation

$$s\epsilon = (2i)^{1/(n+1)}\epsilon$$

3. FOURTH ORDER LEAPFROG ALGORITHM

3. 1. Study of the i parameter and the stepsize

Stepsize	Iteration of convergence	Convergence time	Acceptance
i=1			
ϵ	650	$108,29\ min$	$95,0\%$
2ϵ	250	$45,53\ min$	$70,2\%$
4ϵ	230	$73,47\ min$	$35,2\%$
6ϵ	250	$110,21\ min$	$23,8\%$
8ϵ	260	$148,32\ min$	$13,8\%$
10ϵ	230	$189,09\ min$	$12,6\%$
i=2			
ϵ	68	$18,95\ min$	$94,6\%$
2ϵ	53	$23,85\ min$	$64,2\%$
4ϵ	46	$36,05\ min$	$36,2\%$
6ϵ	36	$50,06\ min$	$23,4\%$
8ϵ	46	$73,40\ min$	$18,0\%$
10ϵ	40	$106,48\ min$	$16,8\%$
i=3			
ϵ	32	$19,05\ min$	$88,4\%$
2ϵ	29	$25,54\ min$	$51,6\%$
4ϵ	20	$25,97\ min$	$27,4\%$
6ϵ	24	$43,54\ min$	$17,0\%$
8ϵ	25	$67,26\ min$	$13,8\%$
10ϵ	27	$67,27\ min$	$26,0\%$

For the second order Leapfrog was found to be optimal a stepsize $\boldsymbol{\epsilon = 0.06}$

3. FOURTH ORDER LEAPFROG ALGORITHM

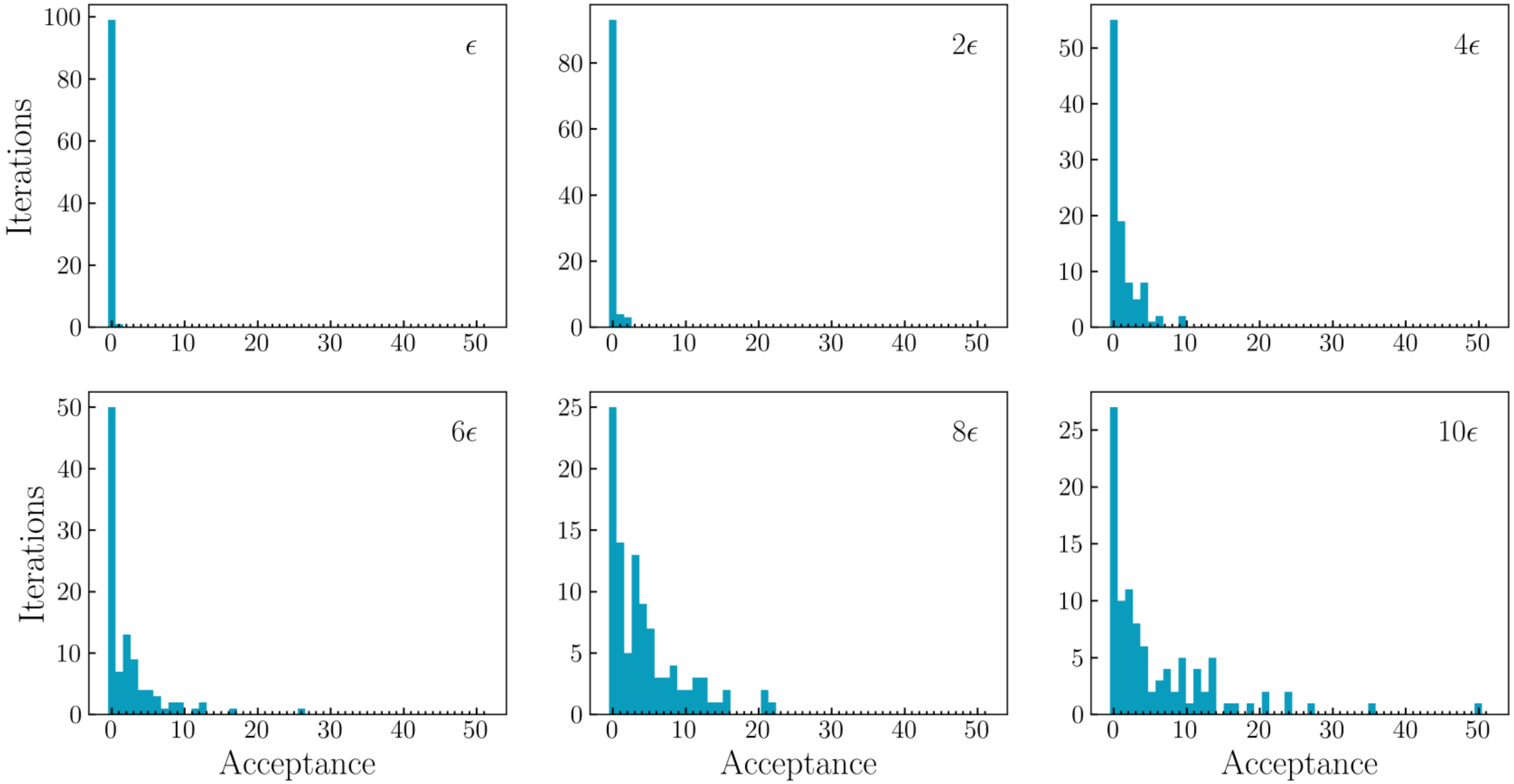
3. 1. Study of the *i* parameter and the stepsize

Stepsize	Iteration of convergence	Convergence time	Acceptance
i=1			
ϵ	650	$108,29\ min$	$95,0\%$
2ϵ	250	$45,53\ min$	$70,2\%$
4ϵ	230	$73,47\ min$	$35,2\%$
6ϵ	250	$110,21\ min$	$23,8\%$
8ϵ	260	$148,32\ min$	$13,8\%$
10ϵ	230	$189,09\ min$	$12,6\%$
i=2			
ϵ	68	$18,95\ min$	$94,6\%$
2ϵ	53	$23,85\ min$	$64,2\%$
4ϵ	46	$36,05\ min$	$36,2\%$
6ϵ	36	$50,06\ min$	$23,4\%$
8ϵ	46	$73,40\ min$	$18,0\%$
10ϵ	40	$106,48\ min$	$16,8\%$
i=3			
ϵ	32	$19,05\ min$	$88,4\%$
2ϵ	29	$25,54\ min$	$51,6\%$
4ϵ	20	$25,97\ min$	$27,4\%$
6ϵ	24	$43,54\ min$	$17,0\%$
8ϵ	25	$67,26\ min$	$13,8\%$
10ϵ	27	$67,27\ min$	$26,0\%$

3. FOURTH ORDER LEAPFROG ALGORITHM

3. 1. Study of the i parameter and the stepsize

- Acceptance: $i = 1$



3. FOURTH ORDER LEAPFROG ALGORITHM

3. 2. Convergence of the Method

- Comparison between second and fourth order Leapfrgog algorithm:

	Iteration of convergence	Convergence time (h)	Acceptance
$2^{o} order$ ϵ	2500	$55,81$	$52,0\%$
4^{o} order $2\epsilon,\ i=1$	340	$13,29$	$51,75\%$
4^{o} order $\epsilon,\ i=2$	100	$6,96$	$83,75\%$
4^{o} order $\epsilon,\ i=3$	33	$3,12$	$78,75\%$

3. FOURTH ORDER LEAPFROG ALGORITHM

3. 2. Convergence of the Method

- Comparison between second and fourth order Leapfrgog algorithm:

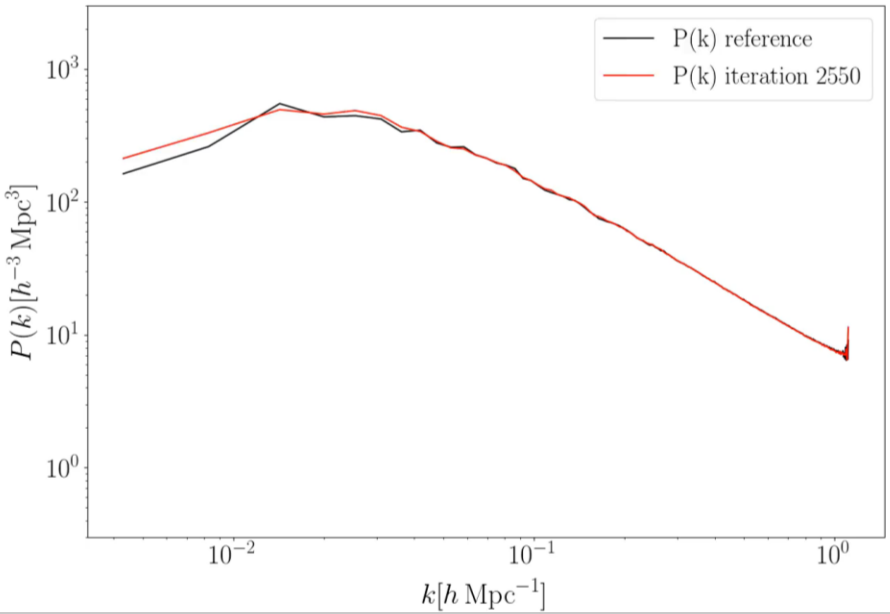
	Iteration of convergence	Convergence time (h)	Acceptance
$2^{o} order$ ϵ	2500	55, 81	52, 0%
4^{o} order $2\epsilon,\ i=1$	340	13, 29	51, 75%
4^{o} order $\epsilon,\ i=2$	100	6, 96	83, 75%
4^{o} order $\epsilon,\ i=3$	33	3, 12	78, 75%

Fourth order Leapfrog algorithm is **18 times** faster than the second order algorithm

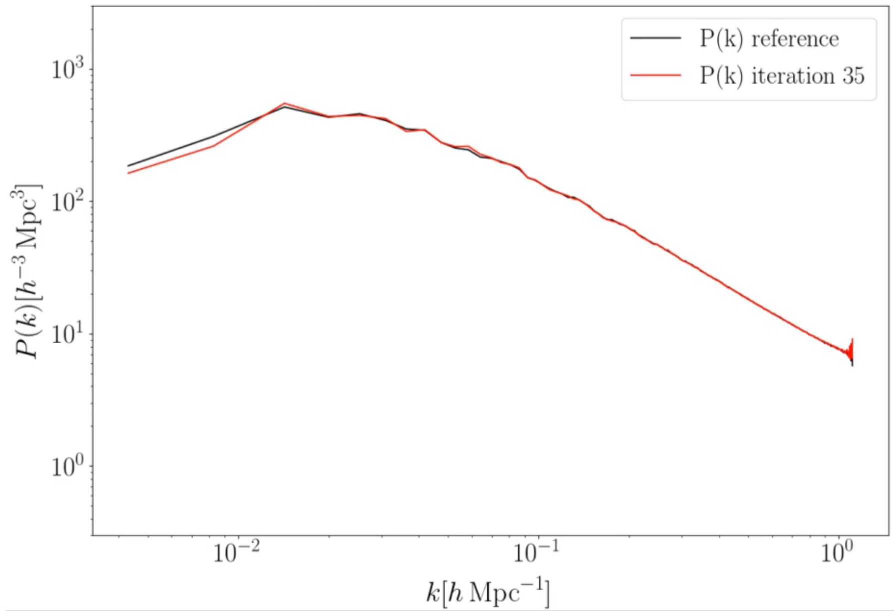
3. FOURTH ORDER LEAPFROG ALGORITHM

3. 2. Convergence of the Method

Second order Leapfrog



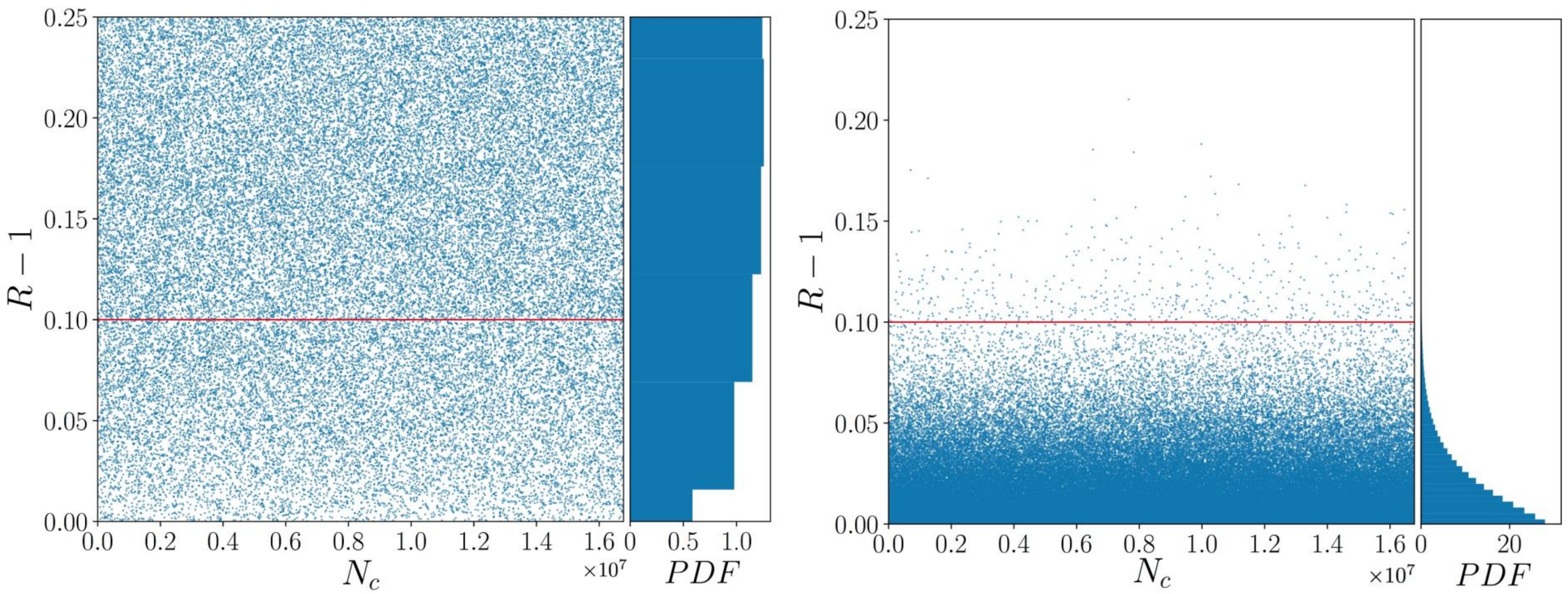
Fourth order Leapfrog



3. FOURTH ORDER LEAPFROG ALGORITHM

3. 2. Convergence of the Method

- Gelman-Rubin test:



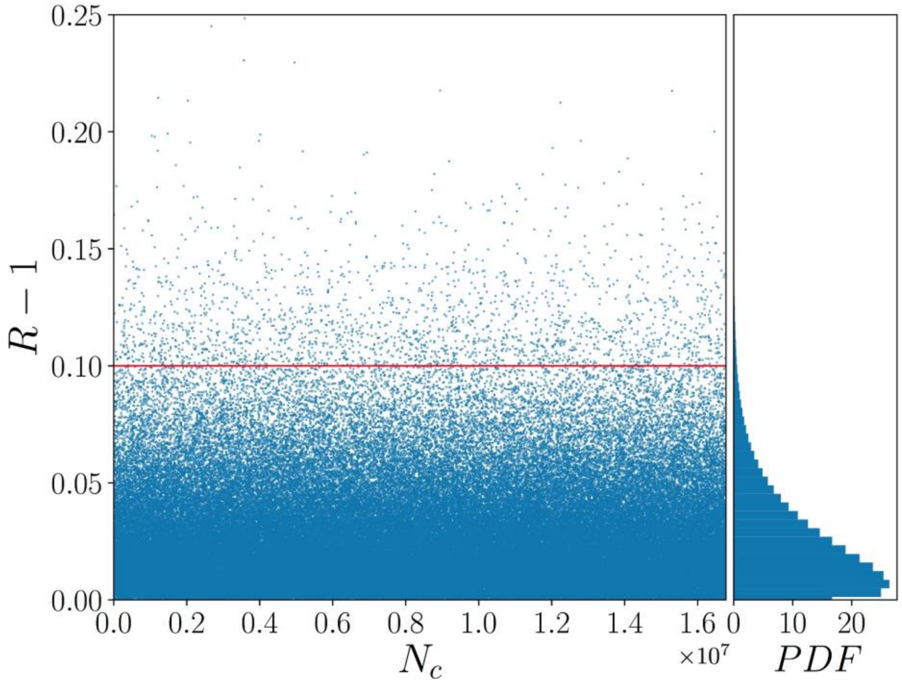
Second order Leapfrog
3000-3460 iterations

Fourth order Leapfrog
40-500 iterations

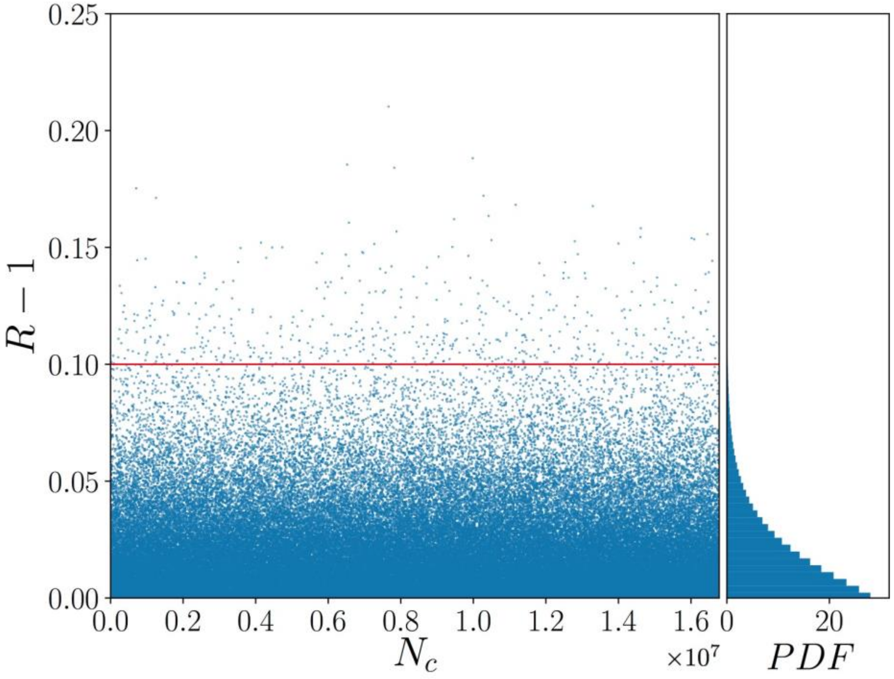
3. FOURTH ORDER LEAPFROG ALGORITHM

3. 2. Convergence of the Method

- Gelman-Rubin test:



Second order Leapfrog
3000-12000 iterations

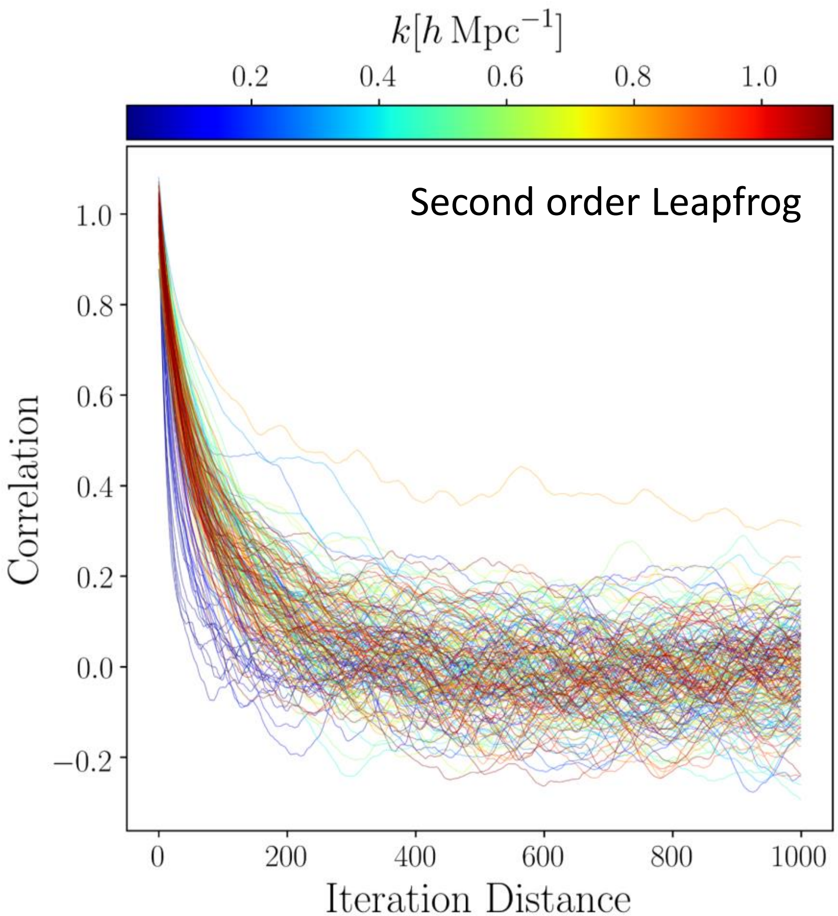


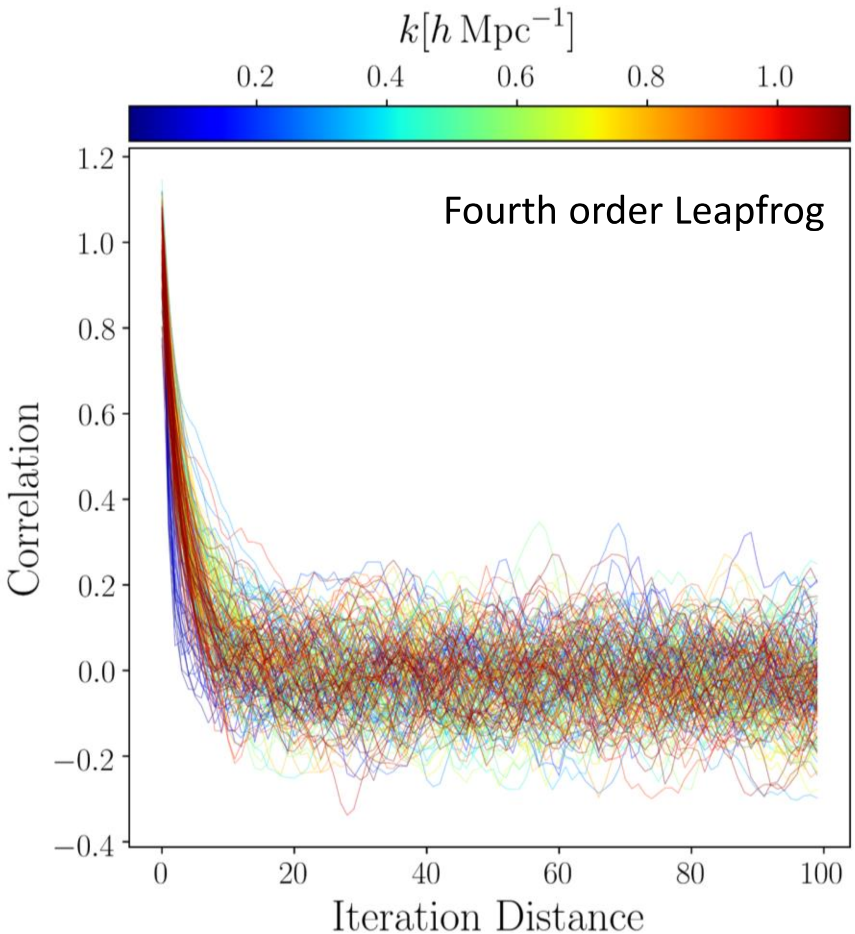
Fourth order Leapfrog
40-500 iterations

3. FOURTH ORDER LEAPFROG ALGORITHM

3. 2. Convergence of the Method

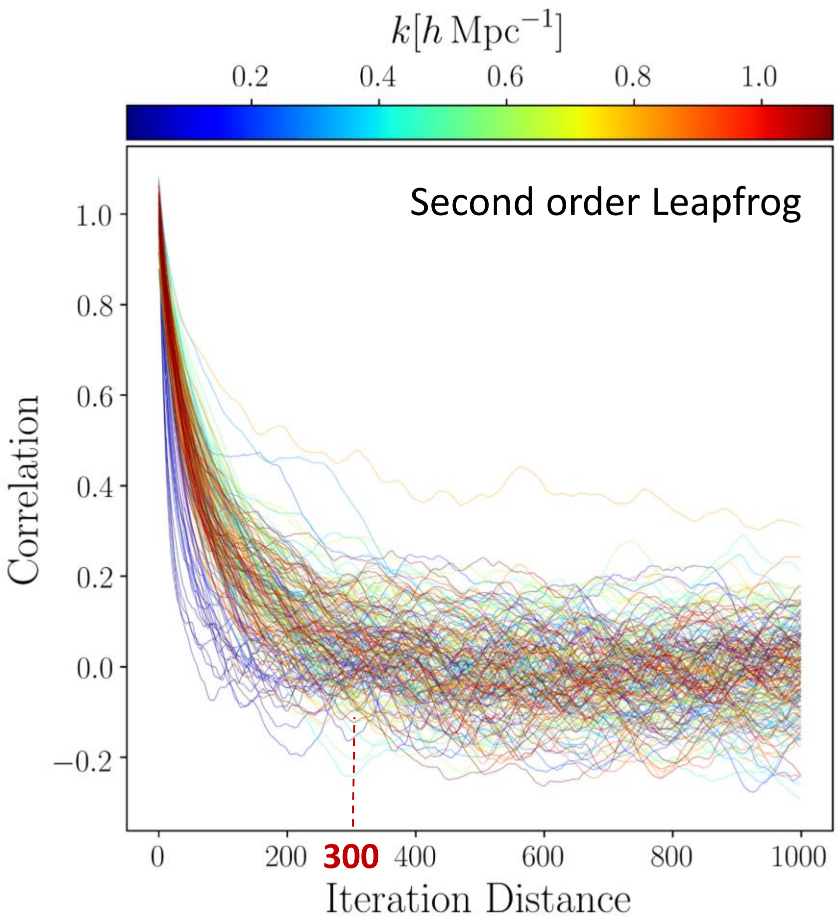
- Correlation Length:

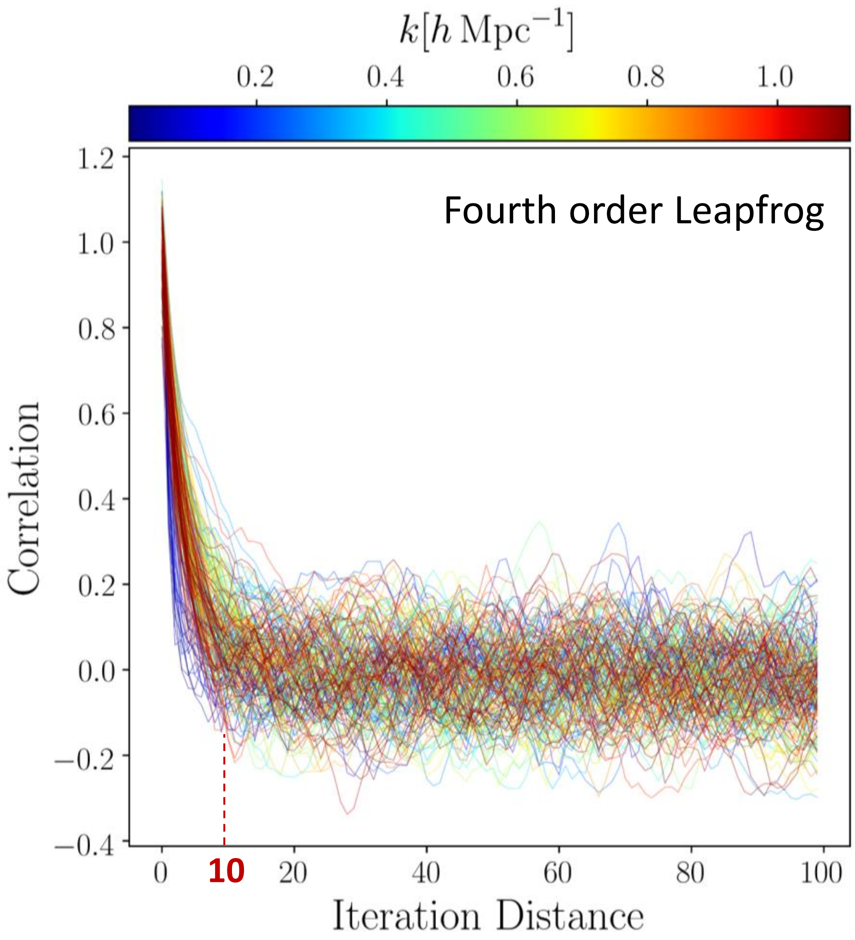




3. FOURTH ORDER LEAPFROG ALGORITHM
3. 2. Convergence of the Method

- Correlation Length:

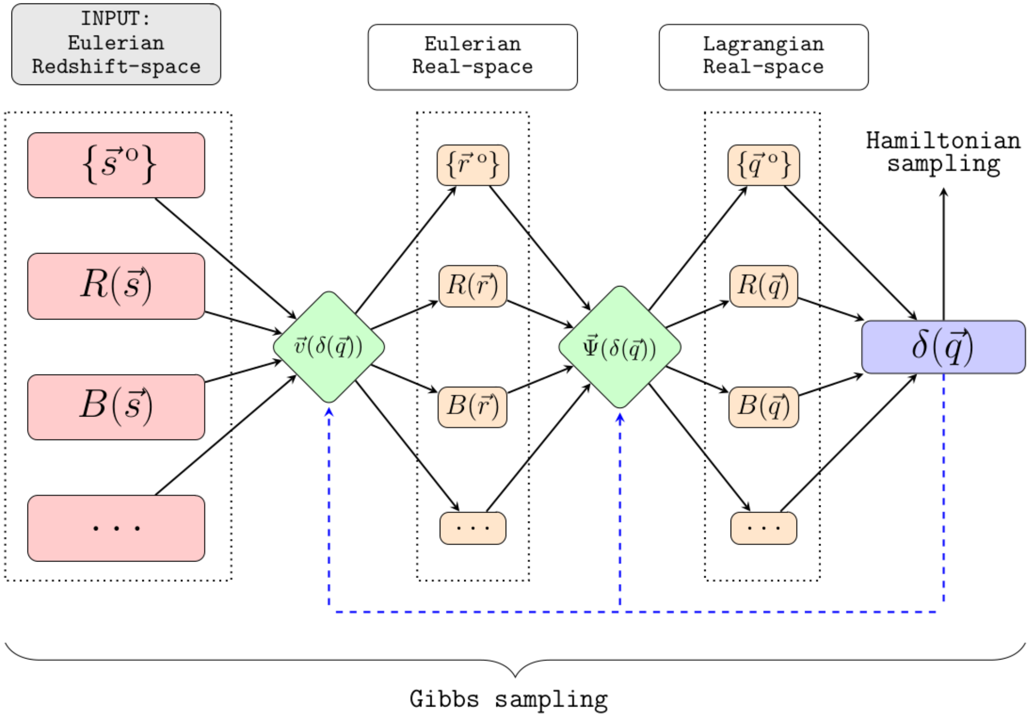




4. BIRTH RESULTS

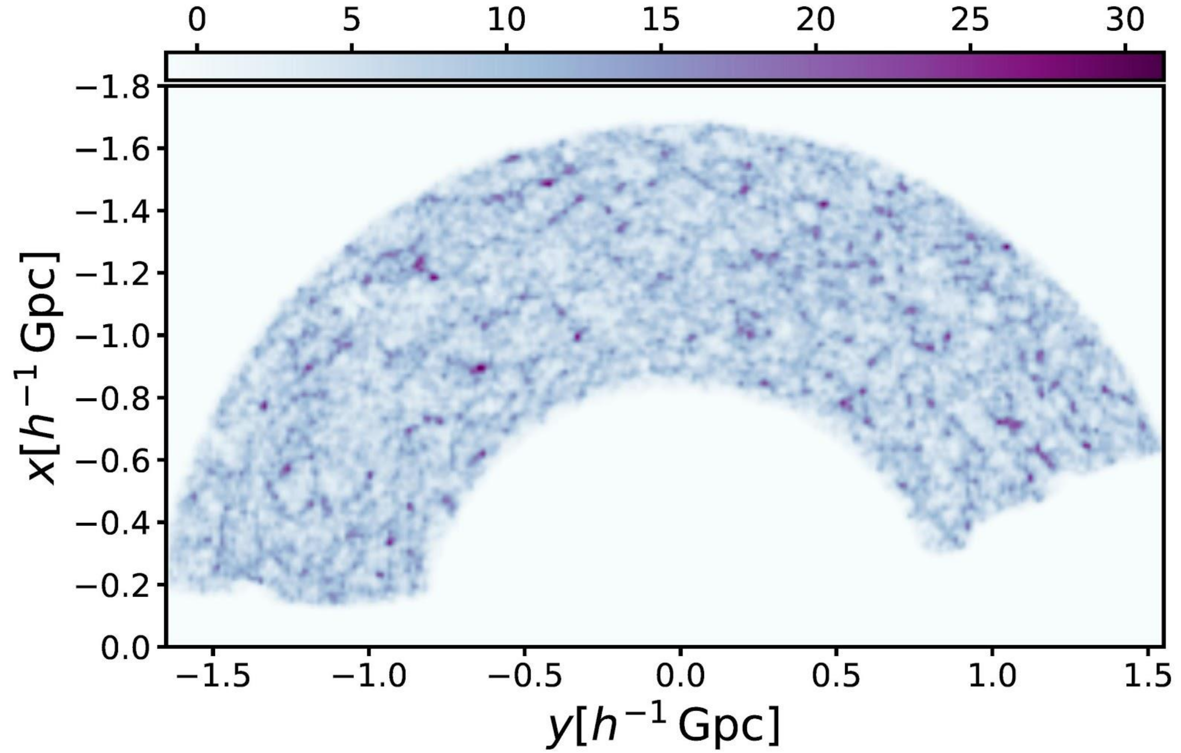
$$\delta(q) \curvearrowleft \mathscr{P}\left(\delta(q)|\{q\},\{b\},\{R\}\right)$$

$$\{q\} \curvearrowleft \mathscr{P}\left(\{q\}|\{s^o\},\delta(q),\mathcal{M}\right)$$



4. BIRTH RESULTS

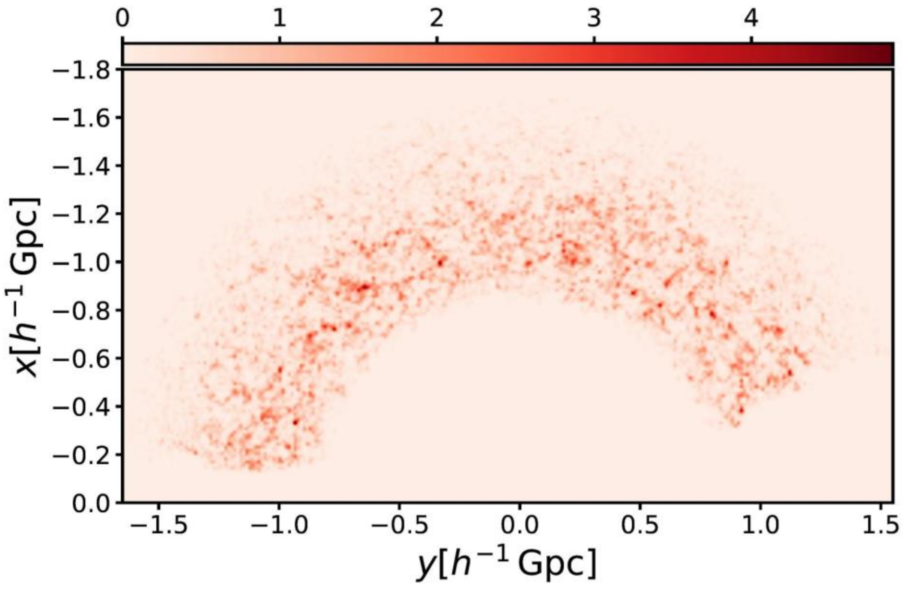
Dark matter from the BigMD simulation with light-cone evolution



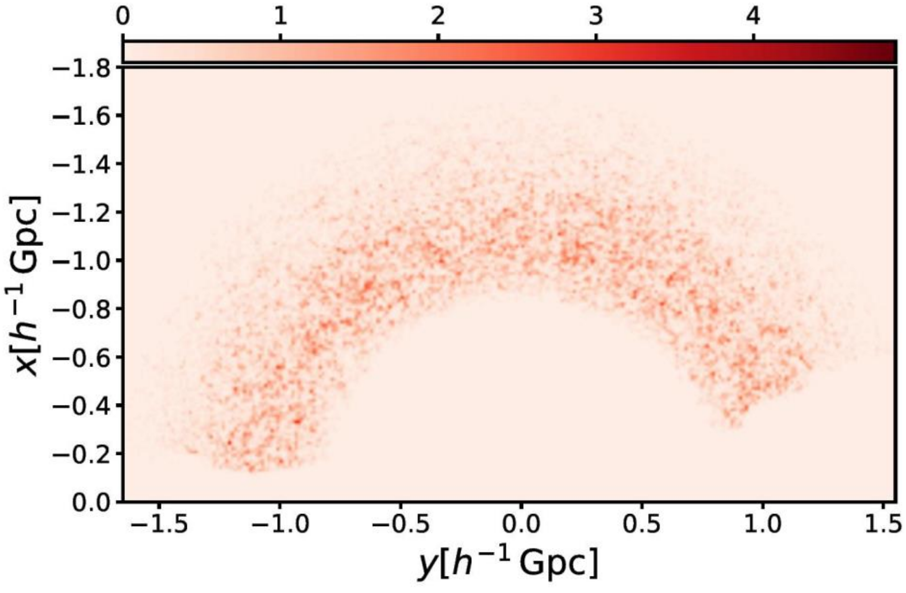
Klypin et al, 2016

4. BIRTH RESULTS

Galaxy number counts in Eulerian

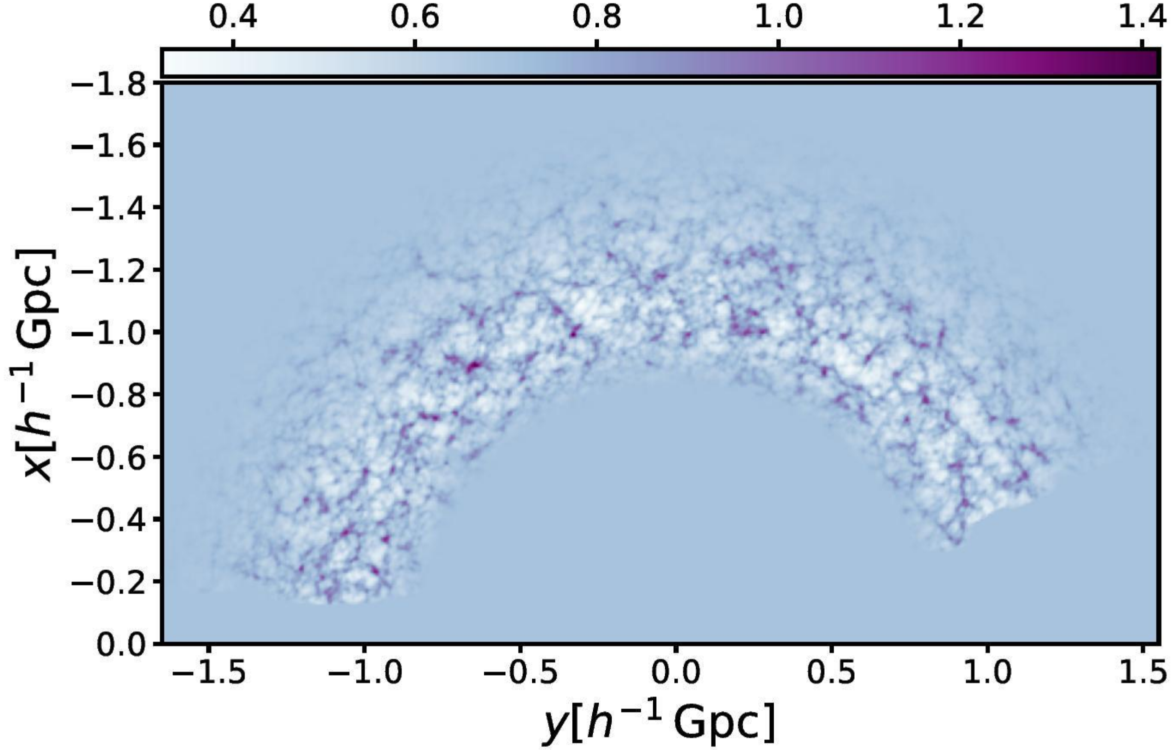


Galaxy number counts in Lagrangian



4. BIRTH RESULTS

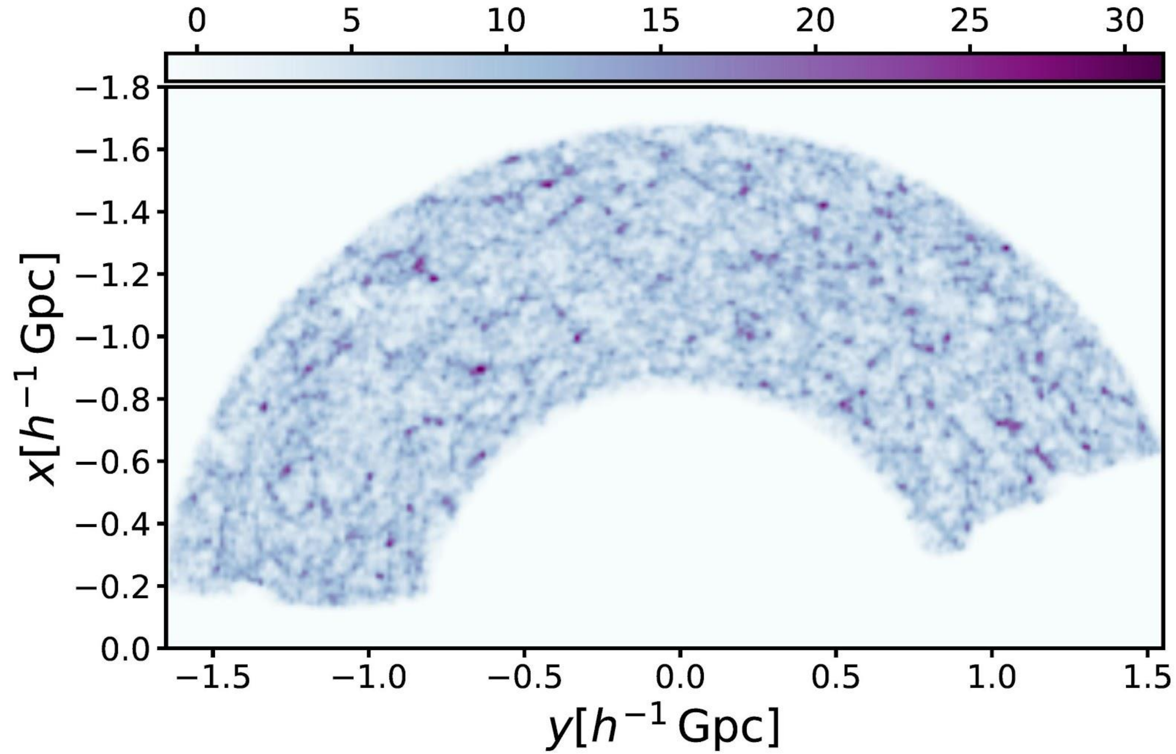
Dark matter reconstruction



Kitaura et al. (in prep)

4. BIRTH RESULTS

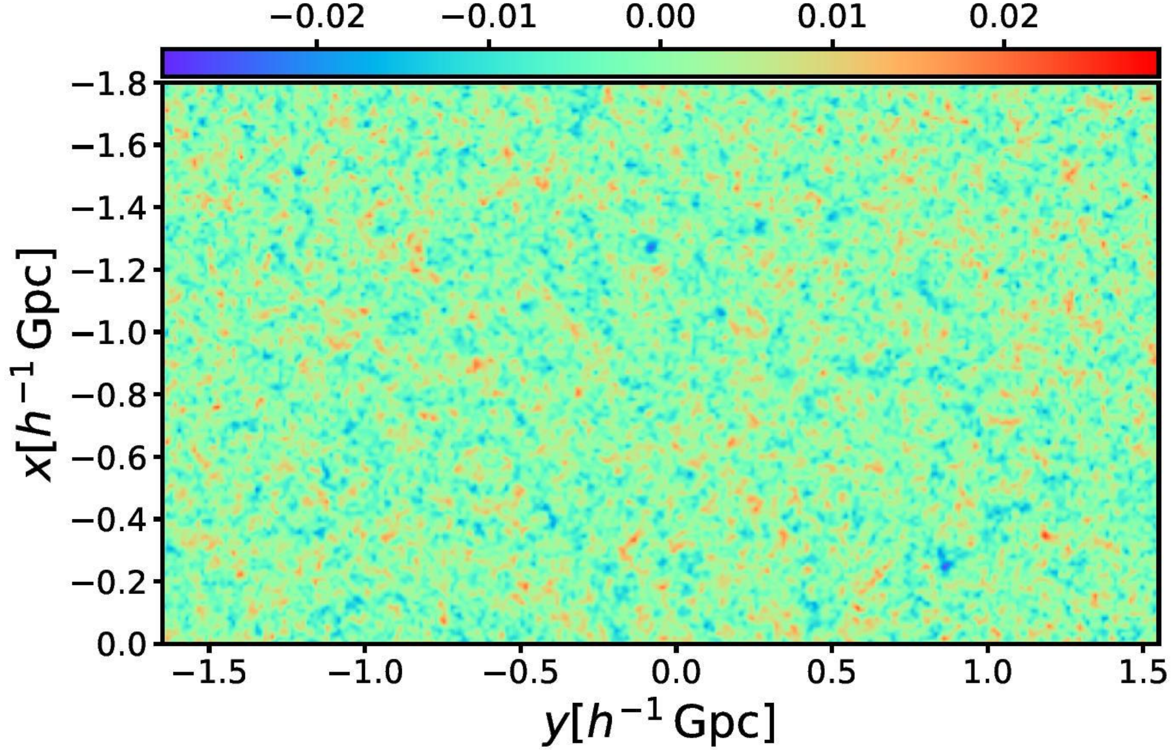
Dark matter from the BigMD simulation with light-cone evolution



Kitaura et al. (in prep)

4. BIRTH RESULTS

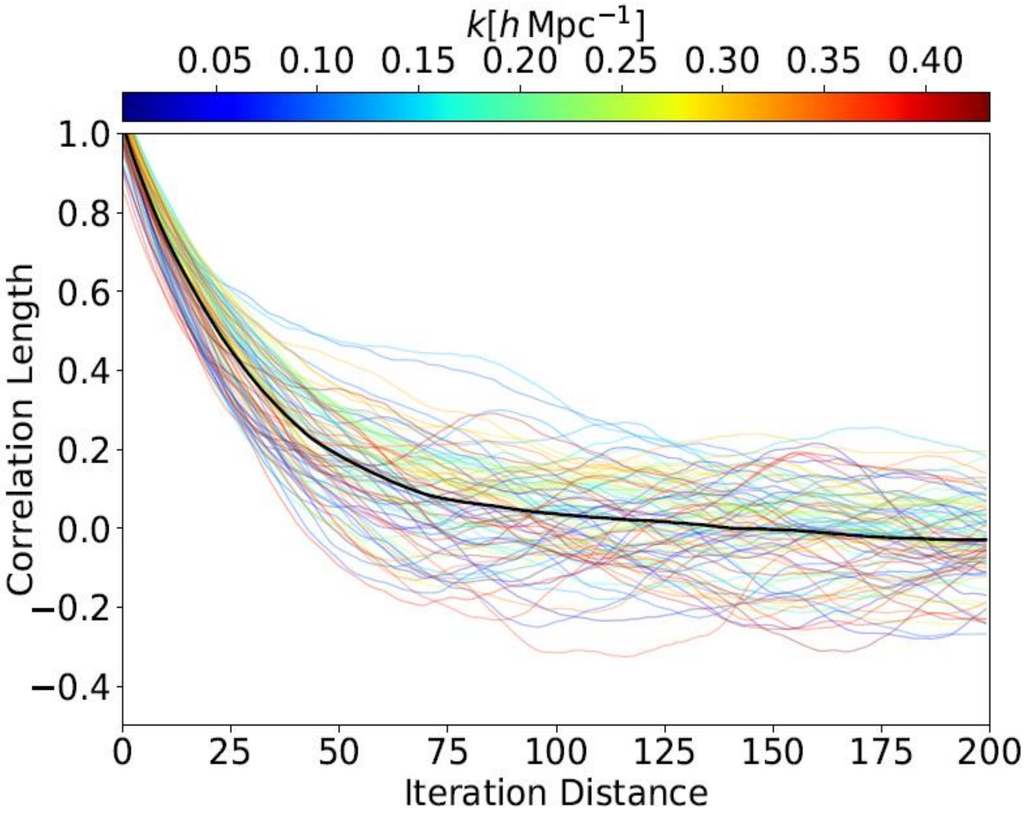
Sampled Gaussian field at z=100



Kitaura et al. (in prep)

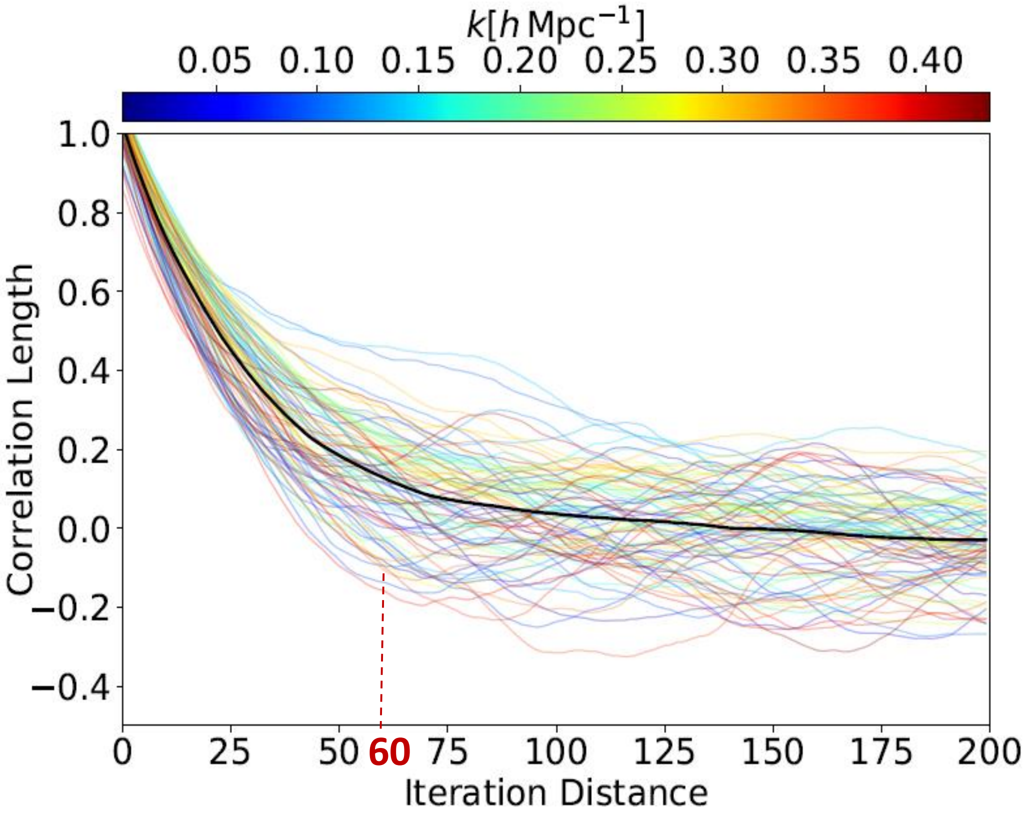
4. BIRTH RESULTS

- Correlation length:



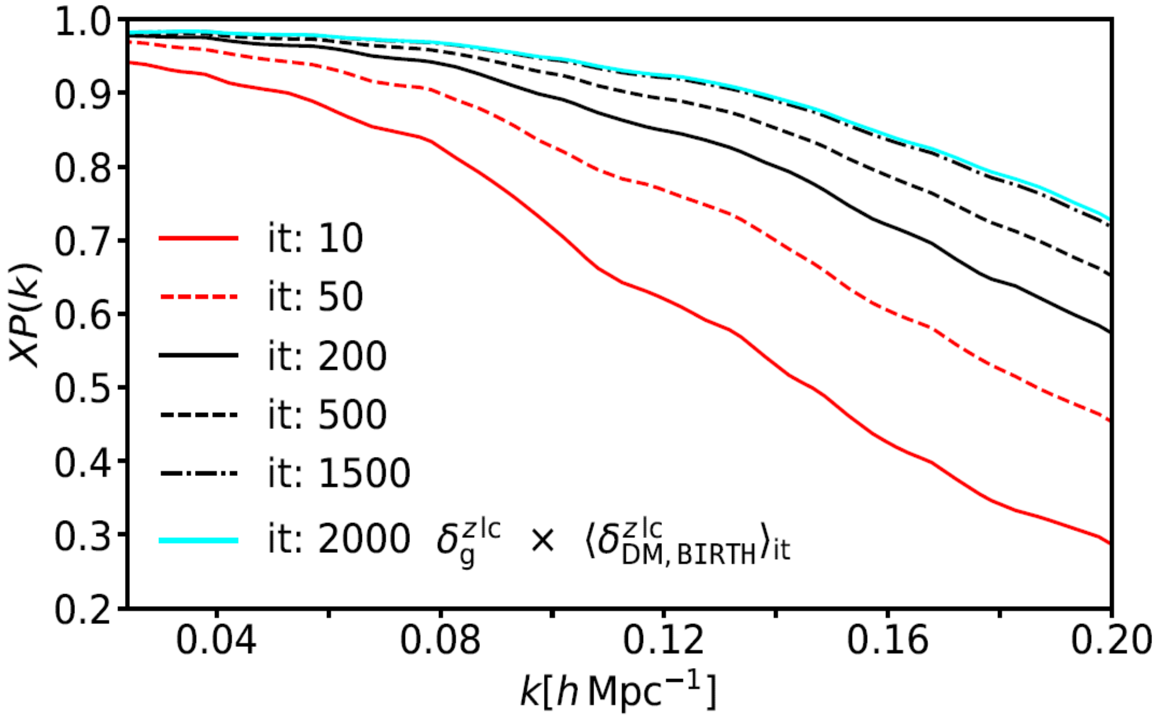
4. BIRTH RESULTS

- Correlation length:



4. BIRTH RESULTS

- Number of independent samples for a fair estimation of the posterior mean:



Kitaura et al. (in prep)

5. CONCLUSIONS

- We have implemented a **fourth order Leapfrog algorithm** for the discretizaton of the Hamilton's equations.

- Several tests have been developed to study the convergence, the computational time, the acceptance of the iterations and the correlation length:

  - We get convergence in ~ 30 iterations, **two orders of magnitude less** than with the second order algorithm.

  - We have reduced the computational time needed to reach the convergence a **factor** $\sim \mathbf{20}$.

  - We can obtain independent samples **each** $\mathbf{10}\boldsymbol{th}$ **iterations** as opposed to every 300 iterations with the old scheme.

- We have implemented this method in a realistic case with light-cone evolution, survey geometry, selection function, non-linar bias, RSD, displacements…

THANK YOU FOR YOUR ATTENTION