

Wavelet Analysis

Spectral Analysis

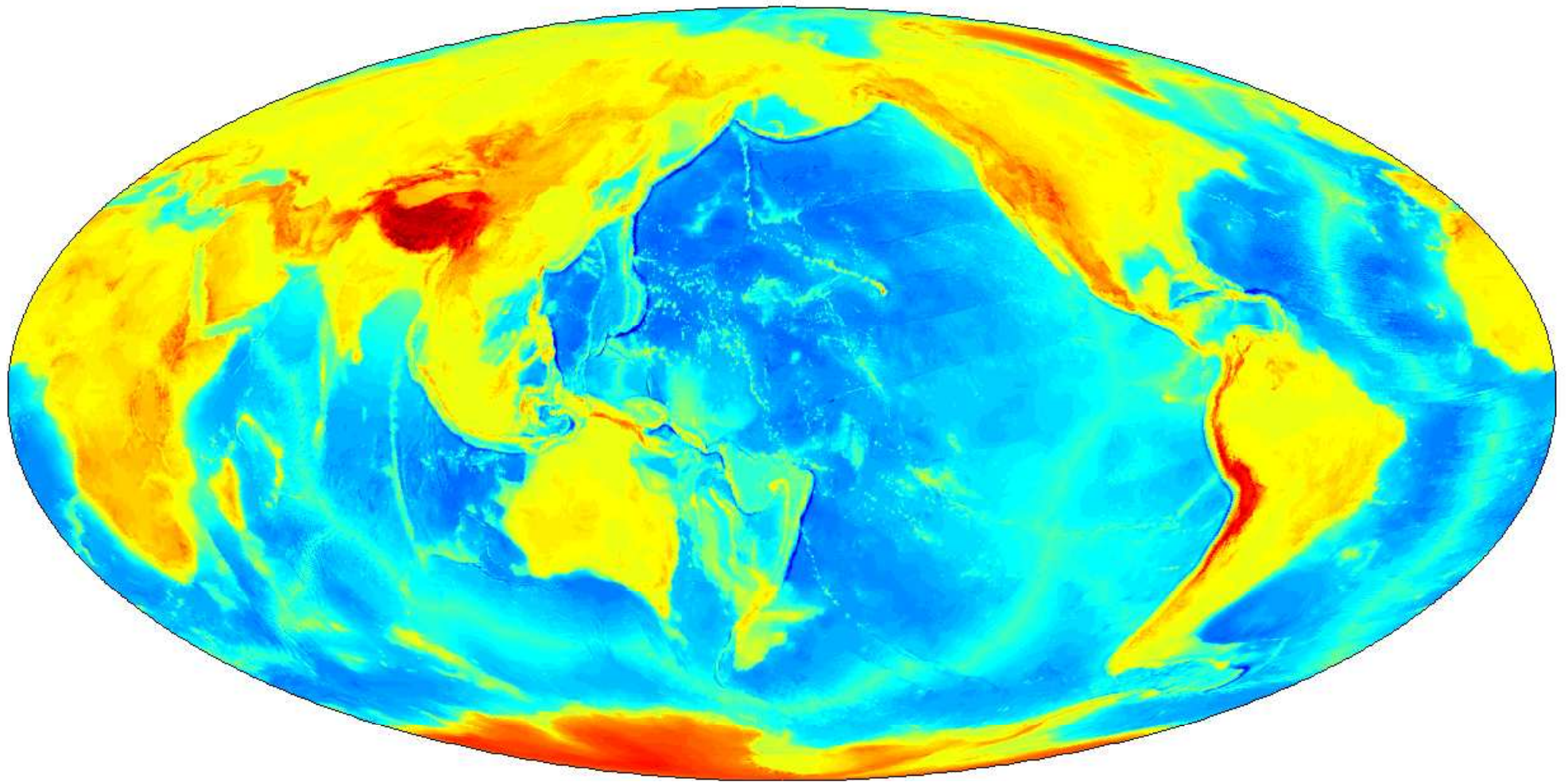
...on the sphere! ...in MATLAB!

Frederik J. Simons

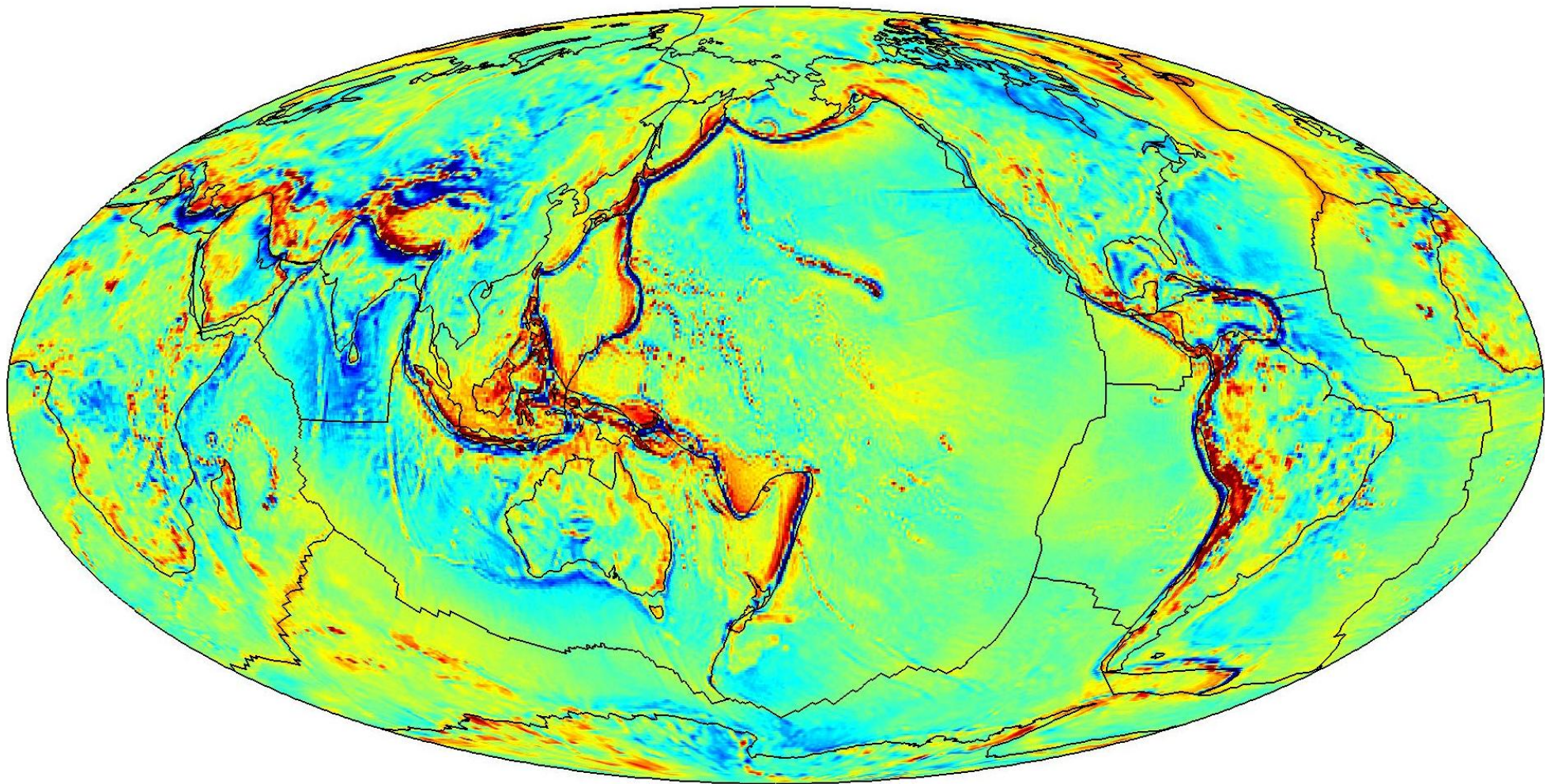
Princeton University



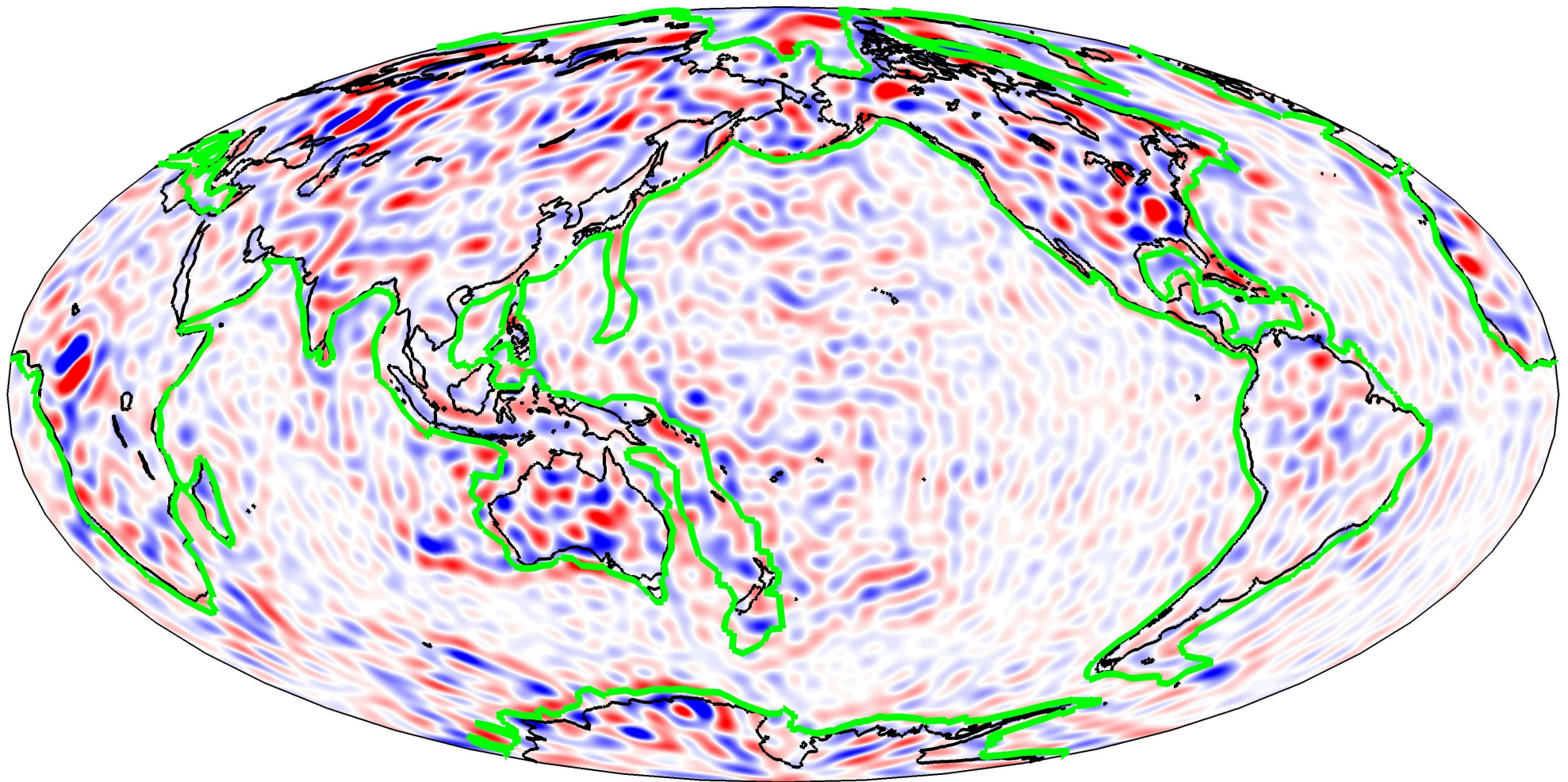
Planetary topography



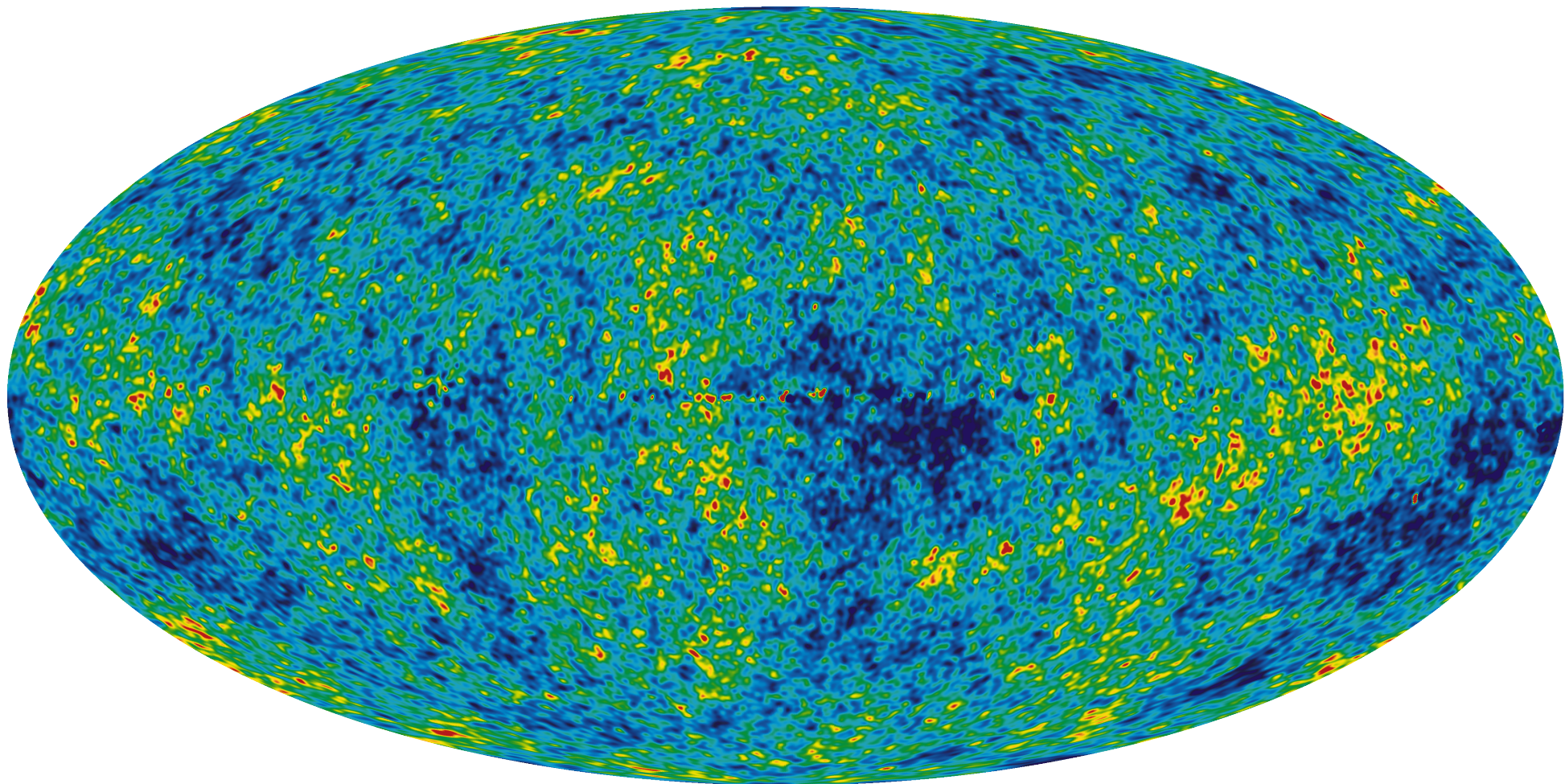
Planetary gravity



Planetary geomagnetism



Sky temperature



Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform



Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform

|

- The spherical *harmonic* transform

||

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

The spherical *wavelet* transform

I

The spherical *harmonic* transform

II

The spherical *Slepian* transform

III

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform

I

- The spherical *harmonic* transform

II

- The spherical *Slepian* transform

III

- Estimate **power spectra**; but keep them **localized**

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform I

- The spherical *harmonic* transform II

- The spherical *Slepian* transform III

- Estimate **power spectra**; but keep them **localized**

- Non-parametrically: the multitaper method IV

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform

I

- The spherical *harmonic* transform

II

- The spherical *Slepian* transform

III

- Estimate **power spectra**; but keep them **localized**

- Non-parametrically: the multitaper method

IV

- Parametrically: the Matérn class

V

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform I

- The spherical *harmonic* transform II

- The spherical *Slepian* transform III

- Estimate **power spectra**; but keep them **localized**

- Non-parametrically: the multitaper method IV

- Parametrically: the Matérn class V

- Estimate **cross-power spectra**: multivariate & multidimensional

Looking at planets

- Study data $d(\theta, \varphi)$ on a *sphere*.

Move **beyond pixels**. Study **wavelengths**. Respect **geography**.

- Find **transforms** that illuminate **spatio-spectral** structure, **parsimoniously**

- The spherical *wavelet* transform I

- The spherical *harmonic* transform II

- The spherical *Slepian* transform III

- Estimate **power spectra**; but keep them **localized**

- Non-parametrically: the multitaper method IV

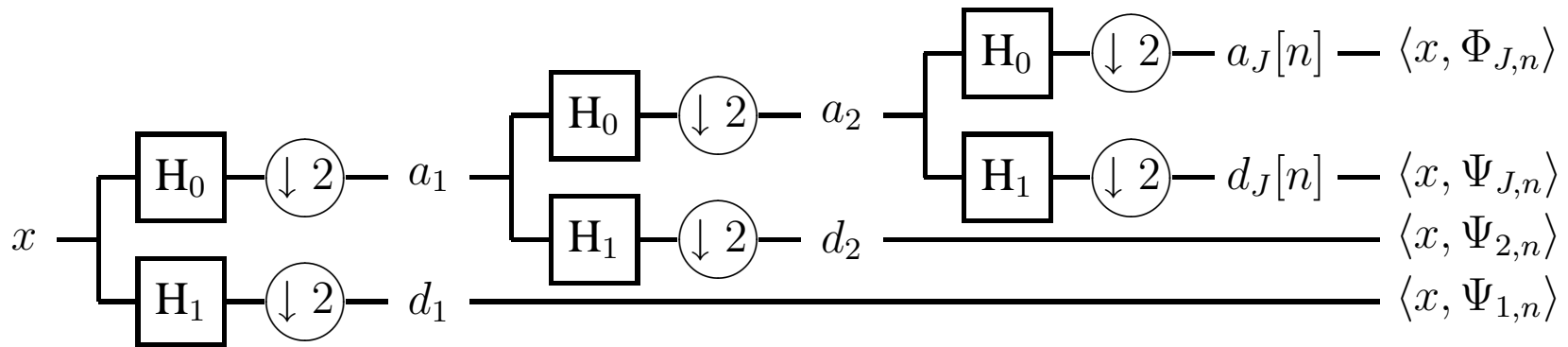
- Parametrically: the Matérn class V

- Estimate **cross-power spectra**: multivariate & multidimensional

- Revisiting **flexure** via maximum-likelihood estimation VI

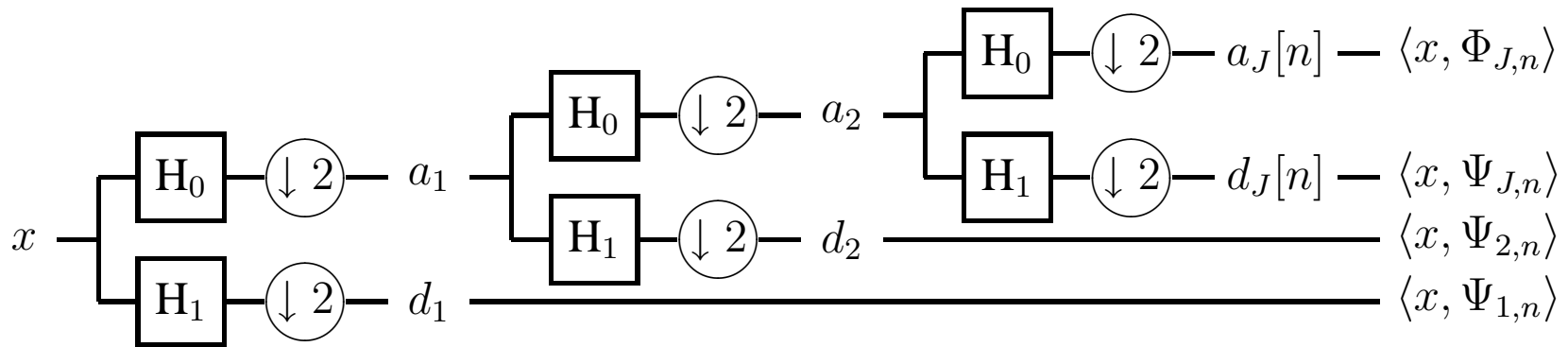
I The wavelet transform

Cartesian wavelets are “easy” — well understood, fast algorithms, and proper accounting of boundaries (wavelets “*on the interval*”).



I The wavelet transform

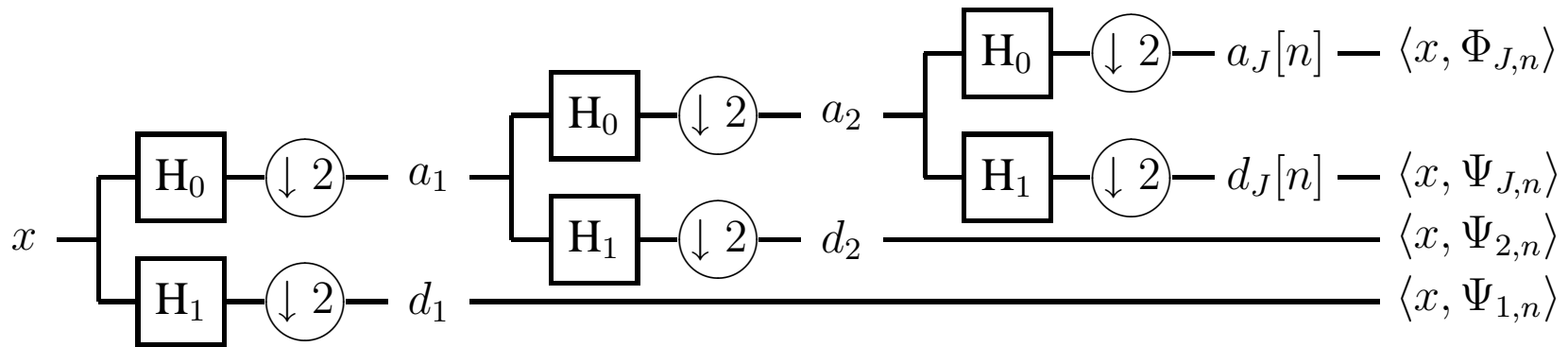
Cartesian wavelets are “easy” — well understood, fast algorithms, and proper accounting of boundaries (wavelets “*on the interval*”).



If we could treat the Earth like a **three-dimensional Cartesian** box, we would.

I The wavelet transform

Cartesian wavelets are “easy” — well understood, fast algorithms, and proper accounting of boundaries (wavelets “*on the interval*”).

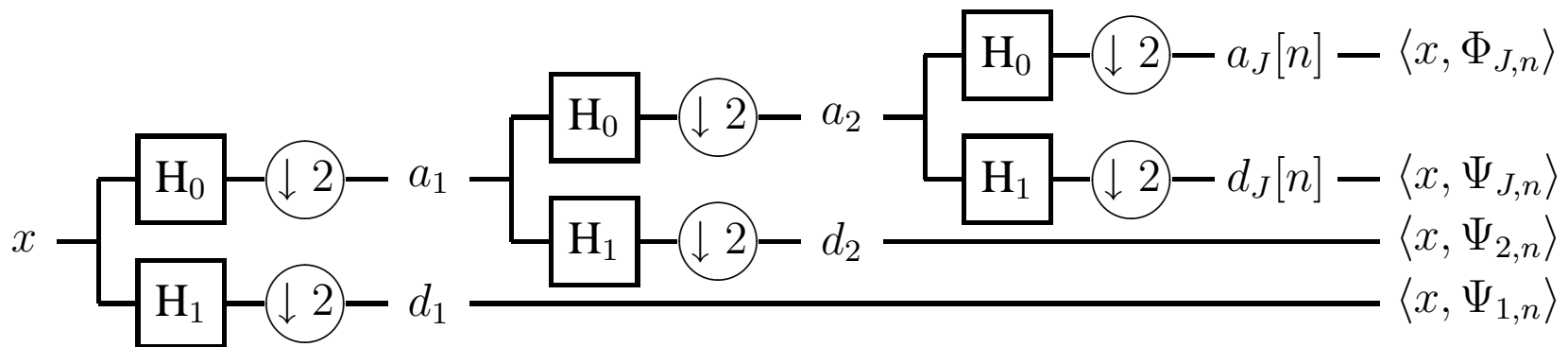


If we could treat the Earth like a **three-dimensional Cartesian** box, we would.

We'd take the **Haar** basis in depth and the **Daubechies** basis on the surface.

I The wavelet transform

Cartesian wavelets are “easy” — well understood, fast algorithms, and proper accounting of boundaries (wavelets “*on the interval*”).



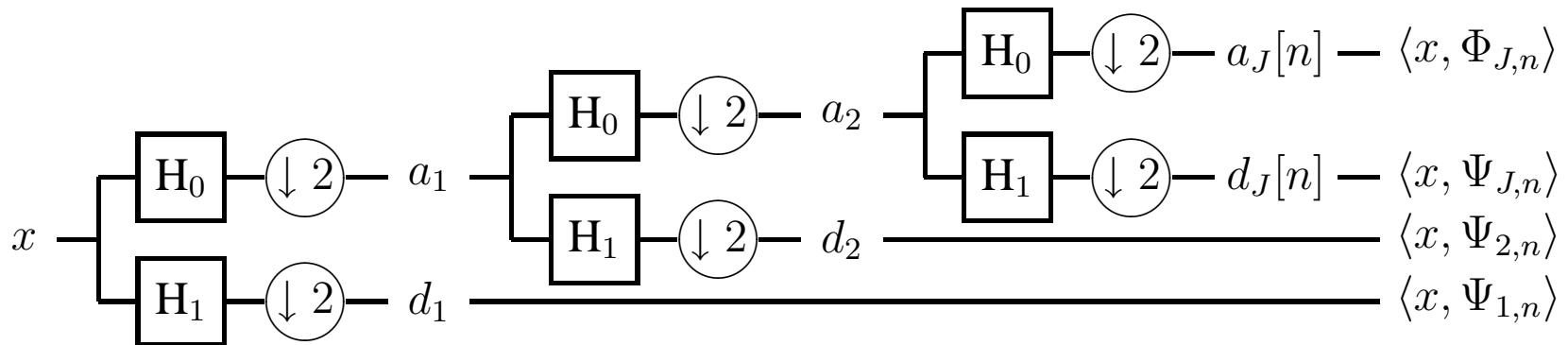
If we could treat the Earth like a **three-dimensional Cartesian** box, we would.

We'd take the **Haar** basis in depth and the **Daubechies** basis on the surface.

We'd account for the edges with **preconditioners** and special **boundary filters**.

I The wavelet transform

Cartesian wavelets are “easy” — well understood, fast algorithms, and proper accounting of boundaries (wavelets “*on the interval*”).



If we could treat the Earth like a **three-dimensional Cartesian** box, we would.

We'd take the **Haar** basis in depth and the **Daubechies** basis on the surface.

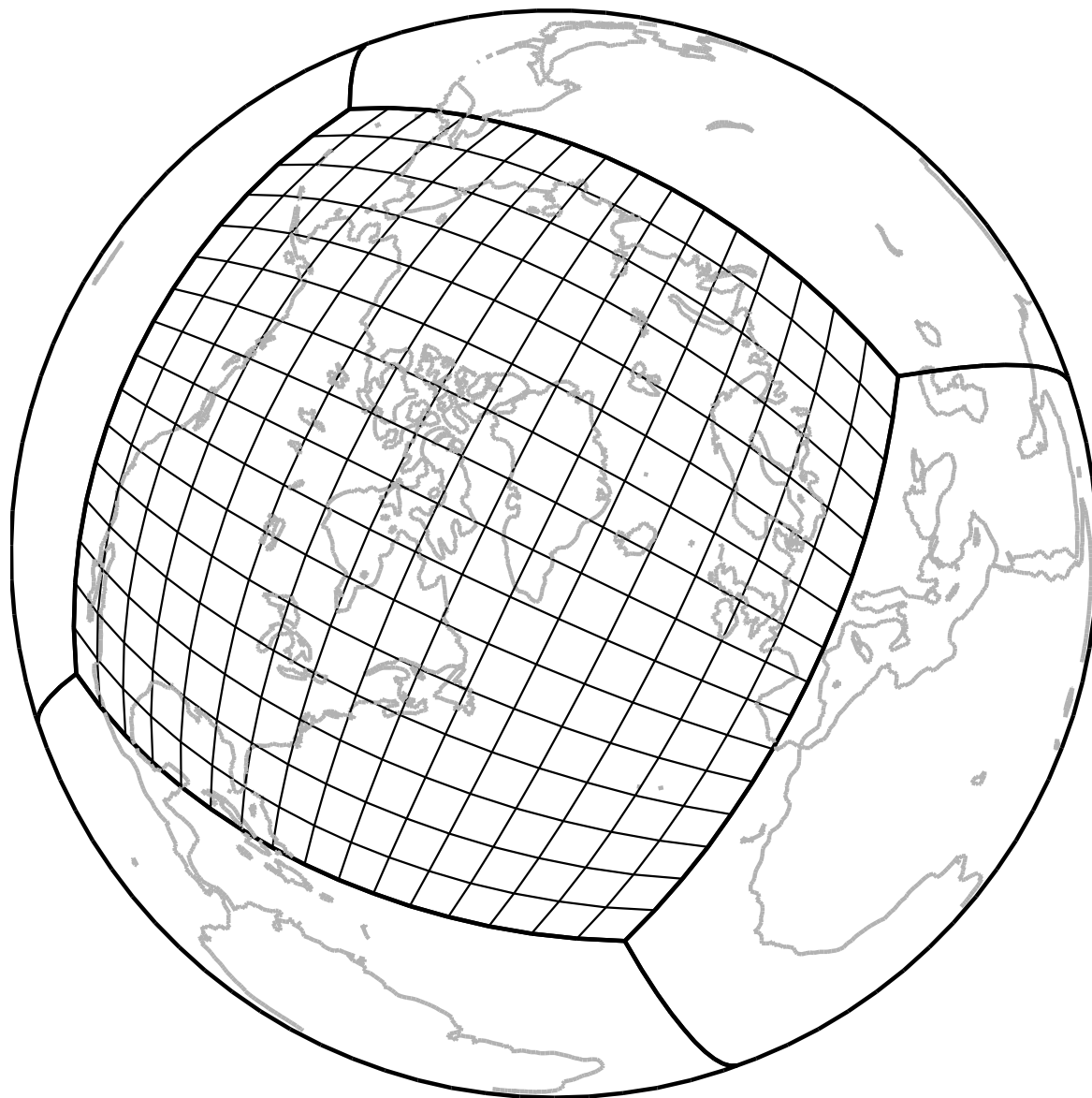
We'd account for the edges with **preconditioners** and special **boundary filters**.

And then we'd **map** the box onto a sphere — the **cubed sphere**.

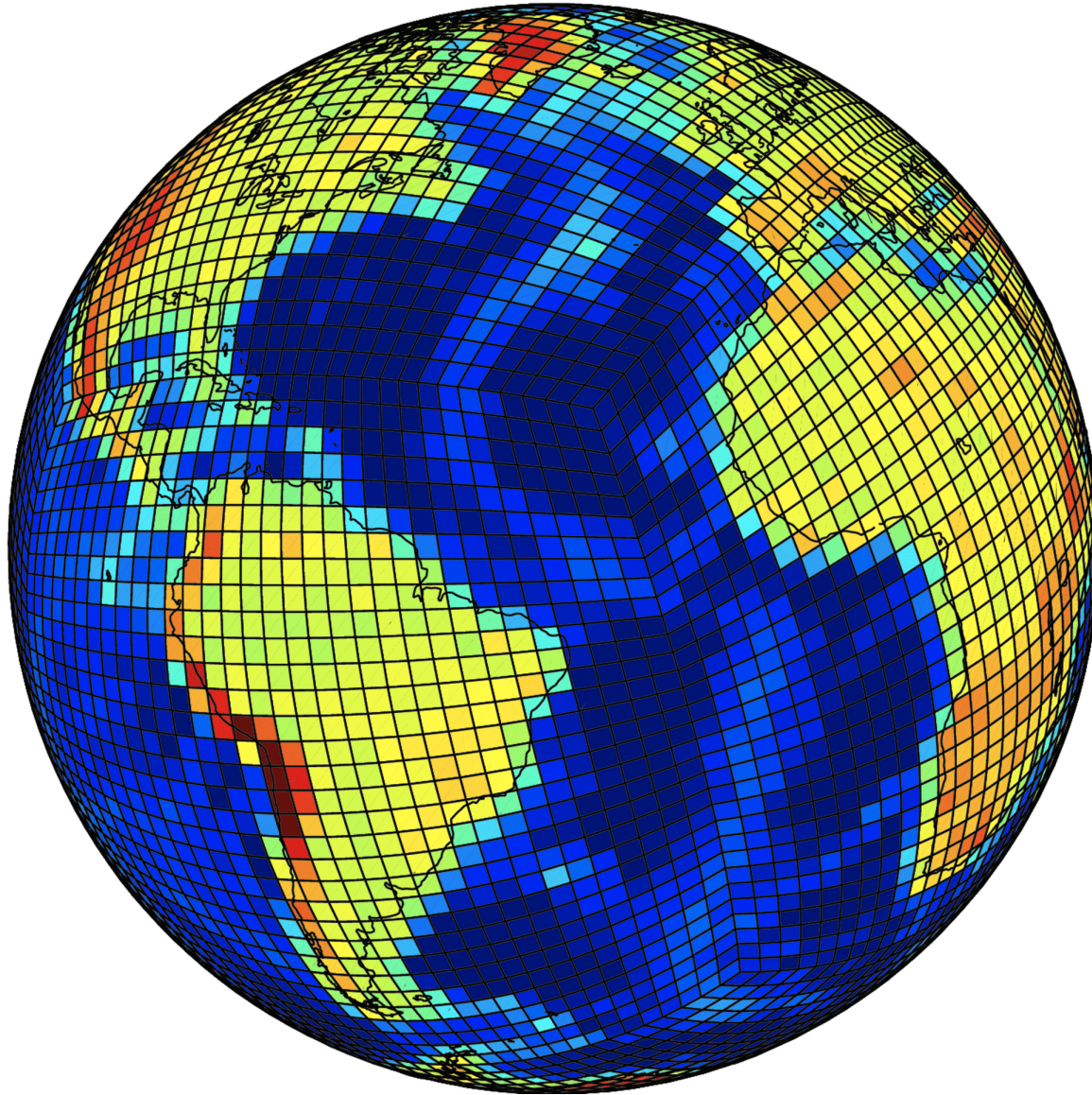
I The wavelet transform

Can we *adapt* the **wavelet transform**,
so powerful in **Cartesian** applications, to the **sphere**,
where it is known to be — “*complicated*”?

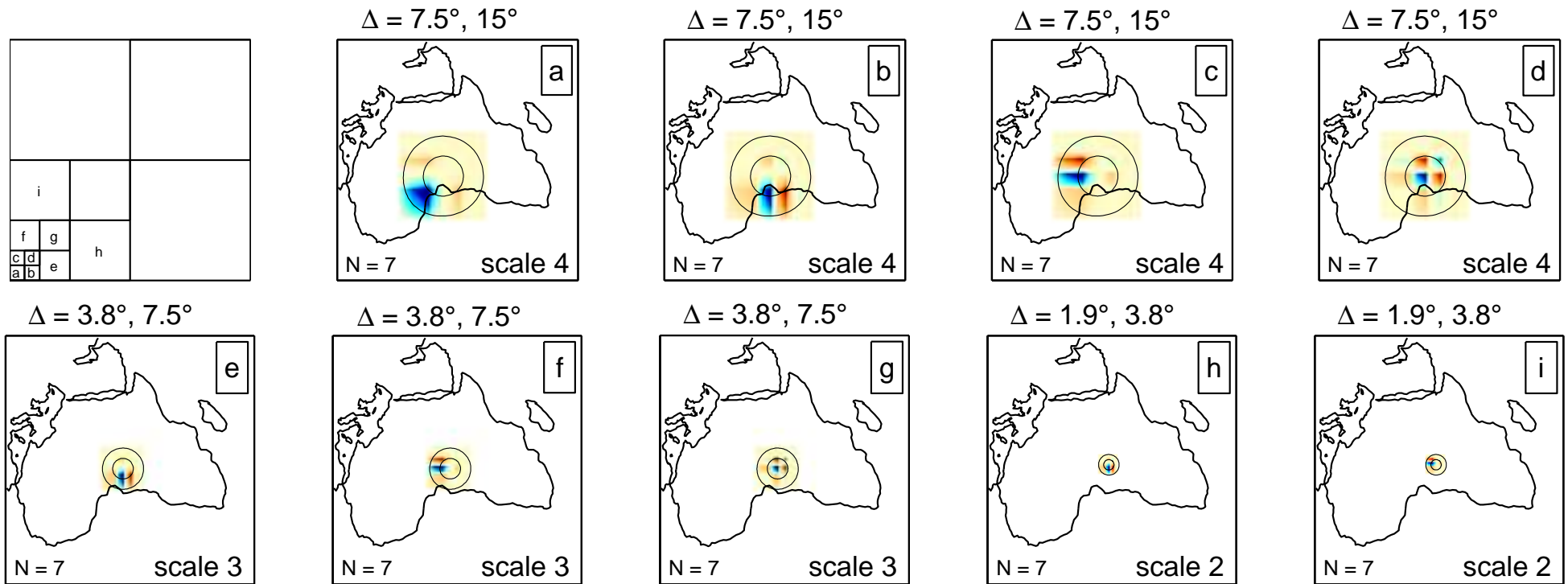
| The *cubed* sphere



| The *cubed* sphere

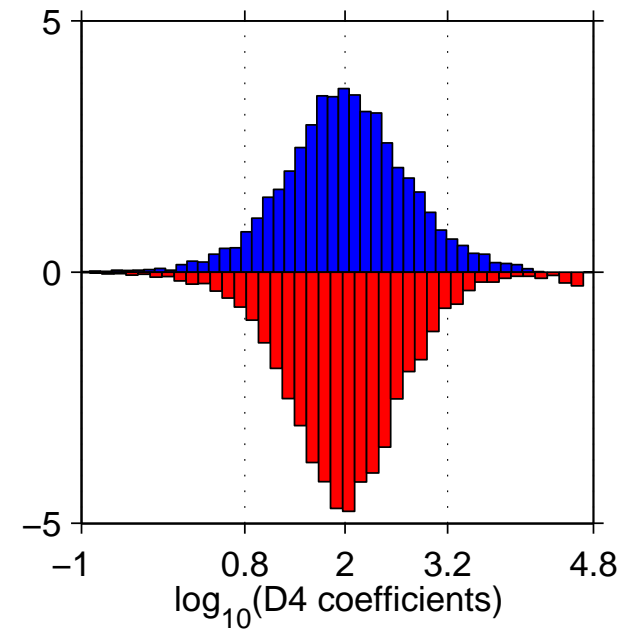
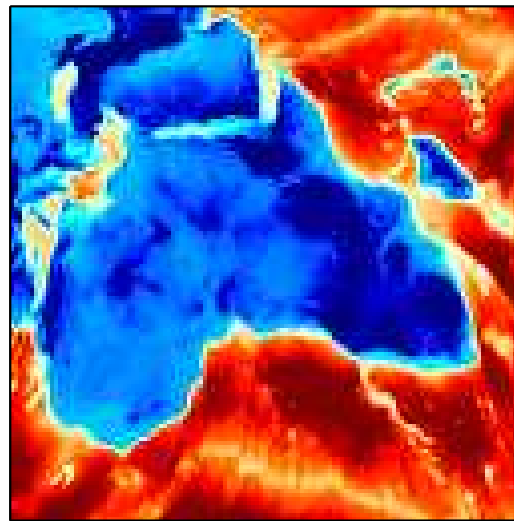
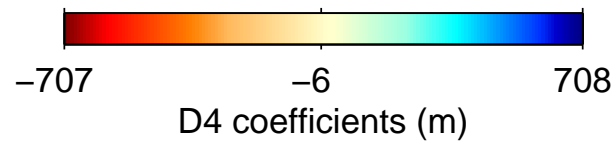
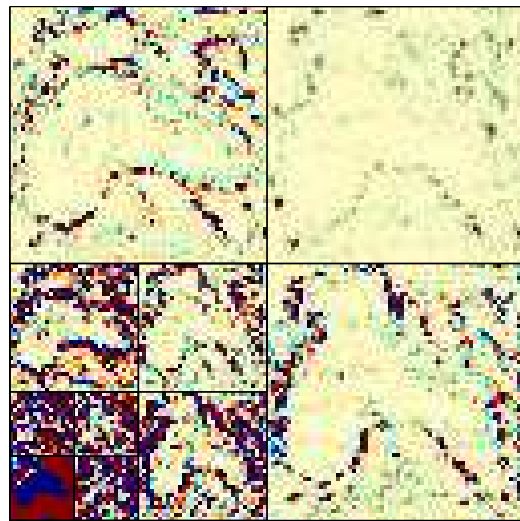


Cubed-spherical wavelets



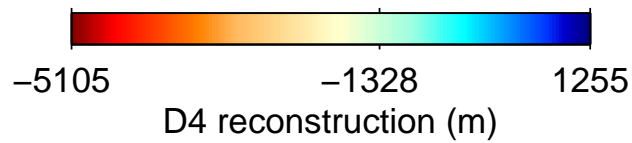
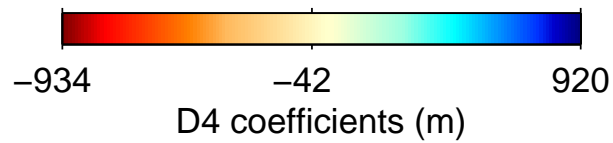
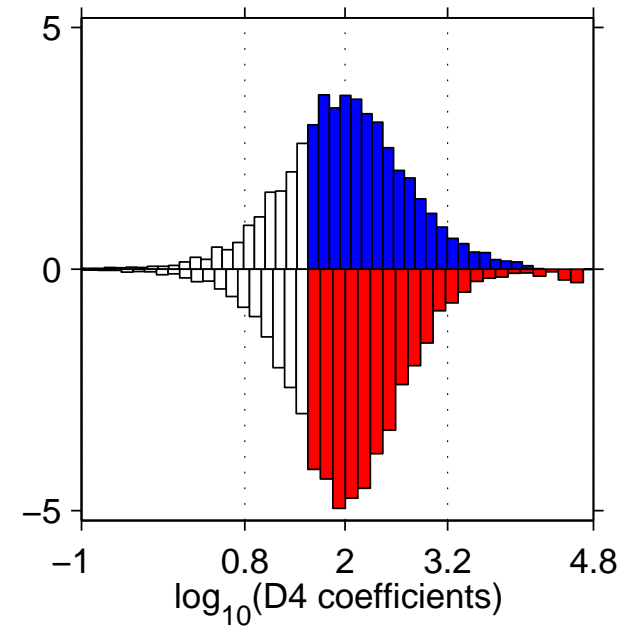
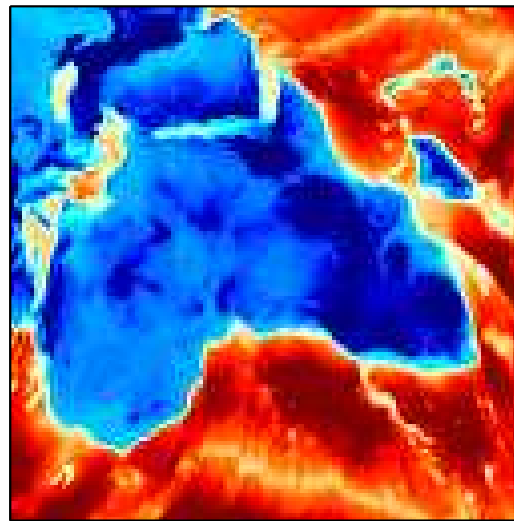
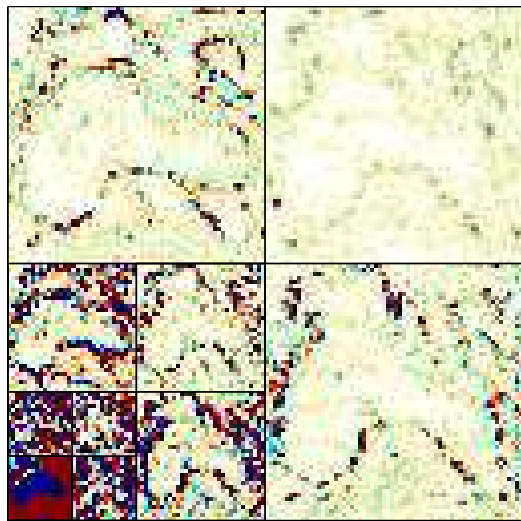
| A sparsifying transform

rmse 0.0% | 0%ile | 0% zero



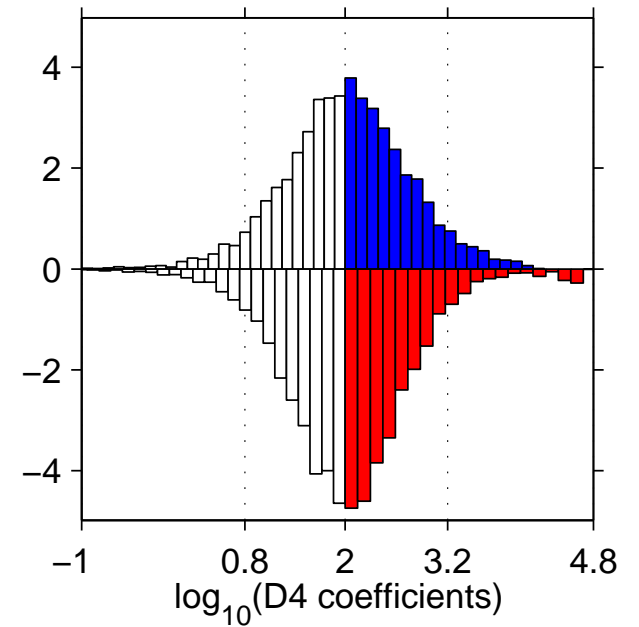
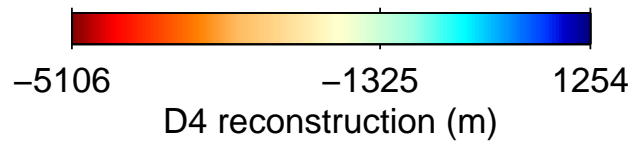
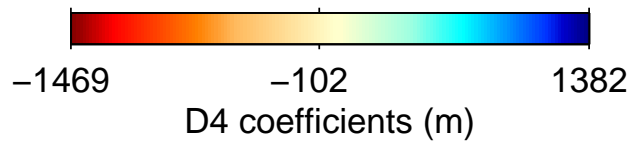
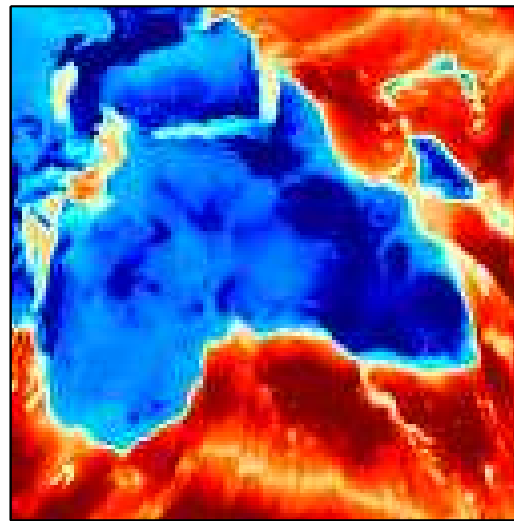
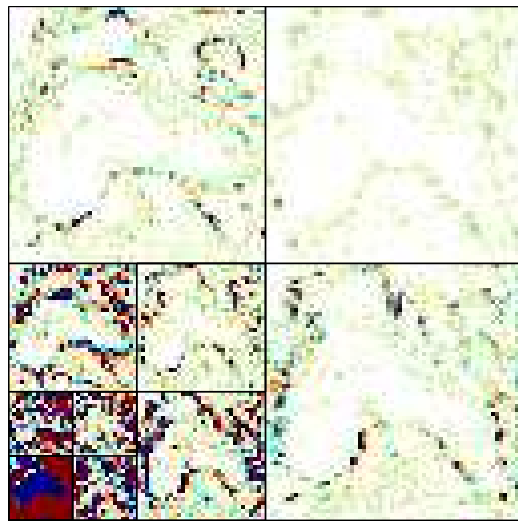
| A sparsifying transform

rmse 0.4% | 25%ile | 25% zero



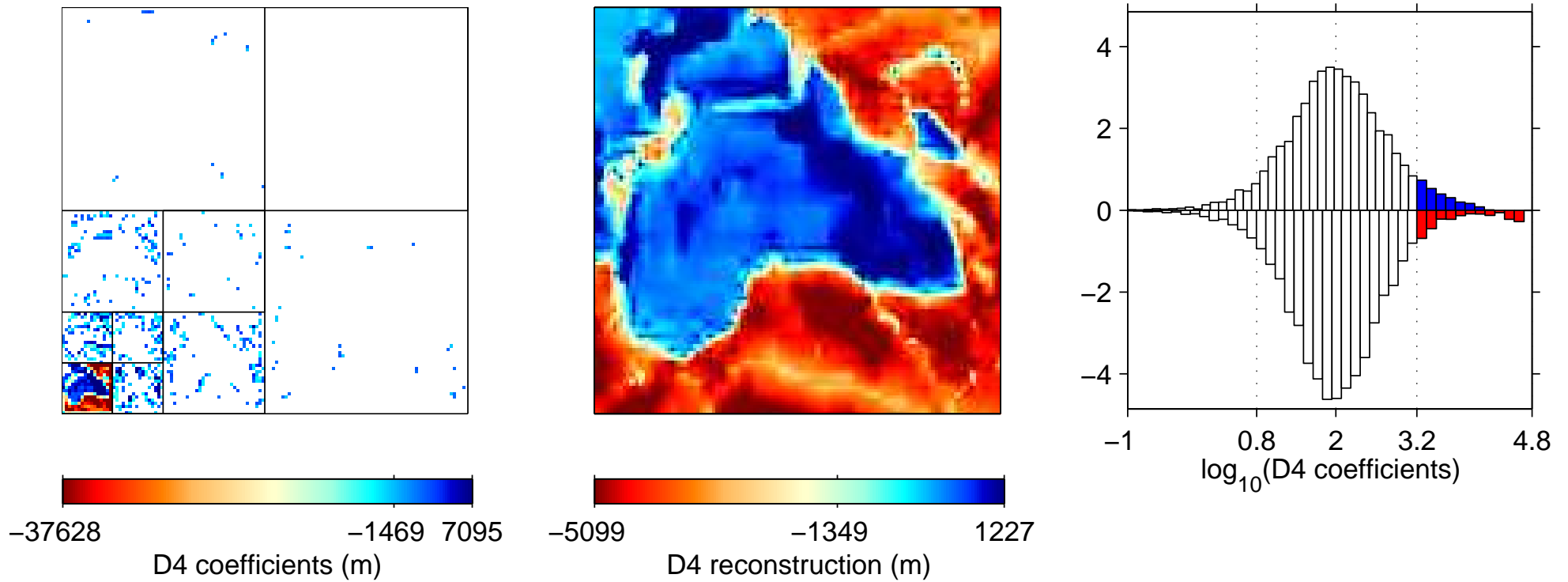
| A sparsifying transform

rmse 1.2% | 50%ile | 50% zero



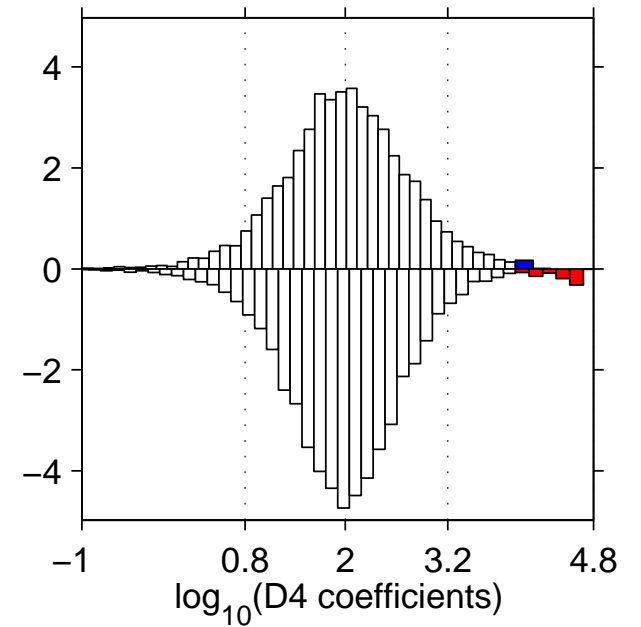
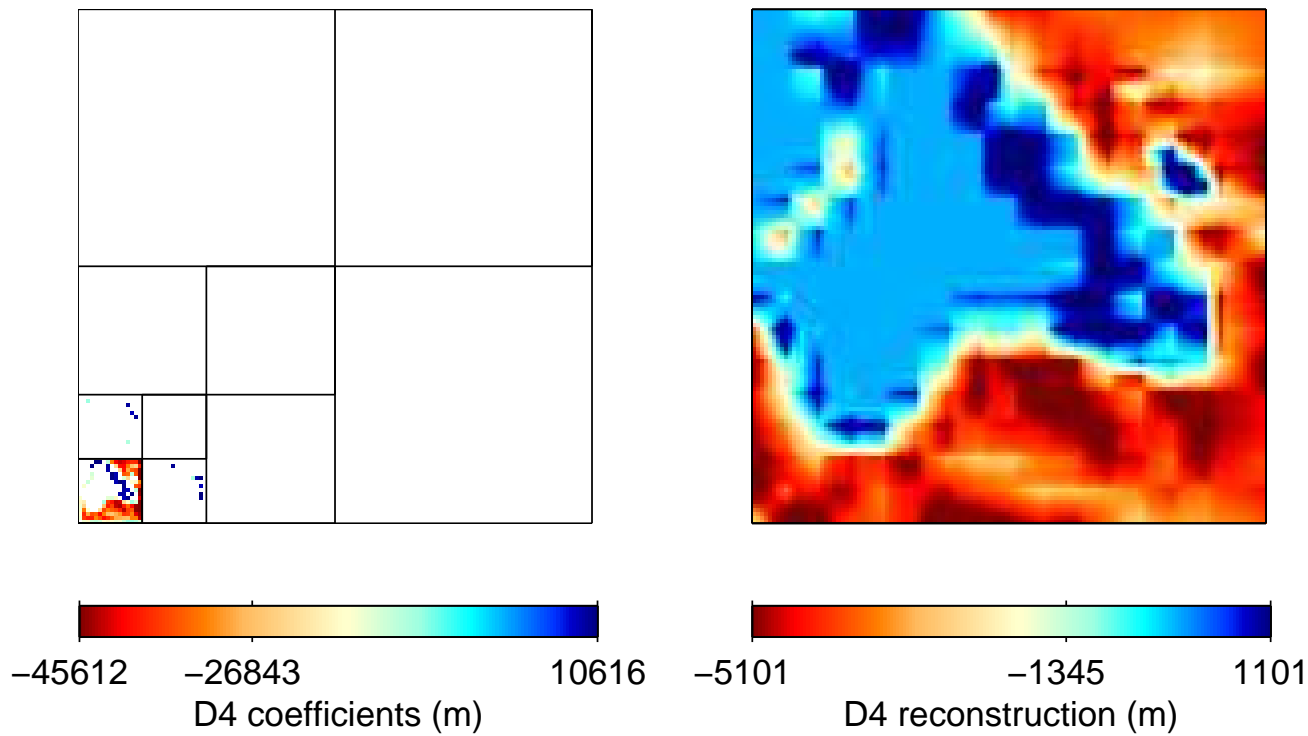
| A sparsifying transform

rmse 9.9% | 95%ile | 95% zero



A sparsifying transform

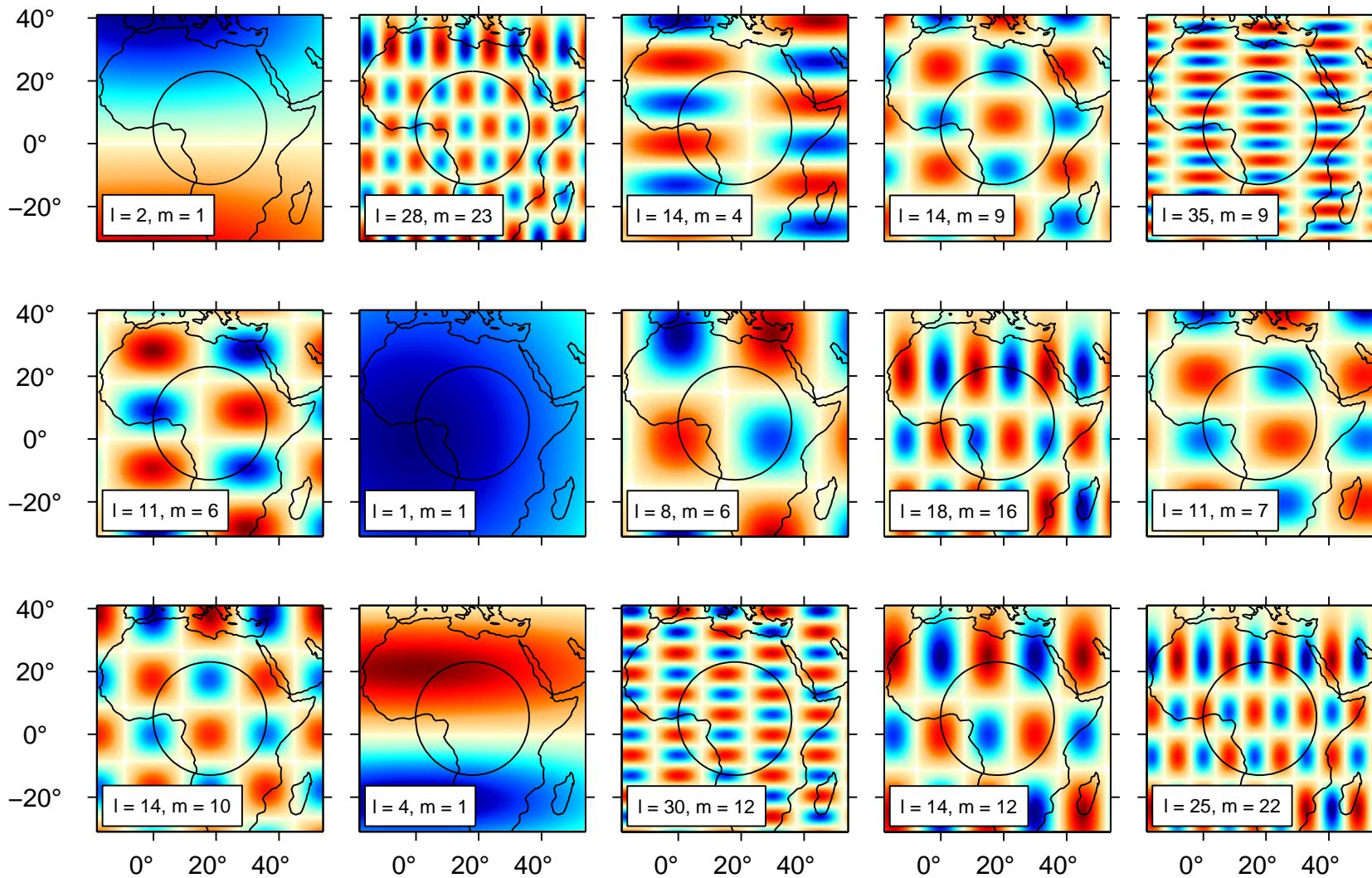
rmse 24.5% | 99%ile | 99% zero



I Applications

So far, mostly for *inverse problems* in **seismology**. But the topographic example shows there is much promise for the study of planetary **surfaces** also.

Spherical harmonics

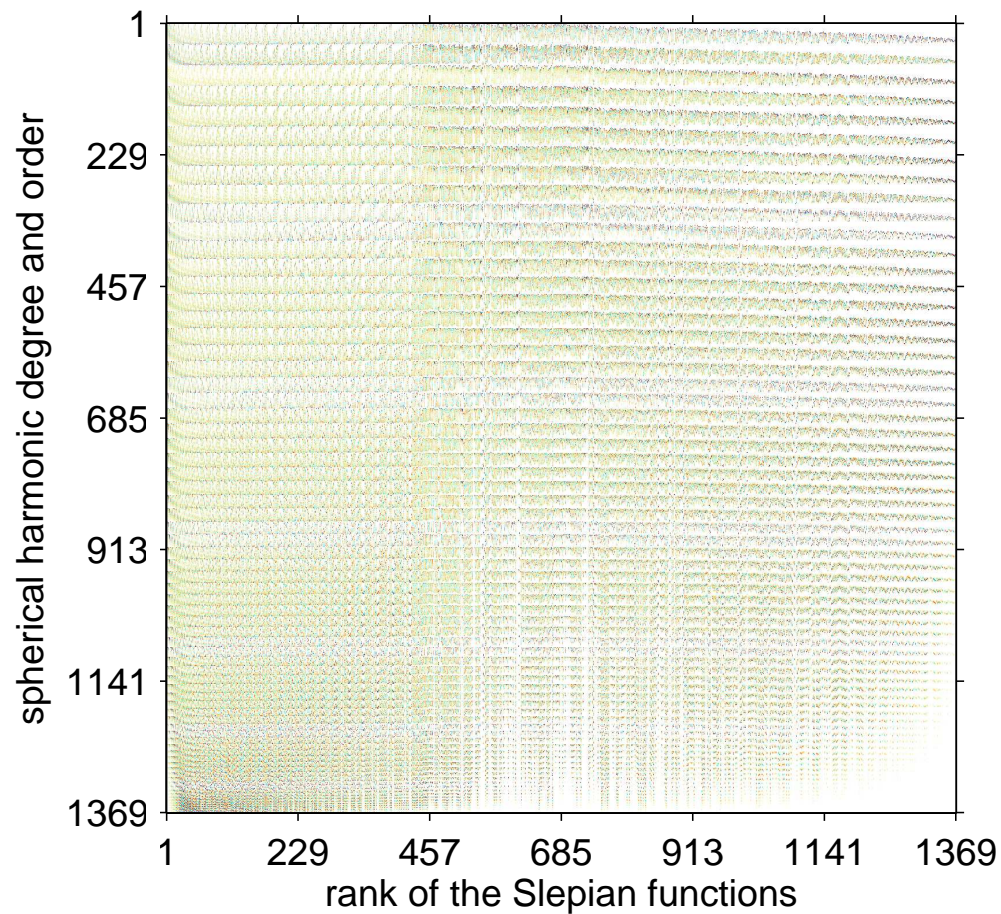


III The Slepian transform

Can we *combine* **spherical harmonics**, which normally spread their energy over the **whole globe**, into Slepian functions — for a *particular region*?

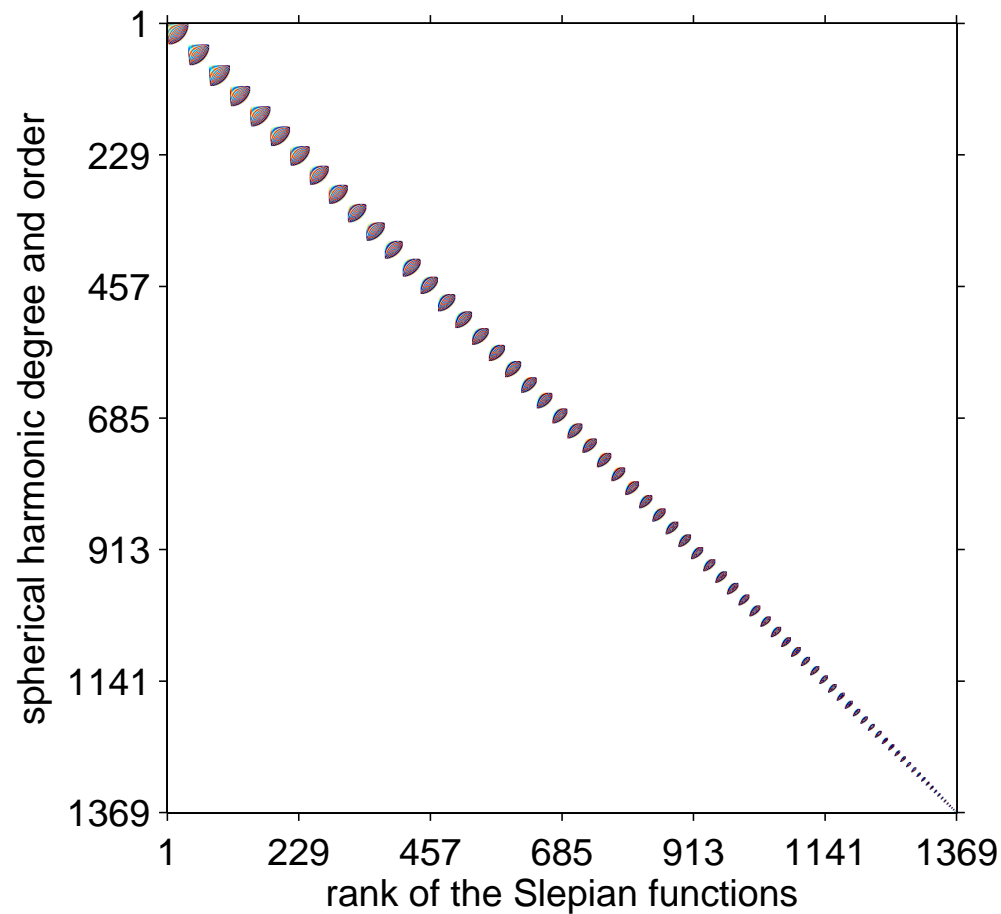
II \rightarrow III Spherical harmonics \rightarrow Slepian functions

An optimal **orthogonal transform** introduces sparsity. Here's two looks at it:



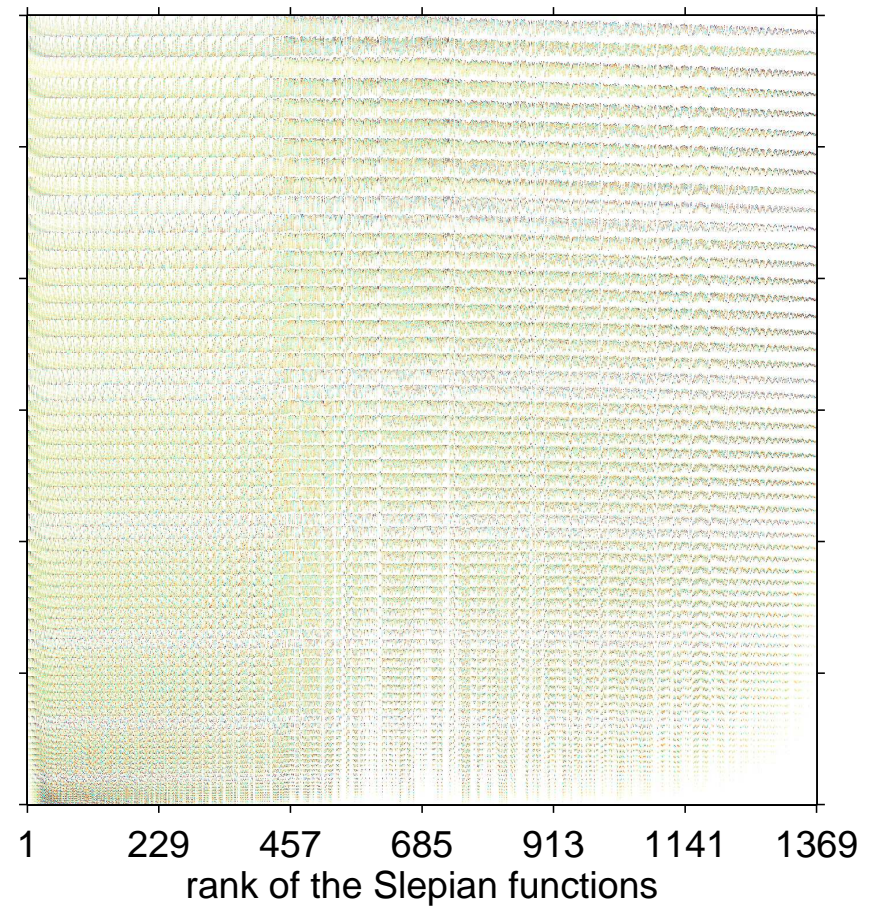
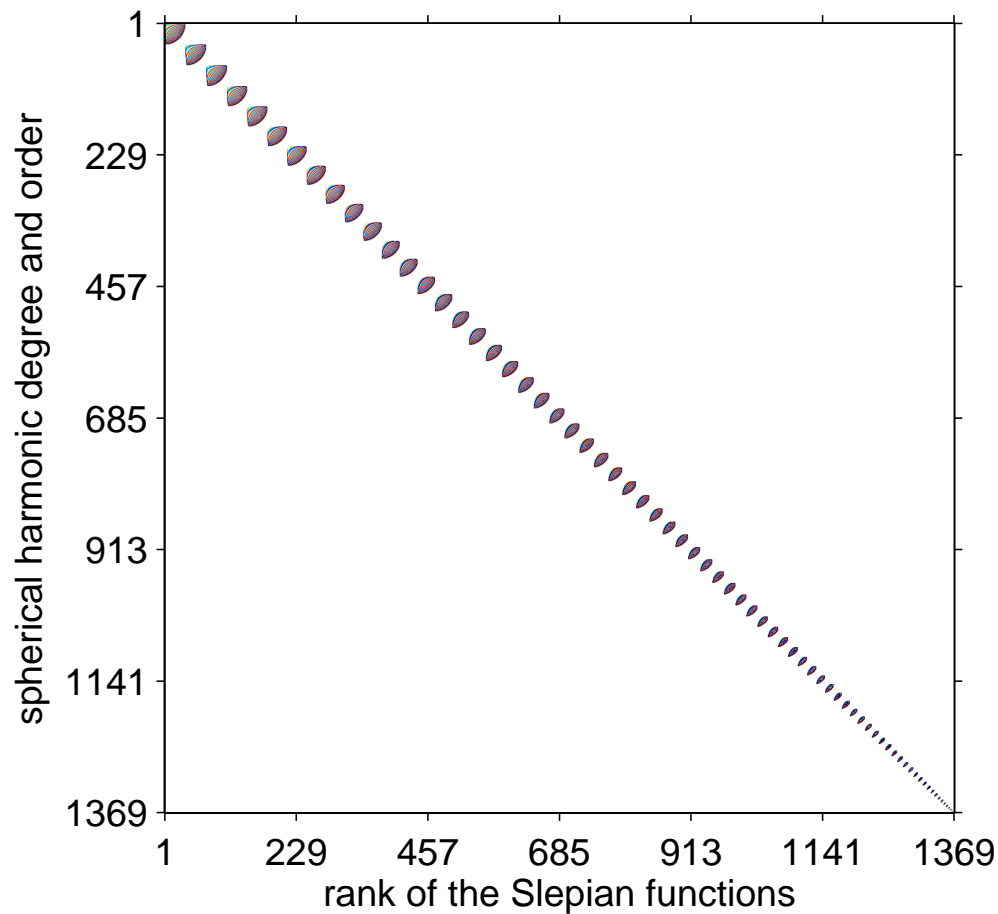
II \rightarrow III Spherical harmonics \rightarrow Slepian functions

An optimal **orthogonal transform** introduces sparsity. Here's two looks at it:

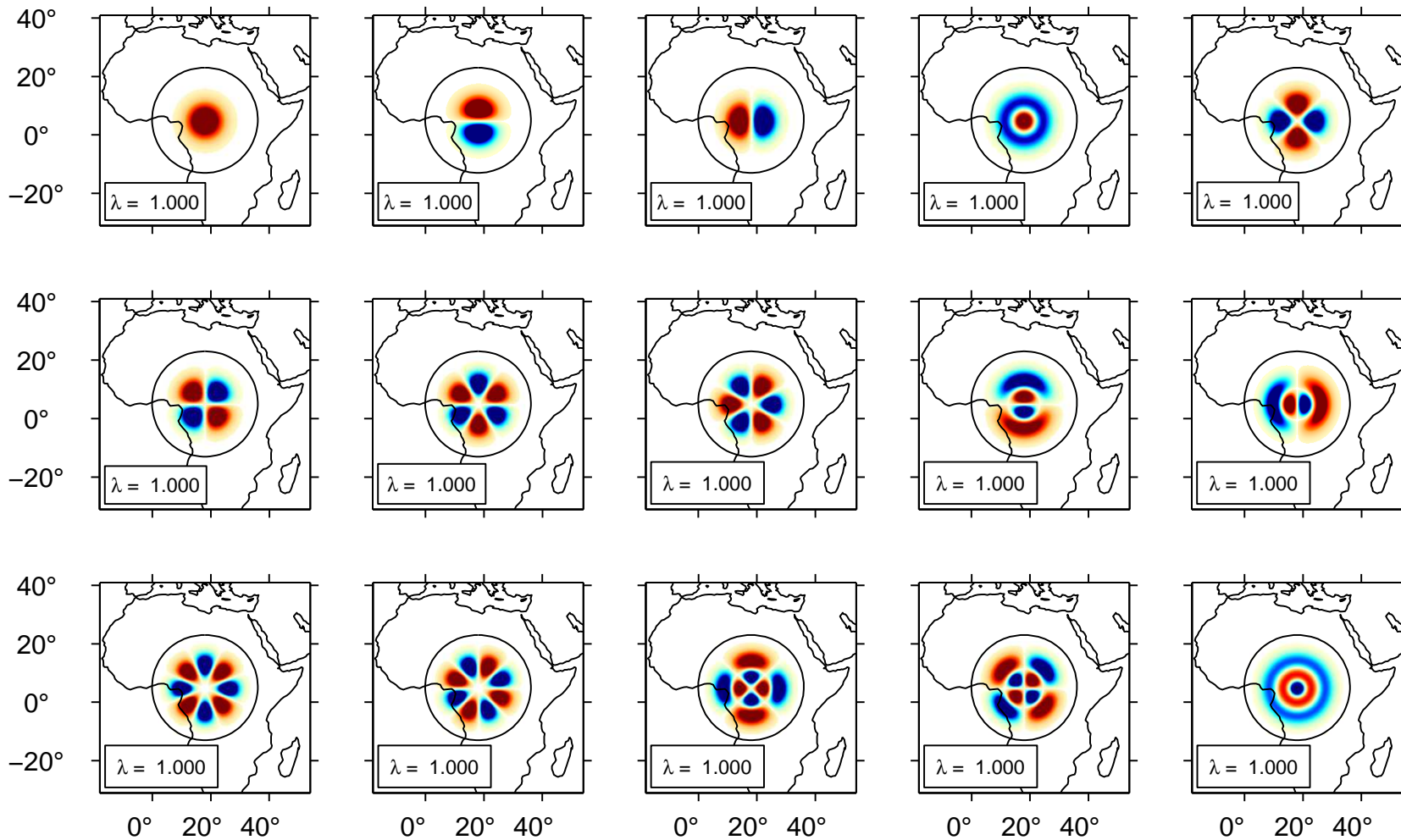


II \rightarrow III Spherical harmonics \rightarrow Slepian functions

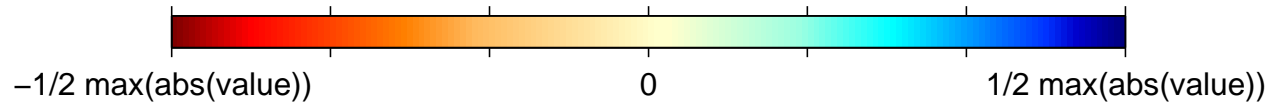
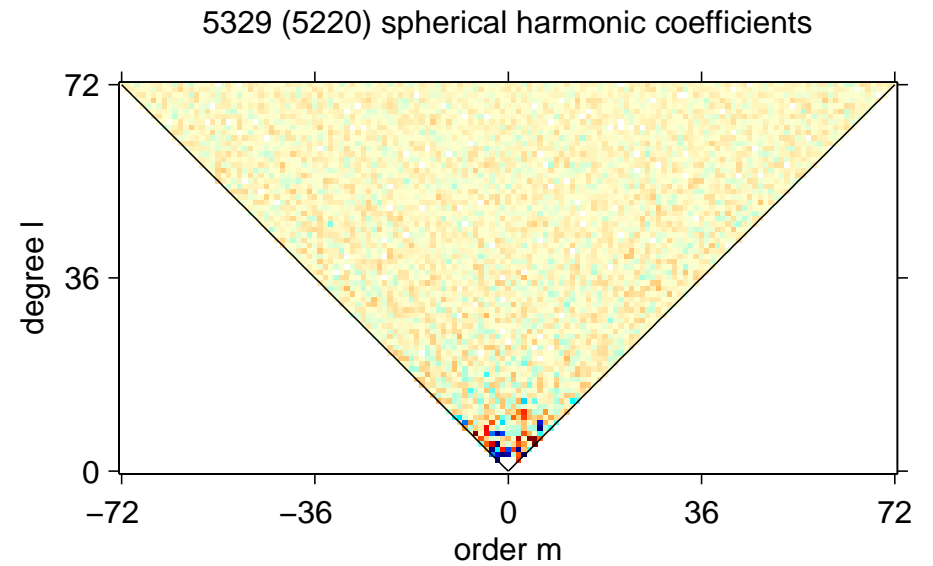
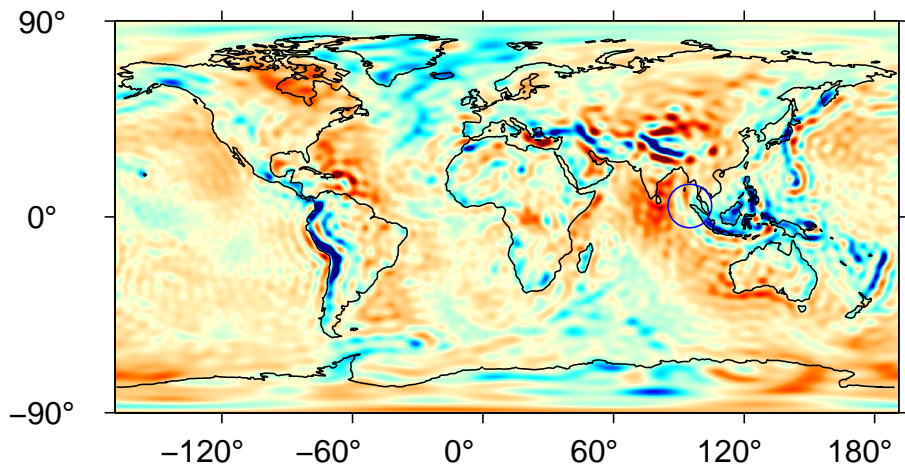
An optimal **orthogonal transform** introduces sparsity. Here's two looks at it:



III Slepian functions

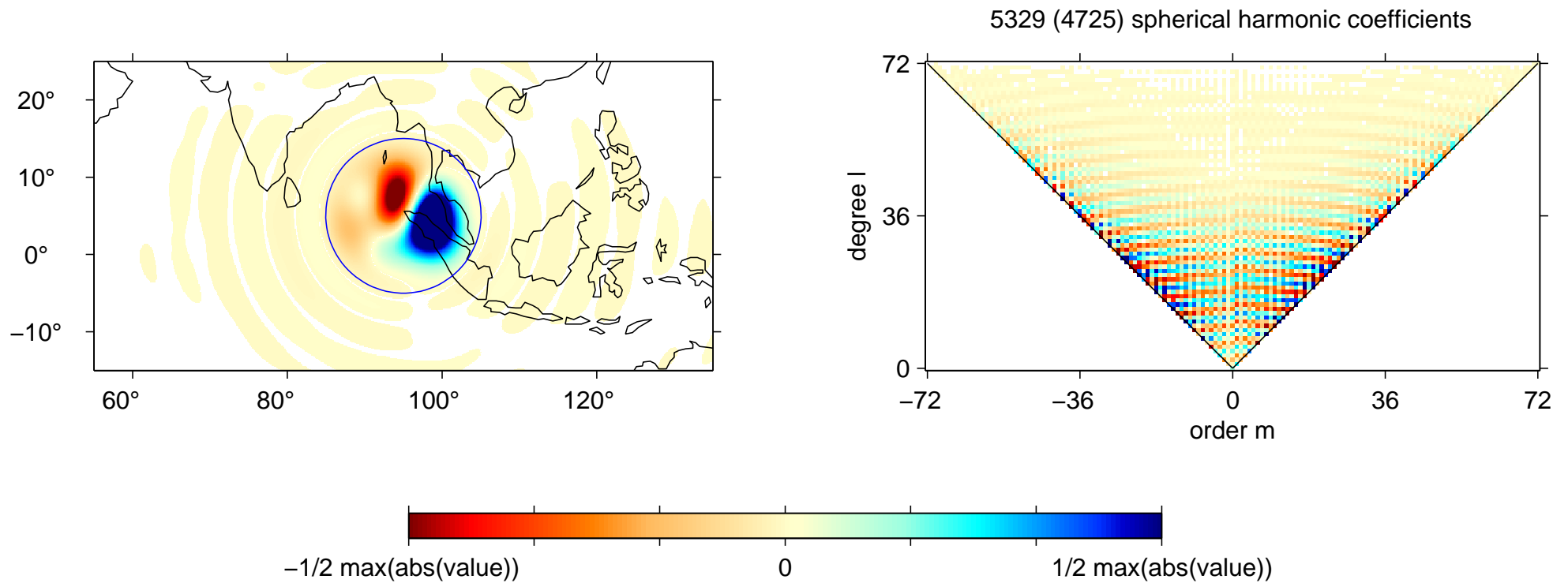


II Spherical harmonics



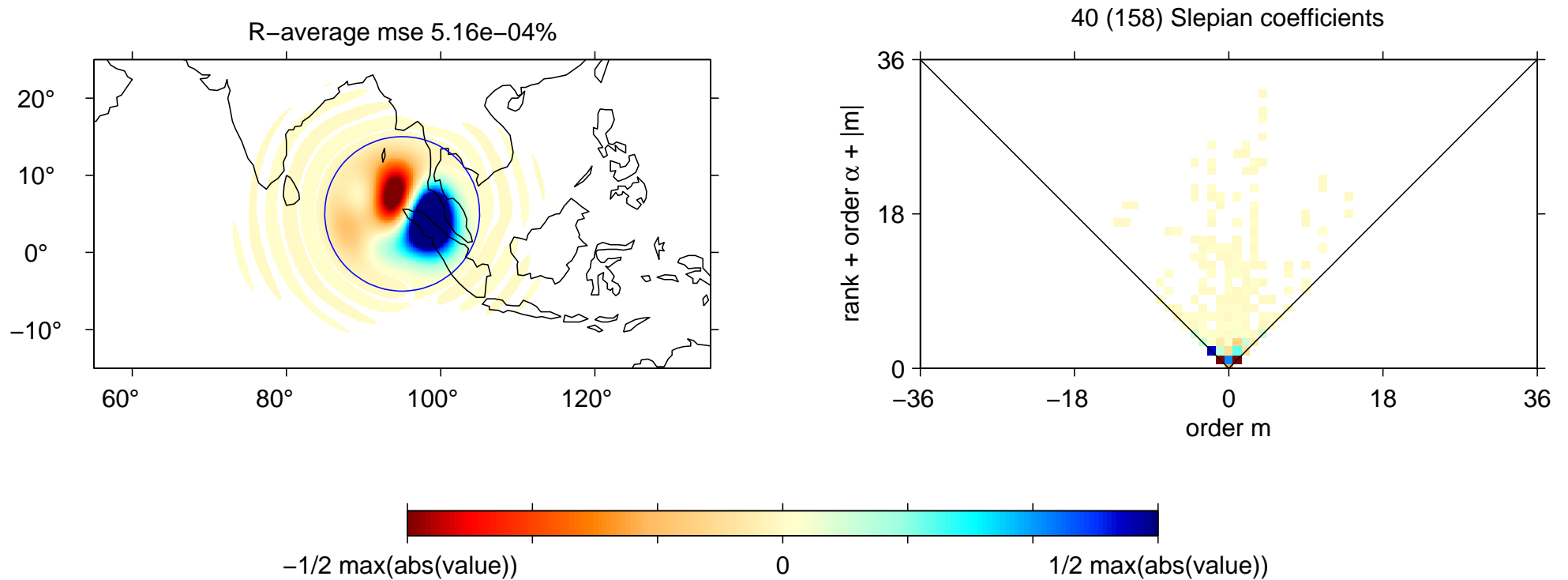
A *global* basis, **good** for *global* problems.

II Spherical harmonics



A *global* basis, **bad** for *local* problems.

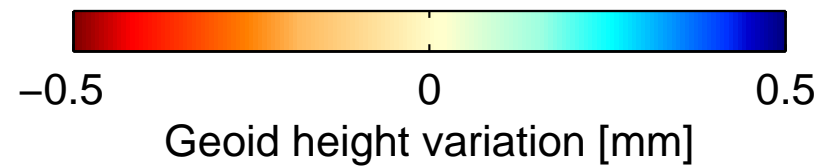
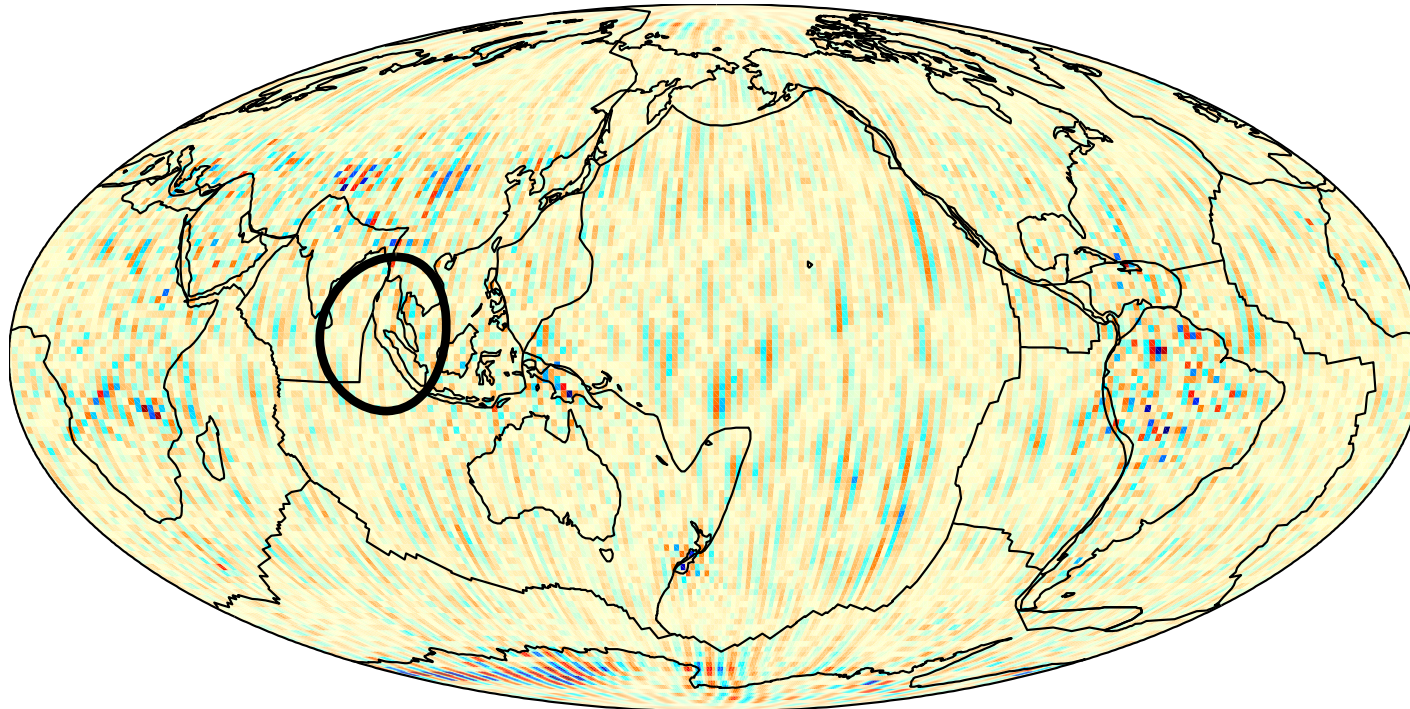
III Slepian functions



A *local* basis, **good** for *local* problems. Sparsity!

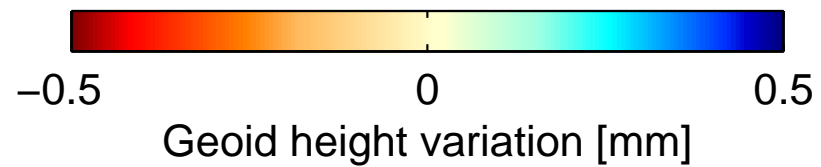
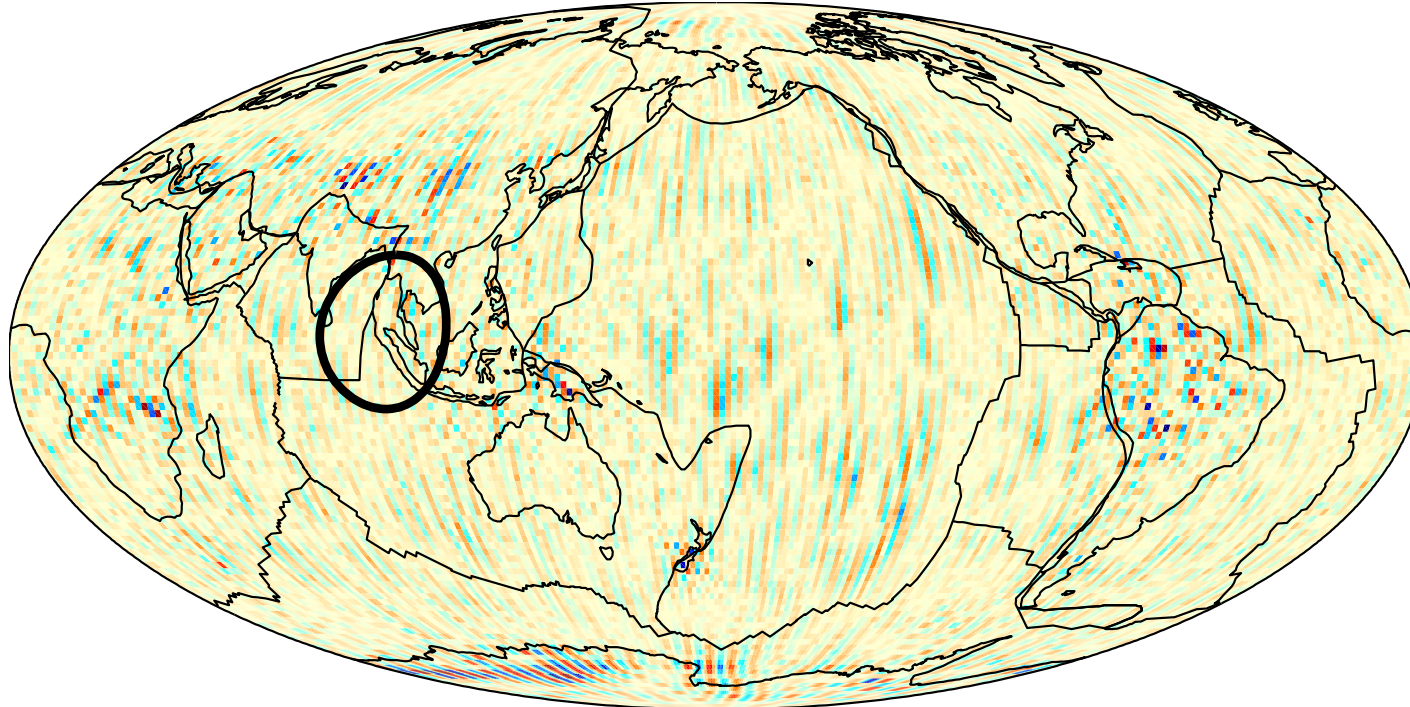
II Application to the analysis of GRACE

dGSM.2005.001.2005.031.0K20.geo



II Application to the analysis of GRACE

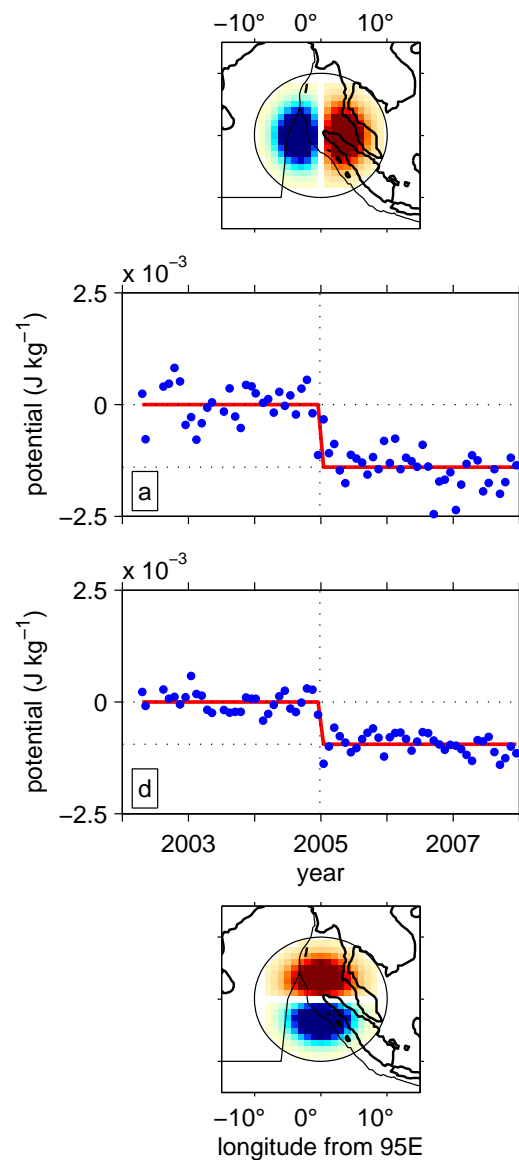
dGSM.2004.336.2004.366.0K20.geo



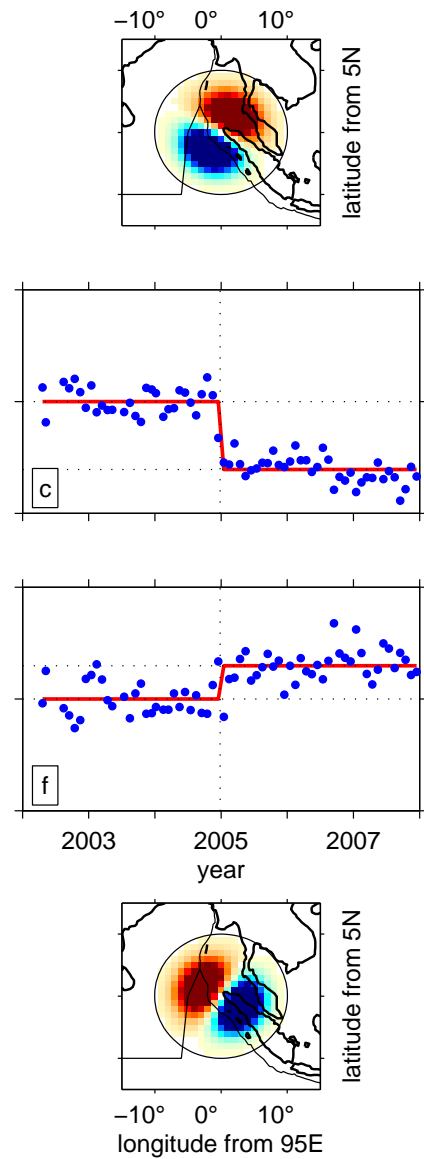
II → III Harmonics → Slepian is *sparsifying*

Expand the GRACE-derived geopotential into Slepian functions — designed for a region around *Sumatra*. The next plot shows just *two of the terms* in the expansion.

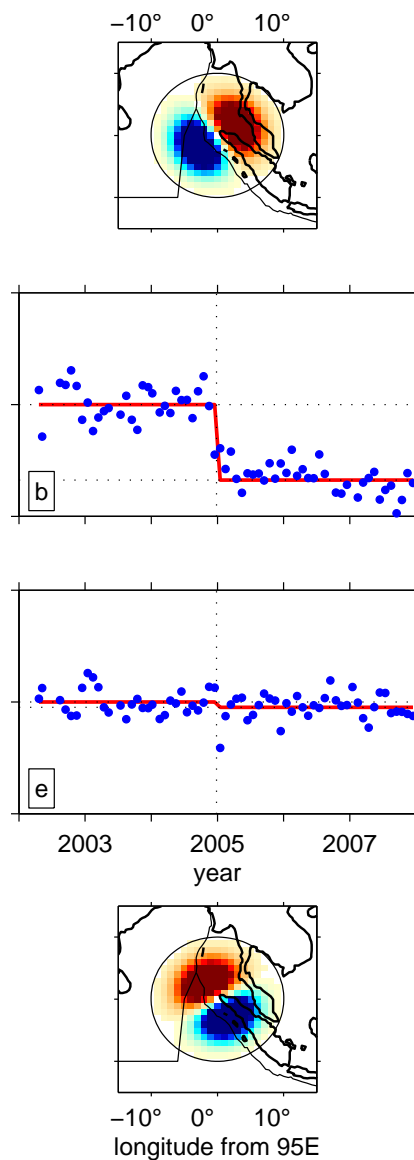
III Application to the analysis of GRACE



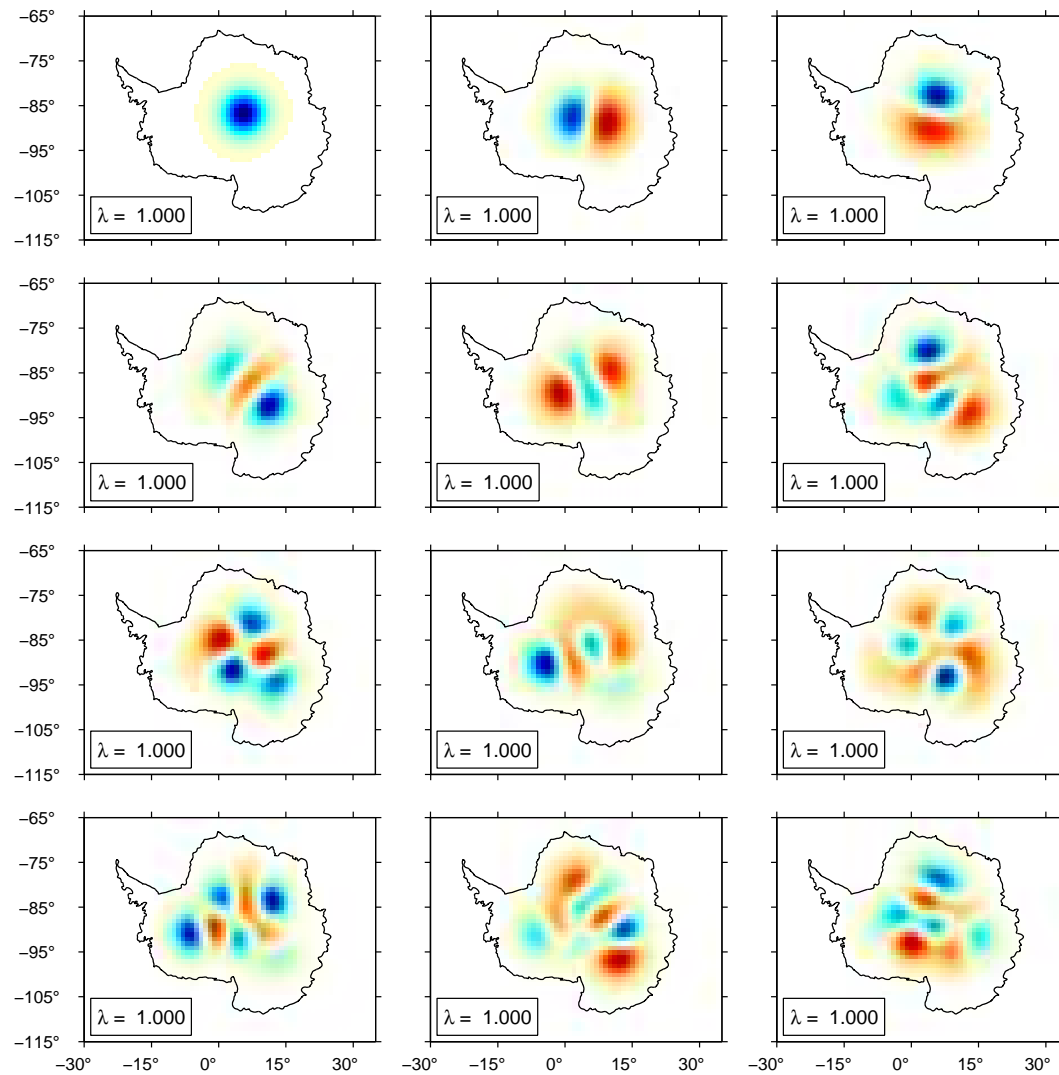
III Application to the analysis of GRACE



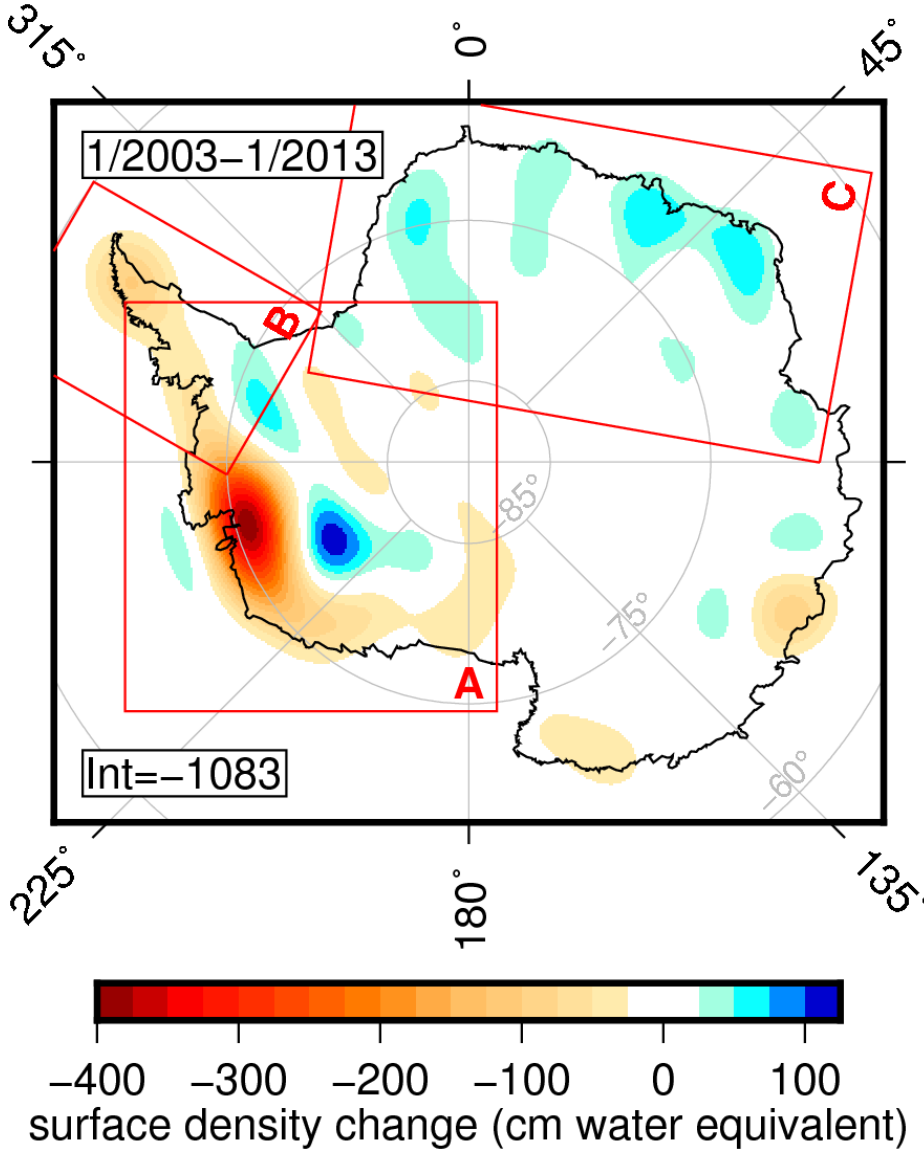
III Application to the analysis of GRACE



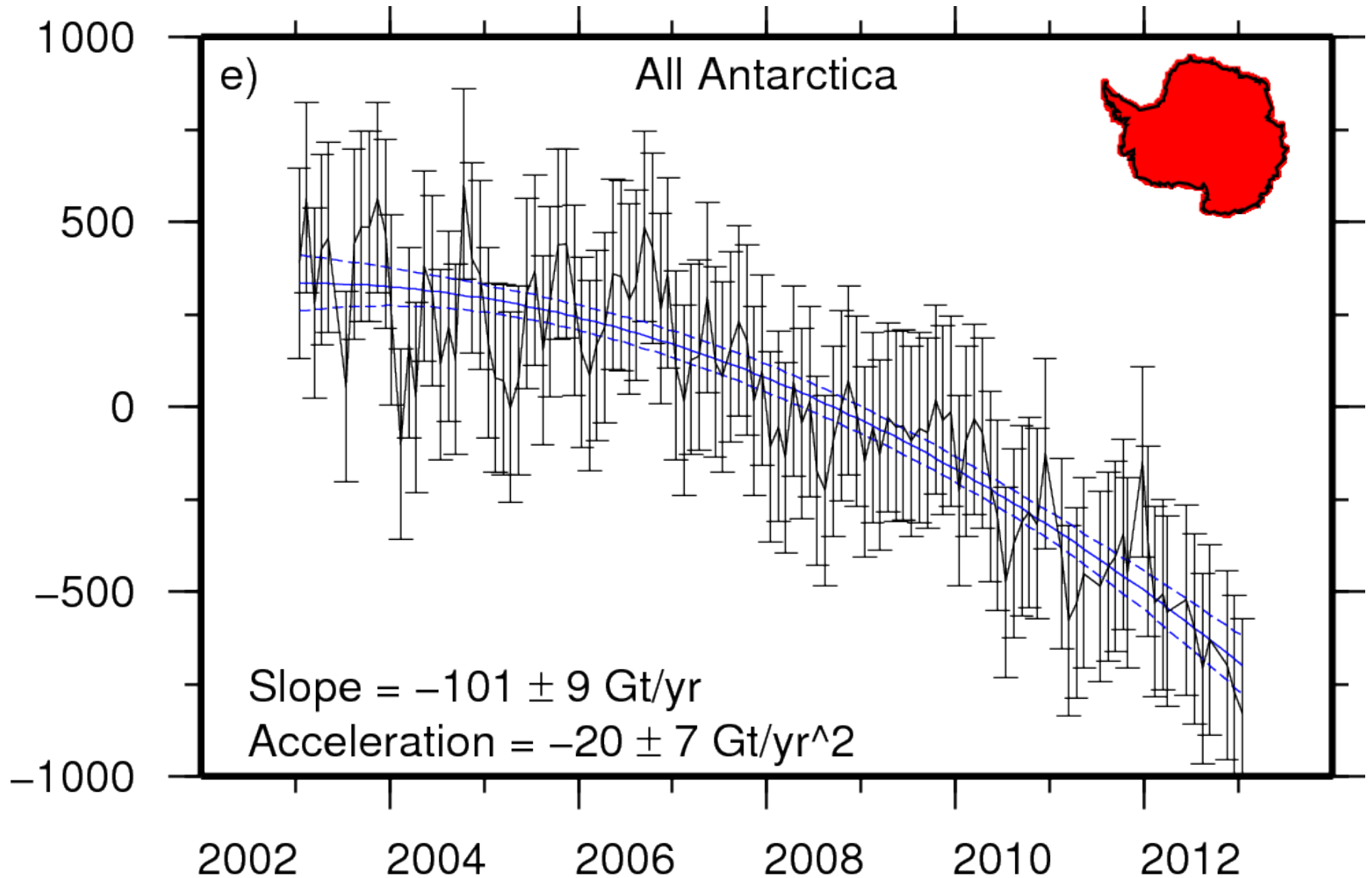
Slepian functions for arbitrary regions



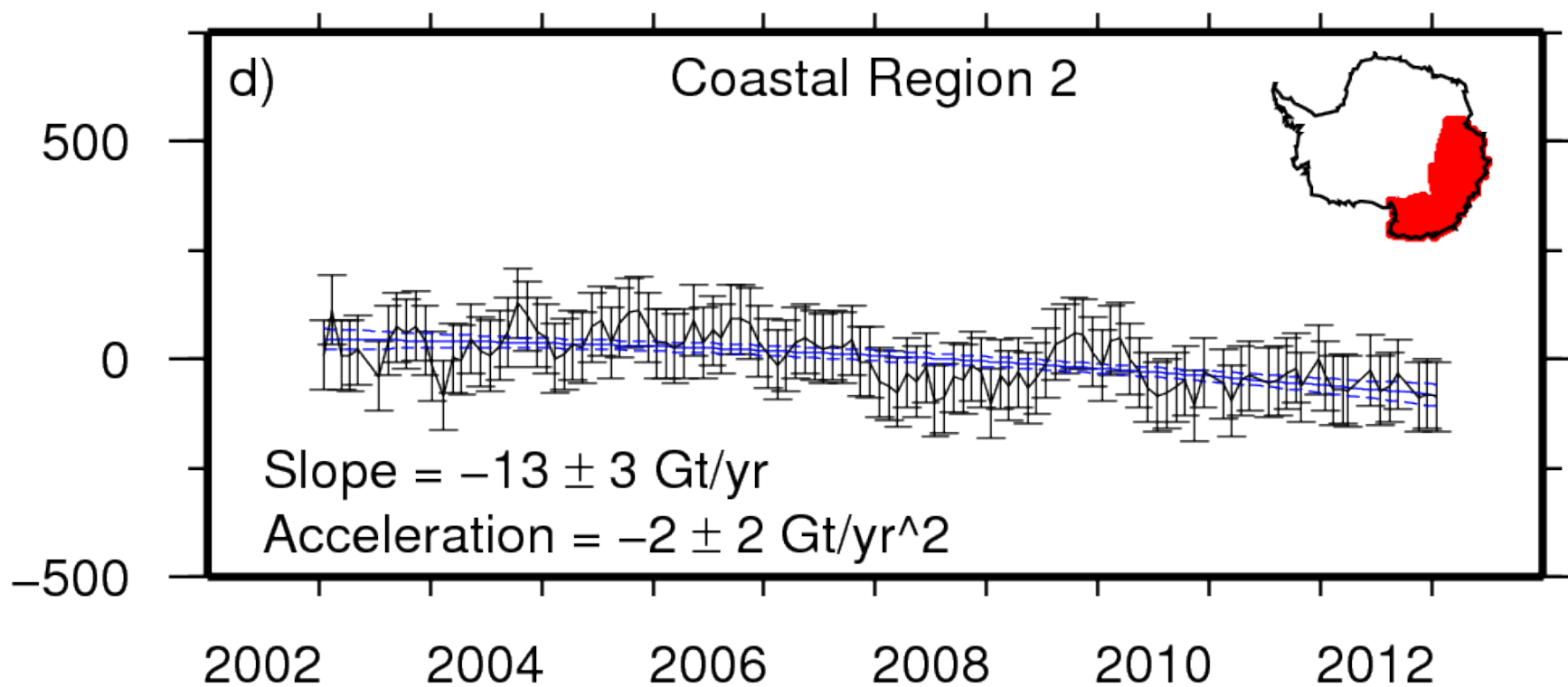
Whither Antarctica?



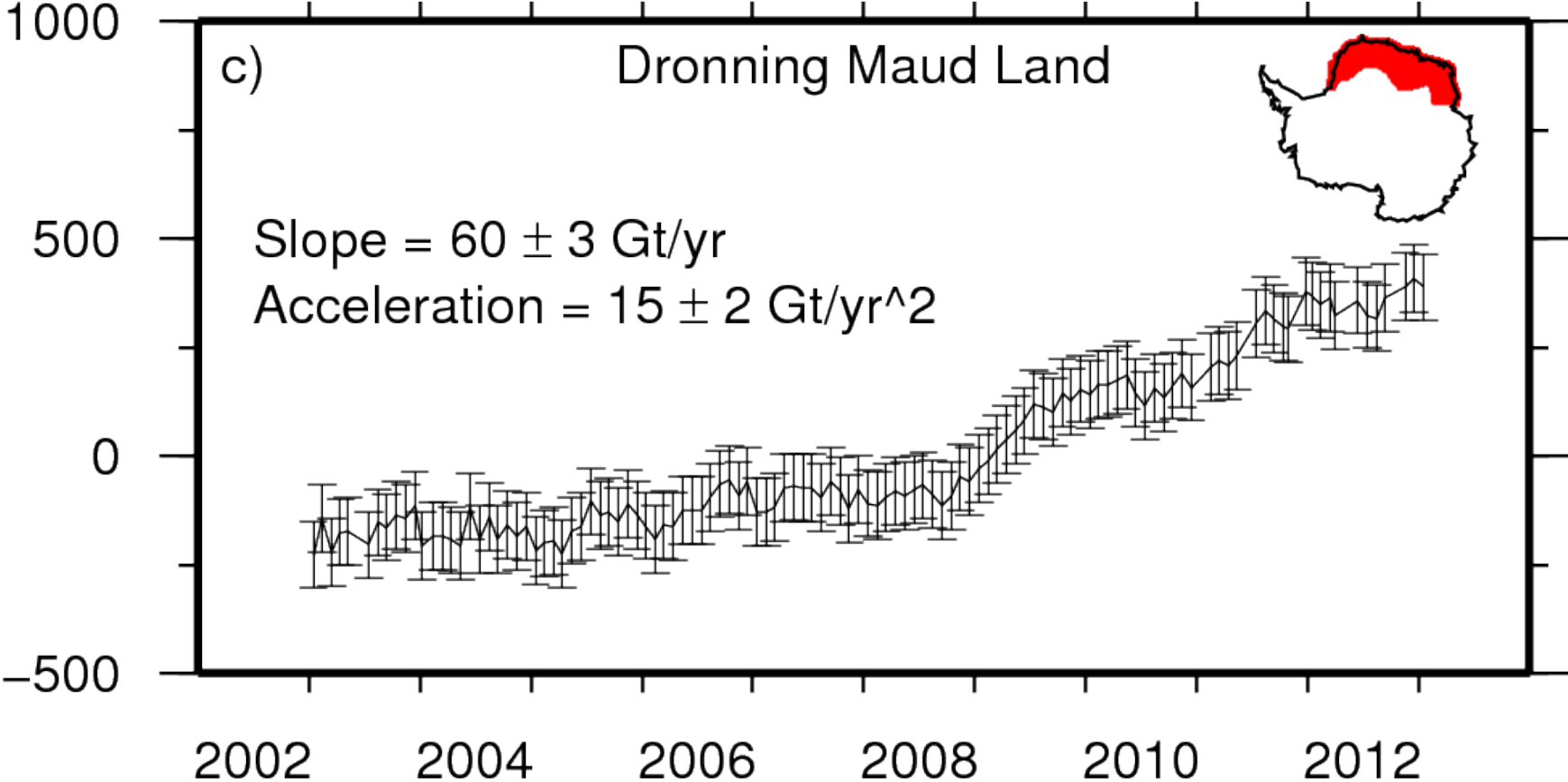
Ice mass loss 2002–2013



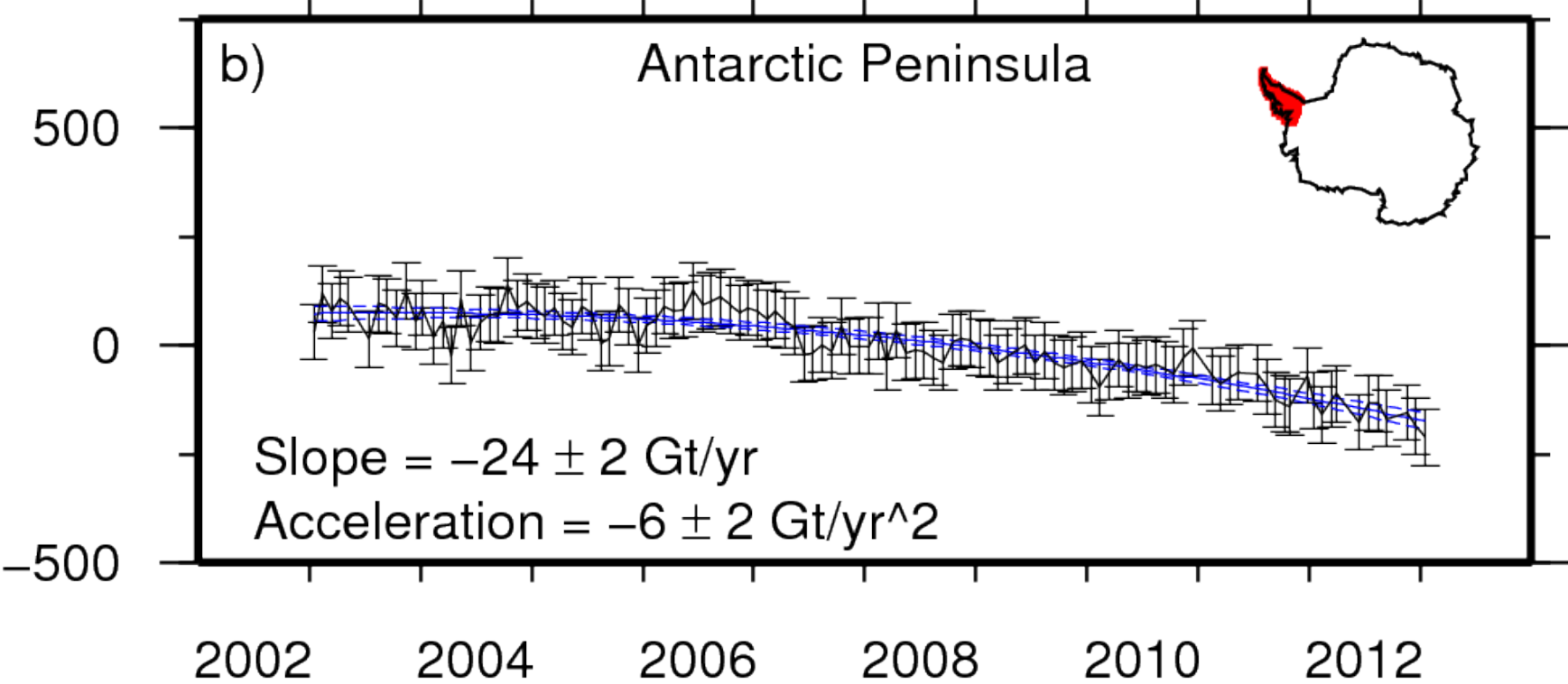
Ice mass loss 2002–2013



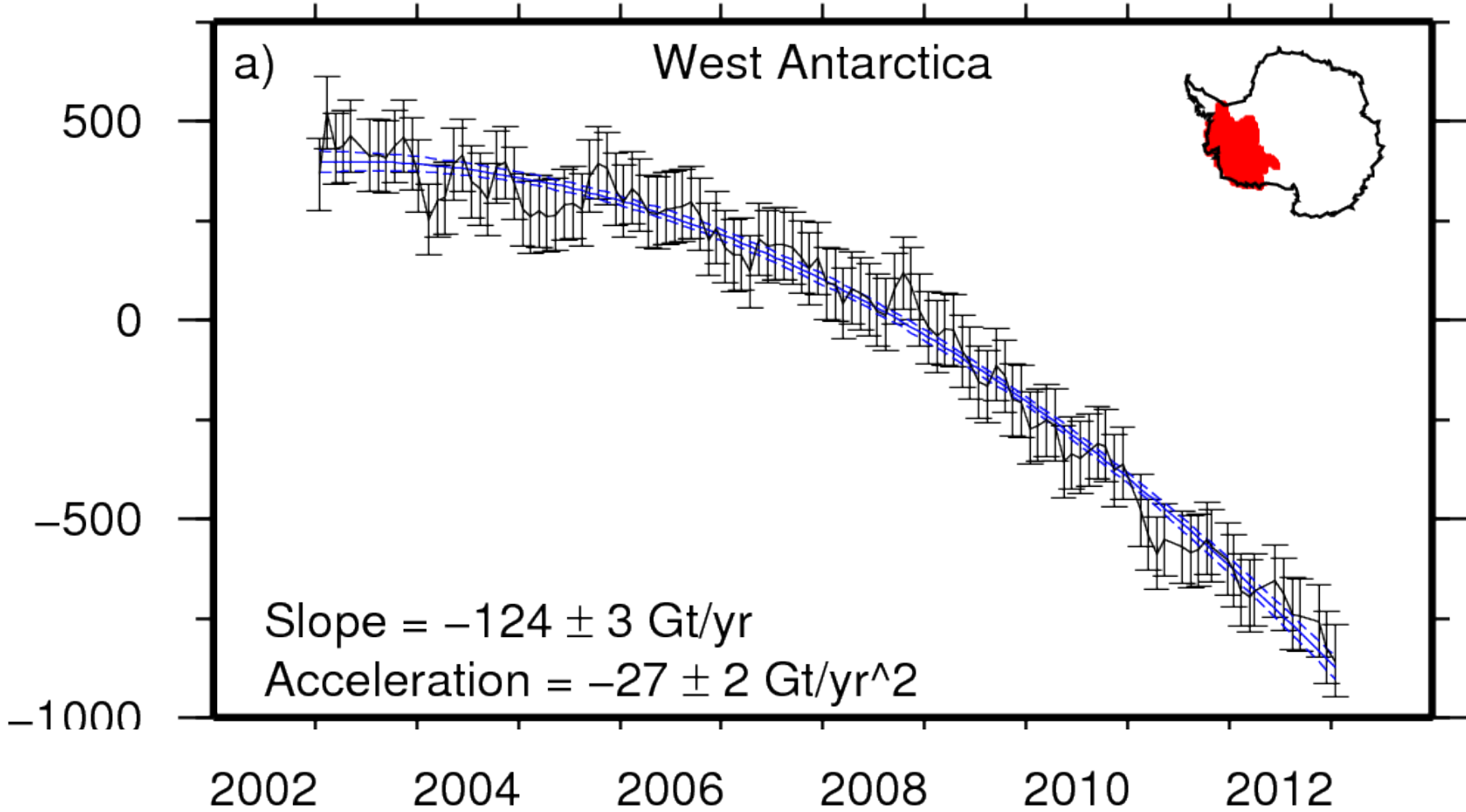
Ice mass loss 2002–2013



Ice mass loss 2002–2013



Ice mass loss 2002–2013



Conclusions

The screenshot shows a web browser displaying the GitHub profile of Frederik J Simons. The browser's address bar shows the URL `https://github.com/fjsimons`. The profile page includes a profile picture, a bio, a list of pinned repositories, and a contribution calendar for the last year.

Profile Information:

- Name:** Frederik J Simons
- Username:** fjsimons
- Bio:** I am a geologically inspired, geophysically educated, computationally motivated and mathematically minded geoscientist, passionate about reproducible research.
- Location:** Princeton, NJ, 08544
- Website:** <http://www.frederik.net>
- Joined:** Feb 26, 2014

Pinned repositories:

- `csdms-contrib/slepian_alpha`: Scalar spherical-harmonic analysis and Slepian functions. 6 stars, Matlab.
- `csdms-contrib/slepian_echo`: Cubed-sphere wavelets. 1 star, Matlab.
- `csdms-contrib/slepian_juliet`: Maximum-likelihood analysis of univariate isotropic Matérn random fields. 1 star, Matlab.
- `csdms-contrib/slepian_oscar`: Seismic processing bits and pieces. 0 stars, Matlab.

Contributions: 159 contributions in the last year. The contribution calendar shows activity from May to September, with a peak in July.

`https://github.com/fjsimons`