# The CLASS framework for computing cross-correlation spectra

J. Lesgourgues

TTK, RWTH Aachen U.

28.10.2025

LPNHE

Need exact $C_\ell$'s

get $C_\ell$'s from CLASS or CAMB(_source)

more accurate w/o Limber + $K \neq 0$

Need Limber $C_\ell$'s

or

get $S_{\Omega}(k,z)$, $D(z)$, $\delta_x(k,z)$... from CLASS/CAMB and assemble in likelihood

full control

→ $C_\ell^{X\neq Y}$ in public CLASS:

[CMB T] × [CMB E]

[CMB T] × [CMB $\phi$]

[CMB E] × [CMB $\phi$]

[cosmic shear bin i] × [cosmic shear bin j]

[biased tracer bin i] × [biased tracer bin j]

[cosmic shear bin i] × [biased tracer bin j]

→ not implemented (but should be easy):

[CMB T] × [cosmic shear bin i] → done externally with Limber

[CMB T] × [biased tracer bin i] →

[CMB $\phi$] × [cosmic shear bin i] → can be achieved "manually"

[CMB $\phi$] × [biased tracer bin i] →

# General framework for $C_\ell$ spectra

→ this presentation:
only scalar modes, adiabatic IC

→ final integral over modes $q$
(= wavenumber $k$ in flat FLRW).

$$C_\ell^{XY} = 4\pi \int \frac{dq}{q} \Delta_\ell^X(q,\tau_0)\, \Delta_\ell^Y(q,\tau_0)\, \mathcal{P}_\mathcal{R}(q)$$

→ types X implemented in public CLASS:

– CMB modes T, E, B
▷ option for T: select one or more of:
[T+SW], [EISW], [LISW], [Doppler], [Pol.]

– projected CMB lensing potential $\phi$

– cosmic shear for sources in user-defined list of bins with selection function $\frac{dN}{dz} \times$ (Gaussian, Dirac, top-hat),

– galaxy number count following Bonvin & Durrer notations, {1105.5280, 1307.1459 CLASSgal}
▷ user selects one or more of:
[density], [rsd], [lensing], [add. GR corrections]

▷ user defines list of bins with selection function
$\frac{dN}{dz} \times$ (Gaussian, Dirac, top-hat),
bias, magnification bias, evolution bias

- the latter may also account for other tracers (e.g. GW from mergers)
- MultiCLASS: two types of tracers 1808.03528
- GW_CLASS: anisotropies of CGWB 2305.01602

- breakdown in tasks and modules

Task of the harmonic.c module:

$$C_\ell^{XY} = 4\pi \int \frac{dq}{q}\, \Delta_\ell^X(q,\tau_0)\, \Delta_\ell^Y(q,\tau_0)\, \mathcal{P}_\mathcal{R}(q)$$

transfer.c primordial.c

fourier.c if NL

perturbations.c

→ $\Delta_\ell^X(q)$ from line-of-sight integral:

$$\Delta_\ell^X(q) = \int_{\tau_{ini}}^{\tau_0} d\tau \; W^X(\tau)\, S^X(k(q),\tau)\, \phi_\ell^X(q,(\tau_0-\tau))$$

transfer function — window function — source function — radial function

– flat FLRW:

$$\phi_\ell^X(q,(\tau_0-\tau)) = j_\ell(k(\tau_0-\tau))$$ for many X ($T_0$, lensing, density)

$= $ combination of $(j_\ell, j_\ell', j_\ell'')$ for other types ($T_1, T_2, E, B$, galaxies: RSD/GR)

Task of the transfer.c module

$$\Delta_\ell^X(q) = \int_{\tau_{ini}}^{\tau_0} d\tau \; W^X(\tau)\, S^X(k(q),\tau)\, \phi_\ell^X(q,(\tau_0-\tau))$$

fourier.c if NL

↑

perturbations.c

- perturbations.c defines quantities with index

index_pt_<name> → all perturbations necessary to integrate ODE

index_tp_<name> → all combinations needed for observables and other modules, $S^X(q,\tau)$ ("source function")

▷ Dictionary defined in perturbation_sources()

ex: index_pt → index_tp

$\delta_\gamma, \phi, \phi', \theta_b, \theta_b' \longrightarrow S_0^T(q,\tau)$

$\left.\begin{array}{l} \phi, \psi \text{ in Newt.} \\ \eta, h' \text{ in Syn.} \end{array}\right\} \longrightarrow [\Phi' + \Psi']_{G.I.}$

all $\delta_x, \theta_x \longrightarrow \left[\frac{\delta\rho_m}{\rho_m}\right]_{G.I.}$

- fourier.c calls NL algorithm (HMcode, Halofit, OneLoop)

→ stores $r_{nl}(k,\tau) = \left(\frac{P_m^{NL}(k,\tau)}{P_m^L(k,\tau)}\right)^{1/2}$

and/or $\left(\frac{P_{cb}^{NL}(k,\tau)}{P_{cb}^L(k,\tau)}\right)^{1/2}$

→ multiplies some selected $S^X(k,\tau)$ in transfer.c

→ **transfer.c** defines quantities $[W^x(\tau)\,S^x(k,\tau)]$

with index index_tt_<name>_<bin>

Dictionary in transfer_sources()

ex: CMB lensing:

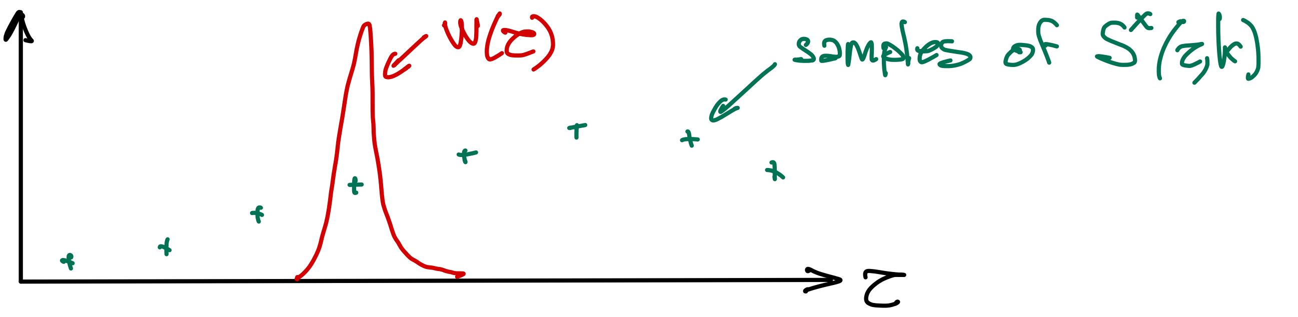
$$[W\cdot S] = \frac{\tau-\tau_{rec}}{(\tau_0-\tau)(\tau_0-\tau_{rec})} \times \overset{\text{perturbations.c}}{\overbrace{[\Phi'+\Psi']_{GI}}} \times \overset{\text{fourier.c}}{\overbrace{\Gamma_{nl}}}$$

ex: cosmic shear:

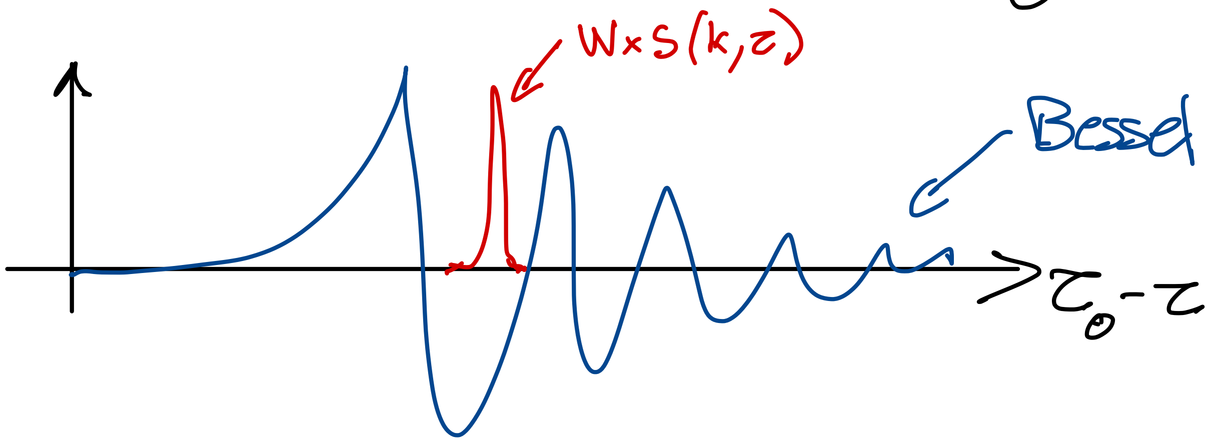
$$[W\cdot S] = \int d\tau_s\, \overset{\text{bin index: } b}{W}(\tau_s) \times \frac{\tau-\tau_s}{(\tau_0-\tau)(\tau_0-\tau_s)} \times \overset{\text{perturbations.c}}{\overbrace{[\Phi'+\Psi']_{GI}}} \times \overset{\text{fourier.c}}{\overbrace{\Gamma_{nl}}}$$

⇒ sometimes $W(\tau)$ is narrow and peaked (galaxy density):

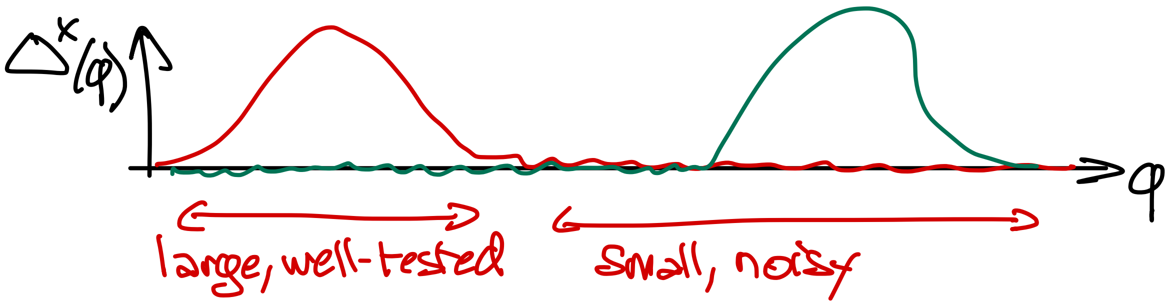
→ triggers re-sampling of $\tau$ in $S^x(\tau,k)$



⇒ resampling also needed when for some $(k,\ell)$ $[W \cdot S]$ becomes less smooth than $j_\ell(k(\tau_0-\tau))$




⇒ Another difficulty in final $C_\ell = \int \frac{dk}{k} \Delta_\ell^X \Delta_\ell^Y \mathcal{P}_\mathcal{R}$ when combining $\Delta_\ell^X(k)$ and $\Delta_\ell^Y(k)$ with different supports:



→ Limber approximation inside CLASS:

- ▷ user chooses $\ell_*$ such that $C_{\ell \geq \ell_*}$ = Limber
- ▷ independent $\ell_*$ for
  - [CMB $\phi$], [cosmic shear] (default $\ell_* = 10$)
  - [biased tracer: density] (default $\ell_* = 100 \bar{z}_i$)
  - [ " " : lensing corr.] (default $\ell_* = 30 \bar{z}_i$)

→ line-of-sight integral replaced by

$$\Delta_\ell^X(q) = \int dz \, [W \cdot S](q, \tau) \, j_\ell(k(\tau_0 - \tau))$$

$$\underset{\text{Limber}}{\simeq} [W.S]\left(q, \frac{\ell+\frac{1}{2}}{q}\right) \sqrt{\frac{\pi}{2\ell}} \frac{\left(1 - \frac{1}{4\ell} + \frac{1}{32\ell^2}\right)}{\ell+\frac{1}{2}}$$

⚠ more complicated for non-flat FLRW and RSD / GR corrections to galaxy number count

→ [W·S] defined in single place.

$\left\{ \begin{array}{l} \text{Either integrated in transfer\_integrate()} \\ \text{or interpolated in transfer\_limber()} \end{array} \right.$

(no redundancy ⇒ no mistakes)

## → Getting $C_\ell^{XY}$ from classy :

- .lensed_cl() returns $C_\ell^{TT,\,EE,\,TE,\,BB,\,\phi\phi,\,T\phi}$

  $E\phi$ omitted from output, can be restored

- raw_cl() returns the same with unlensed TT, EE, TE, BB

- density_cl() returns all combinations of galaxy count / lensing bins

(different functions because CMB x LSS not computed ; behavior easy to change)

## → Getting building blocks for your own Limber :

Goal:

express each $\Delta_\ell^X(k)$ as $(\ldots)\,\delta_m(k, z_*)$ such that:

$$C_\ell = 4\pi \int \frac{dk}{k} (\ldots)^X (\ldots)^Y \underbrace{\delta_m^2(k, z_*)\, \mathcal{S}_R(k)}_{P_m(k,\, z_*)}$$

e.g. [CMB T] x [galaxy density] Stölzner et al.
(LISW) 1710.03238

$$\Delta_\ell^{gal}(k) = \int d\tau \; b(\tau)\, \delta_m(k,\tau)\, j_\ell(k(\tau_0-\tau))$$
$$\simeq b\left(\tau_0-\frac{\ell}{k}\right) \delta_m\left(k, \tau_0-\frac{\ell}{k}\right)$$

$$\Delta_\ell^{LISW}(k) = \int d\tau \left(\phi'+\psi'\right)_{(k,\tau)} j_\ell(k(\tau_0-\tau))$$
$$\simeq 2\phi'\left(k, \tau_0-\frac{\ell}{k}\right)$$
$$\simeq 2 \frac{d}{d\tau}\left(\frac{3}{2}\frac{\Omega_m H_0^2}{a k^2}\delta_m(k,\tau)\right)\Bigg|_{\tau_0-\frac{\ell}{k}}$$
$$\simeq 3\frac{\Omega_m H_0^2}{k^2}\frac{(\delta_m/a)'}{\delta_m}\,\delta_m\left(k,\tau_0-\frac{\ell}{k}\right)$$
$$\simeq 3\frac{\Omega_m H_0^2}{k^2}\left(\frac{(D(\tau)/a(\tau))'}{D(\tau)}\right)_{\tau_0-\frac{\ell}{k}} \delta_m\left(k,\tau_0-\frac{\ell}{k}\right)$$

linear growth factor

$$\Rightarrow C_\ell^{T\times gal} = 4\pi\int\frac{dk}{k}\, b\left(\tau_0-\frac{\ell}{k}\right)\frac{3\Omega_m H_0^2}{k^2}\left(\frac{D/a}{D}\right)'_{\tau_0-\frac{\ell}{k}} \underbrace{\delta_m^2\left(k,\tau_0-\frac{\ell}{k}\right)\mathcal{S}_{\mathcal{R}}(k)}_{P_m\left(k,\tau_0-\frac{\ell}{k}\right)}$$

⇒ Needed from CLASS:

- matter power spectrum $P_m(\tau, z)$
  - pk (k, z)
- linear growth factor
  - scale_independant_growth_factor (z)

⇒ Also available depending on needs:

- scale_dependant_growth_factor(z)
- scale_independant_growth_factor_f(z)

+ H(z), distances(z), etc.

+ all individual perturbations

$\delta_x(k,z)$, $\theta_x(k,z)$, metric fluctuations

always normalised to $\mathcal{R}(k, z \to 0) = 1$

(So they are in fact $\frac{\delta_x(k,z)}{\mathcal{R}(k,0)}$, etc...)

- get_transfer(z)

⇒ approximate results:

- higher-order corrections in Limber
- full line-of-sight corrections at small $\ell$
- corrections to density from lensing, RSD, GR
- corrections from spatial curvature