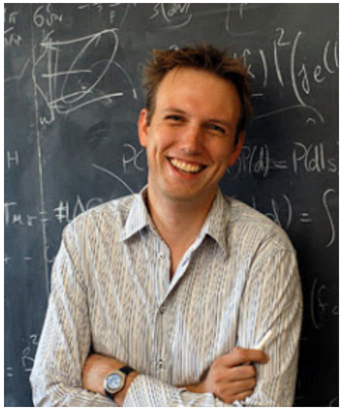
An optimal Bayesian solution to the CMB delensing problem

Marius Millea

Institut Lagrange de Paris

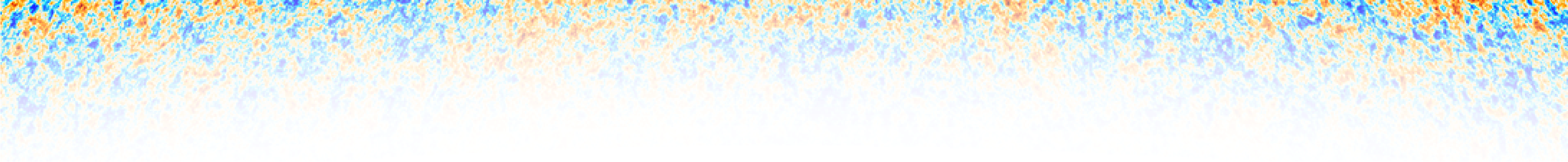
with

Ethan Anderes



Ben Wandelt

Nordita July 21, 2017



How do we optimally delense future CMB data to obtain the best possible estimates of r?

CMB “fields”

$f \equiv (T, Q, U)$

Lensing potential

Cosmo params

Data

$$\boxed{\mathcal{P}(f, \phi, r \,|\, d)}$$

Lensing potential

CMB “fields”

$$f \equiv (T, Q, U)$$

Cosmo params

Data

$$\boxed{\mathcal{P}(f, \phi, r \,|\, d)}$$

Lensing potential

Cosmo params

CMB “fields”

$$f \equiv (T, Q, U)$$

Data

$$\boxed{\mathcal{P}(f, \phi, r \,|\, d)}$$

$$\boxed{\hat{\phi}(\mathbf{L}) = \int d\mathbf{l_1} W(\mathbf{l_1}, \mathbf{l_2}) d(\mathbf{l_1})^* d(\mathbf{l_2})}$$

All current analyses are based on this
Currently near-optimal but will be sub-optimal for next-gen noise levels

CMB “fields”

$$f \equiv (T, Q, U)$$

Lensing potential

Cosmo params

Data

$$\boxed{\mathcal{P}(f, \phi, r \,|\, d)}$$

$$\boxed{\mathcal{P}(\phi \,|\, r, d)}$$

$$= \int df \mathcal{P}(f, \phi \,|\, r, d)$$

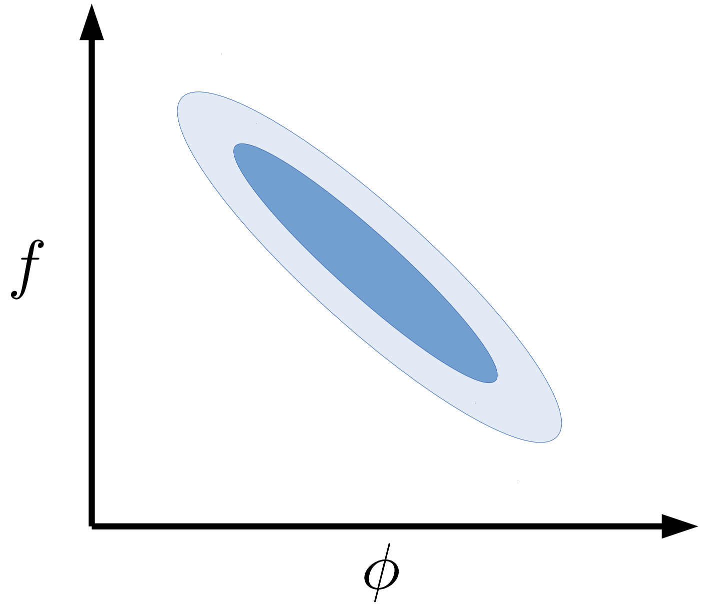
Carron & Lewis (2017), Hirata & Seljak (2003) give algorithm to *maximize* this

$$\boxed{\hat{\phi}(\mathbf{L}) = \int d\mathbf{l_1} W(\mathbf{l_1}, \mathbf{l_2}) d(\mathbf{l_1})^* d(\mathbf{l_2})}$$

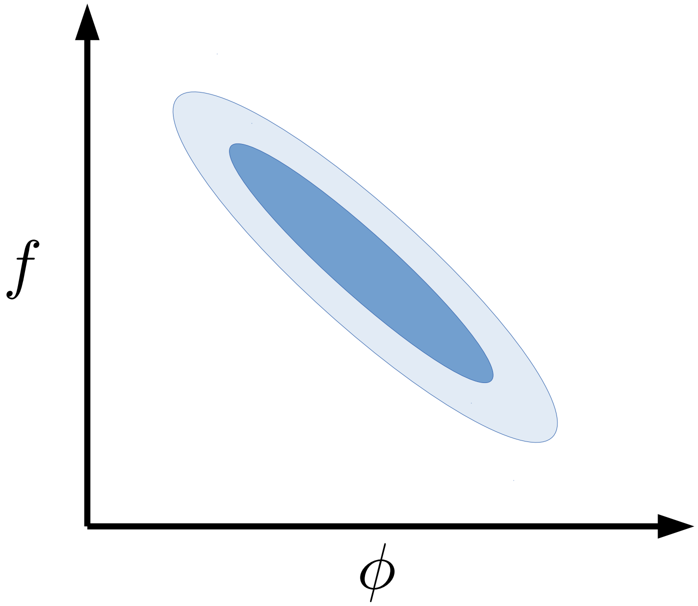
All current analyses are based on this
Currently near-optimal but will be sub-optimal for next-gen noise levels

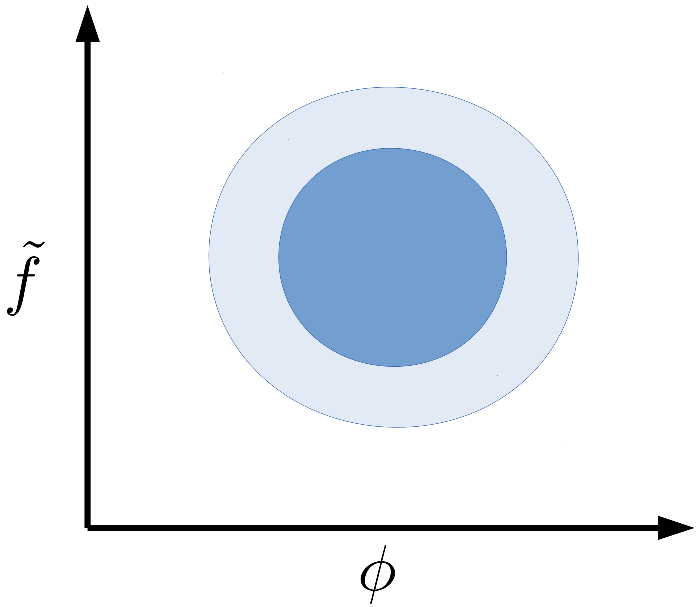
Why is sampling/minimizing $\mathcal{P}(f, \phi \,|\, d)$ such a hard problem?

Why is sampling/minimizing $\mathcal{P}(f, \phi \,|\, d)$ such a hard problem?

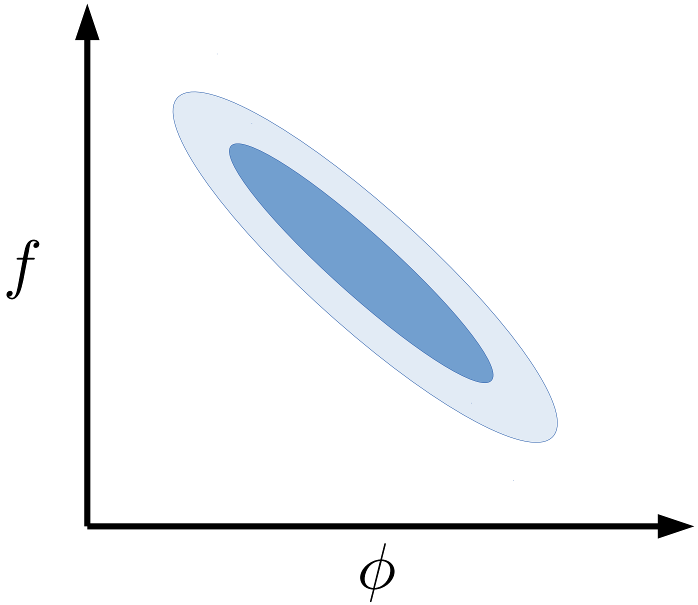


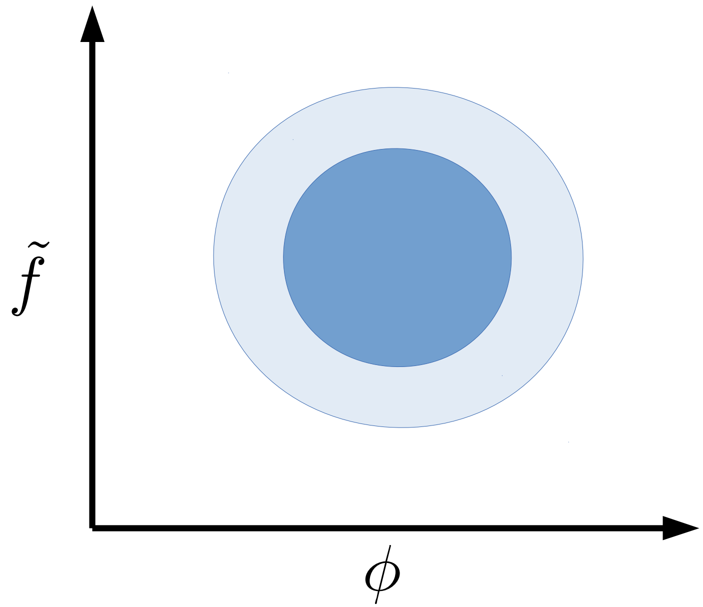
Why is sampling/minimizing $\mathcal{P}(f,\phi\,|\,d)$ such a hard problem?





Why is sampling/minimizing $\mathcal{P}(f, \phi \,|\, d)$ such a hard problem?





So, as pointed out by Anderes et al. 2015, its *very* beneficial to reparametrize,

$$\mathcal{P}(\tilde{f}, \phi \,|\, d) = \mathcal{P}(f(\tilde{f}), \phi \,|\, d) \left| \frac{df}{d\tilde{f}} \right|$$

But now we introduce this determinant...

where $\quad \tilde{f} = \mathcal{L}(\phi) f \implies \left| \frac{df}{d\tilde{f}} \right| = 1/\left|\mathcal{L}(\phi)\right|$

What is the determinant of lensing?

What is the determinant of lensing?

- Infinite resolution: lensing is a remapping (i.e. permutation) so $\det|\mathcal{L}(\phi)| = 1$

What is the determinant of lensing?

- Infinite resolution: lensing is a remapping (i.e. permutation) so $\det|\mathcal{L}(\phi)| = 1$
- This is not the case when we have *pixelization*. Consider the Taylor series approx:

What is the determinant of lensing?

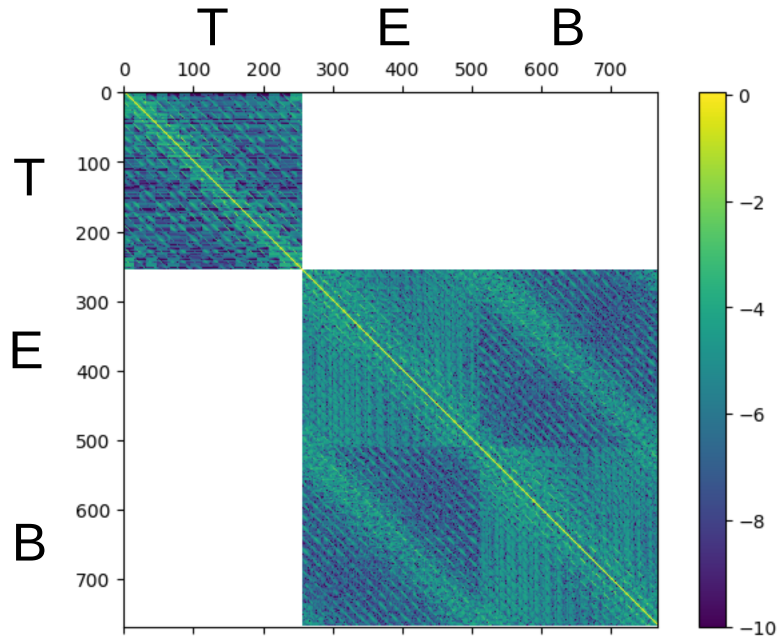
- Infinite resolution: lensing is a remapping (i.e. permutation) so $\det|\mathcal{L}(\phi)| = 1$
- This is not the case when we have *pixelization*. Consider the Taylor series approx:

$$\tilde{f}(x) = f(x + \nabla\phi(x)) \approx \underbrace{\left[1 + \nabla\phi(x) \cdot \nabla + ...\right]}_{\mathcal{L}(\phi)} f(x)$$

What is the determinant of lensing?

- Infinite resolution: lensing is a remapping (i.e. permutation) so $\det|\mathcal{L}(\phi)| = 1$
- This is not the case when we have *pixelization*. Consider the Taylor series approx:

$$\tilde{f}(x) = f(x + \nabla\phi(x)) \approx \underbrace{\left[1 + \nabla\phi(x)\cdot\nabla + ...\right]}_{\mathcal{L}(\phi)} f(x)$$



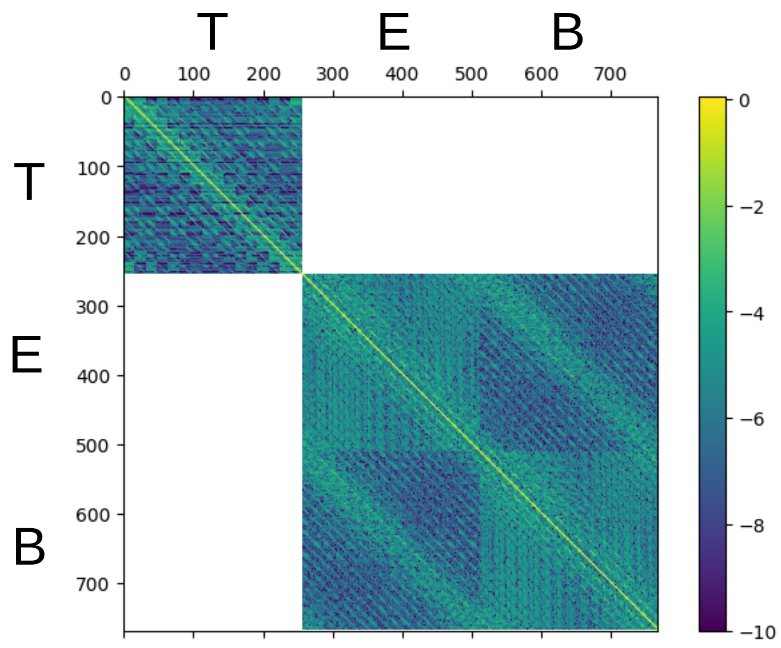
$\log(\mathrm{abs}(\mathcal{L}(\phi)_{ij}))$

Matrix representation of $\mathcal{L}(\phi)$ for 16x16 1’ pixel TEB maps for 7th order Taylor series approximation

What is the determinant of lensing?

- Infinite resolution: lensing is a remapping (i.e. permutation) so $\det|\mathcal{L}(\phi)| = 1$
- This is not the case when we have *pixelization*. Consider the Taylor series approx:

$$\tilde{f}(x) = f(x + \nabla\phi(x)) \approx \underbrace{[1 + \nabla\phi(x)\cdot\nabla + ...]}_{\mathcal{L}(\phi)} f(x)$$



$\log(\mathrm{abs}(\mathcal{L}(\phi)_{ij}))$

Matrix representation of $\mathcal{L}(\phi)$ for 16x16 1' pixel TEB maps for 7th order Taylor series approximation

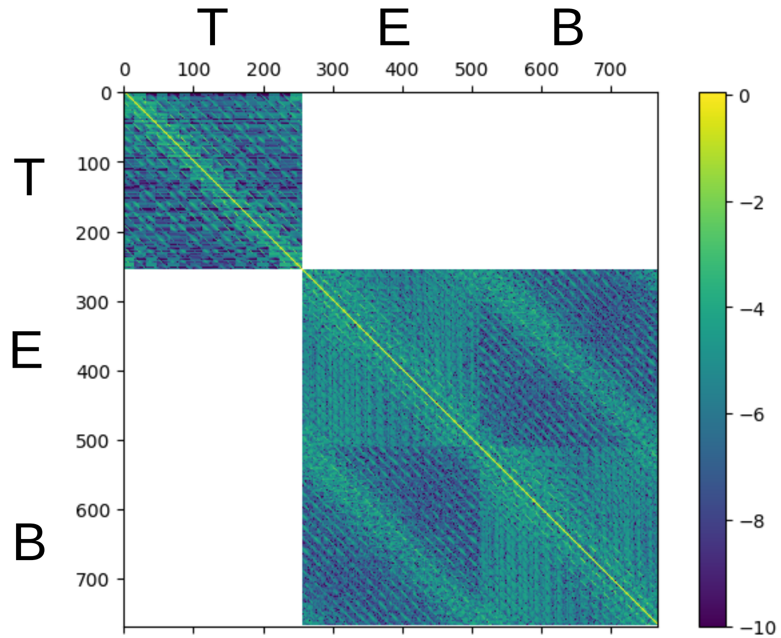
not close to 1!

$$\boxed{\det|\mathcal{L}(\phi)| = 1.9 \times 10^{-9}}$$

What is the determinant of lensing?

- Infinite resolution: lensing is a remapping (i.e. permutation) so $\det|\mathcal{L}(\phi)| = 1$
- This is not the case when we have *pixelization*. Consider the Taylor series approx:

$$\tilde{f}(x) = f(x + \nabla\phi(x)) \approx \underbrace{[1 + \nabla\phi(x)\cdot\nabla + ...]}_{\mathcal{L}(\phi)} f(x)$$



$\log(\mathrm{abs}(\mathcal{L}(\phi)_{ij}))$

Matrix representation of $\mathcal{L}(\phi)$ for 16x16 1' pixel TEB maps for 7th order Taylor series approximation

not close to 1!

$$\boxed{\det|\mathcal{L}(\phi)| = 1.9 \times 10^{-9}}$$

Additionally, the variation of the determinant with ϕ is significant.

A solution: **LenseFlow**

A solution: **LenseFlow**

Define $f_t(x) \equiv f(x + t\nabla\phi(x))$ s.t. $f_{t=0}(x) = f(x)$

$f_{t=1}(x) = \tilde{f}(x)$

A solution: # LenseFlow

Define $f_t(x) \equiv f(x + t\nabla\phi(x))$ s.t.

$$f_{t=0}(x) = f(x)$$
$$f_{t=1}(x) = \tilde{f}(x)$$

One can show f_t obeys an ODE “flow” equation

$$\frac{df_t(x)}{dt} = \nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1} \cdot \nabla f_t(x)$$

A solution: LenseFlow

Define $f_t(x) \equiv f(x + t\nabla\phi(x))$ s.t. $f_{t=0}(x) = f(x)$
$f_{t=1}(x) = \tilde{f}(x)$

One can show f_t obeys an ODE "flow" equation

$$\frac{df_t(x)}{dt} = \nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1} \cdot \nabla f_t(x)$$

- To *lense* a map, just run the ODE from t=0 to t=1

A solution: # LenseFlow

Define $f_t(x) \equiv f(x + t\nabla\phi(x))$ s.t. $f_{t=0}(x) = f(x)$, $f_{t=1}(x) = \tilde{f}(x)$

One can show f_t obeys an ODE "flow" equation

$$\frac{df_t(x)}{dt} = \nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1} \cdot \nabla f_t(x)$$

- To *lense* a map, just run the ODE from t=0 to t=1
- To *delense* a map, just run it backwards from t=1 to t=0

A solution: LenseFlow

Define $f_t(x) \equiv f(x + t\nabla\phi(x))$ s.t. $f_{t=0}(x) = f(x)$

$f_{t=1}(x) = \tilde{f}(x)$

One can show f_t obeys an ODE “flow” equation

$$\frac{df_t(x)}{dt} = \nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1} \cdot \nabla f_t(x)$$

- To *lense* a map, just run the ODE from t=0 to t=1
- To *delense* a map, just run it backwards from t=1 to t=0
- This operation provably has determinant = 1

A solution: # LenseFlow

Define $f_t(x) \equiv f(x + t\nabla\phi(x))$ s.t. $f_{t=0}(x) = f(x)$

$f_{t=1}(x) = \tilde{f}(x)$

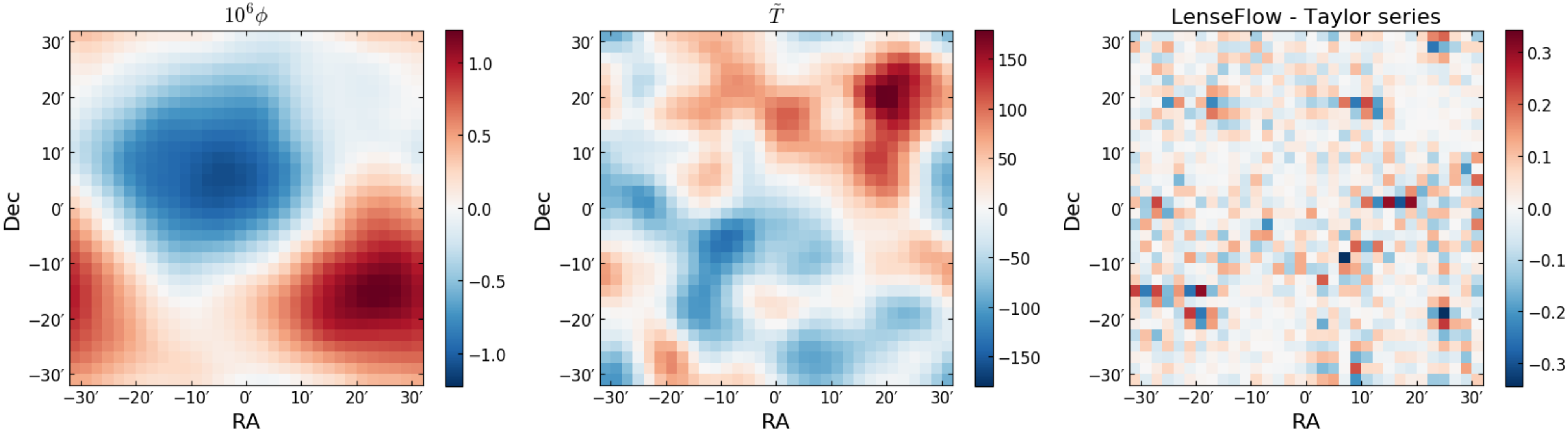
One can show f_t obeys an ODE “flow” equation

$$\frac{df_t(x)}{dt} = \nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1} \cdot \nabla f_t(x)$$

- To *lense* a map, just run the ODE from t=0 to t=1
- To *delense* a map, just run it backwards from t=1 to t=0
- This operation provably has determinant = 1

(In practice we use 4$^{\text{th}}$ order Runge-Kutta with 7 time-steps.)

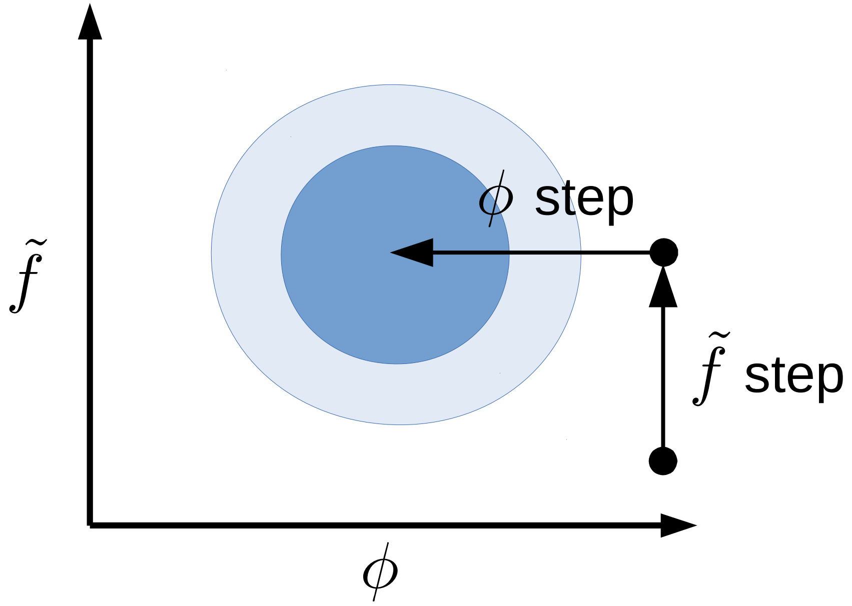
LenseFlow vs. Taylor series



Differences between the two which lead to different determinants

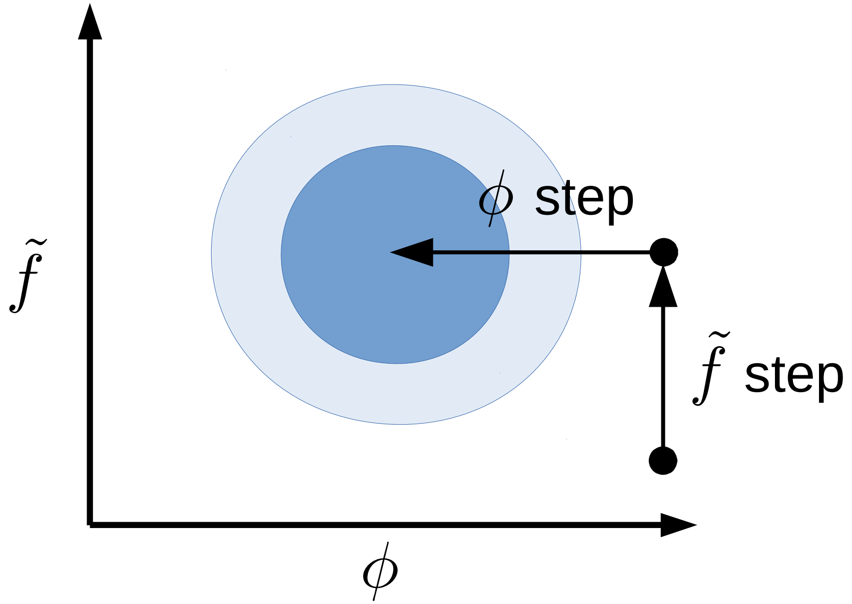
Ok, let's maximize & sample!

The algorithm we devise is a *coordinate descent*



Ok, let's maximize & sample!

The algorithm we devise is a *coordinate descent*

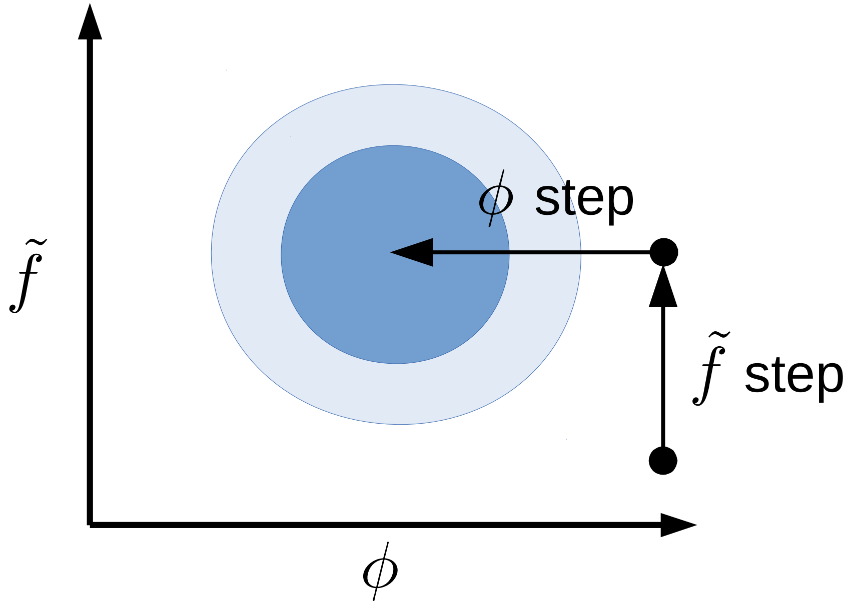


$$-2\ln\,\mathcal{P}(\tilde{f},\phi\,|\,d) =$$

$$= \underbrace{(d-\tilde{f})^{\dagger}\mathcal{C}_n^{-1}(d-\tilde{f})}_{\text{likelihood}} + \underbrace{\tilde{f}^{\dagger}\mathcal{L}(\phi)^{-\dagger}\mathcal{C}_f^{-1}\mathcal{L}(\phi)^{-1}\tilde{f}}_{\text{prior on } f} + \underbrace{\phi^{\dagger}\mathcal{C}_{\phi}{}^{-1}\phi}_{\text{prior on } \phi}$$

Ok, let's maximize & sample!

The algorithm we devise is a *coordinate descent*

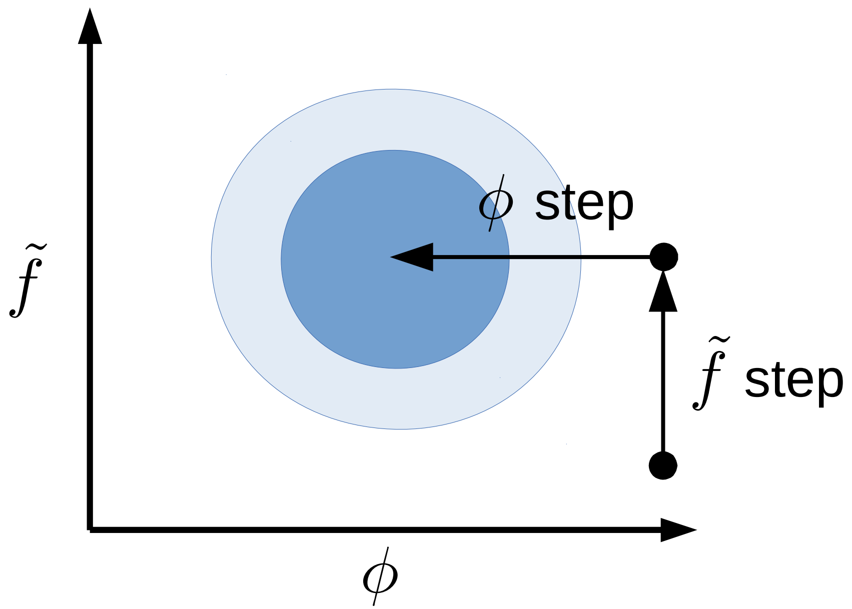


$$-2\ln\mathcal{P}(\tilde{f},\phi\,|\,d) =$$

$$= \overbrace{\underbrace{(d-\tilde{f})^\dagger \mathcal{C}_n^{-1}(d-\tilde{f})}_{\text{likelihood}} + \underbrace{\tilde{f}^\dagger \mathcal{L}(\phi)^{-\dagger}\mathcal{C}_f^{-1}\mathcal{L}(\phi)^{-1}\tilde{f}}_{\text{prior on } f}}^{\tilde{f}\text{ step : a Wiener filter}} + \underbrace{\phi^\dagger \mathcal{C}_\phi{}^{-1}\phi}_{\text{prior on } \phi}$$

Ok, let's maximize & sample!

The algorithm we devise is a *coordinate descent*



$$-2\ln \mathcal{P}(\tilde{f},\phi\,|\,d) =$$

$$= \overbrace{\underbrace{(d-\tilde{f})^\dagger \mathcal{C}_n^{-1}(d-\tilde{f})}_{\text{likelihood}} + \underbrace{\tilde{f}^\dagger \mathcal{L}(\phi)^{-\dagger}\mathcal{C}_f^{-1}\mathcal{L}(\phi)^{-1}\tilde{f}}_{\text{prior on } f}}^{\tilde{f}\text{ step : a Wiener filter}} + \underbrace{\phi^\dagger {\mathcal{C}_\phi}^{-1}\phi}_{\text{prior on } \phi}$$

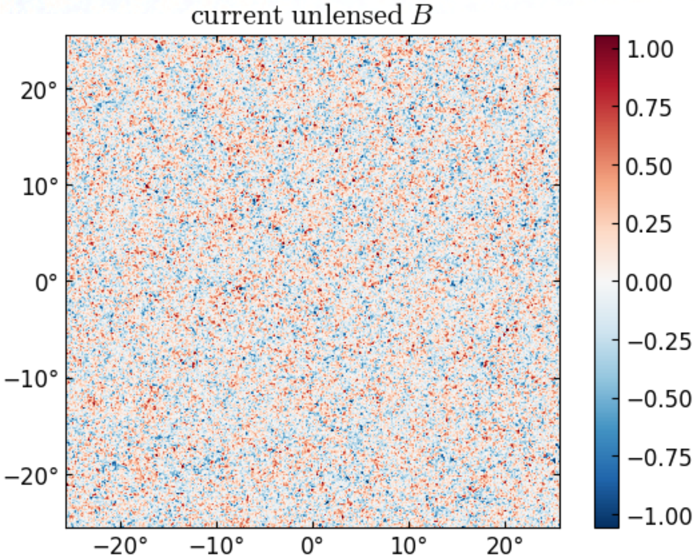
ϕ step

Starting point: $\phi = 0$

Simulated data with: 1uK-arcmin noise, r=0.05

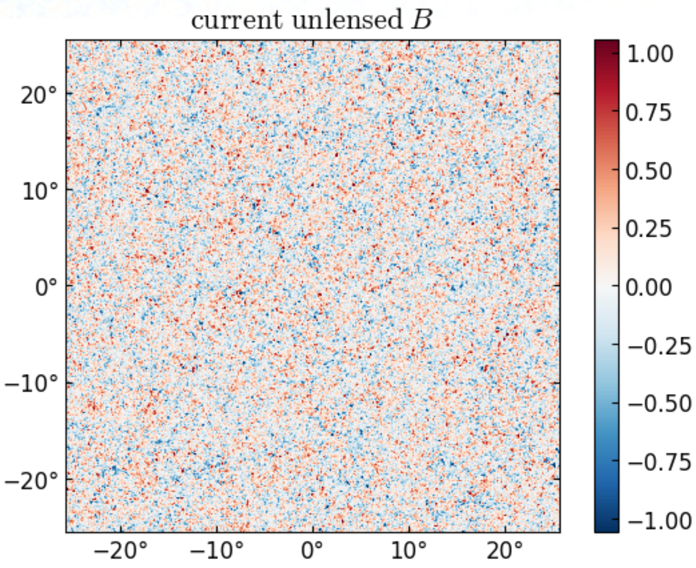
Starting point: $\phi = 0$

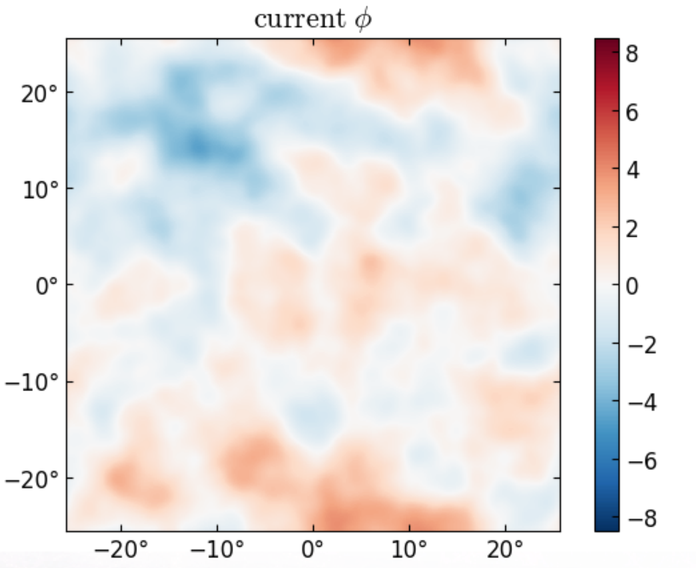
Simulated data with: 1uK-arcmin noise, r=0.05



Starting point: $\phi = 0$

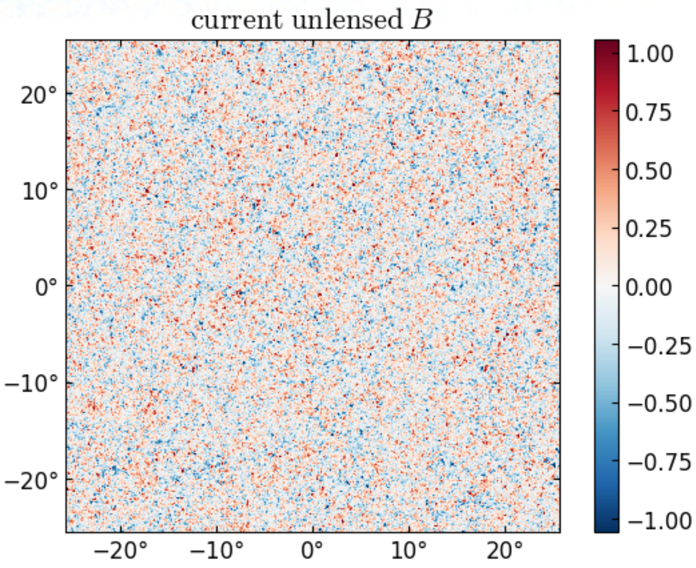
Simulated data with: 1uK-arcmin noise, r=0.05

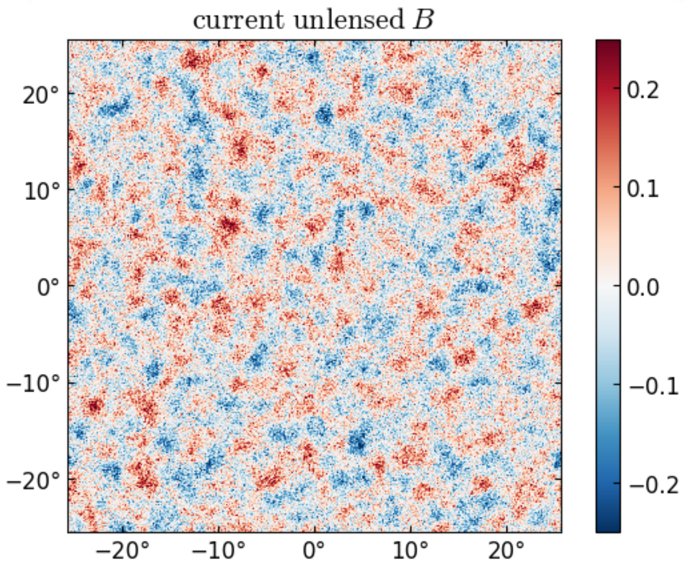


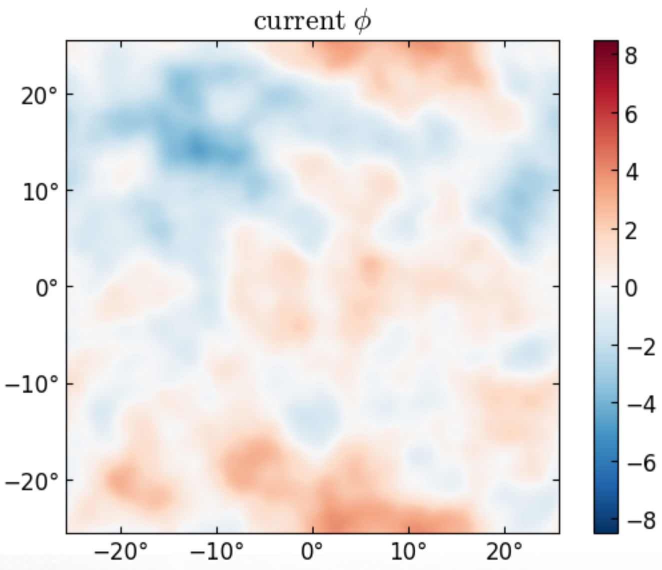


Starting point: $\phi = 0$

Simulated data with: 1uK-arcmin noise, r=0.05

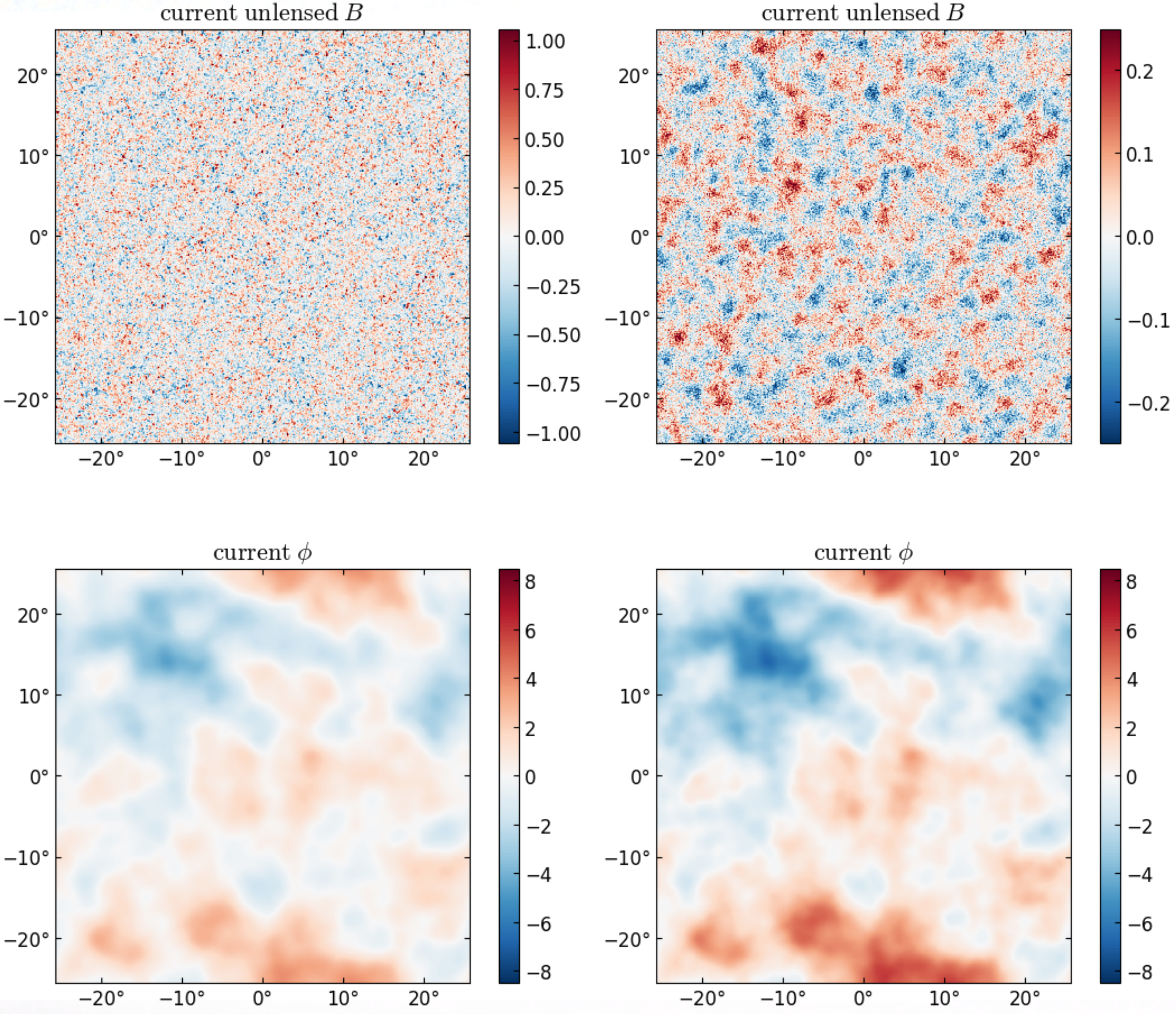






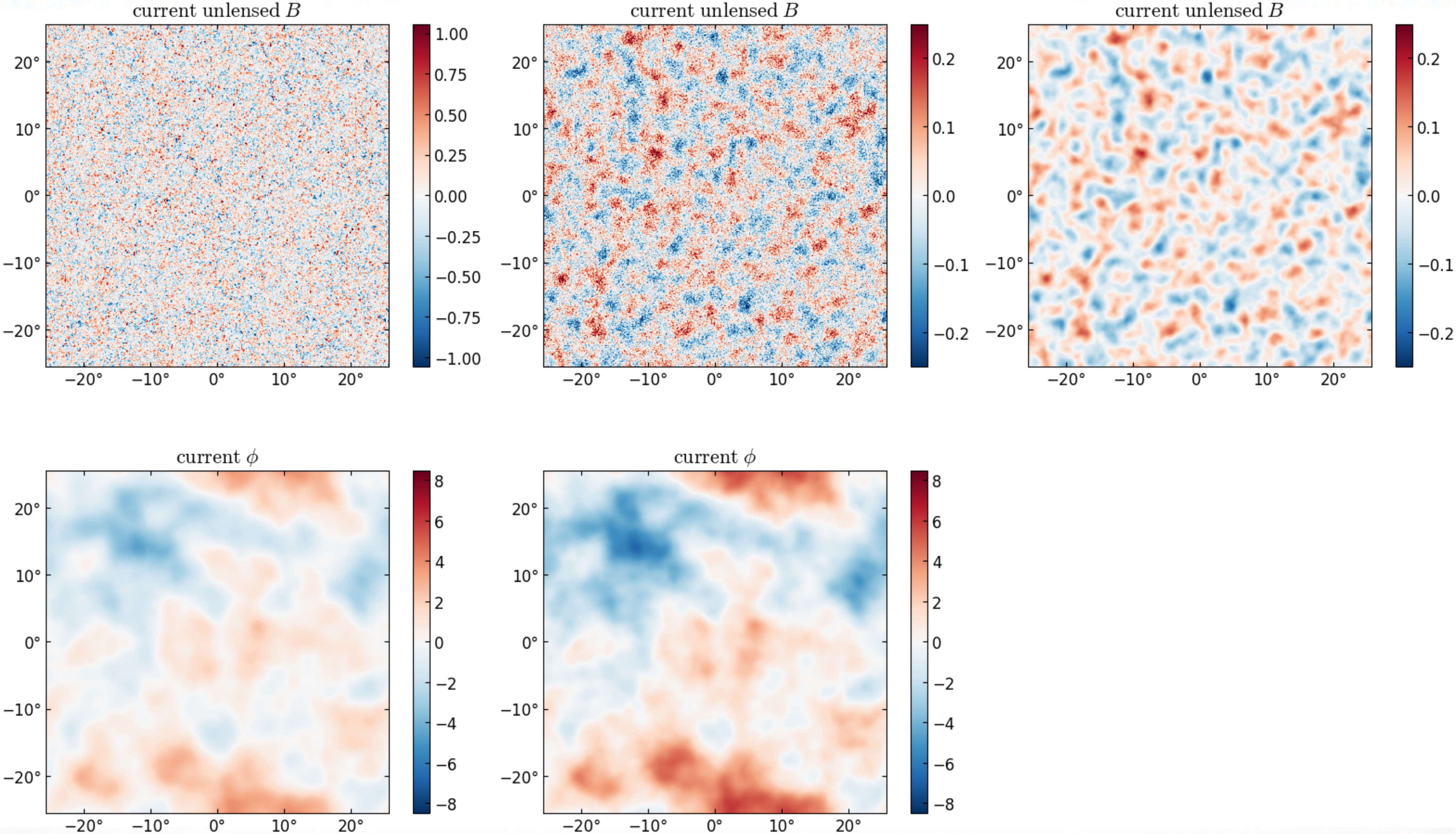
Starting point: $\phi = 0$

Simulated data with: 1uK-arcmin noise, r=0.05



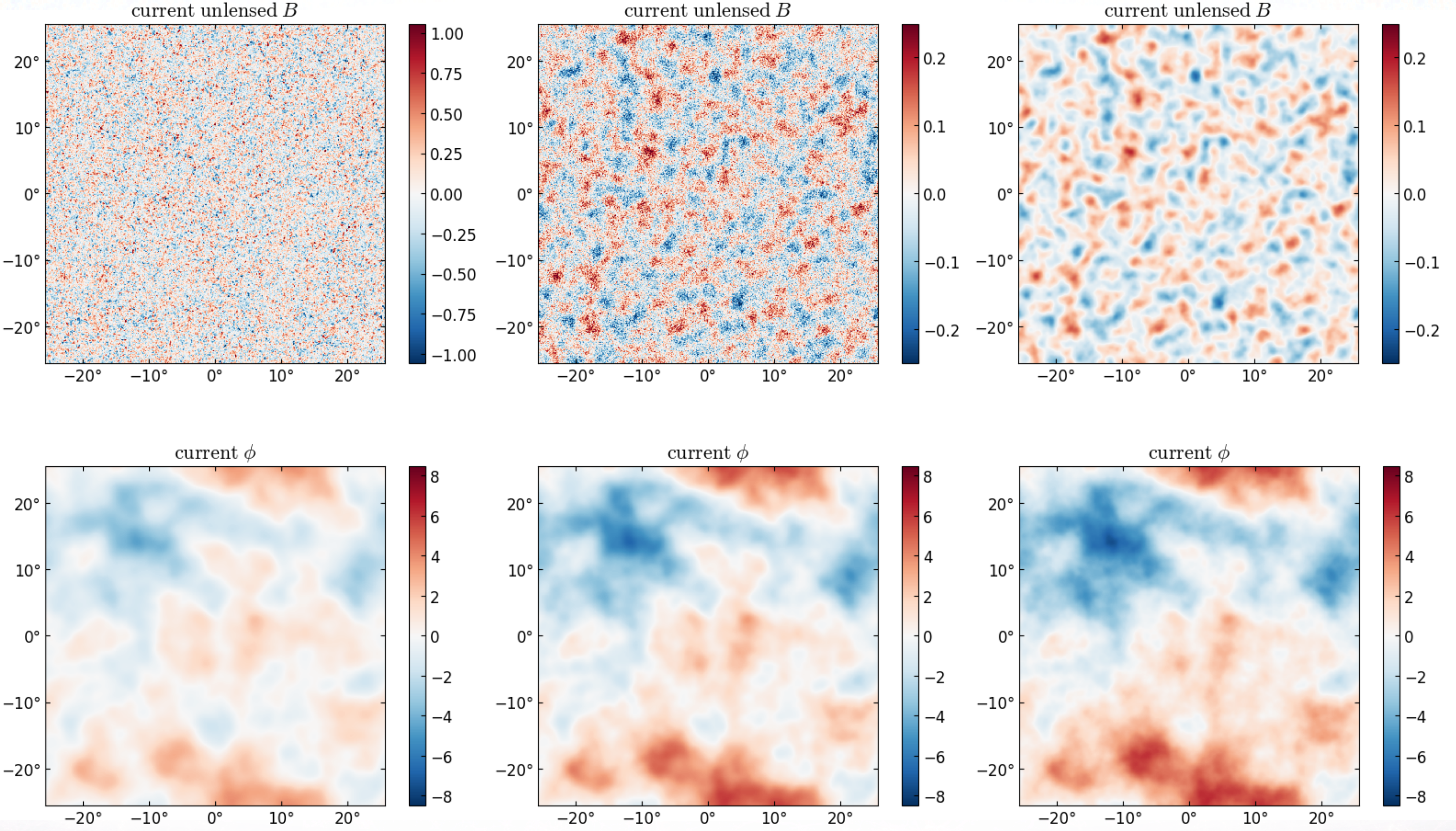
Starting point: $\phi = 0$

Simulated data with: 1uK-arcmin noise, r=0.05



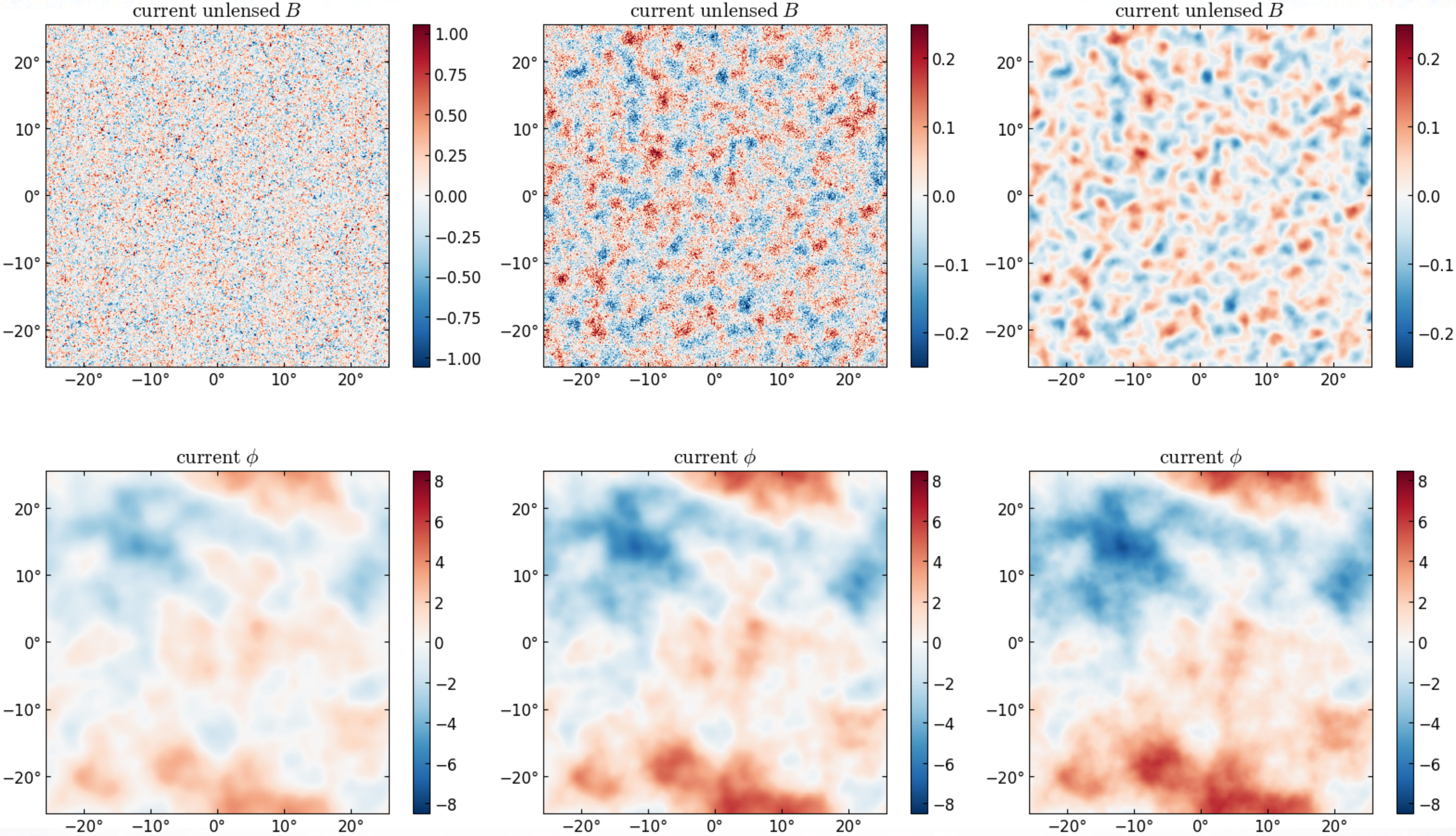
Starting point: $\phi = 0$

Simulated data with: 1uK-arcmin noise, r=0.05



Starting point: $\phi = 0$

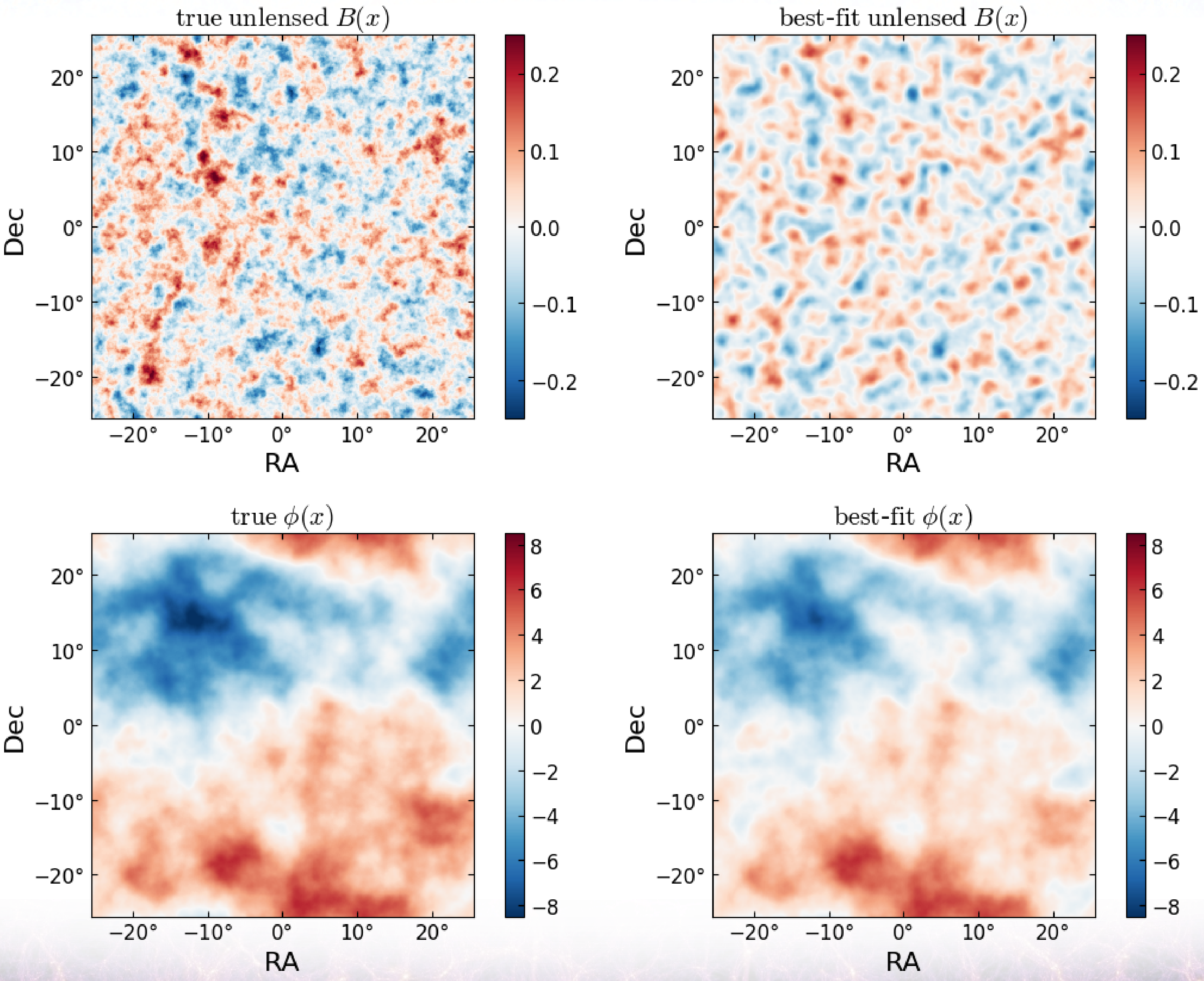
Simulated data with: 1uK-arcmin noise, r=0.05



Step 1 Step 3 Step 30

30min on 1 single multi-core CPU for these 2500deg^2

1024x1024, 3 arcmin pixels

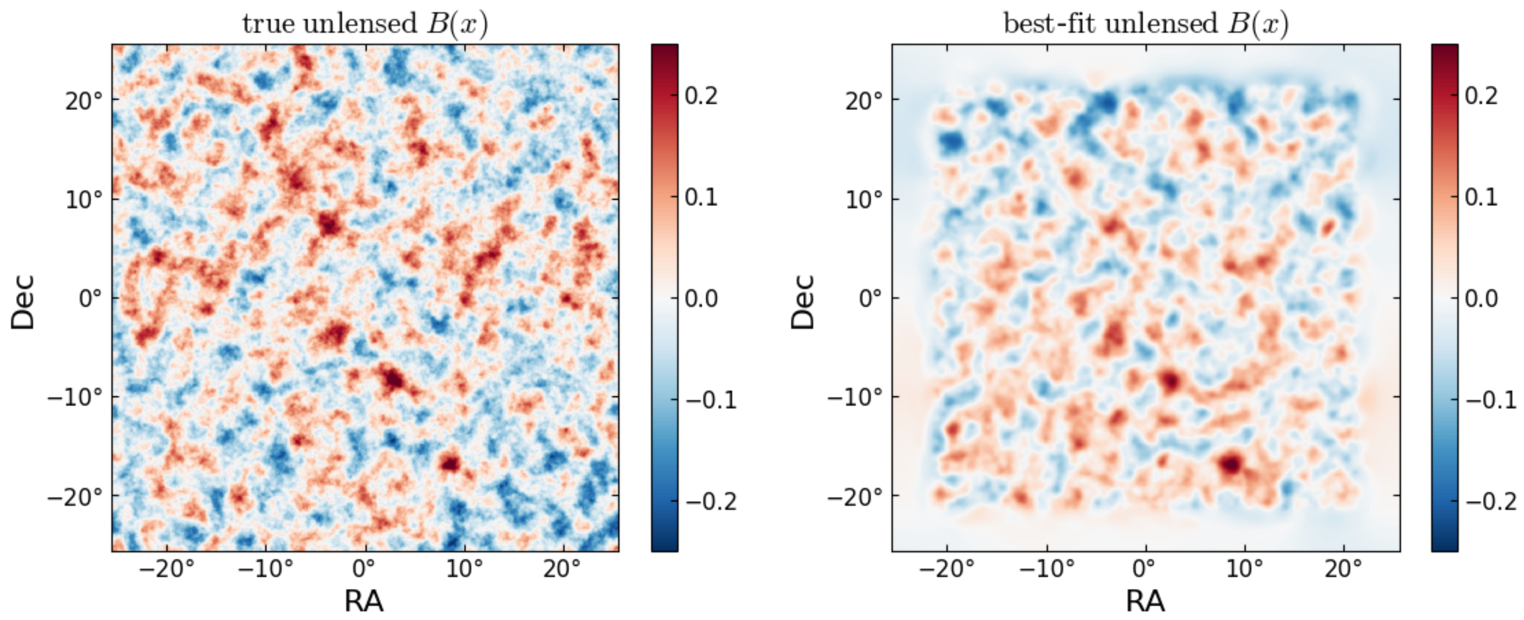


Masking works too

(Only affects the Wiener filter step which needs more conjugate gradient steps => 4 hours)

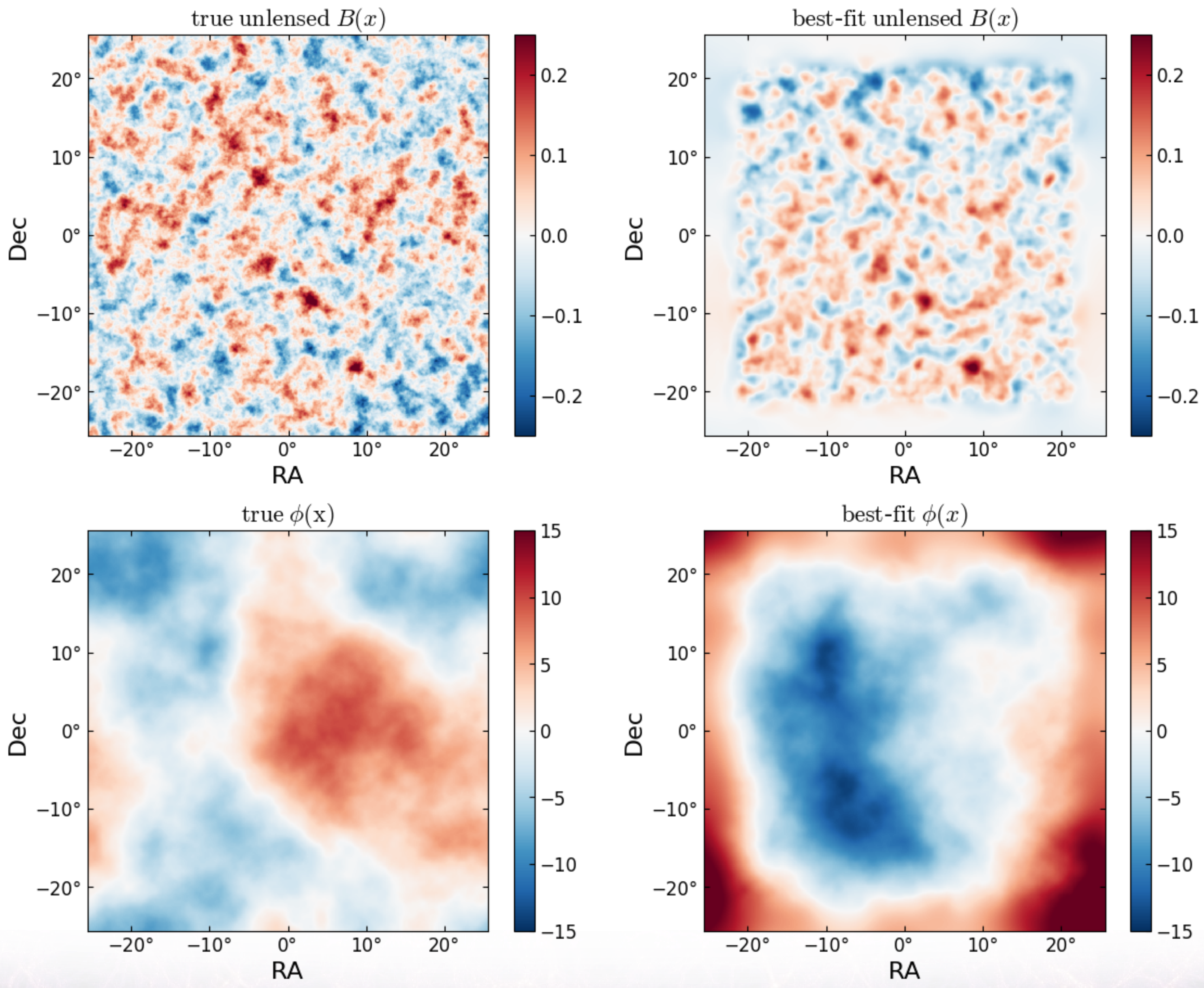
Masking works too

(Only affects the Wiener filter step which needs more conjugate gradient steps => 4 hours)



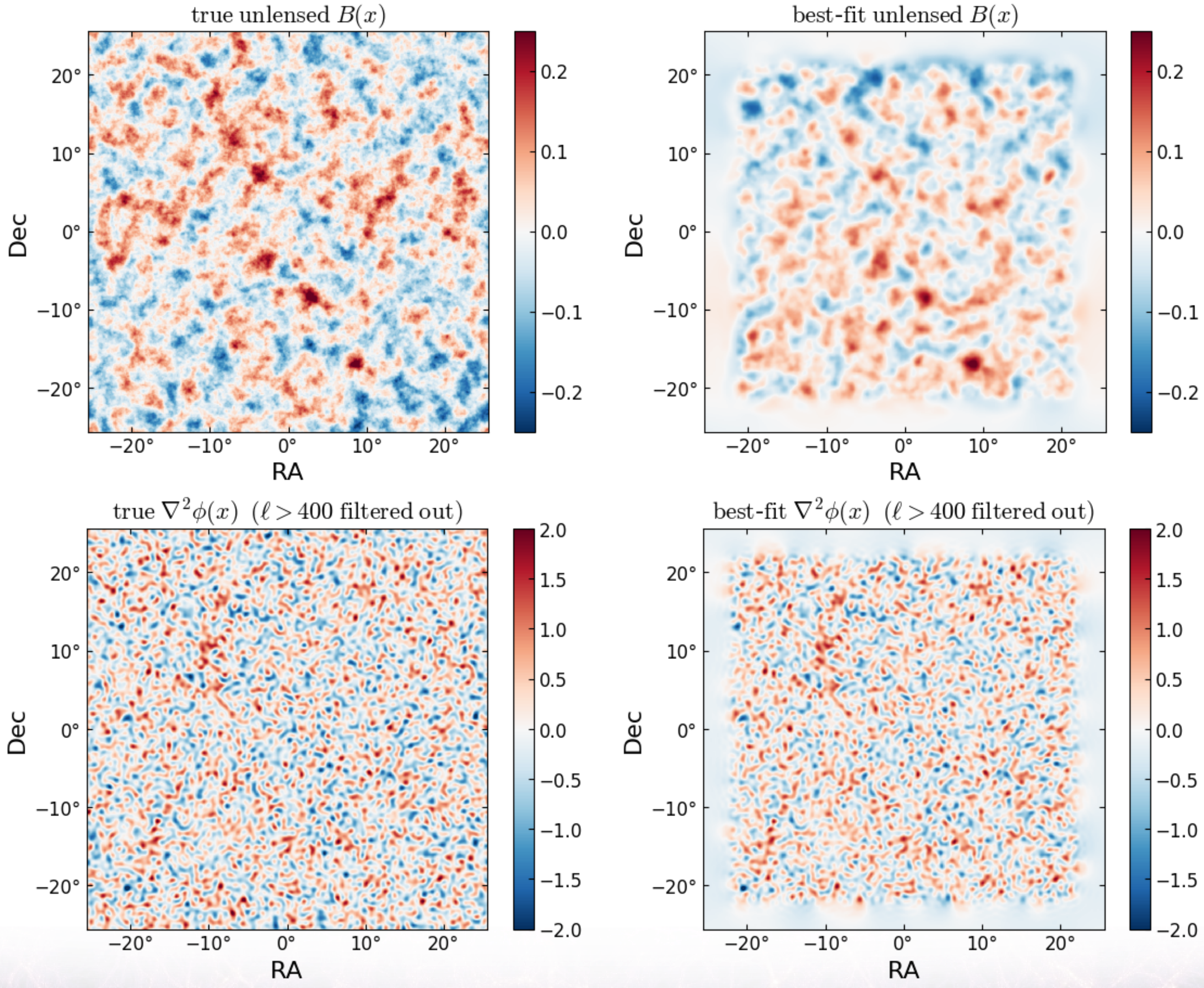
Masking works too

(Only affects the Wiener filter step which needs more conjugate gradient steps => 4 hours)



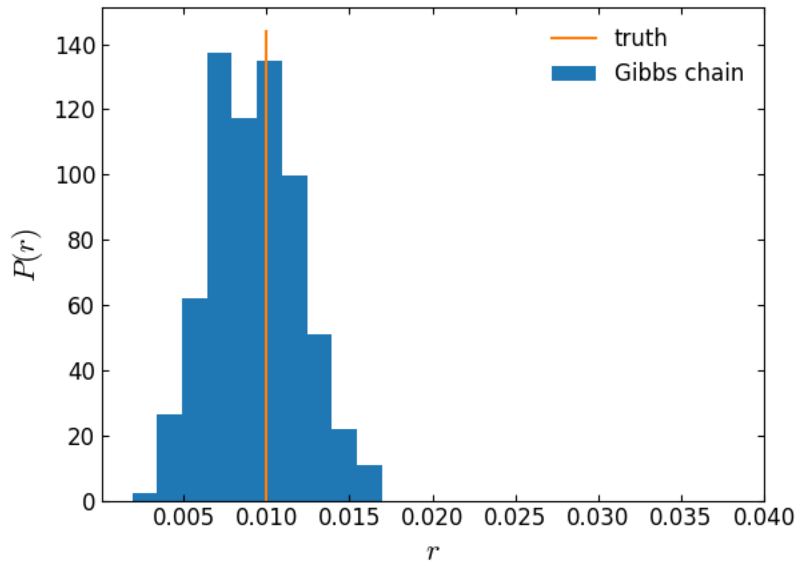
Masking works too

(Only affects the Wiener filter step which needs more conjugate gradient steps => 4 hours)

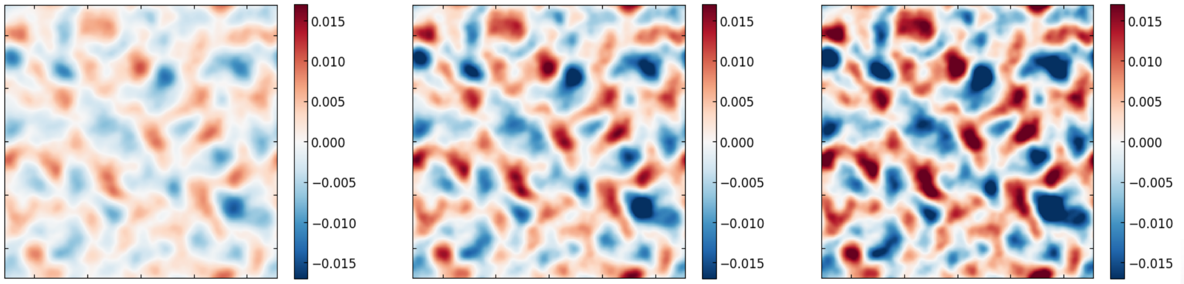


What about r?

For now, a slightly simplified preview: $\mathcal{P}(f, \hat{\phi}, r \,|\, d)$



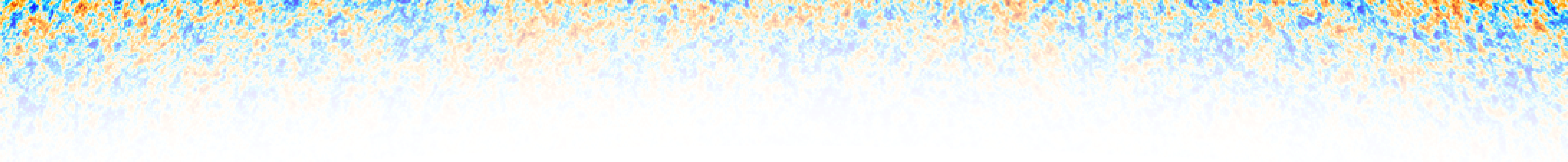
Samples of :

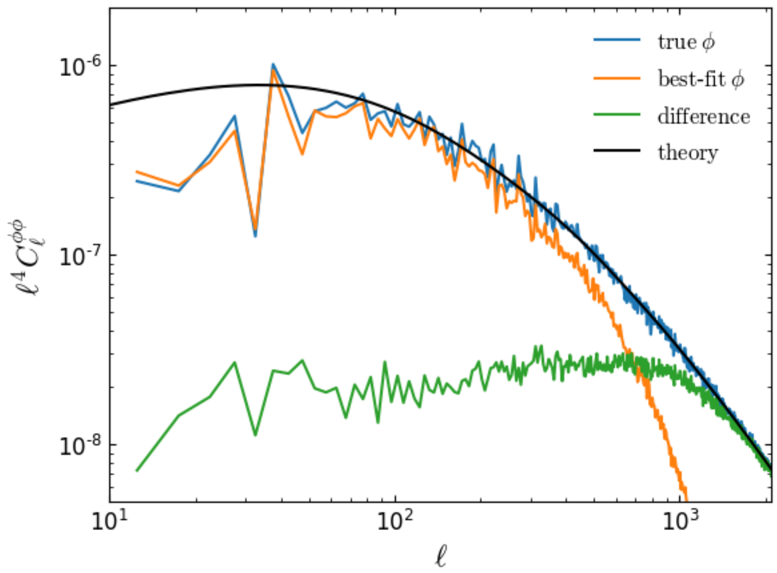


...

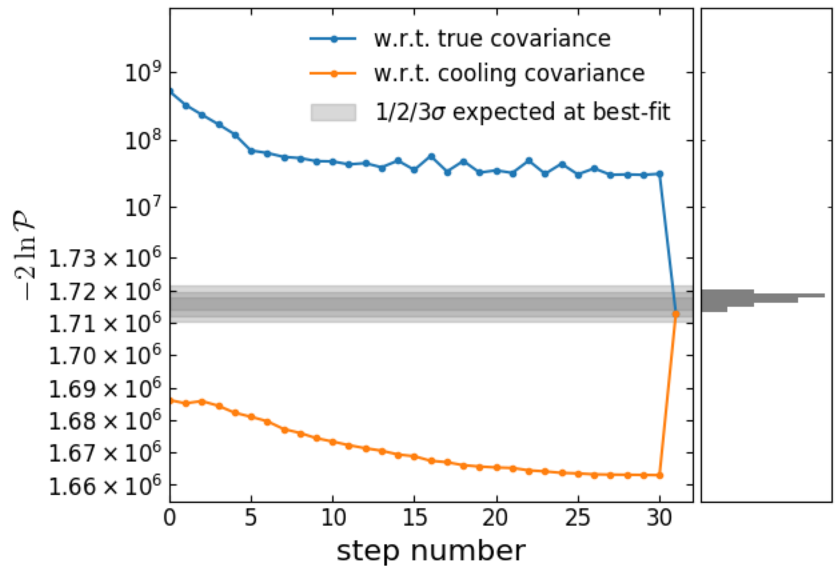
Conclusions

- We can maximize $\mathcal{P}(f, \phi, r \,|\, d)$
- Sampling is coming up and I've given you a preview of it
- Looking forward to more improvement, application to data, and feedback from the community (see our paper soon!)





true ϕ
best-fit ϕ
difference
theory
$\ell^4 C_\ell^{\phi\phi}$
ℓ
10^{-6}
10^{-7}
10^{-8}
10^1
10^2
10^3



w.r.t. true covariance
w.r.t. cooling covariance
1/2/3σ expected at best-fit
$-2\ln\mathcal{P}$
step number
10^9
10^8
10^7
1.73×10^6
1.72×10^6
1.71×10^6
1.70×10^6
1.69×10^6
1.68×10^6
1.67×10^6
1.66×10^6
0
5
10
15
20
25
30

LenseFlow determinant

LenseFlow determinant

$$\frac{df_t(x)}{dt} = \underbrace{\nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1}}_{p_t} \cdot \nabla f(x)$$

LenseFlow determinant

$$\frac{df_t(x)}{dt} = \underbrace{\nabla\phi(x)\cdot\left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1}}_{p_t}\cdot\nabla f(x)$$

$$\mathcal{L}(\phi) = \left[\mathbb{1} + \varepsilon\; p_{t_n}\cdot\nabla\right]\cdots\left[\mathbb{1} + \varepsilon\; p_{t_0}\cdot\nabla\right]$$

LenseFlow determinant

$$\frac{df_t(x)}{dt} = \underbrace{\nabla\phi(x) \cdot [\mathbb{1} + t\nabla\nabla\phi(x)]^{-1}}_{p_t} \cdot \nabla f(x)$$

$$\mathcal{L}(\phi) = [\mathbb{1} + \varepsilon \; p_{t_n} \cdot \nabla] \cdots [\mathbb{1} + \varepsilon \; p_{t_0} \cdot \nabla]$$

$$\operatorname{logdet}[\mathbb{1} + \varepsilon \; p_t \cdot \nabla] = \varepsilon \operatorname{Tr}[p_t \cdot \nabla] + \mathcal{O}(\epsilon^2)$$

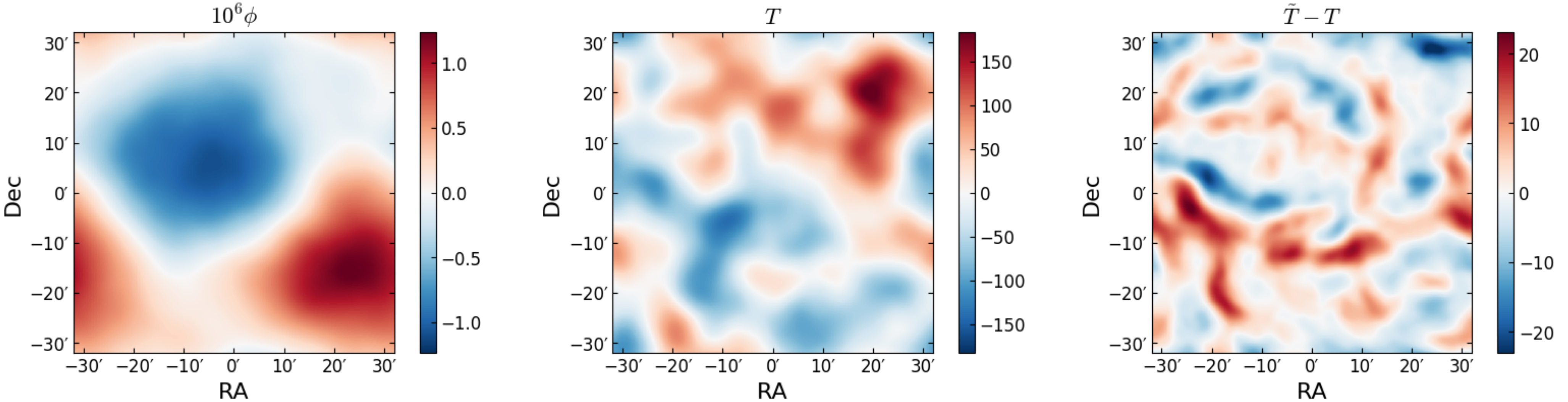
LenseFlow determinant

$$\frac{df_t(x)}{dt} = \underbrace{\nabla\phi(x) \cdot \left[\mathbb{1} + t\nabla\nabla\phi(x)\right]^{-1}}_{p_t} \cdot \nabla f(x)$$

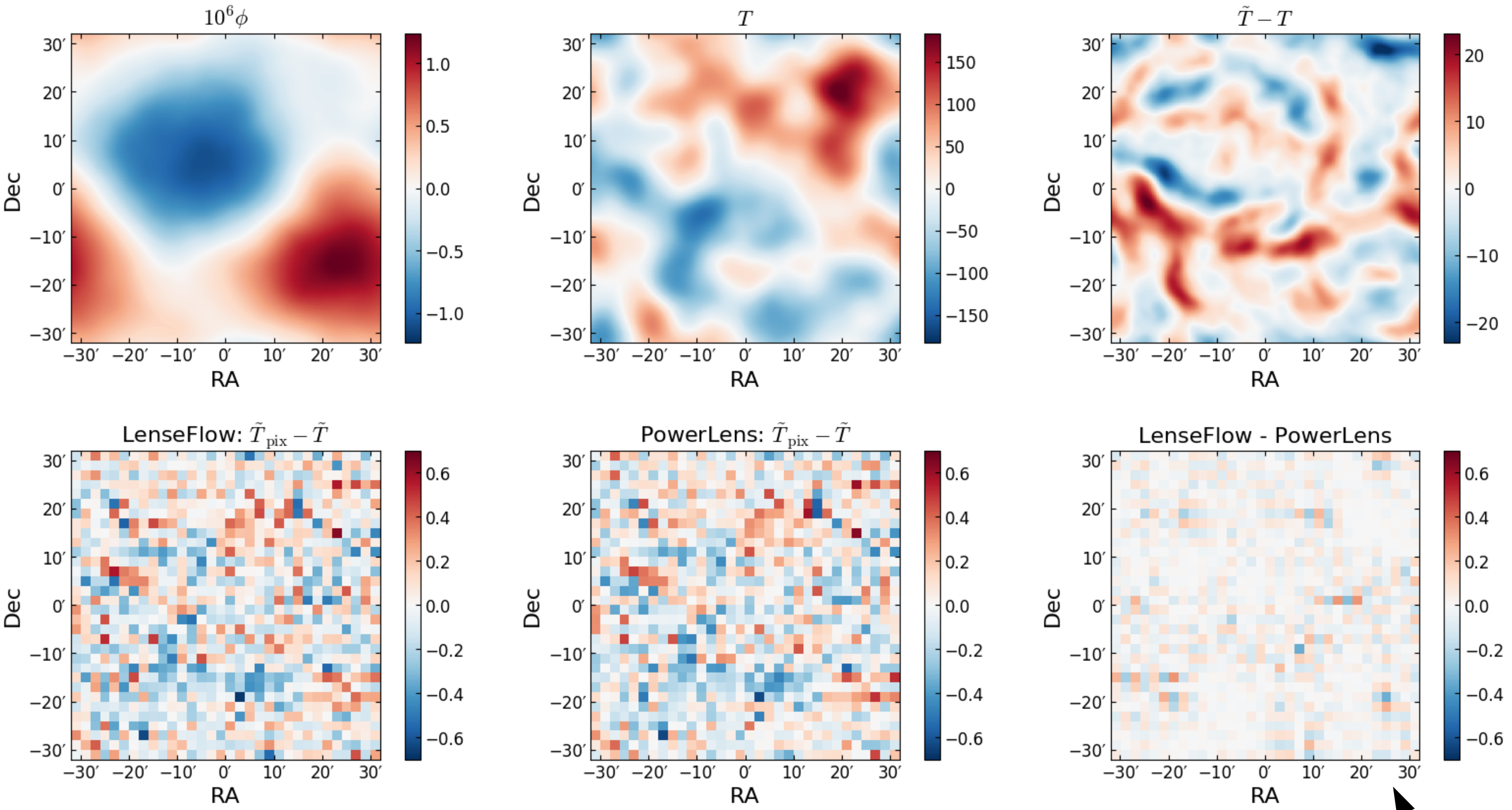
$$\mathcal{L}(\phi) = \left[\mathbb{1} + \varepsilon\; p_{t_n} \cdot \nabla\right] \cdots \left[\mathbb{1} + \varepsilon\; p_{t_0} \cdot \nabla\right]$$

$$\mathrm{logdet}\left[\mathbb{1} + \varepsilon\; p_t \cdot \nabla\,\right] = \varepsilon\, \mathrm{Tr}\left[\,p_t \cdot \nabla\,\right] + \mathcal{O}(\epsilon^2)$$

So for LenseFlow $\det|\mathcal{L}(\phi)| = 1$ so we can ignore it!



$10^6\phi$
T
$\tilde{T} - T$
Dec
RA
−30′ −20′ −10′ 0′ 10′ 20′ 30′
1.0 0.5 0.0 −0.5 −1.0
150 100 50 0 −50 −100 −150
20 10 0 −10 −20



LenseFlow

Taylor series

Differences between two which lead to different determinants