Analytical 3x2pt Covariance

this goes on every 2nd and 4th Friday of the month @ 15.00 CET

IST:NL leaders
~ 20 people participating, ~ 7 really active

Davide Sciotti & Marco Bonici et al.

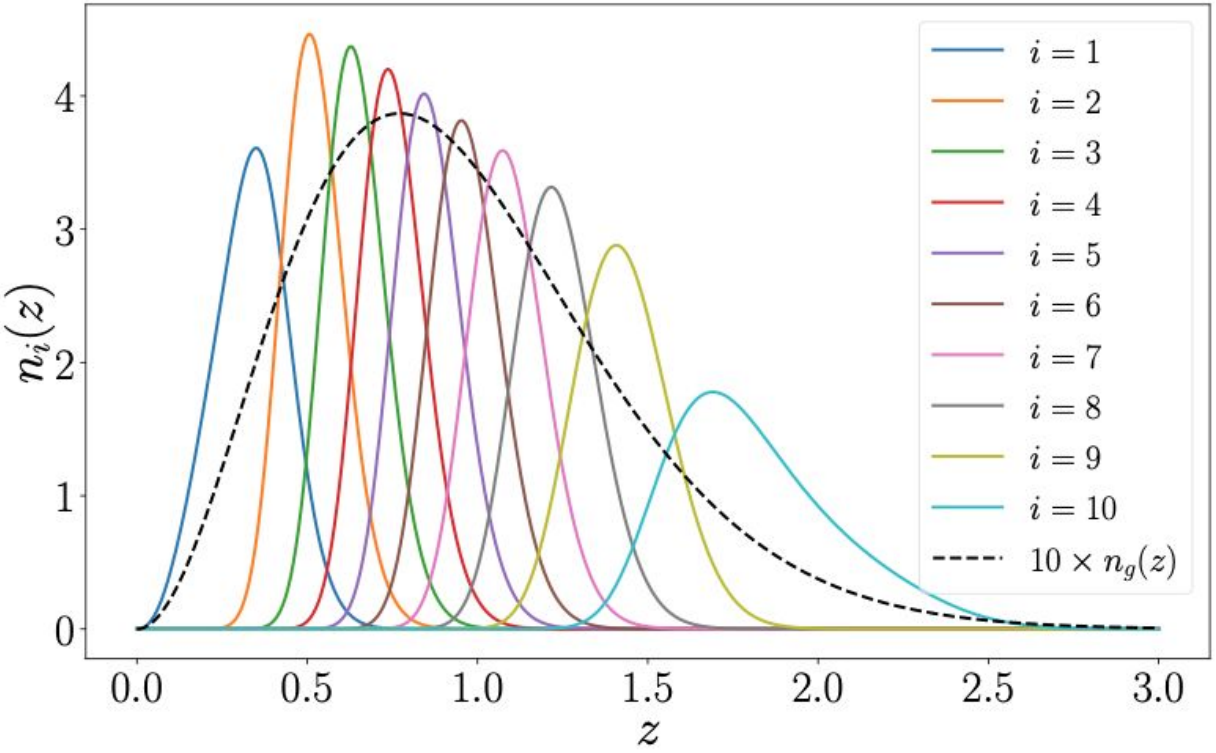
General thoughts

- the covariance will be computed from CLOE recipes but external to the code

- it will be computed once before the MCMC runs

- covariance setups need to be defined: window functions, masks, sky fractions, redshift distributions, non-linear reference models, scale-cuts or not scale-cuts, BNT transformation (**K. Benabed, F. Bernardeau, M. Delaire**, and action in **IST:L**) for the weak lensing part etc…

Photometric Catalogue(s)



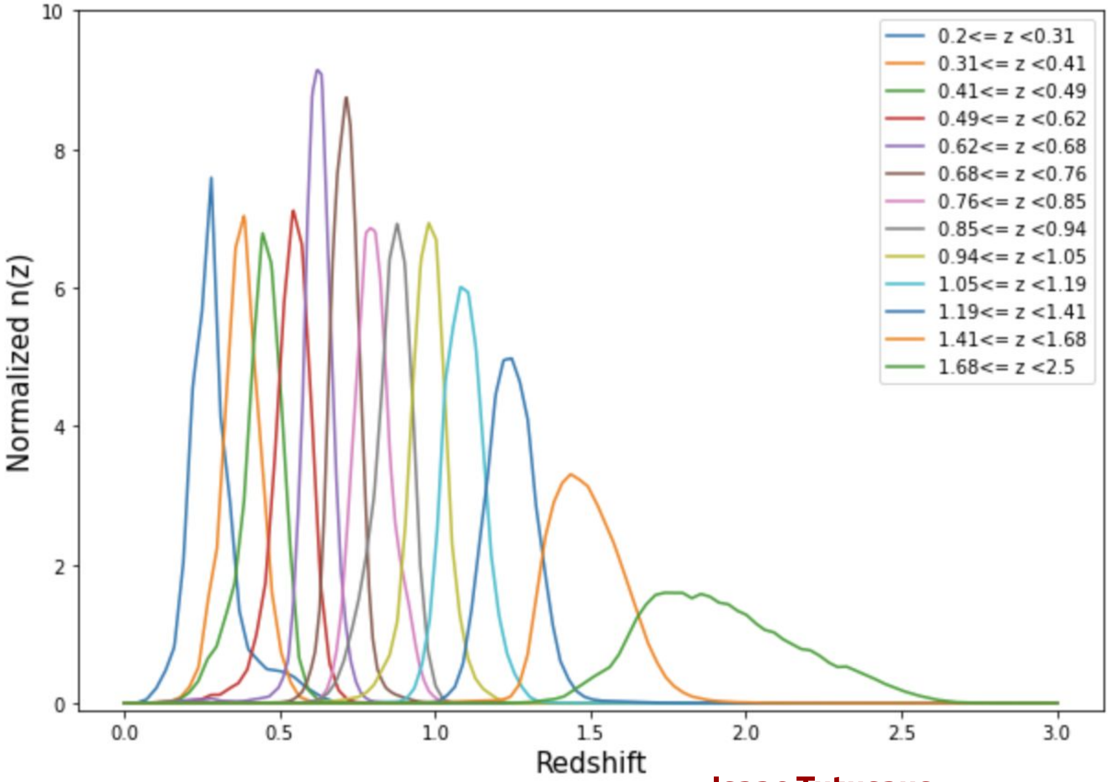
Photometric Catalogue(s)

From the F2 simulations:

After testing different types and number of tomographic bins, the following selection has been chosen as fiducial for both lenses and sources:

- 13 tomographic bins
- equipopulated bins (same number density of galaxies per bin)
- minimum photo-z value: 0.2
- maximum photo-z value: 2.5

24.3 galaxies / arcmin^2



Isaac Tutusaus

Photometric Catalogue(s)

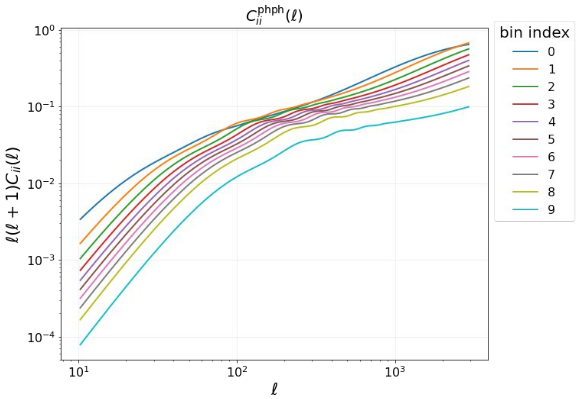
$$W_i^{\mathrm{A}}(z) = b^{\mathrm{A}}(z)\frac{H(z)}{c}n_i^{\mathrm{A}}(z)$$ Clustering

$$W_i^{\gamma}(z) = \frac{3}{2}\left(\frac{H_0}{c}\right)^2\Omega_{\mathrm{m}}(1+z)\chi(z)\int_z^{z_{\max}}\mathrm{d}z'\,n_i^{\mathrm{ph}}(z')\left[1-\frac{\chi(z)}{\chi(z')}\right]$$ Weak Lensing *(+IA components)*

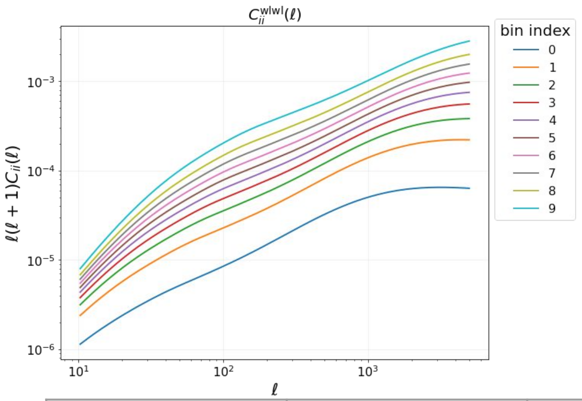
$$C_{ij}^{AB}(\ell) \simeq \frac{c}{H_0}\int_{z_{\min}}^{z_{\max}}\mathrm{d}z\frac{W_i^A(z)W_j^B(z)}{E(z)\chi^2(z)}P_{\delta\delta}\left[k=\frac{\ell+1/2}{r(z)},z\right]$$ In the Limber approximation

Baseline recipes for 3x2pt

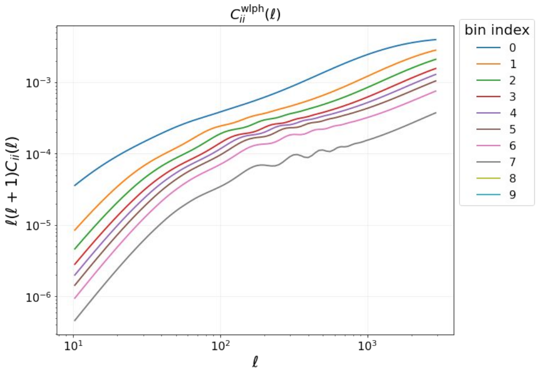
ph-ph



wl-wl



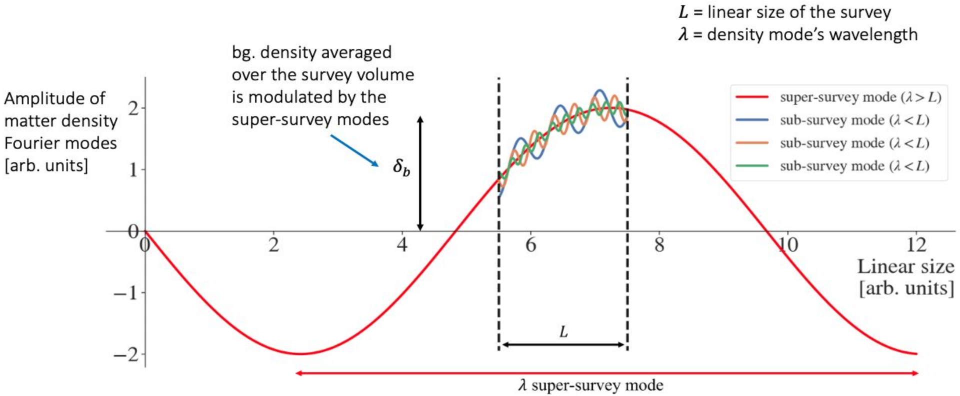
cross: ph-wl



	l_{min}	l_{max}	n_{bin} ($\log_{10}$)
ph-ph	10	3000	200
wl-wl	10	5000	100
ph-wl	10	3000	100

from CLOE: Marco Bonici

Why do we care about Super Sample Covariance?



Power spectrum : all scales probes react to δ_b

→ more important when more modes

All probes react → more important when more probes

Separate universe (e.g. Wagner et al. 2015) :

can mimick δ_b with a change of cosmology

Current Status

- **Gaussian**: published package for easy computation, saves a bit of ordering-related headaches

- **SSC**: three (and a half) codes available
    - *PySSC*:
        - Fast, good control over observables and responses
        - Slowly varying response, may be inaccurate for WL?
    - *PyCCL*:
        - Slower, less control over the observables (but still ok in general)
        - Dirac delta approximation for σ^2, suggested to be less accurate for GCph (**Lacasa+19**)
    - ***Spaceborne***:
        - Slowest
        - No approximation (except full-sky, as with the others)
        - Custom-built, full control over recipe, observables and responses
    - *CosmoLike*:
        - Same recipe as PyCCL, excellent agreement found with input files of R. Upham
        - Runnable on a Docker image by Marco, least control over observables, tests ongoing
        - Already used in the literature

- **cNG**: ongoing tests with PyCCL

SSC Covariance: $\sigma^2(z)$ CosmoLike, PyCCL

$$\mathrm{Cov}^{\mathrm{SSC}}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \int \mathrm{d}\chi \, \frac{q^i_A(\chi) q^j_B(\chi) q^k_C(\chi) q^l_D(\chi)}{\chi^4} \, \frac{\partial P_{AB}(l_1/\chi, z(\chi))}{\partial \delta_b} \, \frac{\partial P_{CD}(l_2/\chi, z(\chi))}{\partial \delta_b} \sigma_b(\Omega_s; z(\chi)),$$

with $\sigma_b(\Omega_s, z(\chi))$ the variance of the background mode over the survey window,

$$\sigma_b(\Omega_s; z) = \int \frac{d^2 k_\perp}{(2\pi)^2} P_{\mathrm{lin}}(k_\perp, z) |\tilde{W}_s(k_\perp, z)|^2 \approx \int \frac{d^2 k_\perp}{(2\pi)^2} P_{\mathrm{lin}}(k_\perp, z) \left[\frac{2 J_1(k_\perp \chi(z) \theta_s)}{k_\perp \chi(z) \theta_s} \right]^2 ,$$

Krause & Eifler 2016
(assumes cylindrical window function)

with q^i_A the weight functions.
Approximation used by DES and KiDS analyses. (KiDS: different pipeline, not public)

- Single redshift integral: approximates $\sigma^2(z)$ as a Dirac delta at $z_1 = z_2$
- If no mask is passed as input, assumes the 3D window function $W(x)$ to be much wider in the radial direction than in the transverse direction.
- **Validity: expected OK for broad overlapping kernels (WL), but not for narrow independent kernels (GC)** - opposite as slowly varying approximation!

For cluster counts, Lacasa+2018 showed it works well in auto-redshift but not in cross-redshift

SSC approximations : slowly varying response (PySSC)

If the response $\frac{\partial \mathcal{O}}{\partial \delta_b}$ varies slowly with redshift compared to $\sigma^2(z_1, z_2)$

$$\mathrm{Cov}_{\mathrm{SSC}}\left[\mathcal{O}(i,\alpha), \mathcal{O}(j,\beta)\right] = \partial_{\delta_b}\mathcal{O}(i,\alpha)\ \partial_{\delta_b}\mathcal{O}(j,\beta)\ S_{\alpha,\beta}$$

= R(i,α) O(i,α) (definition of R)

Output of PySSC
(essentially, $\sigma^2(z_1, z_2)$ term weighted by the kernels)

$\mathcal{O}(i,\alpha)$ cosmological probe: cluster counts, C_l of WL / GCp / XC / CMBX, P(k) of GCs, high order statistic…

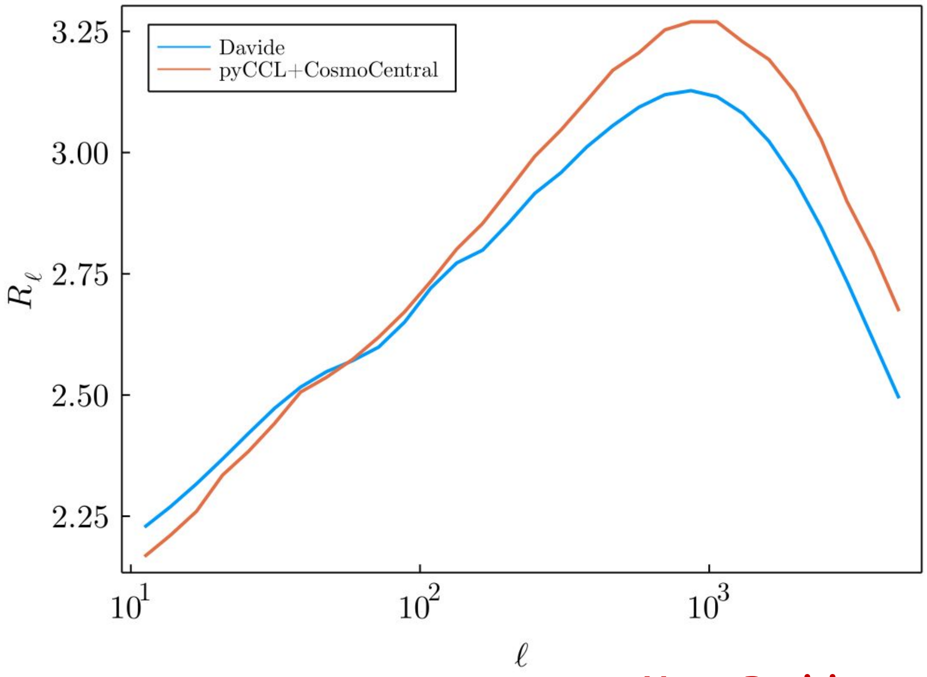
R(i,α) probe response: from theory, ansatz or SU simulations. Depends on probe, redshift (α), scale / other param (i).

$S_{\alpha,\beta}$ covariance matrix of the δ_b of each redshift bin
Can be computed with PySSC (Lacasa & Grain 2019)
Sensitive to sky fraction / mask (Gouyou Beauchamps et al. 2022) - *fsky* rescaling works well for large survey areas (in general SSC does **not** scale as 1/*fsky*)

Validity: **expected OK for narrow independent kernels (GC), worse for broad overlapping kernels with low-z contribution (WL)**

Probe response

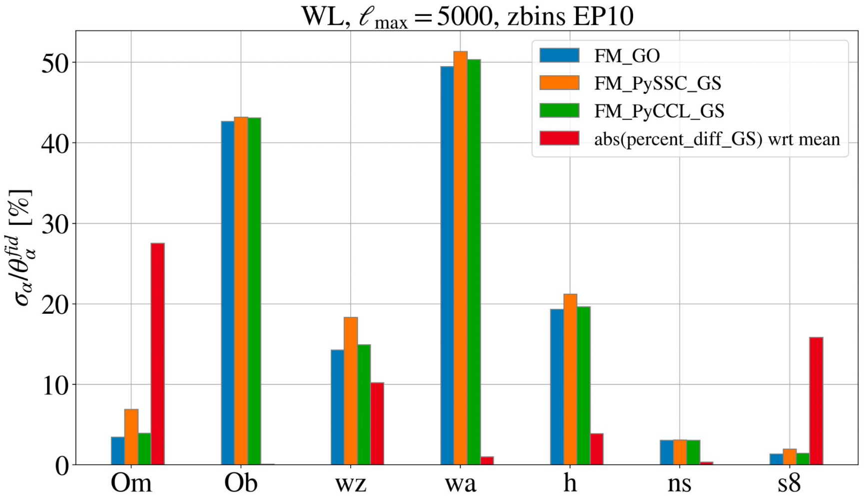
- SU probe response vs probe response from PyCCL (thanks to M. Bonici)
- Good overall agreement
- Next step: gauge the effect of the different recipes by plugging PyCCL responses into PySSC (spoiler alert: small impact on the final constraints)

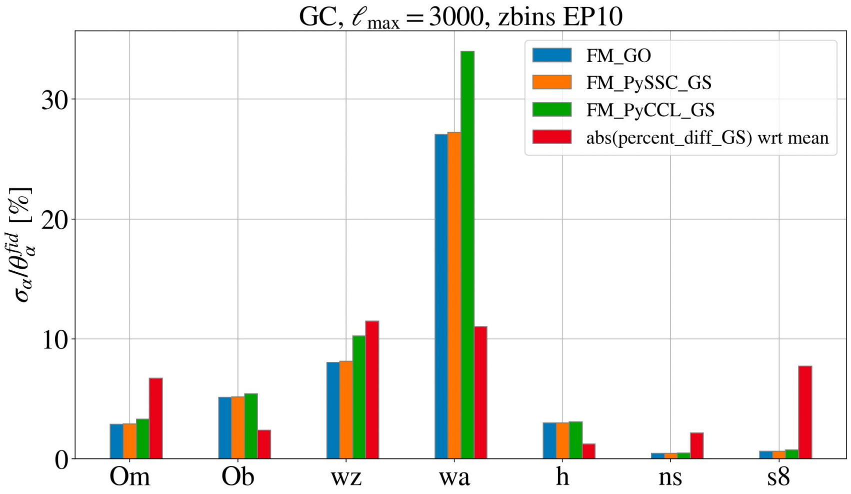


Marco Bonici

PySSC vs PyCCL: FM constraints, WL, GCph

- Same pipeline with SSC matrix from PySSC or PyCCL
- Good agreement found, except on the most impacted parameters for WL
- Difference between GS constraints wrt the *mean* shown in the plot
- Not the best metric for agreement between codes?
  - Direct comparison of the SSC covariance trickier, shows large discrepancies

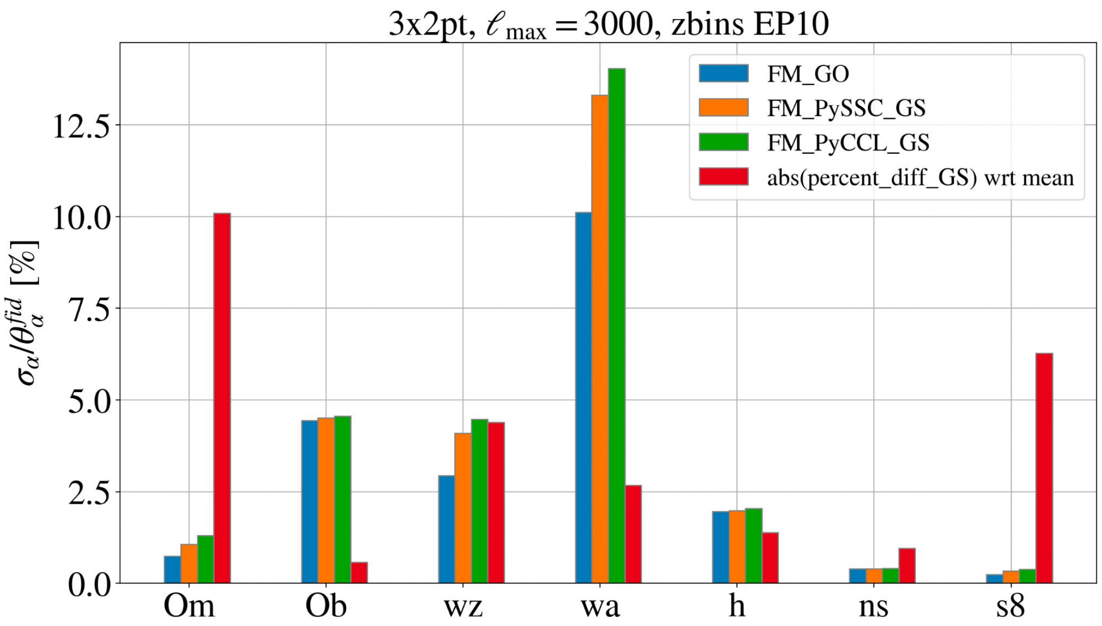




Davide Sciotti

PySSC vs PyCCL: FM constraints, 3x2pt

- Good agreement, 10% max discrepancy
- PyCCL more pessimistic (than PySSC) for GCph, more optimistic for WL

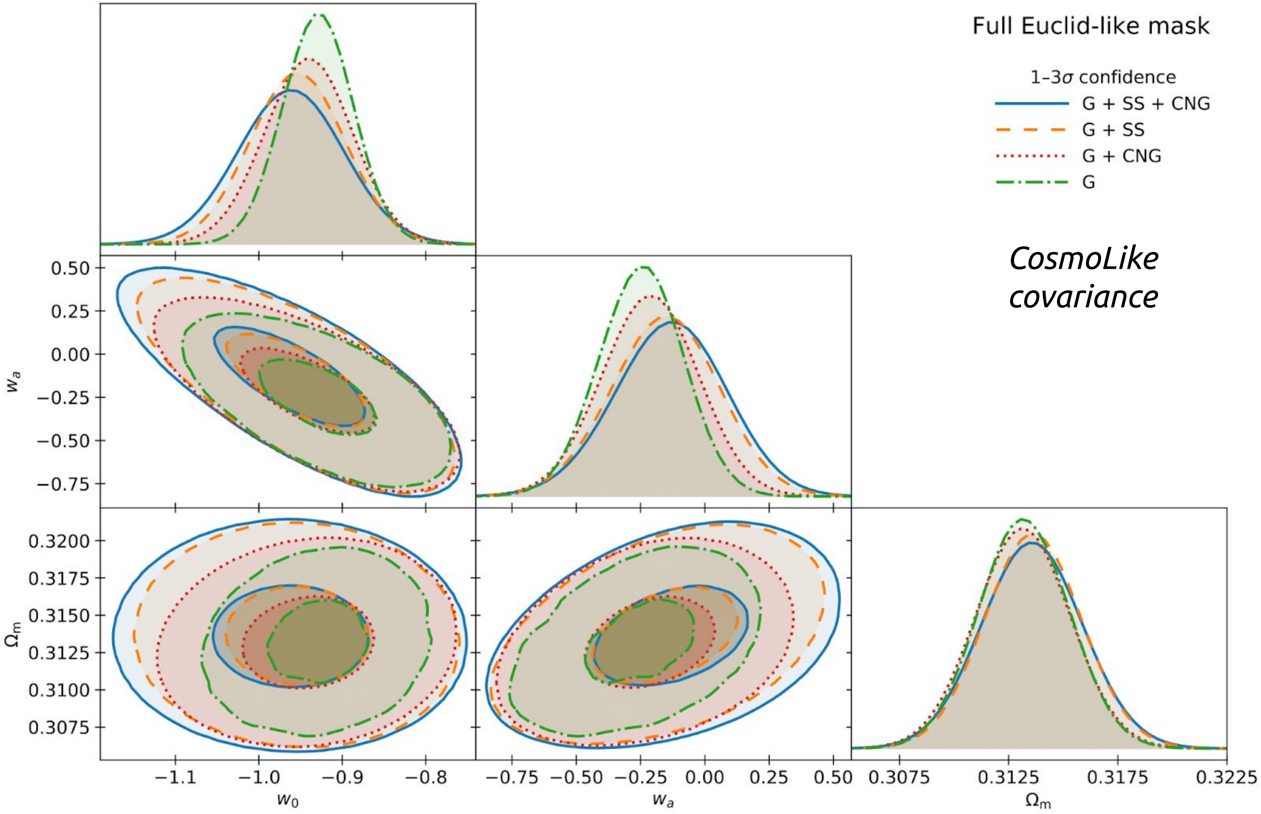


Davide Sciotti

Shear pseudo-Cl covariance

Importance of non-Gaussian terms:

- Mock parameter constraints including different contributions to covariance



CosmoLike covariance

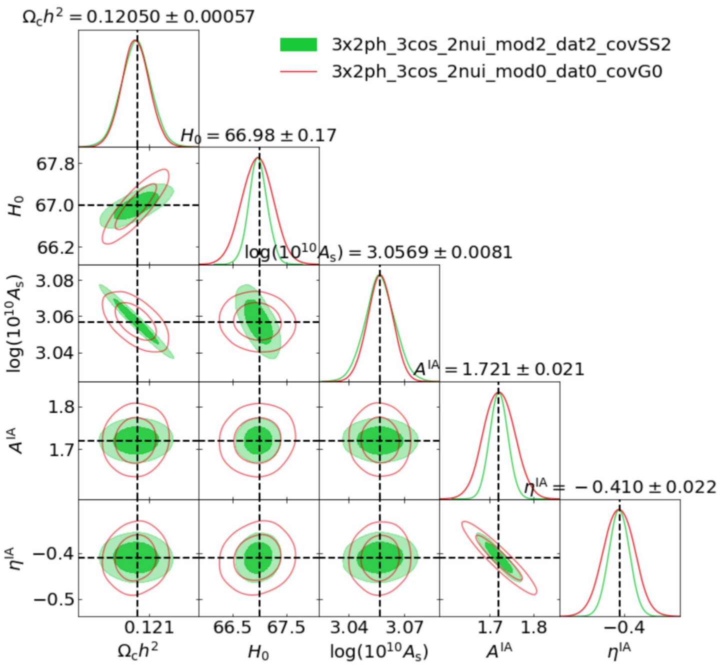
Robin Upham

MCMC analyses

Improvement in constraints using non-linear modelling

3x2pt photometric

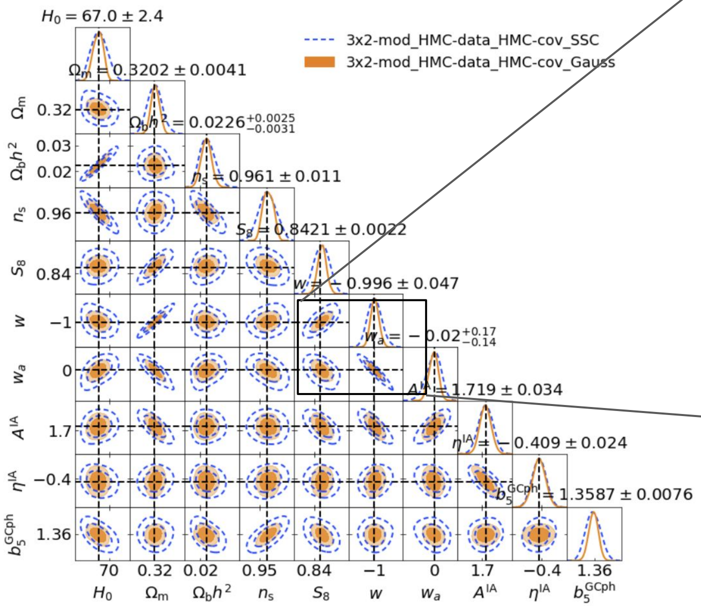
- **Red** lines: Linear model and linear data vector, Gaussian diagonal covariance
- **Green** solid: Nonlinear model HMcode, with same model for data vector and covariance. DM-only.
  SSC + Gaussian covariance.

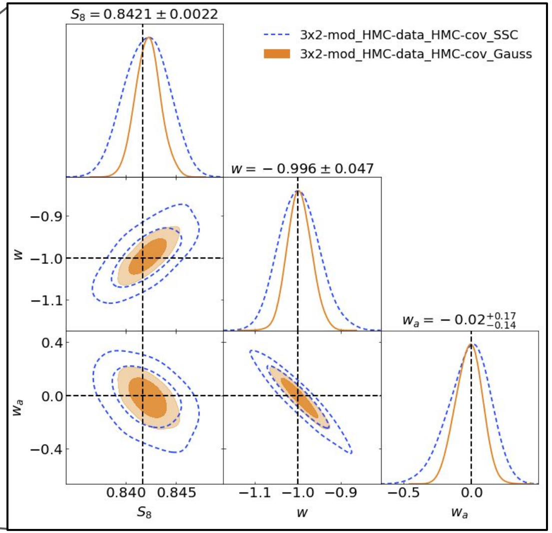


Santiago Casas, Pedro Carrilho & Davide Sciotti

MCMC analyses

Impact of SSC on parameter constraints





3x2pt photometric data: HMCode
model: HMCode
covariance: HMCode

- **Blue dashed** lines: SSC+Gaussian covariance
- **Orange** solid: Gaussian covariance.

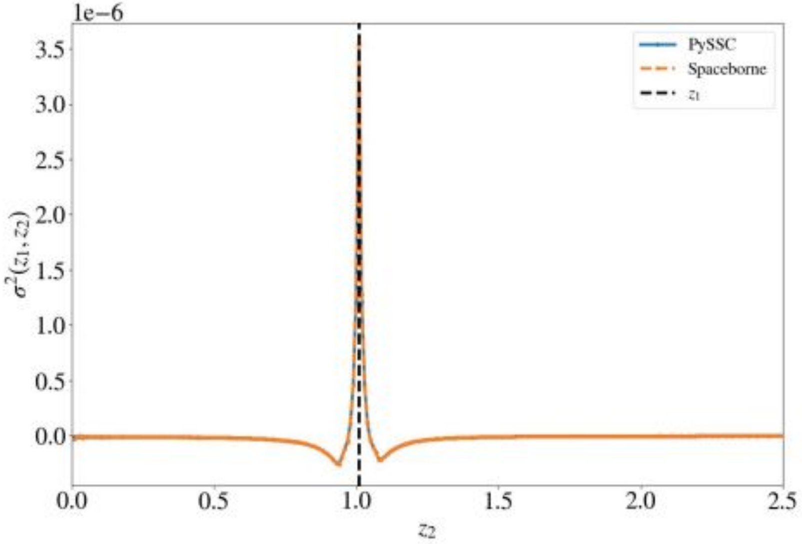
FoM: 401 (with SSC), FoM: 1073 (just Gauss)

Santiago Casas, Pedro Carrilho & Davide Sciotti

Spaceborne SSC

- Possible test of σ^2 against latest version of PySSC, excellent agreement ✅
- Responses from SU simulations, good match against PyCCL (extracted by Marco) ✅
- Including magnification bias (for latest SPV3 forecasts) creates some issues with PySSC and PyCCL, work in progress to bring these up to speed

$$\mathrm{Cov_{SSC}}\left[C_{ij}^{AB}(\ell), C_{kl}^{CD}(\ell')\right] = \int \mathrm{d}V_1 \mathrm{d}V_2 \, W_i^A(z_1)\, W_j^B(z_1)\, W_k^C(z_2)\, W_l^D(z_2) \times \frac{\partial P_{AB}(k_\ell, z_1)}{\partial \delta_b} \frac{\partial P_{CD}(k_{\ell'}, z_2)}{\partial \delta_b} \sigma^2(z_1, z_2)$$



Davide Sciotti & Marco Bonici

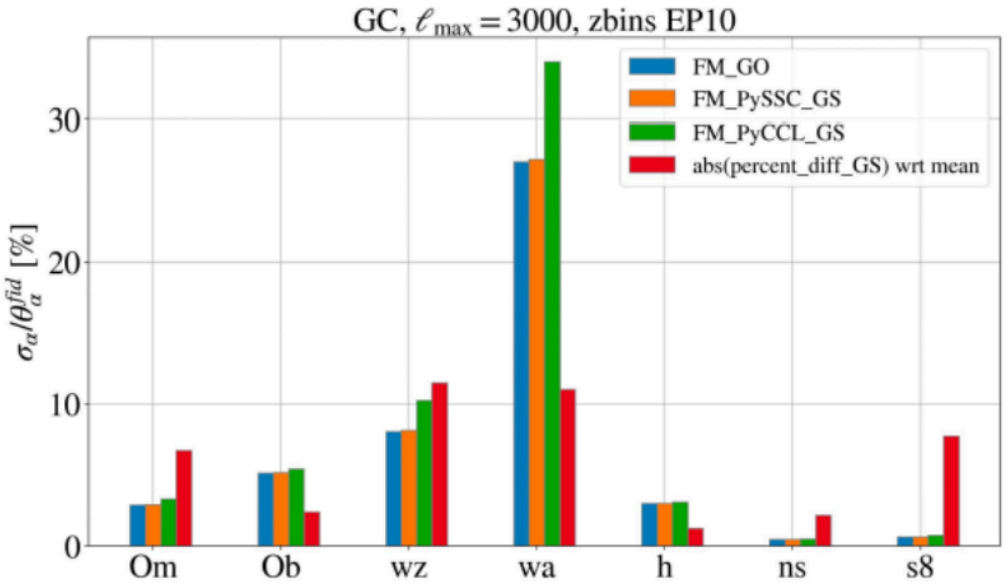
Spaceborne SSC: computing the integral

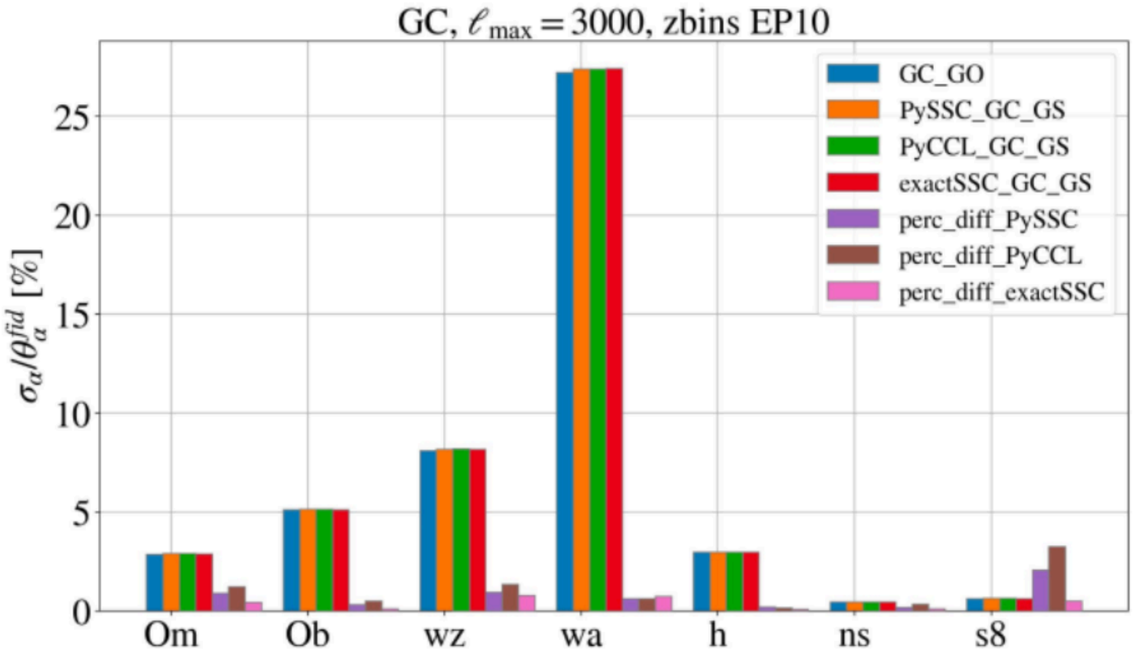
- Previous analysis: proof of concept with 2 \ell bins, integral feasible in Python
- Full matrix (32 \ell bins, 13 *z* bins) **much more** demanding:
  - Developed a Tensorflow version for GPU/TPU, good speed but memory still seems to be the bottleneck
  - `Julia` version by Marco, needs to be run on a cluster (10-ish hours for full matrix) ✅
  - Full control over recipe and approximations

```
 67838032 Sep 19 04:43 cov_SSC_GGGG_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
125984848 Sep 19 03:52 cov_SSC_GGGL_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
 67838032 Sep 19 02:21 cov_SSC_GGLL_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
125984848 Sep 19 01:30 cov_SSC_GLGG_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
233971792 Sep 18 23:55 cov_SSC_GLGL_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
125984848 Sep 18 21:06 cov_SSC_GLLL_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
 67838032 Sep 18 19:31 cov_SSC_LLGG_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
125984848 Sep 18 18:40 cov_SSC_LLGL_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
 67838032 Sep 18 17:09 cov_SSC_LLLL_4D_nbl32_ellmax5000_zbins13_zsteps2899_k1overMpc_conventionPySSC.npy
```

Results GCph

- Low impact in general for GCph
- PyCCL approximation (σ^2 approximated as a Dirac Delta) seems to be good for GCph
- More accurate PyCCL GCph ✅
- Good match with Spaceborne ✅

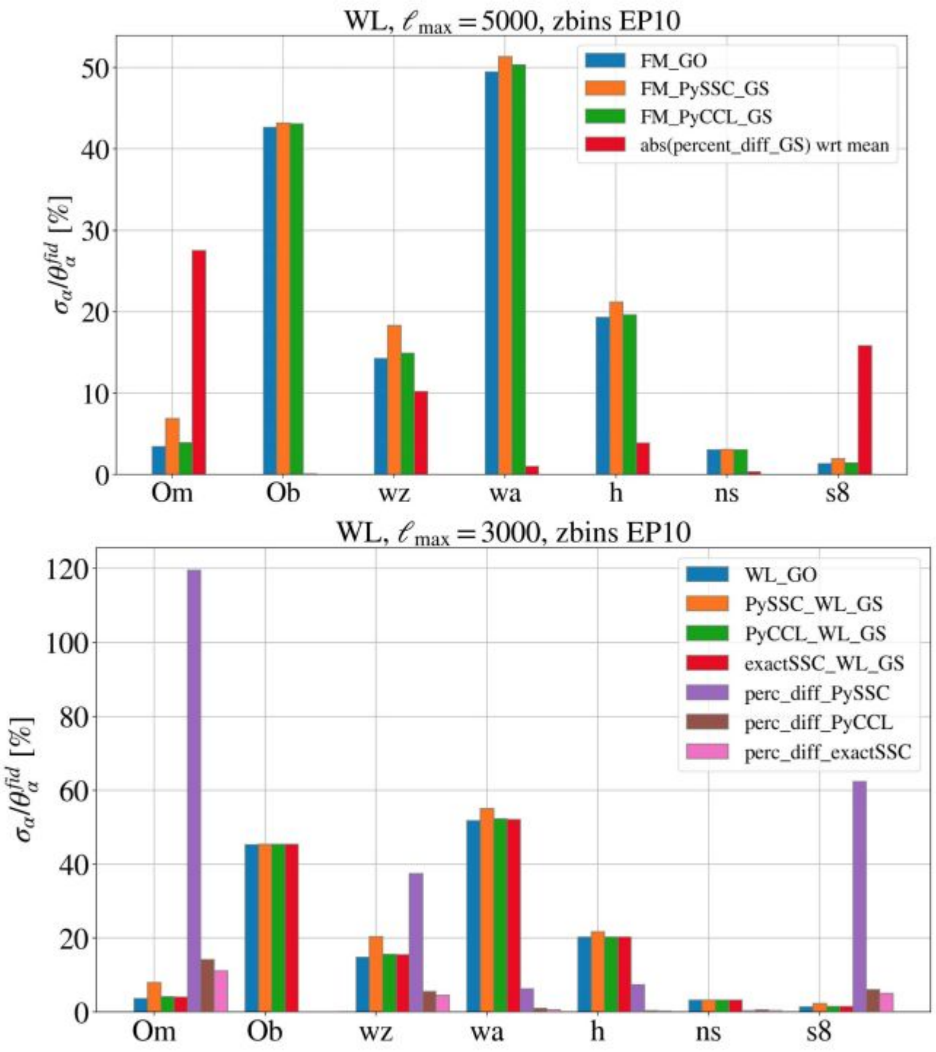




Davide Sciotti

Results WL

- (previously seen) mismatch between PySSC and other codes for most affected parameters
- Good match between PyCCL and exact SSC ✅
- Found variability of results with varying precision parameter in PySSC
- Impact is now too small?

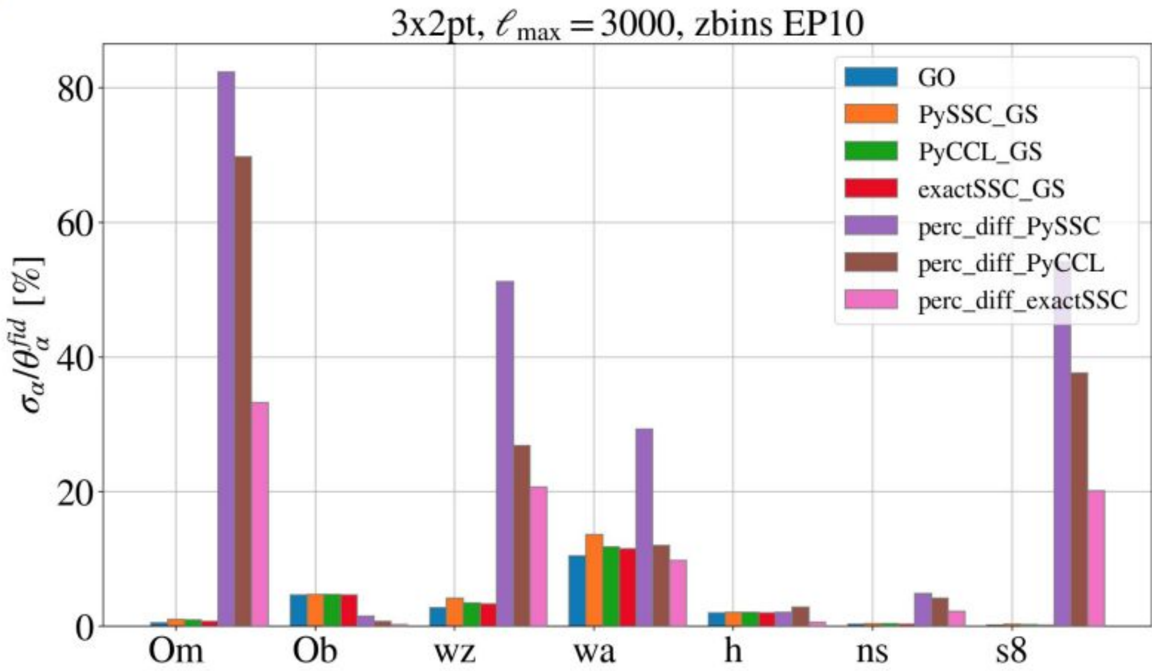


Davide Sciotti

Results 3x2pt

- Largest spread among the 3 codes: issues in the XC term?

	FoM degradation [%]
PySSC	49
PyCCL	41
Spaceborne	26



Davide Sciotti

Next Steps and Future prospects

- *Better understand PySSC WL mismatch and 3x2pt discrepancy*

- Finish building independent PyCCL pipeline

- **Finish implementing the cNG term**

- **Publish code on GitLab**

- Migrate to PyCCL v3

- *Improve integration routine in the Julia SSC integral (from trapz to simps), to reduce a bit number of steps*

- **Compare results against simulations!**

- **run MCMC with NL models in CLOE**