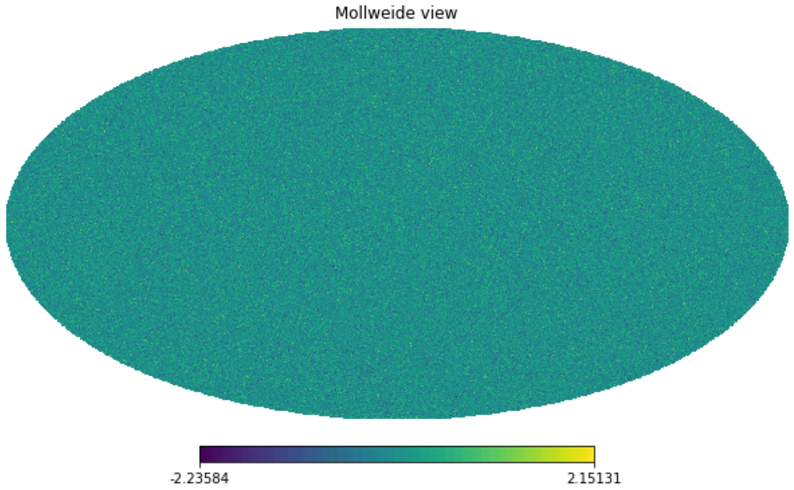
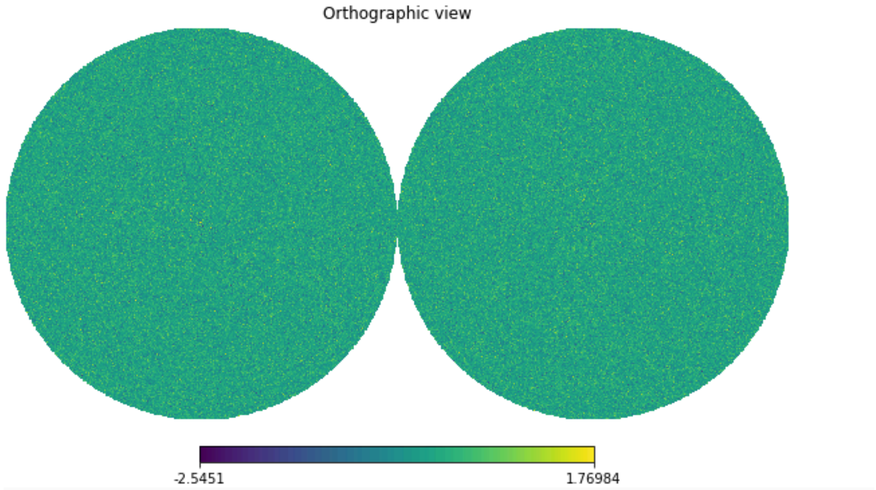
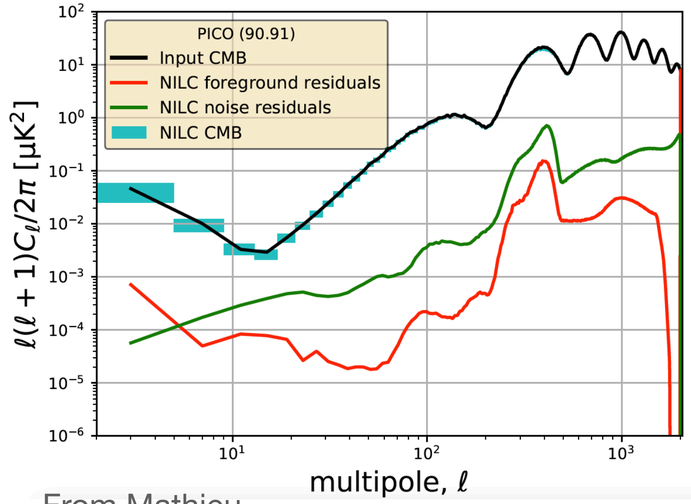
- Very first PICO lensing B-template reconstruction on Mathieu NILC 90.91 sim 001 (with our own E)

Julien C, Sebastian B, 13/01/22

INPUTS

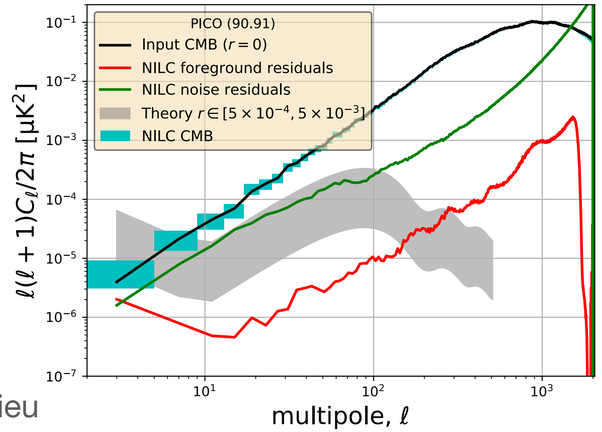






From Mathieu

/project/projectdirs/pico/reanalysis/nilc/ns2048/py91_ns2048_0001/NILC_PICO91_B_reso8acm.fits



From Mathieu

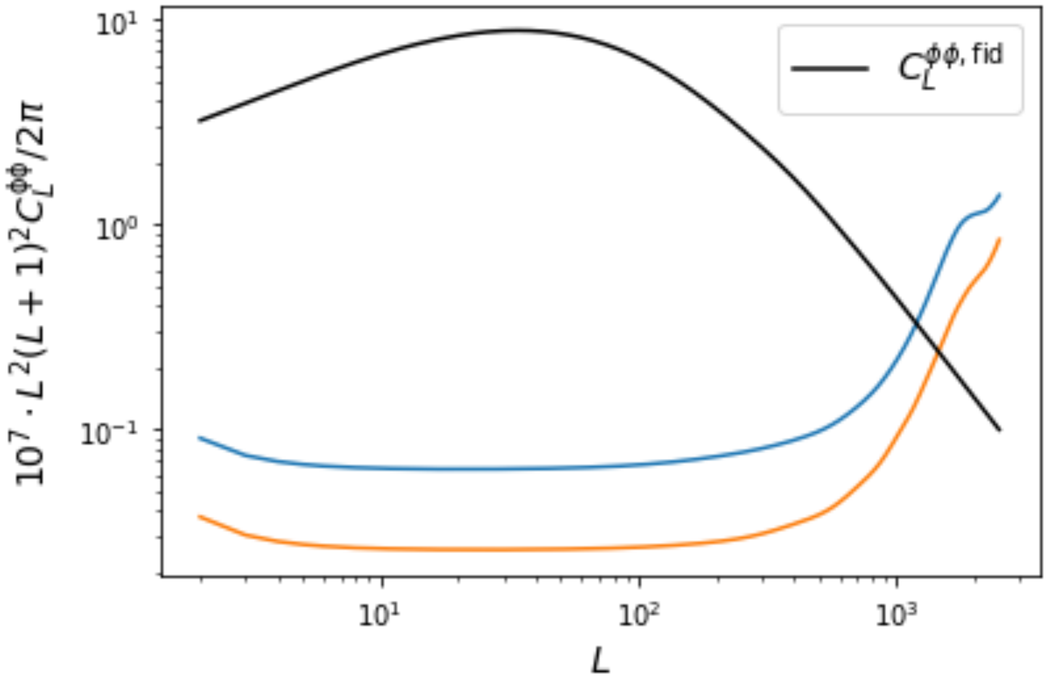
Forecast

```
removed B power in percent  64.2
0.64 0.64 0.64
removed B power in percent  77.7
0.78 0.77 0.78
```

QE

MAP

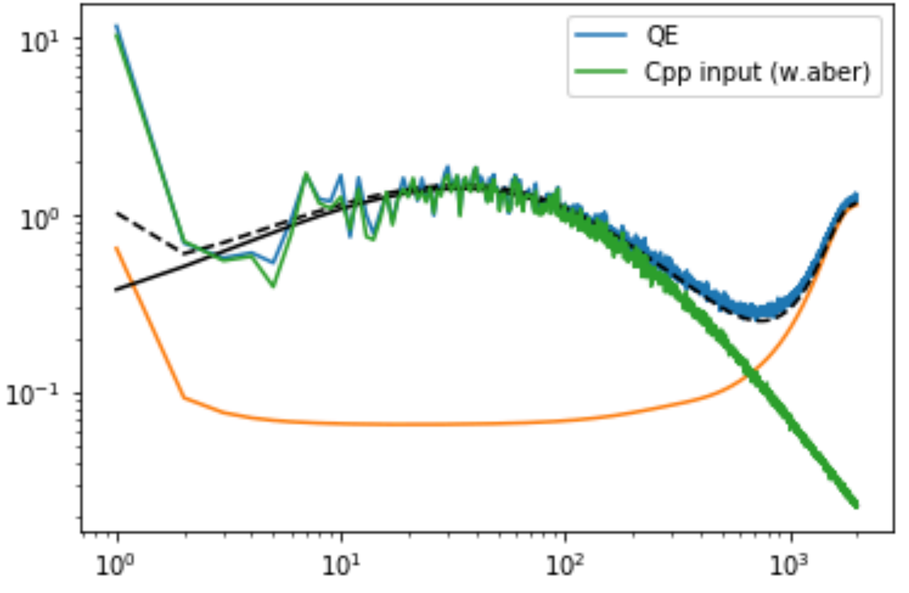
```
[151]: <matplotlib.legend.Legend at 0x2aab1d6ca0d0>
```



```
[14]: w = ls ** 2 * (ls + 1) ** 2  * 1e7 /2 /np.pi
      from plancklens.sims import planck2018_sims

      pl.loglog(ls,w *  hp.alm2cl(qlm)[ls]  / R[ls] ** 2, label='QE')
      pl.plot(ls, w / R[ls])
      pl.plot(ls, w * cls_unl['pp'][ls], c='k')
      pl.plot(ls, w * (cls_unl['pp'][ls] + 1/R[ls]), c='k', ls='--')
      pl.plot(ls, w * hp.alm2cl(planck2018_sims.cmb_unl_ffp10.get_sim_plm(1))[ls], label='Cpp input (w.aber)')
      pl.legend()
```

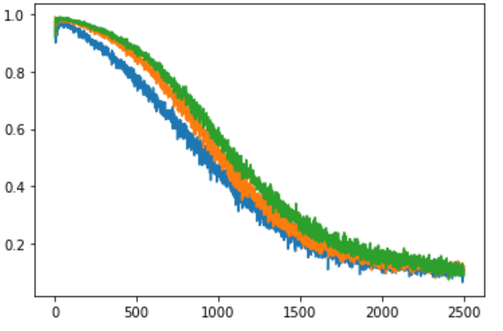
```
[14]: <matplotlib.legend.Legend at 0x2aab035d5550>
```



QE on NILC B-modes plus noisy E inputs

```
[15]: from lenscarf.iterators.statics import rec
      from plancklens.sims import planck2018_sims
      plms = rec.load_plms('/global/cscratch1/sd/jcarron/lenscarfrecs/PICO_idealized/p_p_sim0001/', [0, 1, 2])
      plm_in = utils.alm_copy(planck2018_sims.cmb_unl_ffp10.get_sim_plm(1), lmax=2500)
      ls = np.arange(2, 2500)
      for plm in plms:
          pl.figure('x')
          pl.plot(ls, hp.alm2cl(plm, plm_in)[ls] / np.sqrt(hp.alm2cl(plm_in)[ls] * hp.alm2cl(plm)[ls]))
          pl.figure('a')
          pl.loglog(ls,(2 * ls + 1) * ls ** 2 *(ls + 1) ** 2 * hp.alm2cl(plm)[ls])
      pl.loglog(ls,(2 * ls + 1) * ls ** 2 *(ls + 1) ** 2 * cls_unl['pp'][ls])
```

```
[15]: [<matplotlib.lines.Line2D at 0x2aab03aa0590>]
```

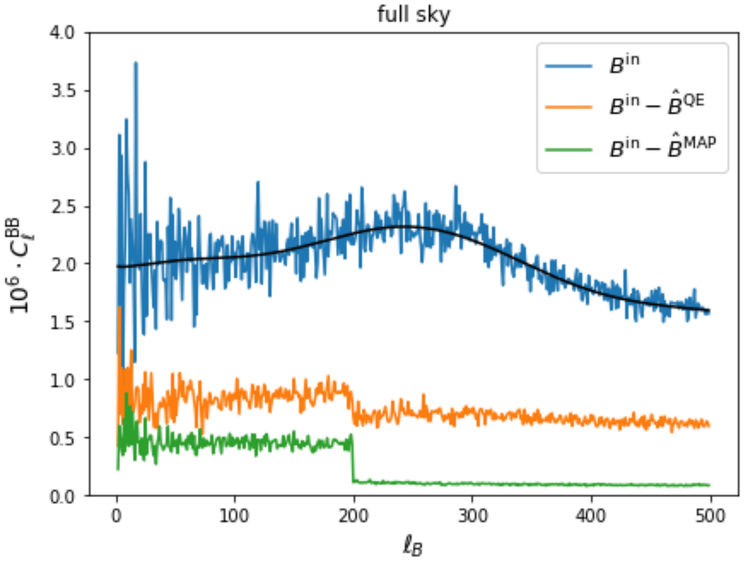


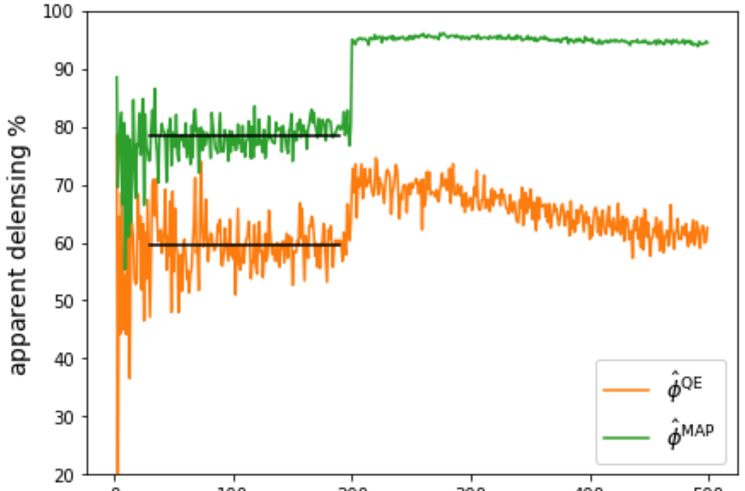
Cross-correlation to input lensing

Results

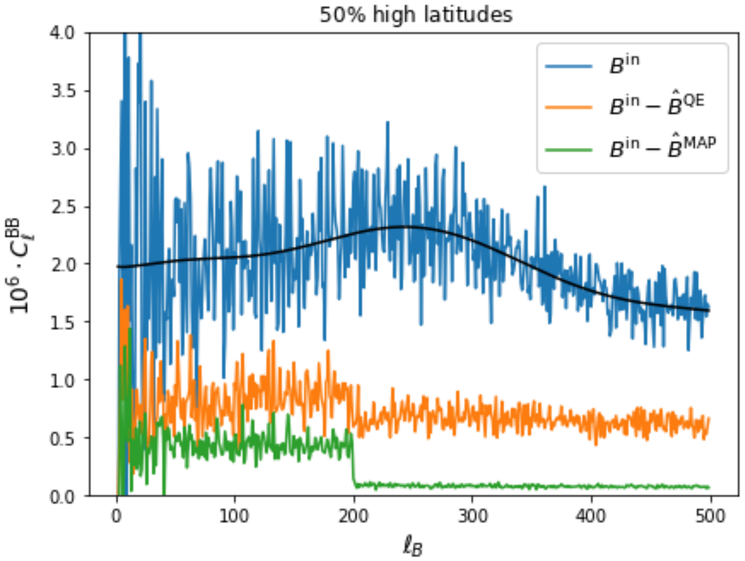
-v hbump -imin 1 -imax 1 -itmax 10

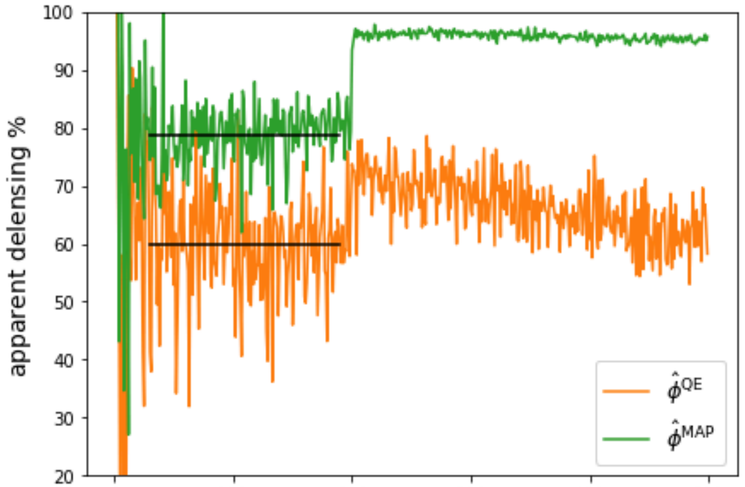
```
fsky 1.00
fsky 1.00 app del QE 59.5
fsky 1.00 app del MAP (itr 10) 78.3
```

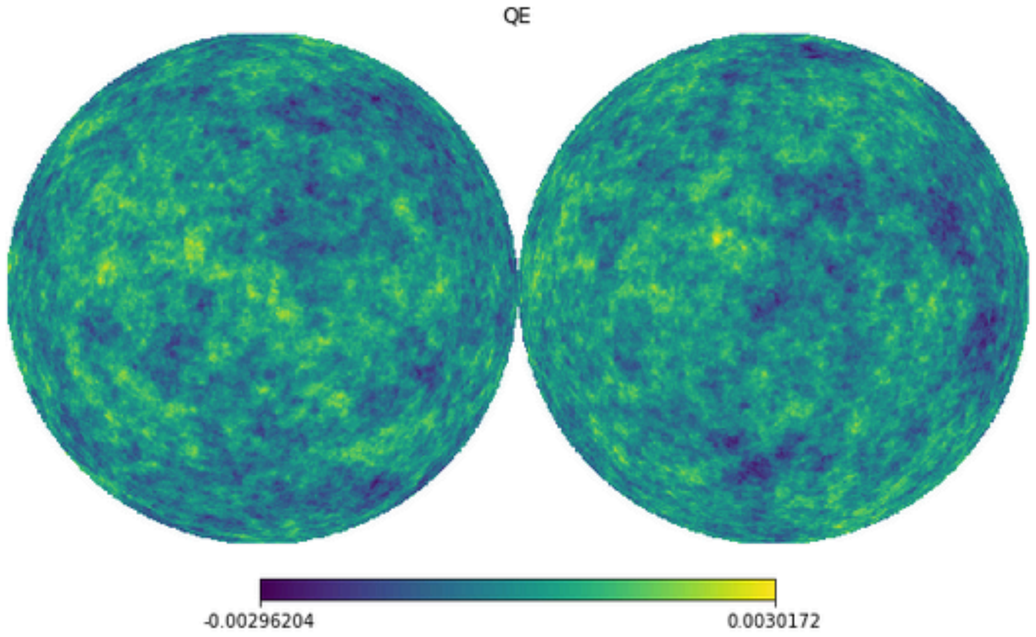


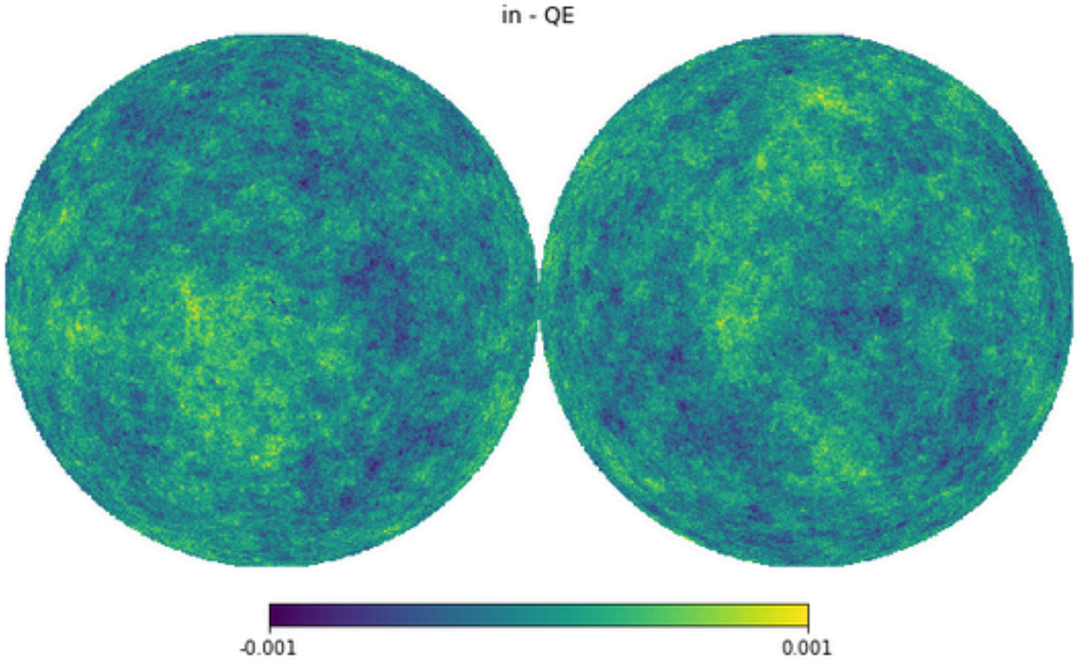


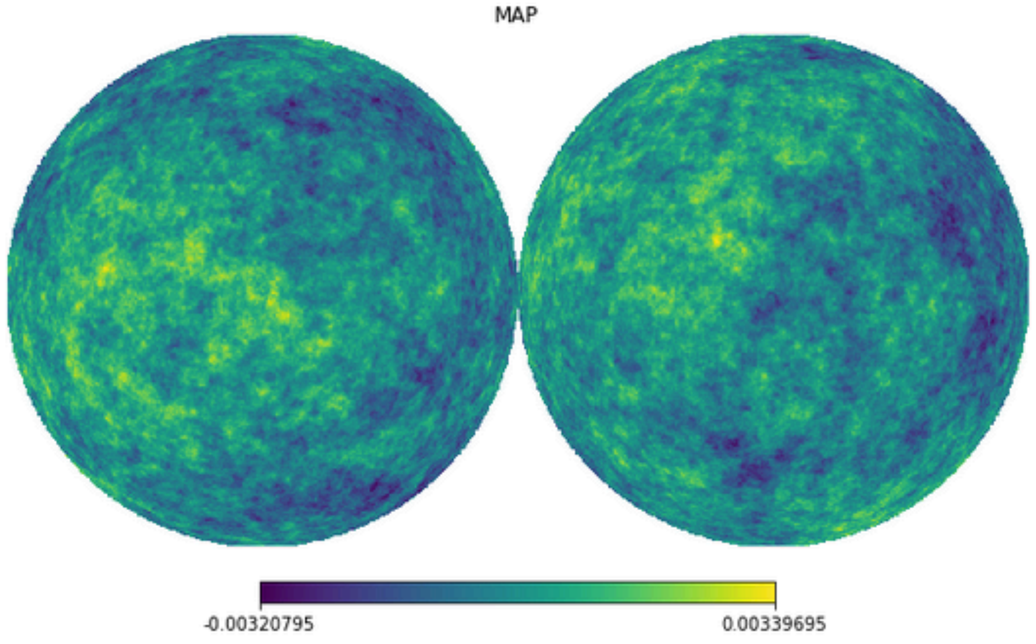

```
fsky 0.44
fsky 0.44 app del QE 59.8
fsky 0.44 app del MAP (itr 10) 78.7
```

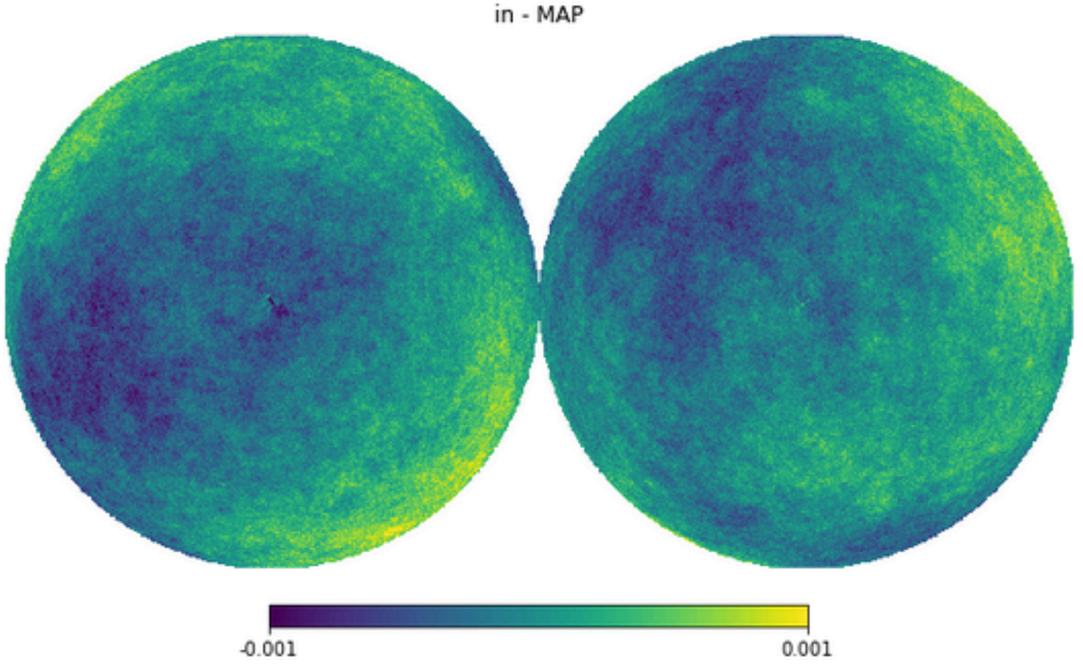





QE
-0.00296204
0.0030172

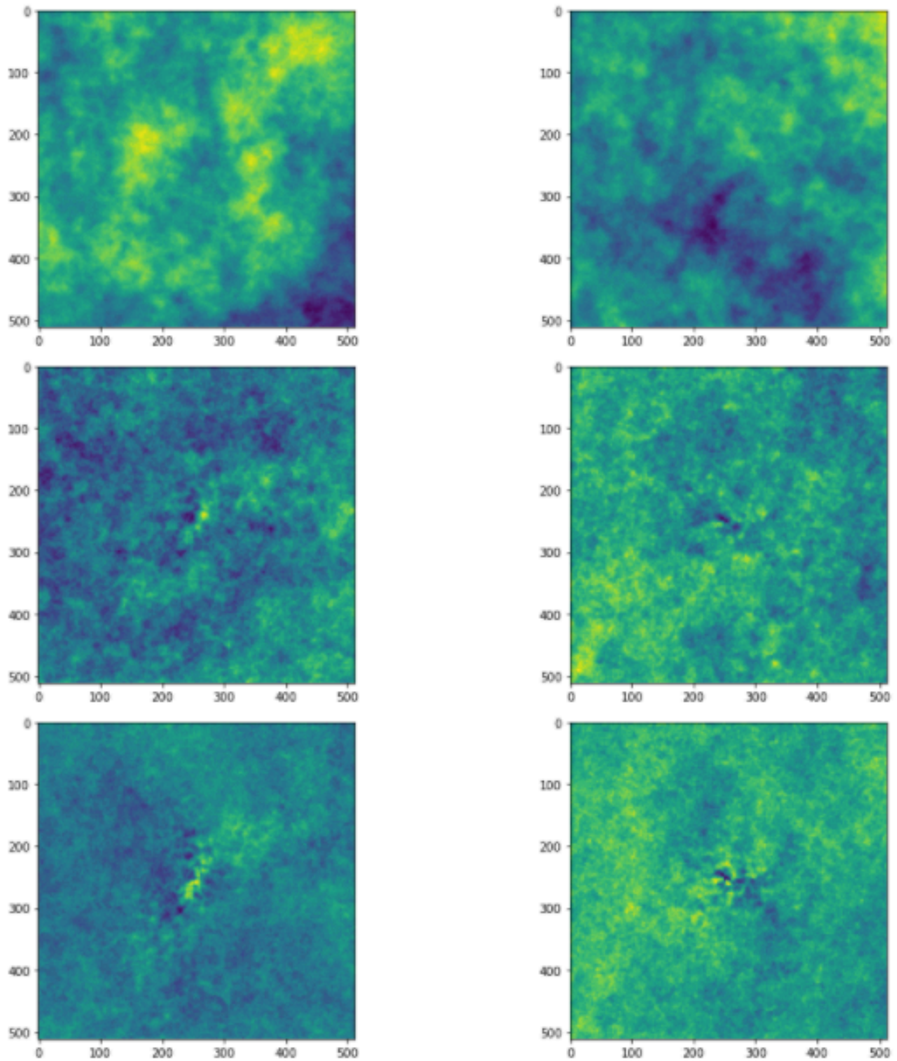

in - QE
-0.001
0.001


MAP
-0.00320795
0.00339695


in - MAP
-0.001
0.001

[174]:
```
N = 512
fig, axes = pl.subplots(1, 2, figsize=(15, 5))
axes[0].imshow(proj(d_i, 1.7, N, lat=-90.)[0])
axes[1].imshow(proj(d_i, 1.7, N, lat=+90.)[0])
fig, axes = pl.subplots(1, 2, figsize=(15, 5))
axes[0].imshow(proj(QE - d_i, 1.7, N, lat=-90.)[0])
axes[1].imshow(proj(QE - d_i, 1.7, N, lat=+90.)[0])
fig, axes = pl.subplots(1, 2, figsize=(15, 5))
axes[0].imshow(proj(MAP - d_i, 1.7, N, lat=-90.)[0])
axes[1].imshow(proj(MAP - d_i, 1.7, N, lat=+90.)[0])
```

[174]:
```
<matplotlib.image.AxesImage at 0x2aab22237cd0>
```



Summary:

- Very first reconstruction on Mathieu NILC 90.91 (with our own E), on the full sky seems to work fine so far
- Forecast and achieved delensing efficiency of ~78%, using $10 < l_E < 2000$, $200 < l_B < 2000$