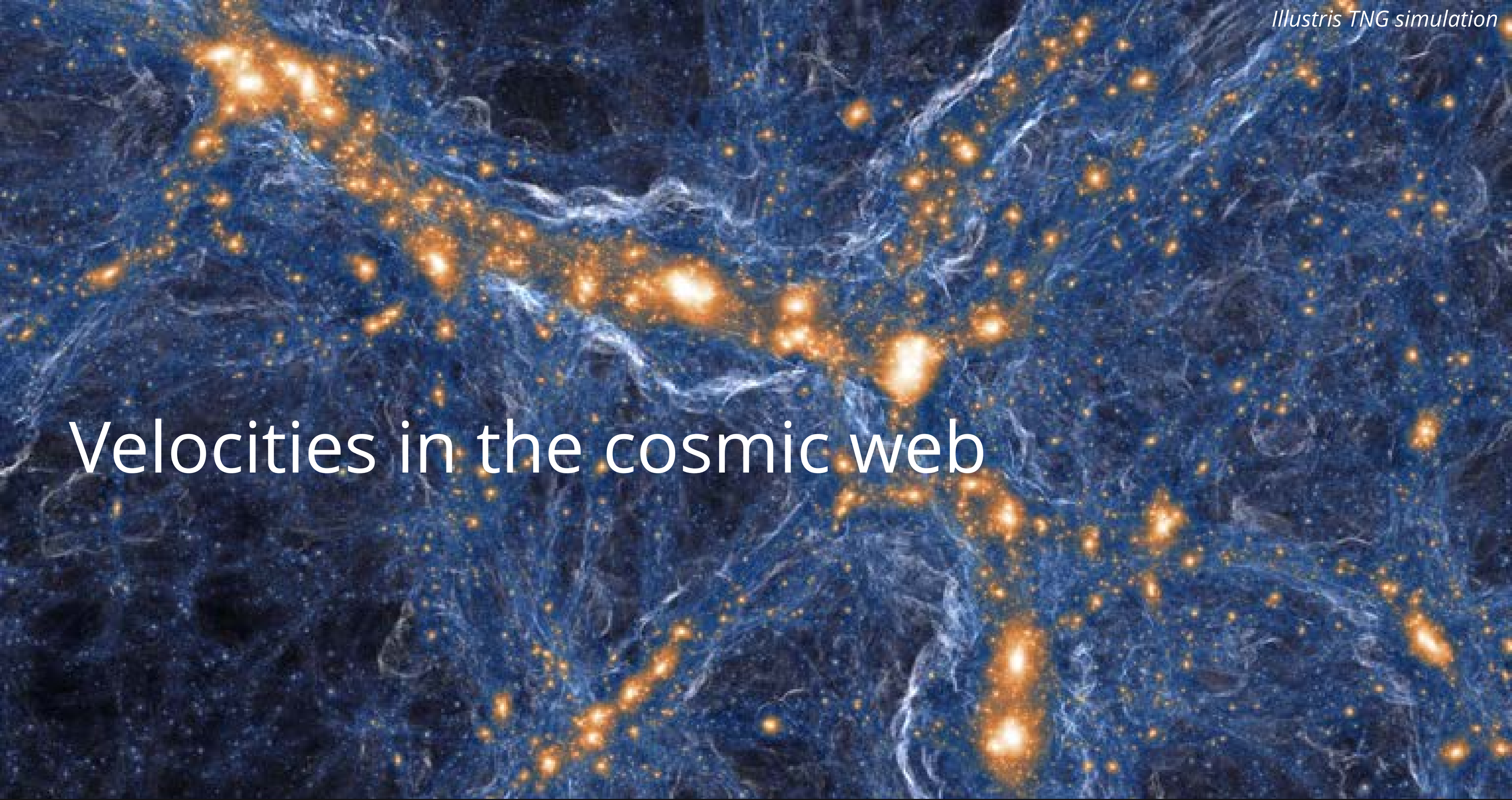
Illustris TNG simulation

Field-level inference of growth rate

Corentin Ravoux

Collaborators: Bastien Carreres, Julian Bautista, Damiano Rosseli

Velocities in the cosmic web

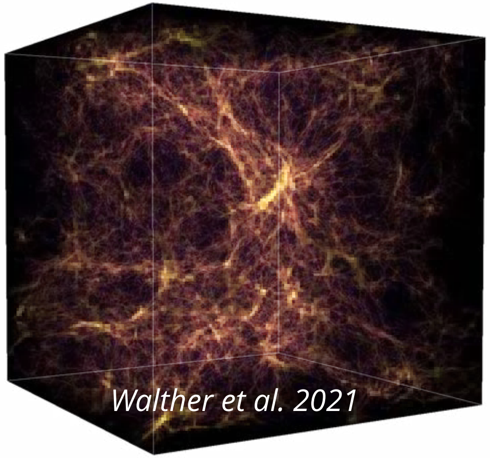


Illustris TNG simulation

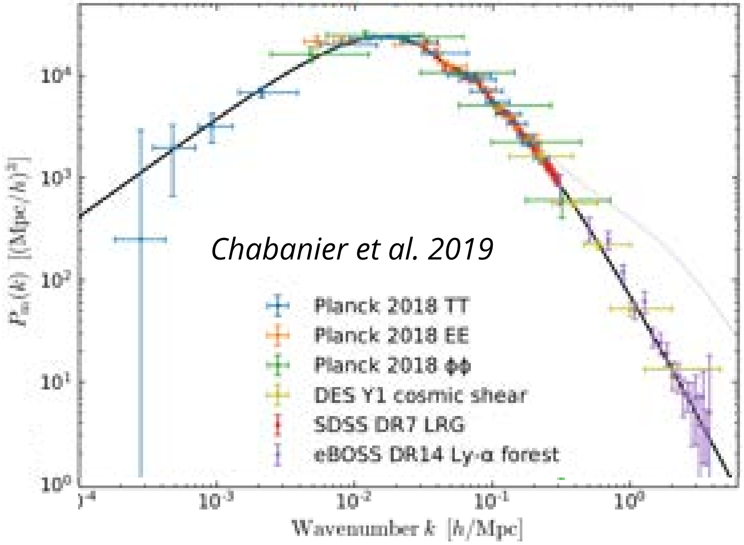
Cosmic web

- Matter perturbations in the primordial Universe grew to form the cosmic web
- **Static description**: matter density contrast (δ) follows linear power spectrum at large-scales
- **Dynamic description**: Velocity of the cosmic web in linear perturbation theory:

$$\boxed{\nabla \cdot \mathbf{v} \propto -aHf\delta}$$



Walther et al. 2021



Chabanier et al. 2019

Growth rate

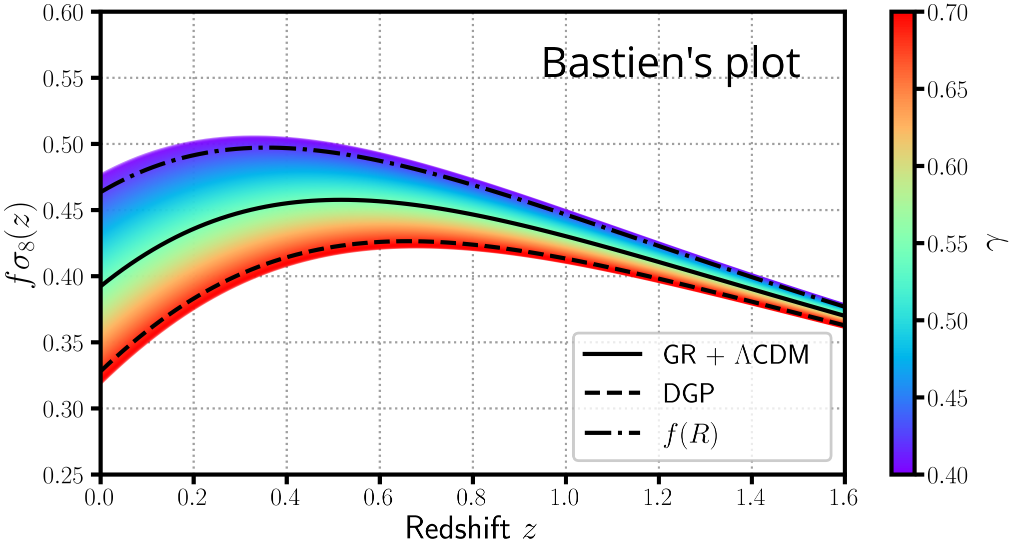
$$\nabla \cdot \mathbf{v} \propto -aHf\delta \quad \longrightarrow \quad \boxed{f = \frac{d\ln \delta_+(a)}{d\ln a}}$$

Logarithmic growth rate of linear perturbations

- Degeneracy with the amplitude of the linear power spectrum

$$P_{\mathrm{m}} \propto \sigma_8^2$$

- **Applications**: Test of general relativity, constraints on dark energy properties



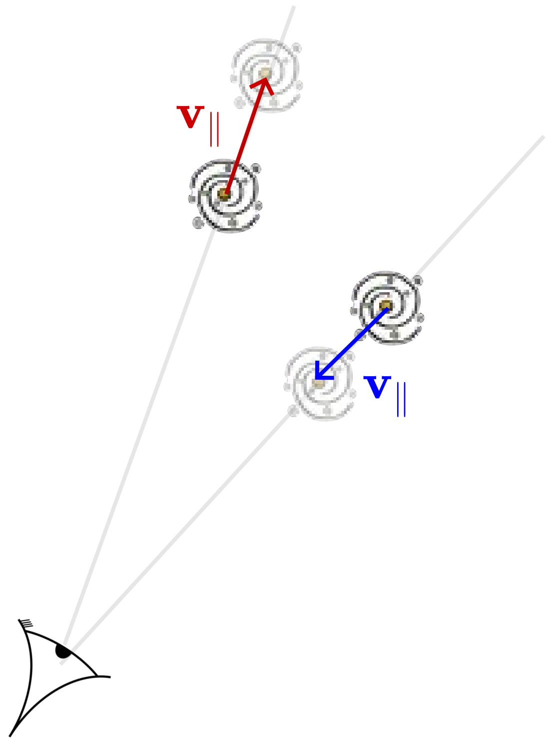
$$f = \Omega_{\mathrm{m}}^{\gamma}$$

Redshift Space Distortion

- Redshifts precisely measured by spectroscopic surveys:
  - Universe expansion + peculiar velocity
  - Position shifted by the line-of-sight peculiar velocity

- **Redshift Space Distortions (RSD)**: Distortion of cosmological observable due to peculiar velocities

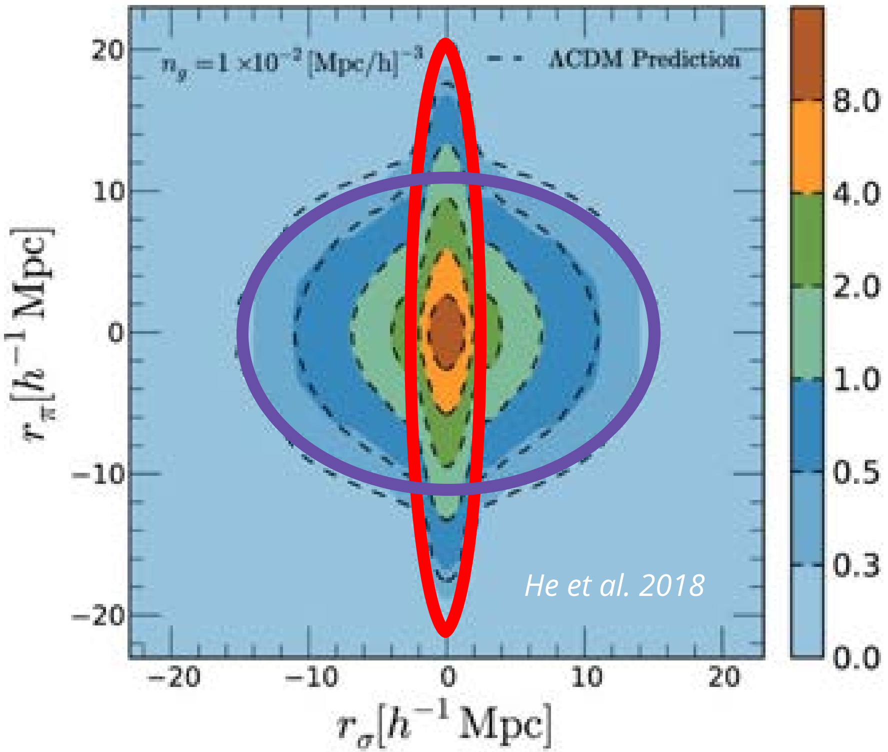
$$(1 + z_{\mathrm{obs}}) = (1 + z_{\mathrm{cosmo}})(1 + z_{\mathrm{peculiar}})$$



RSD example

- Galaxy-galaxy auto-correlation

- **Two effects:**
  - **Linear RSD (or Kaiser):** cohesive movement of galaxies around over densities
  - **Non-linear RSD (or Finger-of-god):** small-scale random velocities due to collapse



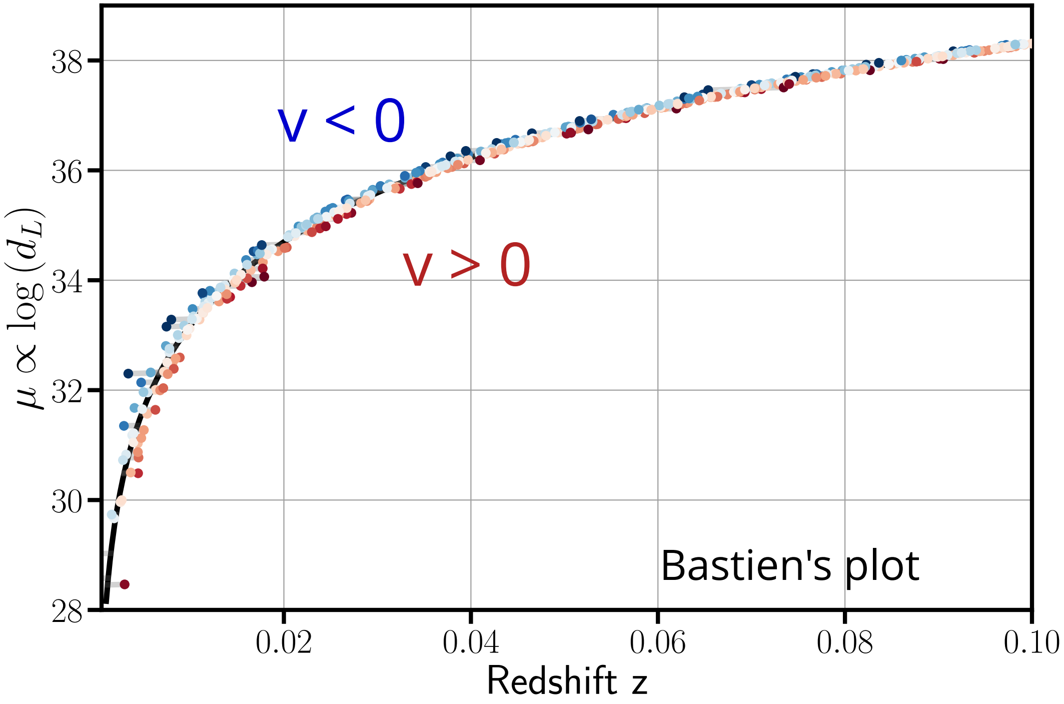
Peculiar velocities

- Direct estimation of the galaxy peculiar velocity.

- Distance estimated with Type Ia supernovae (ZTF, LSST):

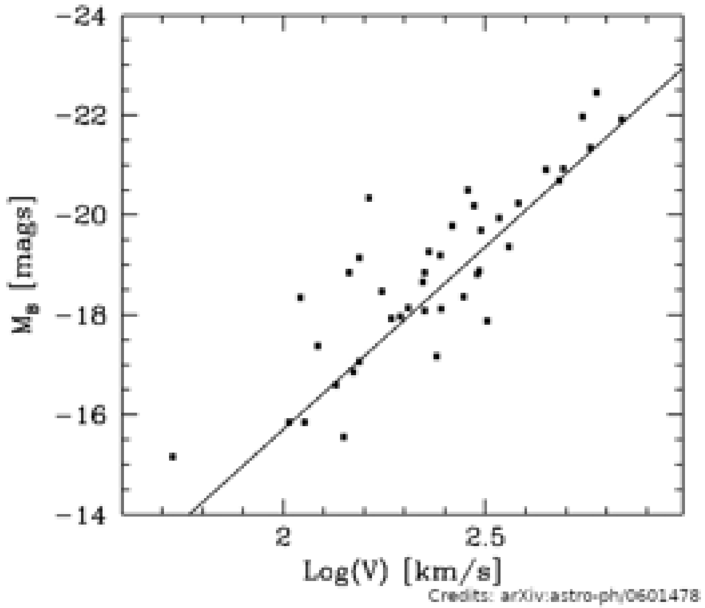
$$\sigma_D / D \sim 7\%$$

- Comparison between distance and redshift gives velocity

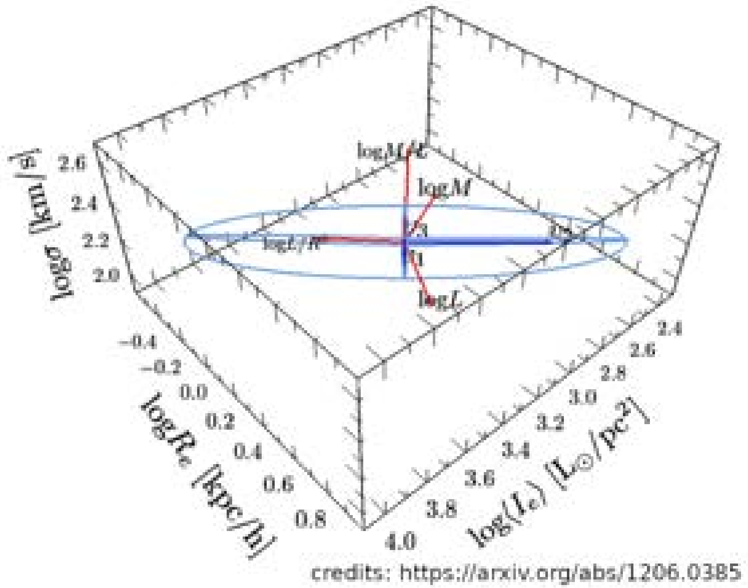


Peculiar velocities

- **Other estimators**: empirical relations $\sigma_D/D \sim 20\%$



Tully-Fisher for spiral galaxies
(luminosity/mass vs rotational velocity)

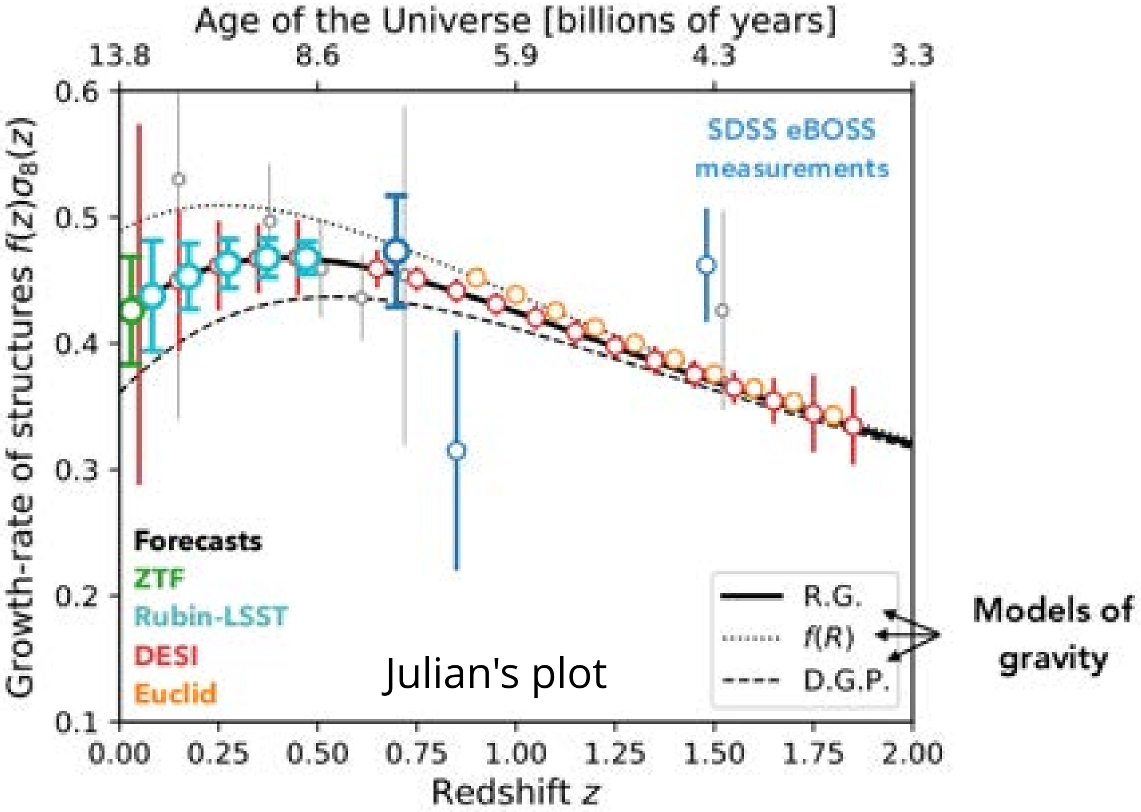


Fundamental plan for elliptical galaxies
(effective angular radius vs velocity dispersion vs luminosity)

Forecasts

- **RSD** very effective for high-z
- **Peculiar velocities** for low-z

- Constraint improvement with combination of methods



Julian's plot

Methods

- Growth rate measurement methods with peculiar velocities:

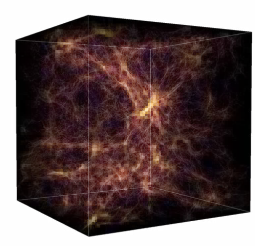
Density/momentum power spectra

$$\langle \delta_{\rm g}(k)\delta_{\rm g}(k)\rangle \; \langle \delta_{\rm g}(k)\mathbf{v}(k)\rangle \; \langle \mathbf{v}(k)\mathbf{v}(k)\rangle$$

Density-velocity comparison

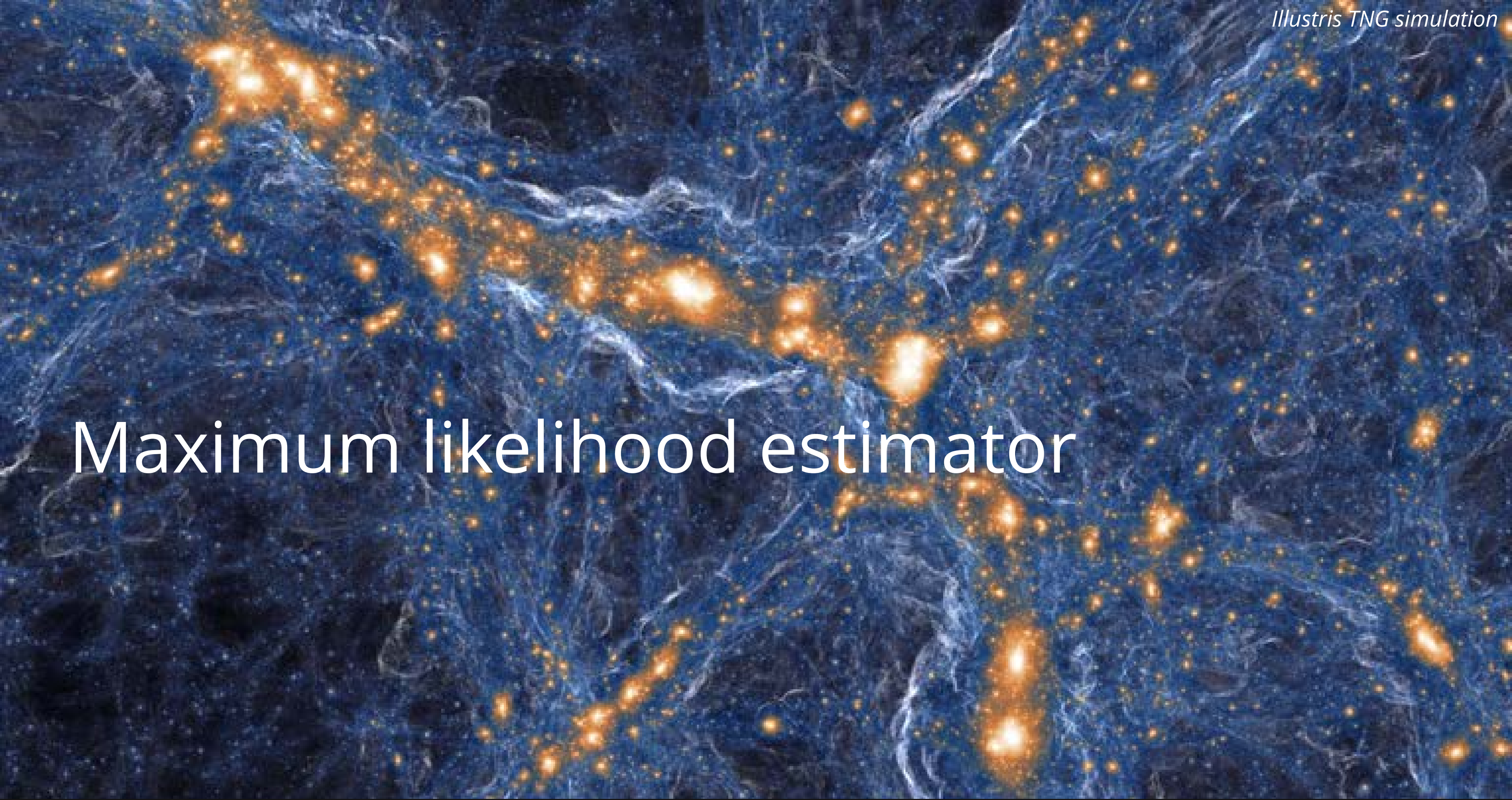
$$\mathbf{v}_{\rm measured} \longleftrightarrow \nabla\cdot\mathbf{v}_{\rm pred} \propto -aHf\delta$$

Forward modeling



$$\longrightarrow (\delta_{\rm g}, \mathbf{v})$$

Maximum likelihood



Illustris TNG simulation

Maximum likelihood estimator

Maximal likelihood estimator

- **Purpose**: Minimize a likelihood computed from all coordinates of the data
- **p = parameters**: contains cosmology and Hubble diagram parameters

$$\mathcal{L}(\boxed{p}) \propto \exp\left[-\frac{1}{2}\boxed{v}^T(p)\boxed{C_{vv}(p)}^{-1}v(p)\right]$$

Carreres et al. 2023

$f\sigma_8 \in p$

peculiar velocities

Covariance matrix computed from theory and SNIa coordinates

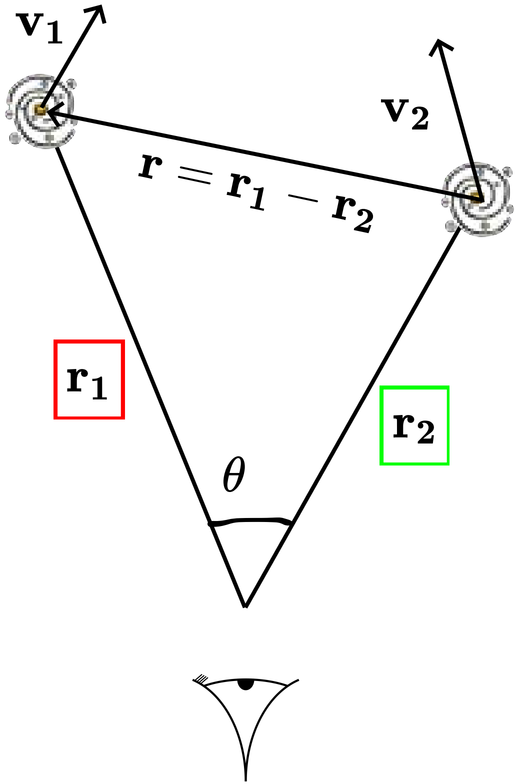
Computation of covariance matrix

$$\boxed{\mu_i} = \mathbf{k} \cdot \mathbf{r_i}$$

$$C_{\mathrm{vv}}(\boxed{\mathbf{r}_1}, \boxed{\mathbf{r}_2}) = \frac{1}{(2\pi)^3} \int_{\mathbf{k}} \boxed{P_{\mathrm{vv}}}(k, \boxed{\mu_1, \mu_2}) e^{i\mathbf{k}.\mathbf{r}} d^3\mathbf{k}$$

$$\boxed{P_{\mathrm{vv}}} = (aHf)^2 \frac{\mu_1 \mu_2}{k^2} P_{\theta\theta}(k) D_u^2(k, \sigma_u)$$

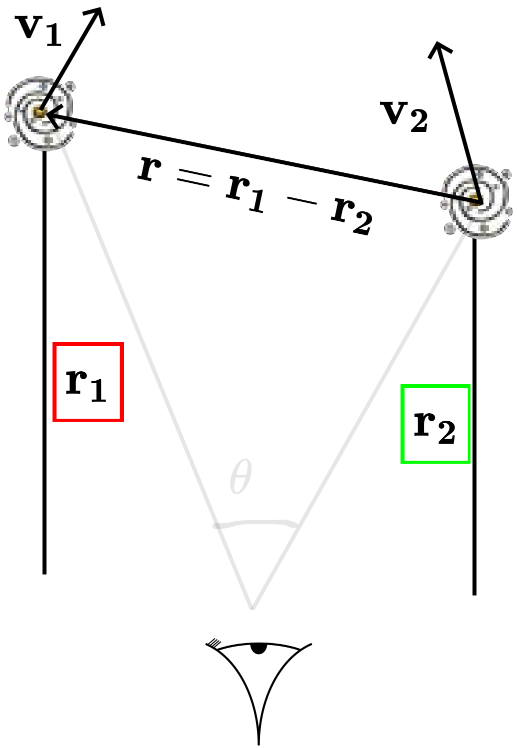
linear perturbation theory



Estimator with wide-angle effects

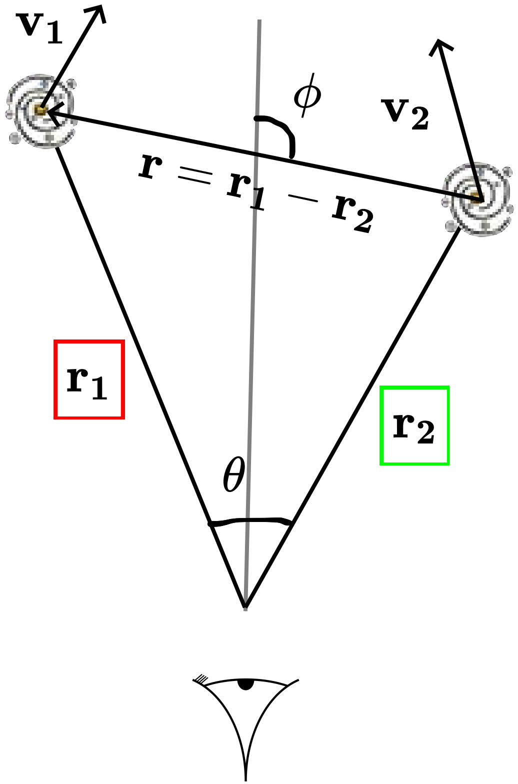
A word on wide-angle in clustering

Plane parallel models	Wide angle models



$$\mu = \mu_1 = \mu_2$$

$$\mu_1 \neq \mu_2$$

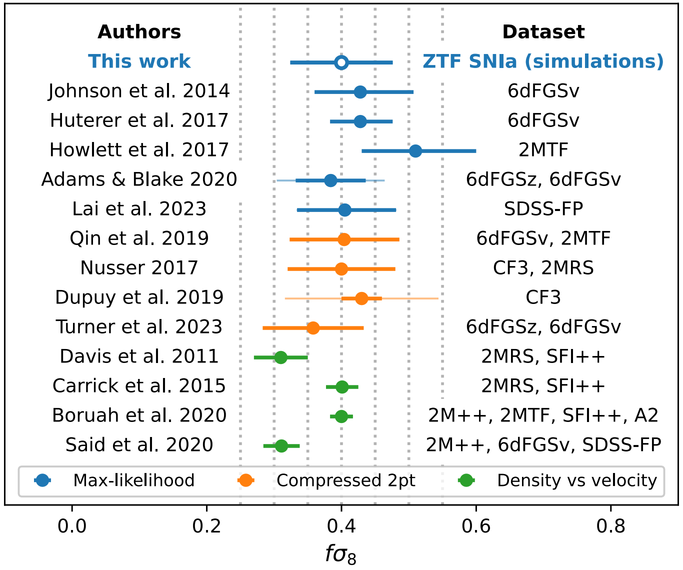


$$\mu_i = \mathbf{k} \cdot \mathbf{r_i}$$

Result for ZTF simulation

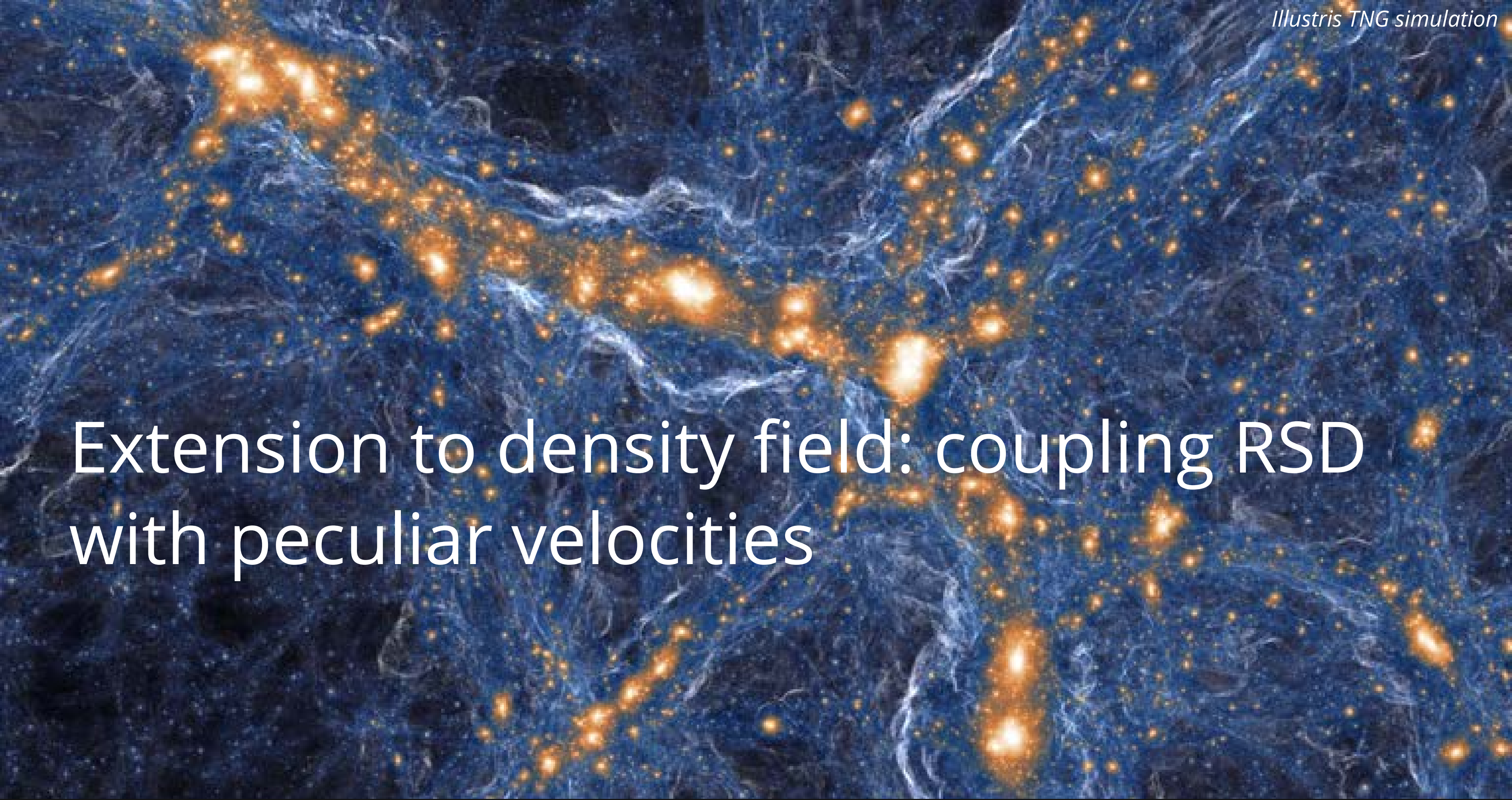
- Use of SNsim package based on SNcosmo
- ZTF survey parameters, and SNIa model

Carreres et al. 2023



- Other measurements performed with FP and TF

- ZTF will be competitive with only 1600 SNIa (compare to ~10k TF/FP)



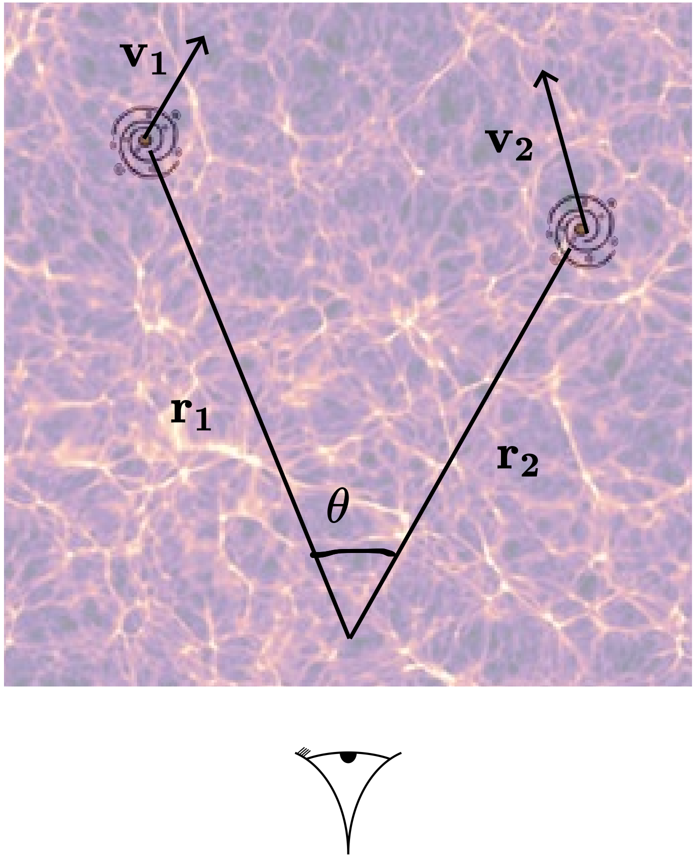
Illustris TNG simulation

Extension to density field: coupling RSD with peculiar velocities

Objective

- Improvement of growth rate measurement using simultaneously velocities and densities

- Improvement of the python package for maximum likelihood fitting



Covariance matrix

$$\mathcal{L}(p) \propto \exp\left[-\frac{1}{2}\begin{bmatrix}\delta \\ v\end{bmatrix}^T \begin{bmatrix}(C_{gg}) & (C_{gv}) \\ (C_{gv}) & (C_{vv})\end{bmatrix}^{-1} \begin{bmatrix}\delta \\ v\end{bmatrix}\right]$$

Density field from galaxy surveys (DESI, Euclid)

Velocities from SNIa (ZTF, LSST)

Covariance matrice computed from theory and coordinates

Linear power spectrum model with wide-angle

$$\begin{cases} P_{\rm gg} = \left[b^2 P_{\rm mm}(k) + bf(\mu_1^2 + \mu_2^2) P_{\rm m\theta}(k) + f^2 \mu_1^2 \mu_2^2 P_{\theta\theta}(k)\right] \exp\left[\frac{-k^2(\mu_1^2+\mu_2^2)\sigma_{\rm g}^2}{2}\right] \\ P_{\rm gv} = iaH\frac{\mu_2}{k^2}\left(bfP_{\rm m\theta}(k) + f^2\mu_1^2 P_{\theta\theta}(k)\right) \exp\left[\frac{-k^2\mu_1^2\sigma_{\rm g}^2}{2}\right] D_u(k,\sigma_u) \\ P_{\rm vv} = (aHf)^2 \frac{\mu_1\mu_2}{k^2} P_{\theta\theta}(k) D_u^2(k,\sigma_u) \end{cases}$$

Very complicated (especially with the exponential term)

$$\longrightarrow C_{\rm ab}(\mathbf{r_1},\mathbf{r_2}) = \frac{1}{(2\pi)^3}\int_{\mathbf{k}} P_{\rm ab}(k,\mu_1,\mu_2) e^{i\mathbf{k}.\mathbf{r}} d^3\mathbf{k} \longrightarrow C_{\rm gg}, C_{\rm gv}, C_{\rm vv}$$

Lai et al. 2022 method

- Taylor expansion of the finger-of-god term

$$\exp\left[\frac{-k^2\mu^2\sigma_\mathrm{g}^2}{2}\right] = \sum_{i=0}^{\infty} \frac{(-1)^i (k\sigma_g)^{2i}}{2^i i!}\mu^{2i}$$

- Complete equations:

$$\mathbf{C}_\mathrm{gg}\left(\mathbf{s_1},\mathbf{s_2}\right) = \sum_{pq}\frac{(-1)^{p+q}}{2^{p+q}p!q!}\sigma_g^{2(p+q)}\sum_l i^l\Big[b^2\xi_{mm,l}^{p,q,0}(s,0)H_{p,q}^l\left(\mathbf{s_1},\mathbf{s_2}\right)$$

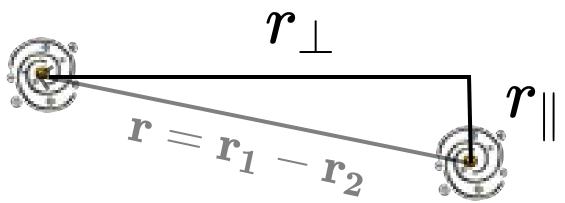
$$+ f^2\xi_{\theta\theta,l}^{p,q,0}(s,0)H_{p+1,q+1}^l\left(\mathbf{s_1},\mathbf{s_2}\right) + bf\xi_{m\theta,l}^{p,q,0}(s,0)\left(H_{p+1,q}^l\left(\mathbf{s_1},\mathbf{s_2}\right) + H_{p,q+1}^l\left(\mathbf{s_1},\mathbf{s_2}\right)\right)\Big]$$

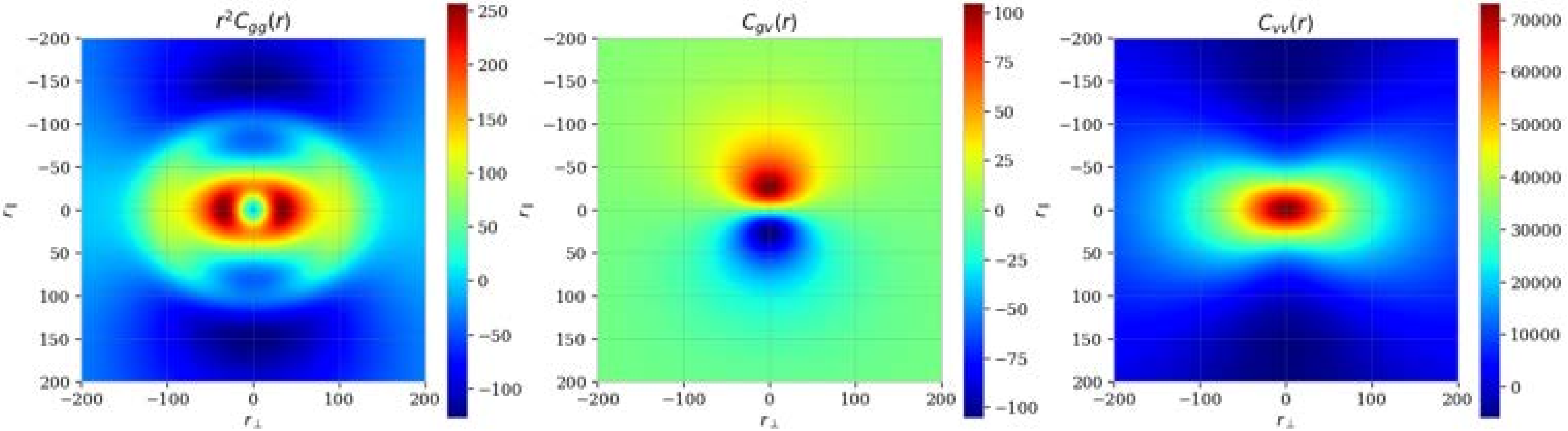
$$\mathbf{C}_{gv}\left(s,\sigma_u\right) = (aHf)\sum_p\frac{(-1)^p}{2^p p!}\sigma_g^{2p}\sum_l i^{l+1}\left(b\xi_{m\theta,l}^{p,-0.5,1}\left(s,\sigma_u\right)H_{p,0.5}^l\left(\mathbf{s_1},\mathbf{s_2}\right) + f\xi_{\theta\theta,l}^{p,-0.5,1}\left(s,\sigma_u\right)H_{p+1,0.5}^l\left(\mathbf{s_1},\mathbf{s_2}\right)\right)$$

$$\mathbf{C}_{vv}\left(s,\sigma_u\right) = (aHf)^2\sum_l i^{l+2}\xi_{\theta\theta,l}^{-0.5,-0.5,2}\left(s,\sigma_u\right)H_{0.5,0.5}^l\left(\mathbf{s_1},\mathbf{s_2}\right)$$

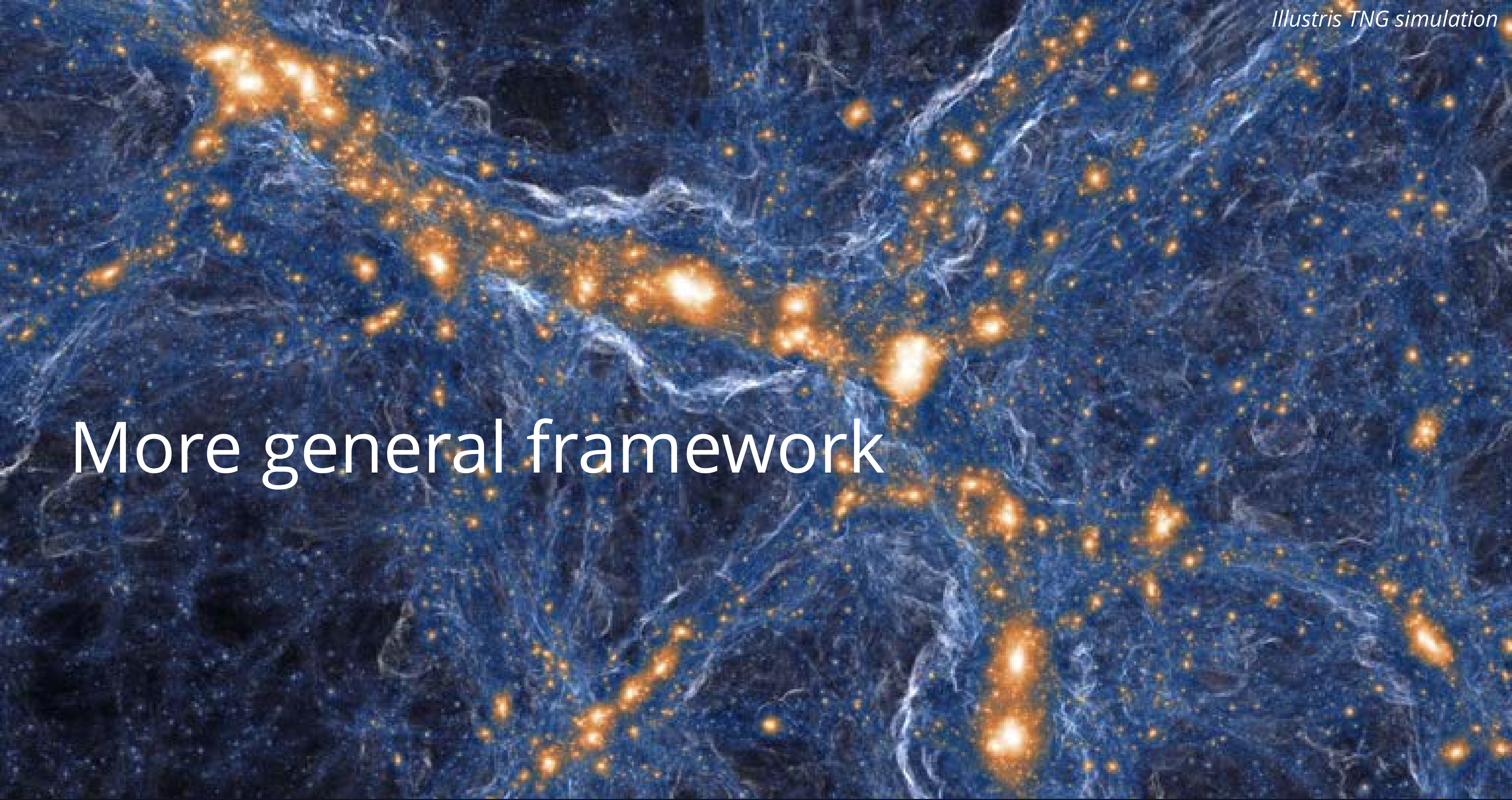
Model evaluation

- **Test**: expressing covariance matrix as a function of parallel and transverse separation





Shape similar to correlation function models



Illustris TNG simulation

More general framework

Idea: general covariance calculation

- **Objectives**:
  - Test more density models
  - Possible extension to non linear corrections

Plane parallel model

- Let's start "simple":

$$C_{\mathrm{ab}}(\mathbf{r_1},\mathbf{r_2}) = \frac{1}{(2\pi)^3}\int_{\mathbf{k}} P_{\mathrm{ab}}(k,\mu)e^{i\mathbf{k}.\mathbf{r}}d^3\mathbf{k}$$

$$\downarrow$$

Using some spherical harmonic definition and multipole theorems

$$C_{\mathrm{ab}}(\mathbf{r_1},\mathbf{r_2}) = \sum_\ell i^\ell \int_k \frac{k^2 dk}{2\pi^2} j_\ell(kr) L_\ell(\hat{\mathbf{d}}\cdot\hat{\mathbf{r}}) \int_{\mu=-1}^{1} \frac{2\ell+1}{2} L_\ell(\mu) P_{\mathrm{ab}}(k,\mu) d\mu$$

Plane parallel model

- Express power spectrum model in a general way:

$$P_{\mathrm{ab}}(k,\mu) = \sum_i \boxed{A_{\mathrm{ab},i}}\,\boxed{B_{\mathrm{ab},i}}(k,\mu)\,\boxed{P_{\mathrm{ab},i}}(k)$$

Parameters to fit

Terms to compute analytically

Analytical power spectra terms

- vv example (Bastien model):

$$P_{\mathrm{vv}} = \boxed{(f^2)} \times \boxed{\left(a^2 H^2 \frac{\mu^2}{k^2}\right)} \times \boxed{\left(P_{\theta\theta}(k) D_u^2(k,\sigma_u)\right)}$$

Plane parallel model

- Without loosing generality we can express the covariance:

$$C_{\mathrm{ab}}(\mathbf{r_1}, \mathbf{r_2}) = \sum_i A_{\mathrm{ab},i} \sum_l \xi^i_{\mathrm{ab},\ell}(r)$$

$$\xi^i_{\mathrm{ab},\ell}(r) = L_\ell(\hat{\mathbf{d}} \cdot \hat{\mathbf{r}}) \left[i^\ell \int_k \frac{k^2 dk}{2\pi^2} P_{\mathrm{ab},i}(k) K^i_{\mathrm{ab},\ell}(k) j_\ell(kr) \right]$$

$$K^i_{\mathrm{ab},\ell}(k, \theta, \phi) = \int_{\mu=-1}^{1} \frac{2\ell+1}{2} L_\ell(\mu) B_{\mathrm{ab},i}(k, \mu) d\mu$$

Plane parallel model

- **Some insights:**

$$C_{\mathrm{ab}}(\mathbf{r_1}, \mathbf{r_2}) = \sum_i \boxed{A_{\mathrm{ab},i}} \sum_l \xi^i_{\mathrm{ab},\ell}(r)$$

Parameters
to fit

linear sum of matrix

Plane parallel model

- **Some insights:**

$$C_{\mathrm{ab}}(\mathbf{r_1}, \mathbf{r_2}) = \sum_i A_{\mathrm{ab},i} \sum_l \xi^i_{\mathrm{ab},\ell}(r)$$

Hankel transform

$$\xi^i_{\mathrm{ab},\ell}(r) = L_\ell(\hat{\mathbf{d}} \cdot \hat{\mathbf{r}}) \boxed{\left[i^\ell \int_k \frac{k^2 dk}{2\pi^2} P_{\mathrm{ab},i}(k) K^i_{\mathrm{ab},\ell}(k) j_\ell(kr)\right]}$$

- **Most important**: Algorithmically optimized way to compute power spectrum integral, with FFTLog

Wide angle model

- More complicated formalism derived and implemented for wide-angle

- **Conclusion**:
  - Formalism can reproduce all recent covariance models (*Adams & Blake 2017, Adams & Blake 2020, Lai et al. 2022, Carreres et al. 2023*)
  - Start extension to other models
  - Fast with FFTLog (Hankel transform)

Wide angle model

$$P_{\mathrm{ab}}(k,\mu_1,\mu_2)=\sum_i A_{\mathrm{ab},i}B_{\mathrm{ab},i}(k,\mu_1,\mu_2)P_{\mathrm{ab},i}(k)$$

$$C_{\mathrm{ab}}(\mathbf{r_1},\mathbf{r_2})=\sum_i A_{\mathrm{ab},i}\sum_l \xi^i_{\mathrm{ab},\ell}(r)$$

$$\xi^i_{\mathrm{ab},\ell}(r)=\sum_j N^{i,j}_{\mathrm{ab},\ell}(\theta,\phi)\left[i^\ell\int_k \frac{k^2dk}{2\pi^2}P_{\mathrm{ab},i}(k)M^{i,j}_{\mathrm{ab},\ell}(k)j_\ell(kr)\right]$$

$$M^{i,j=(\ell_1,\ell_2)}_{\mathrm{ab},\ell}(k)=\int_{\mu_1=-1}^{1}\int_{\mu_2=-1}^{1}\frac{1}{4}L_{\ell_1}(\mu_1)L_{\ell_2}(\mu_2)B_{\mathrm{ab},i}(k,\mu_1,\mu_2)d\mu_1 d\mu_2$$

$$N^{i,j=(\ell_1,\ell_2)}_{\mathrm{ab},\ell}(\theta,\phi)=(4\pi)^2\sum_{m,m_1,m_2}G^{m,m_1,m_2}_{l,l_1,l_2}Y_{\ell m}(\hat{\mathbf{r}})^*Y_{\ell_1 m_1}(\hat{\mathbf{r_1}})^*Y_{\ell_2 m_2}(\hat{\mathbf{r_2}})^*$$

$$G^{l_1,l_2,l_3}_{m_1,m_2,m_3}=\sqrt{\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}}\begin{pmatrix} l_1 & l_2 & l_3\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix} l_1 & l_2 & l_3\\ m_1 & m_2 & m_3\end{pmatrix}$$

flip Field Level Inference Package

- **Currently**: python package for this new formalism

https://github.com/corentinravoux/flip

- **Contains:**
  - Symbolic code generation for covariance model
  - Fast generation of covariance
  - Minuit fitter (MCMC in progress)
  - Gridding routines
  - Covariance model evaluation

A word on symbolic calculation

$$K^{i}_{\mathrm{ab},\ell}(k,\theta,\phi) = \int_{\mu=-1}^{1} \frac{2\ell+1}{2} L_{\ell}(\mu) B_{\mathrm{ab},i}(k,\mu) d\mu$$

Computed analytically

- With **Sympy**, similar to mathematica (but free)

$$\begin{aligned} P_{\mathrm{gg}} = &\exp\left(-(k\mu\sigma_{\mathrm{g}})^2\right) \\ &\left[b^2 P_{\mathrm{mm}}(k)\right. \\ &+ 2bf\mu^2 P_{\mathrm{m}\theta}(k) \\ &\left.+ f^2\mu^4 P_{\theta\theta}(k)\right] \end{aligned}$$

$\xrightarrow{\text{pycode generator}}$

```python
def K_gg_0_4_0(theta, phi):
    return (5 / 16) * np.cos(2 * phi) + (35 / 64) * np.cos(4 * phi) + 9 / 64


def M_gg_1_0_0(sig_g):
    def func(k):
        return -np.exp(-(k**2) * sig_g**2) / (k**2 * sig_g**2) + (
            1 / 2
        ) * np.sqrt(np.pi) * scipy.special.erf(k * sig_g) / (k**3 * sig_g**3)

    return func


def K_gg_1_0_0(theta, phi):
    return 1


def M_gg_1_2_0(sig_g):
    def func(k):
        return (
            -5 * np.exp(-(k**2) * sig_g**2) / (k**2 * sig_g**2)
            - 5
            / 4
            * np.sqrt(np.pi)
            * scipy.special.erf(k * sig_g)
            / (k**3 * sig_g**3)
            - 45 / 4 * np.exp(-(k**2) * sig_g**2) / (k**4 * sig_g**4)
            + (45 / 8)
            * np.sqrt(np.pi)
            * scipy.special.erf(k * sig_g)
            / (k**5 * sig_g**5)
        )

    return func


def K_gg_1_2_0(theta, phi):
    return (3 / 4) * np.cos(2 * phi) + 1 / 4
```

Illustris TNG simulation

Conclusion

- Density/velocity maximum likelihood method
- More general framework with flip