Fast lognormal realizations for multi-probe experiments

David Alonso

December 21, 2016

1 Notation

χ is comoving radial density. We use units $c = 1$ throughout.

2 Algorithm

- Generate a Gaussian realization of the linear density field δ_G and the newtonian potential ϕ at $z = 0$. For this we use FFTW in a box able to hold a sufficiently large sphere. In what follows, let Δx be the comoving resolution of this Cartesian grid.
- Interpolate the Cartesian grid into spherical shells. These are generated with a width $\Delta r = f_r \Delta x$ (with $f_r = 1$ by default). The pixels are defined using one of three pixelization schemes:
  - CEA (cylindrical equal-area): with N_θ divisions at equal intervals in $\cos\theta$, and $2N_\theta$ divisions in φ.
  - CAR (equirectangular projection): with N_θ divisions at equal intervals in θ, and $2N_\theta$ divisions in φ.
  - HEALPix: in this case N_θ is the `nside` resolution parameter, and there are $12N_\theta^2$ pixels in total.

  Secondly, the largest transverse scale covered by a pixel is given by $r_\perp^{\rm max} \sim \chi\,\alpha/N_\theta$, where χ is the comoving distance to the shell and $\alpha = \sqrt{2\pi}$ for CEA[1], $\alpha = \pi$ for CAR[2] and $\alpha = \sqrt{\pi/3}$ for HEALPix[3]. We determine N_θ as <u>the smallest power of 2 that satisfies</u> $r_\perp^{\rm max} \leq f_\theta \Delta x$ (with $f_\theta = 1$ by default).
- While doing the interpolation we compute the relevant quantities at $z = 0$ that will be used later **TODO: δ_G is actually linearly evolved at this stage. We should also note the σ_G calculation**. 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) are required.
  - 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 interpolation is carried out in three steps:

[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 scales 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.

– 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_G, 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

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

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:

$$f(x,y,z) = f_{i,j,k}(1-h_x)(1-h_y)(1-h_z) + f_{i,j,k+1}(1-h_x)(1-h_y)h_z + \tag{2}$$
$$f_{i,j+1,k}(1-h_x)h_y(1-h_z) + f_{i,j+1,k+1}(1-h_x)h_yh_z + \tag{3}$$
$$f_{i+1,j,k}h_x(1-h_y)(1-h_z) + f_{i+1,j,k+1}h_x(1-h_y)h_z + \tag{4}$$
$$f_{i+1,j+1,k}h_xh_y(1-h_z) + f_{i+1,j+1,k+1}h_xh_yh_z \tag{5}$$

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

- The different fields are evolved according to their linear perturbation growth factors and integrated along the line of sight in the case of lensing and ISW (see details below).
- The evolved field values are used to Poisson-sample sources, give them RSDs and shapes, produce convergence, ISW and intensity maps.

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:

$$\phi(\mathbf{k}, z=0) = -\frac{3}{2}\Omega_M H_0^2 \frac{\delta_G(\mathbf{k}, z=0)}{k^2}, \tag{6}$$

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

$$v_r(z=0) = -\frac{2f_0}{3H_0^2\Omega_M}(\hat{\mathbf{n}}\cdot\nabla)\phi(z=0), \tag{7}$$

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

$$\hat{H}_\perp\phi = \hat{R}(\hat{H}\phi)\hat{R}^T \tag{8}$$

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

$$\hat{H} \equiv \begin{pmatrix} \partial_x\partial_x & \partial_x\partial_y & \partial_x\partial_z \\ \partial_x\partial_y & \partial_y\partial_y & \partial_y\partial_z \\ \partial_x\partial_z & \partial_y\partial_z & \partial_z\partial_z \end{pmatrix}, \quad \hat{R} \equiv \begin{pmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \end{pmatrix}. \tag{9}$$

The evolution equations for the different quantities are:

$$\delta_G(z) = D(z)\delta(z=0), \quad v_r(z) = \frac{D(z)f(z)H(z)}{f_0H_0}v_r(z=0), \tag{10}$$
$$\phi(z) = (1+z)D(z)\phi(z=0), \quad \dot{\phi}(z) = (1+z)D(z)H(z)[f(z)-1]\phi(z=0) \tag{11}$$

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

$$\begin{pmatrix} \kappa + \gamma_1 & \gamma_2 \\ \gamma_2 & \kappa - \gamma_1 \end{pmatrix} (\chi) = 2 \int_0^\chi d\chi' \chi' \left(1 - \frac{\chi'}{\chi}\right) \hat{H}_\perp \phi(\chi') \tag{12}$$

$$\Delta^{\rm ISW}(\chi) = 2 \int_0^\chi d\chi' a(\chi') \dot{\phi}(\chi') \tag{13}$$

4 Sources

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

$$n_a(\chi, \hat{\mathbf{n}}) = \bar{n}_a(\chi) \, \exp\left[D(\chi) b_a(\chi) \left(\delta_G(\chi, \hat{\mathbf{n}}) - D(\chi) b_a(\chi) \sigma_G^2/2\right)\right], \tag{14}$$

where $\bar{n}_a$ is the mean number density of source, and σ_G^2 is the variance of the density field at $z = 0$. $\bar{n}$ is related to $dn/(d\Omega dz)$ as $dn/(d\Omega dz) = \bar{n}\,\chi^2/H$.

Then, 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

$$\chi = \int_0^{z_c} \frac{dz}{H(z)}. \tag{15}$$

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:

$$\epsilon_1 = 2\gamma_1, \quad \epsilon_2 = 2\gamma_2. \tag{16}$$

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

$$\epsilon_i = 2 \frac{1 - \kappa}{(1 - \kappa)^2 + \gamma_1^2 + \gamma_2^2} \gamma_i. \tag{17}$$

5 Intensity mapping

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

$$T_a(\nu, \hat{\mathbf{n}}) = \bar{T}_a(\nu) \left[1 + \Delta_i^a(\chi \hat{\mathbf{n}})\right]. \tag{18}$$

where the mean brightness temperature is

$$\bar{T}_a(z) = \frac{3 \hbar A_{21} x_2 c^2}{32 \pi G k_B m_a \nu_{21}^2} \frac{H_0^2 \Omega_a(z)(1+z)^2}{H(z)} \tag{19}$$

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:

$$1 + \delta^a = \exp\left[D(\chi) b_a(\chi) \left(\delta_G(\chi, \hat{\mathbf{n}}) - D(\chi) b_a(\chi) \sigma_G^2/2\right)\right]. \tag{20}$$

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

$$T_{a,{\rm sub}} = \bar{T}_a(\nu)\,(1+\delta_i^a)\,\frac{v_{\rm vox}}{N_{\rm sub}v_{\rm IMAP}}, \tag{21}$$

  where $v_{\rm vox}$ is the comoving volume of the voxel and $v_{\rm IMAP}$ 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.

- Each intensity mapping pixel is finally divided by the total comoving volume covered by the pixel.

6 Shear

We compute the shear tensor as

$$\hat{T} \equiv \left(\begin{array}{ccc} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{array}\right), \quad \hat{R} \equiv \left(\begin{array}{ccc} \sin\theta\,\cos\phi & \sin\theta\,\sin\phi & \cos\theta \\ \cos\theta\,\cos\phi & \cos\theta\,\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \end{array}\right), \tag{22}$$

$$\hat{t} \equiv \hat{R}\cdot\hat{T}\cdot\hat{R}^T, \quad \hat{\tau} \equiv \left(\begin{array}{cc} t_{11} & t_{12} \\ t_{21} & t_{22} \end{array}\right), \tag{23}$$

$$\hat{\Gamma}(\chi,\hat{\mathbf{n}}) \equiv \int_0^\chi d\chi'\,\hat{\tau}(\chi'\hat{\mathbf{n}})\,\chi'\left(1-\frac{\chi'}{\chi}\right) = \int_0^\chi d\chi'\,\hat{\tau}(\chi'\hat{\mathbf{n}})\,\chi' - \frac{1}{\chi}\int_0^\chi d\chi'\,\hat{\tau}(\chi'\hat{\mathbf{n}})\,\chi'^2. \tag{24}$$

$$\hat{\Gamma}(\chi_i) = \hat{I}_i - \frac{1}{\chi_{i+1/2}}\hat{J}_i, \quad \hat{I}_i = \hat{I}_{i-1} + \hat{\tau}(\chi_i)\frac{\chi_{i+1/2}^2-\chi_{i-1/2}^2}{2}, \quad \hat{J}_i = \hat{J}_{i-1} + \hat{\tau}(\chi_i)\frac{\chi_{i+1/2}^3-\chi_{i-1/2}^3}{3} \tag{25}$$

7 CMB lensing

For $\kappa \equiv {\rm Tr}(\hat{\Gamma})/2$ we compute, from the simulation, $\kappa(\chi_{\rm max})$, where $\chi_{\rm max}$ is the maximum radial distance that fits in the box. In order to get CMB lensing we need $\kappa(\chi_{\rm LSS}) = \tilde{\kappa}_{\rm max} + \Delta\kappa$, where

$$\tilde{\kappa}_{\rm max} \equiv \int_0^{\chi_{\rm max}} d\chi\delta(\chi)\frac{\chi}{2}\left(1-\frac{\chi}{\chi_{\rm LSS}}\right) \tag{26}$$

$$\Delta\kappa \equiv \int_{\chi_{\rm max}}^{\chi_{\rm LSS}} d\chi\delta(\chi)\frac{\chi}{2}\left(1-\frac{\chi}{\chi_{\rm LSS}}\right). \tag{27}$$

The strategy to compute these is:

- We compute $\tilde{\kappa}_{\rm max}$ from $\kappa(\chi_{\rm max})$ using the same strategy used for shear, but taking care to divide by $\chi_{\rm LSS}$ instead of $\chi_{\rm max}$.

- We compute $\Delta\kappa$ as a Gaussian realization constrained to have the right correlation with $\tilde{\kappa}_{\rm max}$. Do do this we start by rewriting the previous equation as

$$\tilde{\kappa}_{\rm max} \equiv \int d\chi w_1(\chi)\delta(\chi\hat{\mathbf{n}}), \quad \Delta\kappa \equiv \int d\chi w_2(\chi)\delta(\chi\hat{\mathbf{n}}), \tag{28}$$

$$w_1(\chi) \equiv \frac{\chi}{2}\left(1-\frac{\chi}{\chi_{\rm LSS}}\right)\Theta(\chi,0,\chi_{\rm max}), \quad w_2(\chi) \equiv \frac{\chi}{2}\left(1-\frac{\chi}{\chi_{\rm LSS}}\right)\Theta(\chi,\chi_{\rm max},\chi_{\rm LSS}). \tag{29}$$

  where $\Theta(x,x_0,x_f)$ is a top-hat function.

The covariance matrix of the two terms is therefore

$$\langle|\tilde{\kappa}_{\max,\ell m}|^2\rangle = \frac{2}{\pi}\int_0^\infty dk\, k^2\, P(k)\, w_{1,\ell}^2(k), \quad \langle|\Delta\kappa_{\ell m}|^2\rangle = \frac{2}{\pi}\int_0^\infty dk\, k^2\, P(k)\, w_{2,\ell}^2(k) \tag{30}$$

$$\langle\mathrm{Re}(\tilde{\kappa}_{\max,\ell m}\Delta\kappa_{\ell m})\rangle = \frac{2}{\pi}\int_0^\infty dk\, k^2\, P(k)\, w_{1,\ell}(k)\, w_{2,\ell}(k), \quad w_{i,\ell}(k) \equiv \int_0^\infty d\chi\, w_i(\chi)\, j_\ell(k\chi) \tag{31}$$

Thus we generate a realization of $\Delta\kappa$ at each multipole order as a Gaussian number with distribution $\mathcal{N}(\mu,\sigma)$, where

$$\mu = \frac{\langle\mathrm{Re}(\tilde{\kappa}_{\max,\ell m}\Delta\kappa_{\ell m})\rangle}{\langle|\tilde{\kappa}_{\max,\ell m}|^2\rangle}\tilde{\kappa}_{\max,\ell m}, \quad \sigma = \langle|\Delta\kappa_{\ell m}|^2\rangle - \frac{\langle\mathrm{Re}(\tilde{\kappa}_{\max,\ell m}\Delta\kappa_{\ell m})\rangle^2}{\langle|\tilde{\kappa}_{\max,\ell m}|^2\rangle} \tag{32}$$