High precision pixel-based simulations of CMB lensing

Giulio Fabbian

Laboratoire APC AstroParticule et Cosmologie
University of Paris VII Denis Diderot

22nd April 2011

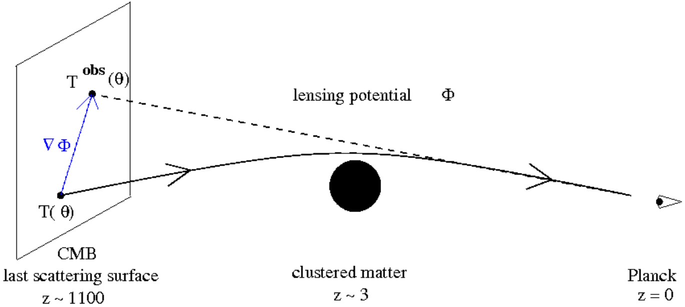
Berkeley CMB lensing workshop

Outline

How to simulate CMB lensing

- An observed CMB field today is the primordial signal coming from another direction

$$\tilde{X}(\hat{\mathbf{n}}) = X(\hat{\mathbf{n}}') \qquad \hat{\mathbf{n}}' = \hat{\mathbf{n}} + \nabla\psi$$



- Series expansion is not accurate and slowly converging (Lewis 2005)
- Pixel based method: remapping points as function of position
- Displacement computed in Born approximation
- High resolution required!

How to simulate CMB lensing? II

1. CMB: $C_\ell^T\, C_\ell^E, C_\ell^B \rightarrow a_{lm}^X$
2. $C_\ell^\psi \rightarrow \psi_{\ell m} \rightarrow \vec{d}$

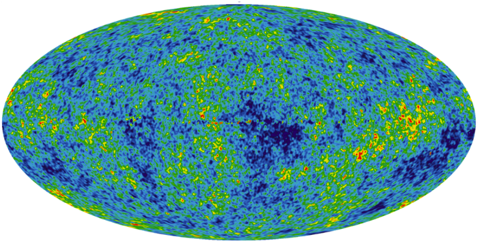
   $${}_1E_{lm}^d = \sqrt{l(l+1)}\psi_{lm} \quad {}_1B_{lm}^d = 0$$

3. $\hat{\mathbf{n}} = (\vartheta, \varphi) \rightarrow \hat{\mathbf{n}}' = (\vartheta', \phi + \Delta\phi)$

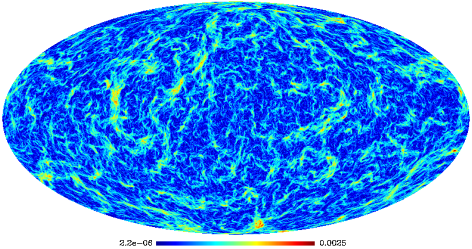
   $$\cos\vartheta' = \cos d \cos\vartheta - \sin d \sin\vartheta \cos\alpha$$

   $$\sin\Delta\varphi = \frac{\sin\alpha \sin d}{\sin\vartheta'}$$

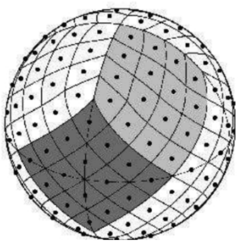
4. Resampling T, P fields at displaced position
5. Smoothing (and repixelization) according to experiment requirements



Realization of the displacement field

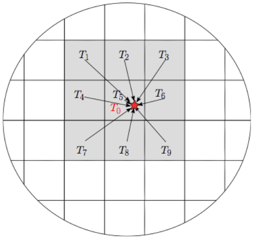


Problems and proposed solutions



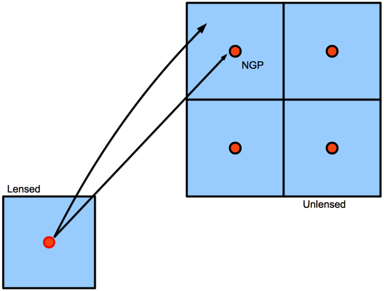
- Direct resummation
- Bicubic interpolation (LENSPix)
- Recasting on 2-D torus + NFFT (Basak et al. 2009)
- Statistical interpolation using spectral information (Lavaux et al. 2010)

- Step 1-3 can be performed with fast SHT
- Step 4 is the problem: we have to be clever
- Lensing changes band width properties
- Does grid properly sample lensed sky?



An alternative approach

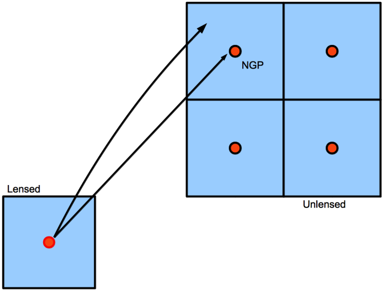
- Is interpolation completely safe?
- How to keep workload constant?
- Alternative approach: Nearest Grid Point
- **Pros**: Fast, as cheap as FSHT
- **Cons**: Calls for very dense sampling



- Parallel HPSC are quickly ubiquitous
- Massively parallel solutions are more and more affordable
- We need a superior SHT algorithm

An alternative approach

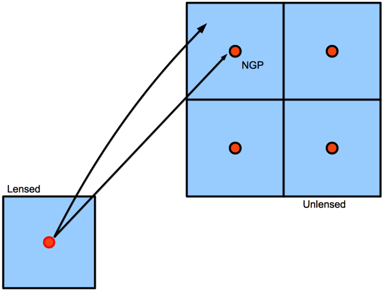
- Is interpolation completely safe?
- How to keep workload constant?
- Alternative approach: Nearest Grid Point
- **Pros**: Fast, as cheap as FSHT
- **Cons**: Calls for very dense sampling



- Parallel HPSC are quickly ubiquitous
- Massively parallel solutions are more and more affordable
- We need a superior SHT algorithm

An alternative approach

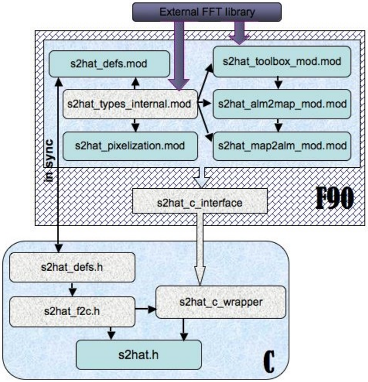
- Is interpolation completely safe?
- How to keep workload constant?
- Alternative approach: Nearest Grid Point
- **Pros**: Fast, as cheap as FSHT
- **Cons**: Calls for very dense sampling



- Parallel HPSC are quickly ubiquitous
- Massively parallel solutions are more and more affordable
- We need a superior SHT algorithm

LenS^{2}HAT : Lensing with S^{2}HAT

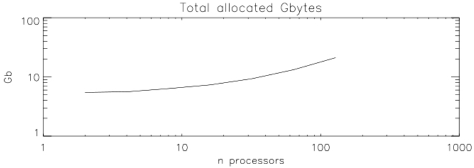
- Distributing both pixel and harmonic domain objects
- S^{2}HAT library: Fourier analysis for spin fields on the sphere

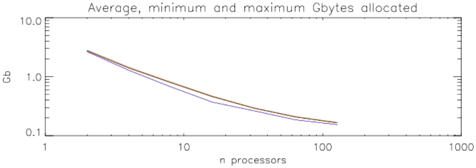


- Designed for parallel architectures
- SPRNG random numbers
- MPI Standard
- Arbitrary pixelization scheme
- Memory requirements $\mathcal{O}(10N_{pix}/n_{proc})$
- Optimized remapping
- T, TQU, QU simulations
- Sub dominant remapping cost
- Nearly perfect scaling

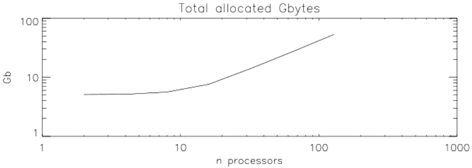
LensPix Vs LenS^{2}HAT

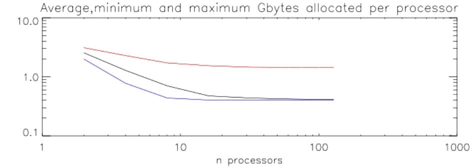
- LenS^{2}HAT memory requirements ($nside = 2048, \ell_{max} = 2048$)





- LensPix memory requirements ($nside = 2048, \ell_{max} = 2048$)





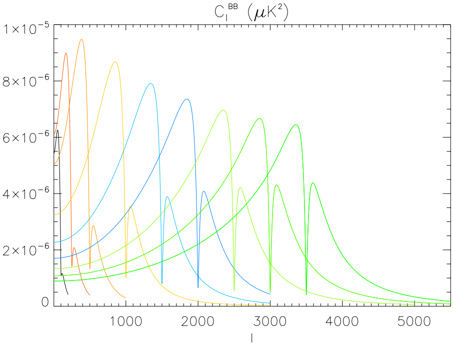
Trade-off: kernels

- Do we correctly compute lensing in the pixel domain?
- The B mode case
- $C_{\ell_1}^{BB_{lens}} = \frac{1}{2\ell_1+1} \sum_{\ell_2 \ell_3}^{\ell_{max}} |f_{\ell_1 \ell_2 \ell_3}^{EB}|^2 C_{\ell_2}^{EE} C_{\ell_3}^{\phi\phi}$
- $f_{\ell_1 \ell_2 \ell_3}^{EB} = \frac{F_{\ell_1 \ell_2 \ell}^{-2} - F_{\ell_1 \ell_2 \ell_3}^{2}}{2i}$
- $F_{\ell_1 \ell_2 \ell_3}^{s} = [-\ell_1(\ell_1+1)+\ell_2(\ell_2+1)+\ell_3(\ell_3+1)] \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}{16\pi}} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -s & s & 0 \end{pmatrix}$
- $C_{\ell_2}^{EE} = \delta_{\ell \ell_2}$
- $C_{\ell_2}^{EE} = \delta_{\ell \ell_2} C_{\ell}^{EE}$
- How LenS^{2}HAT compare?

Trade-off: kernels

- Do we correctly compute lensing in the pixel domain?
- The B mode case

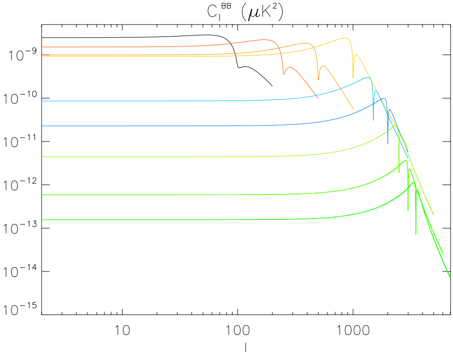
- $C_{\ell_1}^{BB_{lens}} = \frac{1}{2\ell_1+1} \sum_{\ell_2\ell_3}^{\ell_{max}} |f_{\ell_1\ell_2\ell_3}^{EB}|^2 C_{\ell_2}^{EE} C_{\ell_3}^{\phi\phi}$
- $f_{\ell_1\ell_2\ell_3}^{EB} = \frac{F_{\ell_1\ell_2\ell}^{-2} - F_{\ell_1\ell_2\ell_3}^{2}}{2i}$
- $F_{\ell_1\ell_2\ell_3}^{s} = [-\ell_1(\ell_1+1)+\ell_2(\ell_2+1)+\ell_3(\ell_3+1)] \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}{16\pi}} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -s & s & 0 \end{pmatrix}$
- $C_{\ell_2}^{EE} = \delta_{\ell\ell_2}$
- $C_{\ell_2}^{EE} = \delta_{\ell\ell_2} C_{\ell}^{EE}$
- How LenS^{2}HAT compare?



Trade-off: kernels

- Do we correctly compute lensing in the pixel domain?
- The B mode case

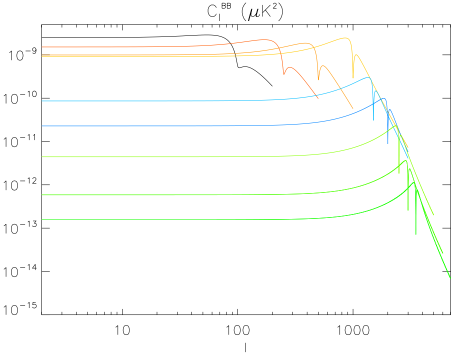
- $C_{\ell_1}^{BB_{lens}} = \frac{1}{2\ell_1+1} \sum_{\ell_2\ell_3}^{\ell_{max}} |f_{\ell_1\ell_2\ell_3}^{EB}|^2 C_{\ell_2}^{EE} C_{\ell_3}^{\phi\phi}$
- $f_{\ell_1\ell_2\ell_3}^{EB} = \frac{F_{\ell_1\ell_2\ell}^{-2} - F_{\ell_1\ell_2\ell_3}^{2}}{2i}$
- $F_{\ell_1\ell_2\ell_3}^{s} = [-\ell_1(\ell_1+1)+\ell_2(\ell_2+1)+\ell_3(\ell_3+1)] \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}{16\pi}} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -s & s & 0 \end{pmatrix}$
- $C_{\ell_2}^{EE} = \delta_{\ell\ell_2}$
- $C_{\ell_2}^{EE} = \delta_{\ell\ell_2} C_{\ell}^{EE}$
- How LenS^{2}HAT compare?



Trade-off: kernels

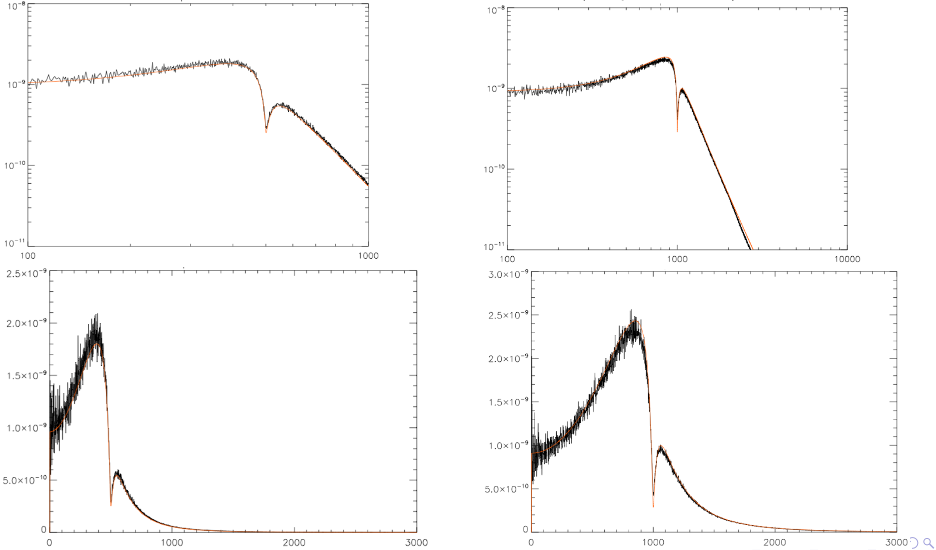
- Do we correctly compute lensing in the pixel domain?
- The B mode case

- $C_{\ell_1}^{BB_{lens}} = \frac{1}{2\ell_1+1} \sum_{\ell_2 \ell_3}^{\ell_{max}} |f_{\ell_1\ell_2\ell_3}^{EB}|^2 C_{\ell_2}^{EE} C_{\ell_3}^{\phi\phi}$
- $f_{\ell_1\ell_2\ell_3}^{EB} = \frac{F_{\ell_1\ell_2\ell}^{-2} - F_{\ell_1\ell_2\ell_3}^{2}}{2i}$
- $F_{\ell_1\ell_2\ell_3}^{s} = [-\ell_1(\ell_1+1)+\ell_2(\ell_2+1)+\ell_3(\ell_3+1)] \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}{16\pi}} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ -s & s & 0 \end{pmatrix}$
- $C_{\ell_2}^{EE} = \delta_{\ell\ell_2}$
- $C_{\ell_2}^{EE} = \delta_{\ell\ell_2} C_{\ell}^{EE}$
- How LenS^{2}HAT compare?

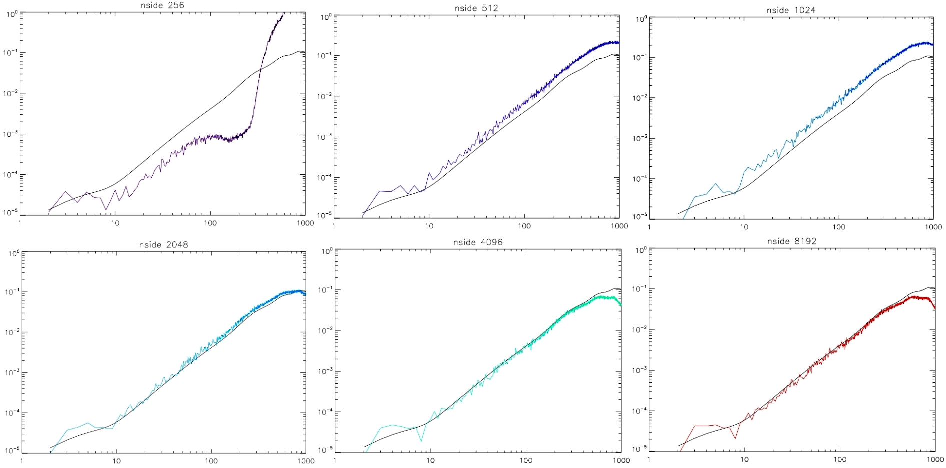


LenS^{2}HAT kernels

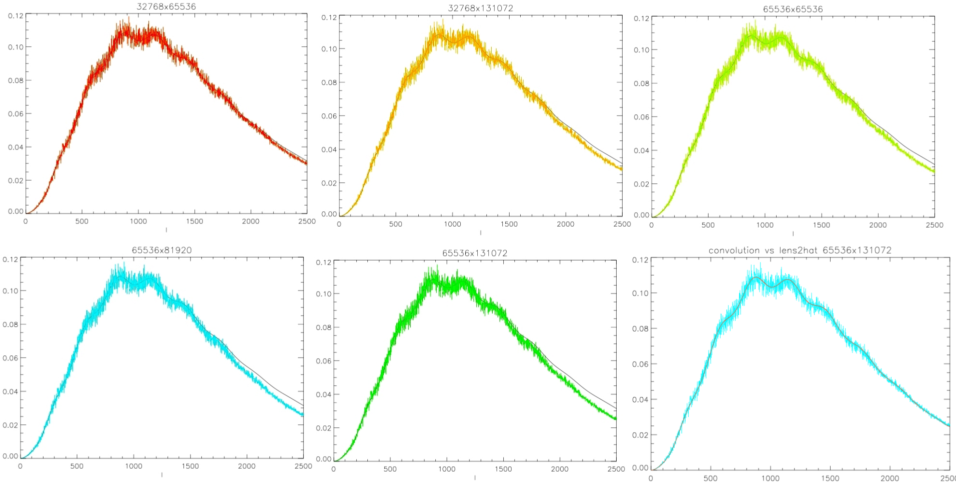
$\ell = 512, \ell_{max} = 3000, nside = 8192$ $\ell = 1000, \ell_{max} = 3000, nside = 16384$



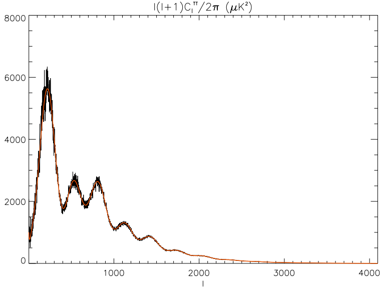
1-D convergence: HEALPix grid $\ell_{max} = 1000$

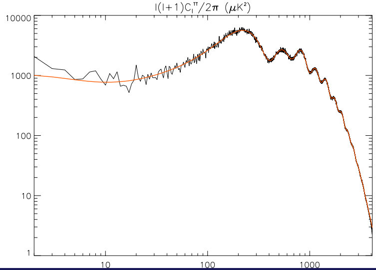


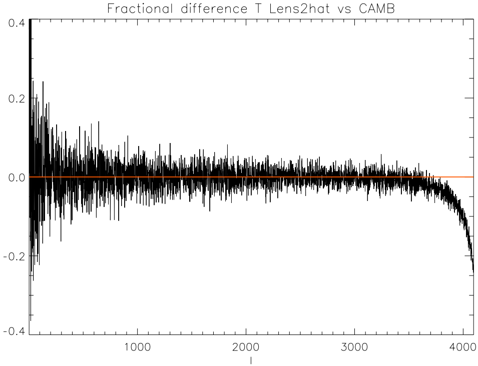
1-D convergence II: ECP grid $\ell_{max} = 2500$



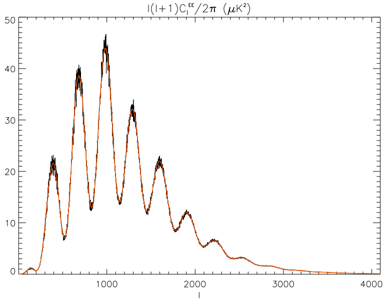
Full sky LenS^{2}HAT simulations

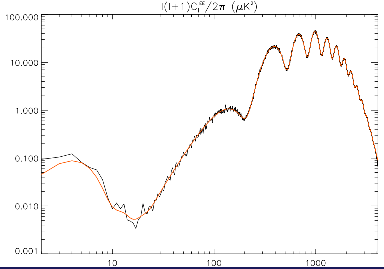


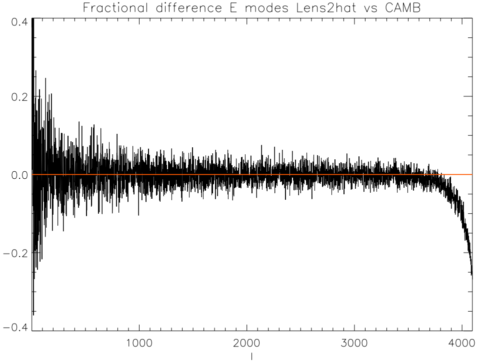




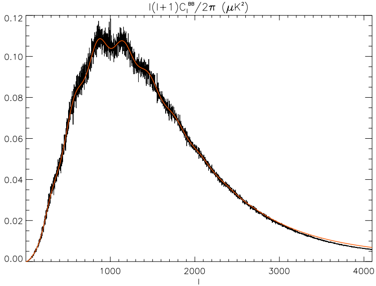
Full sky LenS^{2}HAT simulations

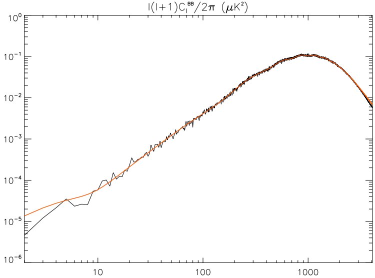


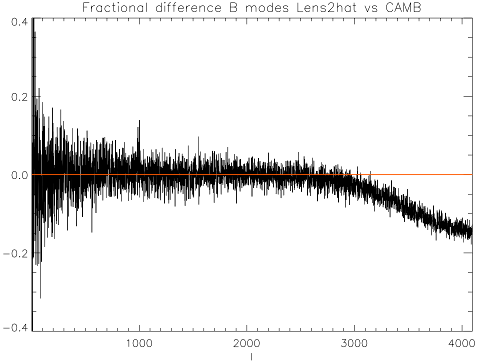




Full sky LenS^{2}HAT simulations

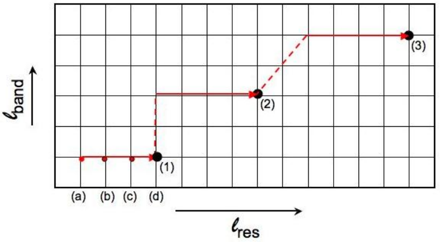






Conclusions

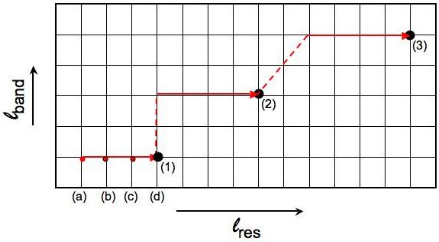
- NGP method
- High performance code LenS^{2}HAT
- Grid resolution and ℓ_{max} affect the B modes accuracy
- Validation of the method
- Useful for MC

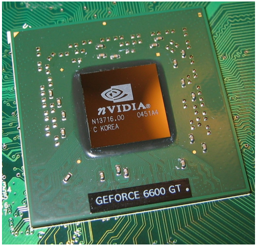


- Pixelization effect and high ℓ behavior ongoing
- Perfect quadrature with ECP
- SHT is the most expensive part of the method
- Towards exact method: lensing with GPUs?

Conclusions

- NGP method
- High performance code LenS^{2}HAT
- Grid resolution and ℓ_{max} affect the B modes accuracy
- Validation of the method
- Useful for MC





- Pixelization effect and high ℓ behavior ongoing
- Perfect quadrature with ECP
- SHT is the most expensive part of the method
- Towards exact method: lensing with GPUs?