CCAPP

THE OHIO STATE UNIVERSITY

FAST-PT

An Extremely Efficient Algorithm for Mode-Coupling Integrals in Cosmological Perturbation Theory

Xiao Fang

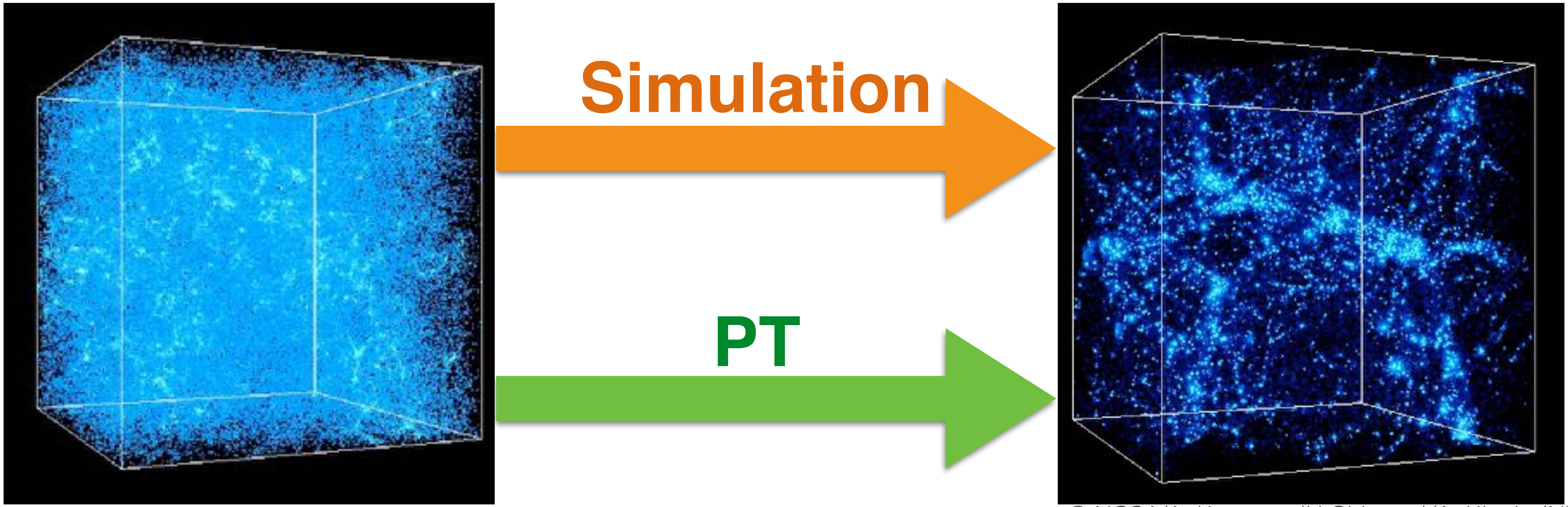
The Ohio State University

Collaborators: Jonathan Blazek, Christopher Hirata, Joseph McEwen

McEwen *et al.* JCAP 09,015(2016)

Fang *et al.* JCAP 02,030(2017)

Perturbation Theory for LSS



Initial: Homogeneous
+ Tiny Density Fluctuations

LSS

PT:

- Much faster
- Valid at linear and mildly nonlinear regime

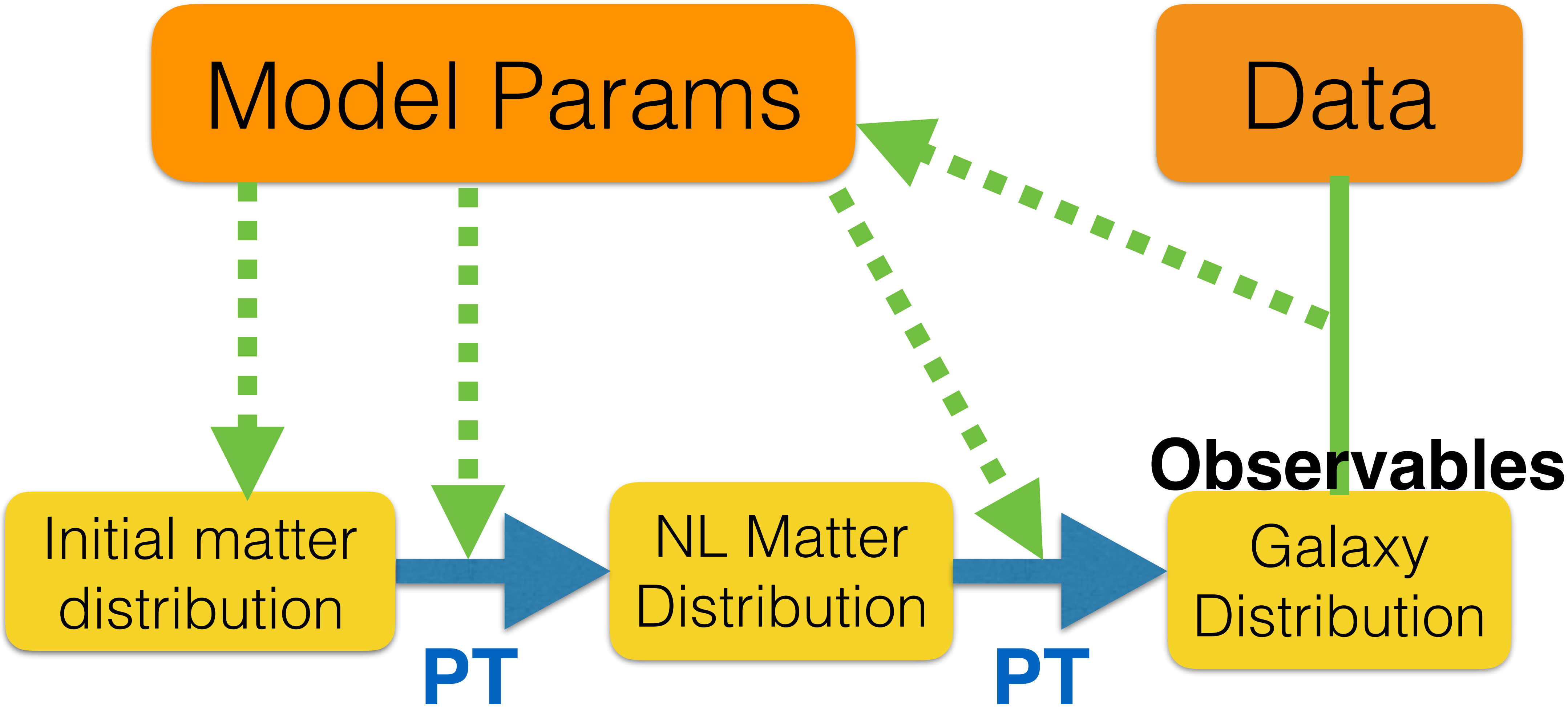
PT is very important for cosmology

Practical Issue in PT

Slow!

How Fast is Fast?

Example Sketch: Galaxy Clustering



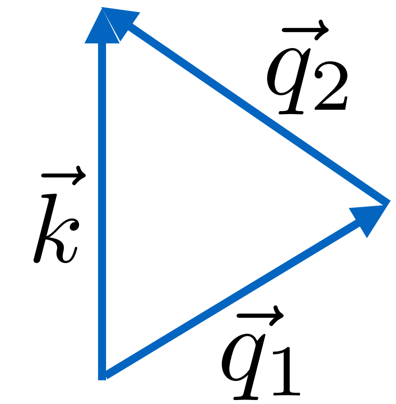
A chain may run 10^4-10^5 or more evaluations!

Solution: FAST-PT

Scalar quantity:

McEwen *et al.* (2016)

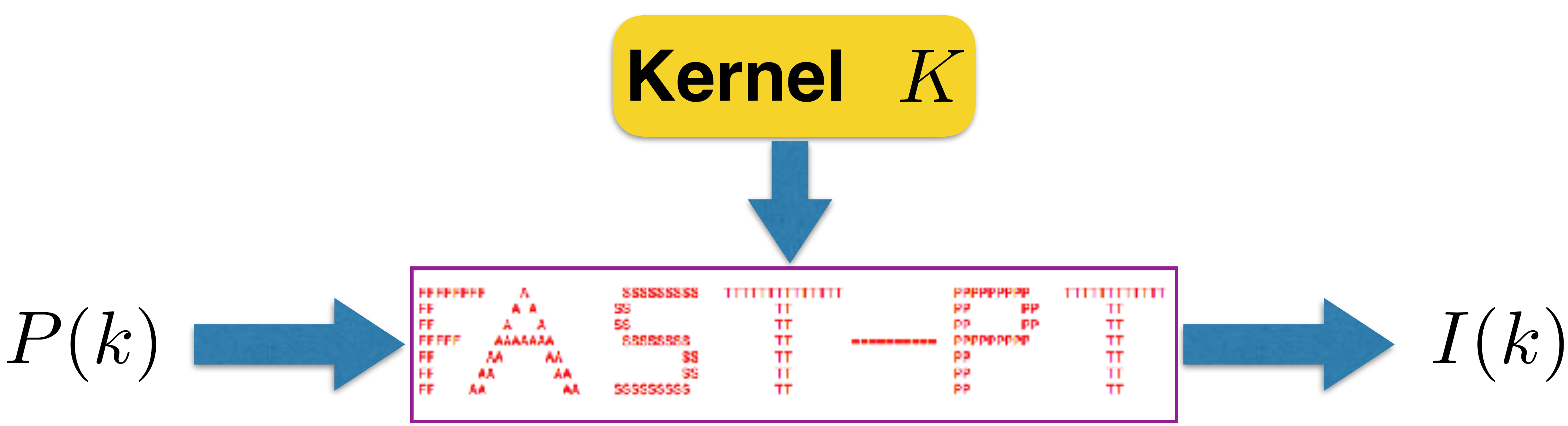
$$I(k) = \int \frac{d^3\boldsymbol{q}_1}{(2\pi)^3} K(\hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{q}}_2, q_1, q_2) P(q_1) P(q_2)$$



Tensor quantity:

Fang *et al.* (2017)

$$I(k) = \int \frac{d^3\boldsymbol{q}_1}{(2\pi)^3} K(\hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{q}}_2, \hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{k}}, \hat{\boldsymbol{q}}_2 \cdot \hat{\boldsymbol{k}}, q_1, q_2) P(q_1) P(q_2)$$



Scalar Quantity

Example: Matter Power Spectrum

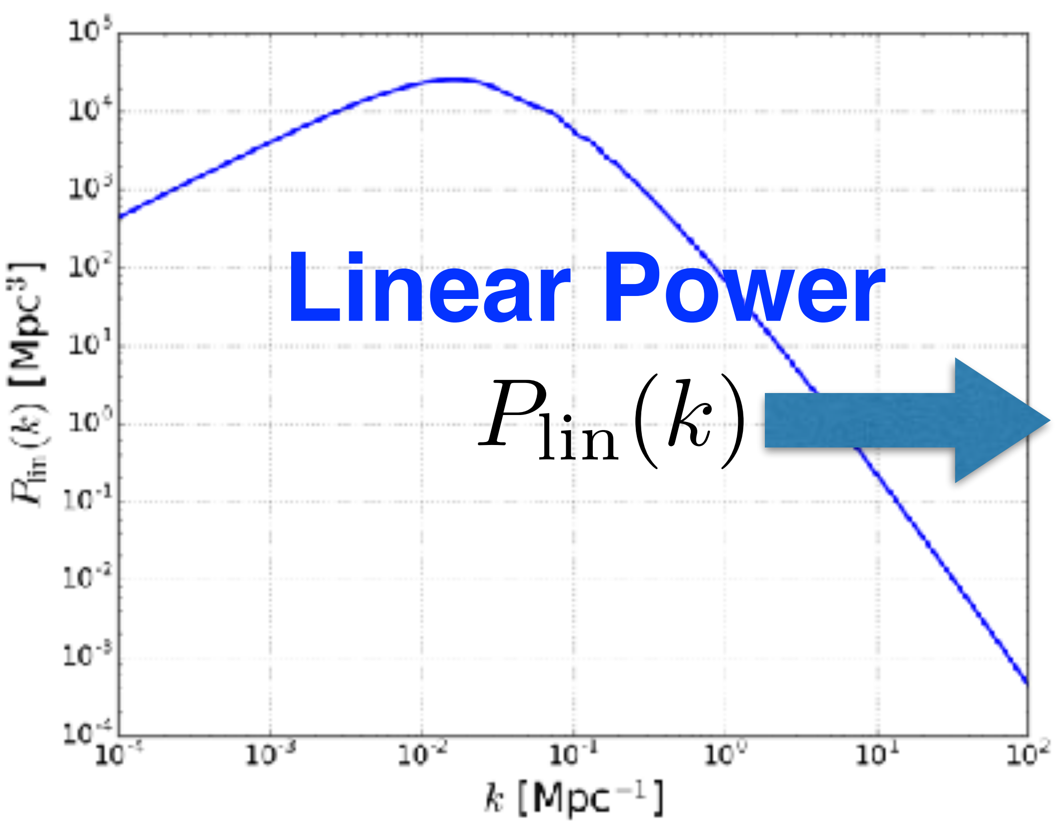
with leading order NL: $P_{\rm NL}(k) = P_{\rm lin}(k) + P_{22}(k) + P_{13}(k)$

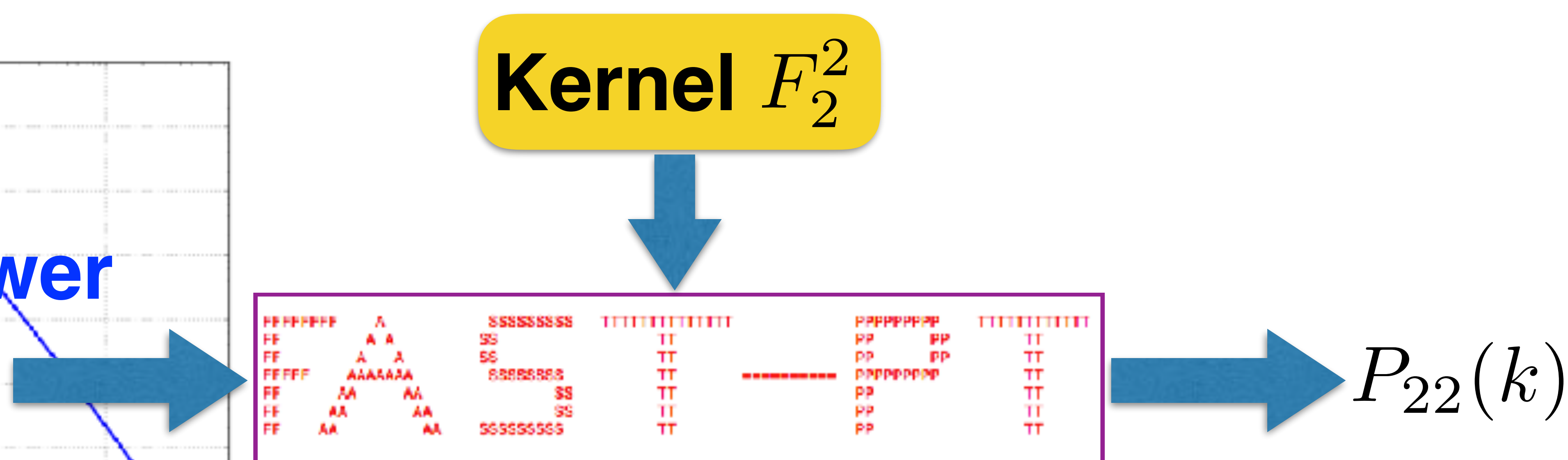
$$P_{22}(k) = 2\int \frac{d^3 q_1}{(2\pi)^3} P_{\rm lin}(q_1) P_{\rm lin}(|\vec{k}-\vec{q}_1|) F_2^2(\vec{q}_1, \vec{k}-\vec{q}_1)$$

Kernel

Brute-force: $\mathcal{O}(N^3)$

Convolution, 3d: SLOW!





Scalar FAST-PT Algorithm

McEwen *et al.* (2016)

Key Ideas:

1. Convolution —F.T.—> Multiplication
2. Dark matter fluid eqs. with gravity: scale-invariant —> Eigenfunction of scale translation: power-law

$$P_{22} \in \{J_{\alpha\beta\ell}\}$$

Basis, Finite

$$J_{\alpha\beta\ell}(k) = \int \frac{d^3q_1}{(2\pi)^3} q_1^\alpha q_2^\beta \mathcal{P}_\ell(\hat{q}_1 \cdot \hat{q}_2) P_{\rm lin}(q_1) P_{\rm lin}(q_2)$$

$$\vec{q}_2 = \vec{k} - \vec{q}_1$$

↓ F.T.

$$\bar{J}_{\alpha\beta\ell}(r) = \frac{(-1)^\ell}{4\pi^4} I_{\alpha\ell}(r) I_{\beta\ell}(r)$$

Reduce to 1d: $I_{\alpha\ell}(r) = \int dk\ k^{\alpha+2} j_\ell(kr) P_{\rm lin}(k) \leftarrow \sum_{m=-N/2}^{N/2} c_m k^{\nu + i\eta_m}$

Power-law decomposition

All we need: log-spaced FFTs —> NlogN

Runtime Achievement

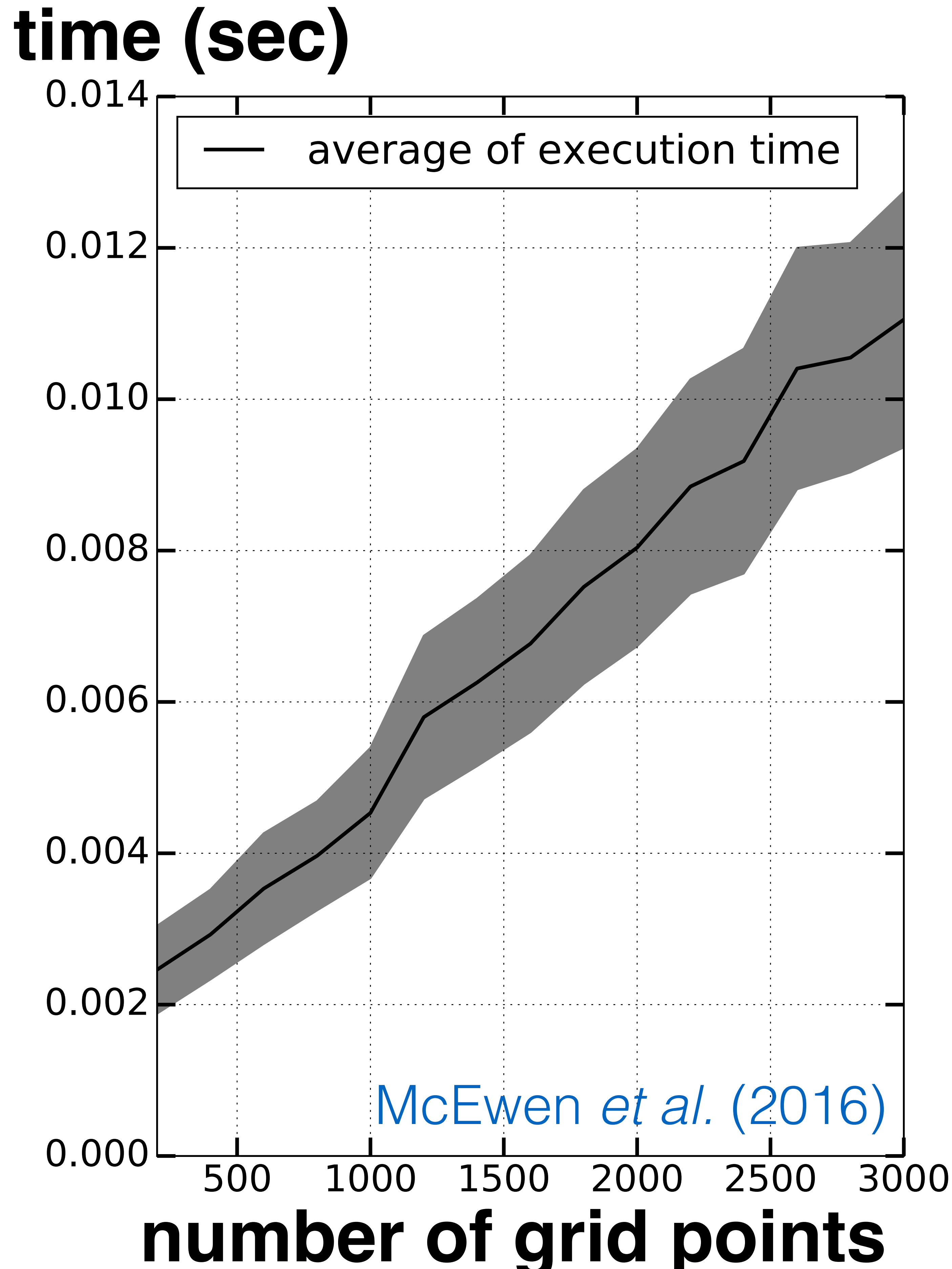
≤0.01sec !

vs.

Conventional Method:

~*minutes for 3000 points*

$$\mathcal{O}(N^3) \longrightarrow \mathcal{O}(N \log N)$$



Tensor FAST-PT Algorithm

Fang *et al.* (2017)

General Form:

$$I(k) = \int \frac{d^3\boldsymbol{q}_1}{(2\pi)^3} K(\hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{q}}_2, \hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{k}}, \hat{\boldsymbol{q}}_2 \cdot \hat{\boldsymbol{k}}, q_1, q_2) P(q_1) P(q_2)$$

$$\vec{q}_2 = \vec{k} - \vec{q}_1$$

$I \in \{I^{\alpha\beta}_{\ell_1\ell_2\ell}\}$ Basis, Finite

$$I^{\alpha\beta}_{\ell_1\ell_2\ell}(k) = \int \frac{d^3\boldsymbol{q}_1}{(2\pi)^3} q_1^\alpha q_2^\beta \mathcal{P}_{\ell_1}(\hat{\boldsymbol{q}}_2 \cdot \hat{\boldsymbol{k}}) \mathcal{P}_{\ell_2}(\hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{k}}) \mathcal{P}_{\ell}(\hat{\boldsymbol{q}}_1 \cdot \hat{\boldsymbol{q}}_2) P(q_1) P(q_2)$$

Change Basis

$$Y_{J_1 M_1}(\hat{\boldsymbol{q}}_1) Y_{J_2 M_2}(\hat{\boldsymbol{q}}_2) Y_{J_k M_k}(\hat{\boldsymbol{k}})$$

k-dependence Separable!

Reduce to scalar FAST-PT !

Applications

Scalar:

Nonlinear matter power in Standard PT

Renormalization Group Approach

Nonlinear Galaxy Bias, …

Tensor:

Intrinsic Alignment

Secondary Effects in CMB

Redshift Space Distortions, …

DES is using FAST-PT for galaxy bias and IA calculations!

Krause *et al*. (2017)

Summary

Scalar: Nonlinear matter power, Renormalization Group Approach, Galaxy Bias, etc.

Tensor: Intrinsic Alignment, Secondary Effects in CMB, Redshift Space Distortions, etc.

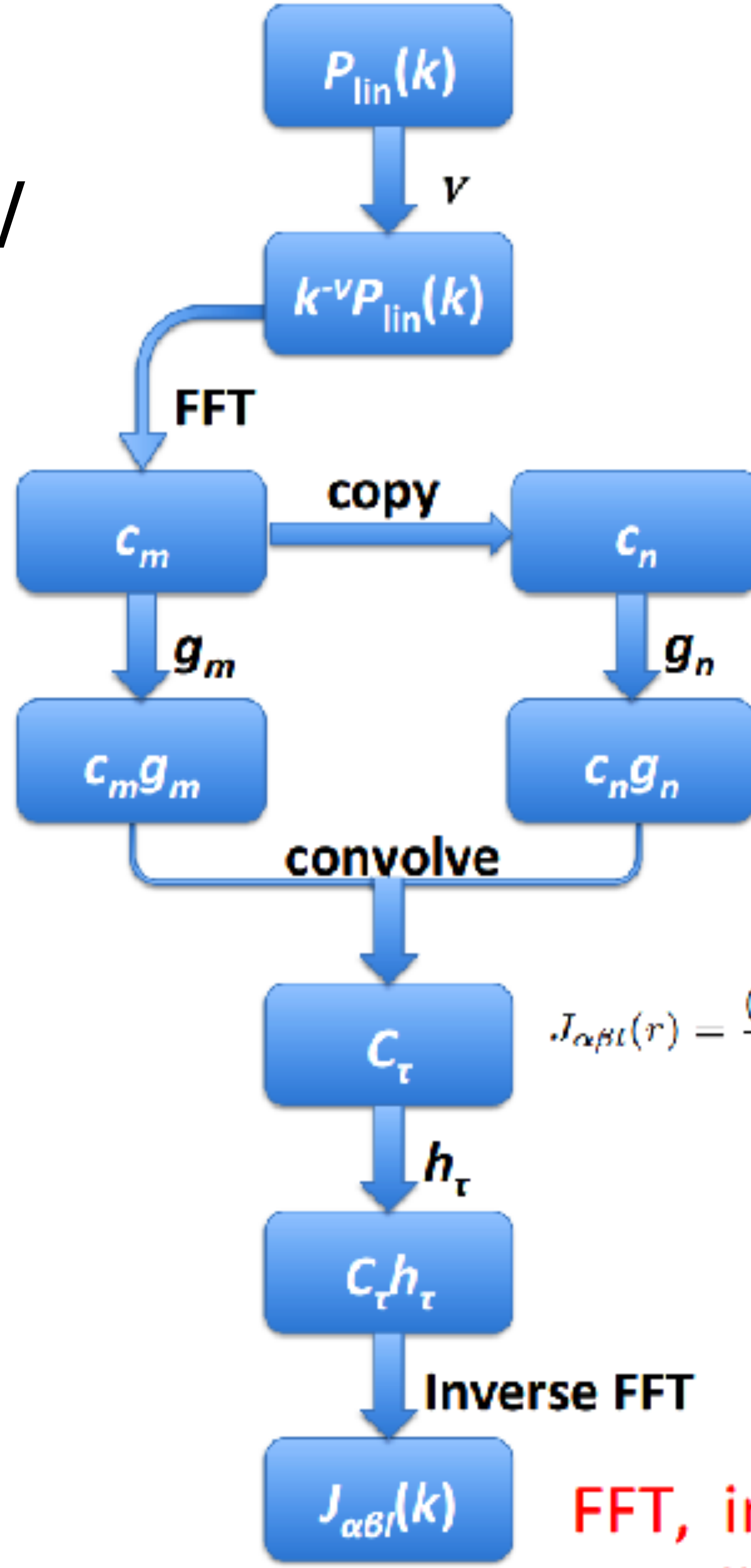
- Versatile
- Fast
- In Python, User-Friendly!
- Public on Github with User Manual
- Incorporated in DES pipeline

https://github.com/JoeMcEwen/FAST-PT

Powerful Tool For Your 1-Loop Calculations!

```
FFFFFFFF      A          SSSSSSSSS   TTTTTTTTTTTTTT              PPPPPPPPP    TTTTTTTTTTTT
FF           A A        SS                 TT                    PP       PP        TT
FF          A   A       SS                 TT                    PP       PP        TT
FFFFF      AAAAAAA       SSSSSSSS          TT        ==========  PPPPPPPPP          TT
FF        AA       AA            SS        TT                    PP                 TT
FF       AA         AA           SS        TT                    PP                 TT
FF      AA           AA  SSSSSSSSS         TT                    PP                 TT
```

Backup: Flow chart



$$P_{\rm lin}(k) = \sum_{m=-N/2}^{N/2} c_m k^{\nu + i\eta_m}$$

$$\eta_m = \frac{2\pi m}{N\Delta} \qquad \Delta = \ln k_{i+1} - \ln k_i$$

$$g_m = \frac{\Gamma[(l+3+\nu+\alpha+i\eta_m)/2]}{\Gamma[(l-\nu-\alpha-i\eta_m)/2]}$$

$$g_n = \frac{\Gamma[(l+3+\nu+\beta+i\eta_n)/2]}{\Gamma[(l-\nu-\beta-i\eta_n)/2]}$$

$$J_{\alpha\beta l}(r) = \frac{(-1)^l}{8\pi^3} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} c_m g_m c_n g_n 2^{3+2\nu+i(\eta_m+\eta_n)}$$

FFT, inverse FFT and convolution are all *NlogN* algorithms
→ Much faster than N^3

Backup Slide

Power spectrum:

$$\langle \delta(\vec{k})\delta^*(\vec{k}')\rangle = (2\pi)^3\delta_D^3(\vec{k}-\vec{k}')P(k)$$

Perturbative Expansion:

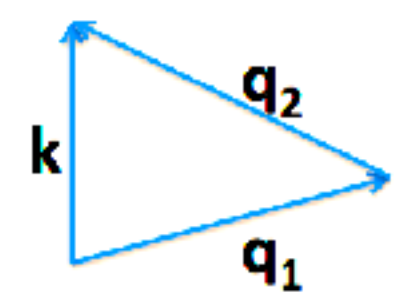
$$\delta(\vec{k}) = \delta^{(1)}(\vec{k}) + \delta^{(2)}(\vec{k}) + \delta^{(3)}(\vec{k}) + \cdots$$

$$\delta^{(n)}(\vec{k}) = \int \frac{d^3q_1}{(2\pi)^3}\cdots\frac{d^3q_n}{(2\pi)^3}\delta_D^3\left(\vec{k}-\sum_{i=1}^{n}\vec{q}_i\right) \underbrace{F_n(\vec{q}_1,\cdots,\vec{q}_n)}_{\text{Kernel}}\delta^{(1)}(\vec{q}_1)\cdots\delta^{(1)}(\vec{q}_n)$$

Equations for 1-Loop

$$P_{1-\text{loop}}(k) = P_{\text{lin}}(k) + P_{22}(k) + P_{13}(k)$$

$$P_{22}(k) = 2\int \frac{d^3q}{(2\pi)^3} P_{\text{lin}}(q) P_{\text{lin}}(|\vec{k}-\vec{q}|) F_2^2(\vec{q},\vec{k}-\vec{q})$$



$$F_2(\vec{q}_1,\vec{q}_2) = \frac{5}{7} + \frac{\vec{q}_1\cdot\vec{q}_2}{2q_1q_2}\left(\frac{q_2}{q_1}+\frac{q_1}{q_2}\right) + \frac{2}{7}\left(\frac{\vec{q}_1\cdot\vec{q}_2}{q_1q_2}\right)^2$$

$$P_{13}(k) = \frac{k^3}{252(2\pi)^2} P_{\text{lin}}(k)\int dr\; r^2 P_{\text{lin}}(kr) Z(r)$$

$$Z(r) = \frac{12}{r^4} - \frac{158}{r^2} + 100 - 42r^2 + \frac{3}{r^5}(7r^2+2)(r^2-1)^3\ln\left(\frac{r+1}{|r-1|}\right)$$

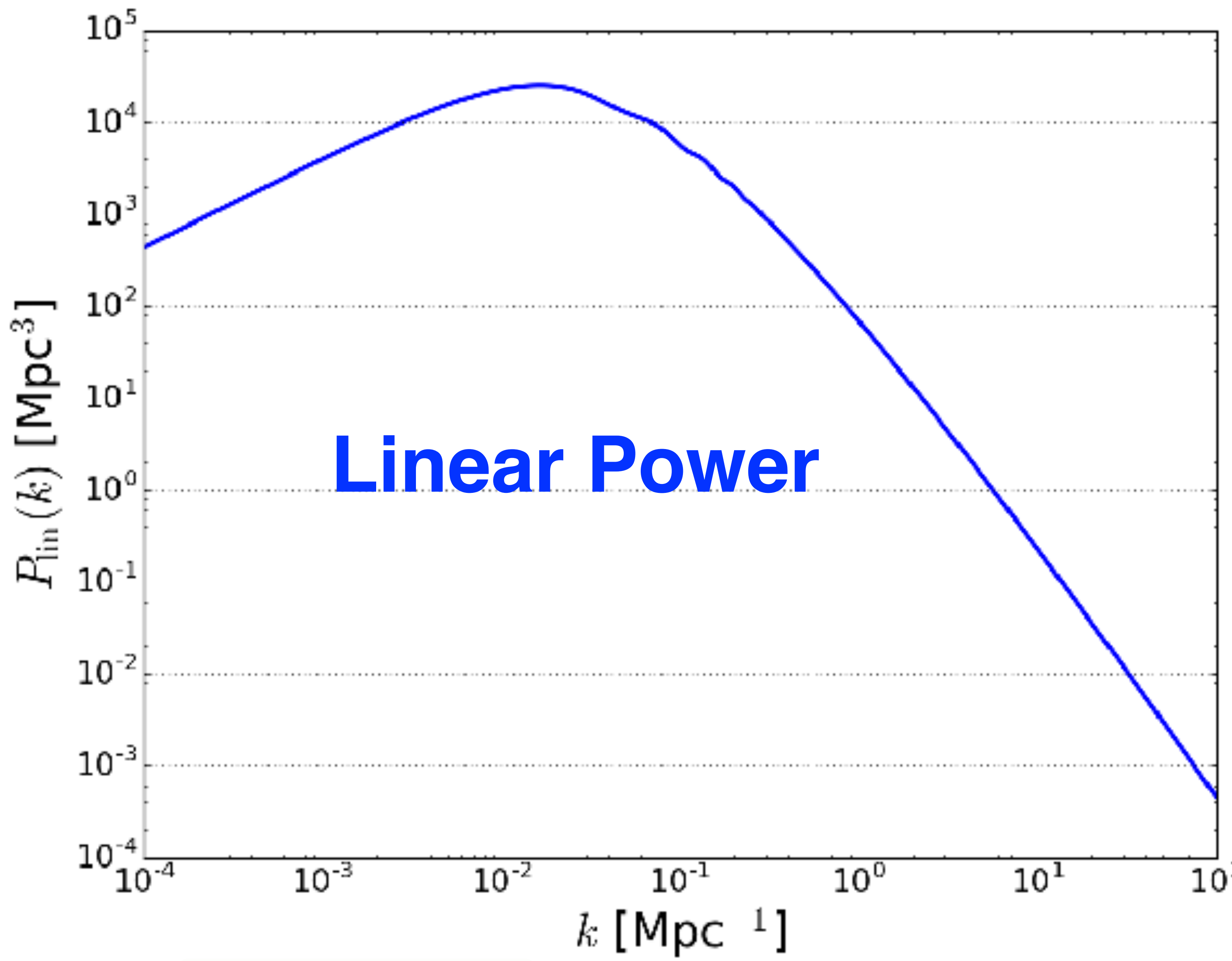
Scalar Quantity

Example: Matter Power Spectrum

$$\langle \delta(\vec{k})\delta^*(\vec{k}')\rangle \equiv (2\pi)^3\delta_D^3(\vec{k}-\vec{k}')P(k)$$

δ^* \ δ	$\delta^{(1)}$	$\delta^{(2)}$	$\delta^{(3)}$	...
$\delta^{(1)*}$	$P_{\rm lin}$	0	$\frac{1}{2}P_{13}$	
$\delta^{(2)*}$	0	P_{22}		
$\delta^{(3)*}$	$\frac{1}{2}P_{13}$			
...				

1-loop order



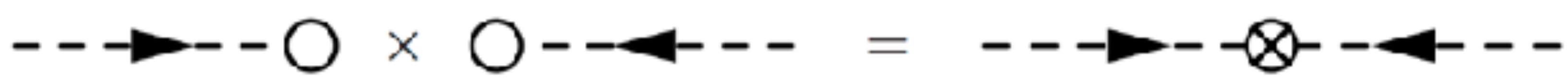
$$P_{22}(k) = 2\int \frac{d^3q_1}{(2\pi)^3} P_{\rm lin}(q_1)P_{\rm lin}(|\vec{k}-\vec{q}_1|)F_2^2(\vec{q}_1,\vec{k}-\vec{q}_1)$$

Kernel

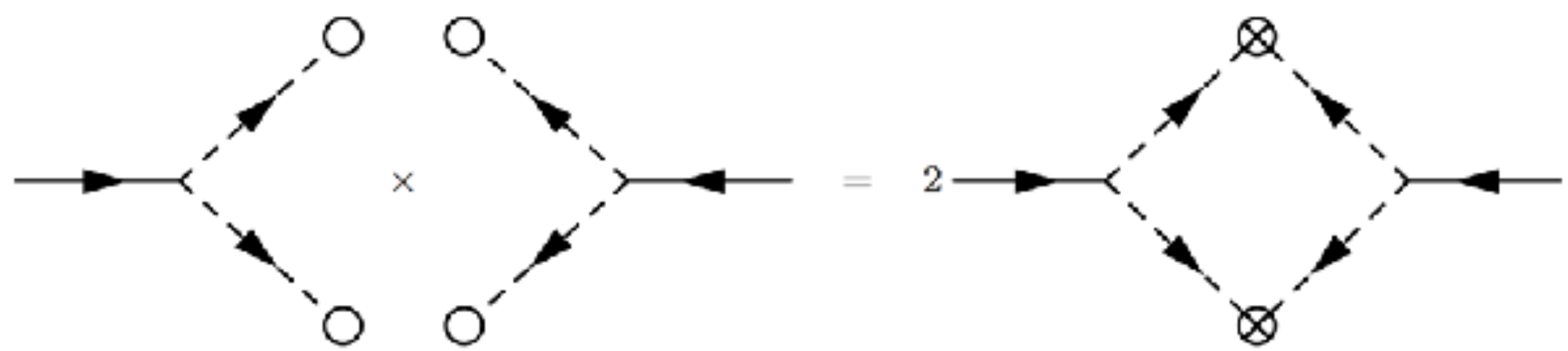
Brute-force: $\mathcal{O}(N^3)$

Convolution, 3d: SLOW!

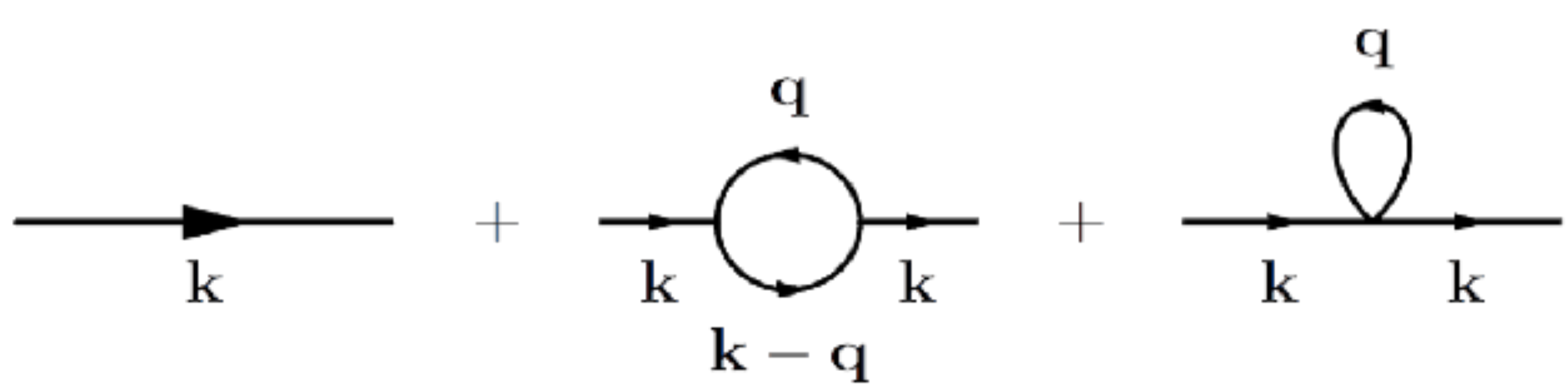
Backup Slide



$$\equiv (2\pi)^3 \delta_D(\boldsymbol{q}+\boldsymbol{q}')P_0(q)$$



$$P_{1-\mathrm{loop}}(k) = P_{\mathrm{lin}}(k) + P_{22}(k) + P_{13}(k)$$



Backup Slide

Key Idea

- Expand P_{22} integral in Legendre polynomials

$$\frac{1}{2}P_{22}(k) = \frac{1219}{1470}J_{000}(k) + \frac{671}{1029}J_{002}(k) + \frac{32}{1715}J_{004}(k) + \frac{1}{6}J_{2-20}(k) + \frac{1}{3}J_{2-22}(k) + \frac{62}{35}J_{1-11}(k) + \frac{8}{35}J_{1-13}(k)$$

- Each component has the same form:

$$J_{\alpha\beta l}(k) = \int \frac{d^3 q_1}{(2\pi)^3} q_1^{\alpha} q_2^{\beta} \mathcal{P}_l(\hat{q}_1 \cdot \hat{q}_2) P_{\rm lin}(q_1) P_{\rm lin}(q_2)$$

- The Fourier transform of $J(k)$ integral is the product of two 1-dim integrals (Hankel transforms)

$$J_{\alpha\beta l}(r) = \frac{(-1)^l}{4\pi^4} I_{\alpha l}(r) I_{\beta l}(r) \qquad I_{\alpha l}(r) = \int dk\ k^{\alpha+2} j_l(kr) P_{\rm lin}(k)$$

- The integrals can be worked out analytically if $P_{\rm lin}$ has a power law form!

Scale Invariance

Ensured by Locality