JEMU

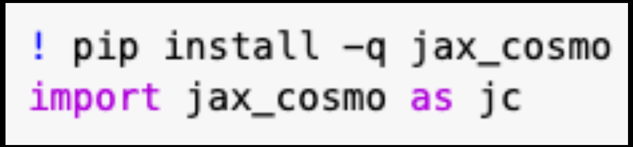
A BRIEF INTRODUCTION TO *ANOTHER* SIMPLE & NON-OFFICIAL P(K,Z) CLASS EMULATOR DESIGNED IN THE CONTEXT OF JAX-COSMO LIB.

J.E Campagne

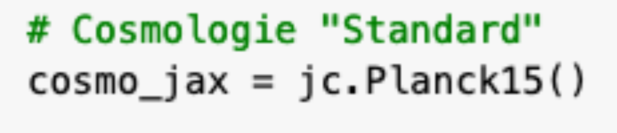
Tools Workshop @ GDR CoPhy / 5-7 June 2023

JAX-COSMO LIBRARY*

Cosmological library…

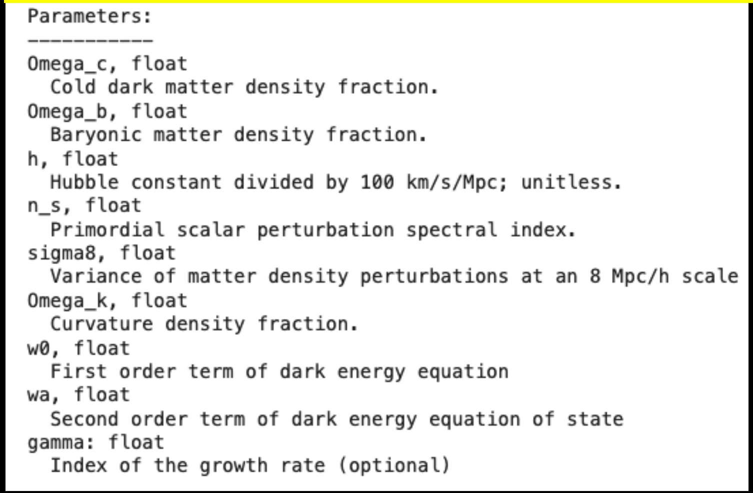


```
! pip install -q jax_cosmo
import jax_cosmo as jc
```

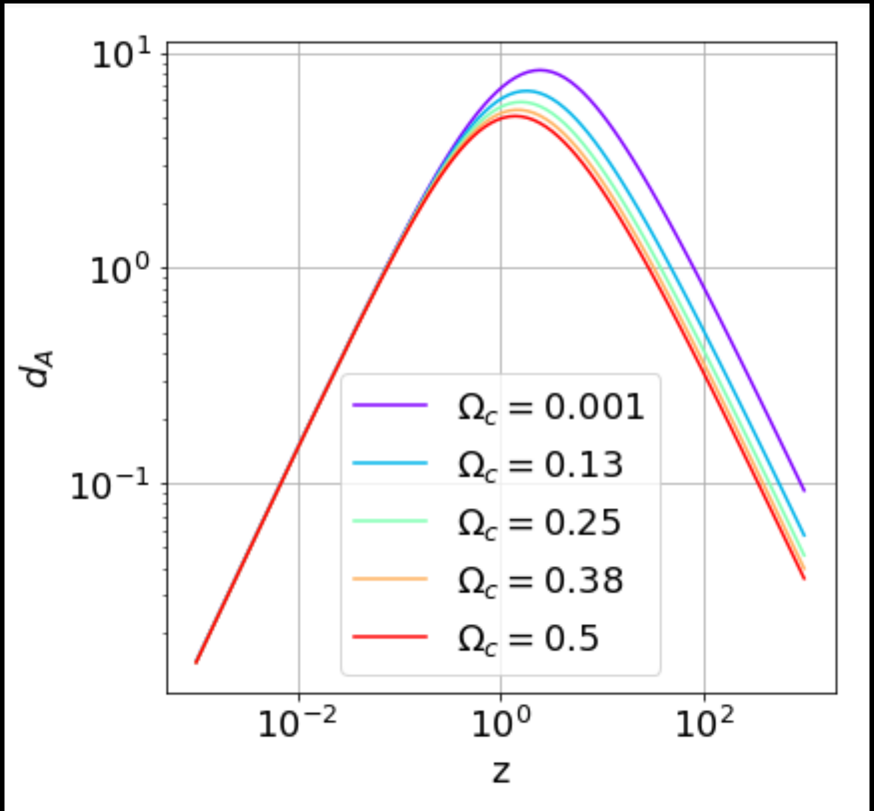


```
# Cosmologie "Standard"
cosmo_jax = jc.Planck15()
```

Cosmology as PyTree structure



```
Parameters:
----------
Omega_c, float
  Cold dark matter density fraction.
Omega_b, float
  Baryonic matter density fraction.
h, float
  Hubble constant divided by 100 km/s/Mpc; unitless.
n_s, float
  Primordial scalar perturbation spectral index.
sigma8, float
  Variance of matter density perturbations at an 8 Mpc/h scale
Omega_k, float
  Curvature density fraction.
w0, float
  First order term of dark energy equation
wa, float
  Second order term of dark energy equation of state
gamma: float
  Index of the growth rate (optional)
```



- ✓ automatic differentiation
- ✓ Just in time compilation
- ✓ Vectorization on single & multi devices
- ✓ Same code for CPU & GPU/TPU

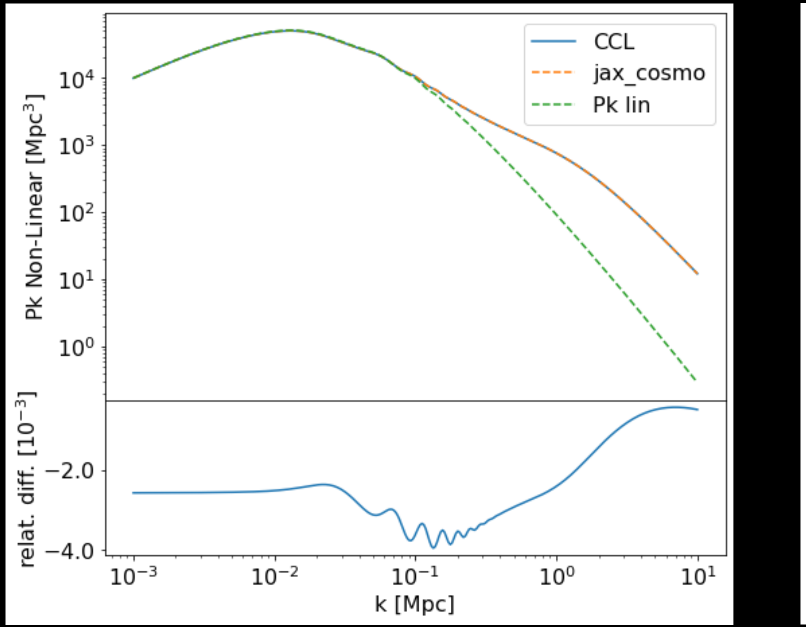
*) JEC, F. Lanusse, J. Zuntz et al arxiv:2302.05163 (2023), OJAp, 6, 15

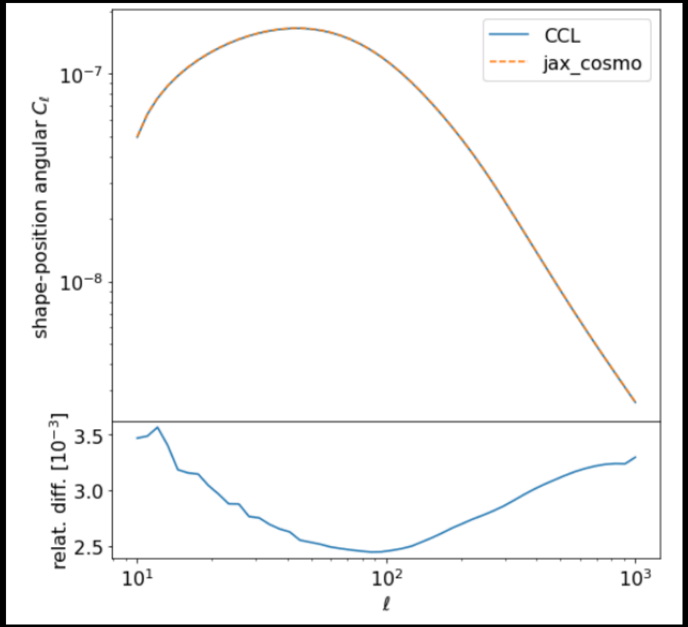
jax-cosmo @ github

JAX-COSMO LIBRARY

Currently: light implementation

- Follow CCL's tracer mechanism
- Unit tested against CCL
- P(k) (Eisenstein-Hu, Halofit),
- Cls (Lensing including NLA, multiplicative bias, number counts)
- Limber approximation





Idea: change the present P(k) by an emulator

CCL: Core Cosmology Library, https://ccl.readthedocs.io; Nora Elisa Chisari *et al* 2019 *ApJS* **242** 2

From original work of CLASS emulator : Mootoovaloo et al (2022), arXiv:2105.02256 A&C, 38, 100508

$$\tilde{D}(k,z) := \frac{P_{lin}(k,z)}{P_{lin}(k,0)} \qquad \tilde{Q}(k,z) := \frac{P_{nl}(k,z)}{P_{nl}(k,0)}$$

$$\boxed{y_{ij}^{X}}$$

Different from original schema

X = $\{\underbrace{P_{lin}(k,z=0), P_{nl}(k,z=0)}_{\text{RBF}}, \underbrace{\tilde{D}(k,z), \tilde{Q}(k,z)}_{\text{Matern12}}\}$

(ki,zj) log-lin grid of 120x20 pts

One GP with polynomial mean per (i,j)

θ : Cosmo param. 10^3 Training on a Latin Hypercube

Model hyper-param

$$y(\theta) = \Phi(\theta)\beta + \mathcal{GP}(0, K(a,\{\ell\}))$$

$\Phi(\theta) = \{1, \theta, \theta^2\}$

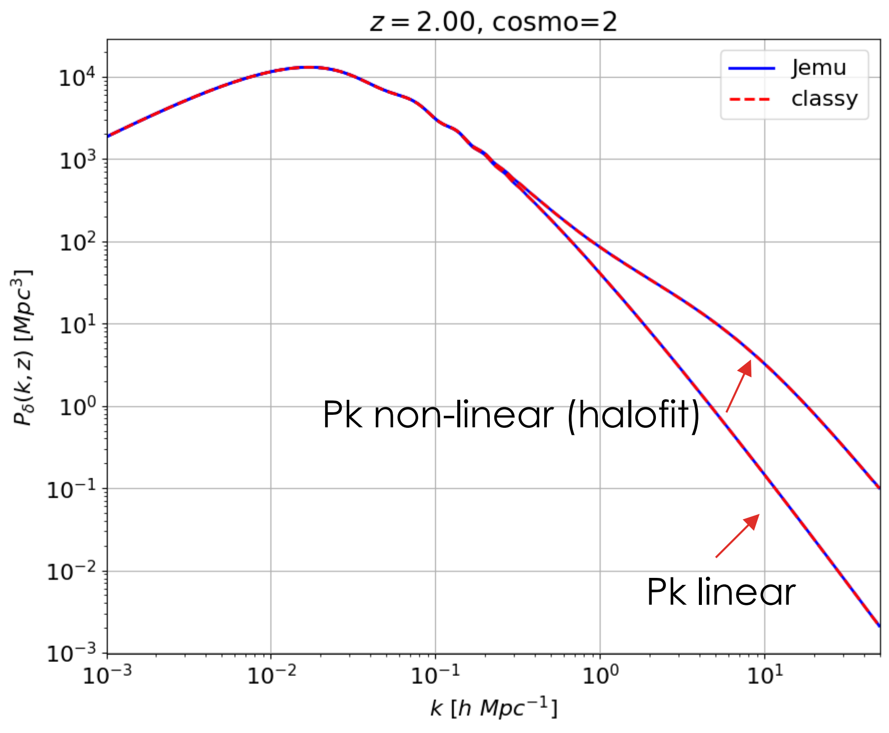
$$\boxed{\alpha_{ij}^{X} = (\beta, a, \{\ell\})_{ij}^{X}}$$

Max. of marginal likelihood with β gaussian prior (Scipy/L-BFGS-B)

$y_{ij}^{X}(\theta_*; \hat{\alpha}_{ij}^{X})$ Interpolation from he ij-grid

Autodiff & vectorization

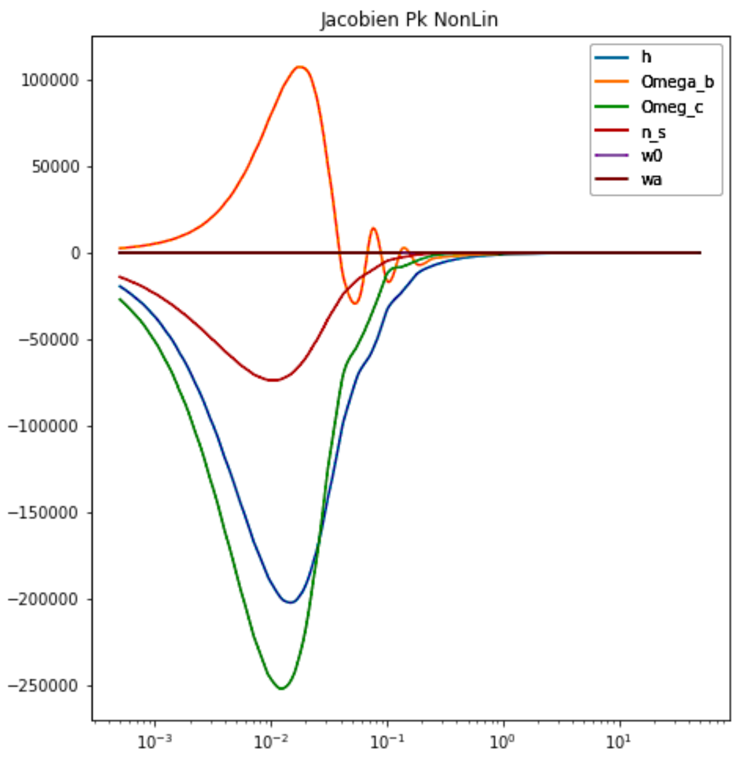
Interface with jax-cosmo Cosmology



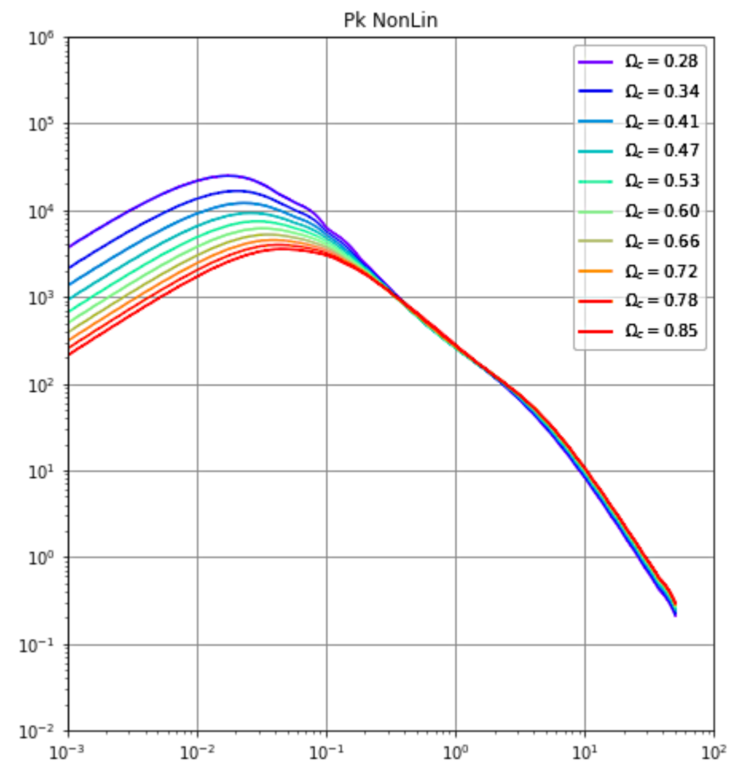
Pk non-linear (halofit)

Pk linear

Jacobian



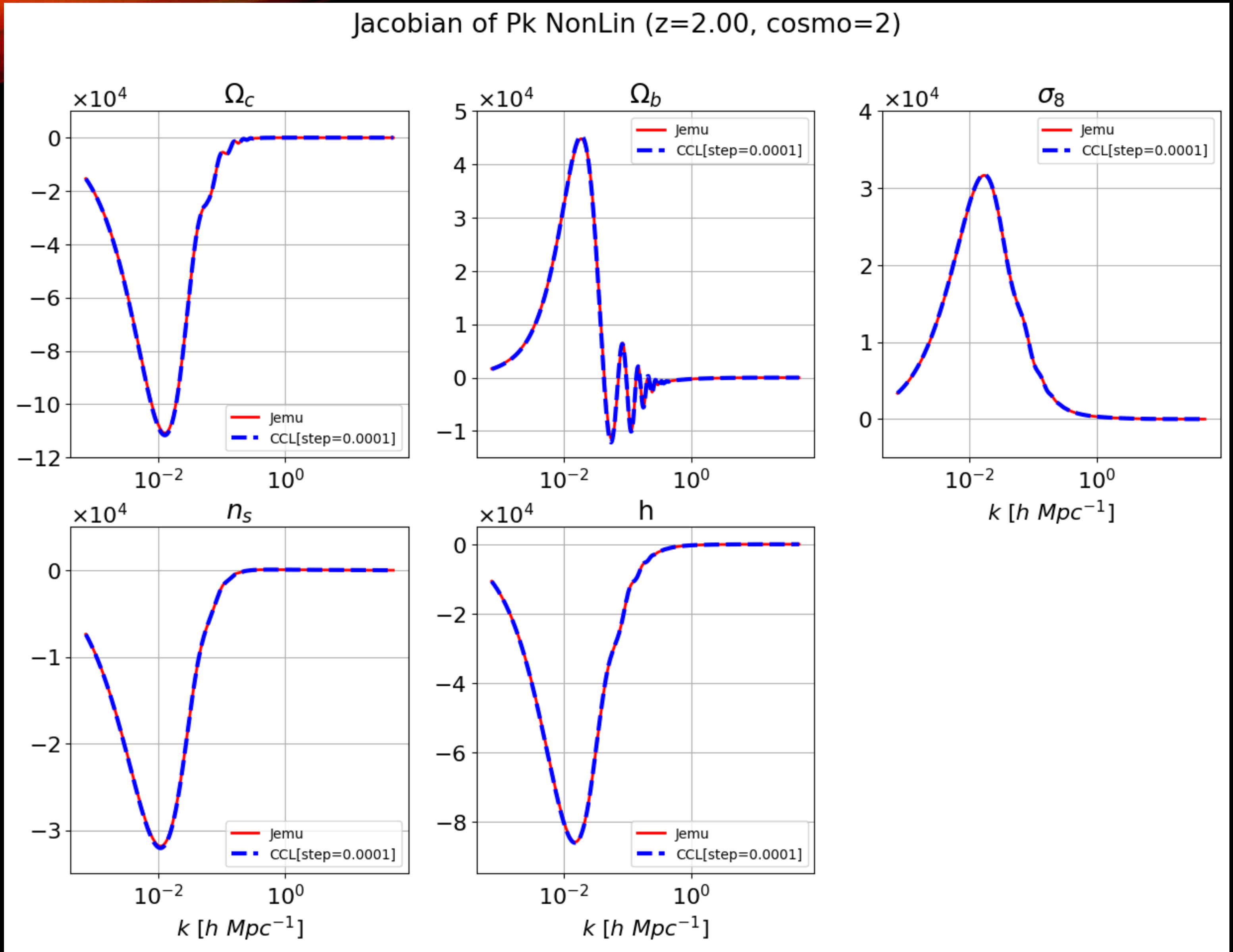
Vectorization (Ω_c)



@ O(1%) level the differences are not visible

CCL vs Jax-Cosmo with Jemu

h= 0.7
Ωc = 0.4
Ωb = 0.045
σ8 = 0.75
ns = 0.95



SUMMARY

✓Cosmo Parameters

- $\Omega_c\ h^2$: [0.06, 0.40]
- $\Omega_b\ h^2$: [0.019, 0.026]
- σ_8: [0.25, 3.25]
- n_s: [0.7, 1.3]
- h : [0.64, 0.82]

Here the settings used for test but it is user-case dependant

3 neutrino, Mtot=0.06eV, T_ncdm: 0.71611, N_ur: 0.00641

✓redshift in [0, $\lesssim$ 3], k in [5 10^{-4}, $\lesssim$ 20] h/Mpc

✓ $|\Delta Pk/Pk| \approx 1\%$ (nb. More the above ranges are squeezed more the accuracy gets better)

✓Coded in JAX with interface with jax-cosmo* (see CosmoPower-JAX** for similar aspects and use)

See the project for a JAX standardisation of CLASS emulator interface
Zelos @ DifferentiableUniverseInitiative

*: Jemu Github

**: D. Piras &A. Spurio Mancini arXiv:2305.06347