Fast lognormal realizations for multi-probe experiments

David Alonso

December 25, 2016

1 Notation

- χ : the comoving radial distance.
- δ_G : Gaussian matter density field.
- δ : physical matter density field.
- ϕ : Newtonian potential (corresponding to δ_G).
- a_g : Cartesian grid spacing.
- $D(\chi)$: linear growth factor of matter perturbations.
- We use units $c = 1$ throughout.

2 Algorithm

1. Generate a Gaussian realization of the linear density field δ_G and the Newtonian potential ϕ at $z = 0$. This is done by drawing from the matter power spectrum in Fourier space on a Cartesian box large enough to hold a sphere of radius $\chi(z_{\rm max})$ where $z_{\rm max}$ is the maximum redshift of the simulation. From now on a_g is the spacing of the Cartesian grid on which the density field is generated.
2. Use δ_G to produce a physical matter density field δ (by physical we mean $\delta > -1$) in the light-cone (i.e. $\delta(\mathbf{x}) = \delta(t(\chi), \chi\hat{\mathbf{n}})$). Currently two methods are supported for this: the lognormal transformation and Lagrangian perturbation theory (LPT).
3. Use δ and ϕ to generate a set of spherical shells containing the main quantities needed to compute the necessary cosmological observables. The details regarding the interpolation scheme used to go from Cartesian to spherical coordinates, as well as the methods used to compute the different quantities are described below.
4. Transform the lightcone quantities into the main cosmological observables. Currently supported obsevables:
   - Galaxy angular positions and redshifts (including RSDs).
   - Cosmic shear contribution to galaxy shapes.
   - Intensity maps for line-emitting species (including RSDs).
   - Lensing convergence maps for individual source plane redshifts.
   - Integrated Sachs-Wolfe perturbation for individual source plane redshifts.

3 Relevant equations

Given the Cartesian realization of the Gaussian density field δ_G at $z = 0$, we compute the Newtonian potential by solving Poisson's equation in Fourier space:

$$\phi(\mathbf{k}, z=0) = -\frac{3}{2}\Omega_M H_0^2 \frac{\delta_G(\mathbf{k}, z=0)}{k^2}, \tag{1}$$

The radial velocity field is computed in terms of the gradient of the Newtonian potential as

$$v_r(z=0) = -\frac{2f_0}{3H_0^2\Omega_M}(\hat{\mathbf{n}}\cdot\nabla)\phi(z=0), \tag{2}$$

where $\hat{\mathbf{n}} \equiv (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$.

The transverse Hessian of the gravitational potential is computed as

$$\hat{H}_\perp\phi = \hat{R}(\hat{H}\phi)\hat{R}^T \tag{3}$$

where $\hat{H}$ is the Cartesian Hessian operator, and $\hat{R}$ is a rotation matrix:

$$\hat{H} \equiv \left(\begin{array}{ccc} \partial_x\partial_x & \partial_x\partial_y & \partial_x\partial_z \\ \partial_x\partial_y & \partial_y\partial_y & \partial_y\partial_z \\ \partial_x\partial_z & \partial_y\partial_z & \partial_z\partial_z \end{array}\right), \quad \hat{R} \equiv \left(\begin{array}{ccc} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \end{array}\right). \tag{4}$$

The evolution equations for the different quantities are:

$$\delta_G(z) = D(z)\delta(z=0), \quad v_r(z) = \frac{D(z)f(z)H(z)}{f_0H_0}v_r(z=0), \tag{5}$$

$$\phi(z) = (1+z)D(z)\phi(z=0), \quad \dot{\phi}(z) = (1+z)D(z)H(z)[f(z)-1]\phi(z=0) \tag{6}$$

Finally, we obtain the shear (γ_1, γ_2), convergence κ and ISW $\Delta^{\rm ISW}$ fields after integrating along the line of sight:

$$\hat{\Gamma}(\chi) \equiv \left(\begin{array}{cc} \kappa+\gamma_1 & \gamma_2 \\ \gamma_2 & \kappa-\gamma_1 \end{array}\right)(\chi) = 2\int_0^\chi d\chi'\chi'\left(1-\frac{\chi'}{\chi}\right)\hat{H}_\perp\phi(\chi') \tag{7}$$

$$\Delta^{\rm ISW}(\chi) = 2\int_0^\chi d\chi' a(\chi')\dot{\phi}(\chi') \tag{8}$$

4 Generating the physical density field

4.1 Lognormal transformation

At each cell we compute the physical density field in terms of the Gaussian field as:

$$1+\delta(\chi\hat{\mathbf{n}}) = \exp\left[D(\chi)\delta_G(\chi\hat{\mathbf{n}}) - D^2(\chi)\frac{\sigma_G^2}{2}\right], \tag{9}$$

where σ_G^2 is the variance of the Gaussian density field at $z = 0$ and $D(\chi)$ is the linear growth factor of matter perturbations.

4.2 Lagrangian perturbation theory

In this case the density field is generated in two steps:

1. Compute the (first- or second-order) displacement field $\mathbf{\Psi}$ corresponding to δ_G in the lightcone.
2. Interpolate the perturbed positions $\mathbf{x} = \mathbf{q} + \mathbf{\Psi}$ (where $\mathbf{q}$ is the comoving position of each grid point) into a density grid. The code allows for three different interpolation schemes: NGP, CIC and TSC.

4.2.1 LPT notes

Start by writing the displacement in a perturbation series $\mathbf{\Psi} = \sum_{i=1} \mathbf{\Psi}^{(i)}$. The first-order displacement can be computed in terms of the linear density field by solving the equation

$$\nabla \mathbf{\Psi}^{(1)} = -\delta_G. \tag{10}$$

This is done in Fourier space by assuming zero vorticity:

$$\mathbf{\Psi}^{(1)}_{\mathbf{k}} = i\frac{\mathbf{k}}{k^2}\delta_{G,\mathbf{k}}. \tag{11}$$

The second-order displacement can be computed in terms of the first-order displacement as

$$\mathbf{\Psi}^{(2)}_{\mathbf{k}} = -i\frac{\mathbf{k}}{k^2}\Upsilon_{\mathbf{k}}, \quad \Upsilon(\mathbf{x}) = \frac{1}{2}\sum_{i\neq j}\left[\partial_i\Psi^{(1)}_i\partial_j\Psi^{(1)}_j - \partial_j\Psi^{(1)}_i\partial_i\Psi^{(1)}_j\right]. \tag{12}$$

The total displacement in the lightcone is given by $\mathbf{\Psi}(\chi\hat{\mathbf{n}}) = D(\chi)\mathbf{\Psi}^{(1)}(\chi\hat{\mathbf{n}}) + D_2(\chi)\mathbf{\Psi}^{(2)}(\chi\hat{\mathbf{n}})$, where D_2 is the second-order growth factor, which we approximate as $D_2(a) = -\frac{3}{7}D^2(a)[\Omega_M(a)]^{-1/143}$.

5 Cartesian to spherical

The Cartesian fields are interpolated into spherical shells as follows. All shells have a comoving width Δr and are divided into a $N_{\rm pix}$ pixels, where $N_{\rm pix}$ varies as a function of the shell radius as described below. Three pixelization schemes are supported:

- CEA (cylindrical equal-area): with N_θ divisions at equal intervals in $\cos\theta$, and $2N_\theta$ divisions in φ (i.e. $N_{\rm pix} = 2N_\theta^2$).
- CAR (equirectangular projection): with N_θ divisions at equal intervals in θ, and $2N_\theta$ divisions in φ (i.e. $N_{\rm pix} = 2N_\theta^2$).
- HEALPix: in this case N_θ is the `nside` resolution parameter (i.e. $N_{\rm pix} = 12N_\theta^2$).

The size of the voxels defined by these pixels is defined by Δr and $N_\theta(r)$. These are defined by trying to preserve the resolution of the Cartesian grid after interpolation. Thus

- We define $\Delta r = f_r\, a_{\rm grid}$, with $f_r = 1$ by default.
- N_θ is defined for each shell by relating the maximum comoving transverse scale of the voxels at the comoving distance of that shell to the grid spacing. At a shell at comoving distance χ, this is given by $r_\perp^{\rm max} \sim \alpha\chi/N_\theta$, with $\alpha = \sqrt{2\pi}$ for CEA[1], $\alpha = \pi$ for CAR[2] and $\alpha = \sqrt{\pi/3}$ for HEALPix[3]. We determine N_θ as the smallest power of 2 that satisfies $r_\perp^{\rm max} \leq f_\theta\Delta x$ (with $f_\theta = 1$ by default). The requirement that N_θ be a power of 2 is motivated by the need to compute integrated quantities, for which it is convenient to be able to relate any pixel at a given resolution to another pixel at a lower resolution.

While doing the interpolation we compute the relevant quantities in the lightcone that will be used later. These are:

- The radial velocity v_r, proportional to the radial gradient of ϕ. This is always computed, and is needed for RSDs.
- The second derivatives of ϕ in the transverse directions. This is only computed if any lensing quantity (galaxy shear or convergence maps) is required.

[1]This is a mere approximation. The largest scale probed by a pixel for CEA corresponds to the width in θ of the pixel closest to the pole, and in that case the resolution would scale with $\arccos(1-2/N_\theta)$ instead of $1/N_\theta$. However, this scaling causes an excessive number of pixels at large χ, and thus we use the scaling $\sqrt{2\pi}/N_\theta$, corresponding to the square-root of the pixel area.

[2]The largest angle in this case corresponds to the side of pixels around the equator.

[3]Here we use the square root of the pixel area.

- The time derivative of ϕ, $\dot{\phi}$. This is only computed if ISW maps are required.

All radial and transverse derivatives are estimated by first computing the gradient or Hessian in Cartesian coordinates using finite differences in the box and then rotating into spherical coordinates (the corresponding equations are given above). In the case of lensing and ISW, the final quantities are integrated along the line of sight after interpolation.

The interpolation is carried out in three steps:

- First, each spherical voxel is sub-divided into $N_r^{\rm sub}\times(N_\theta^{\rm sub})^2$ sub-voxels, where $N_r^{\rm sub}$ is the number of divisions taken in the radial direction and $N_\theta^{\rm sub}$ is the number of divisions in each of the two transverse directions.
- Each sub-voxel is then assigned a value of the corresponding field (δ, v_r, $\hat{H}_\perp\phi$ or $\dot{\phi}$) by using tri-linear interpolation on the sub-voxel centre. Specifically, let (x,y,z) be comoving Cartesian coordinates of the sub-voxel, and let (i,j,k) denote the cell in the Cartesian grid such that

$$x_i\leq x<x_{i+1},\quad x_j\leq y<x_{j+1},\quad x_k\leq z<x_{k+1}, \tag{13}$$

  where $x_i\equiv i\,\Delta x$. Let $f_{i,j,k}$ be the value of the corresponding field in the Cartesian grid denoted by (i,j,k), and let $h_x=(x-x_i)/\Delta x$ etc.. Then, the field value assigned to the sub-voxel is given by:

$$\begin{aligned}
f(x,y,z)=&f_{i,j,k}(1-h_x)(1-h_y)(1-h_z)+f_{i,j,k+1}(1-h_x)(1-h_y)h_z+ &(14)\\
&f_{i,j+1,k}(1-h_x)h_y(1-h_z)+f_{i,j+1,k+1}(1-h_x)h_yh_z+ &(15)\\
&f_{i+1,j,k}h_x(1-h_y)(1-h_z)+f_{i+1,j,k+1}h_x(1-h_y)h_z+ &(16)\\
&f_{i+1,j+1,k}h_xh_y(1-h_z)+f_{i+1,j+1,k+1}h_xh_yh_z &(17)
\end{aligned}$$

- The field value assigned to the voxel is then computed as an average over sub-voxels.

6 Sources

At each voxel (defined by its coordinates $\chi,\hat{\mathbf{n}}$), we compute the physical source density for each source type a as

$$n_a(\chi\hat{\mathbf{n}})=\bar{n}_a(\chi)\,\frac{1+B(\delta(\chi\hat{\mathbf{n}}),b_a(\chi))}{\langle 1+B(\delta,b_a(\chi))\rangle}, \tag{18}$$

where $B(\delta,b)$ is a bias model relating the matter and source overdensities. The mean number density $\bar{n}$ is related to the counts as a function of redshift $dn/(d\Omega dz)$ as $dn/(d\Omega dz)=\bar{n}\,\chi^2/H$. Currently two bias models are supported:

$$\text{Model A:}\quad 1+B(\delta,b)=(1+\delta)^b, \tag{19}$$

$$\text{Model B:}\quad 1+B(\delta,b)=\frac{(1+\delta)^b}{(1+\delta^2)^{(b-1)/2}}. \tag{20}$$

Both bias models satisfy two sanity requirements:

$$1+B(\delta,b)\geq 0\ \ \forall\,\delta\in[-1,\infty) \tag{21}$$

$$B(\delta\rightarrow 0,b)=b\,\delta, \tag{22}$$

and the second one is designed such that $B(\delta\rightarrow\infty,b)=\delta$ in order to prevent the galaxy density from blowing up in high-density regions for $b>1$. Note that the average $\langle 1+B\rangle$ is precomputed in a previous step.

At each pixel we sample a number of sources from a Poisson distribution with mean $\lambda\equiv V_{\rm vox}n_a$, where $V_{\rm vox}$ is the comoving volume of each spherical voxel. We then place the resulting number of particles inside each voxel at random within it. Each source is given a cosmological redshift z_c corresponding to its comoving distance by inverting

$$\chi=\int_0^{z_c}\frac{dz}{H(z)}. \tag{23}$$

In addition, each source is given a redshift distortion $\Delta z = v_r$ according to the value of the comoving velocity field in their corresponding voxel. Each source is also assigned two ellipticity components based on the value of the local shear field:

$$\epsilon_1 = 2\gamma_1, \quad \epsilon_2 = 2\gamma_2. \tag{24}$$

It is also possible compute the fully non-linear values of the ellipticities, given by

$$\epsilon_i = 2\frac{1-\kappa}{(1-\kappa)^2 + \gamma_1^2 + \gamma_2^2}\gamma_i. \tag{25}$$

7 Intensity mapping

The brightness temperature for a line-emitting species a in a voxel is

$$T_a(\nu, \hat{\mathbf{n}}) = \bar{T}_a(\nu)\left[1 + \Delta^a(\chi\hat{\mathbf{n}})\right]. \tag{26}$$

where the mean brightness temperature is

$$\bar{T}_a(z) = \frac{3\hbar A_{21} x_2 c^2}{32\pi G k_B m_a \nu_{21}^2}\frac{H_0^2\Omega_a(z)(1+z)^2}{H(z)} \tag{27}$$

and Δ_i^a is the redshift-space overdensity of the line-emitting species smoothed over the voxel. Here x_2 is the fraction of atoms in the excited state, Ω_a is the fractional density of the species, ν_{21} is the rest-frame frequency of the transition and A_{21} is the corresponding Einstein coefficient for emission.

The procedure to generate intensity maps is:

- We cycle over each voxel in the spherical shells for which we have stored the value of the density and velocity fields.
- We compute the overdensity in the voxel using a log-normal model:

$$1 + \Delta^a(\chi\hat{\mathbf{n}}) = \frac{1 + B(\delta(\chi\hat{\mathbf{n}}), b_a(\chi))}{\langle 1 + B(\delta, b_a(\chi))\rangle} \tag{28}$$

- We sub-sample the voxel in $N_{\rm sub}$ random points. Each point is assigned a brightness temperature

$$T_{a,{\rm sub}} = \bar{T}_a(\nu)\,(1+\Delta^a)\frac{V_{\rm vox}}{N_{\rm sub}V_{\rm IMAP}}, \tag{29}$$

  where $V_{\rm IMAP}$ is the comoving volume of the intensity mapping pixel. Each point is also assigned a redshift displacement given by v_r, the value of the lightcone-evolved radial velocity field in the voxel.
- We compute the frequency channel corresponding to each point from their redshift (computed as the sum of its cosmological redshift and the RSD term), as well as the pixel index in that frequency channel corresponding to the angular coordinates of the point. We then add the brightness temperature of this point computed in the previous step to the total brightness temperature in the pixel.

8 Shear

The cosmic shear tensor is integrated along the line of sight by breaking it up into two Riemann integrals as (I and J)

$$\hat{\Gamma}(\chi_i) = \hat{I}_i - \frac{1}{\chi_{i+1/2}}\hat{J}_i, \tag{30}$$

$$\hat{I}_i = \hat{I}_{i-1} + [\hat{H}_\perp\phi](\chi_i)\frac{\chi_{i+1/2}^2 - \chi_{i-1/2}^2}{2}, \quad \hat{J}_i = \hat{J}_{i-1} + [\hat{H}_\perp\phi](\chi_i)\frac{\chi_{i+1/2}^3 - \chi_{i-1/2}^3}{3}, \tag{31}$$

where $\chi_i \equiv i\,\Delta r$. Note that the pixels two adjacent spherical shells may have different resolutions. However, due to the hyerarchical nature of the pixelization scheme (going up one level in resolution implies sub-dividing a pixel into 4 sub-pixels), it is always possible to relate pixels between adjacent shells.

9 Convergence and ISW

The simulations only cover a limited redshift range $z \in [0, z_{\rm max}]$, and therefore for source planes at $z_{\rm src} > z_{\rm max}$, the convergence and ISW maps must be supplemented with fictitious fluctuations beyond $z_{\rm max}$. This is done as follows:

Let $f(\chi_{\rm src}, \hat{\mathbf{n}})$ be an integrated field (in our case either κ or $\Delta^{\rm ISW}$). It is always possible to relate f to the intervening density fluctuations as

$$f(\chi_{\rm src}, \hat{\mathbf{n}}) = \int_0^{\chi_{\rm src}} d\chi F_f(t(\chi), \chi\hat{\mathbf{n}})\, w_f(\chi, \chi_{\rm src}), \tag{32}$$

where w_f is the so-called *window function*, and F_f is a fluctuation (e.g. δ or $\dot{\phi}$. For κ and $\Delta^{\rm ISW}$ these are given by:

$$F_\kappa \equiv \delta, \quad w_\kappa(\chi, \chi_{\rm src}) = \frac{3}{2}\Omega_M H_0^2 \left(1 - \frac{\chi}{\chi_{\rm src}}\right)\frac{\chi}{a(\chi)} \tag{33}$$

$$F_{\rm ISW} \equiv \dot{\phi}, \quad w_{\rm ISW}(\chi, \chi_{\rm src}) = 2a(\chi). \tag{34}$$

We start by splitting f into the contributions covered by the simulation and everything outside:

$$f(\chi_{\rm src}, \hat{\mathbf{n}}) = f_1(\hat{\mathbf{n}}) + f_2(\hat{\mathbf{n}}) \equiv \int_0^\infty d\chi F_f(\chi)\, w_{f,1}(\chi) + \int_0^\infty d\chi F_f(\chi)\, w_{f,2}(\chi) \tag{35}$$

$$w_{f,1}(\chi) = w_f(\chi, \chi_{\rm src})\Theta(\chi; 0, \chi_{\rm max}), \quad w_{f,2}(\chi) = w_f(\chi, \chi_{\rm src})\Theta(\chi; \chi_{\rm max}, \chi_{\rm src}), \tag{36}$$

where $\Theta(\chi; \chi_1, \chi_2)$ is a top-hat window function between χ_1 and χ_2. Since f_1 can be computed from the simulation, and the only remaining task is to draw a random sample of f_2 from the distribution $p(f_2|f_1)$. Assuming Gaussian statistics, this distribution is completely determined by the second order moments of f_1 and f_2, which are given in Fourier space by

$$\begin{aligned}\langle f_{i,\ell,m} f^*_{j,\ell',m'}\rangle &\equiv C_\ell^{f,ij}\delta_{\ell\ell'}\delta_{mm'} && (37)\\ &= \delta_{\ell\ell'}\delta_{mm'}\frac{2}{\pi}\int dk k^2 \int d\chi_1 w_{f,i}(\chi_1) j_\ell(k\chi_1)\int d\chi_1 w_{f,j}(\chi_2) j_\ell(k\chi_2)\, P_F(t(\chi_1), t(\chi_2), k), && (38)\end{aligned}$$

where

$$\langle F_f(t_1, \mathbf{k}) F_f^*(t_2, \mathbf{k}')\rangle \equiv \delta(\mathbf{k} - \mathbf{k}') P_F(t_1, t_2, k). \tag{39}$$

Plugging these into $p(f_2|f_1)$ it's straightforward to prove that $f_{2,\ell,m}$ can be drawn as Gaussian numbers with mean and variance:

$$\langle f_{2,\ell,m}|f_{1,\ell,m}\rangle = \frac{C_\ell^{f,12}}{C_\ell^{f,11}} f_{1,\ell,m}, \qquad {\rm Var}(f_{2,\ell,m}|f_{1,\ell,m}) = C_\ell^{f,22} - \frac{(C_\ell^{f,12})^2}{C_\ell^{f,11}}. \tag{40}$$

For simplicity we will work in the Limber approximation, in which

$$j_\ell(k\chi) = \frac{1}{k}\sqrt{\frac{\pi}{2\ell+1}}\,\delta\left(\chi - \chi_\ell\right). \quad \chi_\ell \equiv \frac{\ell + 1/2}{k} \tag{41}$$

In this case, the second-order moments are

$$C_\ell^{f,ij} = \frac{2}{2\ell+1}\int dk\, w_{f,i}(\chi_\ell)\, w_{f,j}(\chi_\ell)\, P_F(t(\chi_\ell), t(\chi_\ell), k). \tag{42}$$

Since $w_{f,i}(\chi)\, w_{f,j}(\chi) = \delta_{ij}\left[w_{f,i}(\chi)\right]^2$, the cross-correlation is zero, and thus f_2 can be drawn as a zero-mean Gaussian number with variance given by its auto-power spectrum. For lensing convergence and ISW these are given by

$$C_\ell^{\kappa,22} = \frac{2}{2\ell+1}\int dk\, \left[\frac{3}{2}\Omega_M H_0^2 \frac{\ell(\ell+1)}{k^2}\frac{\chi_{\rm src} - \chi_\ell}{\chi_\ell\chi_{\rm src}}\frac{D(\chi_\ell)}{a(\chi_\ell)}\Theta(\chi_\ell; \chi_{\rm max}, \chi_{\rm src})\right]^2 P(k), \tag{43}$$

$$C_\ell^{\rm ISW,22} = \frac{2}{2\ell+1}\int dk\, \left[\frac{3\Omega_M H_0^2}{k^2} D(\chi_\ell)\, H(\chi_\ell)\left[1 - f(\chi_\ell)\right]\Theta(\chi_\ell; \chi_{\rm max}, \chi_{\rm src})\right]^2 P(k), \tag{44}$$

where $P(k)$ is the density power spectrum at $z = 0$.