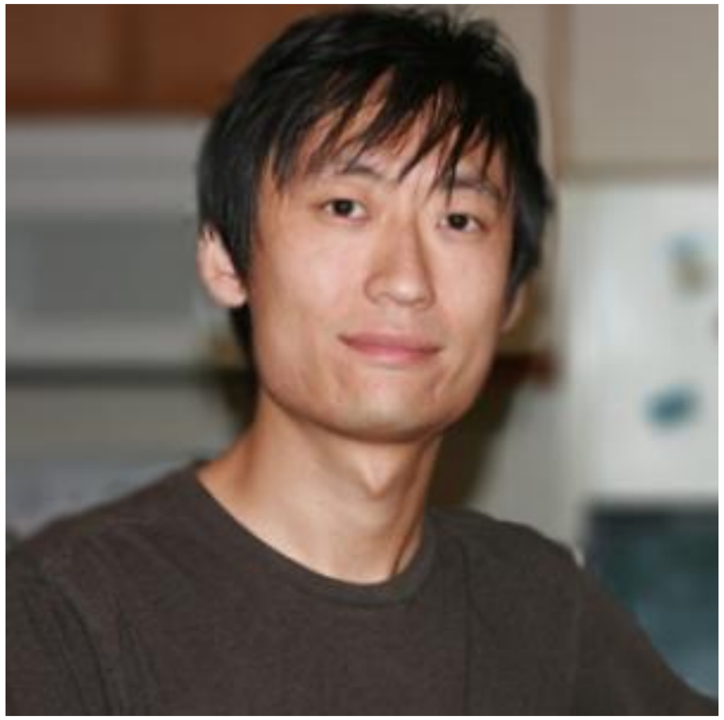
Sampling versus optimization in very high dimensional parameter spaces

Grigor Aslanyan
Berkeley Center for Cosmological Physics
UC Berkeley

Statistical Challenges in Modern Astronomy VI
Carnegie Mellon University
June 6, 2016

Collaborators

Uroš Seljak



Yu Feng

Chirag Modi

Sampling: Hamiltonian Monte Carlo (HMC)

- Each parameter becomes a “particle” position. Momentum variables are introduced. Particles follow Hamiltonian dynamics. U = -ln(posterior).

- Huge advantage over random walk: Information in the derivatives is used to walk “in the right direction”.

- Acceptance rate = 1 **theoretically**.

- For each iteration need to integrate equations of motion numerically using staggered leapfrog (or similar) methods. Typically ~10 numerical integration steps are taken per iteration.

- Method of choice for sampling high dimensional parameter spaces.

- **Tuning**: masses, integration steps, integration time.

Optimization: BFGS

Quasi-Newton method.

Needs the first, but not second derivatives.

At each iteration the inverse Hessian is estimated from previous iterations (never stored explicitly). A direction of move is deduced, followed by line search.

Line search: Moré-Theunte 1992

L-BFGS: Limited memory BFGS. Store and use only a few previous iterations. Works almost as well as BFGS!

Linear Model

data → $\mathbf{d} = \mathbf{R}\mathbf{s} + \mathbf{n}$ (↑ signal, ↖ noise)

$$\mathbf{S} = \left\langle \mathbf{s}\mathbf{s}^\dagger \right\rangle \quad \mathbf{N} = \left\langle \mathbf{n}\mathbf{n}^\dagger \right\rangle$$

$$\mathbf{C} \equiv \left\langle \mathbf{d}\mathbf{d}^\dagger \right\rangle = \mathbf{R}\mathbf{S}\mathbf{R}^\dagger + \mathbf{N}$$

Binning: $\mathbf{S}_l = \{s_{m_l}\} \ (m_l = 1, \ldots, M_l)$

$$\mathbf{Q}_l = \mathbf{R}\mathbf{\Pi}_l\mathbf{R}^\dagger$$

↑ projection matrix

$$\mathbf{R}\mathbf{S}\mathbf{R}^\dagger = \sum_l \Theta_l \mathbf{Q}_l$$

Likelihood: $\mathcal{L}(\mathbf{d}|\mathbf{\Theta}) = (2\pi)^{-N/2} \det(\mathbf{C})^{-1/2} \exp\left(-\frac{1}{2}\mathbf{d}^\dagger\mathbf{C}^{-1}\mathbf{d}\right)$

Minimum variance estimator (**Wiener Filter**): $\hat{\mathbf{s}} = \mathbf{S}\mathbf{R}^\dagger\mathbf{C}^{-1}\mathbf{d}$

For gaussian fields this is the same as the maximum probability estimator!

U.Seljak *astro-ph/9710269*

Linear Model

Fisher matrix: $$F_{ll'} = -\left\langle \frac{\partial^2 \ln \mathcal{L}(\mathbf{d}|\boldsymbol{\Theta})}{\partial \Theta_l \partial \Theta_{l'}} \right\rangle_{\hat{\boldsymbol{\Theta}}}$$

The inverse is an estimate of the covariance matrix of the parameters:

$$\left\langle \hat{\boldsymbol{\Theta}} \hat{\boldsymbol{\Theta}}^\dagger \right\rangle - \left\langle \hat{\boldsymbol{\Theta}} \right\rangle \left\langle \hat{\boldsymbol{\Theta}}^\dagger \right\rangle = \mathbf{F}^{-1}$$

Calculation: $$F_{ll'} = \frac{1}{2} tr \left(\mathbf{Q}_l \mathbf{C}^{-1} \mathbf{Q}_{l'} \mathbf{C}^{-1} \right)$$

Window: $$W_{ll'} = \frac{F_{ll'}}{\sum_{l'} F_{ll'}}$$

Power spectrum quadratic estimator:

$$\hat{\Theta}_l = \frac{1}{2} \sum_{l'} F^{-1}_{ll'} \left[\mathbf{d}^\dagger \mathbf{C}^{-1} \mathbf{Q}_{l'} \mathbf{C}^{-1} \mathbf{d} - b_{l'} \right]$$

$$b_l = tr \left[\mathbf{N} \mathbf{C}^{-1} \mathbf{Q}_l \mathbf{C}^{-1} \right]$$

$$\left\langle \hat{\Theta}_l \right\rangle = \sum_{l'} W_{ll'} \Theta_{l'}$$

U.Seljak *astro-ph/9710269*

Estimating Noise Bias and Fisher Matrix

Noise bias: simulate noise: $\mathbf{d}_n$ Pass through optimizer: $\hat{\mathbf{s}}_n$

$$b_l = \mathbf{\Pi}_l \mathbf{S}^{-1} \hat{\mathbf{s}}_n^\dagger \hat{\mathbf{s}}_n \mathbf{S}^{-1} \mathbf{\Pi}_l$$

Fisher matrix: simulate signal: $\mathbf{s}_s$ Pass through optimizer: $\hat{\mathbf{s}}_s$

For each bin l' simulate extra signal in that bin only: $\Delta \mathbf{s}_{l'}$

$\mathbf{s}_{l'} = \mathbf{s}_s + \Delta \mathbf{s}_{l'}$ Pass through optimizer: $\hat{\mathbf{s}}_{l'}$

$$F_{ll'} = \frac{K_{l'}}{2\Theta_l^2} \left\langle \frac{\sum_{k_l} |\Delta \hat{s}_l(k_l)|^2}{\sum_{k_{l'}} |\Delta s_{l'}(k_{l'})|^2} \right\rangle$$

with $\Delta \hat{\mathbf{s}}_{l'} = \hat{\mathbf{s}}_{l'} - \hat{\mathbf{s}}_s$

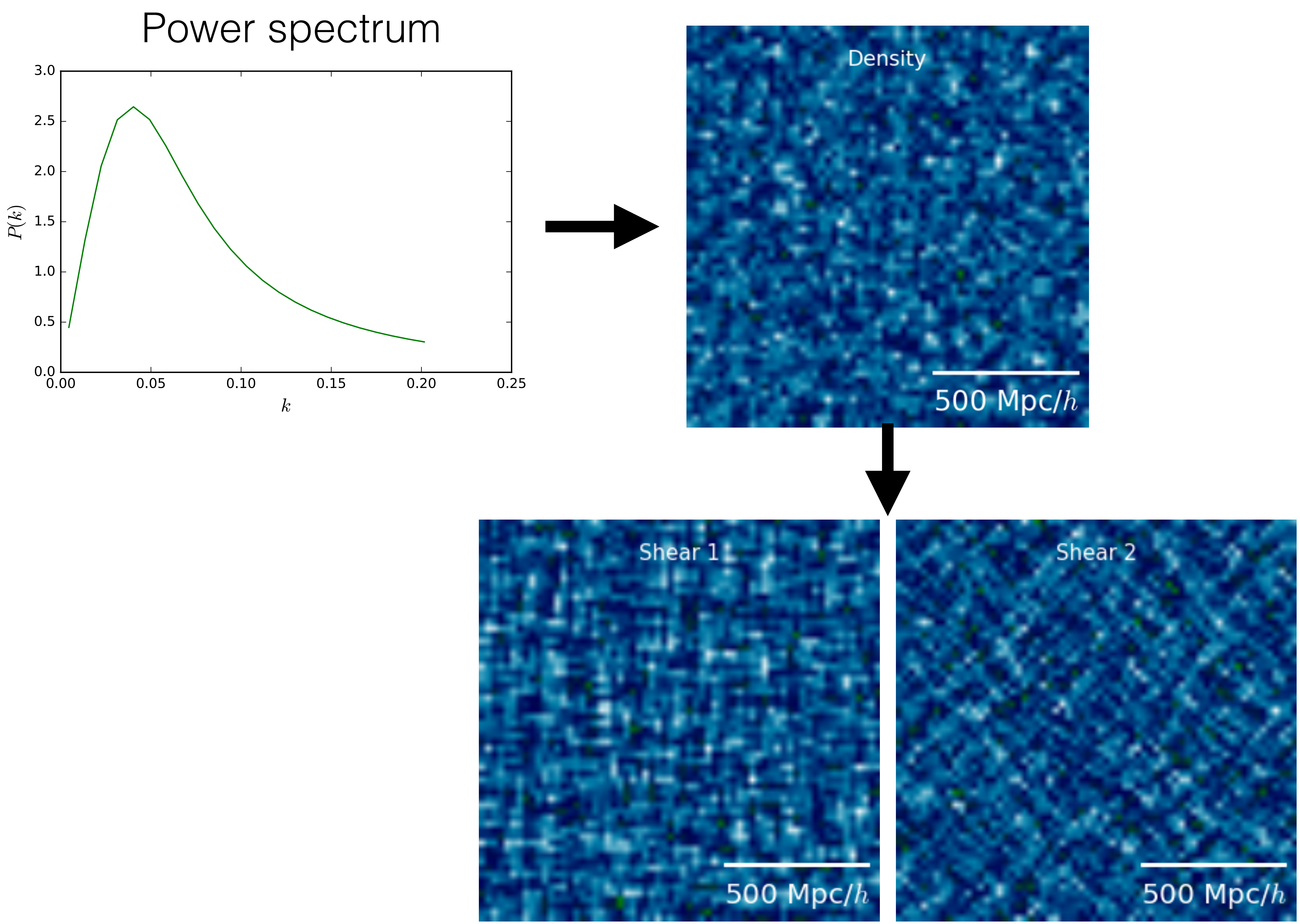
$K_{l'}$ is the number of modes

Power spectrum:

$$\hat{\Theta}_l = \sum_{k_l} \left(|\hat{s}(k_l)|^2 - \langle |\hat{s}_n(k_l)|^2 \rangle \right) \left\langle \frac{\sum_{l'} W_{ll'} K_{l'}^{-1} \sum_{k_{l'}} |s_s(k_{l'})|^2}{\sum_{k_l} |\hat{s}_s(k_l)|^2} \right\rangle$$

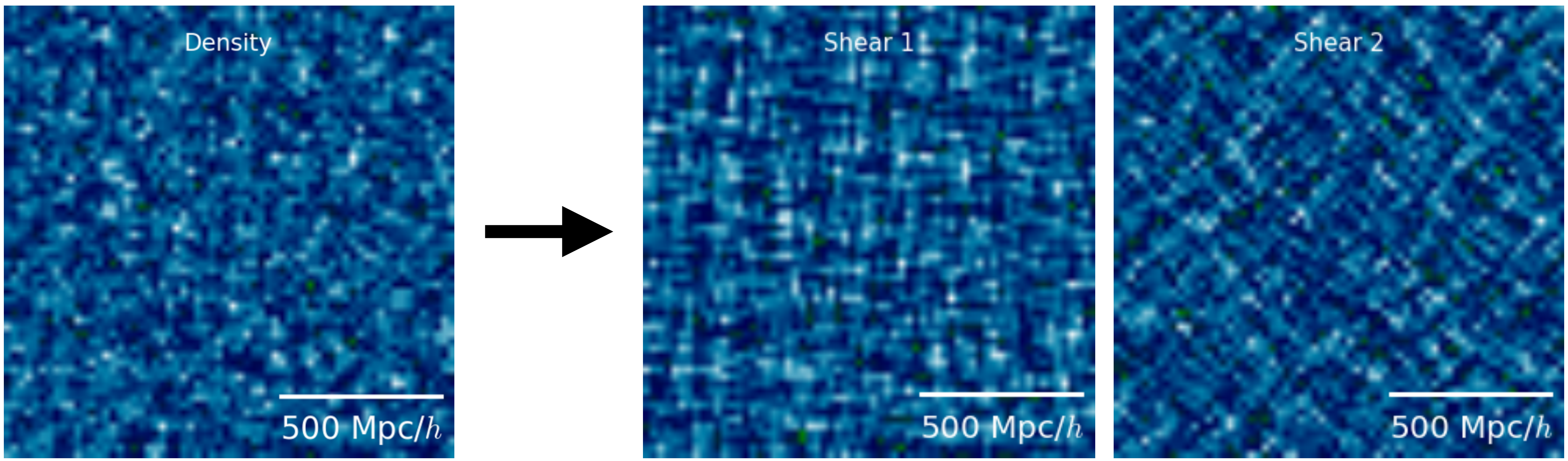
Linear case: Weak Lensing

Toy Example: 64x64 grid



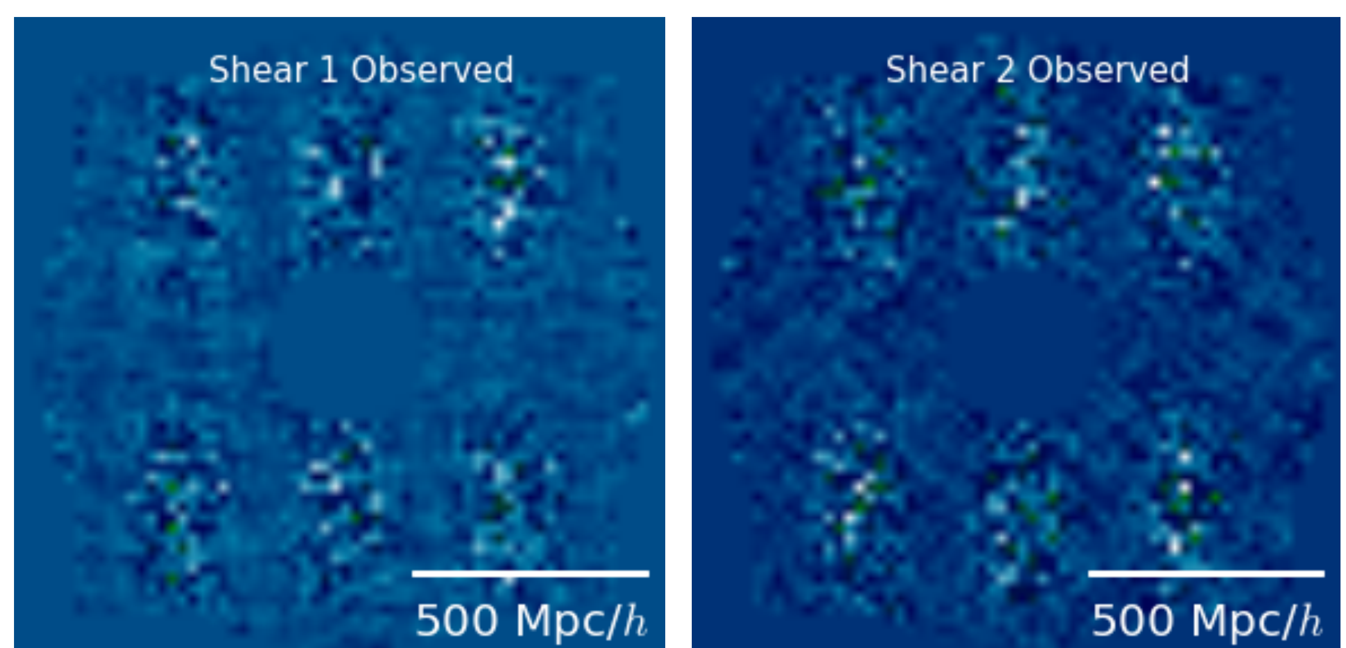
Linear case: Weak Lensing

Toy Example: 64x64 grid



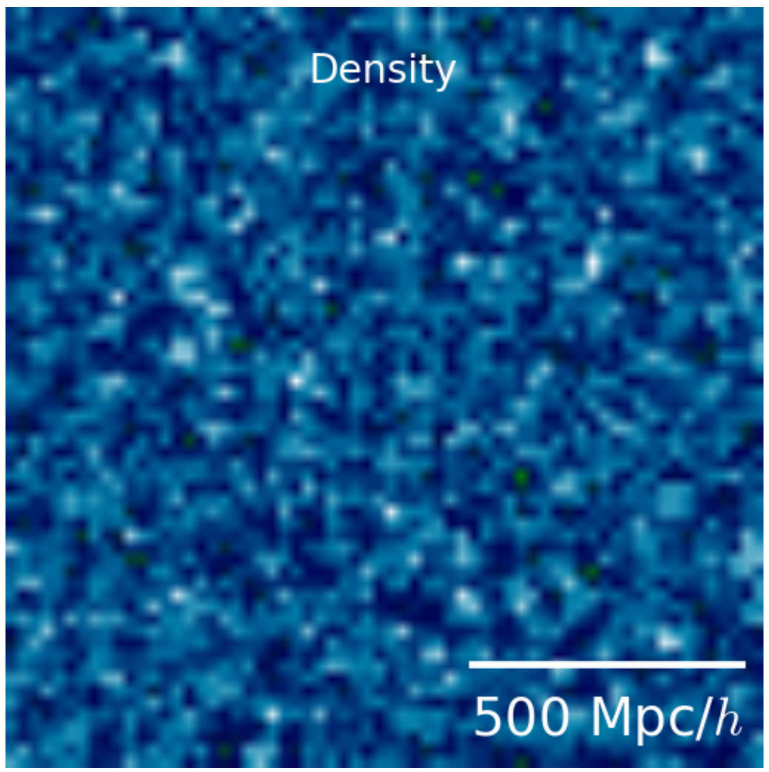
Add noise and mask

Observed data:

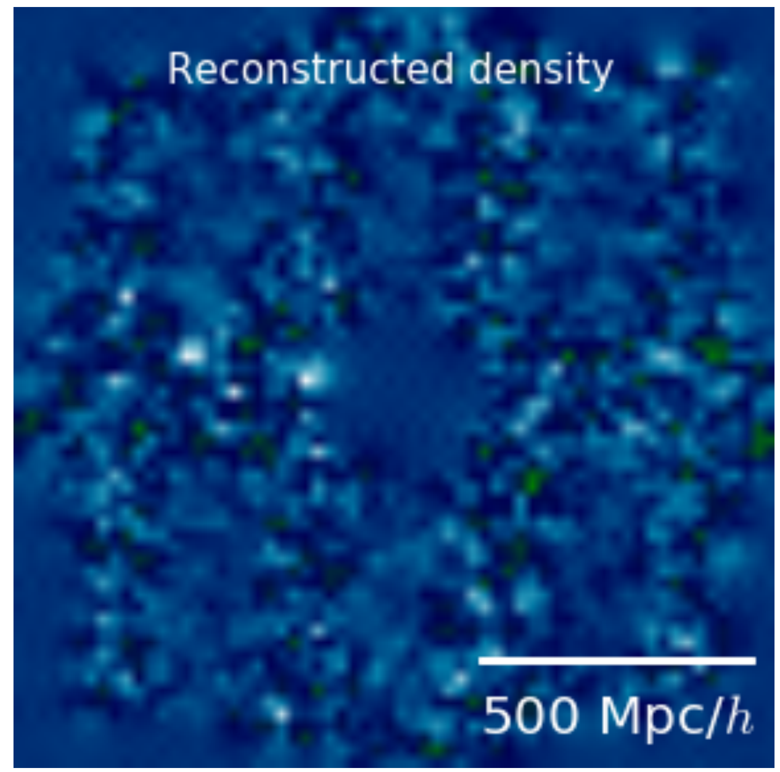


Results

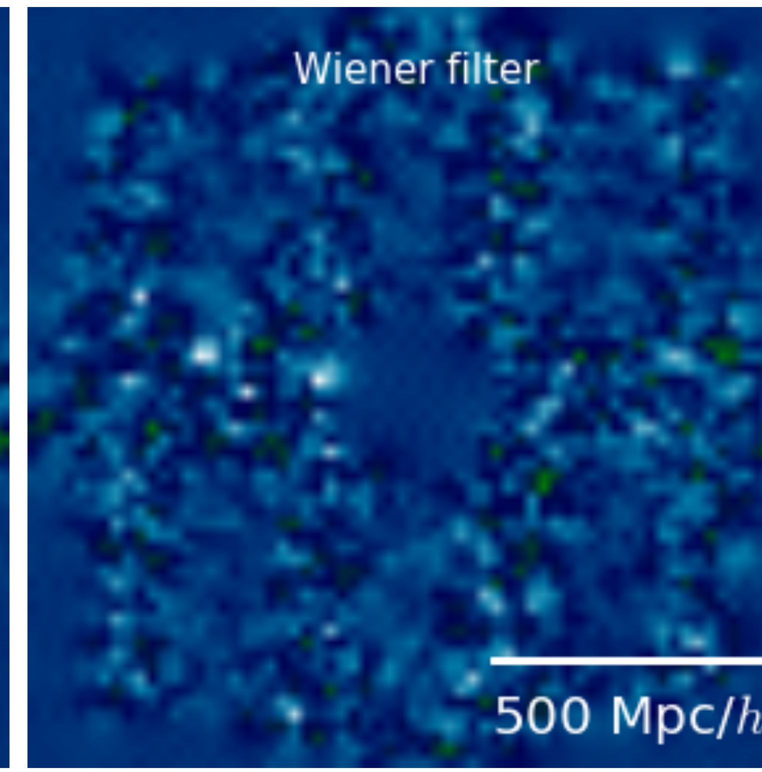
Original

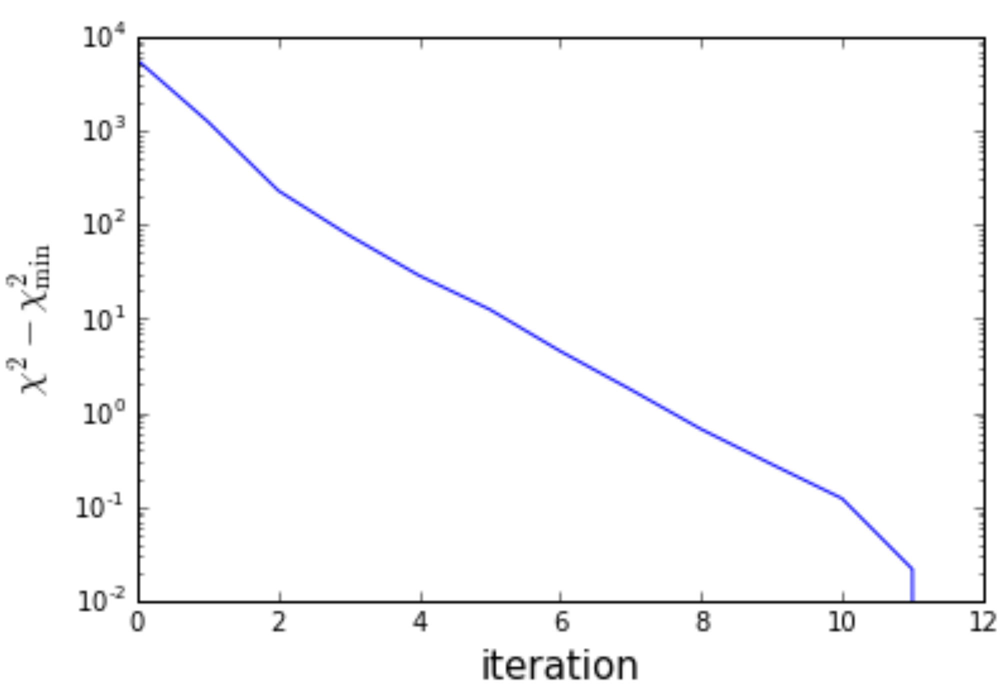


L-BFGS

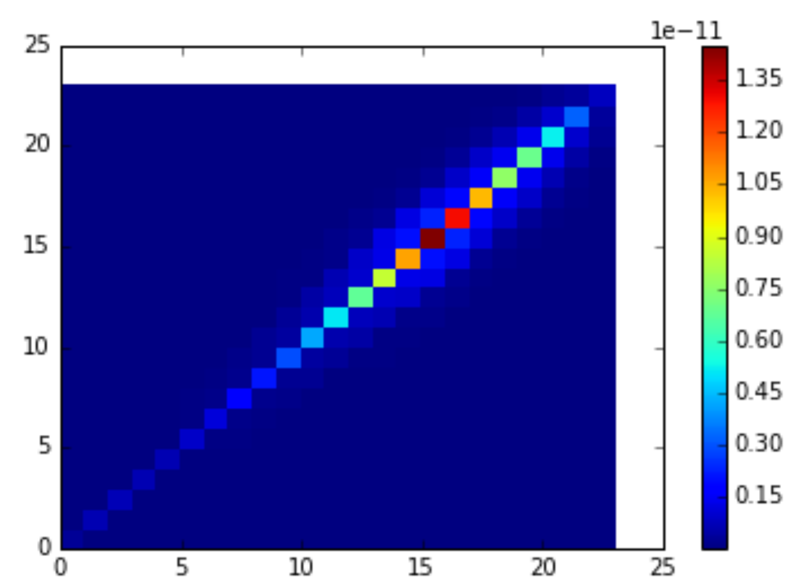


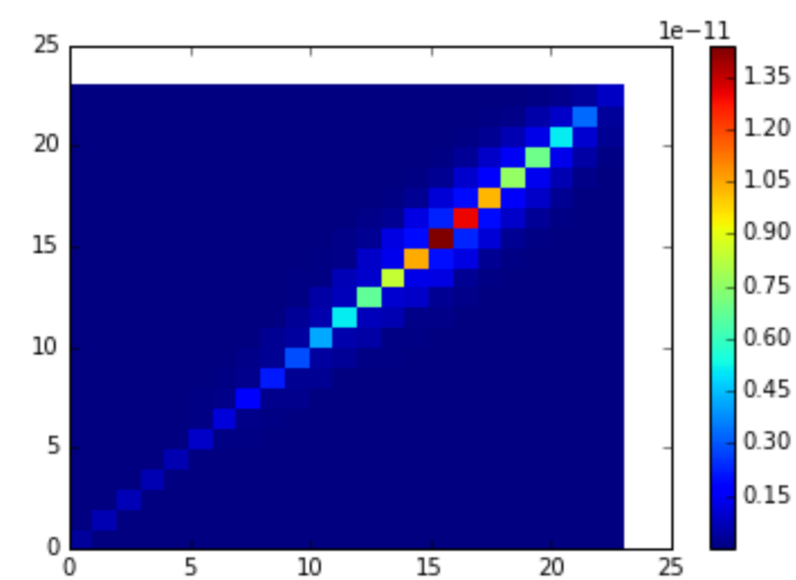
Linear Algebra





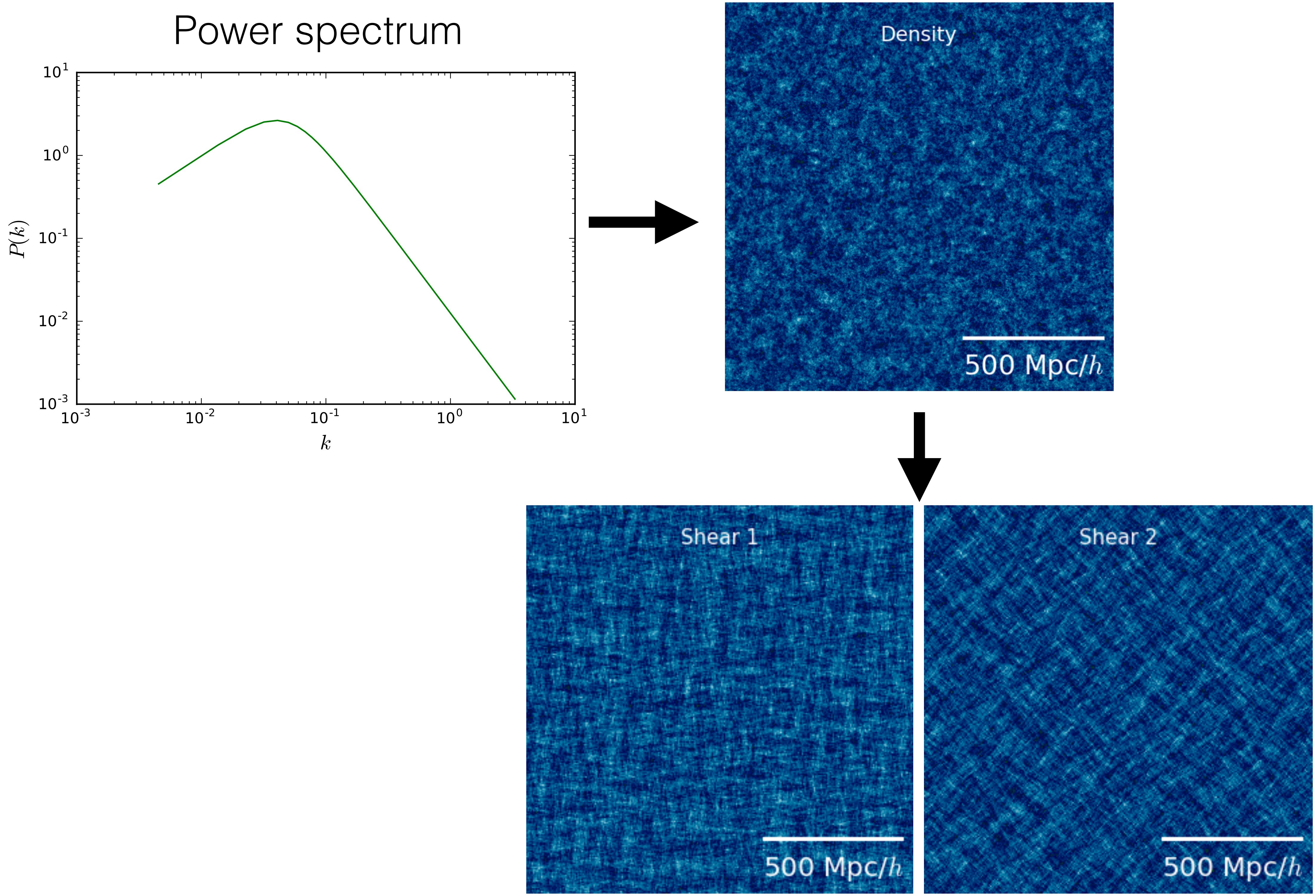
Fisher matrix:





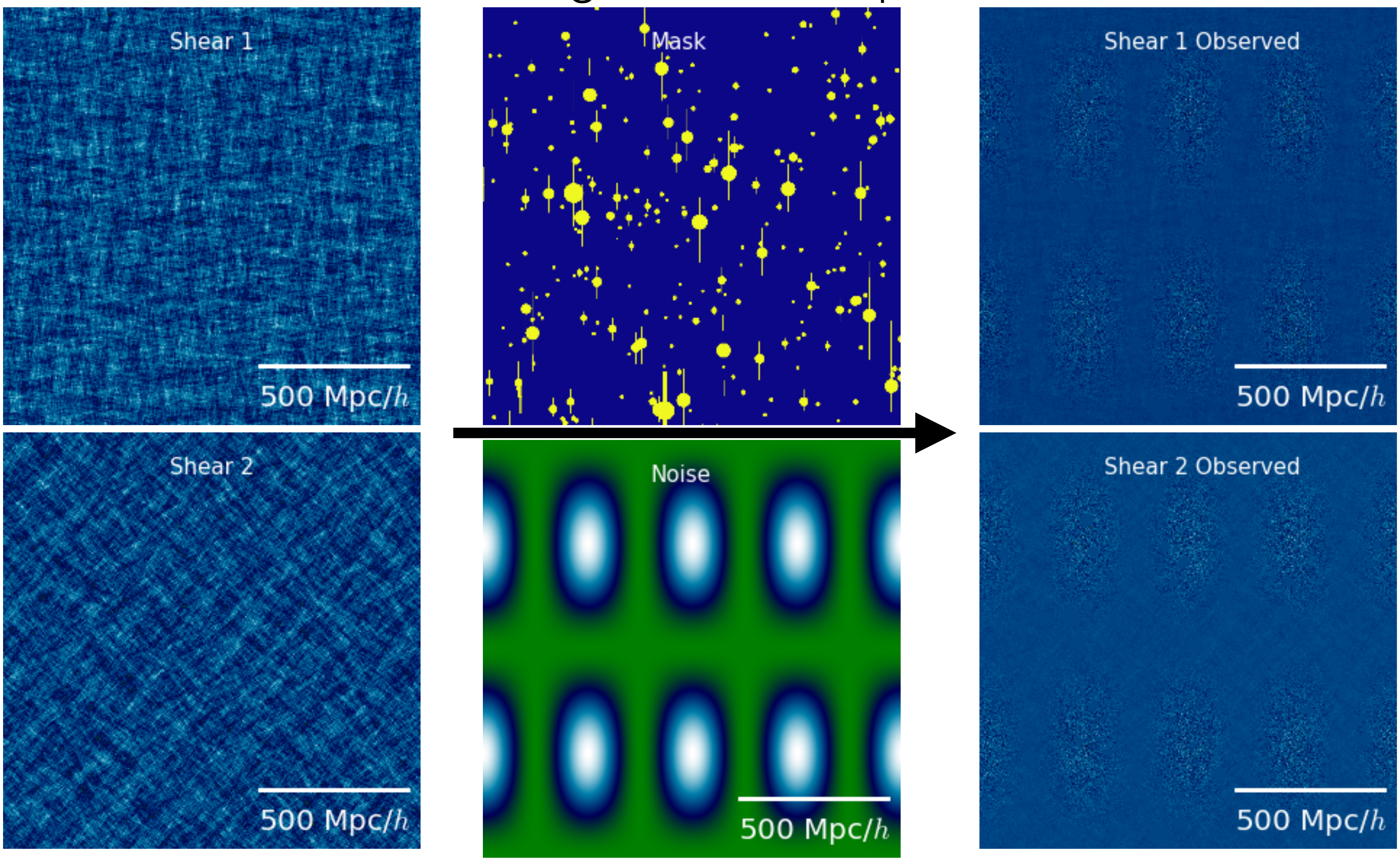
Linear case: Weak Lensing

1024x1024 grid: ~million parameters



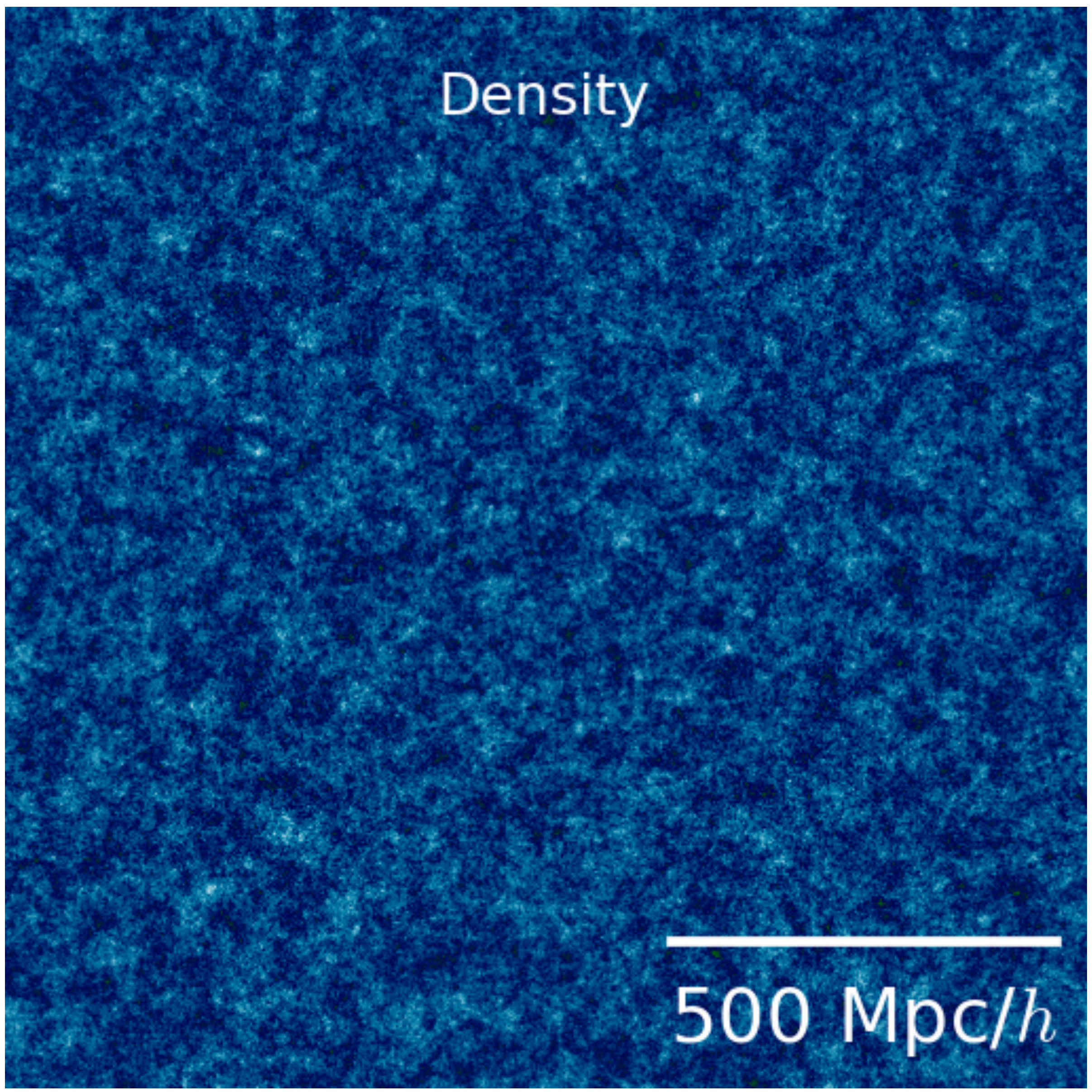
Linear case: Weak Lensing

1024x1024 grid: ~million parameters

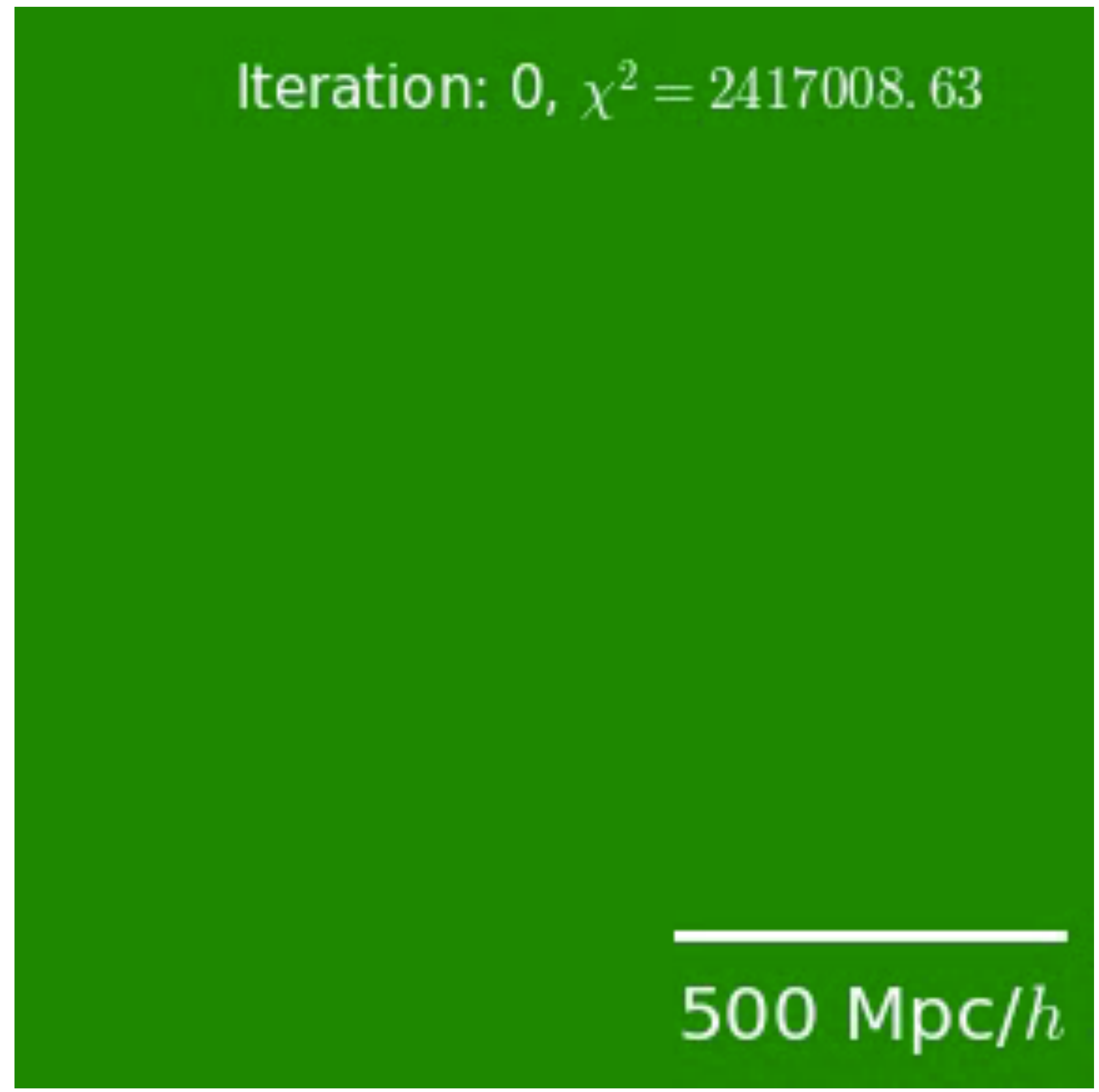


Reconstruction

Original

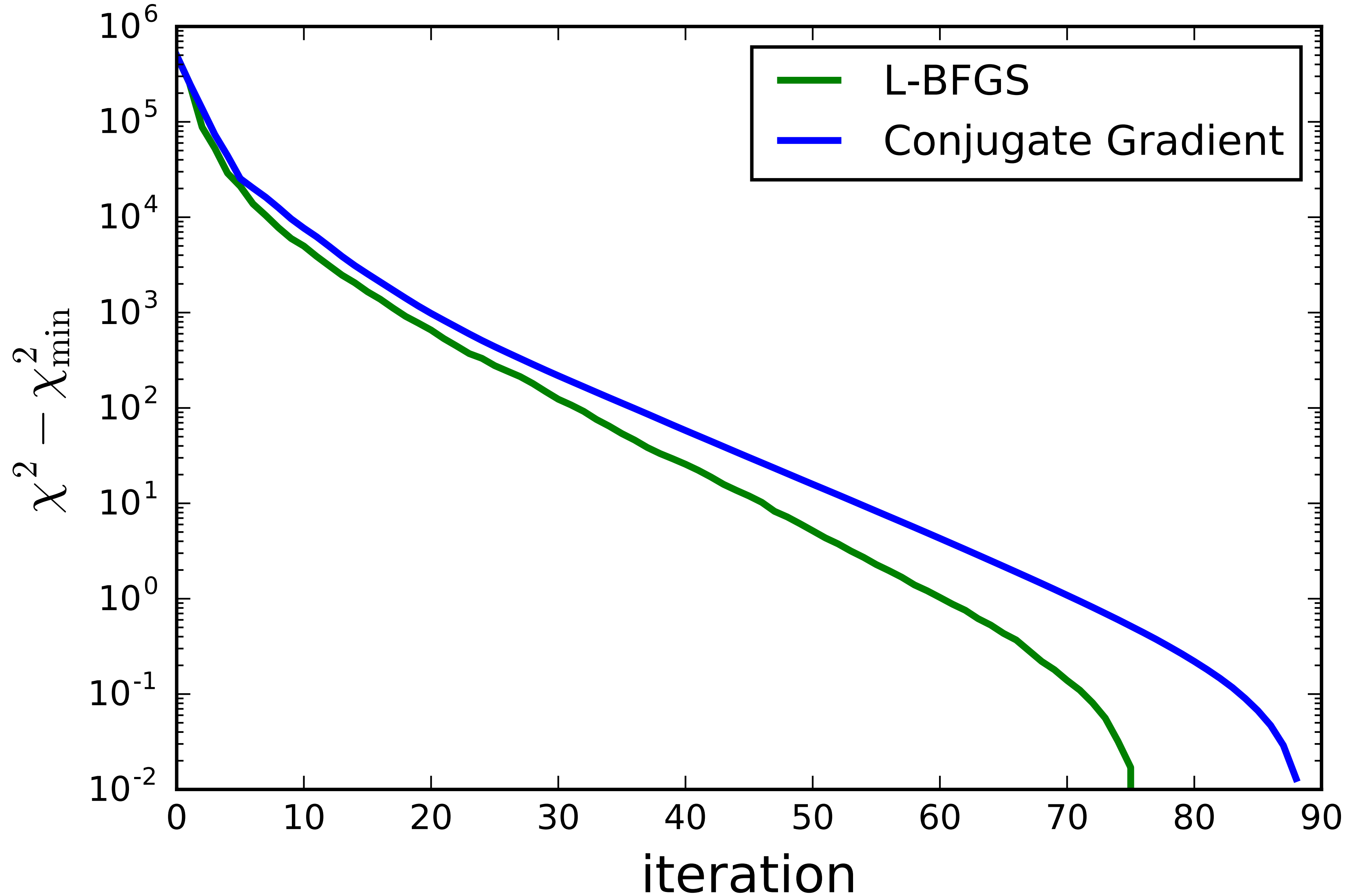


L-BFGS in action

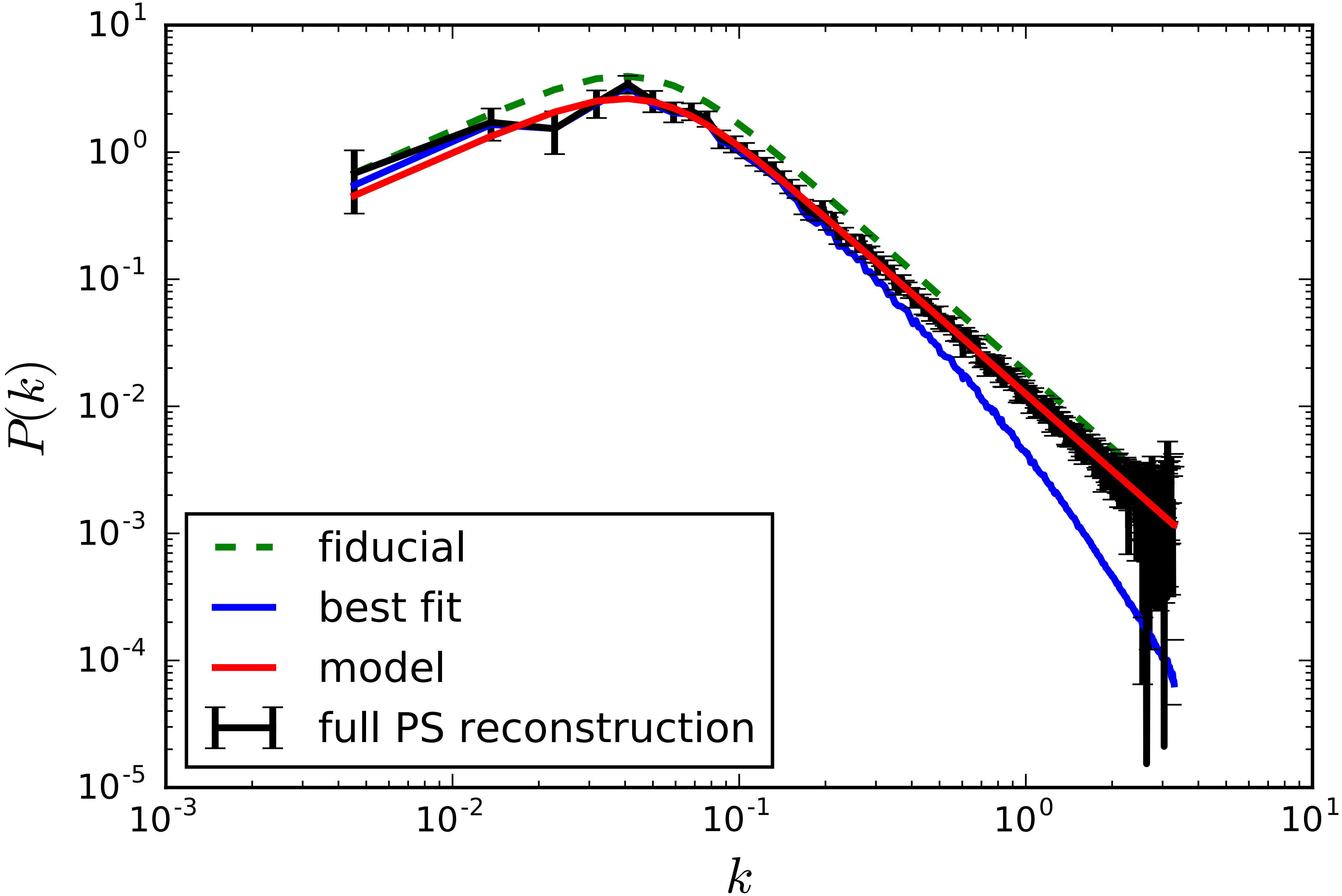


CPU time ~ 1 min.

L-BFGS vs. Conjugate Gradient



Power Spectrum



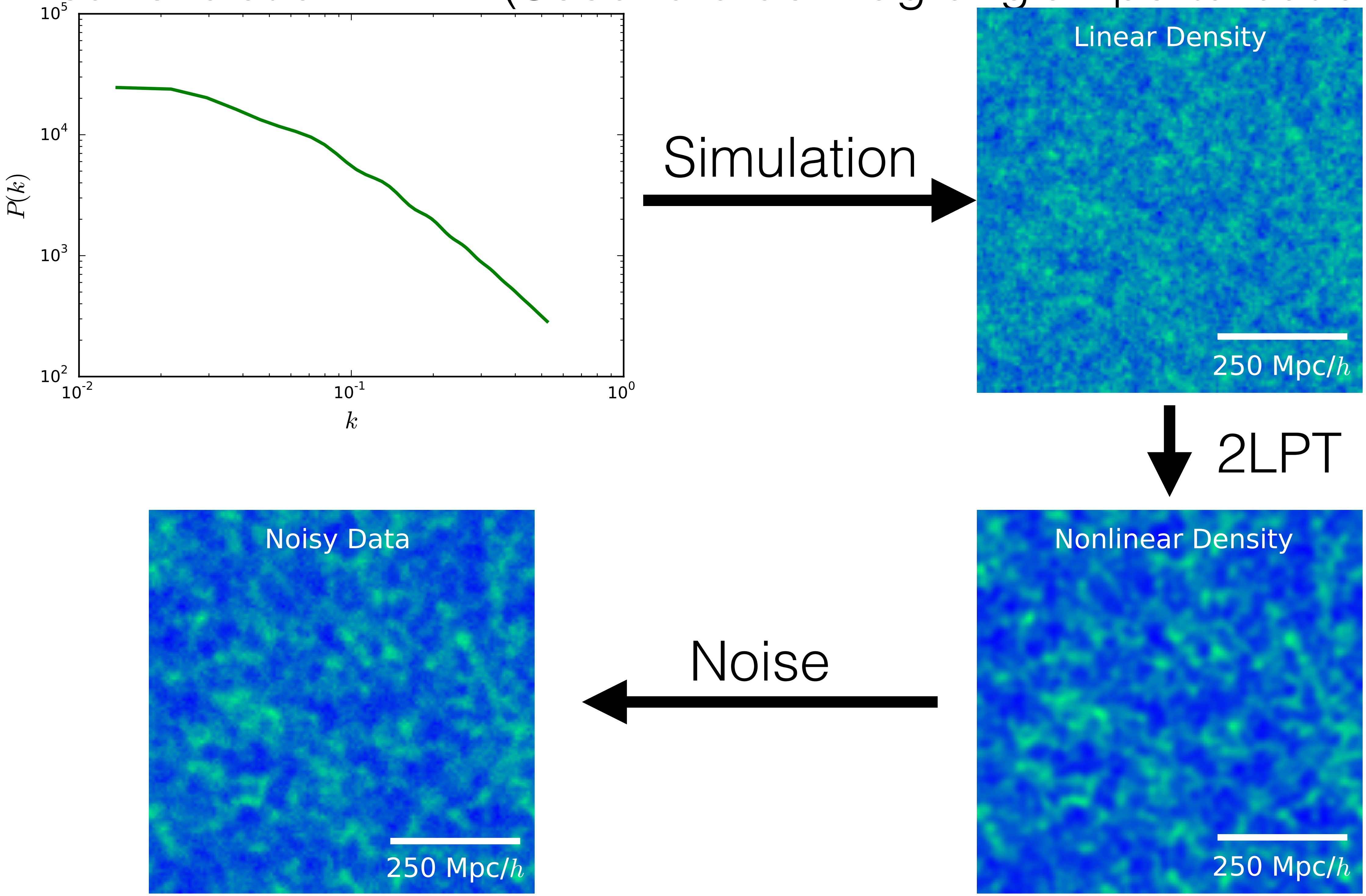
Fiducial power is 50% larger than the true power.

CPU time ~ 1 hour

Nonlinear Case: Large Scale Structure

Grid: 128^3, Box size: $(750\ \mathrm{Mpc/h})^3$

Nonlinear evolution: 2LPT (Second order Lagrangian perturbation theory)

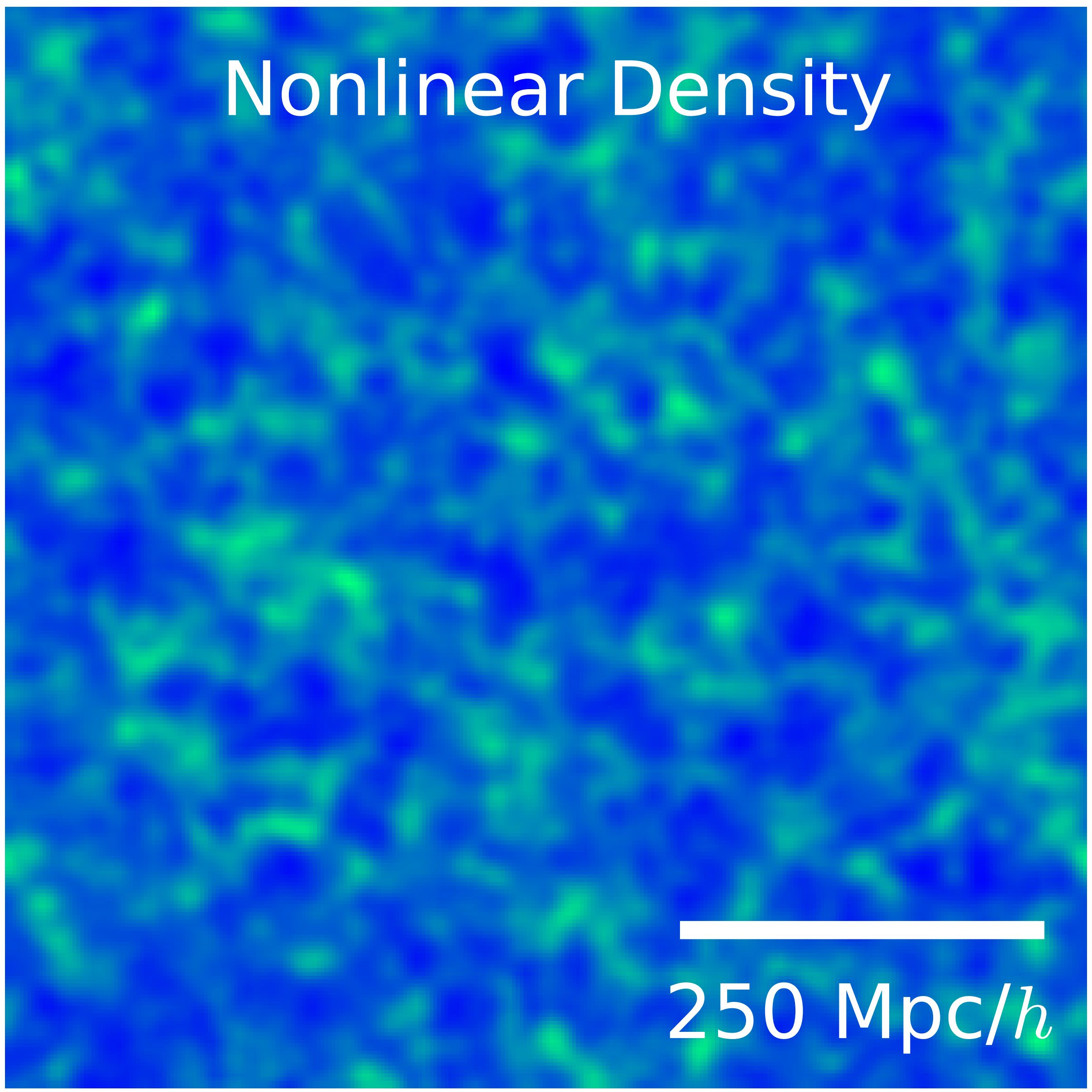


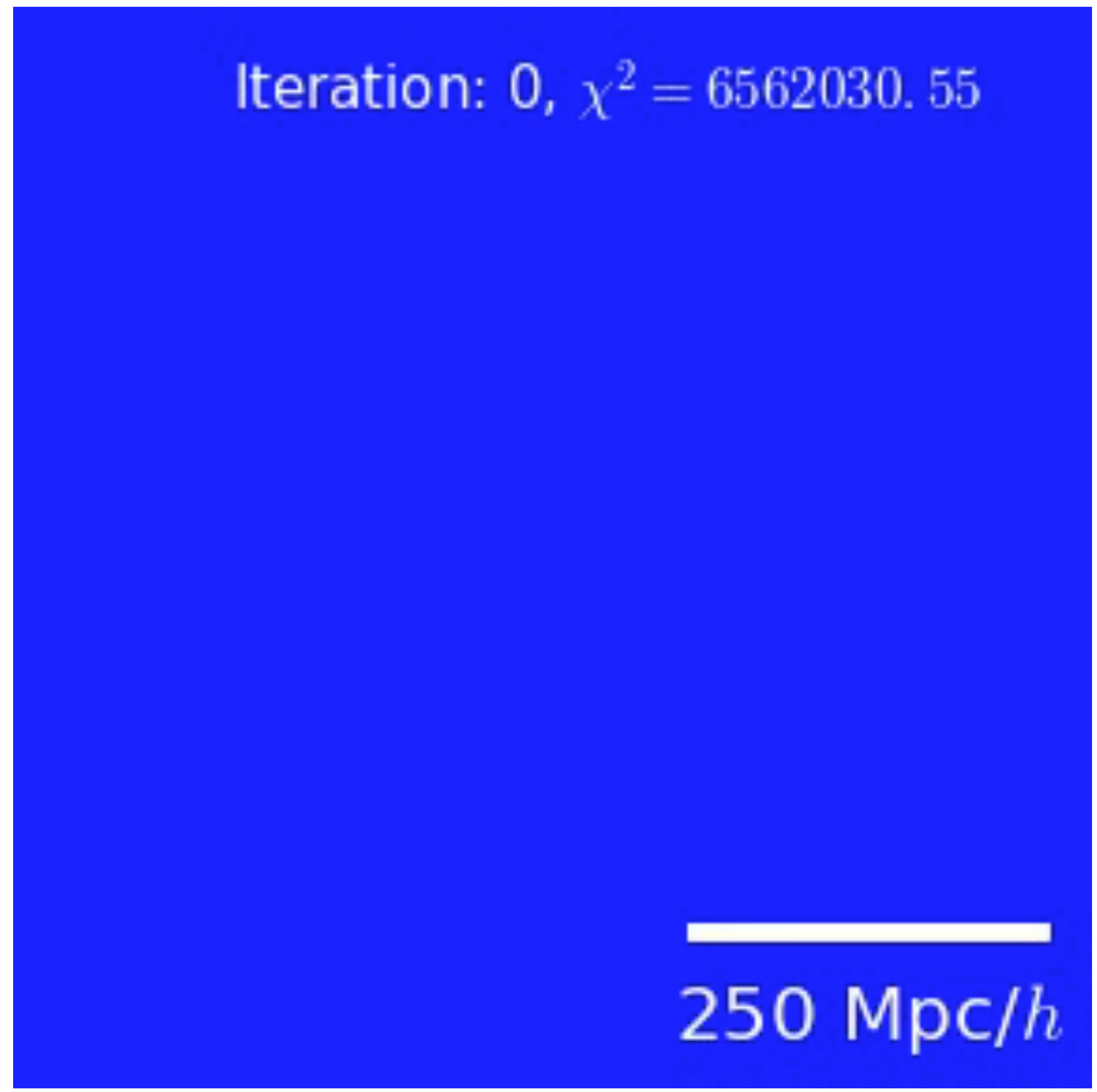
Code used: **fastPM** (Yu Feng, Man-Yat Chu, Uros Seljak, *1603.00476*)

Reconstruction

Nonlinear density

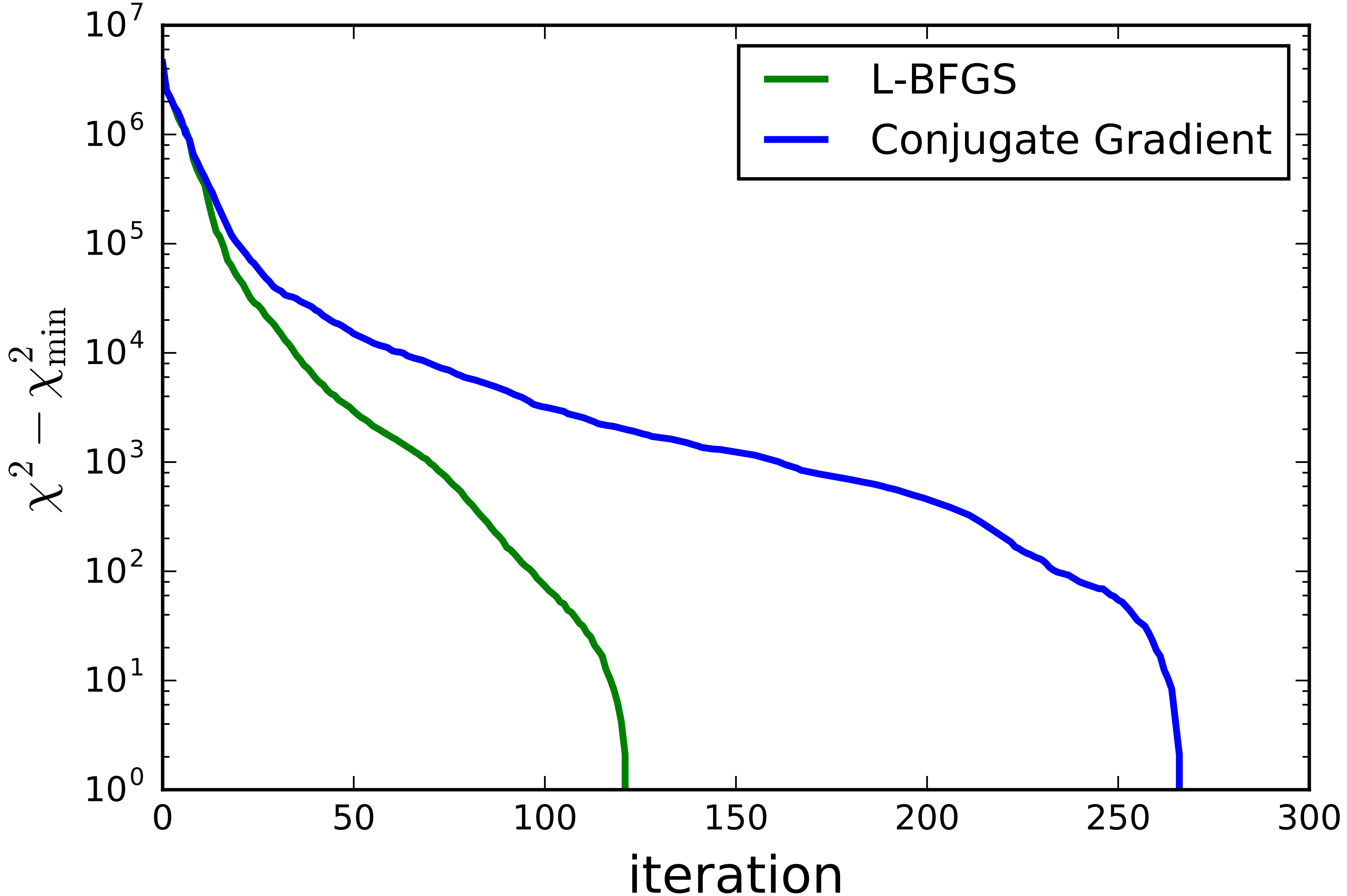
Original	L-BFGS in action



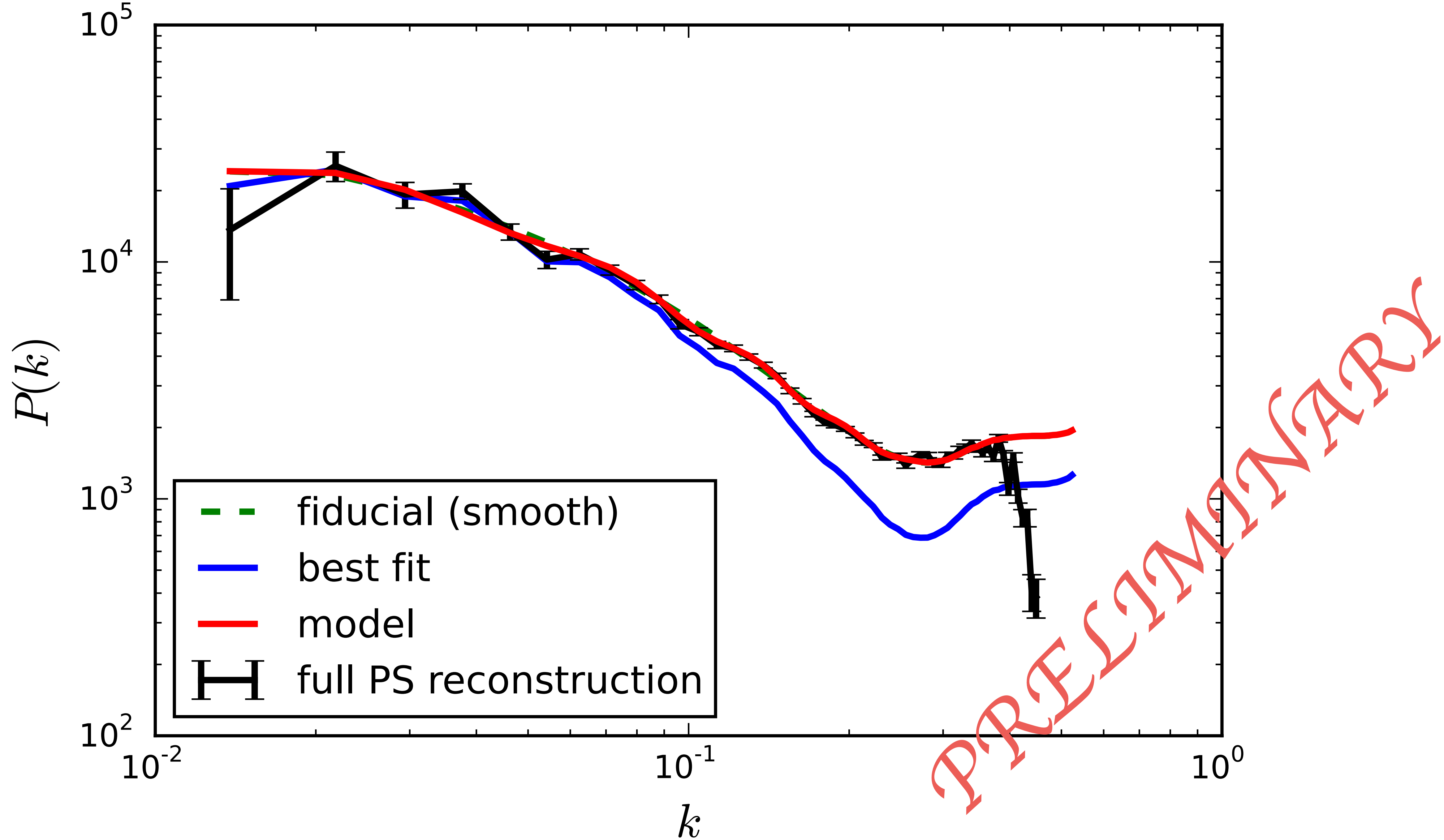


CPU time ~ 1 hour

L-BFGS vs. Conjugate Gradient



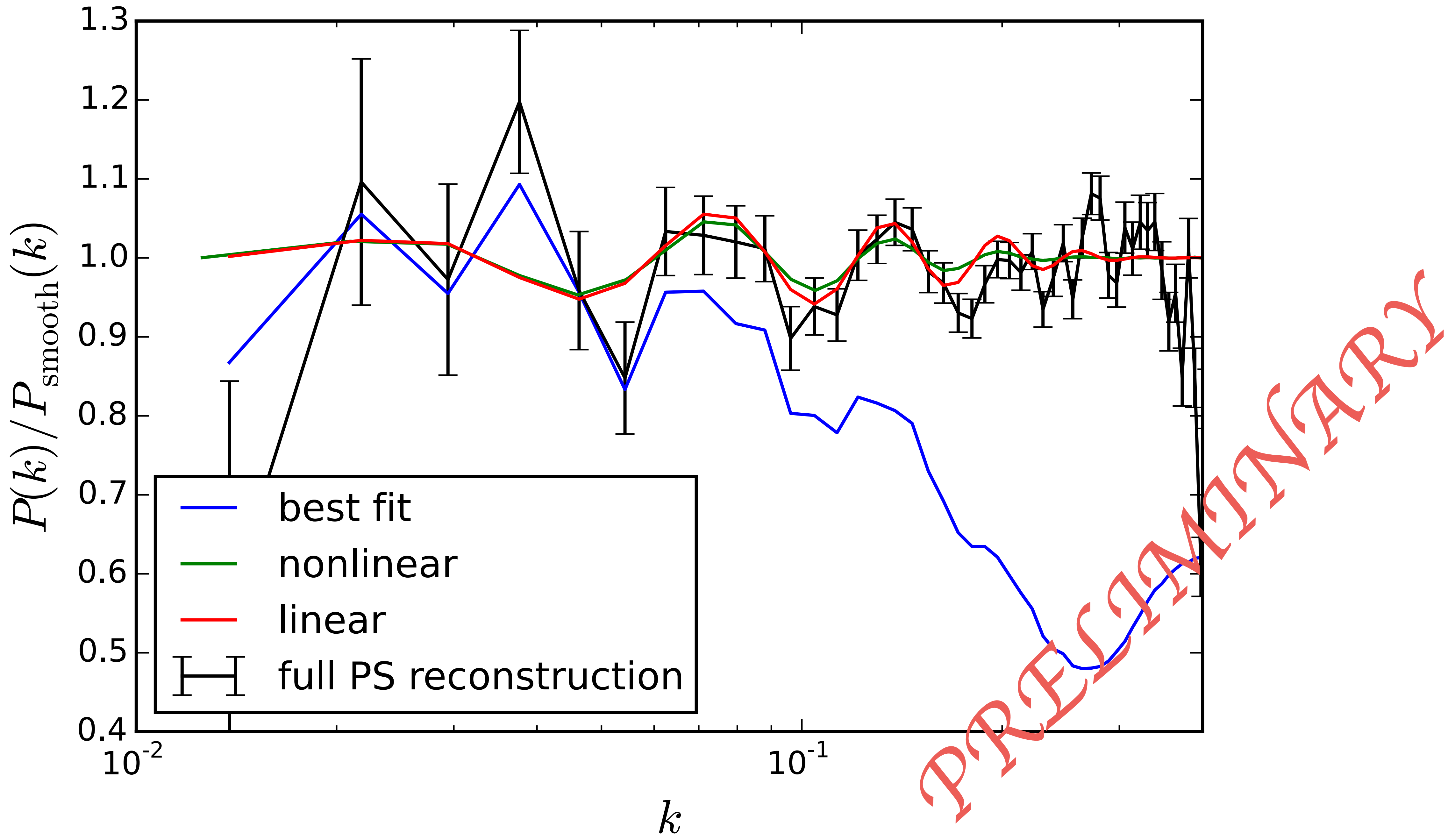
Power Spectrum



Fiducial power has no wiggles. Power spectra are convolved with window.

CPU time ~ 60 hours

Power Spectrum



Showing ratio to fiducial (smooth) power.

CPU time ~ 60 hours

Comparison with HMC

Previously:
HMC: Jasche, Wandelt, *1203.3639,…*
Wang, Mo, Yang, van den Bosch, *1301.1348,…*

- HMC Burnin: ~500 iterations (~5,000 function/derivative calls).
- L-BFGS optimization: ~100 iterations (~100 function/derivative calls).
- HMC correlation length: ~200.
- HMC sample of 10,000: ~100,000 function/derivative calls.
- L-BFGS full fisher matrix/power spectrum estimation: ~5,000 function/derivative calls.

Summary

- L-BFGS is a fast optimizer for very high dimensional parameter spaces.
- Conjugate gradient works almost as well as L-BFGS for the linear case. Not so much for the nonlinear case.
- Our reconstruction method works well for both linear and nonlinear models with ~million parameters at least.
- HMC is at least an order of magnitude more expensive. But if you really need a full sample then HMC is the way to go.
- DO NOT use HMC for minimization!
- Optimizers (L-BFGS, Conjugate gradient) and HMC publicly available (soon) as a part of the **cosmo++** package: cosmopp.com