modelagem direta e inversão

de campos gravitacionais em coordenadas esféricas

Observatório Nacional – 29 de Abril de 2016

Leonardo Uieda

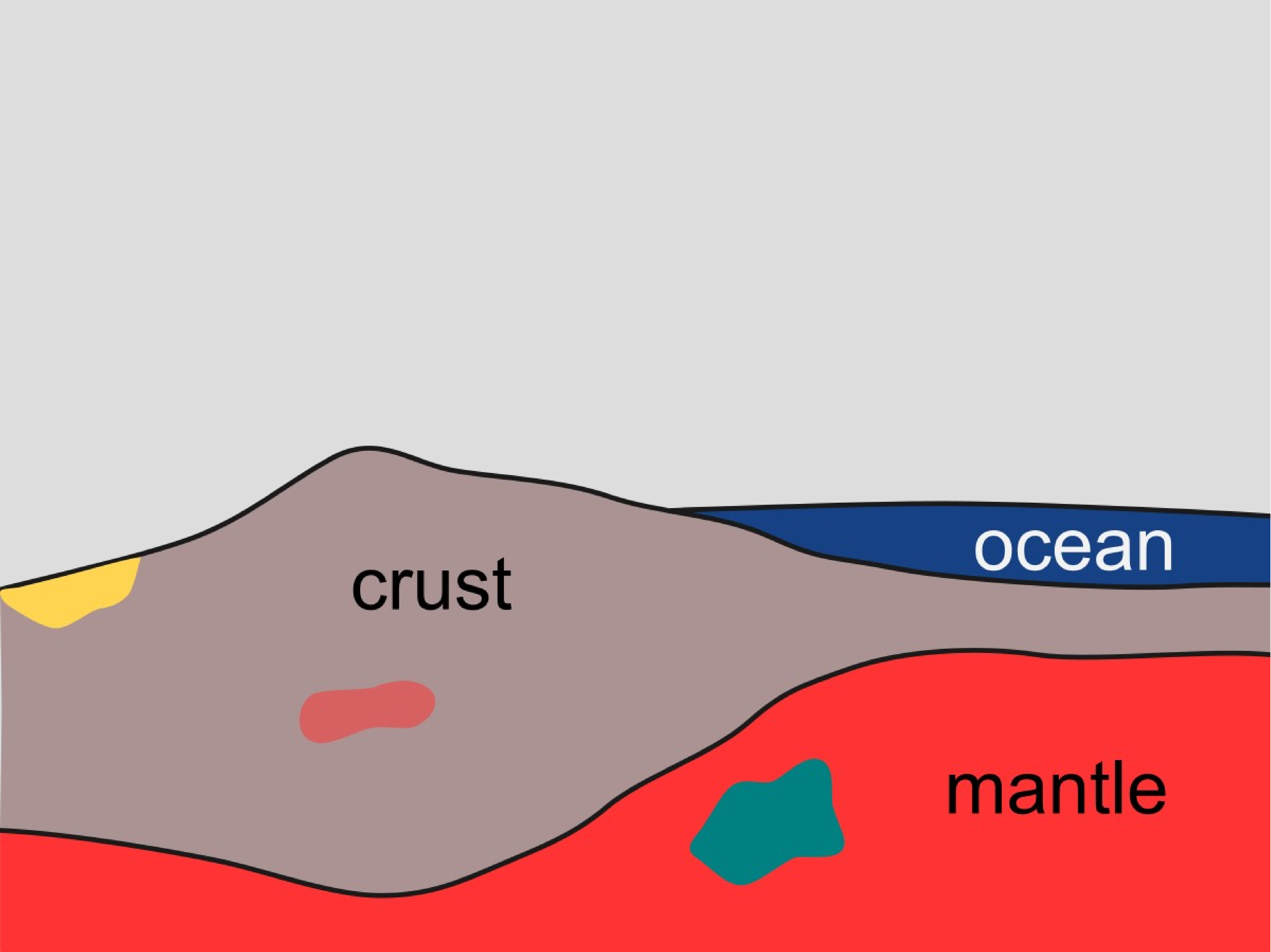
Valéria C F Barbosa (orientadora)

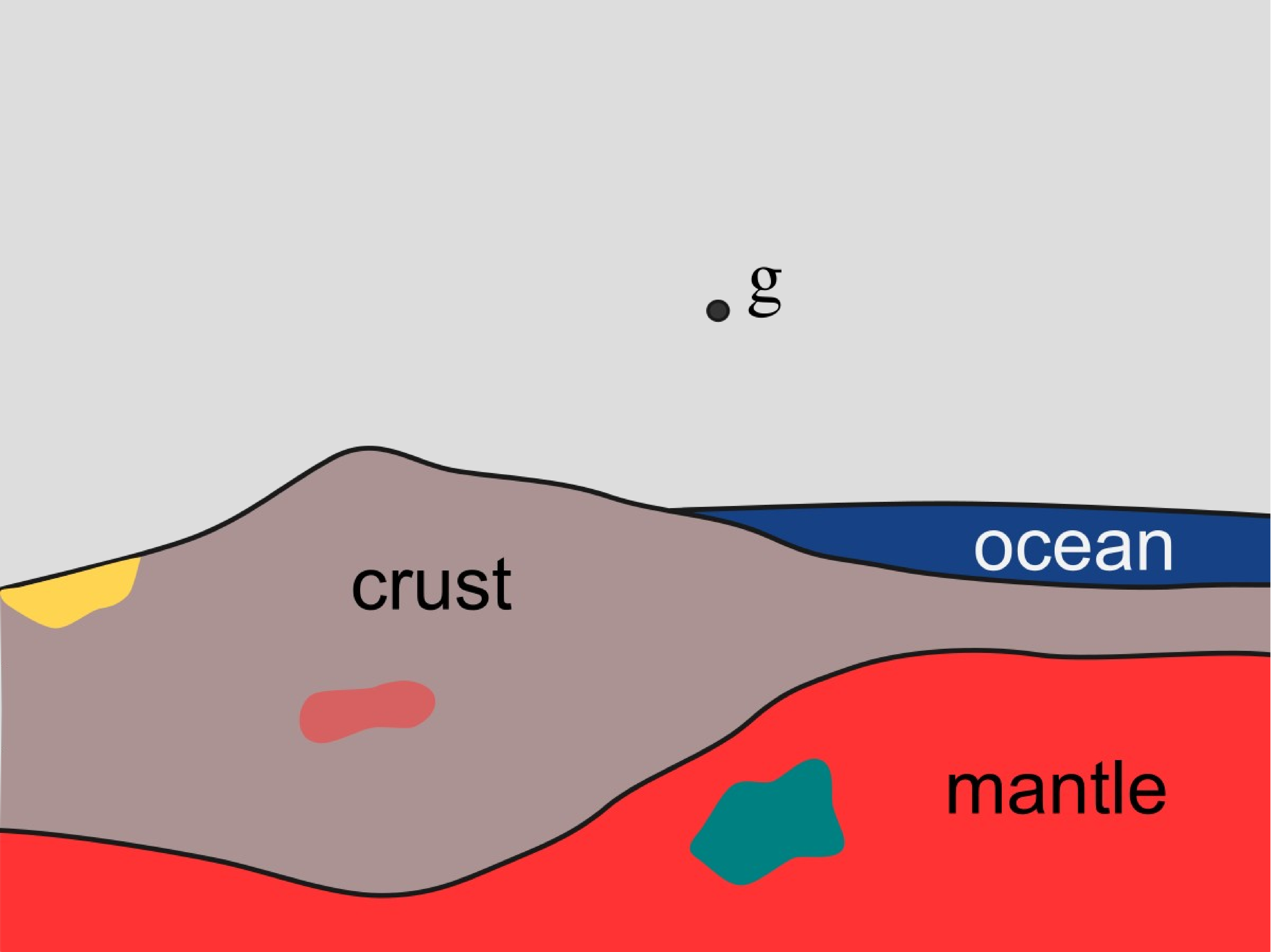
A COORDENADORA DE PÓS-GRADUAÇÃO ADVERTE:

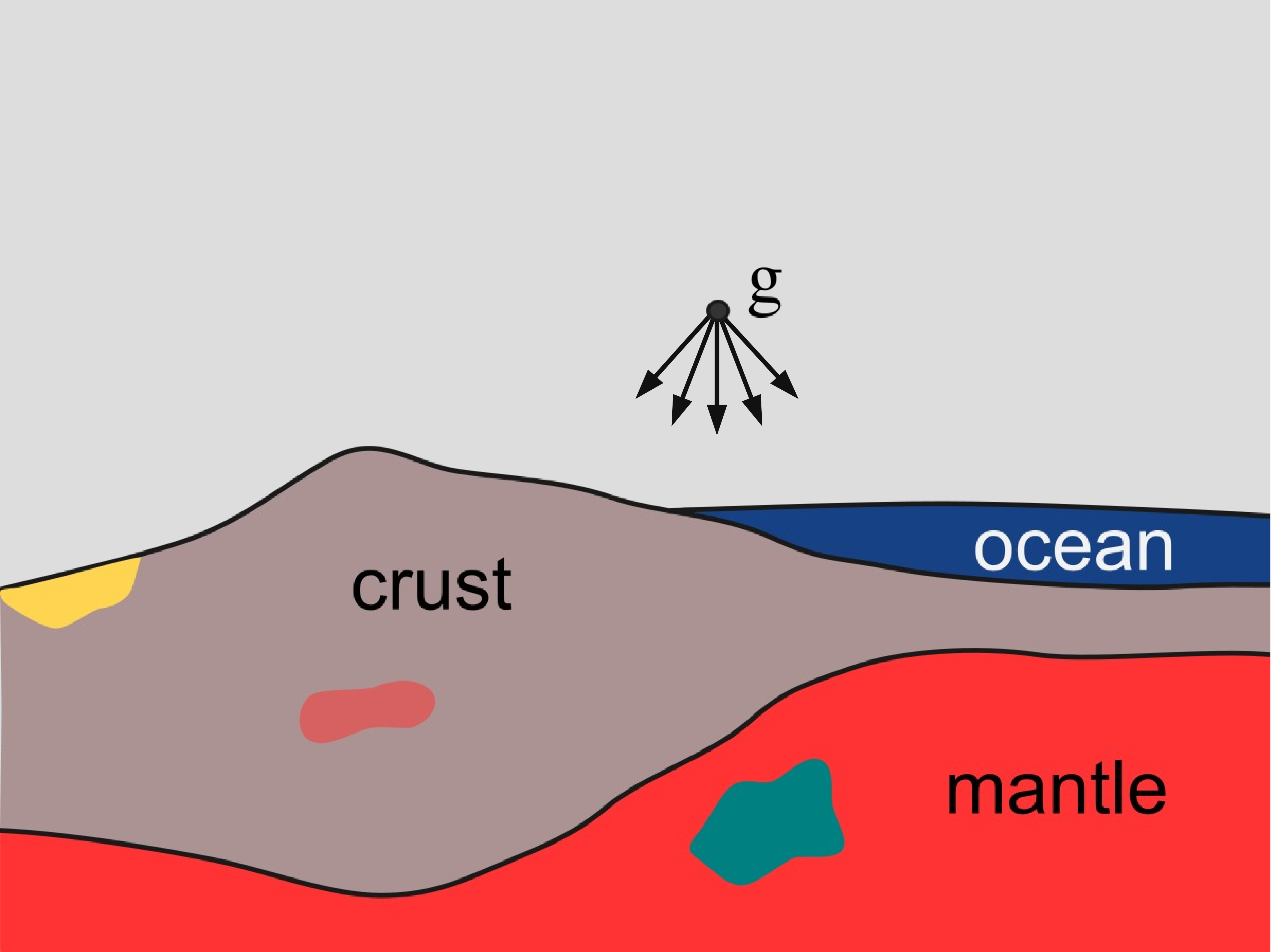
Efeitos colaterais dessa defesa incluem:

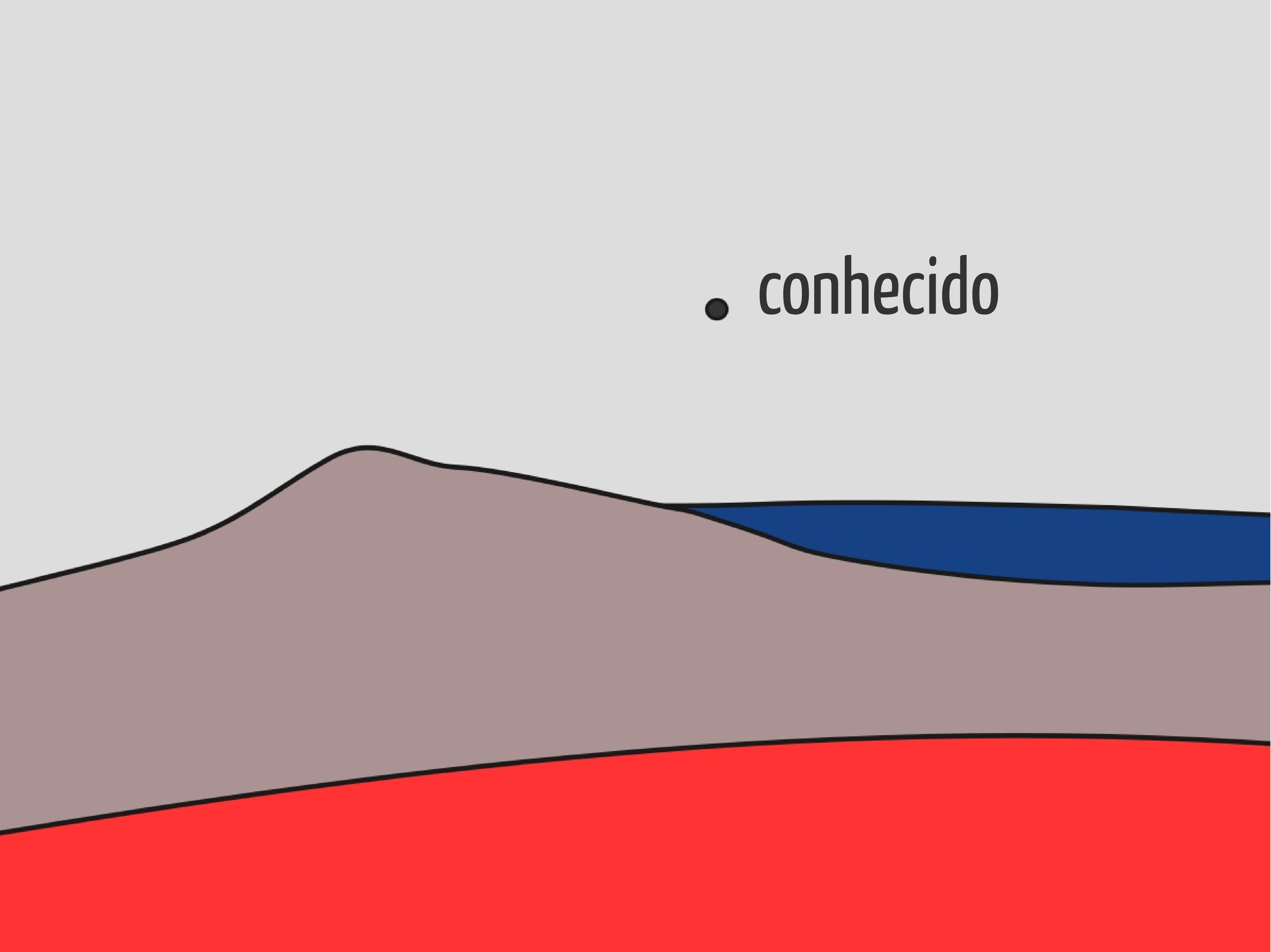
discussão	choro
discórdia	aprovação
crítica	reprovação

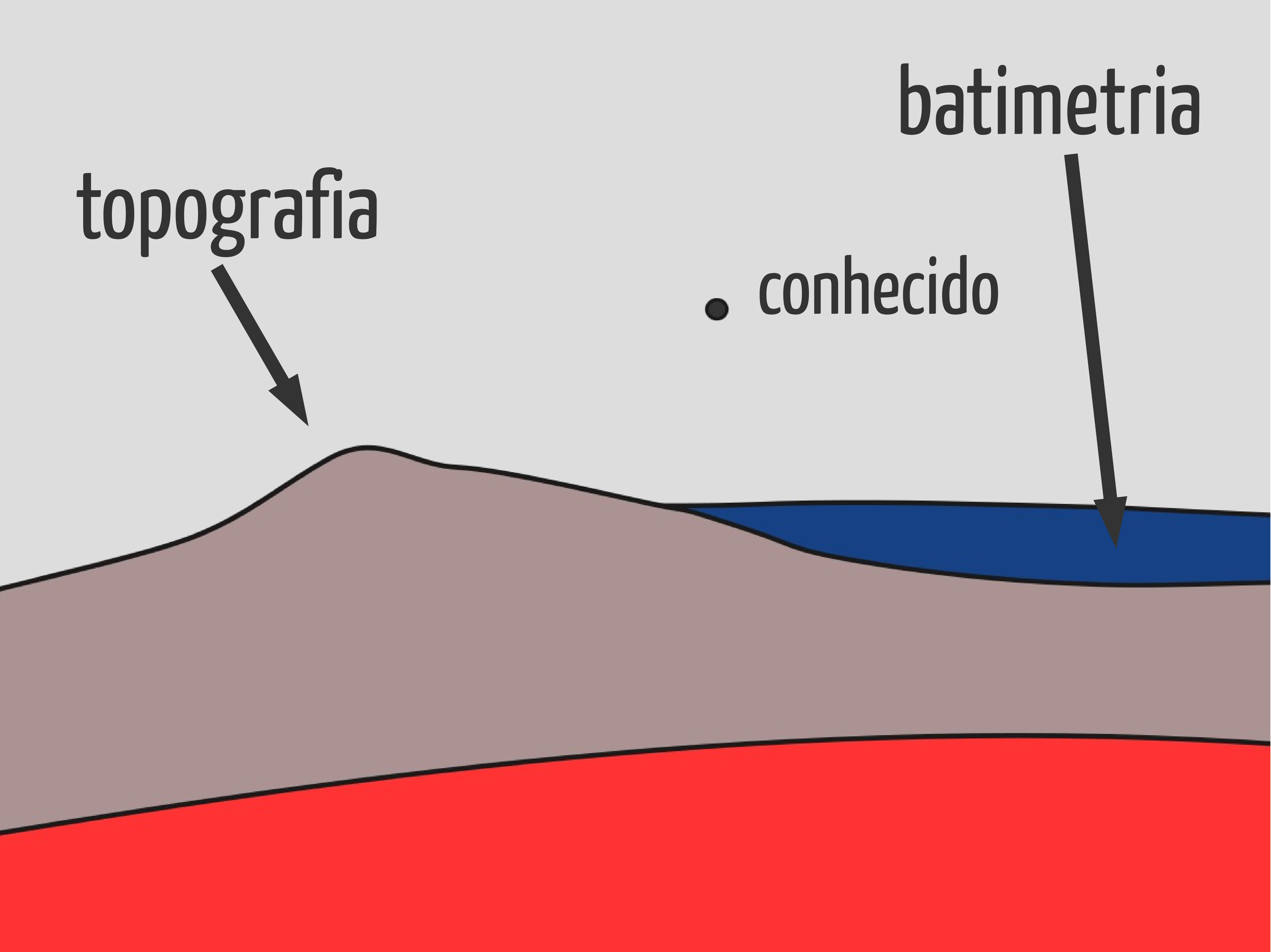
introdução

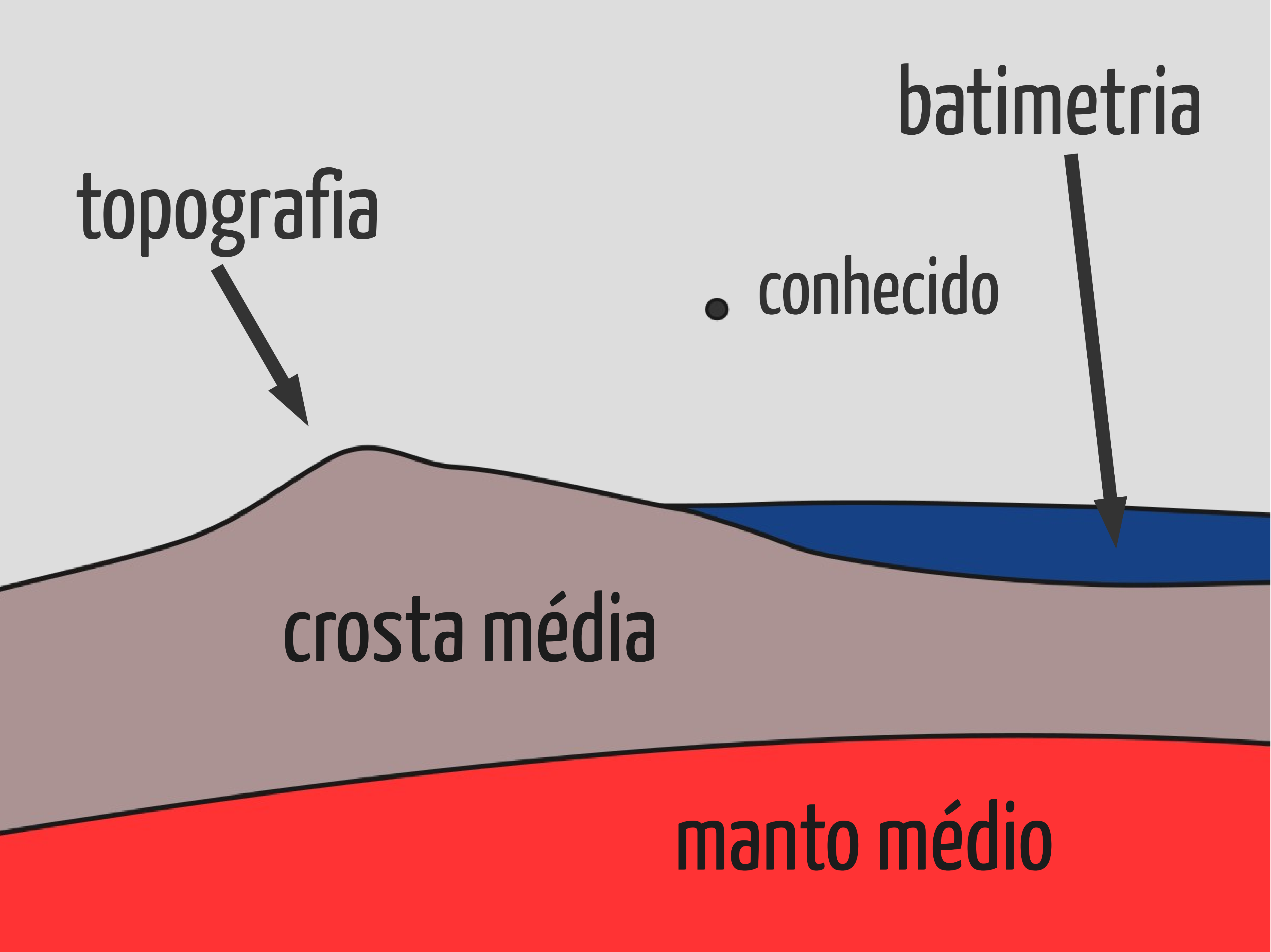

ocean
crust
mantle

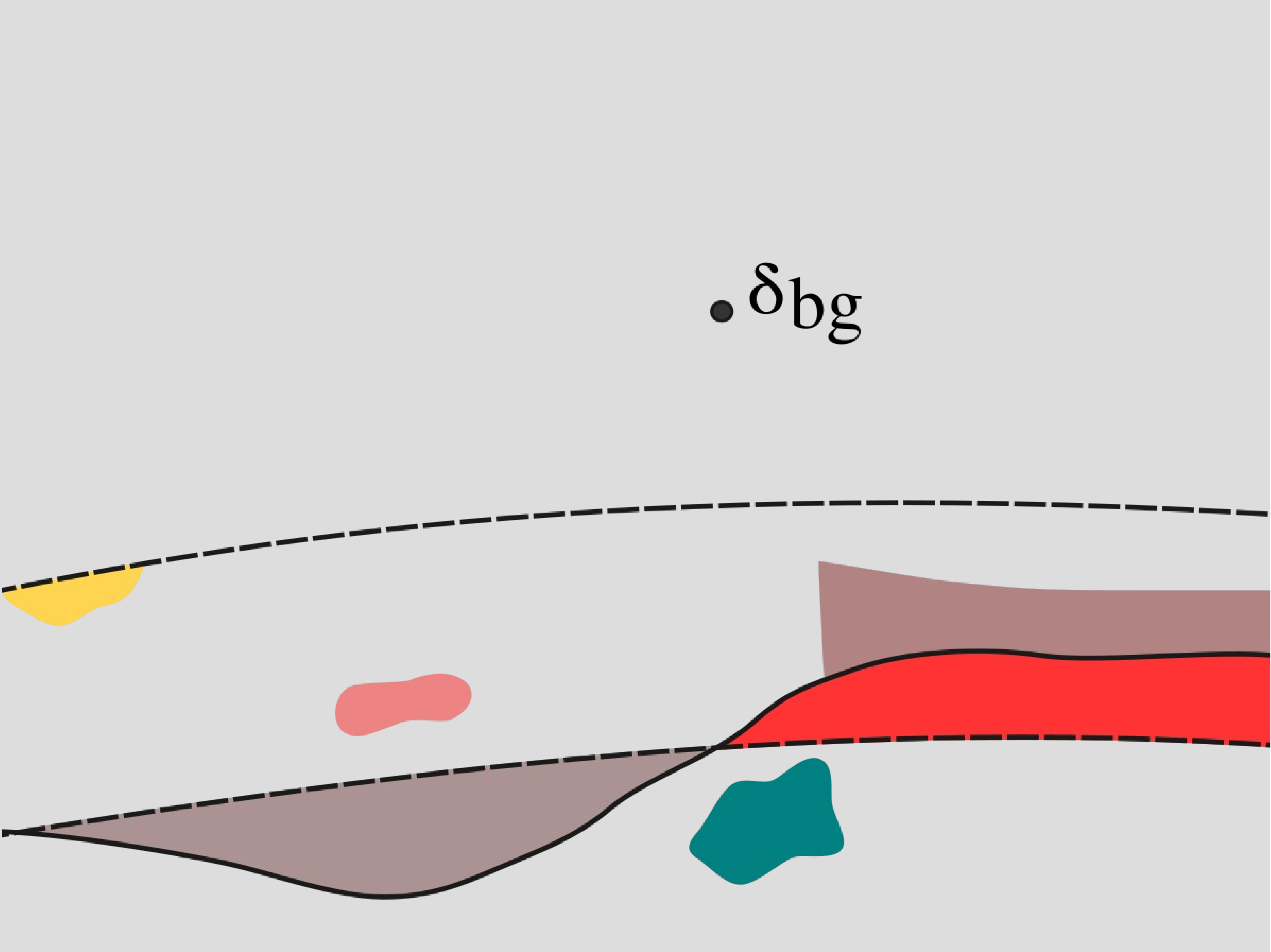

g
crust
ocean
mantle

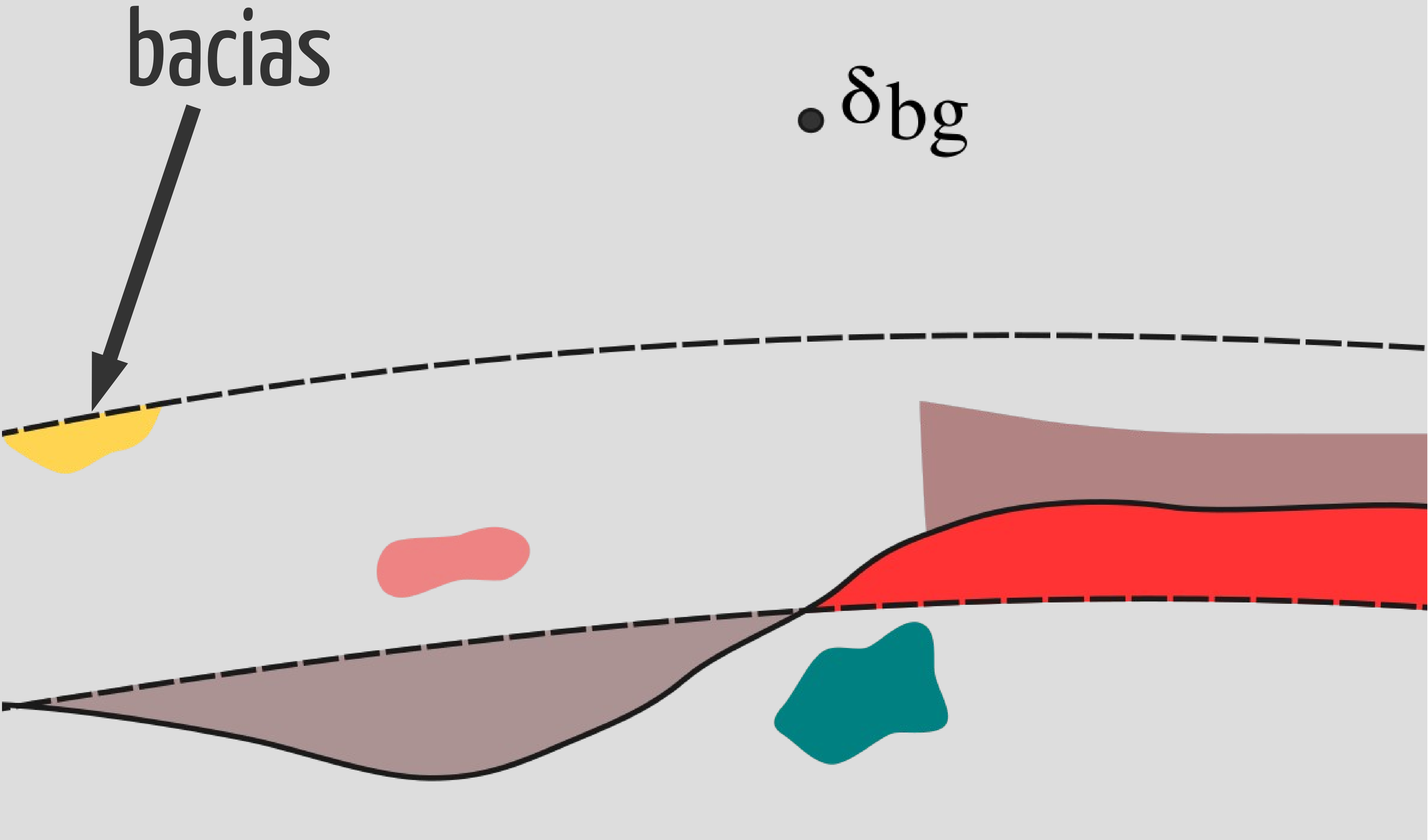

g
ocean
crust
mantle

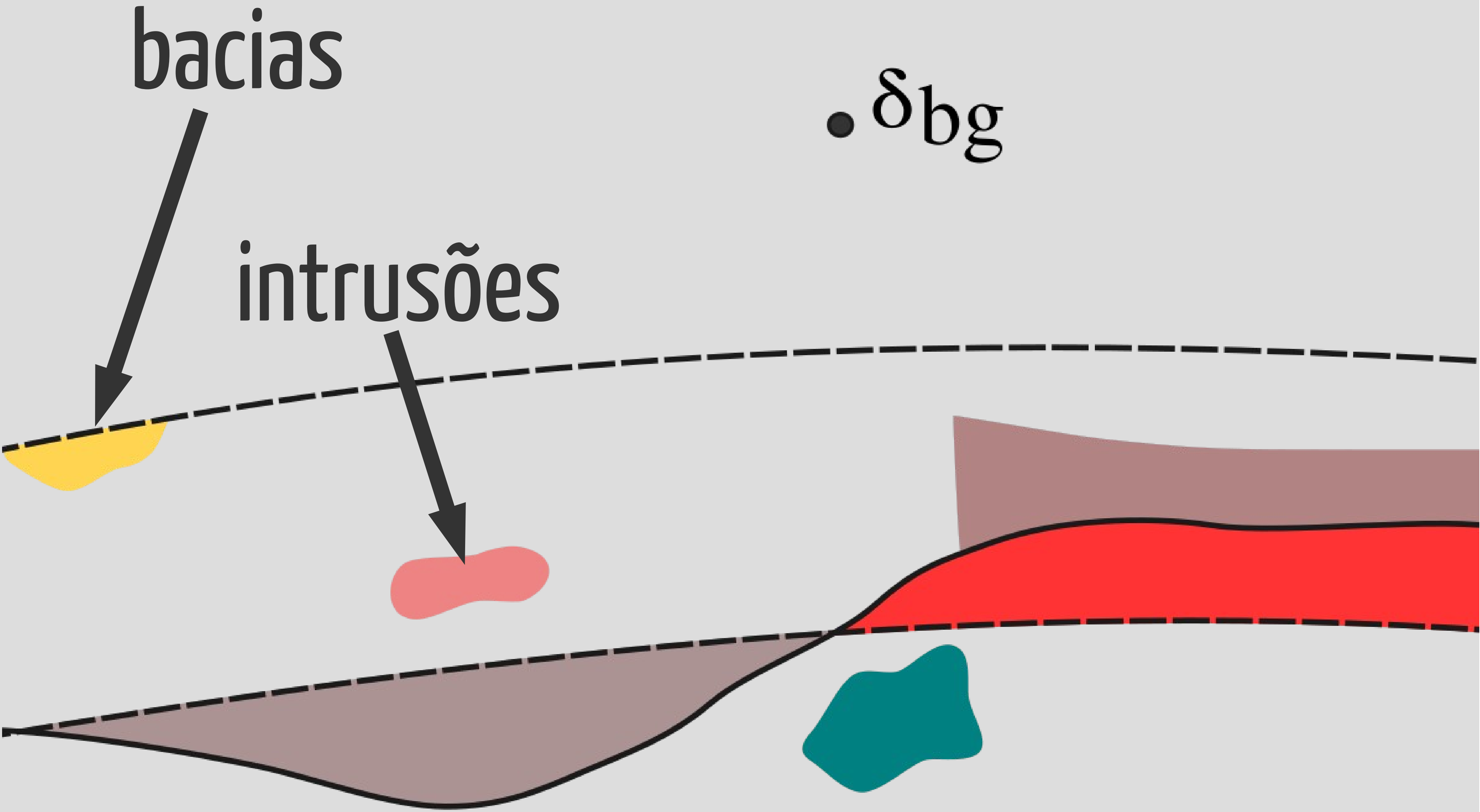

conhecido

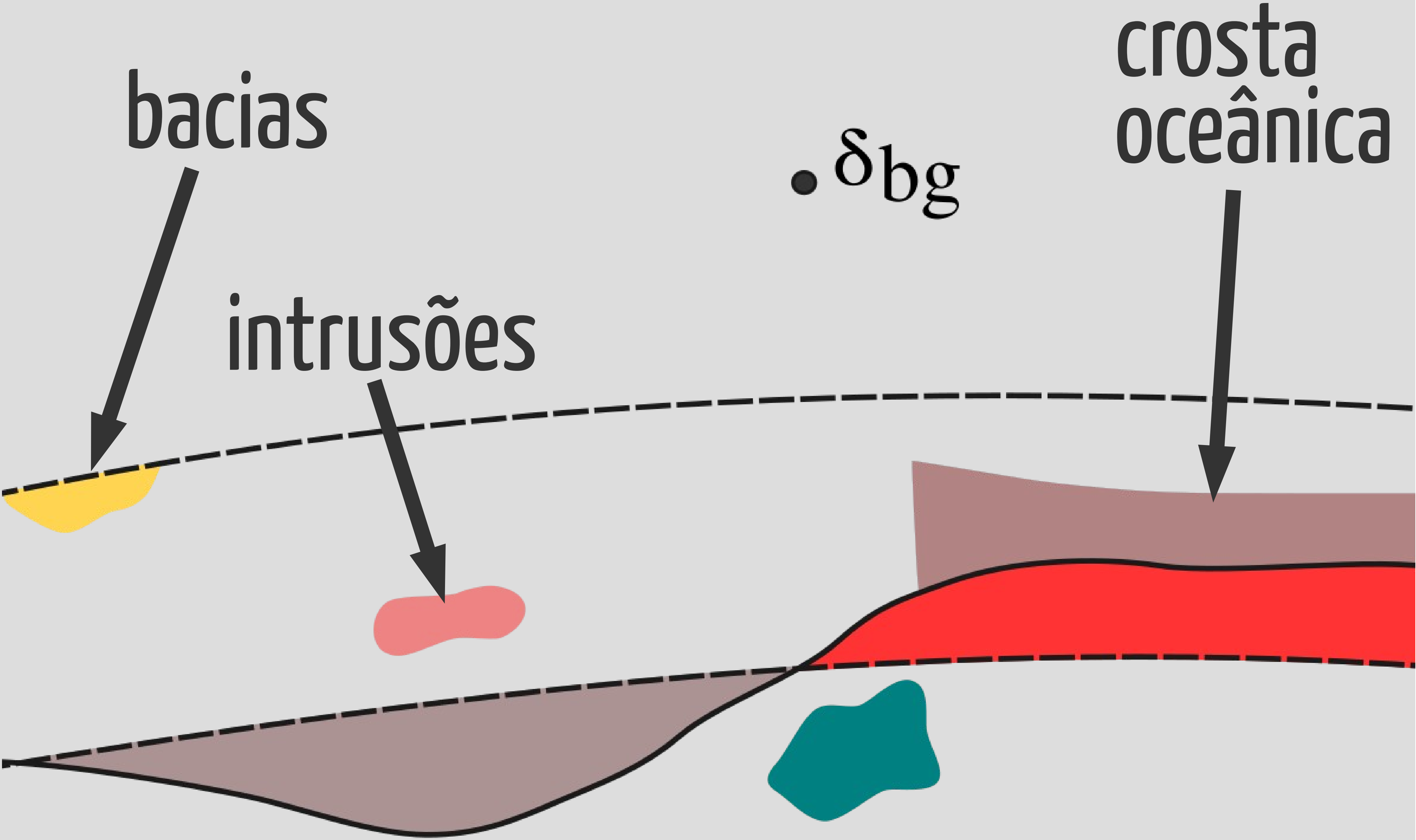

batimetria
topografia
conhecido


batimetria
topografia
conhecido
crosta média
manto médio

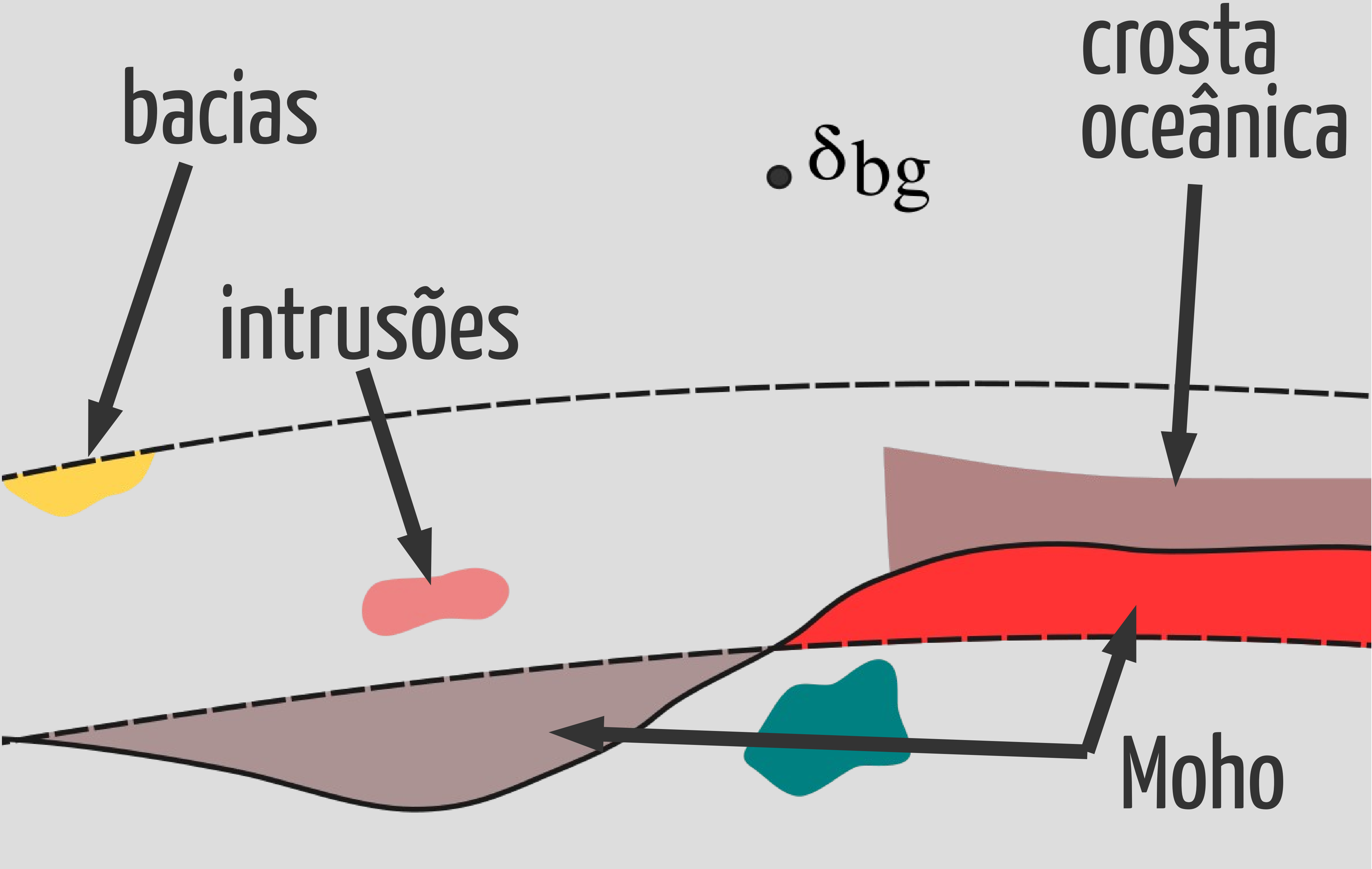

δ_{bg}


bacias
δ_{bg}

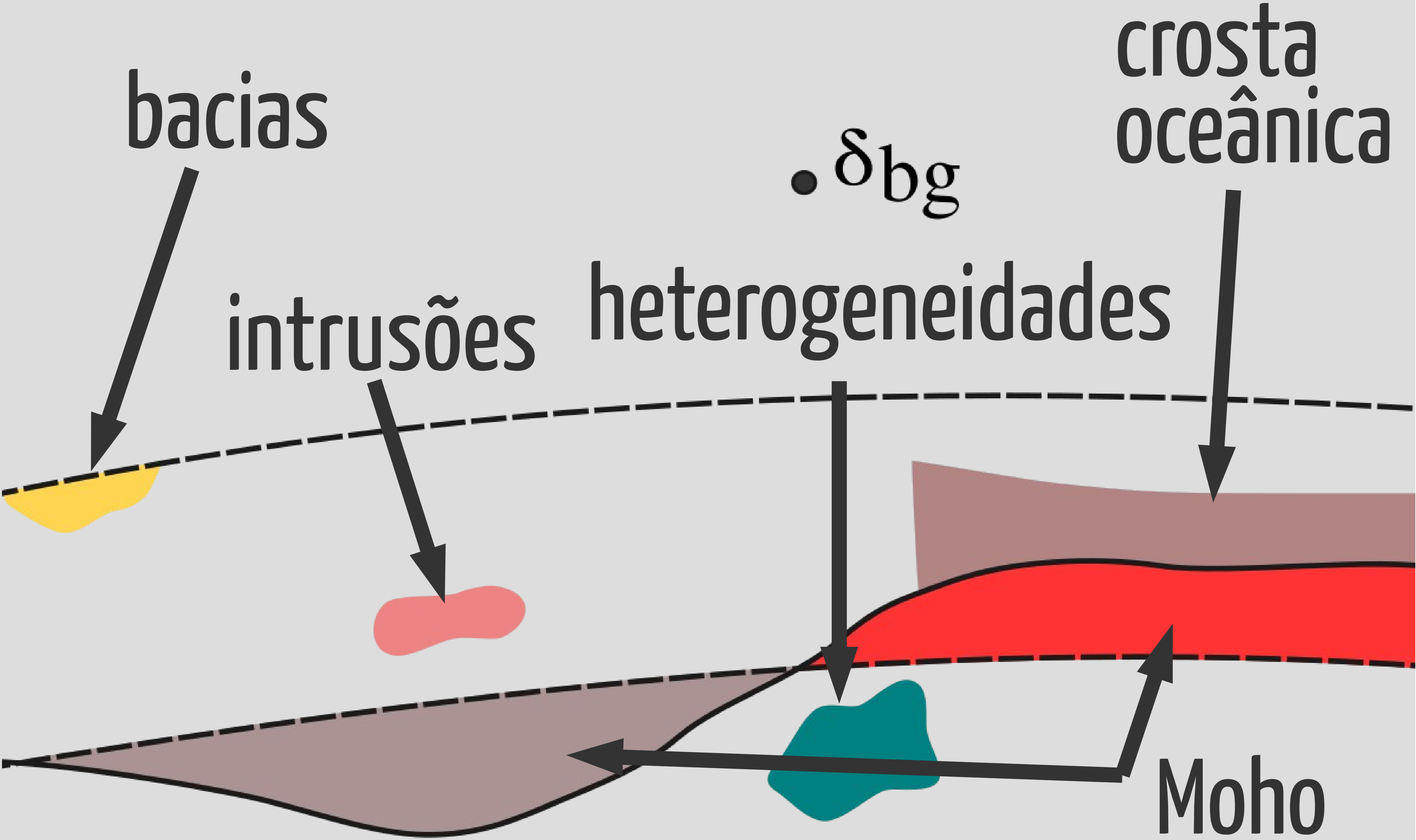

bacias
δ_{bg}
intrusões

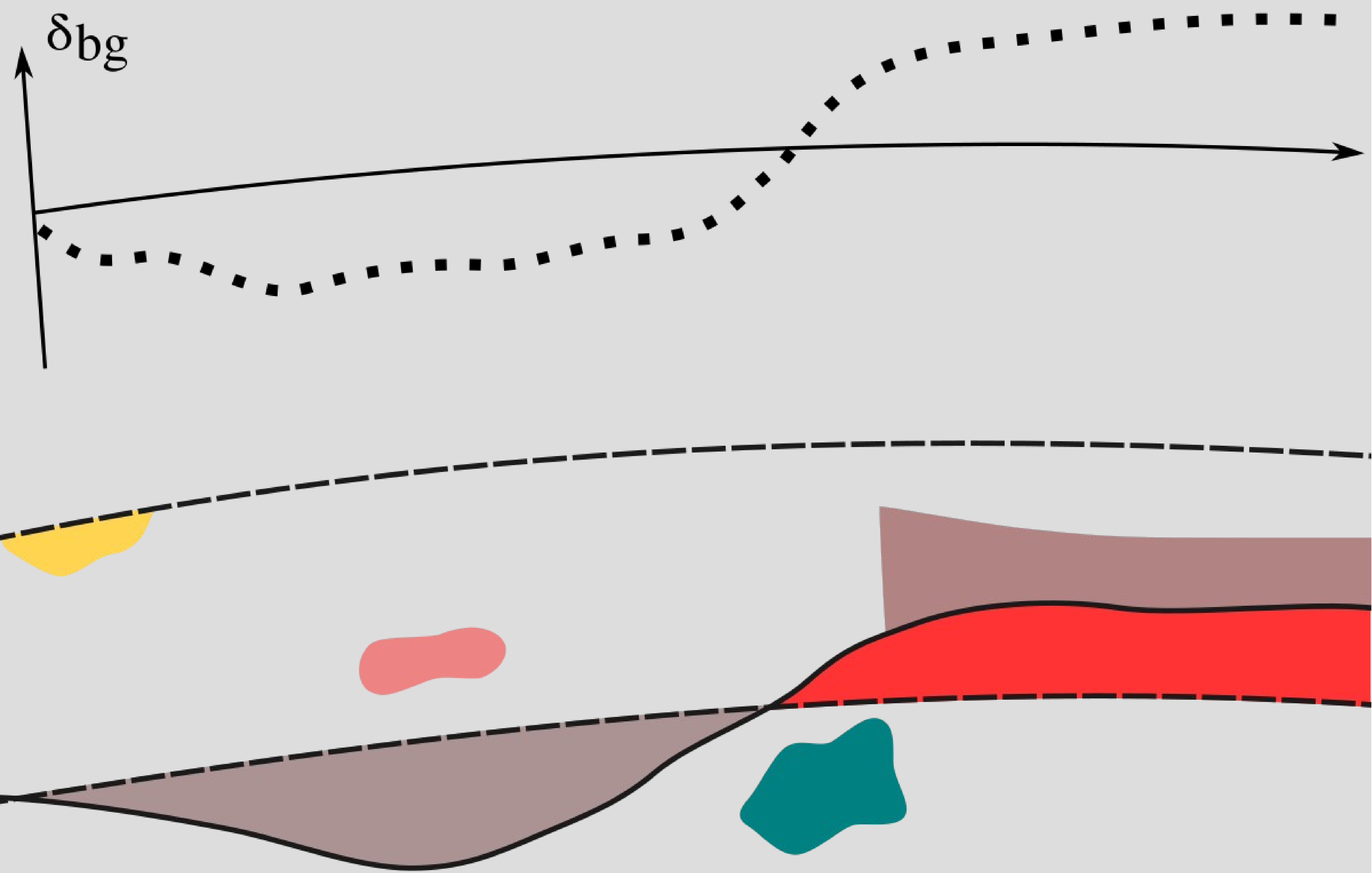


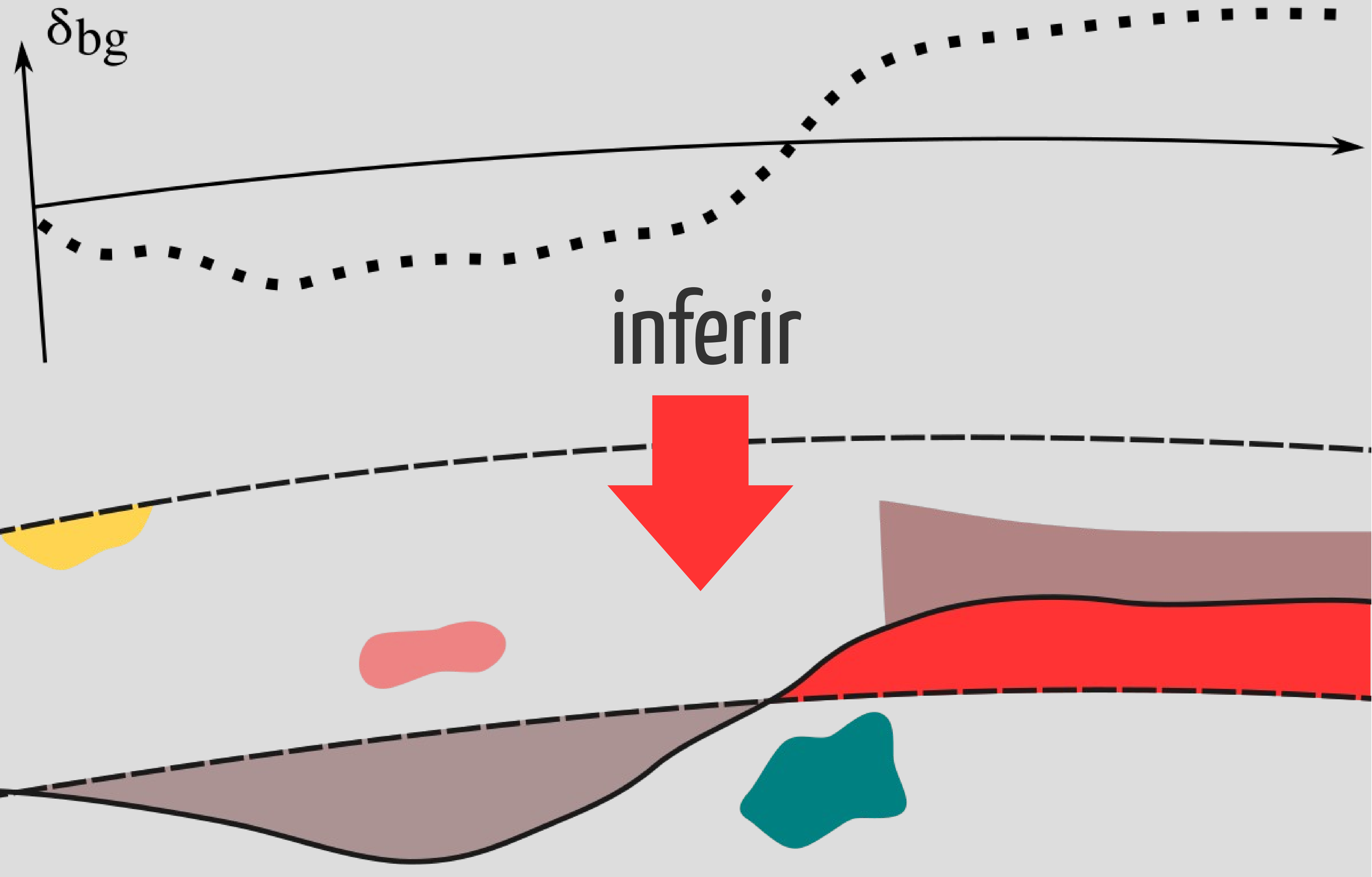
bacias
δ_{bg}
crosta
oceânica
intrusões



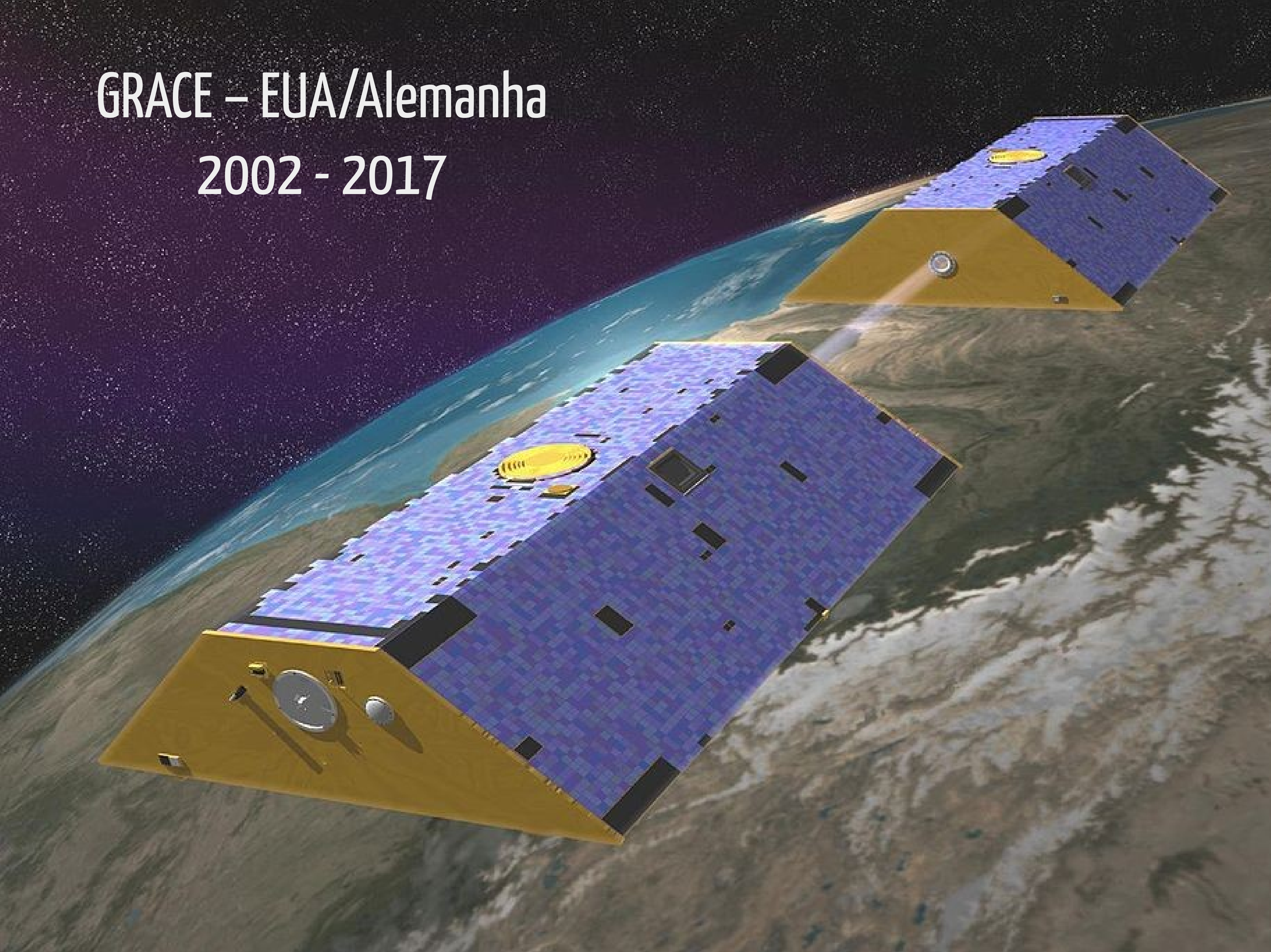
bacias
δ_{bg}
crosta oceânica
intrusões
Moho

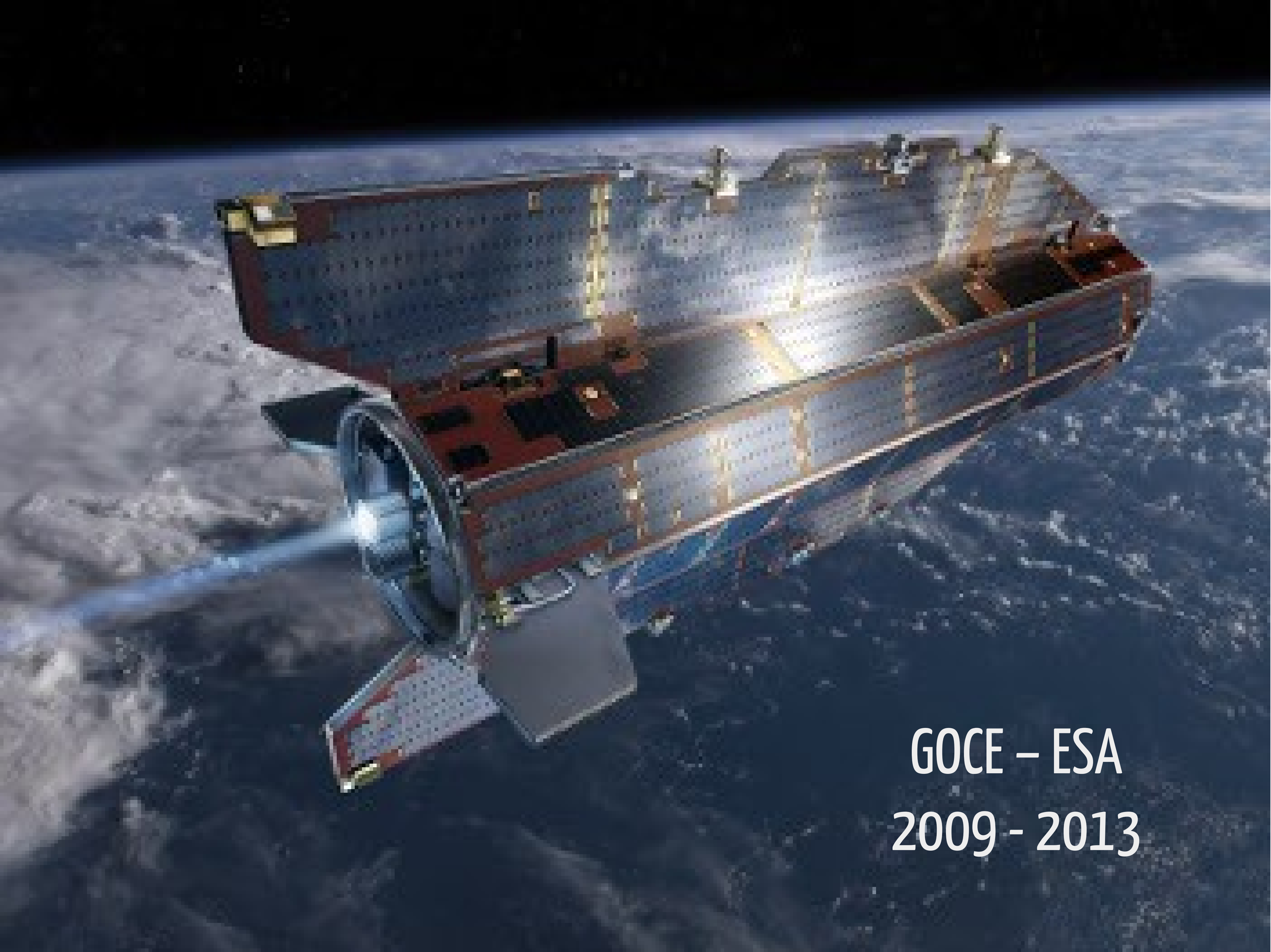

bacias
intrusões
δ_{bg}
heterogeneidades
crosta oceânica
Moho


δ_{bg}

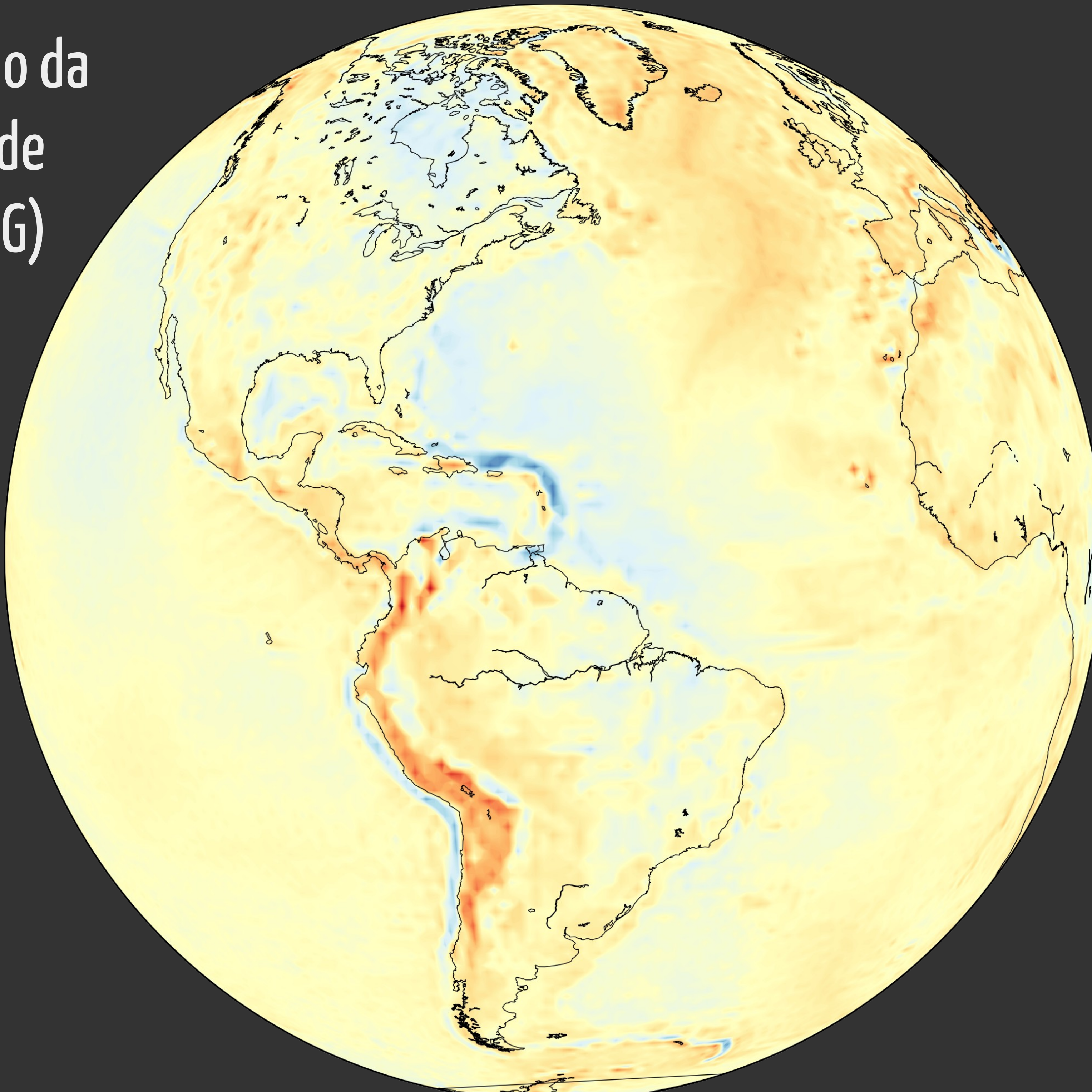


δ_{bg}
inferir


GRACE – EUA/Alemanha
2002 - 2017


GOCE – ESA
2009 - 2013

distúrbio da gravidade (GGM05G)



dados ✔

dados ✔

subsuperfície

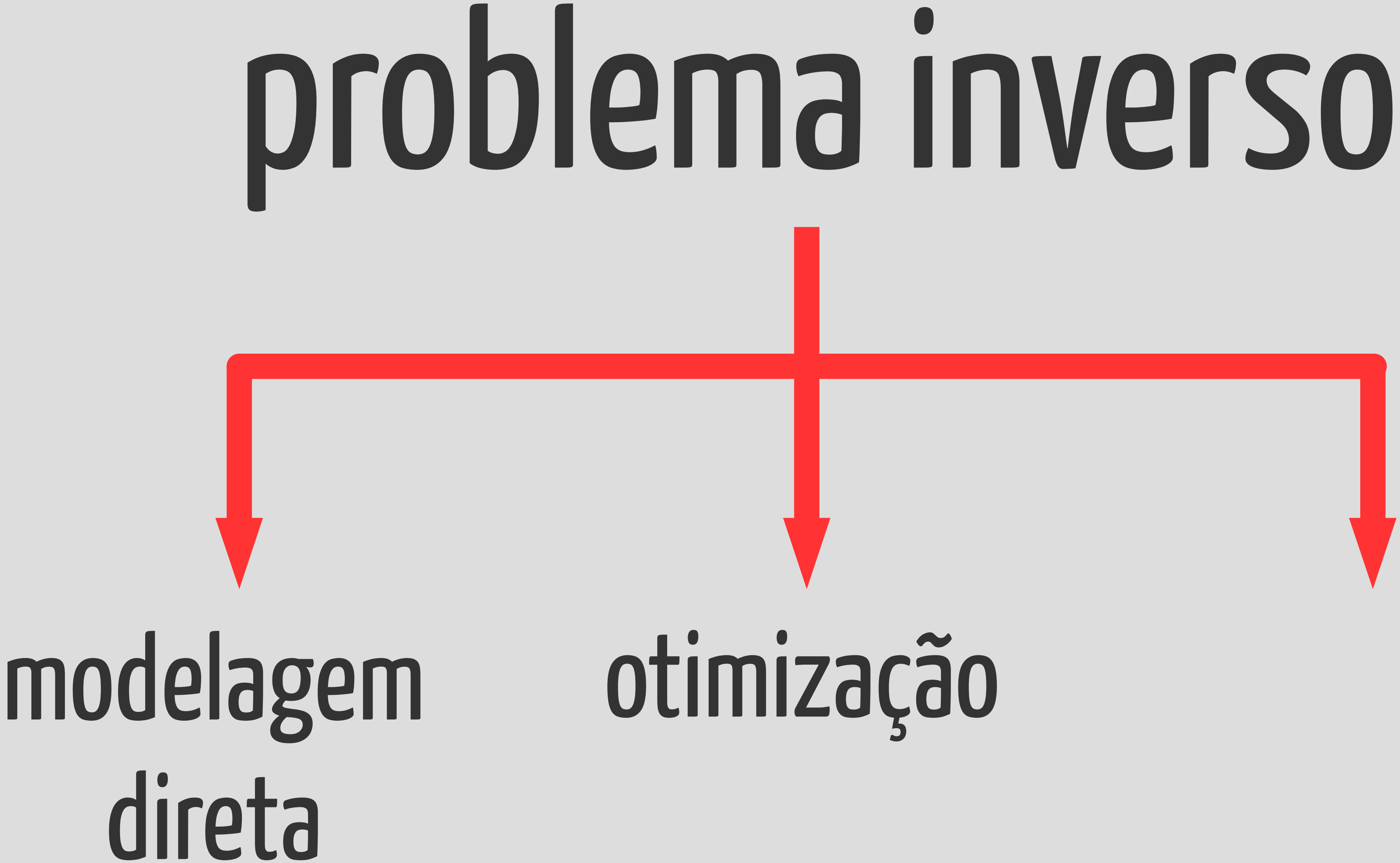
aproximação esférica

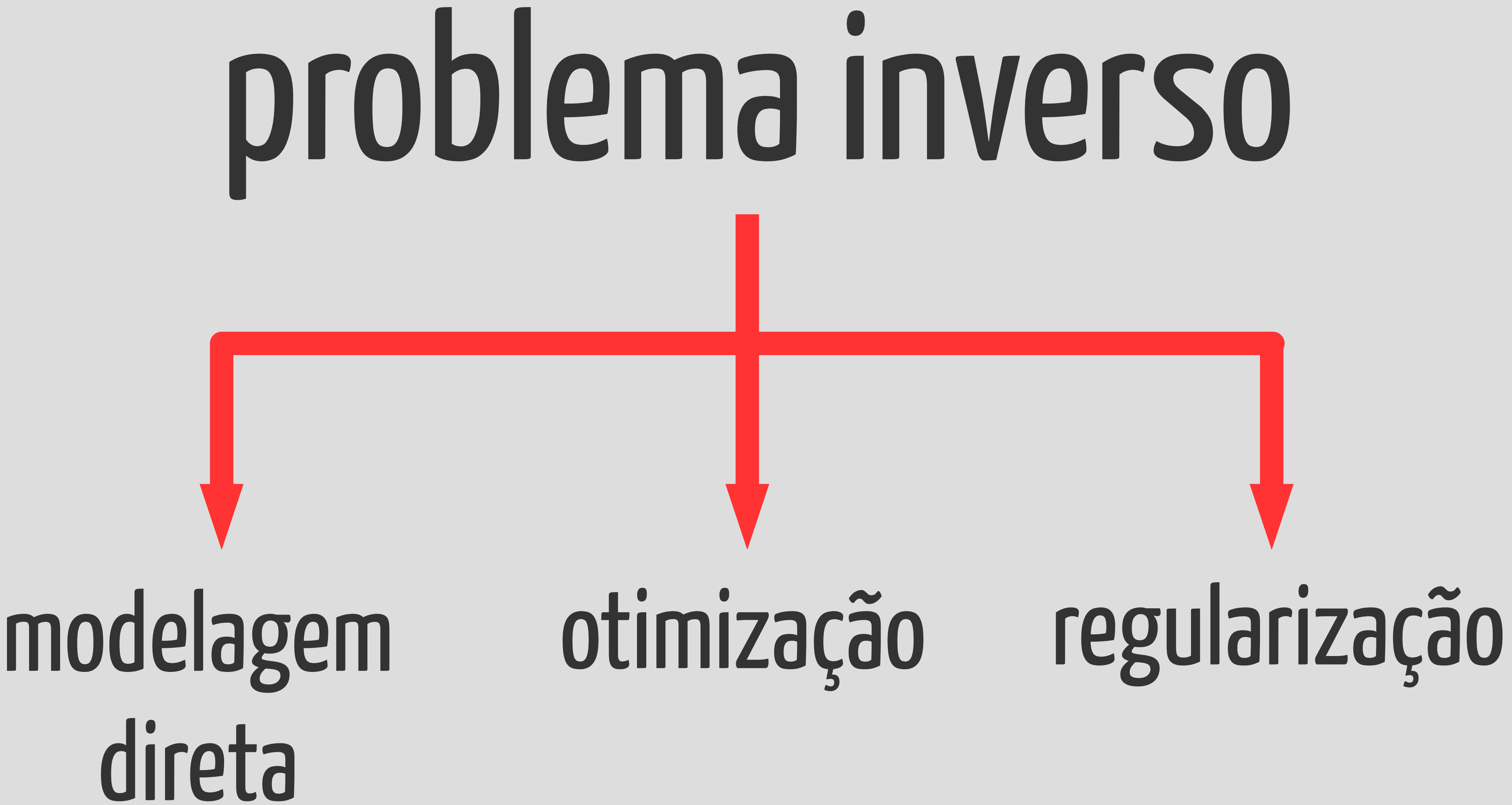
problema inverso

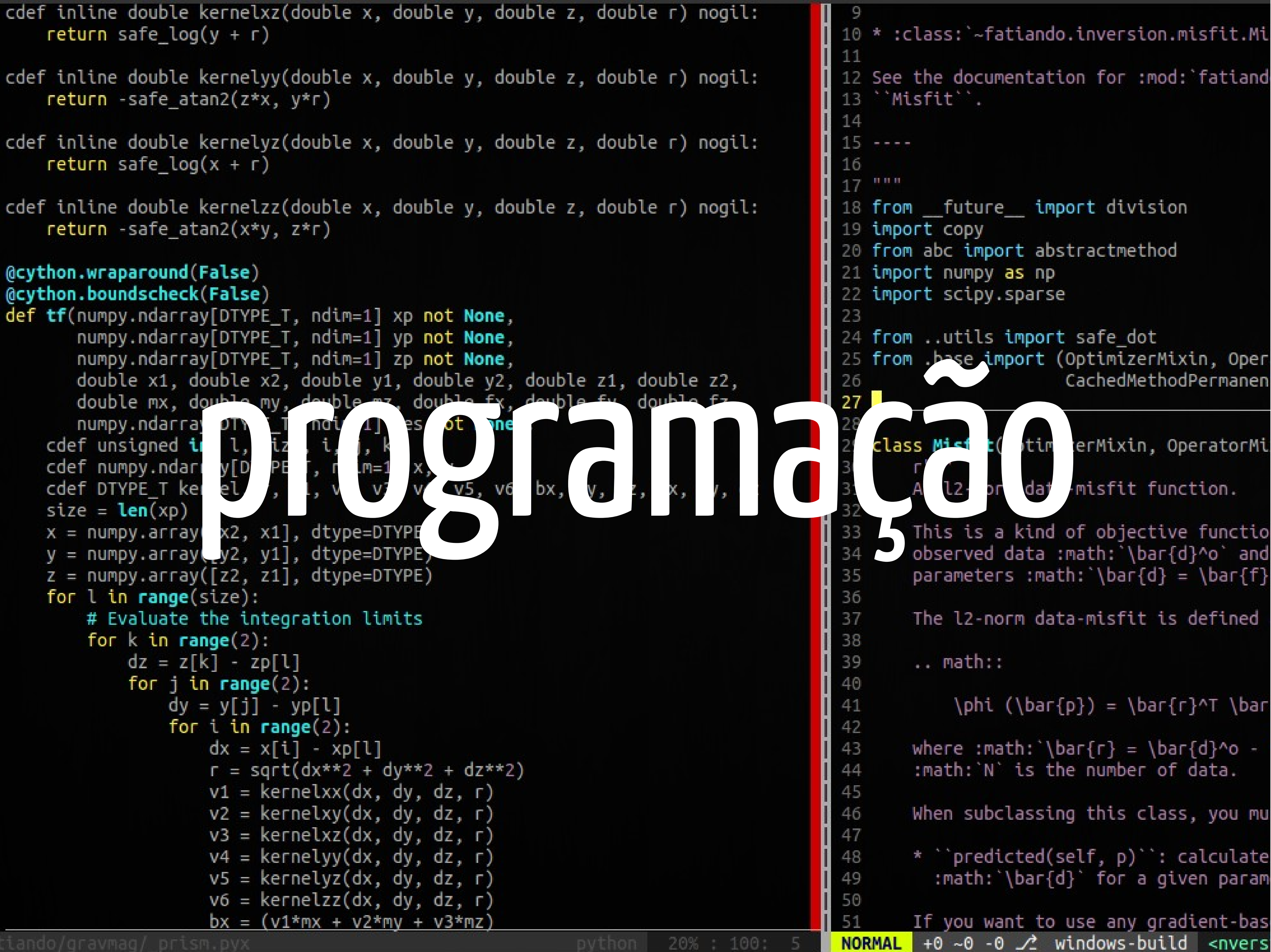
problema inverso

problema inverso

modelagem direta


problema inverso
modelagem direta
otimização


problema inverso
modelagem direta
otimização
regularização


programação

1

2

3

programa A	programa B

Novo método

Tesseroids

modelagem direta
aprox. esférica

programa B

Novo método

Tesseroids

modelagem direta
aprox. esférica

fatiando a terra

modelagem direta
otimização
regularização, etc

Novo método

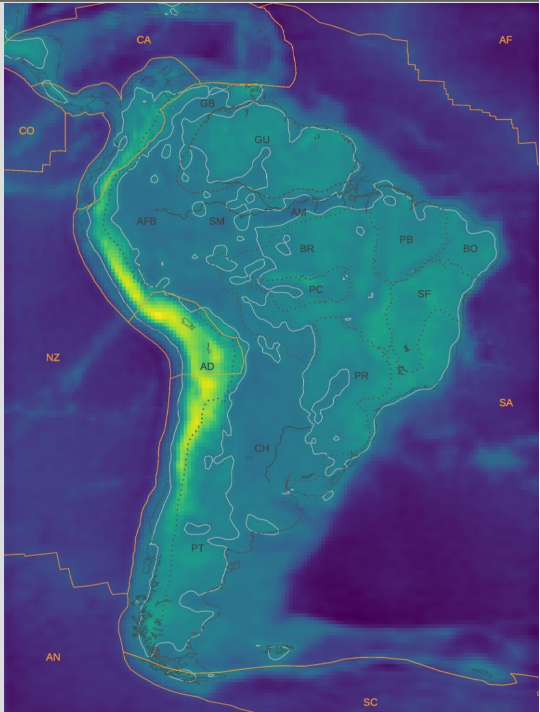
Tesseroids

modelagem direta
aprox. esférica

fatiando a terra

modelagem direta
otimização
regularização, etc

método: inversão não-linear rápida
aplicação: Moho da América do Sul



Introdução

Tesseroids

Fatiando a Terra

Inversão Moho

Conclusão

Introdução

Tesseroids

Fatiando a Terra

Inversão Moho

Conclusão

Tesseroids

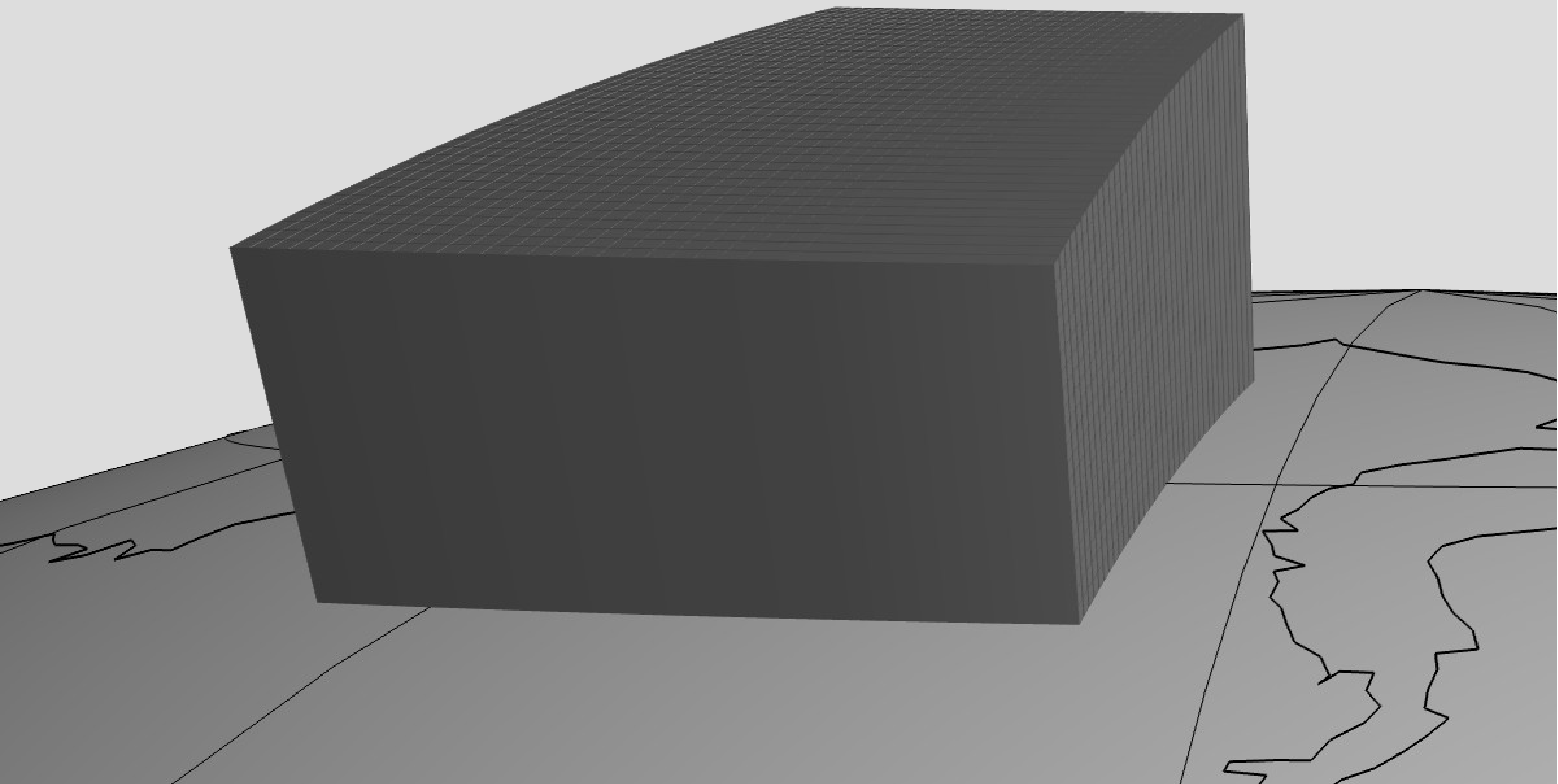
modelagem direta em coordenadas esféricas

GEOPHYSICS

Geophysical Software and Algorithms

Submetido: Mar 2015
R1: Nov 2015
Aceito: Abr 2016

tesseroide = prisma esférico



$$g_z = G\rho \int_{\lambda_1}^{\lambda_2}\int_{\varphi_1}^{\varphi_2}\int_{r_1}^{r_2} K_z \, dr \, d\varphi \, d\lambda$$

cte.

$$g_z = G\rho \iiint\limits_{\lambda_1 \varphi_1 r_1}^{\lambda_2 \varphi_2 r_2} K_z \, dr \, d\varphi \, d\lambda$$

densidade

cte.

$$g_z = G\rho \int_{\lambda_1}^{\lambda_2}\int_{\varphi_1}^{\varphi_2}\int_{r_1}^{r_2} K_z \, dr \, d\varphi \, d\lambda$$

densidade

kernel

(Grombein et al., 2013)

cte.

$$g_z = G\rho \int_{\lambda_1}^{\lambda_2}\int_{\varphi_1}^{\varphi_2}\int_{r_1}^{r_2} K_z \, dr \, d\varphi \, d\lambda$$

Quadratura

Gauss-Legendre

$$\int_{a}^{b} f(x)\,dx$$

$$\int_{a}^{b} f(x)\,dx \approx \frac{b-a}{2} \sum_{i=1}^{N} W_i f(x_i)$$

ordem

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\sum_{i=1}^{N} W_i f(x_i)$$

ordem pesos

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2} \sum_{i=1}^{N} W_i f(x_i)$$

ordem

pesos

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\sum_{i=1}^{N} W_i f(x_i)$$

raízes de $P_N(x)$

$$\iiint\limits_{\Omega} K_z(r, \varphi, \lambda)\, d\Omega$$

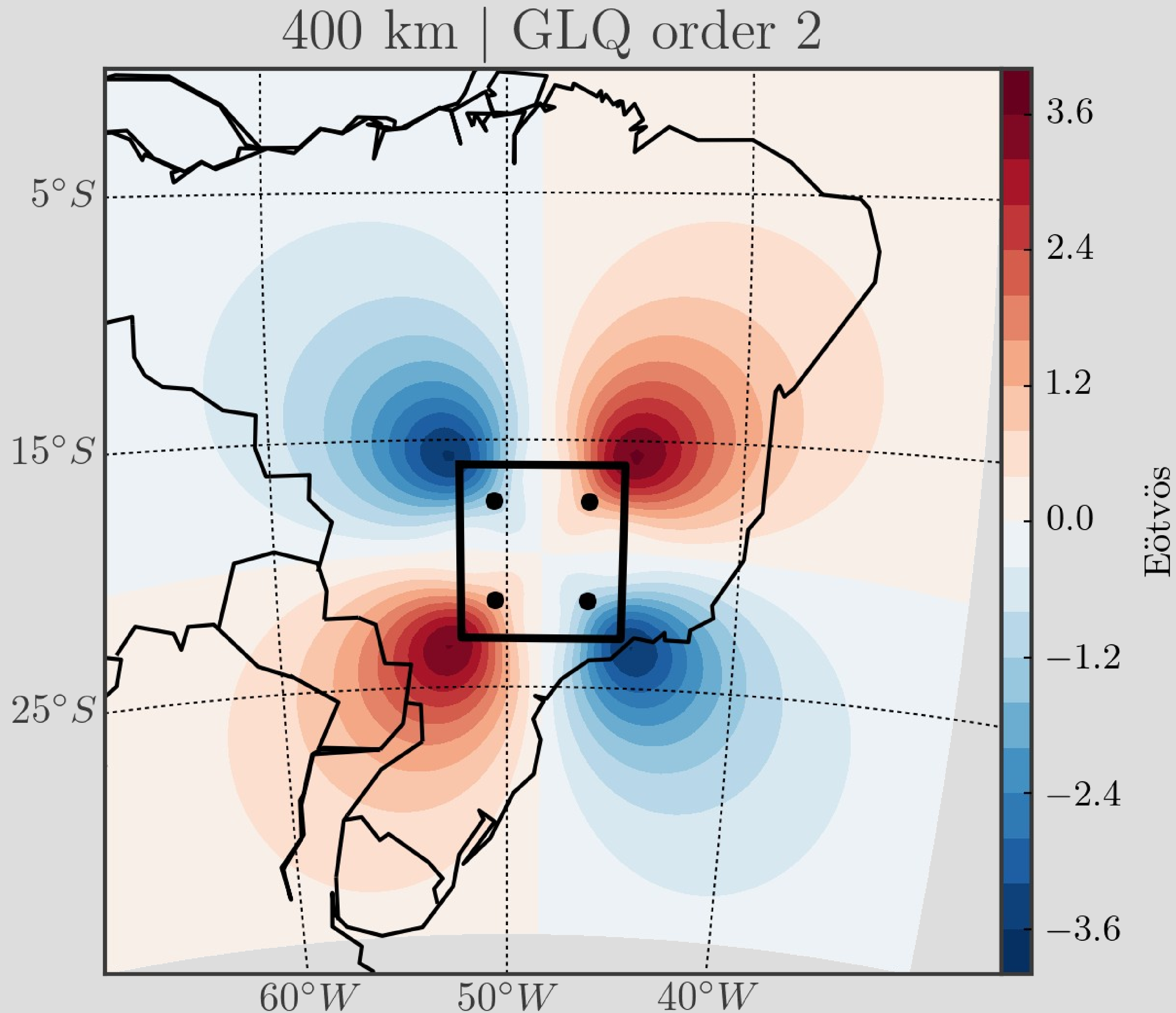
$$\iiint\limits_{\Omega} K_z(r,\varphi,\lambda)\,d\Omega$$

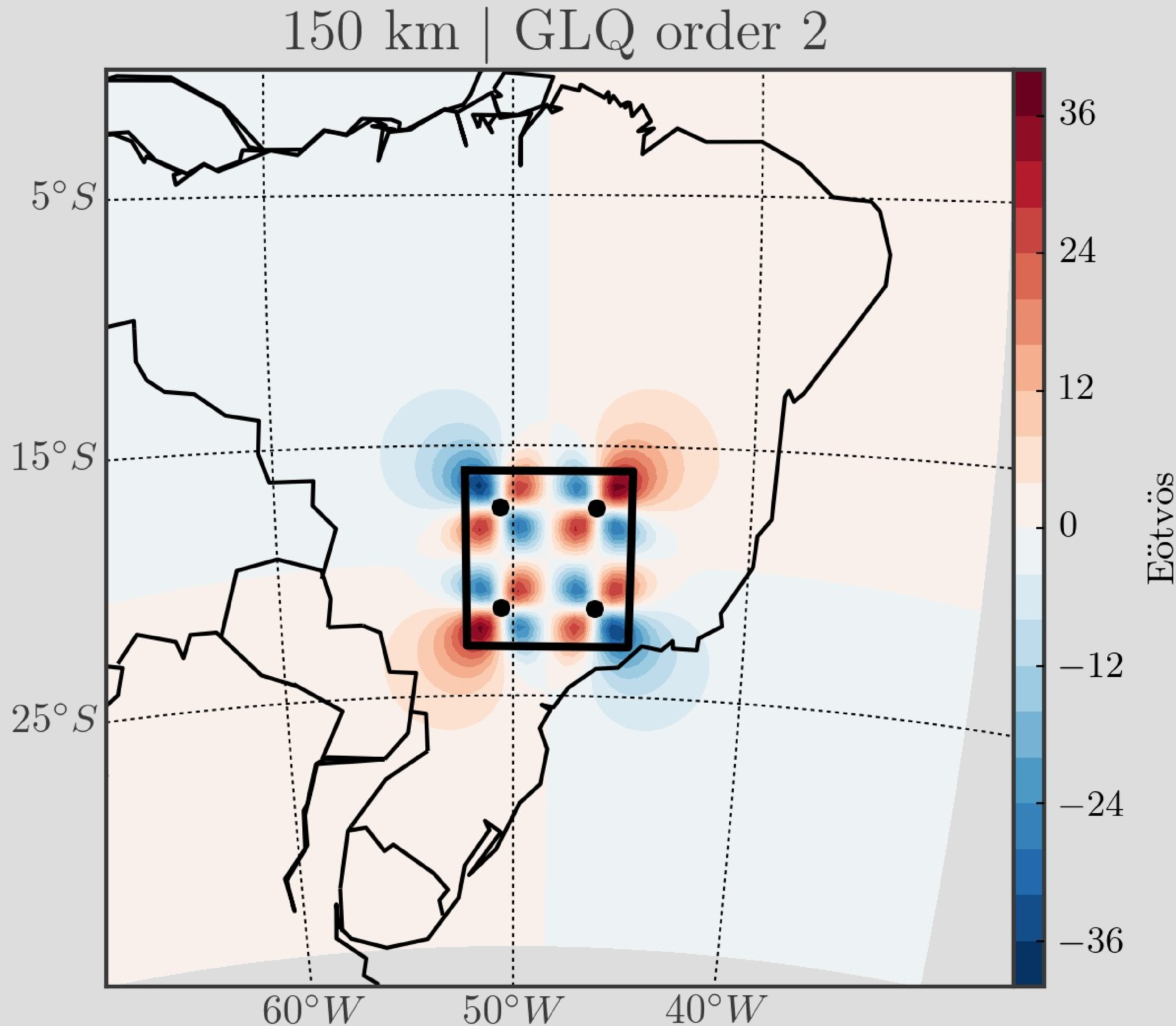
$$A\sum_{i=1}^{N_r}\sum_{j=1}^{N_\varphi}\sum_{k=1}^{N_\lambda} W_i W_j W_k K_z(r_i,\varphi_j,\lambda_k)$$

$$\iiint_{\Omega} K_z(r,\varphi,\lambda)\,d\Omega$$

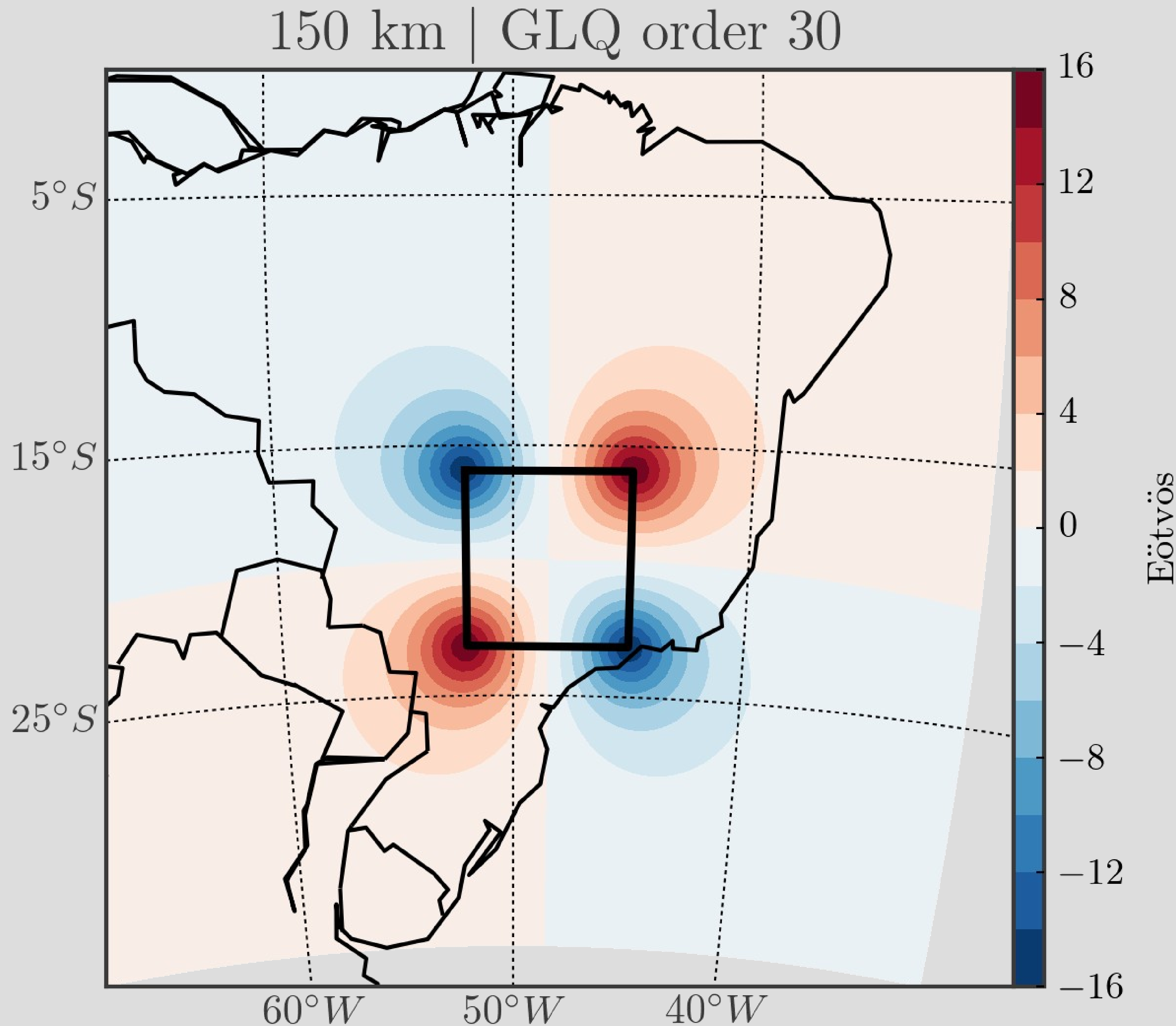
$$A\sum_{i=1}^{N_r}\sum_{j=1}^{N_\varphi}\sum_{k=1}^{N_\lambda} W_i W_j W_k \boxed{K_z(r_i,\varphi_j,\lambda_k)}$$

massa pontual


400 km | GLQ order 2
5°S
15°S
25°S
60°W
50°W
40°W
3.6
2.4
1.2
0.0
−1.2
−2.4
−3.6
Eötvös

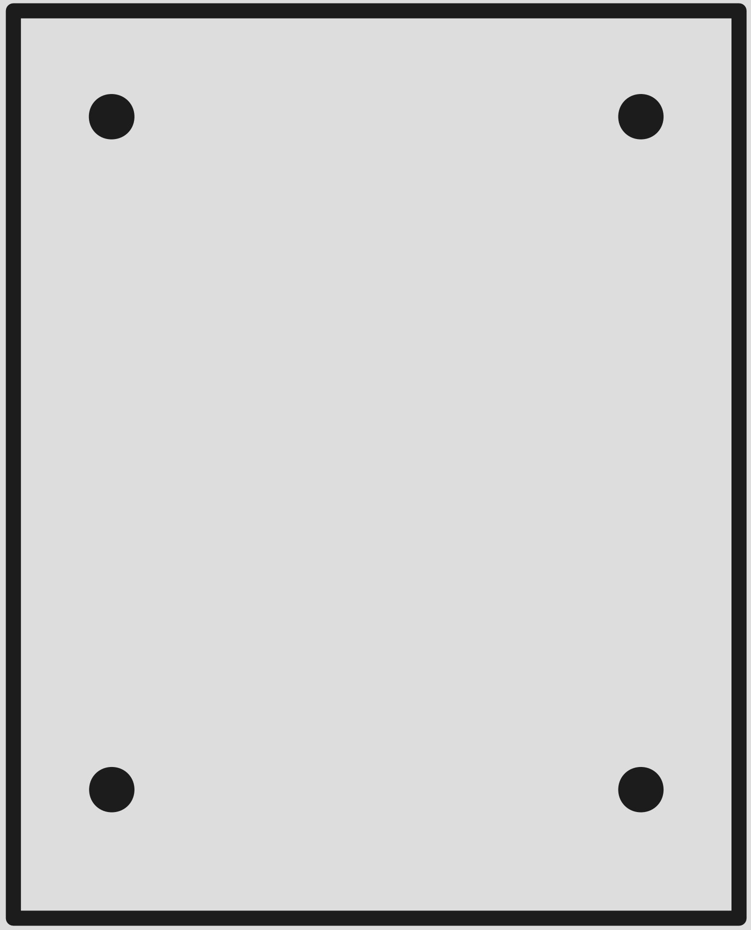


150 km | GLQ order 2
5° S
15° S
25° S
60° W
50° W
40° W
36
24
12
0
−12
−24
−36
Eötvös

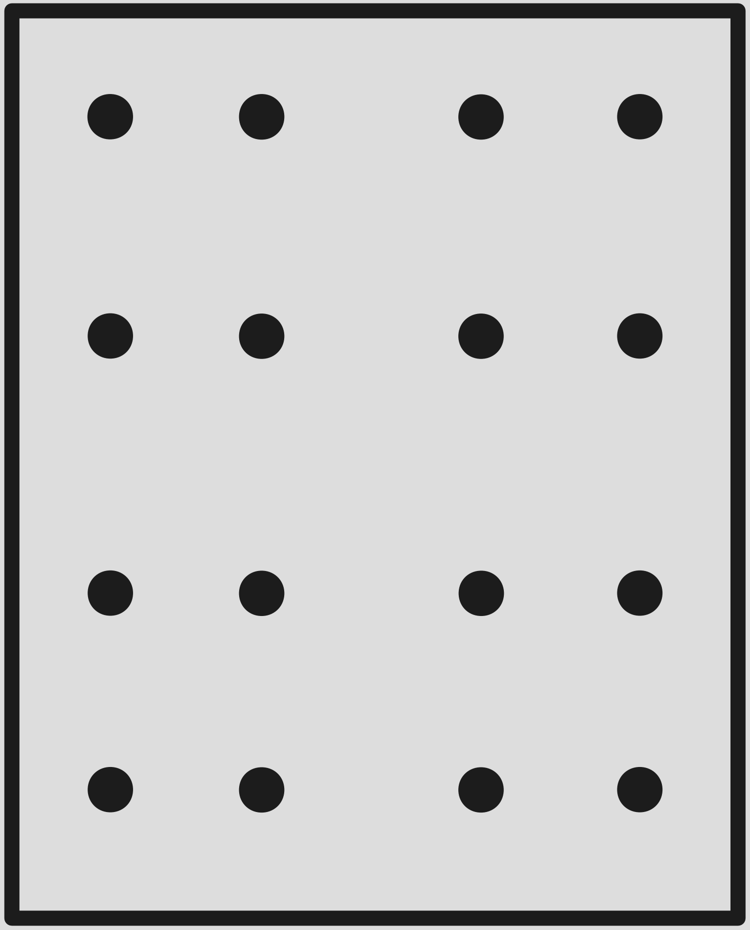

150 km | GLQ order 30
5°S
15°S
25°S
60°W
50°W
40°W
16
12
8
4
0
−4
−8
−12
−16
Eötvös

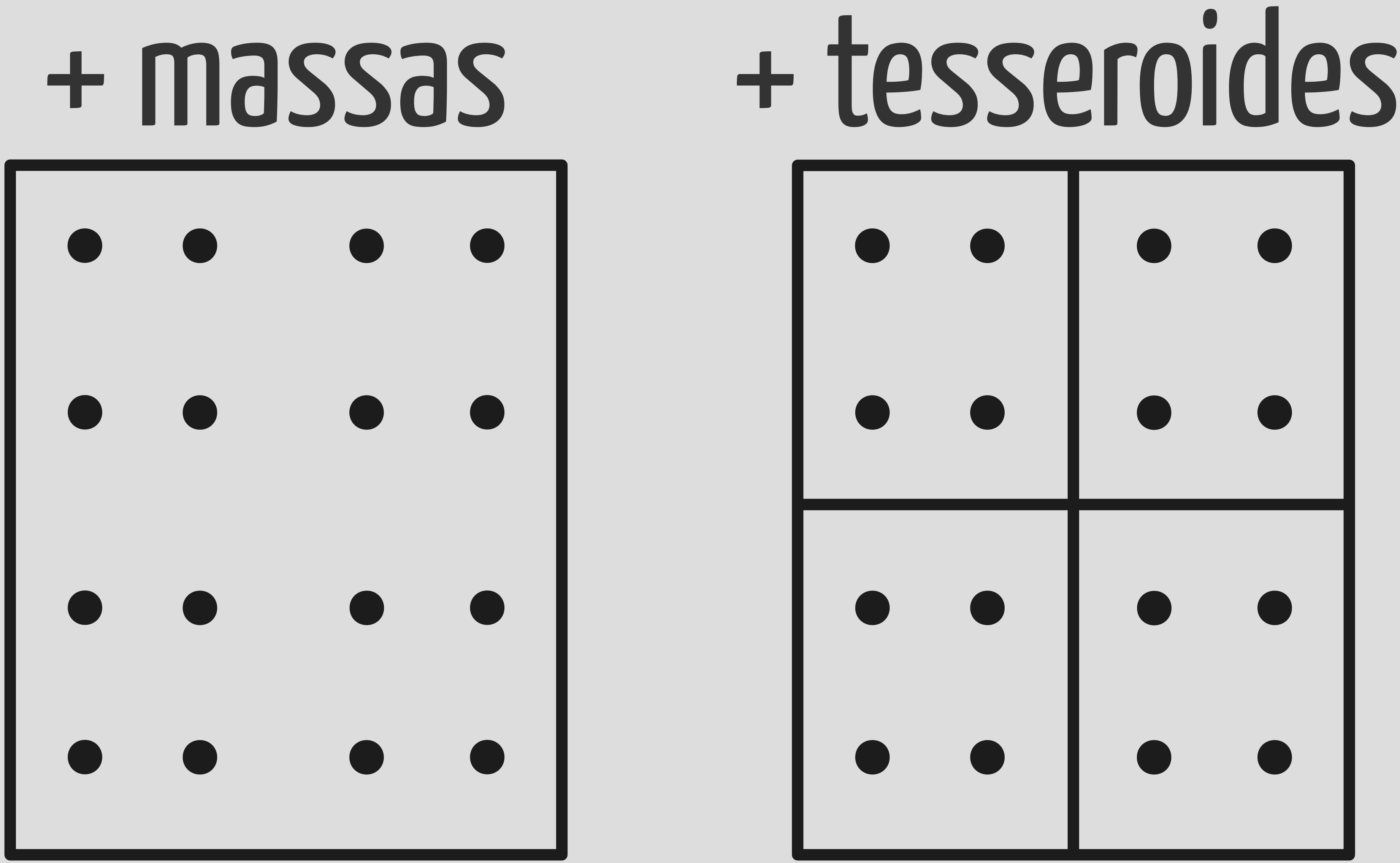
Ku (1977)

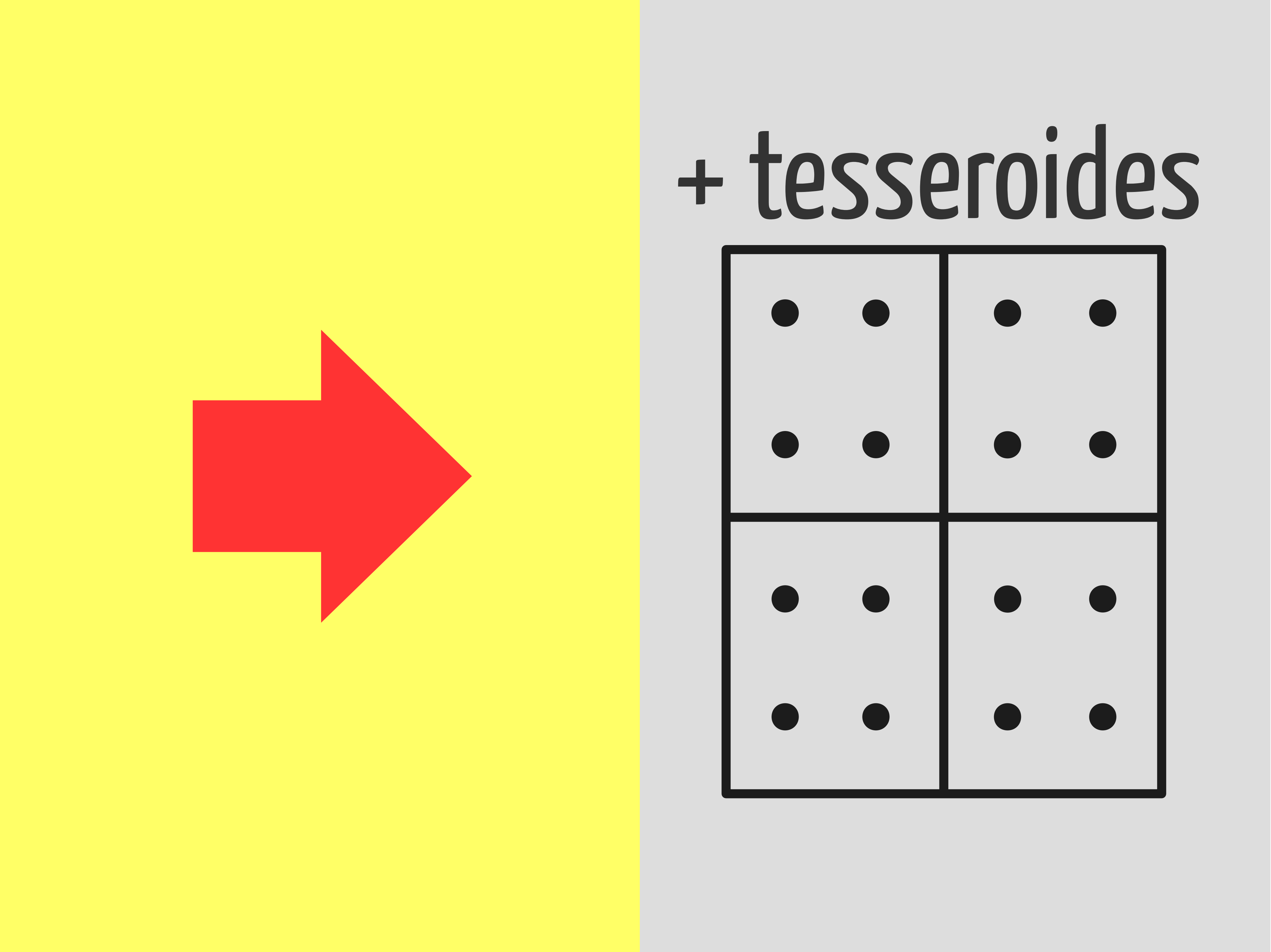
dist. até observação $<$ dist. entre massas

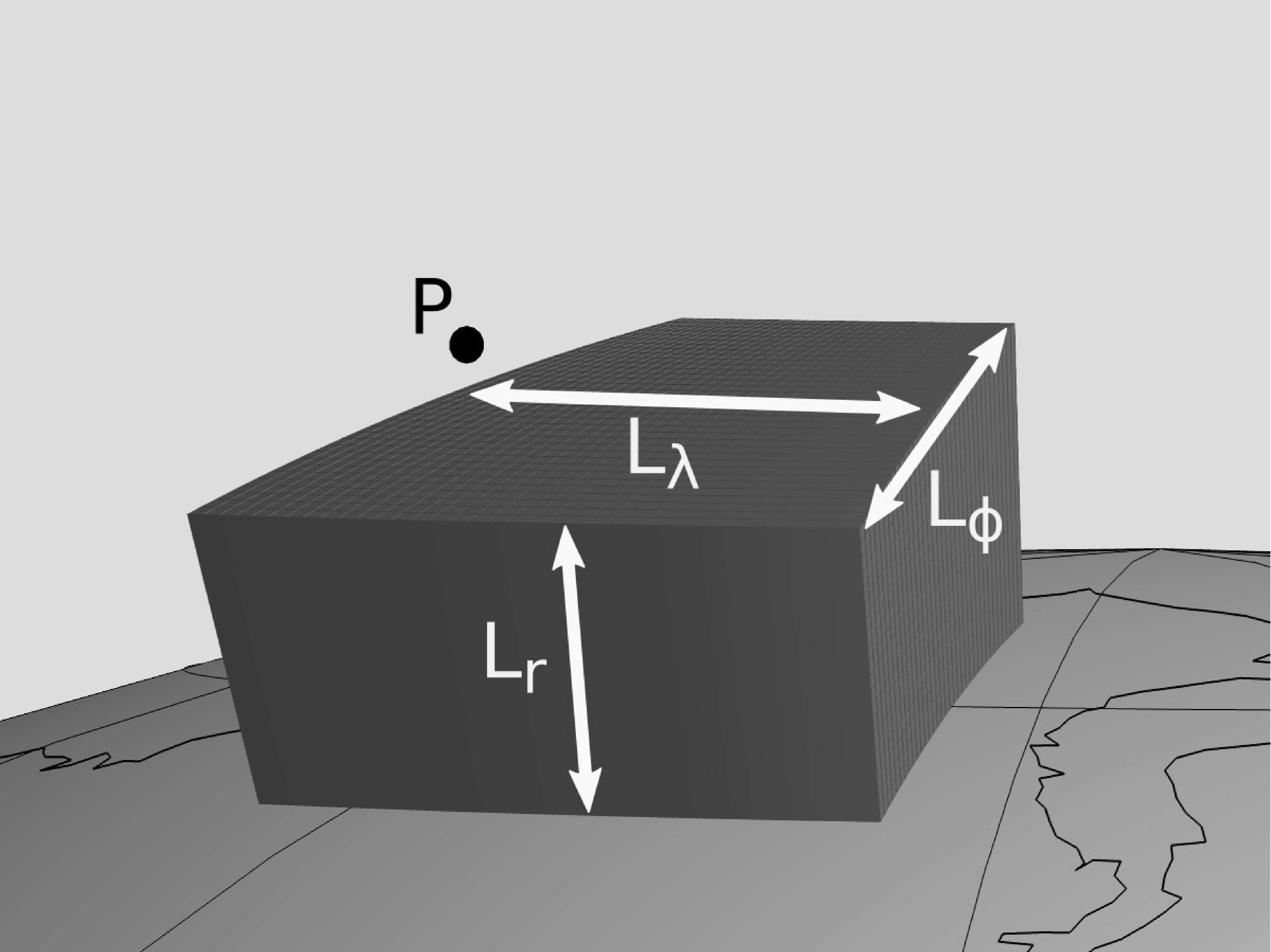


+ massas

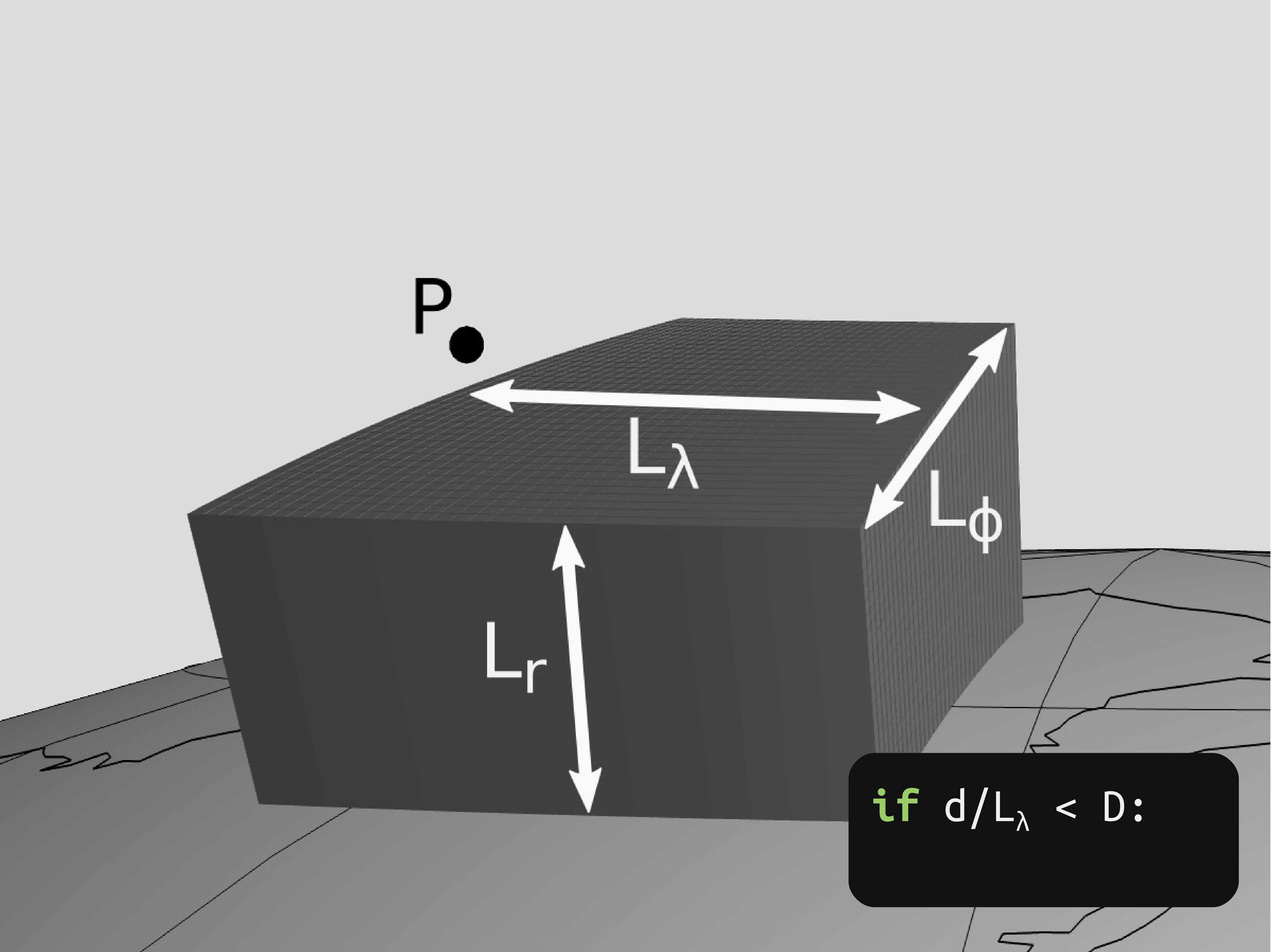


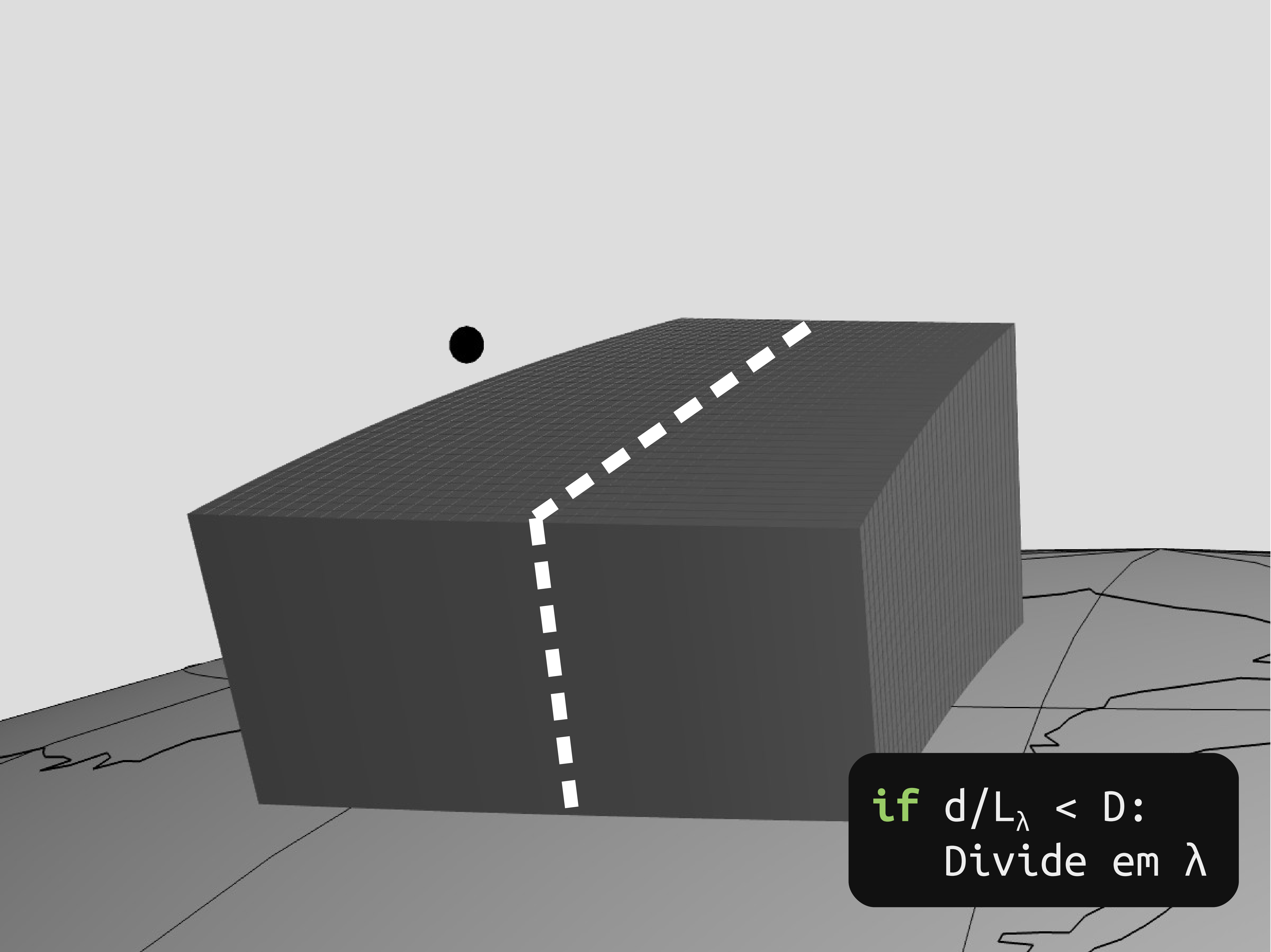

+ massas
+ tesseroides


+ tesseroides

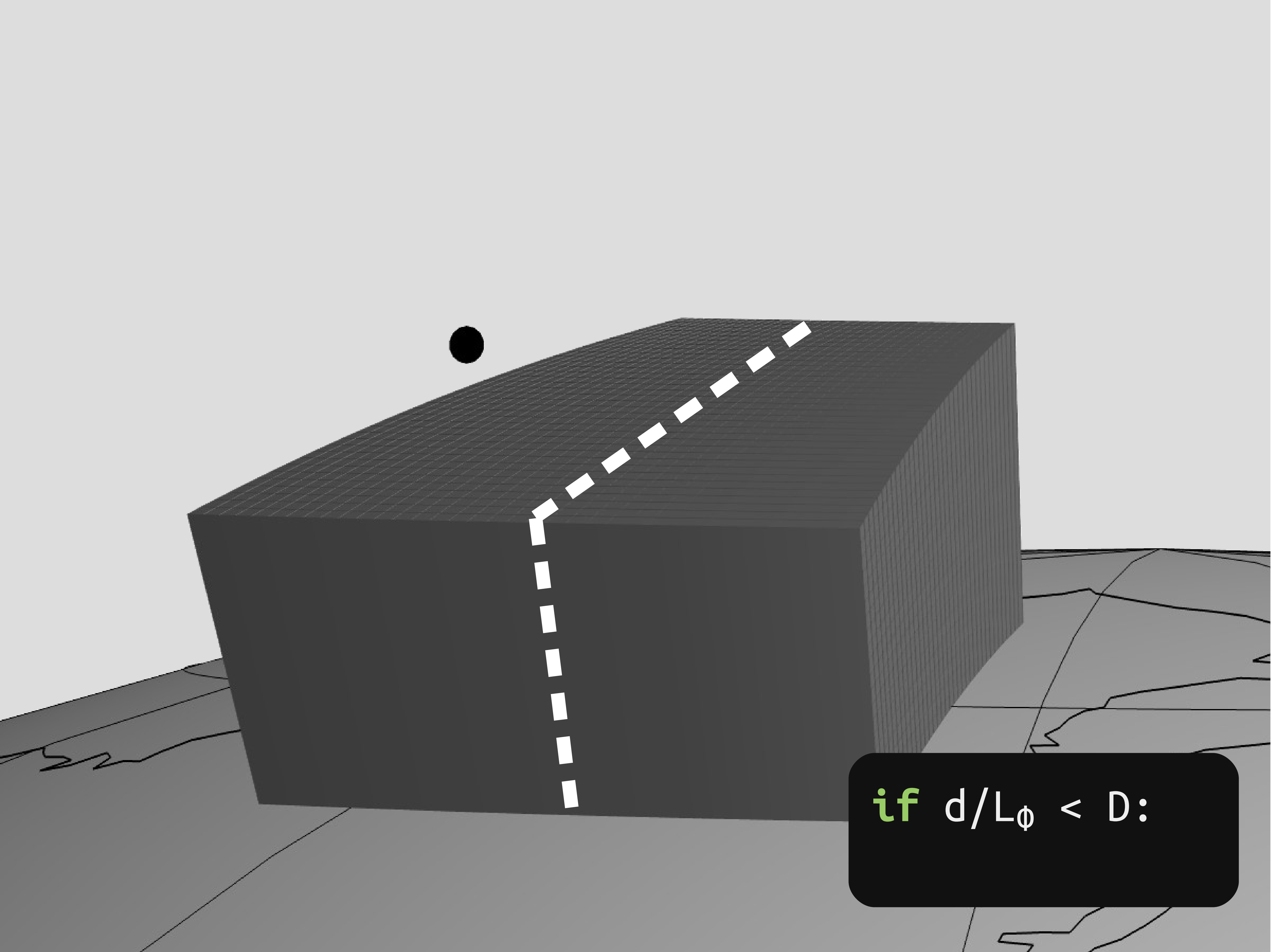


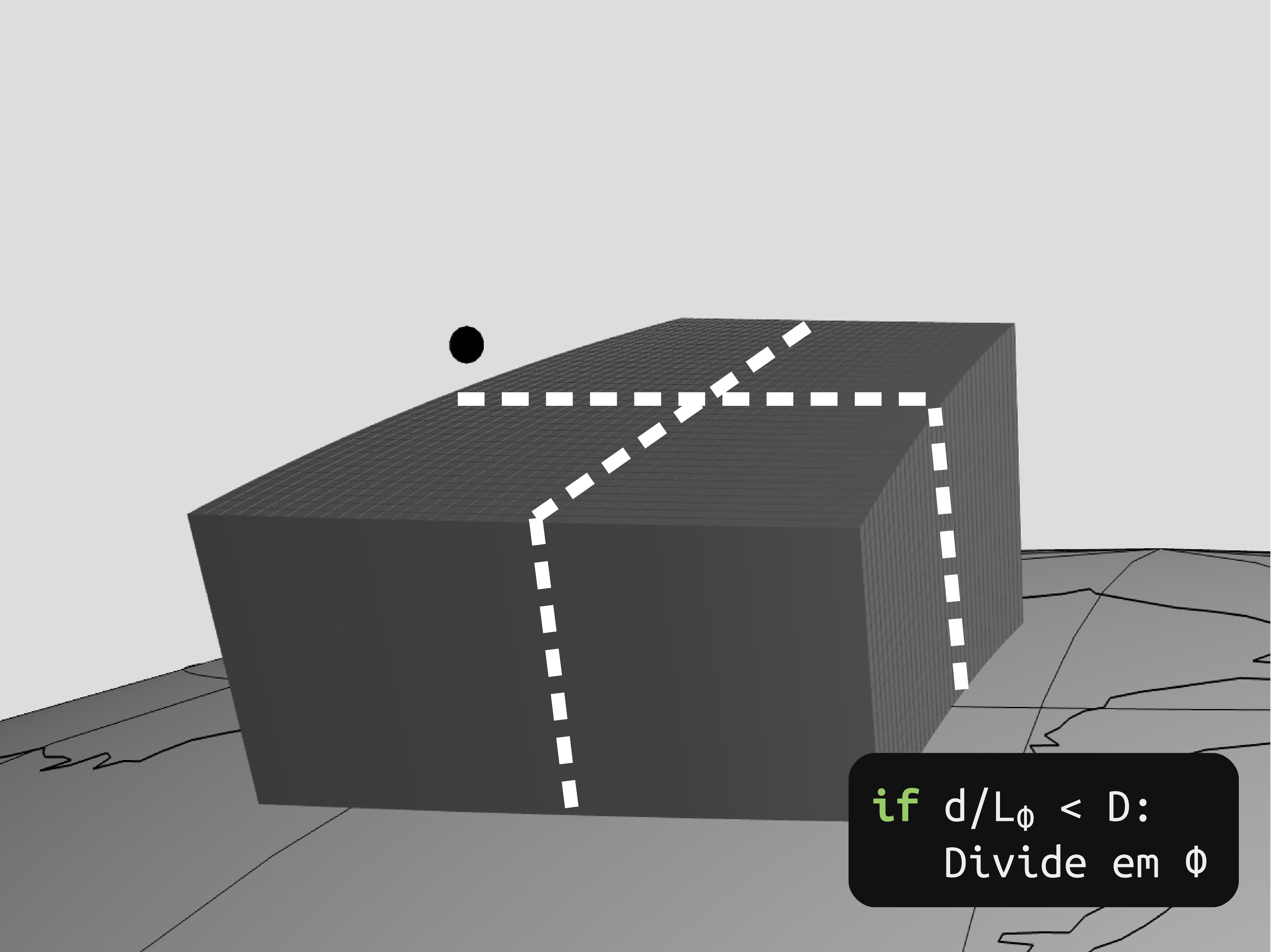
P
L_λ
L_ϕ
L_r

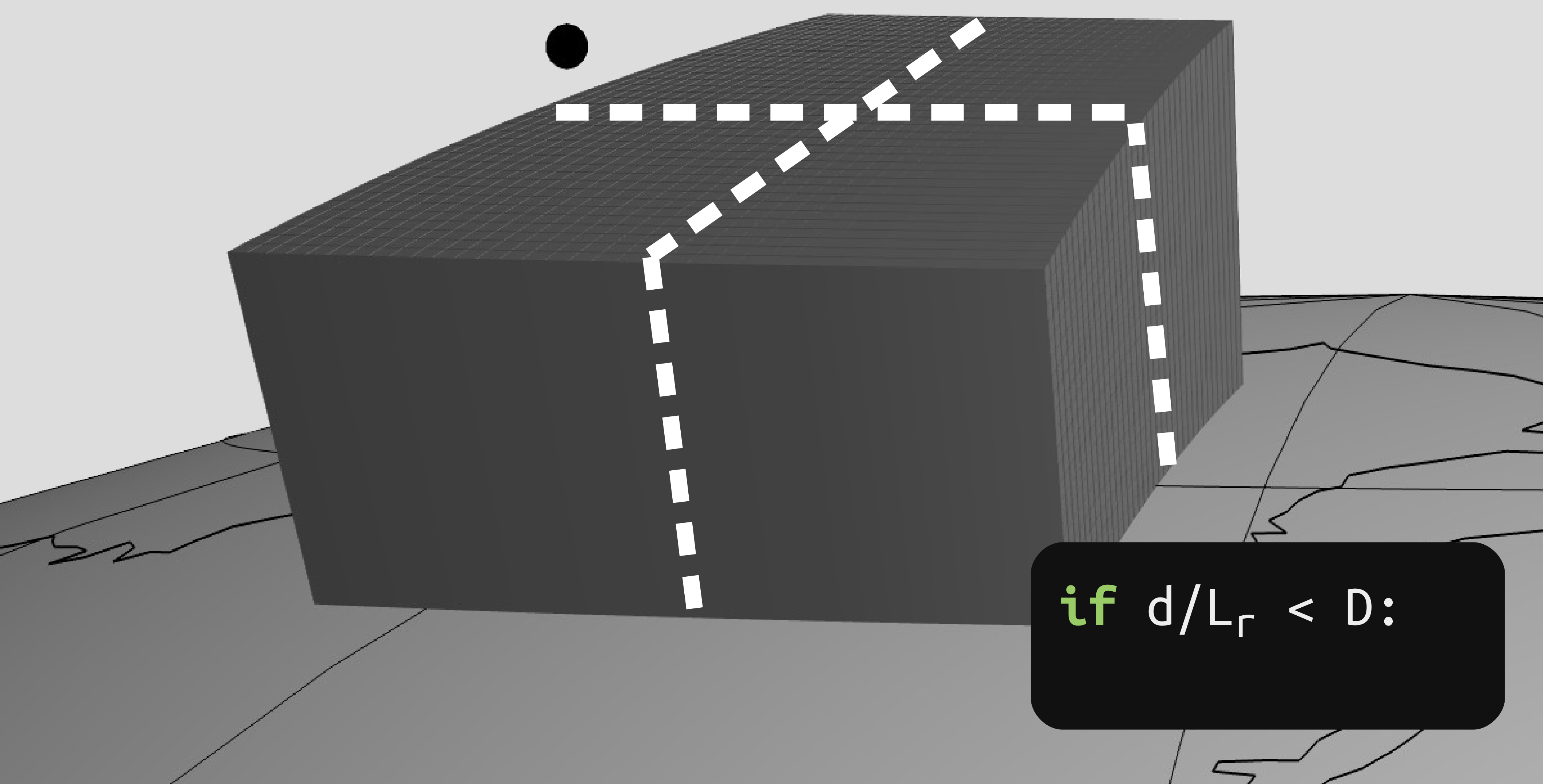

P
L_λ
L_ϕ
L_r
if d/L_λ < D:

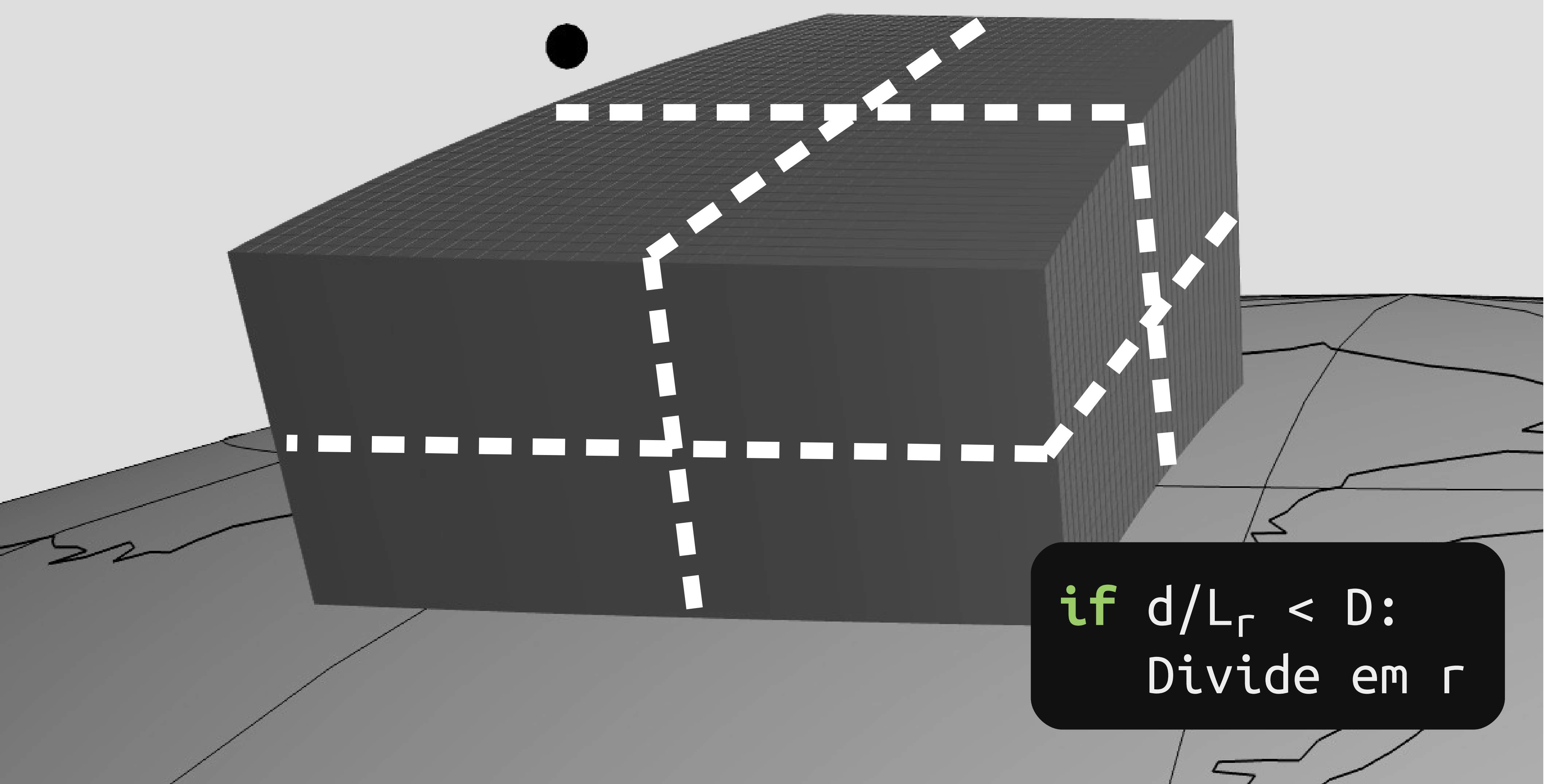


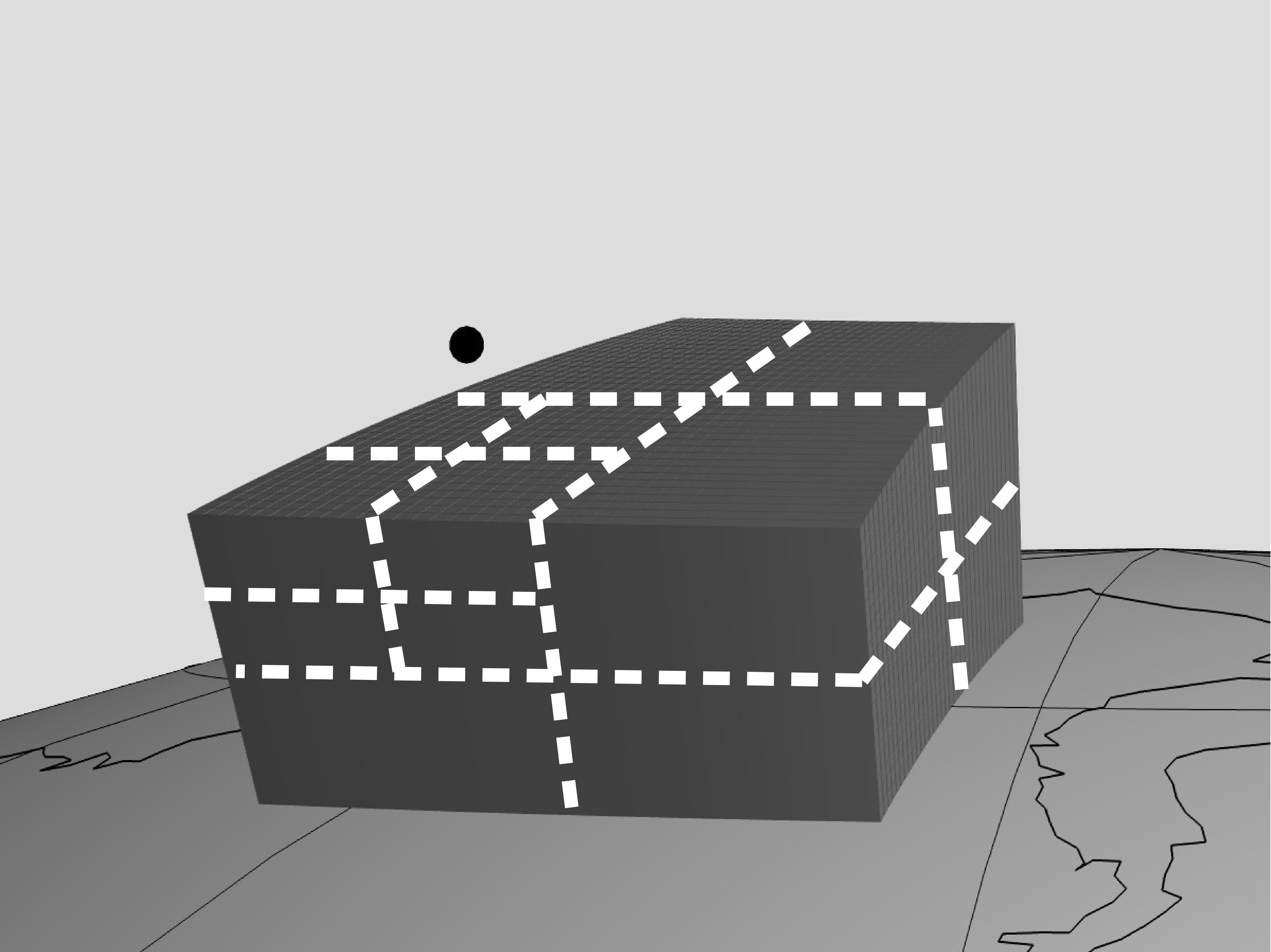
if d/Lλ < D:
Divide em λ

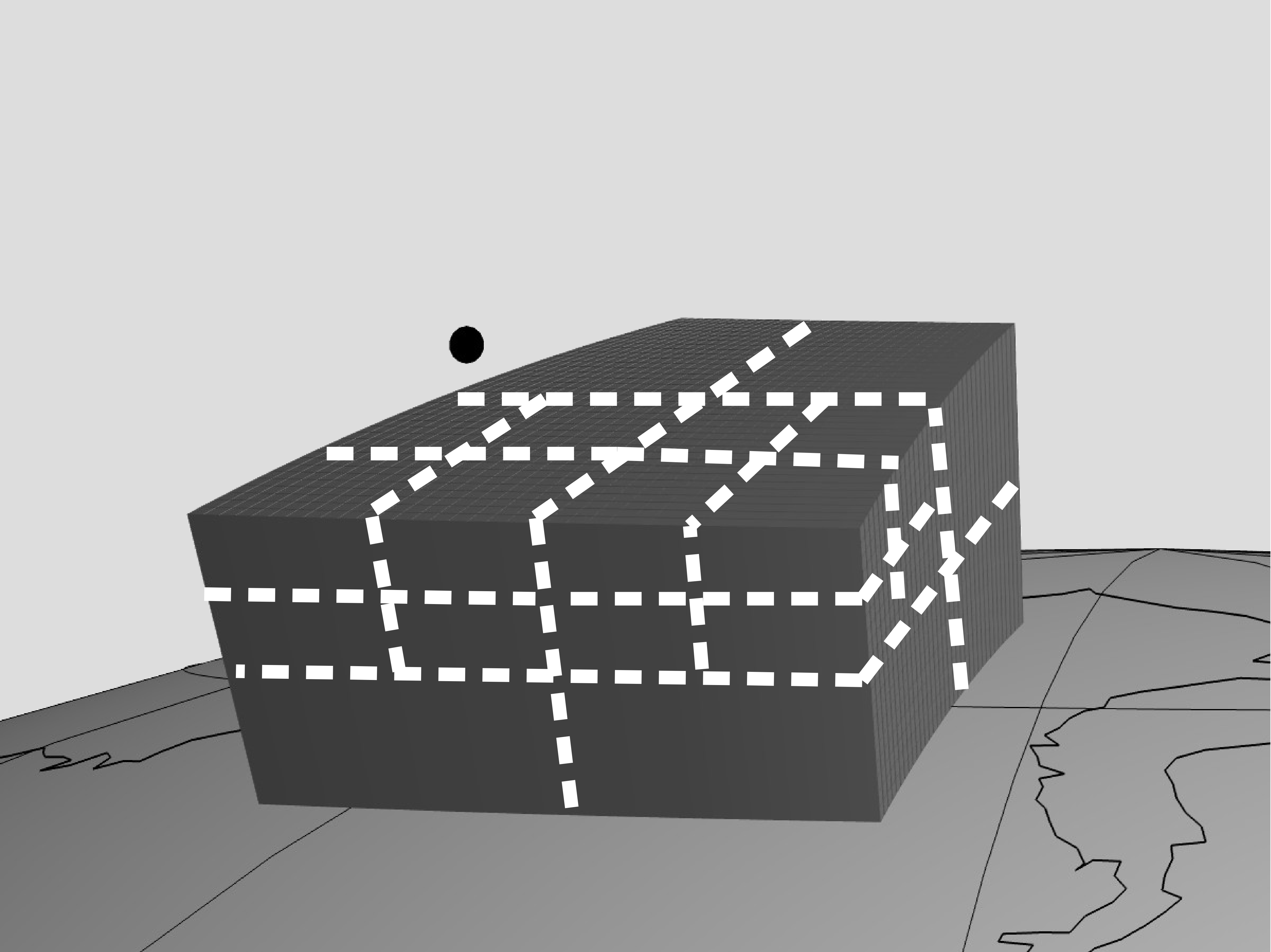

if d/Lϕ < D:

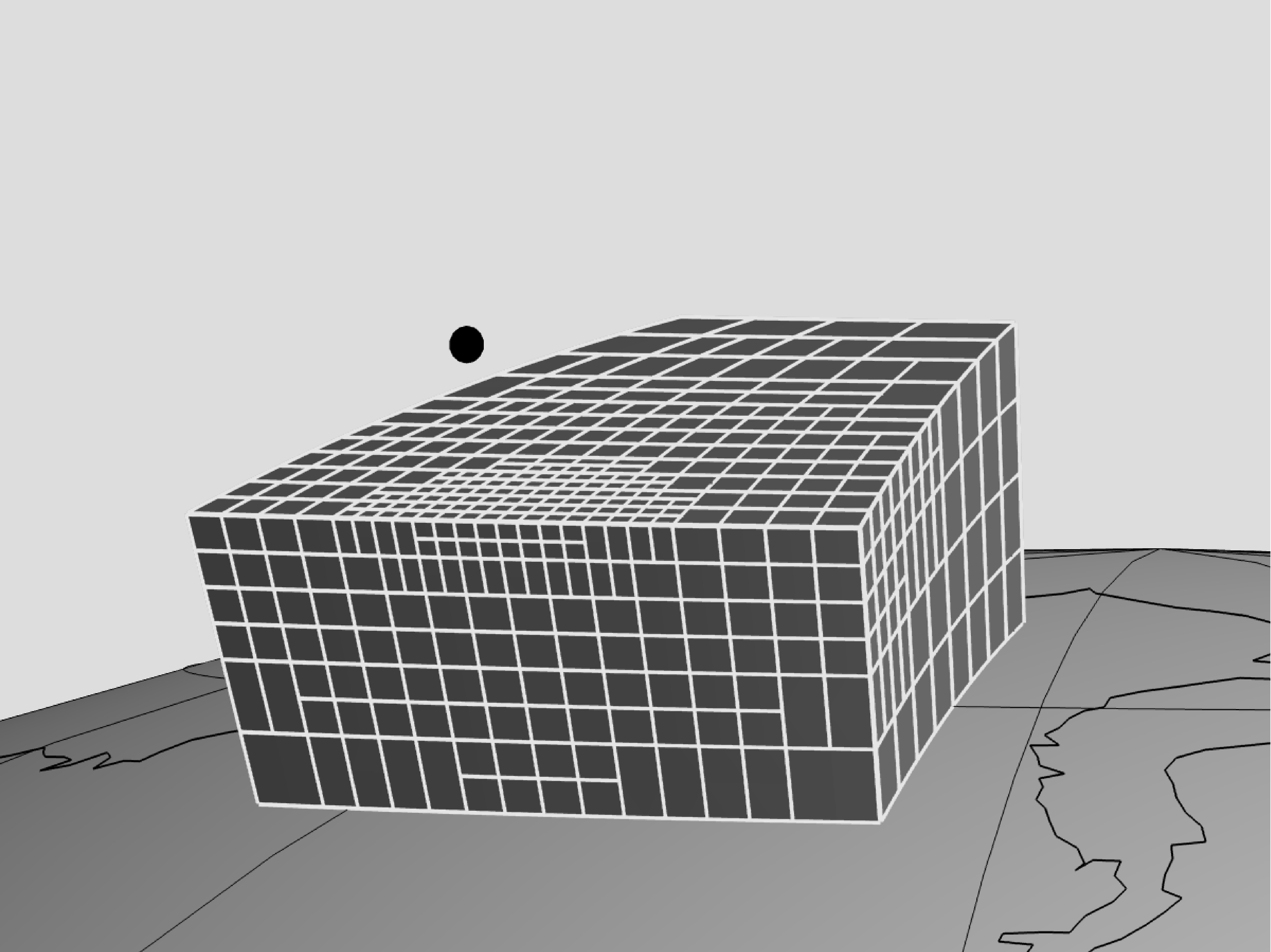

if d/Lϕ < D:
Divide em ϕ


if d/Lr < D:

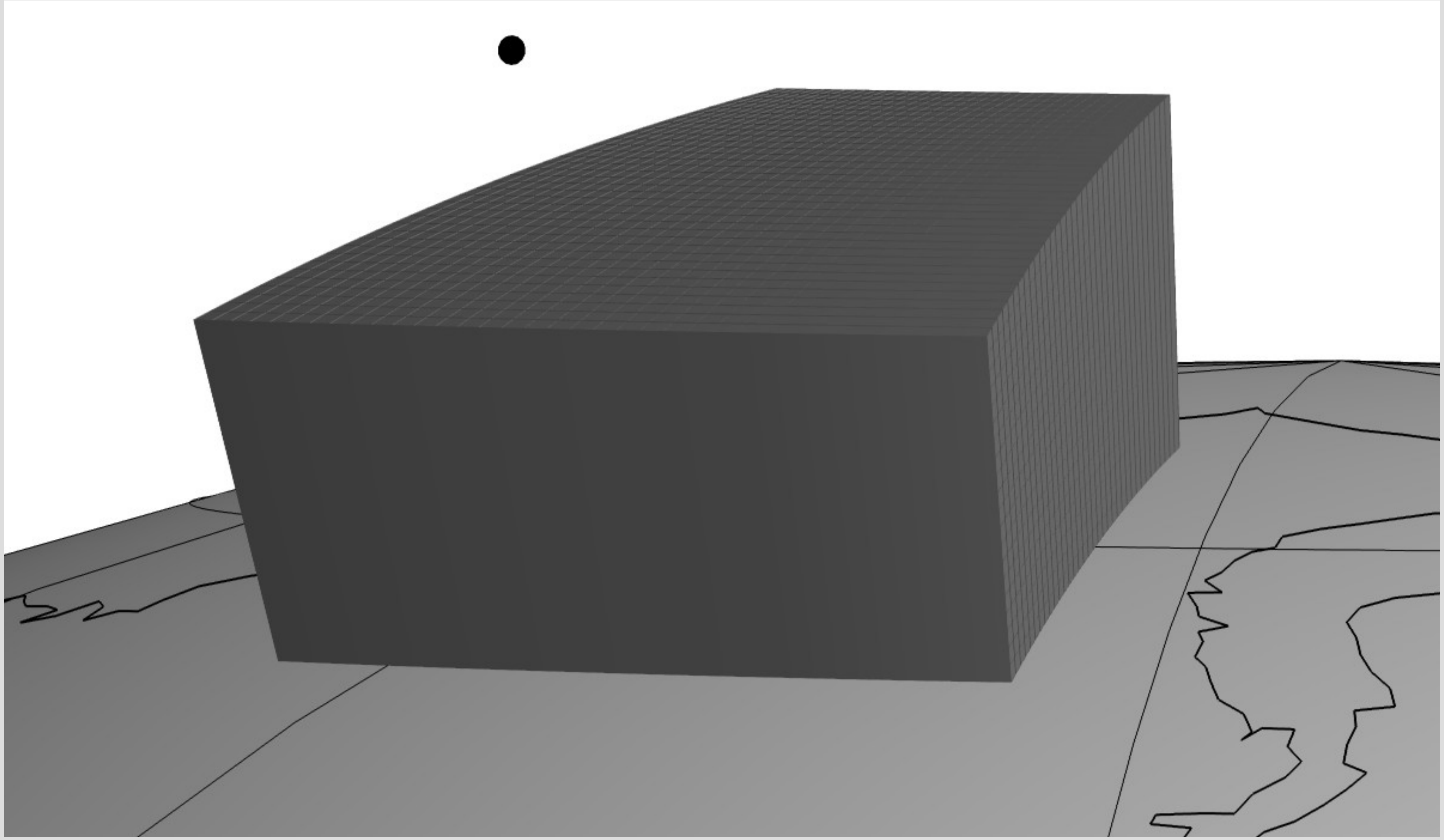

if d/Lr < D:
Divide em r



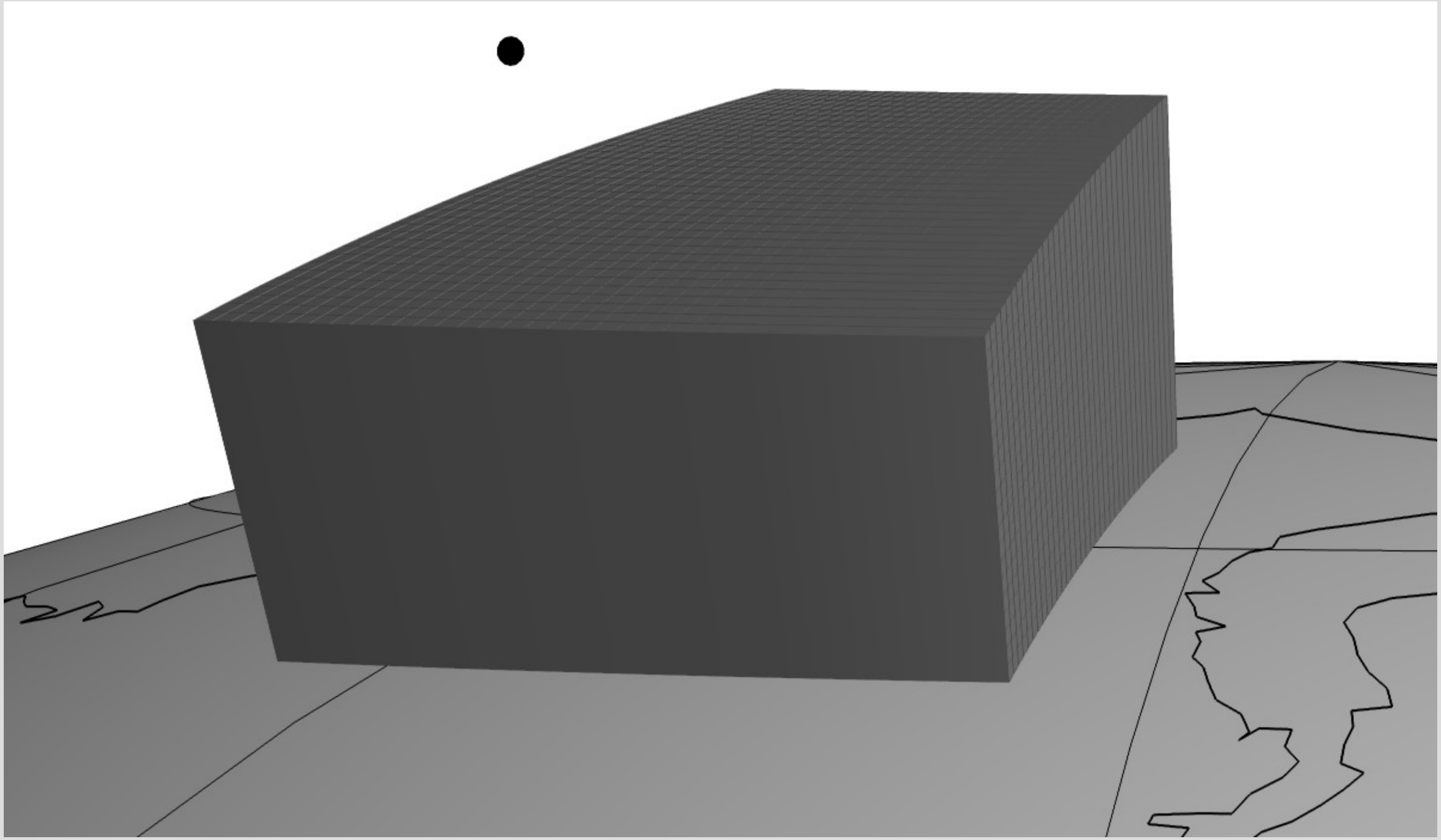




algoritmo

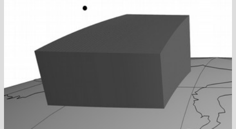


pilha



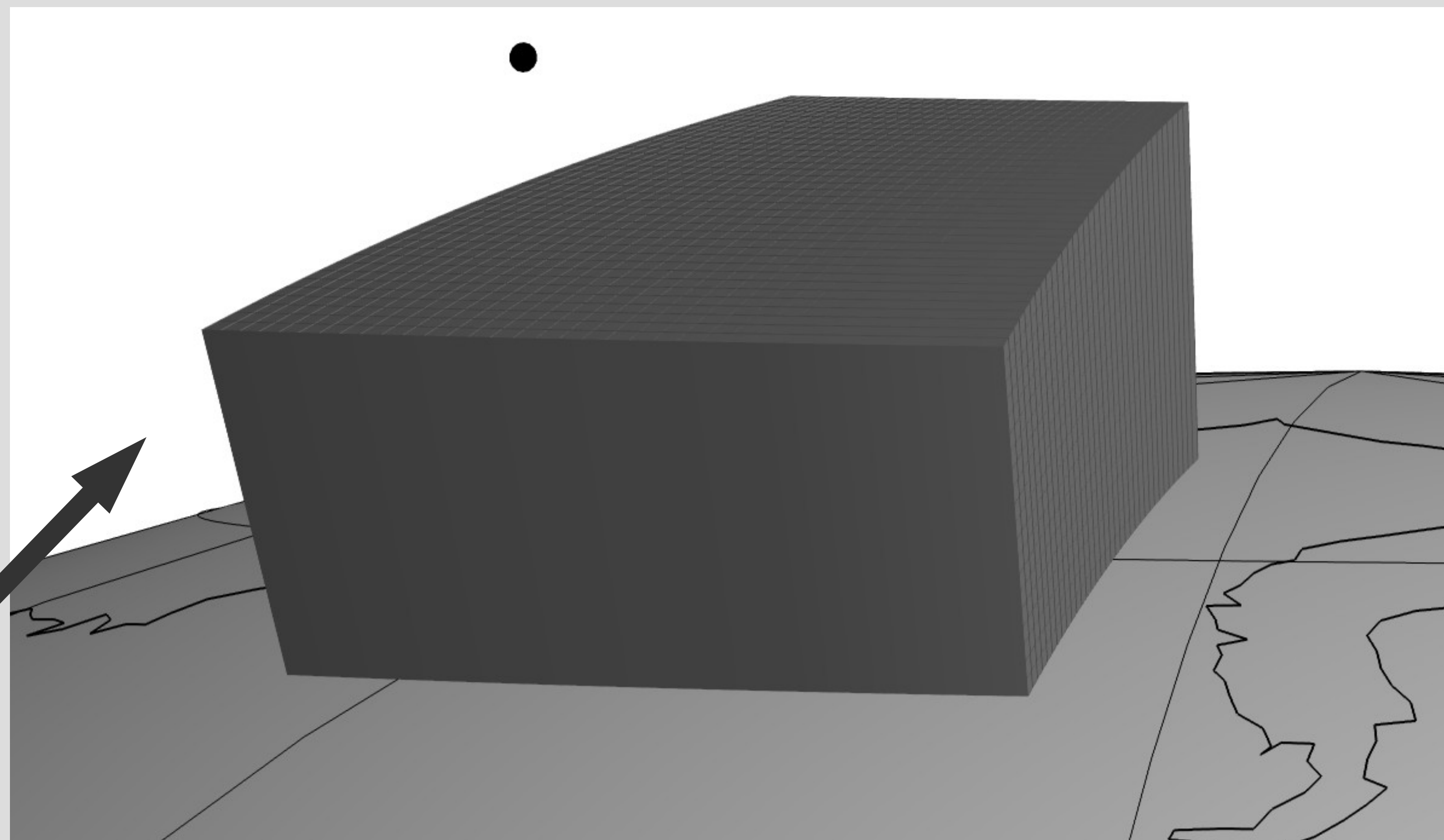
pilha

adicionar no
topo da pilha



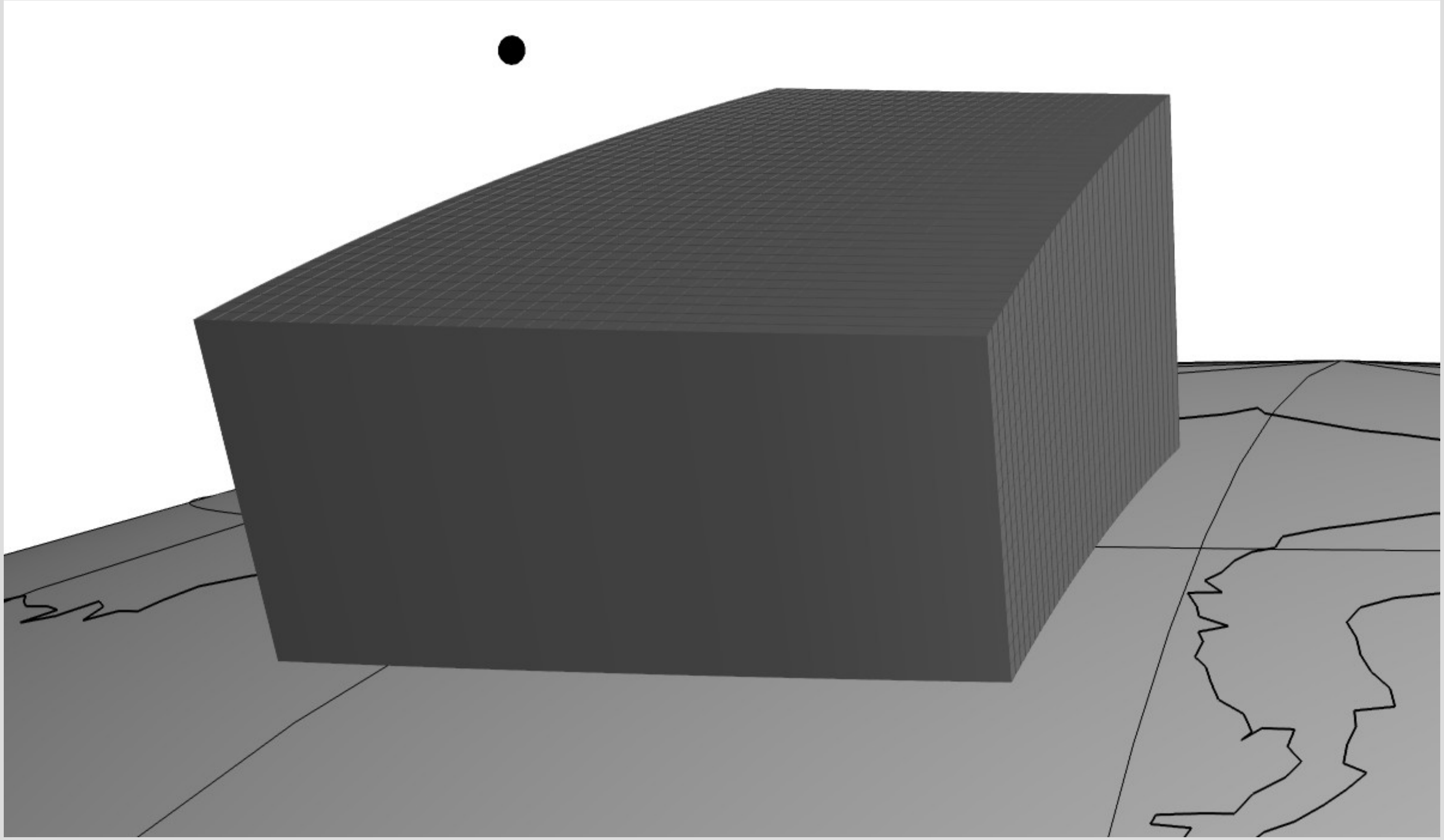
pilha

“pop”



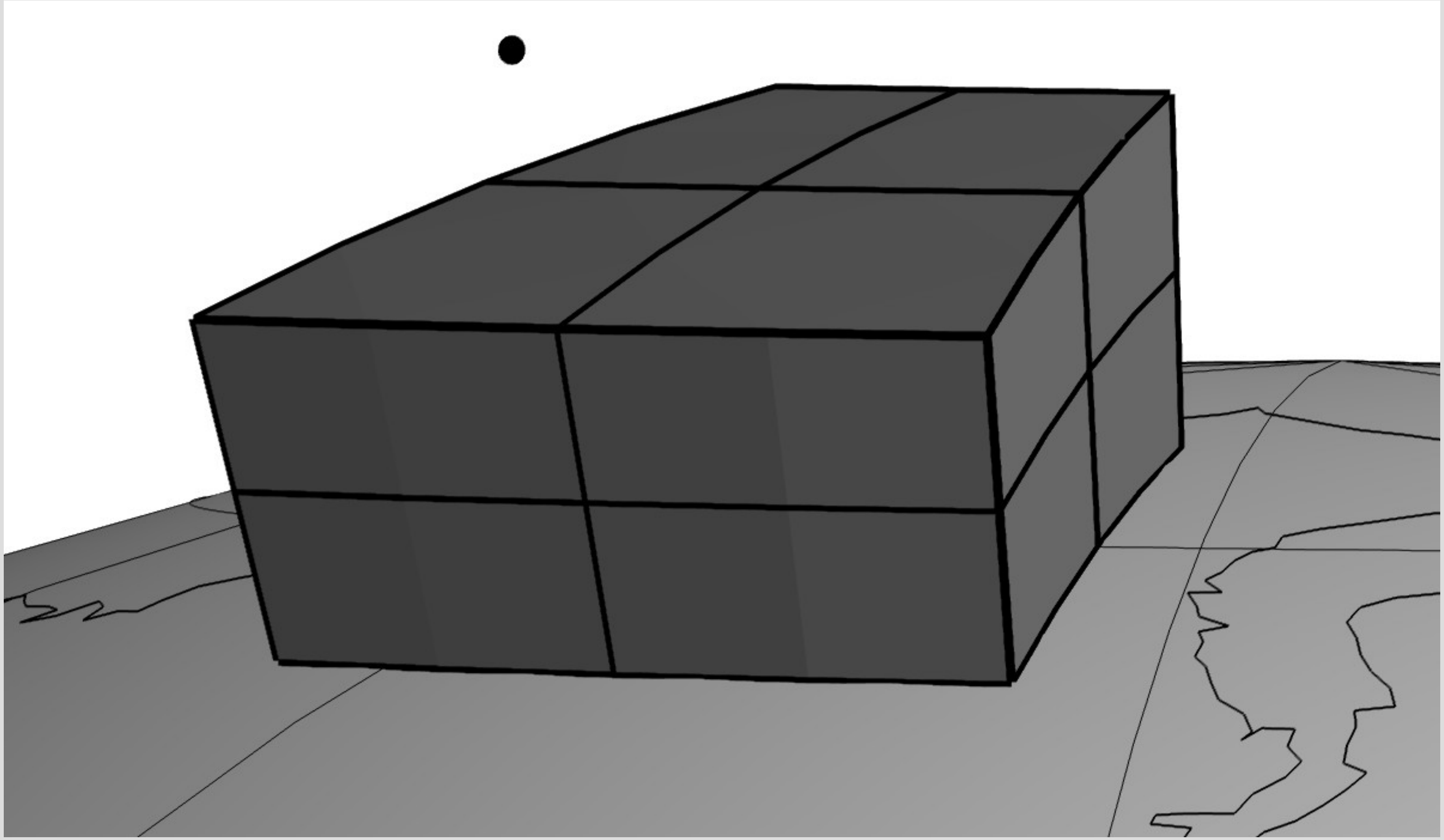
pilha

```
if d/Lx < D:
   Divide em x
```

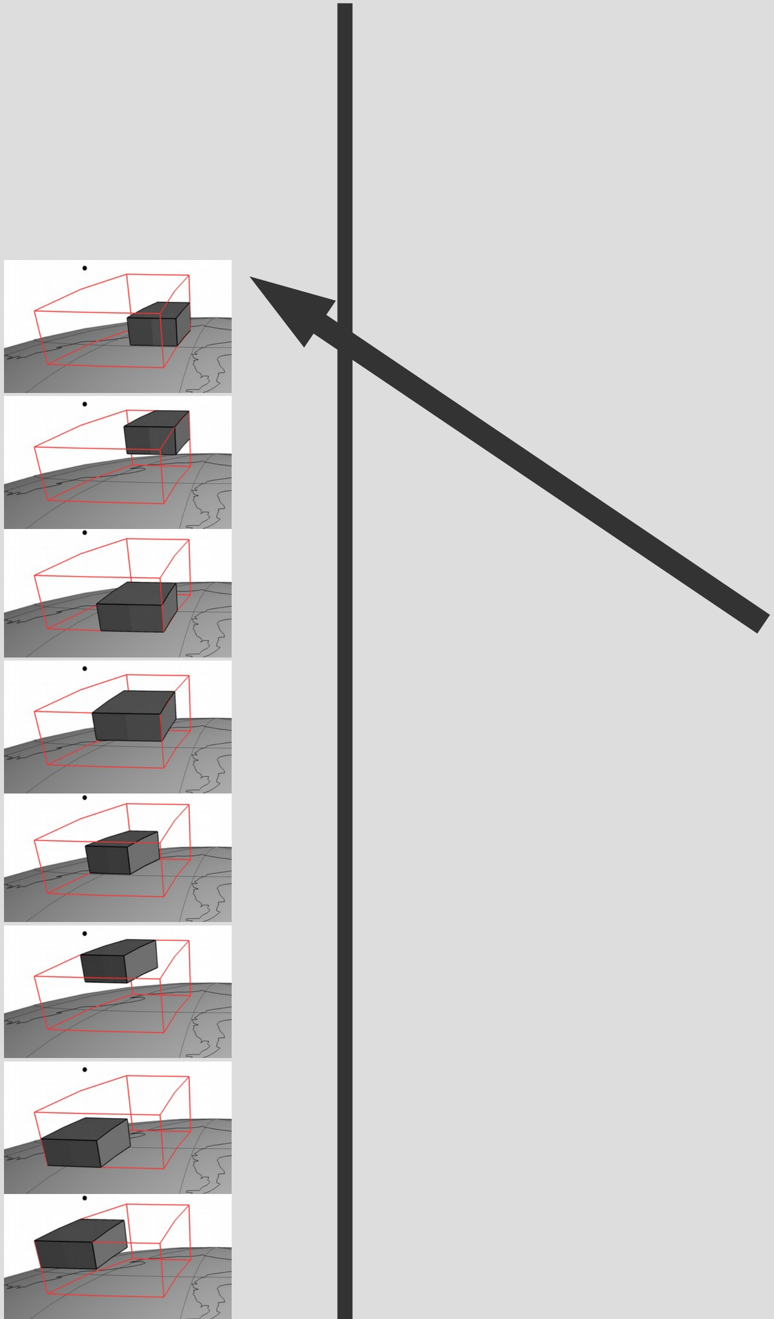


pilha

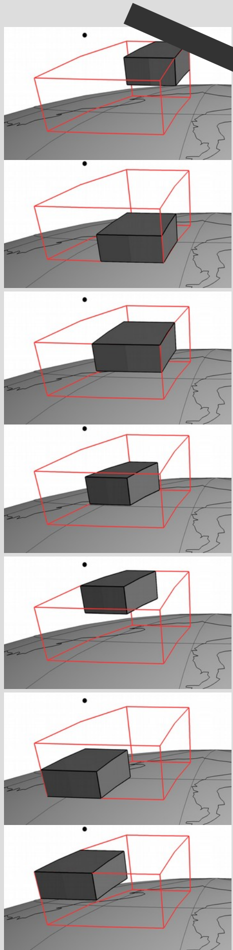
```
if d/Lx < D:
   Divide em x
```



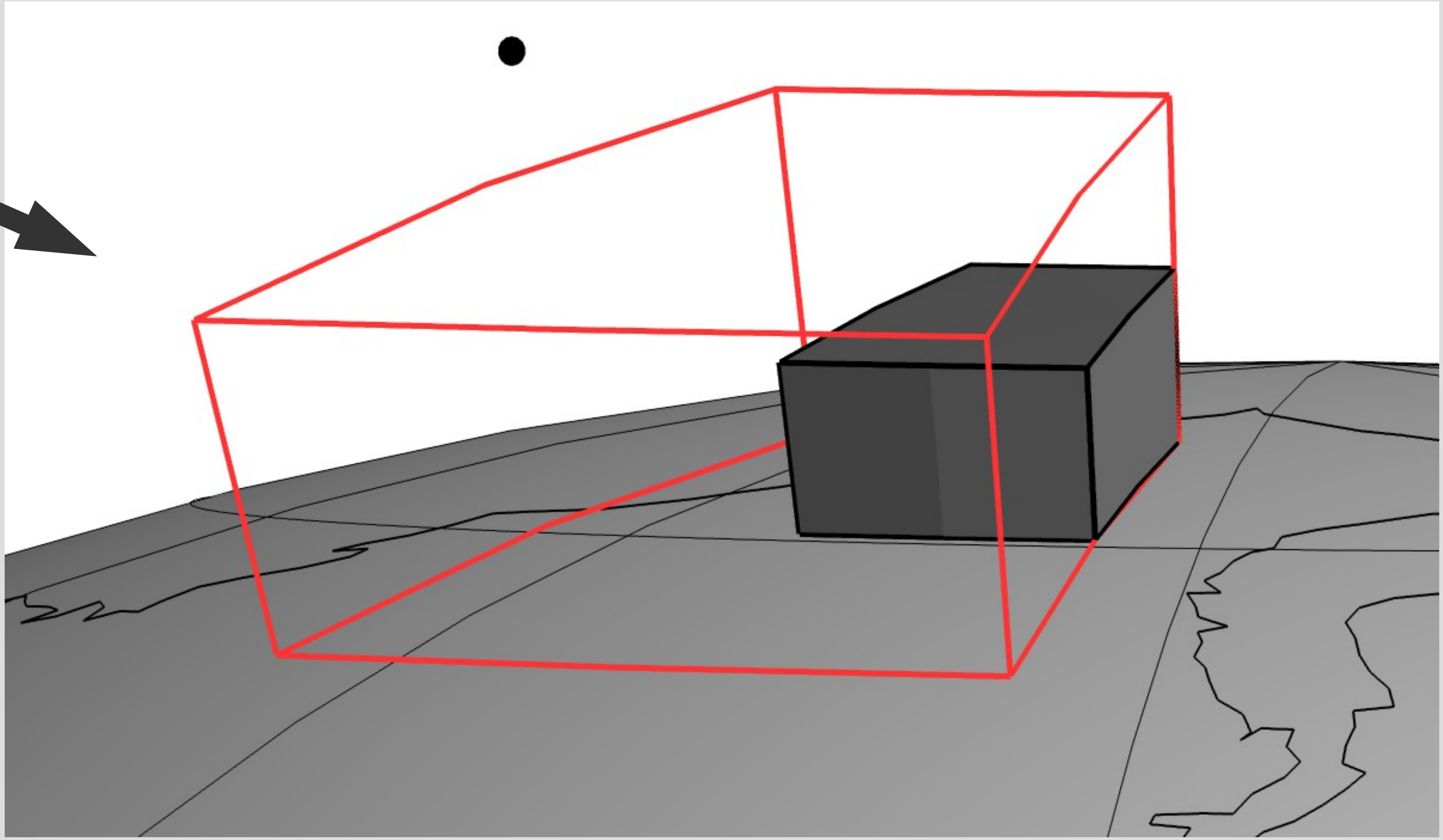
pilha



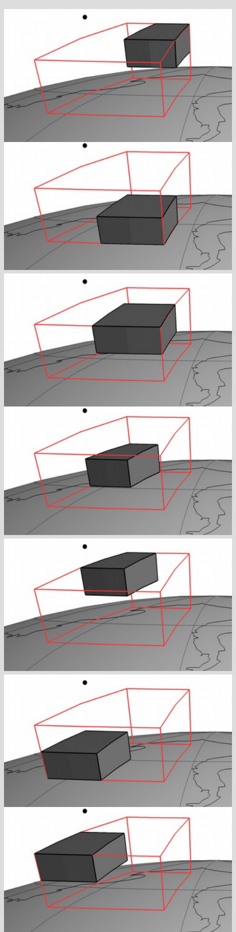
pilha



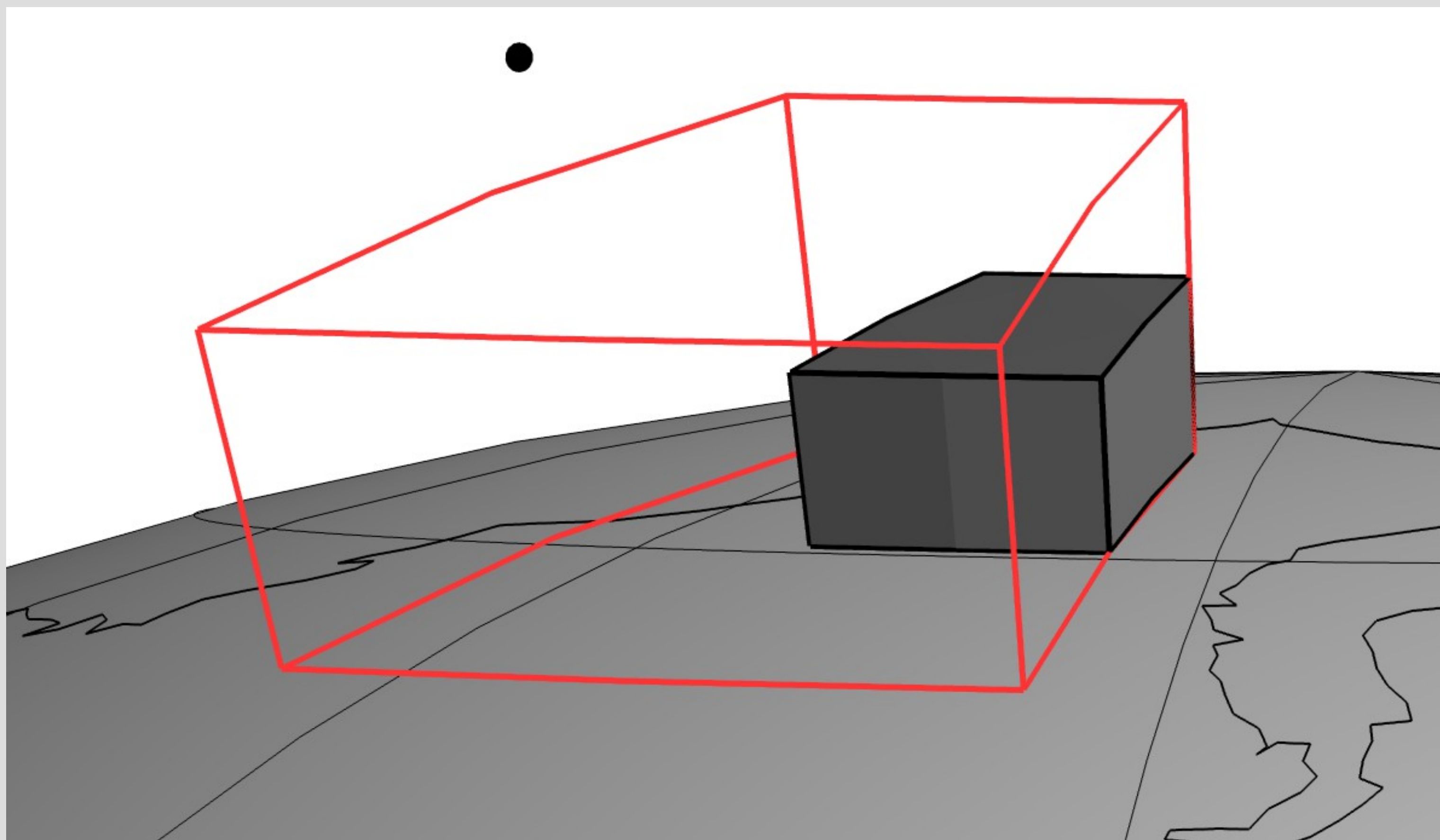
“pop”



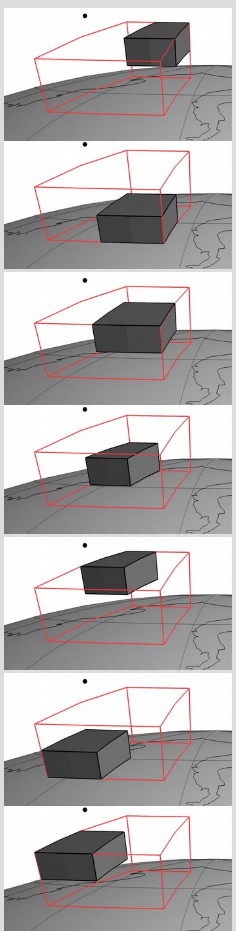
pilha



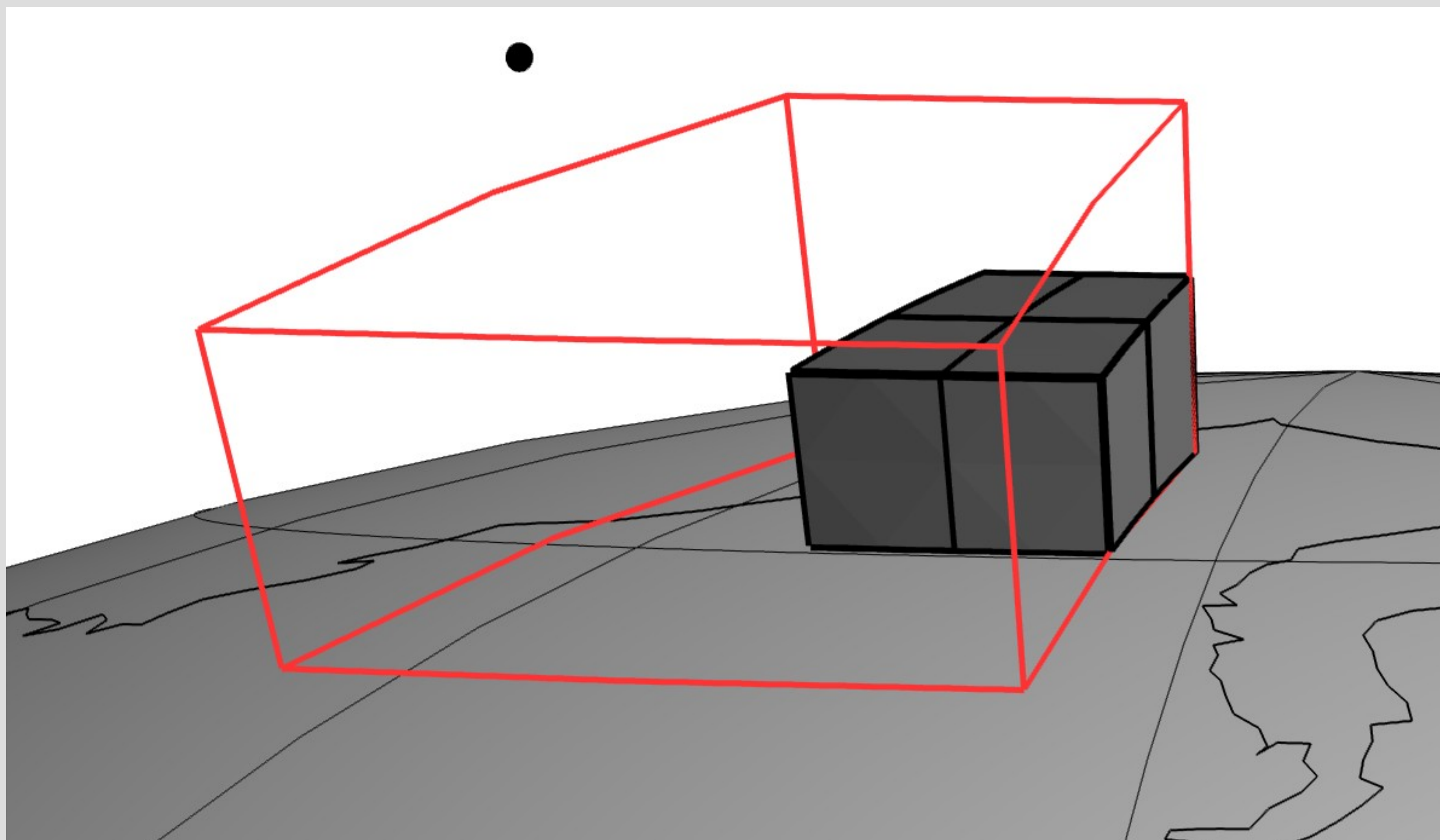
```
if d/Lx < D:
   Divide em x
```



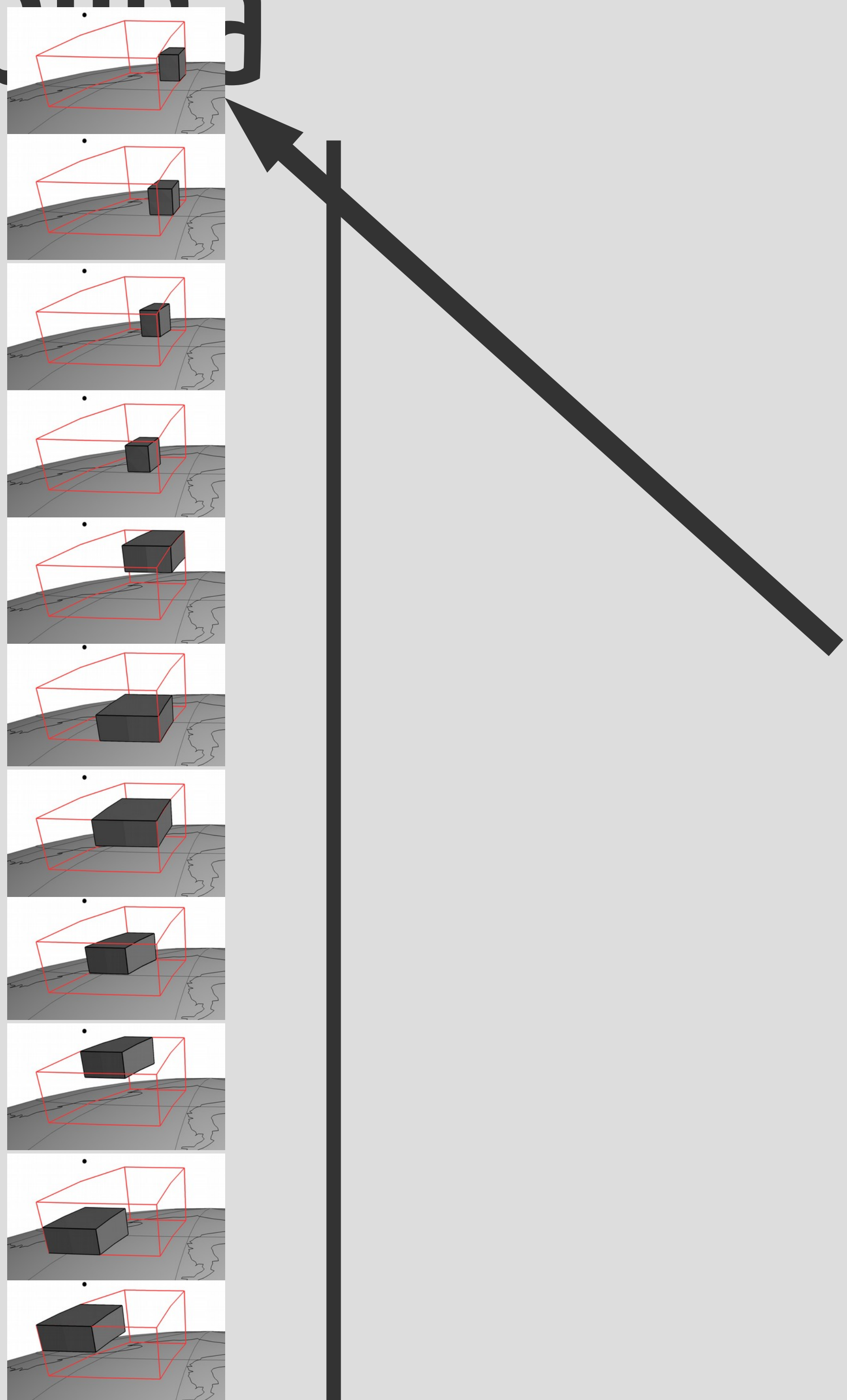
pilha



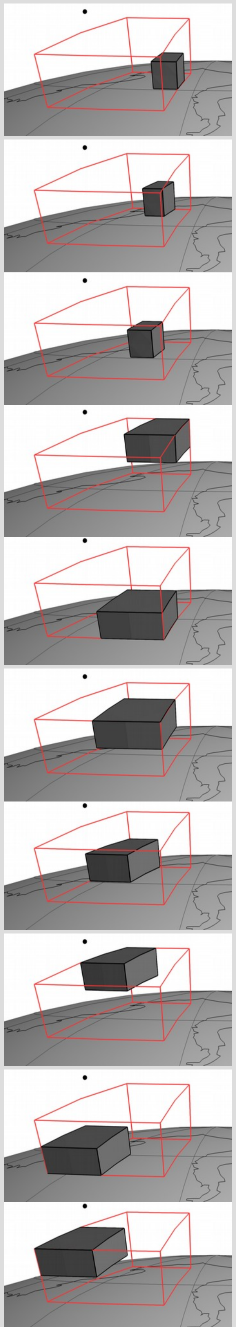
```
if d/L_x < D:
    Divide em x
```



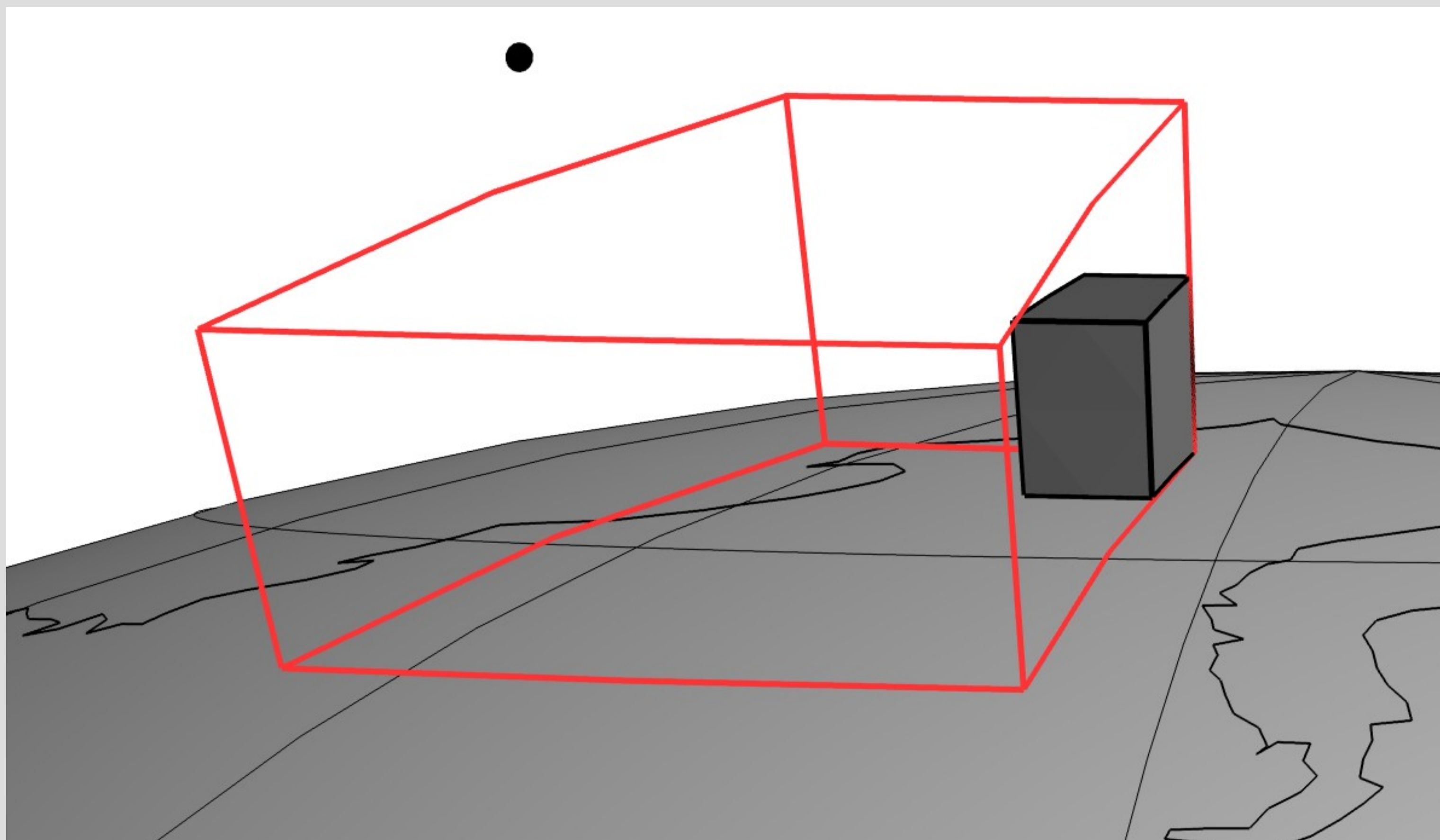
pilba



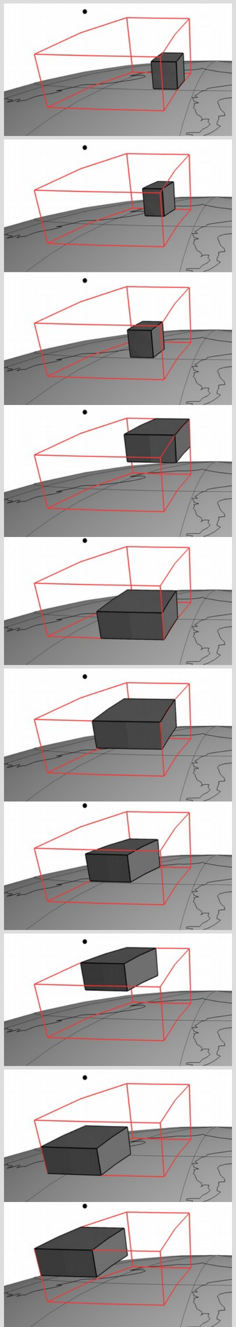
pilha



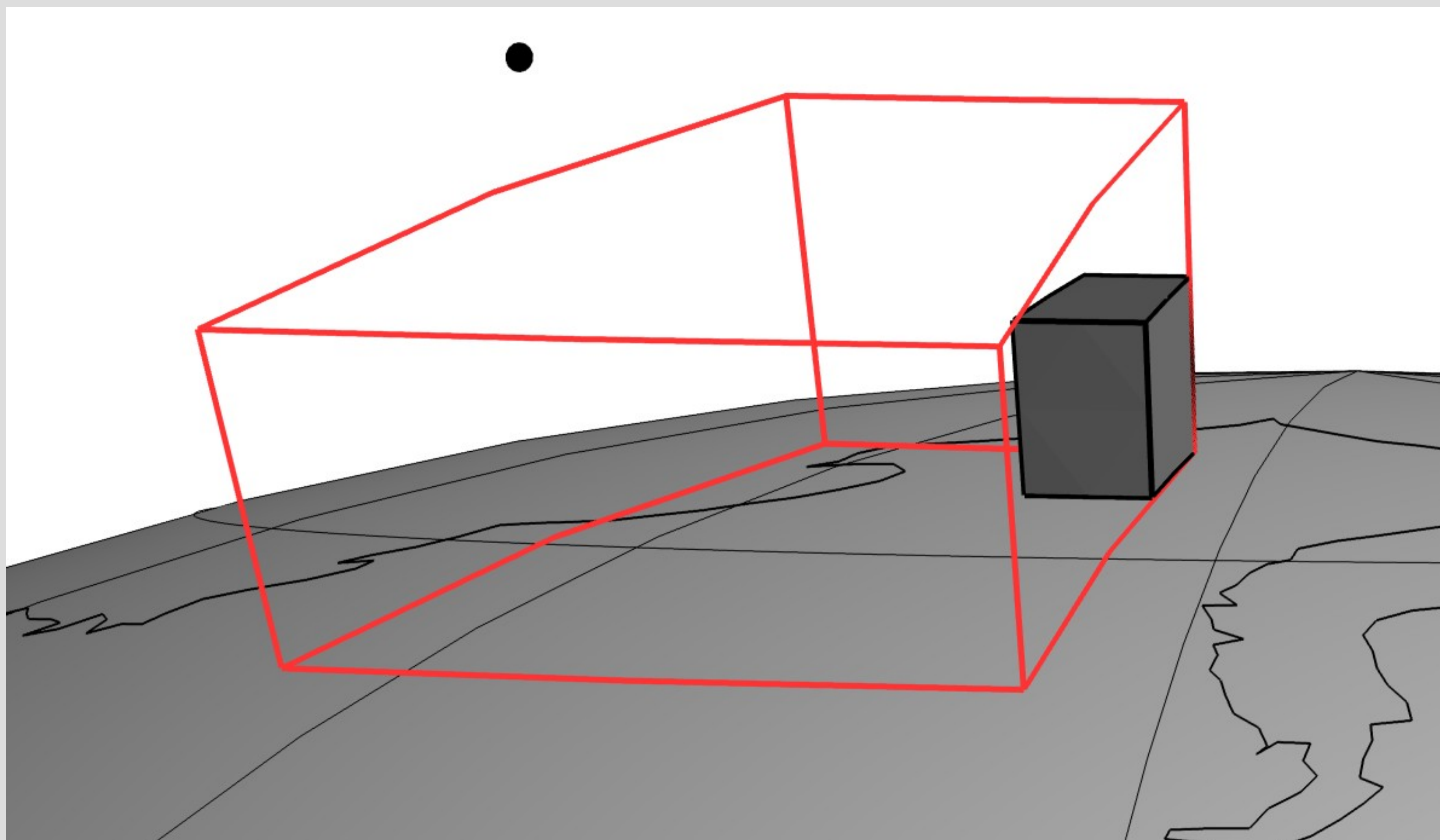
“pop”



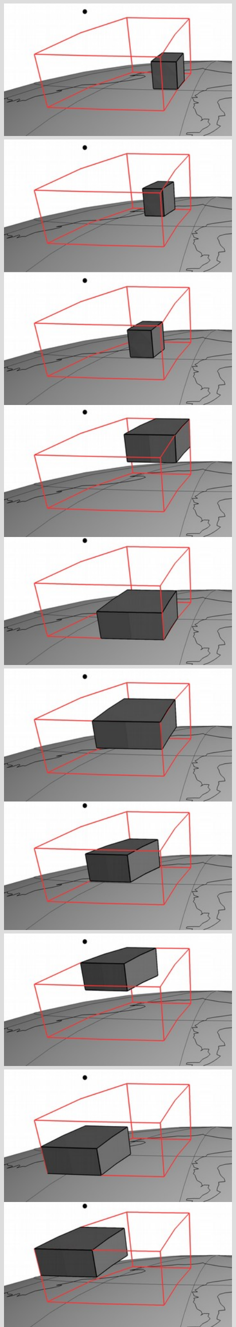
pilha



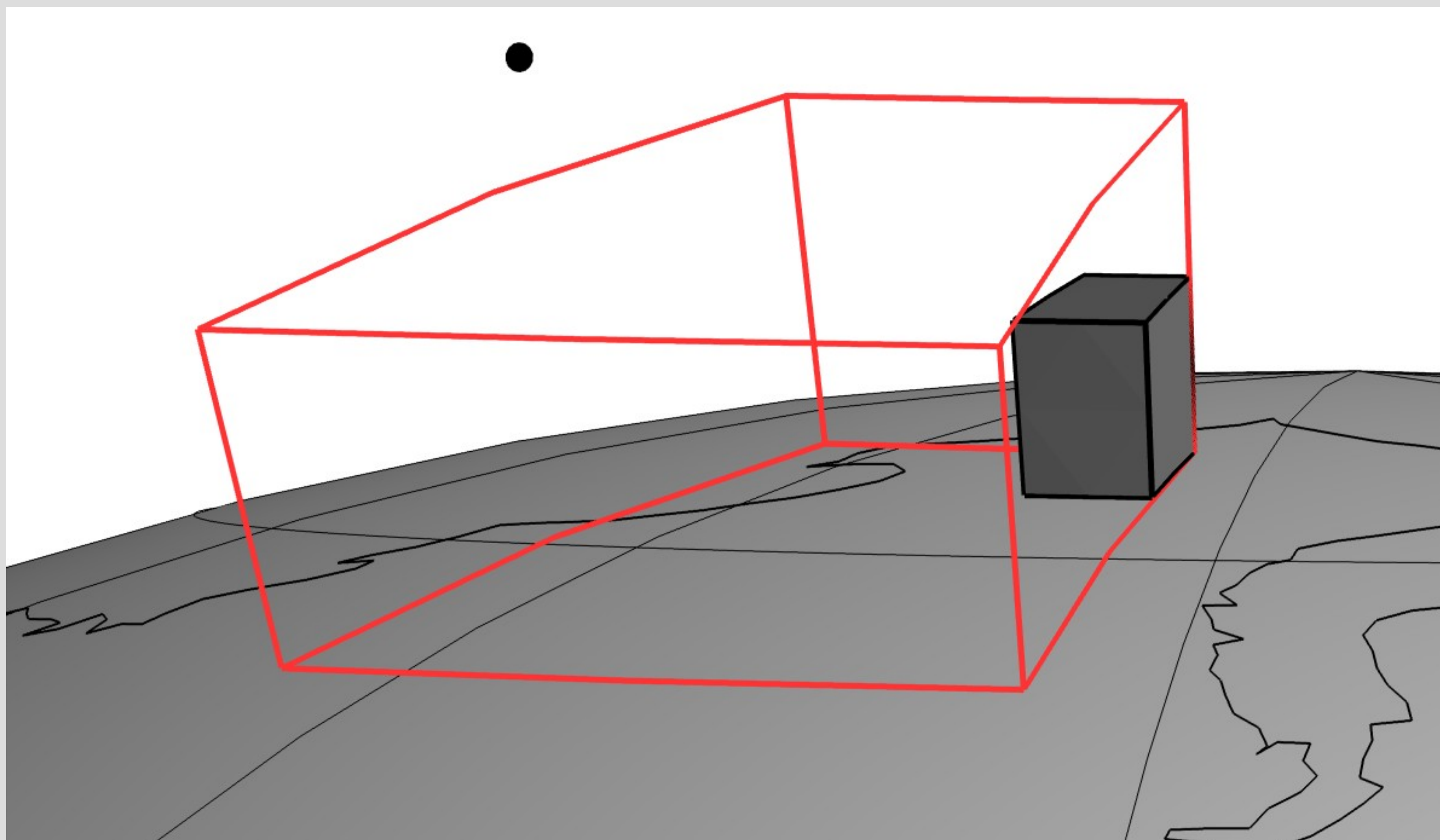
```
if d/L_x < D:
    Divide em x
```



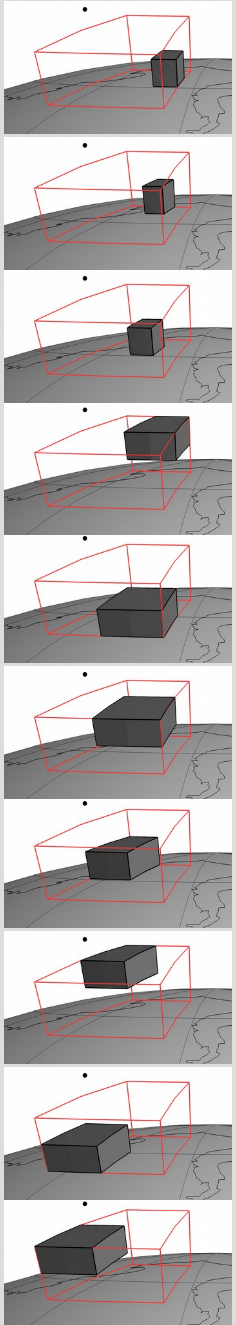
pilha



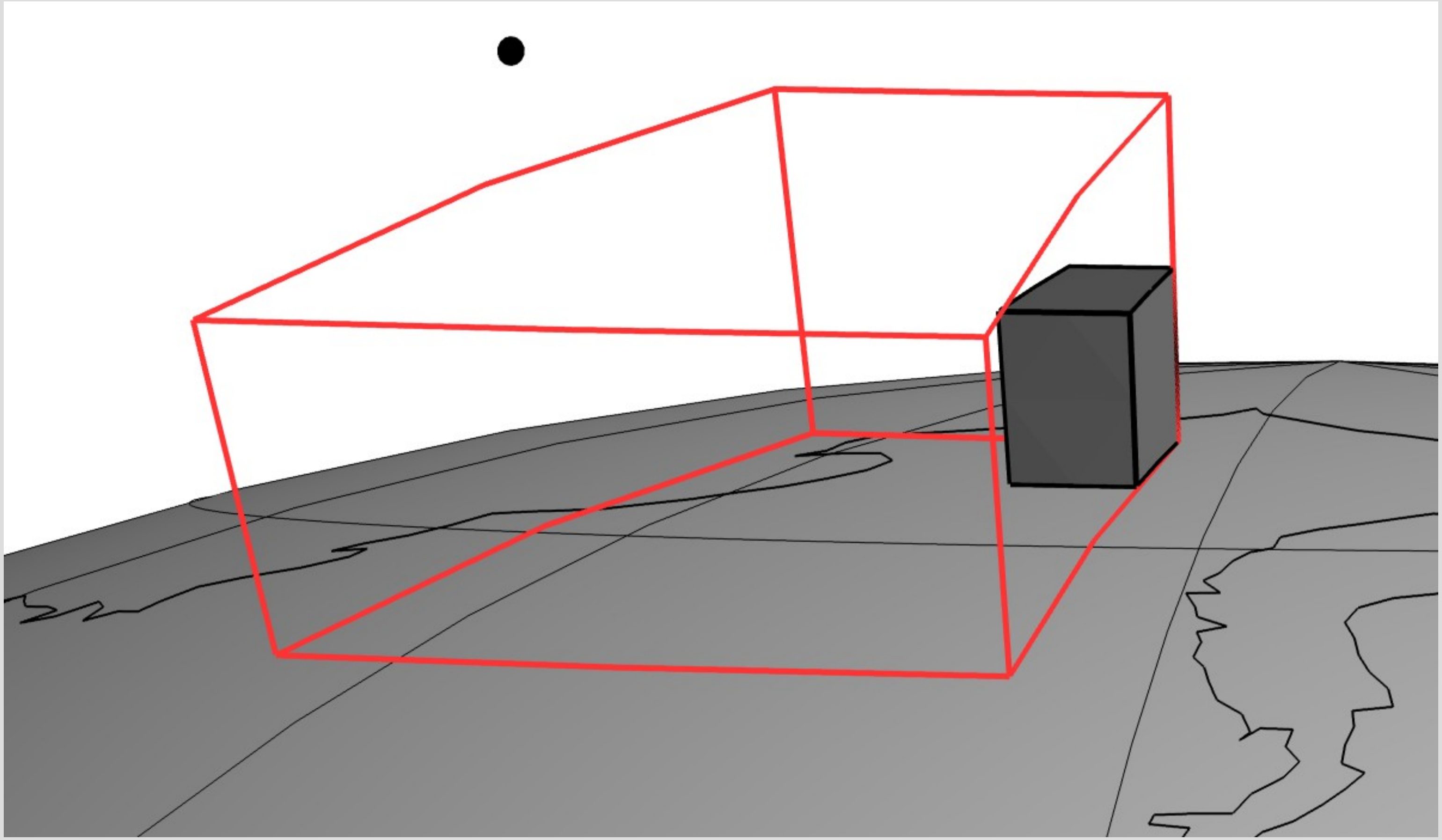
~~if d/L < D:~~
~~Divide em x~~



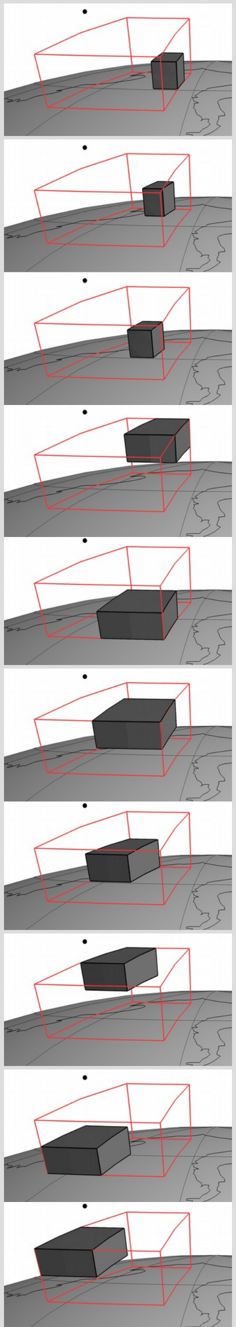
pilha



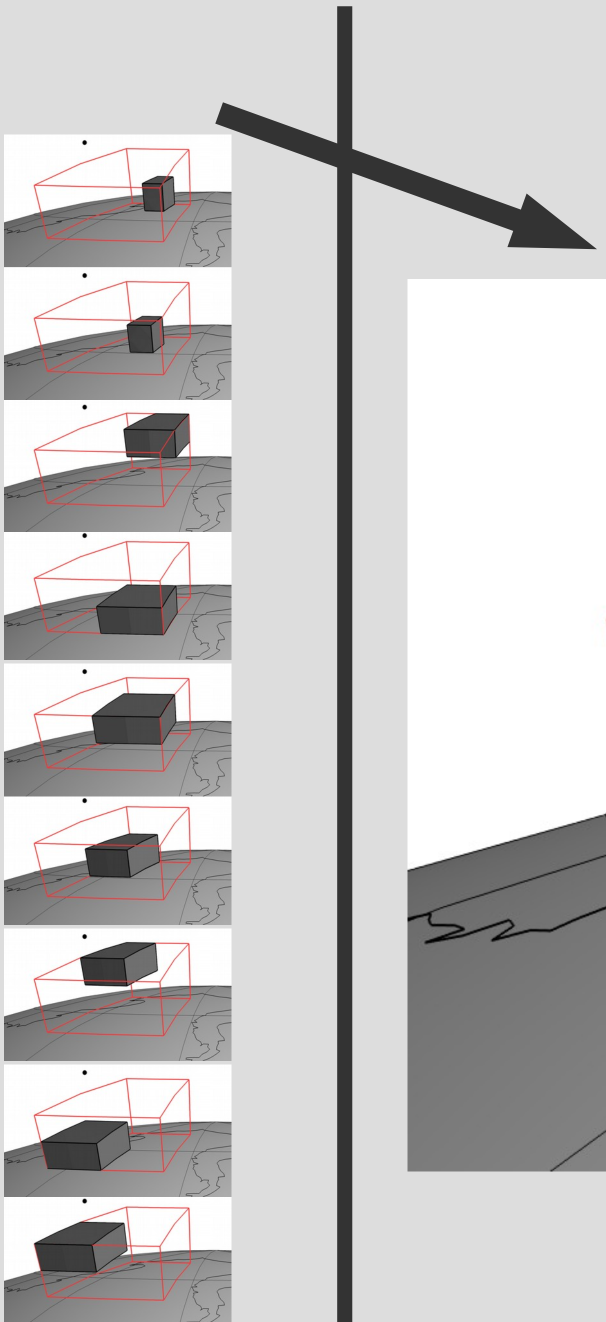
g total += g tesseroide



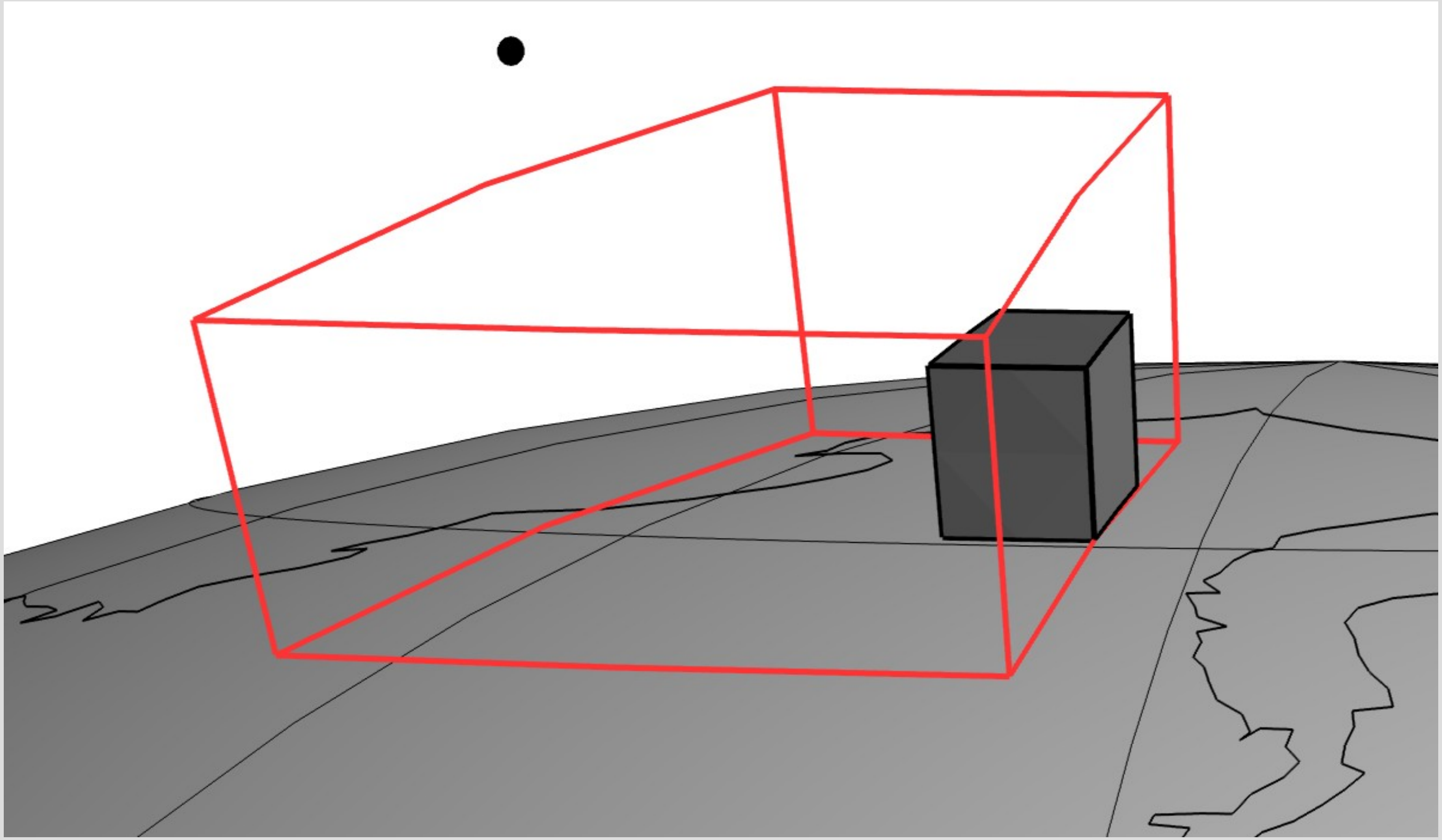
pilha



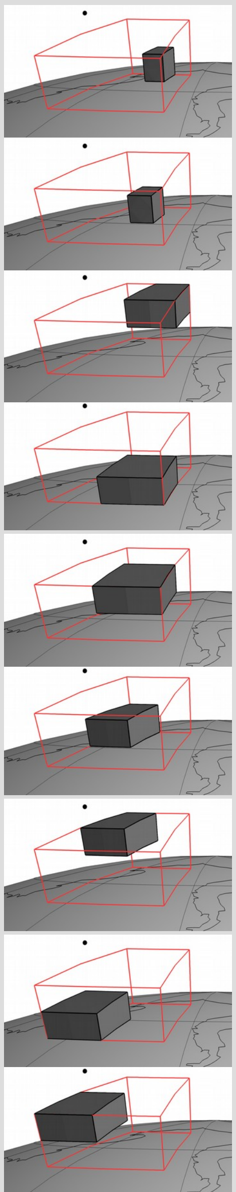
pilha



"pop"

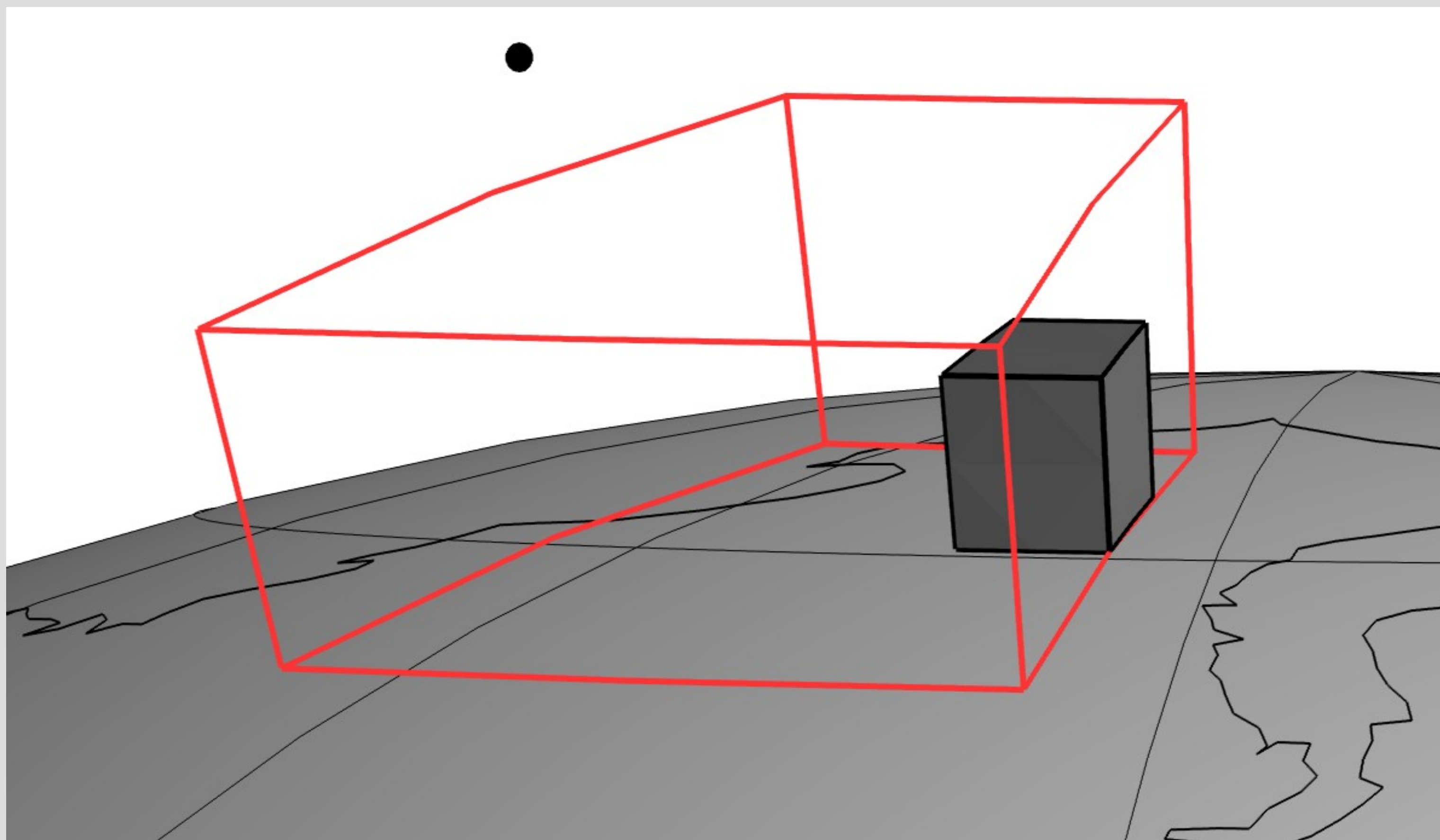


pilha

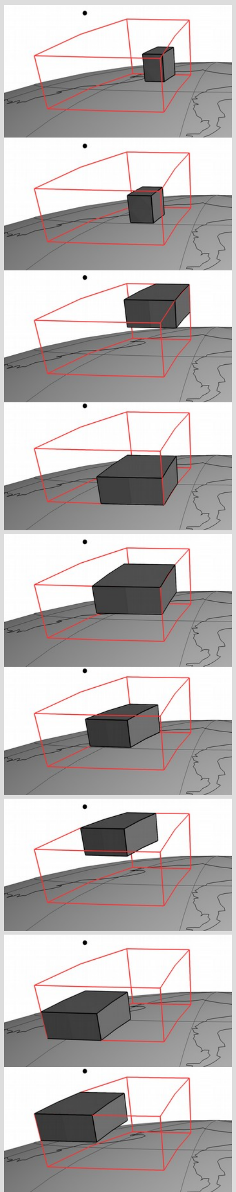


if $d/L_x < D$:

    Divide em x



pilha



pilha

vazia

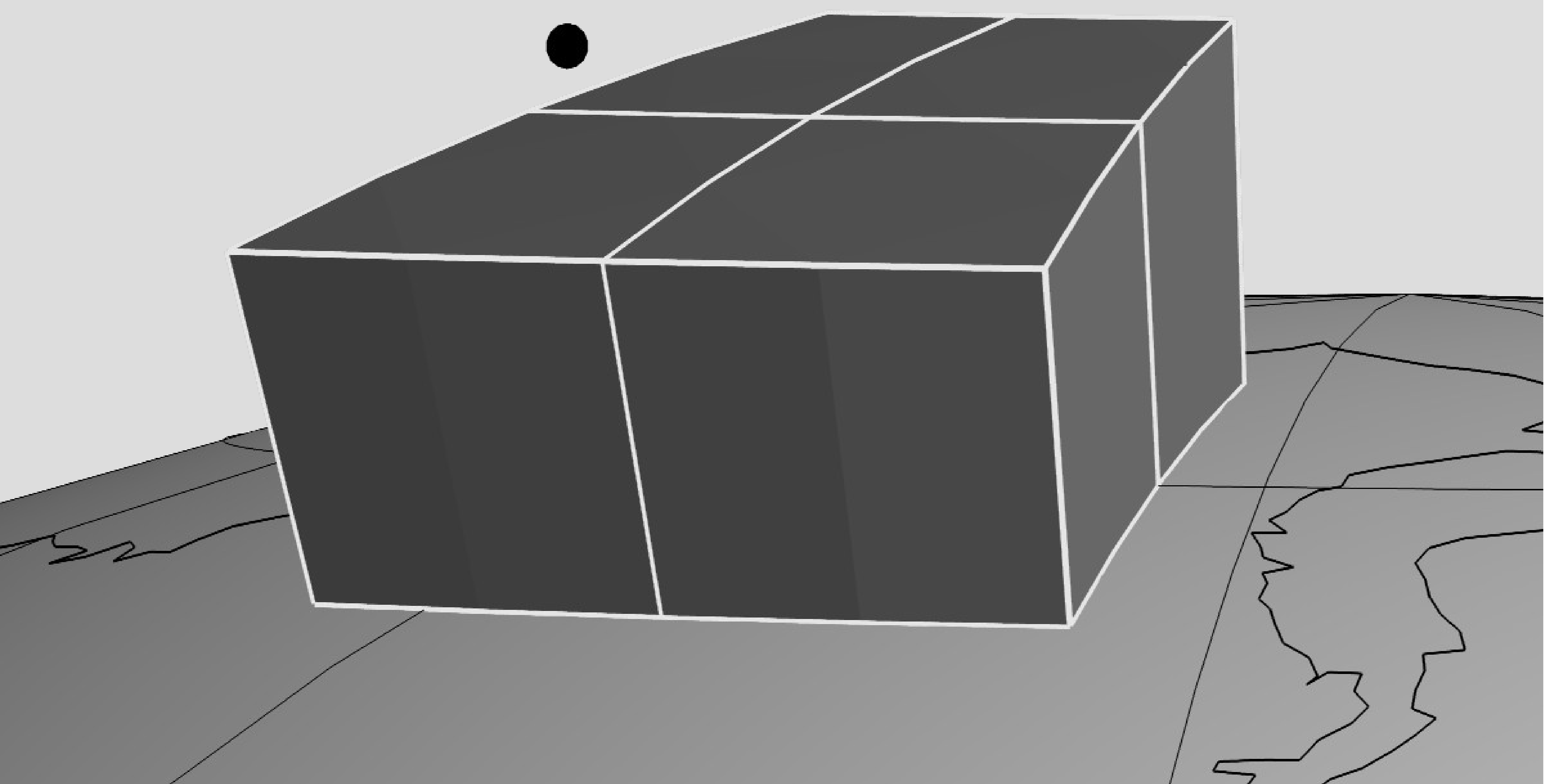
fim

g total

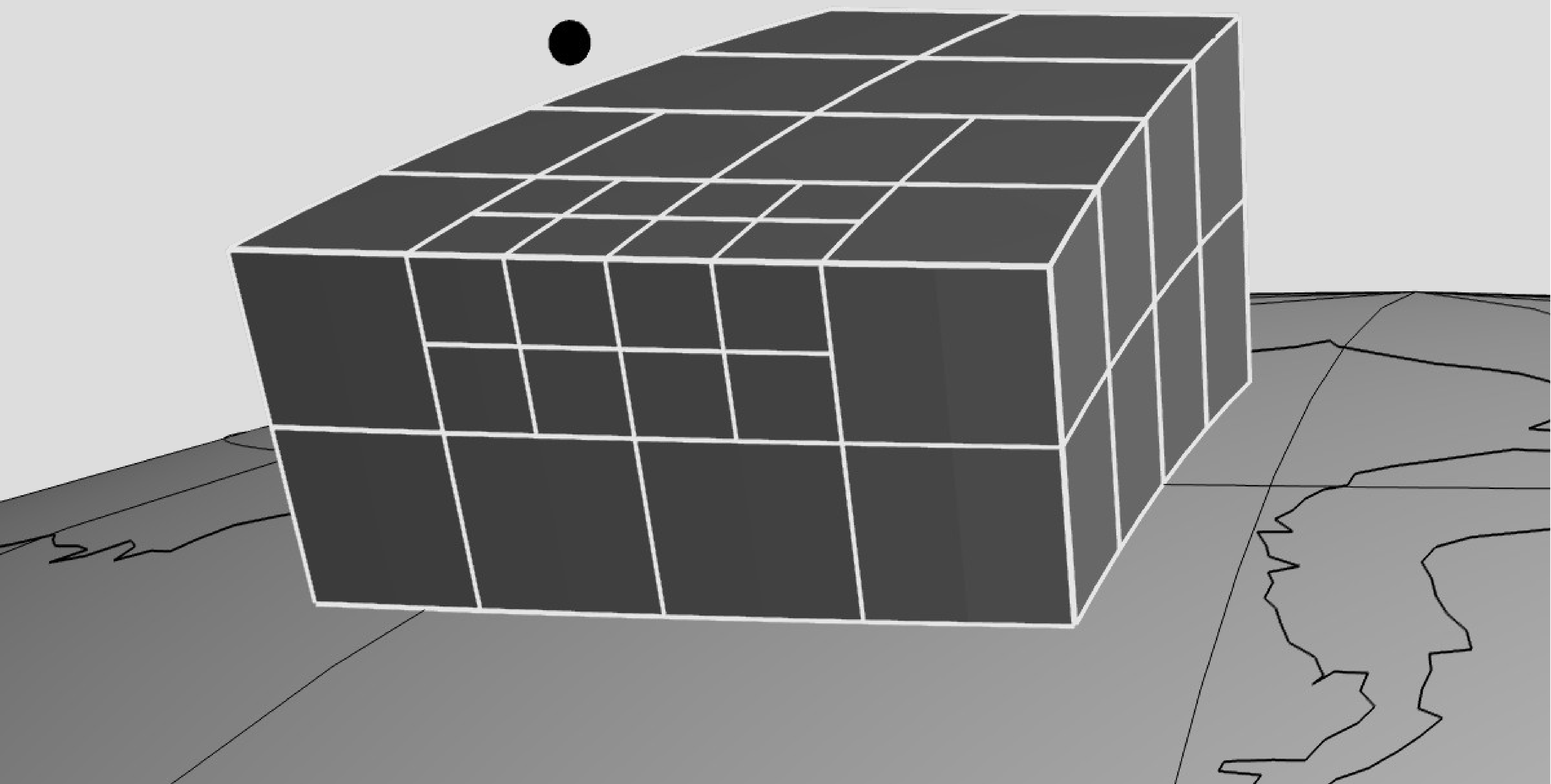
D

$d/L_x < D$

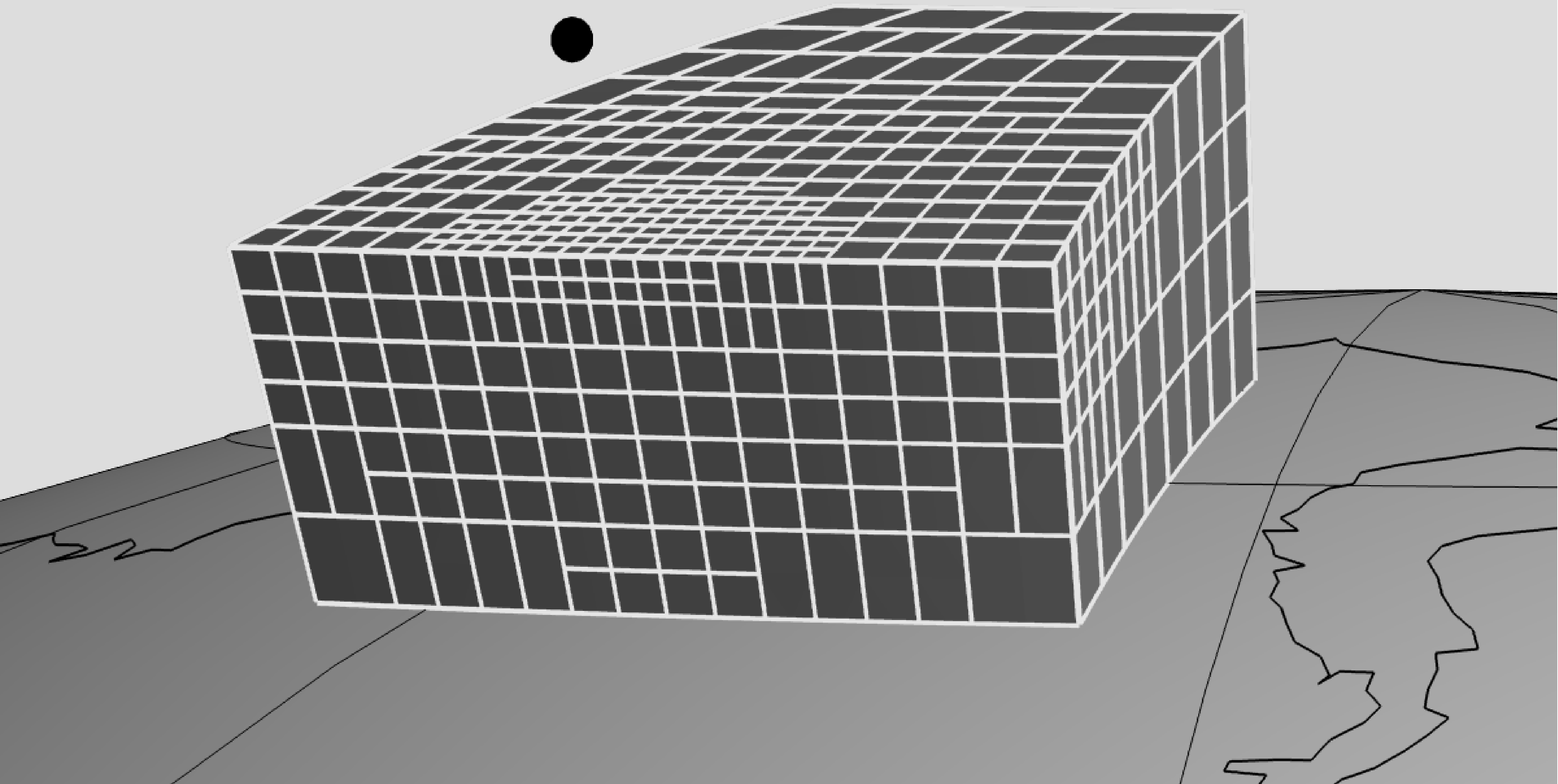
D=1   4 tesseroides

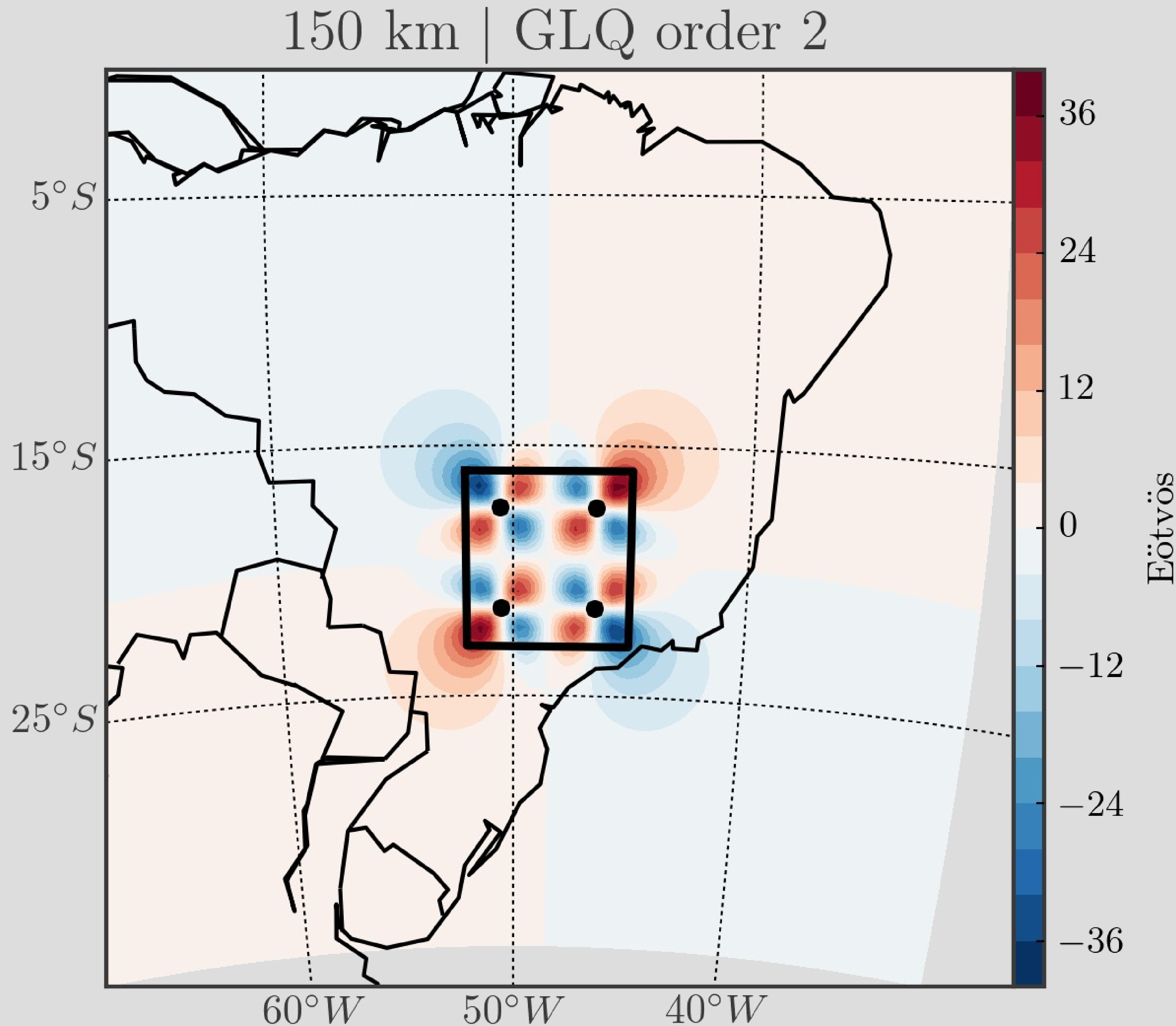


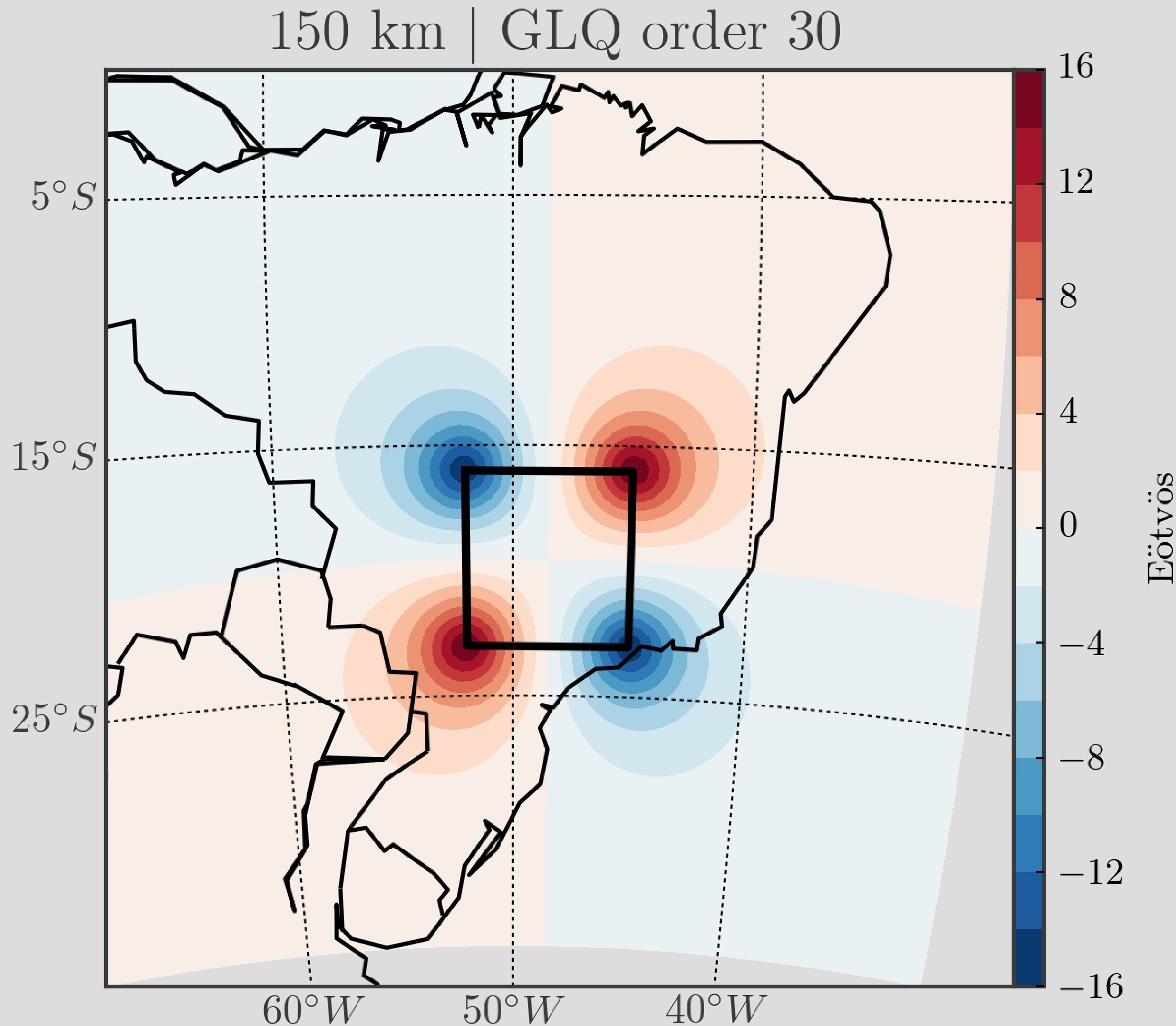
D=2   38 tesseroides



D=6   936 tesseroides

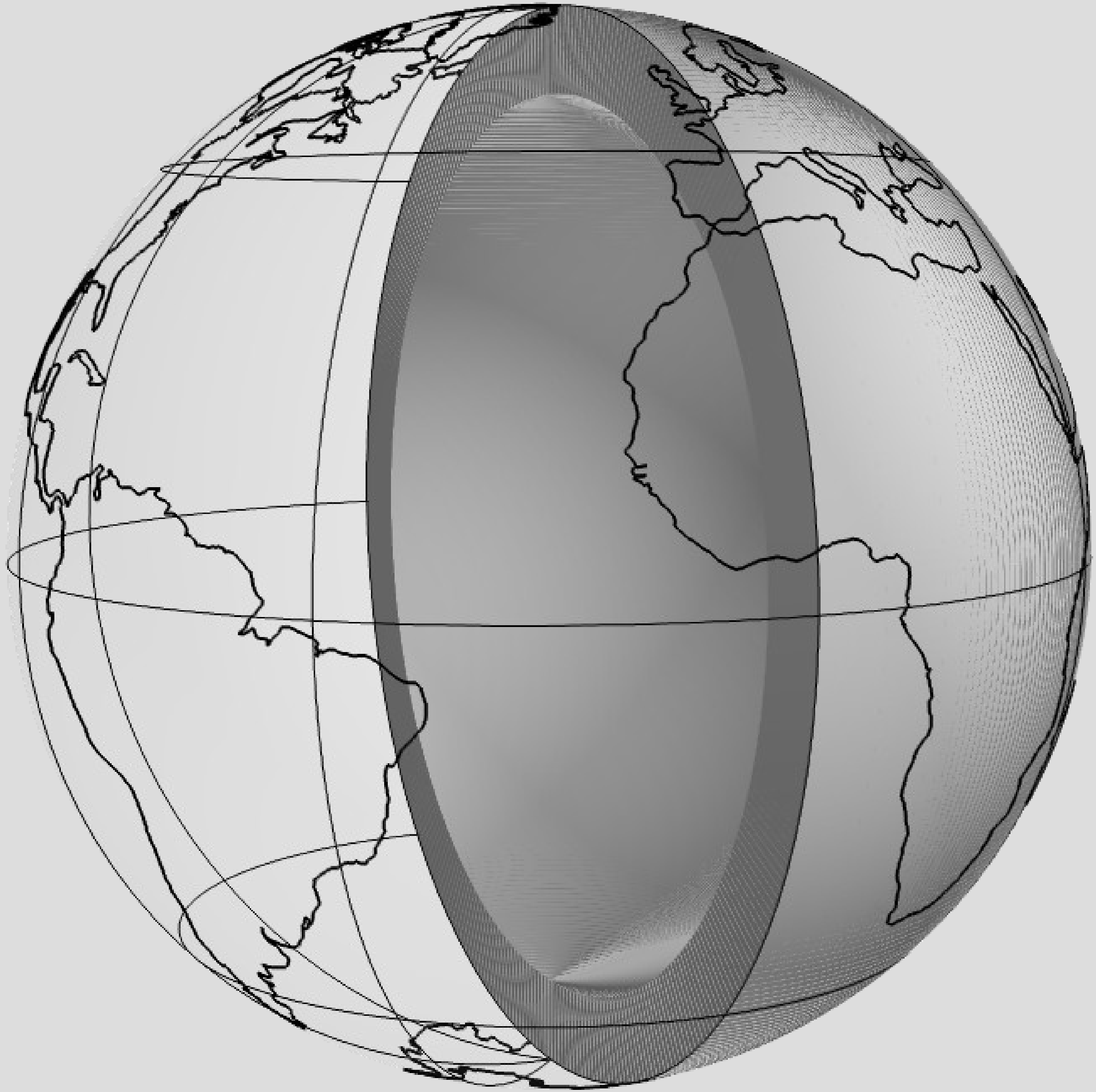



150 km | GLQ order 2
5°S
15°S
25°S
60°W
50°W
40°W
36
24
12
0
−12
−24
−36
Eötvös



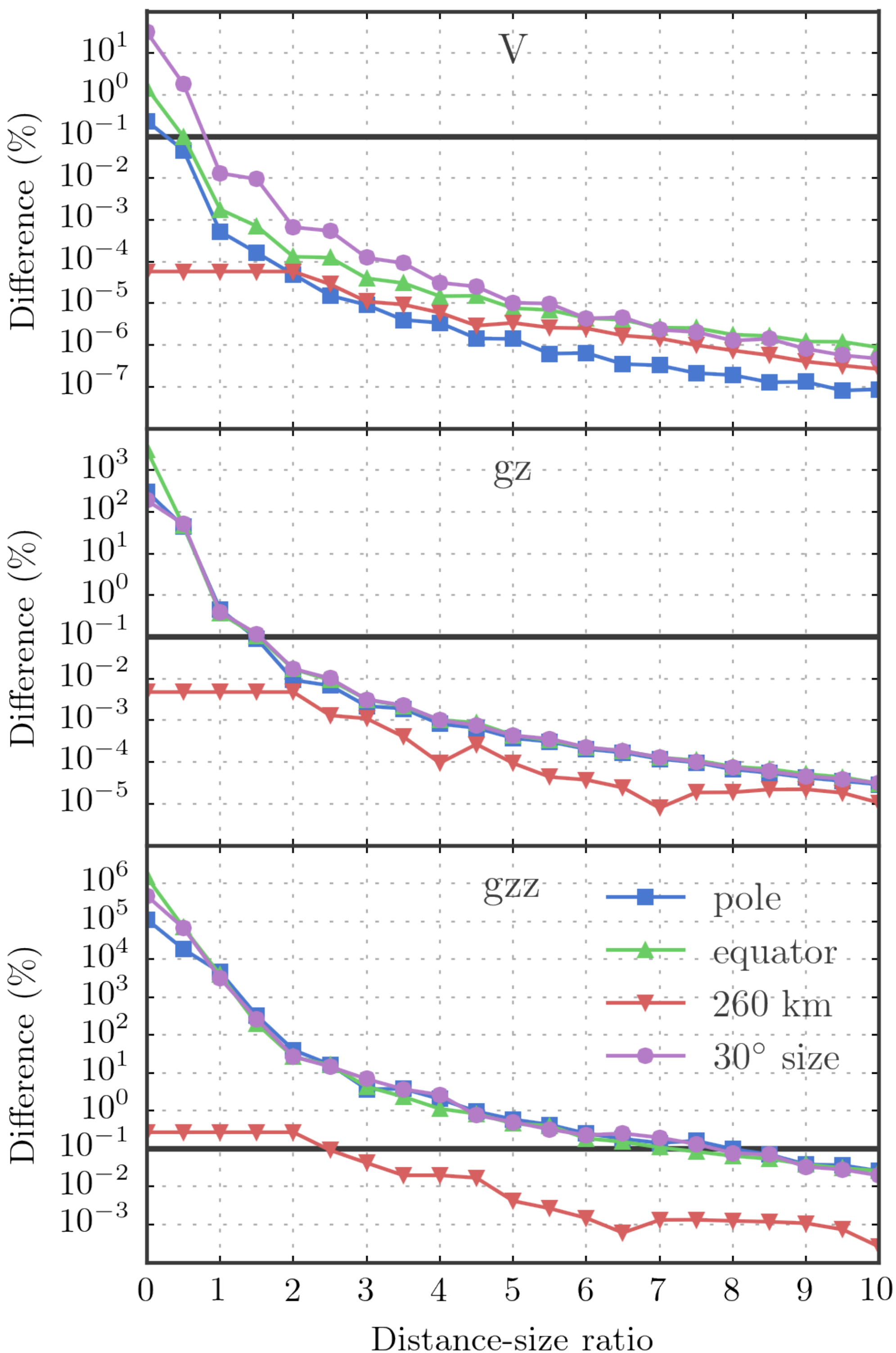
150 km | GLQ order 30
5°S
15°S
25°S
60°W
50°W
40°W
16
12
8
4
0
−4
−8
−12
−16
Eötvös

Melhor D?

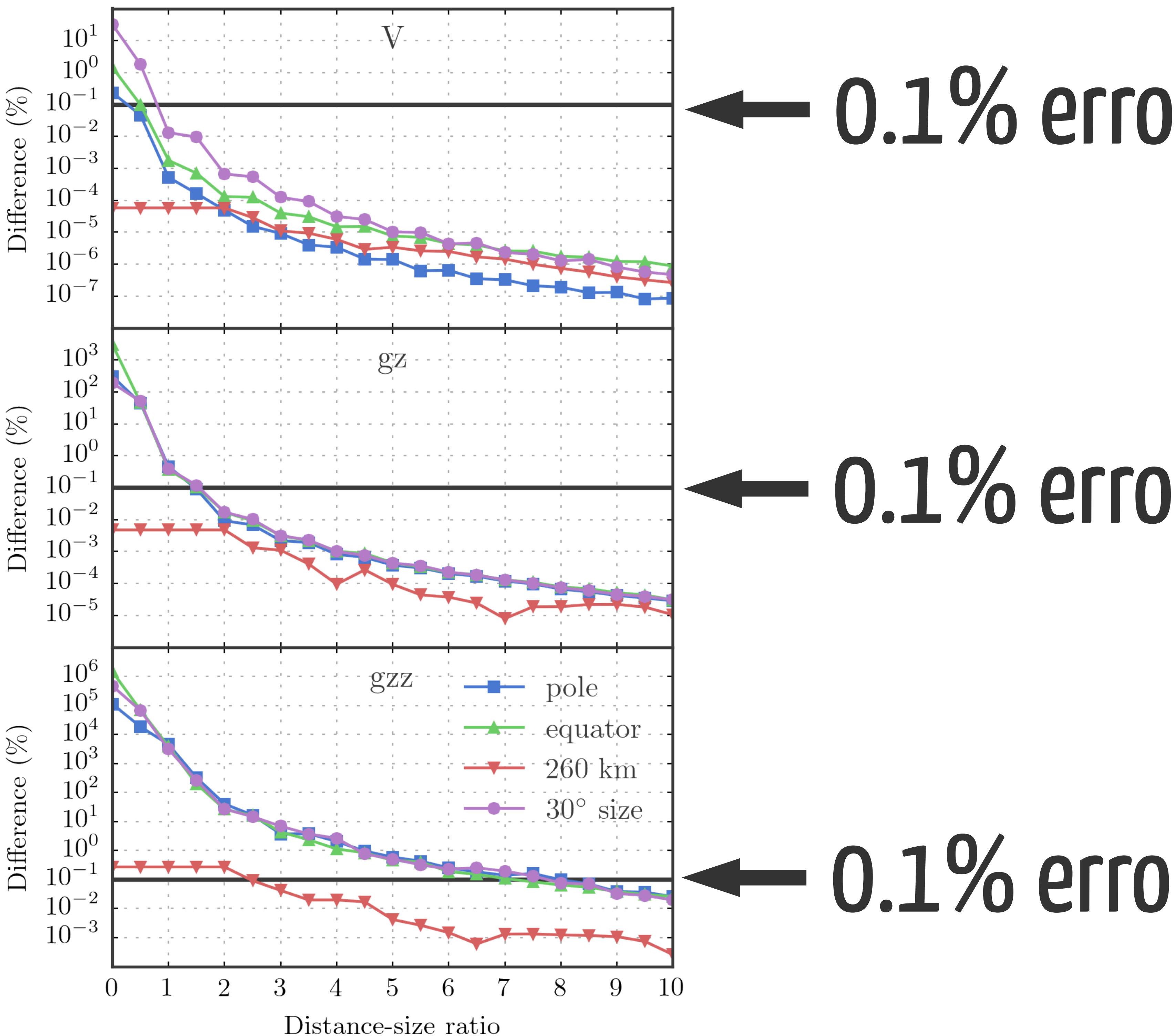


Casca x Quadratura

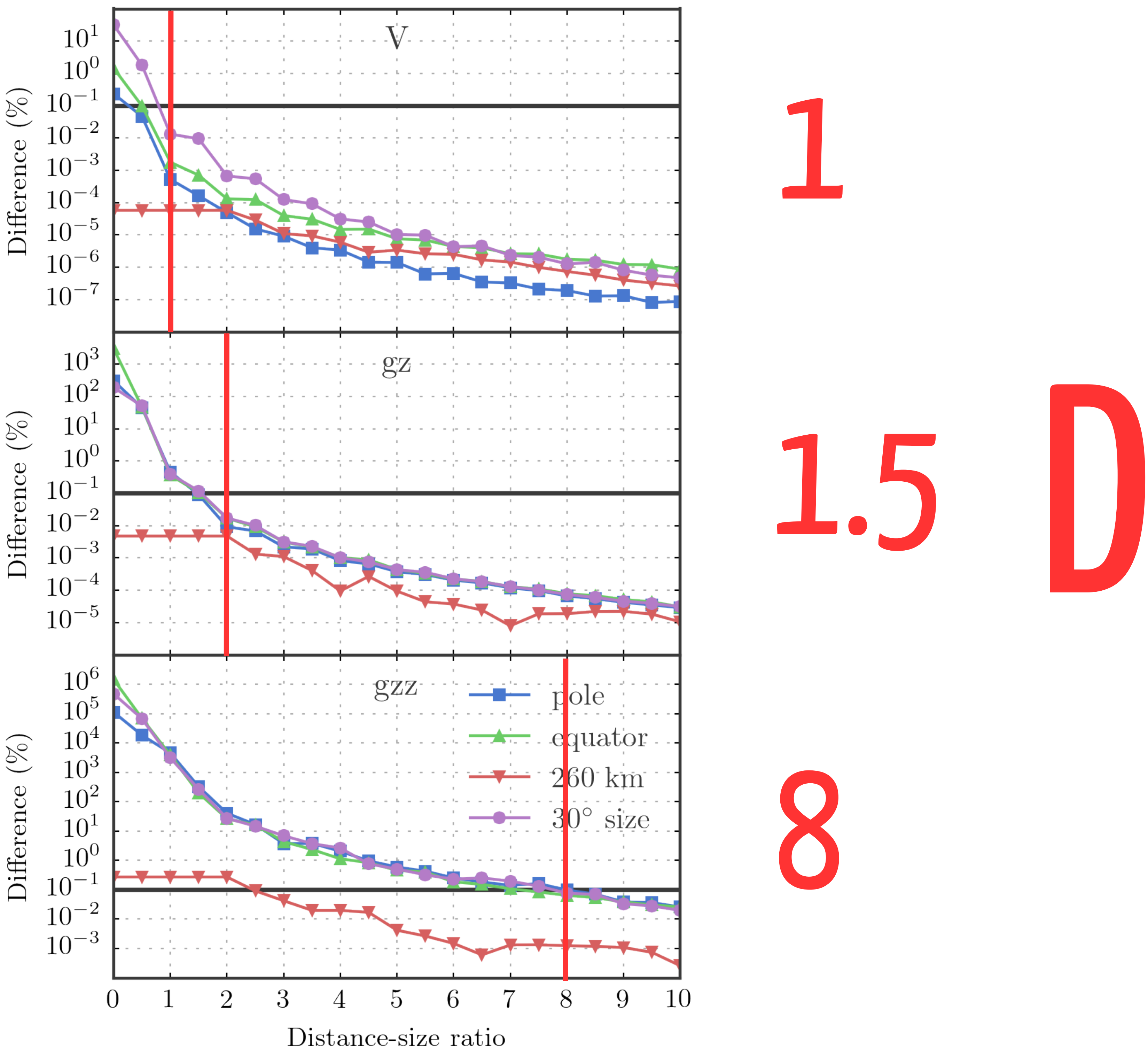
1. pólo	h=2km	1°x 1°x 1km
2. equador	h=2km	1°x 1°x 1km
3. pólo	h=260km	1°x 1°x 1km
4. pólo	h=2km	30°x 30°x 1km



V
gz
gzz
pole
equator
260 km
30° size
Difference (%)
Distance-size ratio

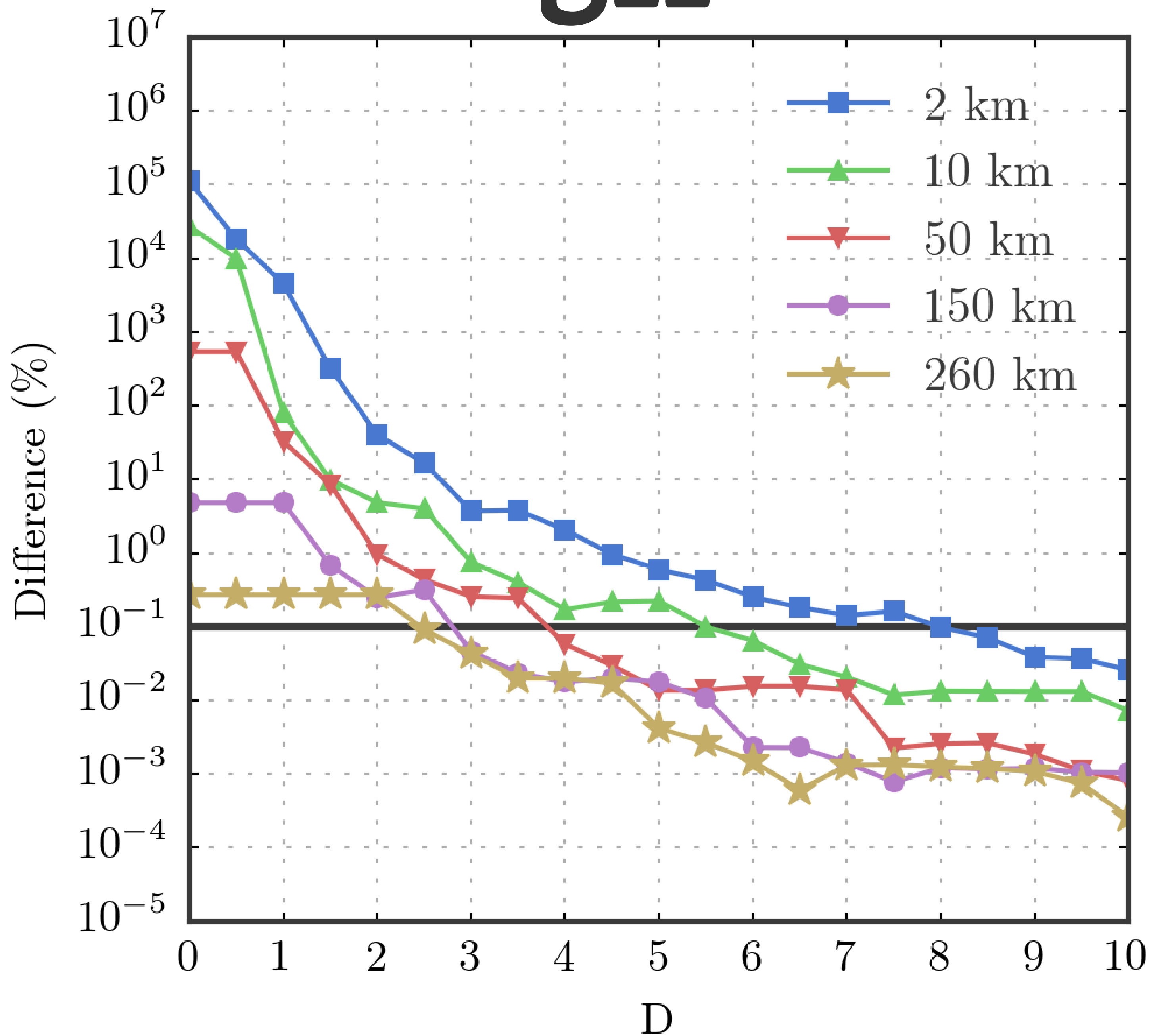


V
gz
gzz
pole
equator
260 km
30° size
Difference (%)
Distance-size ratio
0.1% erro
0.1% erro
0.1% erro



V
gz
gzz
pole
equator
260 km
30° size
Difference (%)
Distance-size ratio
1
1.5
8
D

gzz



software

Tesseroids

29 programas

linguagem C

open-source: BSD license

linha de comando

```
leo@gauss:~$ tess
tess2prism     tessgxx        tessgyy        tesslayers
tessdefaults   tessgxy        tessgyz        tessmass
tessgrd        tessgxz        tessgz         tessmodgen
tessgx         tessgy         tessgzz        tesspot
leo@gauss:~$ tessgz -h
Usage: tessgz MODELFILE [OPTIONS]

Calculate the gz component due to a tesseroid model on
specified observation points.

Values are calculated in the local coordinate system of the
observation point: x-> North  y-> East  z-> Up (away from the
center of the Earth).
In order to maintain mainstream convention, component gz is
calculated with z-> Down.

All units either SI or degrees!
```

Tesseroids

latest

Search docs

- Citing
- Known issues
- Changelog
- Installing
- Releases
- Theorerical background
- Using Tesseroids
- Cookbook
- License

WRITE THE DOCS

Love Documentation? Come to the Write the Docs 2016 conference in Portland.

Docs » Tesseroids: forward modeling in spherical coordinates

Edit on GitHub

Tesseroids: forward modeling in spherical coordinates

Tesseroids

A collection of **command-line programs** for modeling the **gravitational potential, acceleration, and gradient tensor**. *Tesseroids* supports models and computation grids in Cartesian and spherical coordinates.

Developed by Leonardo Uieda in cooperation with Carla Braitenberg.

Official site: http://tesseroids.leouieda.com

License: BSD 3-clause

Source code: https://github.com/leouieda/tesseroids

Latest release: v1.2.0 (doi:10.5281/zenodo.16033)

Note

Tesseroids is **research software**. Please consider citing it in your publications if you use it for your research.

Warning

tesseroids.leouieda.com

Tesseroids: forward modeling in spherical coordinates

Tesseroids

34 citações (google scholar)

…itational potential, acceleration, and
… grids in Cartesian and spherical

…enberg.

License: BSD 3-clause

Source code: https://github.com/leouieda/tesseroids

Latest release: v1.2.0 (doi:10.5281/zenodo.16033)

Note

Tesseroids is research software. Please consider citing it in your publications if you use it for your research.

Warning

tesseroids.leouieda.com

conclusão

Discretização adaptativa:

algoritmo melhor definido

pilha x recursivo

Discretização adaptativa:

algoritmo melhor definido

pilha x recursivo

Quantificação do erro:

D ótimo

Ds diferentes: V, gz, gzz

Discretização adaptativa:

algoritmo melhor definido pilha x recursivo

Quantificação do erro:

D ótimo Ds diferentes: V, gz, gzz

Implementação:

open-source usuários desenvolvedores

Tesseroids

modelagem direta
aprox. esférica

programa B

Novo método

Introdução

Tesseroids

Fatiando a Terra

Inversão Moho

Conclusão

fatiando a terra

modelagem direta e inversa para geofísica

SciPy 2013

"Modeling the Earth with Fatiando a Terra"

SciPy 2013

“Modeling the Earth with Fatiando a Terra”

v0.1

SciPy 2013

“Modeling the Earth with Fatiando a Terra”

v0.1

YouTube

fatiando a terra

An open-source Python library for modeling and inversion in geophysics.

Our goal is provide a comprehensive and extensible framework for geophysical data analysis and the development of new methodologies.

Research: Fatiando allows you to write Python scripts to perform your data analysis and generate figures in a reproducible way.

Development: Designed for extensibility, Fatiando offers tools for users to build upon the existing infrastructure and develop new inversion methods. We take care of the boilerplate.

Teaching: Fatiando can be combined with the Jupyter notebook to make rich, interactive documents. Great for teaching fundamental concepts of geophysics.

Overview

Gravity and magnetics

Modeling, inversion, and processing for potential field methods.

3D forward modeling with prisms, polygonal prisms, spheres, and tesseroids. Handles the potential, acceleration, gradient tensor, magnetic induction, total field magnetic anomaly.

Seismology and Seismics

Simple modeling functions for

Toy problems for: Cartesian st estimation. Experimental finite

fatiando.org

fatiando 0.4 Installing Documentation Cookbook Developer Guide

Search

fatiando a terra

An open-source Python library for modeling and inversion in geophysics

Our goal is provide … development of new

Research: Fatiando all… be combined with the scripts to perform your … rich, interactive generate figures in a rep… ching fundamental

~950 downloads/mês
7 citações

Overview

Gravity and magnetics

Modeling, inversion, and processing for potential field methods.

3D forward modeling with prisms, polygonal prisms, spheres, and tesseroids. Handles the potential, acceleration, gradient tensor, magnetic induction, total field magnetic anomaly.

Seismology and Seismics

Simple modeling functions for

Toy problems for: Cartesian st… estimation. Experimental finite

fatiando.org

NATURE | LETTER

日本語要約

Upper-plate controls on co-seismic slip in the 2011 magnitude 9.0 Tohoku-oki earthquake

Dan Bassett, David T. Sandwell, Yuri Fialko & Anthony B. Watts

Affiliations | Contributions | Corresponding author

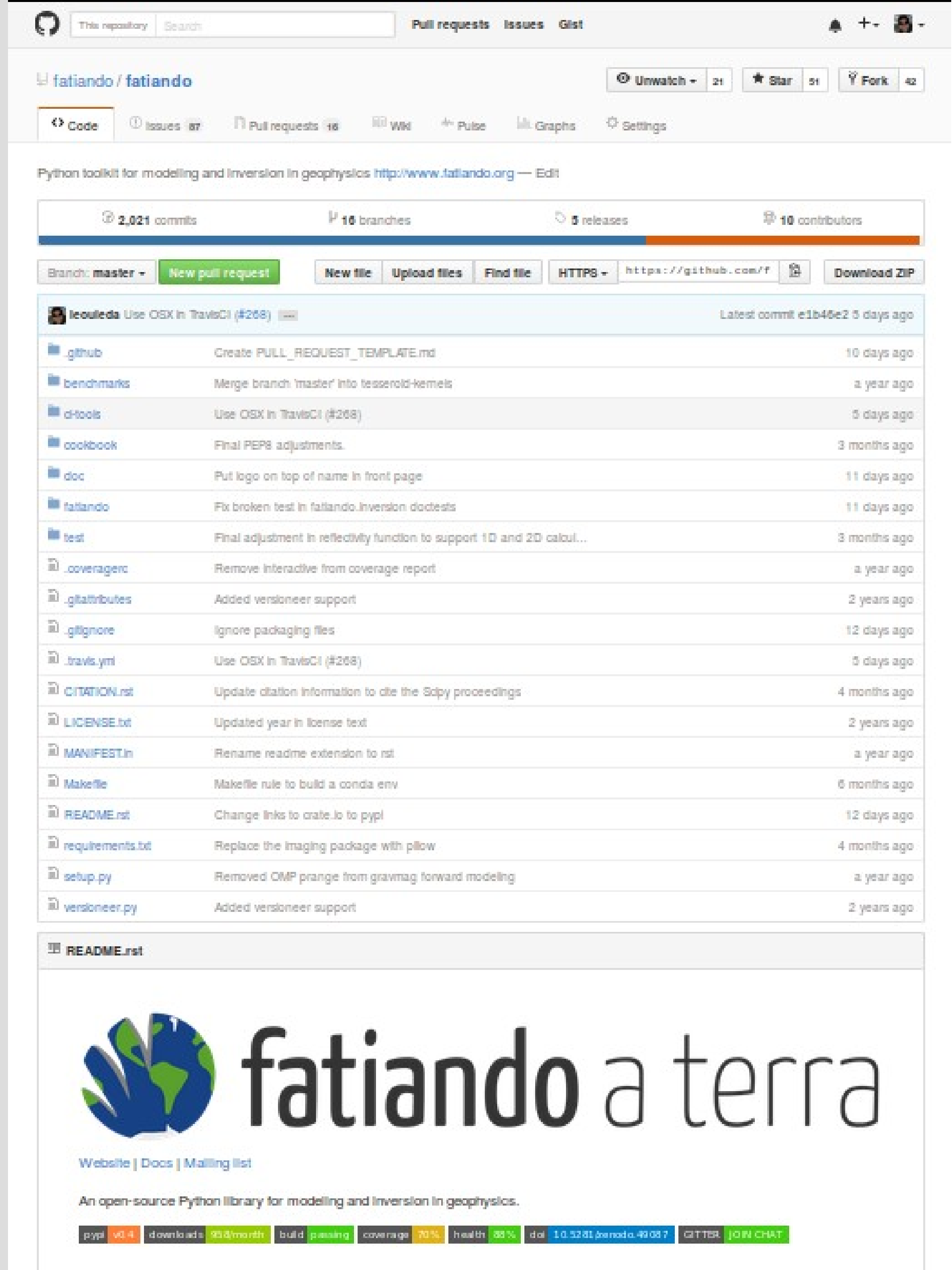
Nature **531**, 92–96 (03 March 2016) | doi:10.1038/nature16945
Received 26 June 2015 | Accepted 11 December 2015 | Published online 02 March 2016

The March 2011 Tohoku-oki earthquake was only the second giant (moment magnitude $M_w \geq 9.0$) earthquake to occur in the last 50 years and is the most recent to be recorded using modern geophysical techniques. Available data place high-resolution constraints on the kinematics of earthquake rupture[1], which have challenged prior knowledge about how much a fault can slip in a

Editor's summary

Dan Bassett *et al.* use residual gravity anomalies to characteri
structures in the region of the T
megathrust earthquake of Marc
point to a …

código
github.com



Implement tilt filter #261

Open **mtb-za** wants to merge 7 commits into `fatiando:master` from `mtb-za:tilt_filter`

Conversation 25 | Commits 7 | Files changed 3

mtb-za commented 25 days ago Fatiando a Terra member

This will create a filter to determine the tilt of a potential field, as developed by Miller and Singh (1994).

I could use some suggestions on how to create a sensible test for this.

This is the first of a few potential field filters that I want to implement.

Checklist:

- [ ] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings
- [ ] Code follows PEP8 style conventions
- [ ] Code and docs have been spellchecked
- [x] Include new dependencies in docs, requirements.txt, README, and .travis.yml
- [ ] Documentation builds properly
- [ ] All tests pass

Implement tilt filter #261

Open **mtb-za** wants to merge 7 commits into `fatiando:master` from `mtb-za:tilt_filter`

Conversation 25 | Commits 7 | Files changed 3

mtb-za commented 25 days ago — Fatiando a Terra member

This will create a filter to determine the tilt of a potential field, as developed by Miller and Singh (1994).

I could use some suggestions on how to create a sensible test for this.

This is the first of a few potential field filters that I want to implement.

Checklist:

- [ ] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings
- [ ] Code follows PEP8 style conventions
- [ ] Code and docs have been spellchecked
- [x] Include new dependencies in docs, requirements.txt, README, and .travis.yml
- [ ] Documentation builds properly
- [ ] All tests pass

Pad arrays #239

Open **drandykass** wants to merge 22 commits into `fatiando:master` from `drandykass:pad_arrays`

Conversation 67 | Commits 22 | Files changed 5

drandykass commented on Dec 2, 2015 — Fatiando a Terra member

This is an incomplete pull request for the addition of array padding and unpadding routines to gridder.

Some talking points:
--The padding function expects the array to be of the same dimension as intended--that is, no 'flattened' arrays.
--The unpadding function currently is set up to return unpadded coordinate vectors. However, I think that may be completely unnecessary.
--Still working on cleaning up the docs, the actual code, and need to create the test functions.

Update 03-12-2015 (**@leouieda**): Added the checklist below.

Checklist:

- [x] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings

Implement tilt filter #261

Open **mtb-za** wants to merge 7 commits into `fatiando:master` from `mtb-za:tilt_filter`

Conversation 25 | Commits 7 | Files changed 3

mtb-za commented 25 days ago — Fatiando a Terra member

This will create a filter to determine the tilt of a potential field, as developed by Miller and Singh (1994).

I could use some suggestions on how to create a sensible test for this.

This is the first of a few potential field filters that I want to implement.

Checklist:

- [ ] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings
- [ ] Code follows PEP8 style conventions
- [ ] Code and docs have been spellchecked
- [x] Include new dependencies in docs, requirements.txt, README, and .travis.yml
- [ ] Documentation builds properly
- [ ] All tests pass

Pad arrays #239

Open **drandykass** wants to merge 22 commits into `fatiando:master` from `drandykass:pad_arrays`

Conversation 67 | Commits 22 | Files changed 5

drandykass commented on Dec 2, 2015 — Fatiando a Terra member

This is an incomplete pull request for the addition of array padding and unpadding routines to gridder.

Some talking points:
--The padding function expects the array to be of the same dimension as intended--that is, no 'flattened' arrays.
--The unpadding function currently is set up to return unpadded coordinate vectors. However, I think that may be completely unnecessary.
--Still working on cleaning up the docs, the actual code, and need to create the test functions.

Update 03-12-2015 (**@leouieda**): Added the checklist below.

Checklist:

- [x] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings

Radial Average of Power Spectrum #230

Open **santis19** wants to merge 2 commits into `fatiando:master` from `santis19:fft-radial-average`

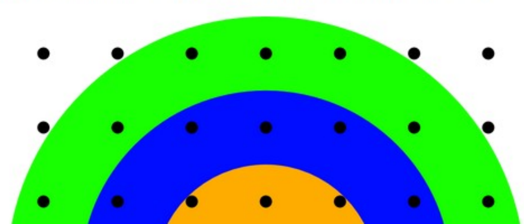
Conversation 7 | Commits 2 | Files changed 2

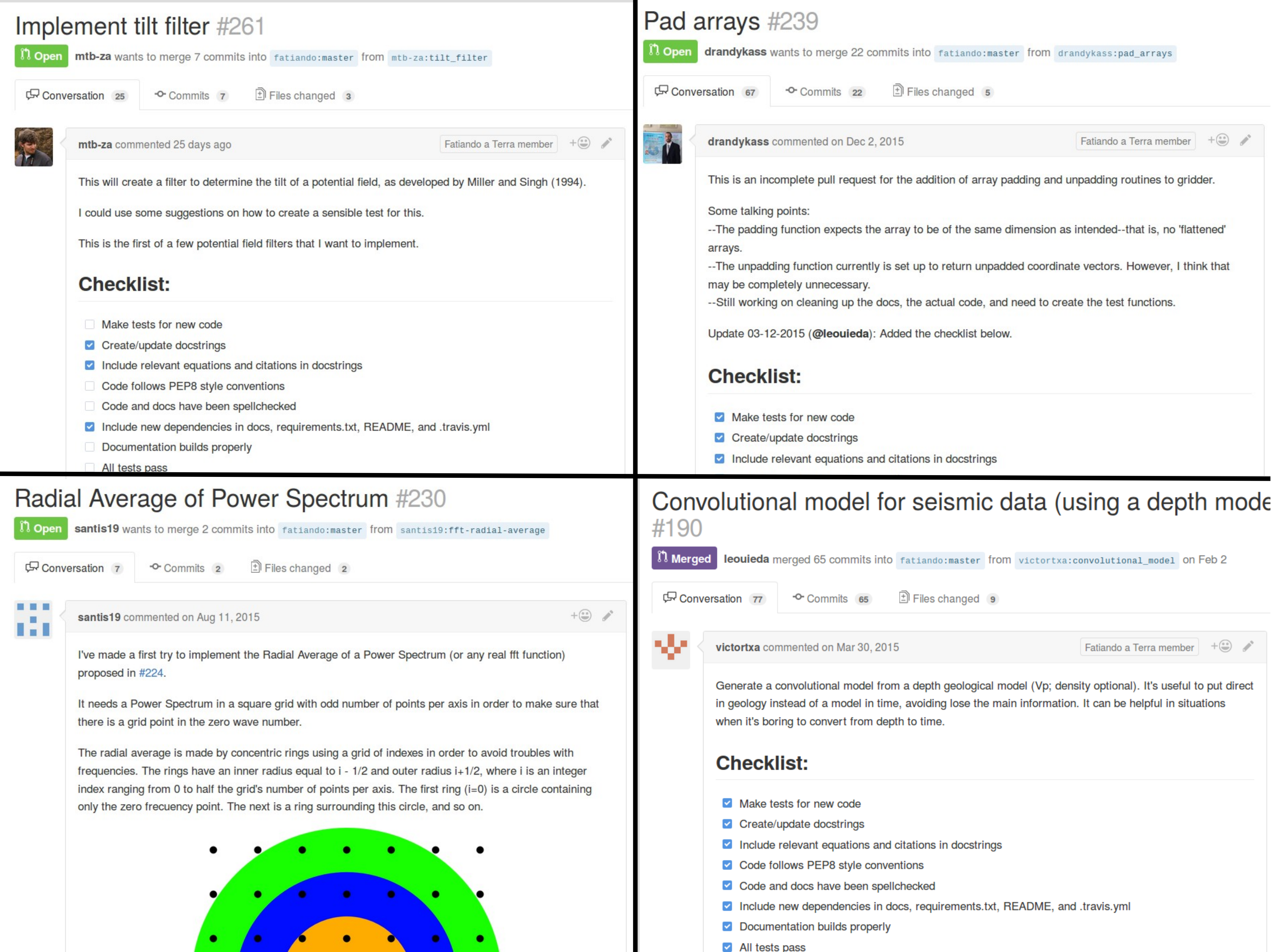
santis19 commented on Aug 11, 2015

I've made a first try to implement the Radial Average of a Power Spectrum (or any real fft function) proposed in #224.

It needs a Power Spectrum in a square grid with odd number of points per axis in order to make sure that there is a grid point in the zero wave number.

The radial average is made by concentric rings using a grid of indexes in order to avoid troubles with frequencies. The rings have an inner radius equal to i - 1/2 and outer radius i+1/2, where i is an integer index ranging from 0 to half the grid's number of points per axis. The first ring (i=0) is a circle containing only the zero frecuency point. The next is a ring surrounding this circle, and so on.





Implement tilt filter #261

Open **mtb-za** wants to merge 7 commits into `fatiando:master` from `mtb-za:tilt_filter`

Conversation 25 | Commits 7 | Files changed 3

mtb-za commented 25 days ago — Fatiando a Terra member

This will create a filter to determine the tilt of a potential field, as developed by Miller and Singh (1994).

I could use some suggestions on how to create a sensible test for this.

This is the first of a few potential field filters that I want to implement.

Checklist:

- [ ] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings
- [ ] Code follows PEP8 style conventions
- [ ] Code and docs have been spellchecked
- [x] Include new dependencies in docs, requirements.txt, README, and .travis.yml
- [ ] Documentation builds properly
- [ ] All tests pass

Pad arrays #239

Open **drandykass** wants to merge 22 commits into `fatiando:master` from `drandykass:pad_arrays`

Conversation 67 | Commits 22 | Files changed 5

drandykass commented on Dec 2, 2015 — Fatiando a Terra member

This is an incomplete pull request for the addition of array padding and unpadding routines to gridder.

Some talking points:
--The padding function expects the array to be of the same dimension as intended--that is, no 'flattened' arrays.
--The unpadding function currently is set up to return unpadded coordinate vectors. However, I think that may be completely unnecessary.
--Still working on cleaning up the docs, the actual code, and need to create the test functions.

Update 03-12-2015 (**@leouieda**): Added the checklist below.

Checklist:

- [x] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings

Radial Average of Power Spectrum #230

Open **santis19** wants to merge 2 commits into `fatiando:master` from `santis19:fft-radial-average`

Conversation 7 | Commits 2 | Files changed 2

santis19 commented on Aug 11, 2015

I've made a first try to implement the Radial Average of a Power Spectrum (or any real fft function) proposed in #224.

It needs a Power Spectrum in a square grid with odd number of points per axis in order to make sure that there is a grid point in the zero wave number.

The radial average is made by concentric rings using a grid of indexes in order to avoid troubles with frequencies. The rings have an inner radius equal to i - 1/2 and outer radius i+1/2, where i is an integer index ranging from 0 to half the grid's number of points per axis. The first ring (i=0) is a circle containing only the zero frecuency point. The next is a ring surrounding this circle, and so on.

Convolutional model for seismic data (using a depth mode #190

Merged **leouieda** merged 65 commits into `fatiando:master` from `victortxa:convolutional_model` on Feb 2

Conversation 77 | Commits 65 | Files changed 9

victortxa commented on Mar 30, 2015 — Fatiando a Terra member

Generate a convolutional model from a depth geological model (Vp; density optional). It's useful to put direct in geology instead of a model in time, avoiding lose the main information. It can be helpful in situations when it's boring to convert from depth to time.

Checklist:

- [x] Make tests for new code
- [x] Create/update docstrings
- [x] Include relevant equations and citations in docstrings
- [x] Code follows PEP8 style conventions
- [x] Code and docs have been spellchecked
- [x] Include new dependencies in docs, requirements.txt, README, and .travis.yml
- [x] Documentation builds properly
- [x] All tests pass

histórico

fatiando / **fatiando**

Unwatch 21 Star 51 Fork 42

Python toolkit for modeling and inversion in geophysics http://www.fatiando.org — Edit

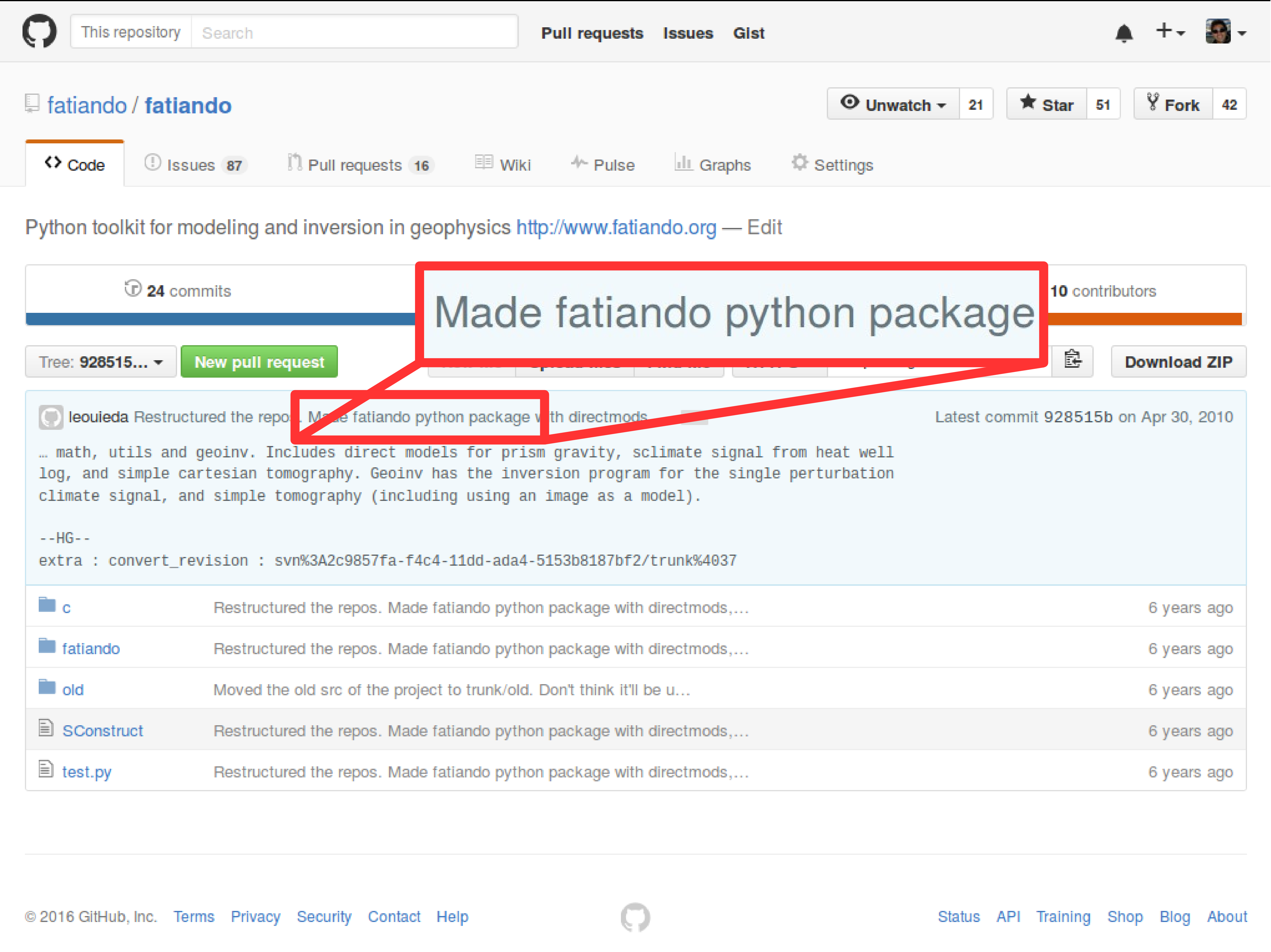
24 commits **16** branches **5** releases **10** contributors

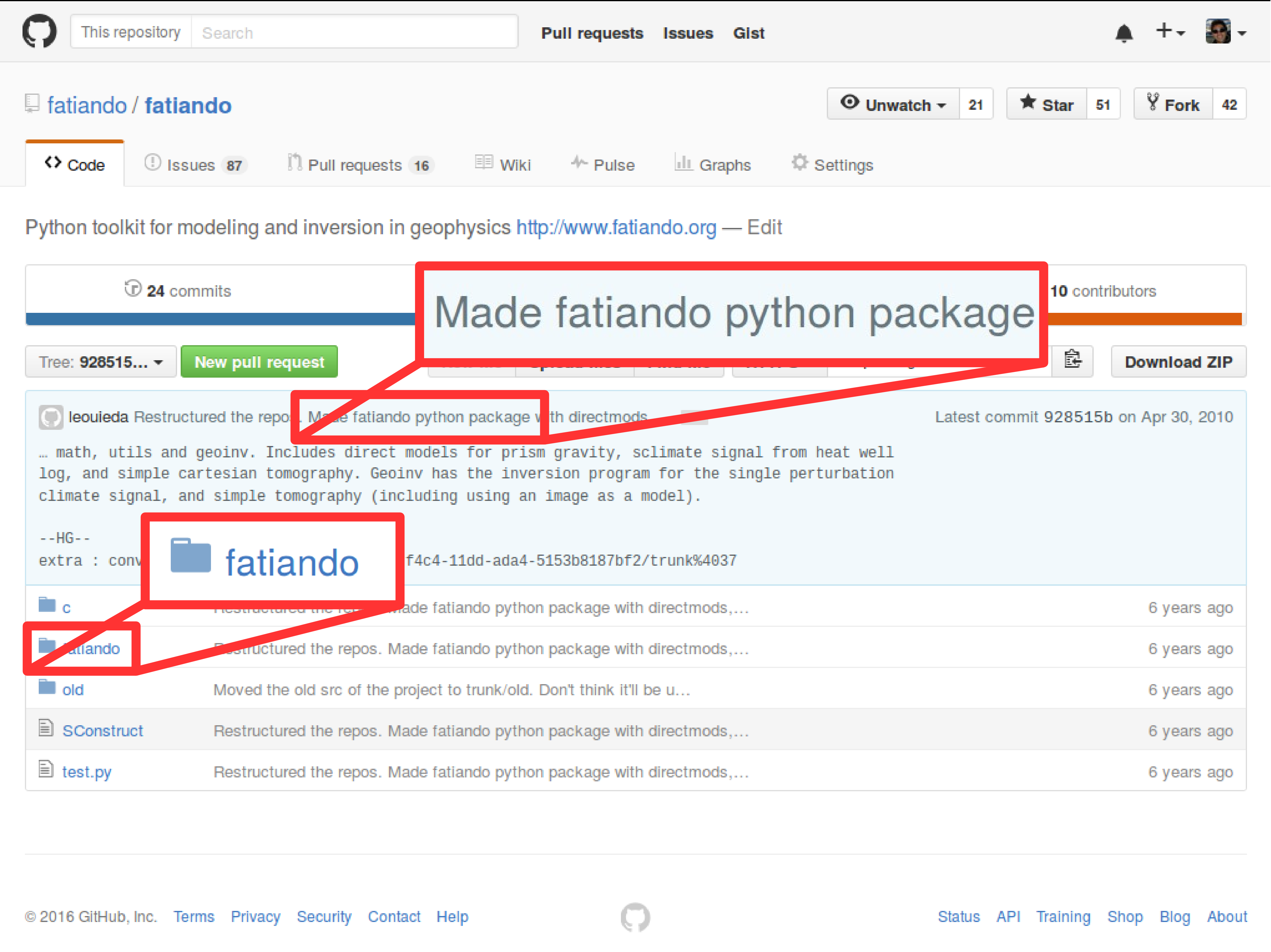
Tree: **928515...** New pull request New file Upload files Find file HTTPS https://github.com/fatia Download ZIP

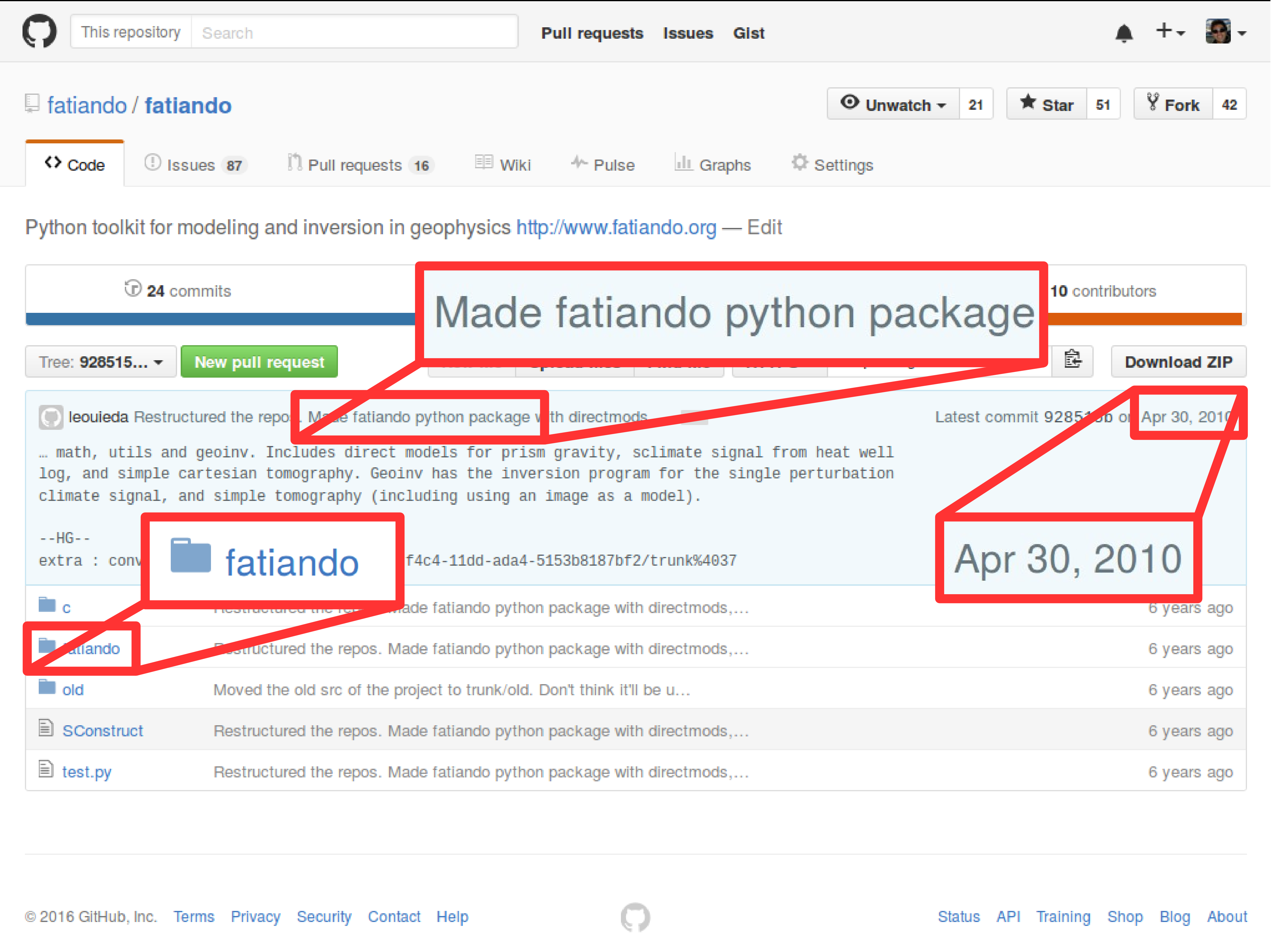
leouieda Restructured the repos. Made fatiando python package with directmods,… … Latest commit 928515b on Apr 30, 2010

```
… math, utils and geoinv. Includes direct models for prism gravity, sclimate signal from heat well
log, and simple cartesian tomography. Geoinv has the inversion program for the single perturbation
climate signal, and simple tomography (including using an image as a model).

--HG--
extra : convert_revision : svn%3A2c9857fa-f4c4-11dd-ada4-5153b8187bf2/trunk%4037
```

c	Restructured the repos. Made fatiando python package with directmods,…	6 years ago
fatiando	Restructured the repos. Made fatiando python package with directmods,…	6 years ago
old	Moved the old src of the project to trunk/old. Don't think it'll be u…	6 years ago
SConstruct	Restructured the repos. Made fatiando python package with directmods,…	6 years ago
test.py	Restructured the repos. Made fatiando python package with directmods,…	6 years ago


This repository
Search
Pull requests
Issues
Gist
fatiando / fatiando
Unwatch 21
Star 51
Fork 42
Code
Issues 87
Pull requests 16
Wiki
Pulse
Graphs
Settings
Python toolkit for modeling and inversion in geophysics http://www.fatiando.org — Edit
24 commits
10 contributors
Made fatiando python package
Tree: 928515...
New pull request
Download ZIP
leouieda Restructured the repos. Made fatiando python package with directmods
Latest commit 928515b on Apr 30, 2010
… math, utils and geoinv. Includes direct models for prism gravity, sclimate signal from heat well
log, and simple cartesian tomography. Geoinv has the inversion program for the single perturbation
climate signal, and simple tomography (including using an image as a model).
--HG--
extra : convert_revision : svn%3A2c9857fa-f4c4-11dd-ada4-5153b8187bf2/trunk%4037
c Restructured the repos. Made fatiando python package with directmods,… 6 years ago
fatiando Restructured the repos. Made fatiando python package with directmods,… 6 years ago
old Moved the old src of the project to trunk/old. Don't think it'll be u… 6 years ago
SConstruct Restructured the repos. Made fatiando python package with directmods,… 6 years ago
test.py Restructured the repos. Made fatiando python package with directmods,… 6 years ago
© 2016 GitHub, Inc. Terms Privacy Security Contact Help
Status API Training Shop Blog About


This repository
Search
Pull requests
Issues
Gist
fatiando / fatiando
Unwatch 21
Star 51
Fork 42
Code
Issues 87
Pull requests 16
Wiki
Pulse
Graphs
Settings
Python toolkit for modeling and inversion in geophysics http://www.fatiando.org — Edit
24 commits
10 contributors
Made fatiando python package
Tree: 928515...
New pull request
Download ZIP
leouieda Restructured the repos. Made fatiando python package with directmods
Latest commit 928515b on Apr 30, 2010
… math, utils and geoinv. Includes direct models for prism gravity, sclimate signal from heat well log, and simple cartesian tomography. Geoinv has the inversion program for the single perturbation climate signal, and simple tomography (including using an image as a model).
--HG--
extra : conv
f4c4-11dd-ada4-5153b8187bf2/trunk%4037
fatiando
c
made fatiando python package with directmods,…
6 years ago
fatiando
Restructured the repos. Made fatiando python package with directmods,…
6 years ago
old
Moved the old src of the project to trunk/old. Don't think it'll be u…
6 years ago
SConstruct
Restructured the repos. Made fatiando python package with directmods,…
6 years ago
test.py
Restructured the repos. Made fatiando python package with directmods,…
6 years ago
© 2016 GitHub, Inc.
Terms
Privacy
Security
Contact
Help
Status
API
Training
Shop
Blog
About


This repository
Search
Pull requests
Issues
Gist
fatiando / fatiando
Unwatch 21
Star 51
Fork 42
Code
Issues 87
Pull requests 16
Wiki
Pulse
Graphs
Settings
Python toolkit for modeling and inversion in geophysics http://www.fatiando.org — Edit
24 commits
10 contributors
Made fatiando python package
Tree: 928515...
New pull request
Download ZIP
leouieda Restructured the repo
th directmods
Latest commit
Apr 30, 2010
… math, utils and geoinv. Includes direct models for prism gravity, sclimate signal from heat well
log, and simple cartesian tomography. Geoinv has the inversion program for the single perturbation
climate signal, and simple tomography (including using an image as a model).
--HG--
extra : conv
f4c4-11dd-ada4-5153b8187bf2/trunk%4037
fatiando
Apr 30, 2010
c
ade fatiando python package with directmods,…
6 years ago
Restructured the repos. Made fatiando python package with directmods,…
6 years ago
old
Moved the old src of the project to trunk/old. Don't think it'll be u…
6 years ago
SConstruct
Restructured the repos. Made fatiando python package with directmods,…
6 years ago
test.py
Restructured the repos. Made fatiando python package with directmods,…
6 years ago
© 2016 GitHub, Inc.
Terms
Privacy
Security
Contact
Help
Status
API
Training
Shop
Blog
About

juntar código

disciplinas, teses

dissertações, etc

biblioteca

funções, classes, etc

python™

fácil de aprender

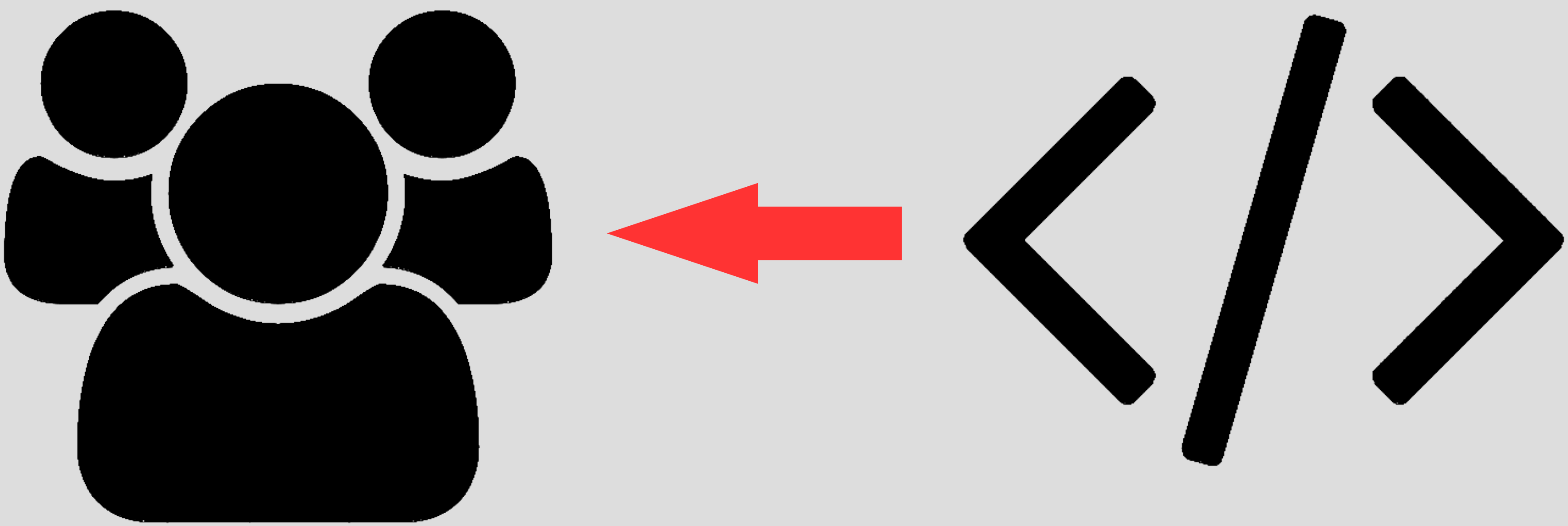
rápido de implementar

open-source

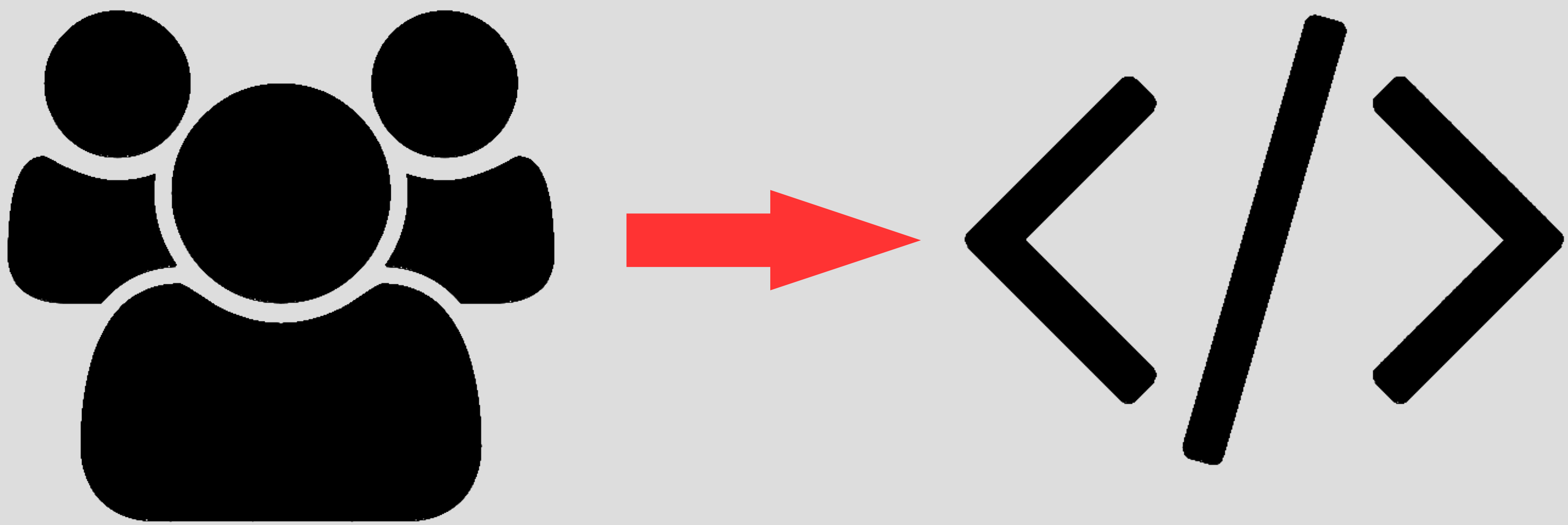
“bateries included”

1 código

++ usuários



1 código
++ programadores



a biblioteca

fatiando.gridder
mesher
utils
vis
gravmag
seismic
inversion

Módulos

fatiando.gridder

mesher

utils

vis

gravmag

seismic

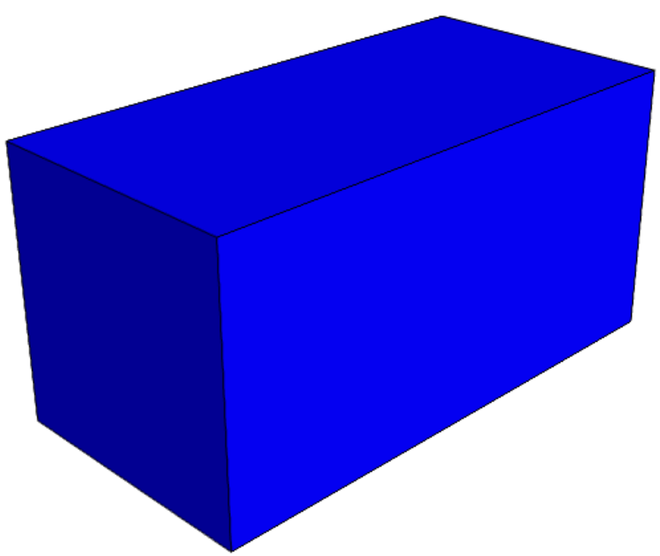
inversion

Módulos

In:

```
from fatiando.mesher import Prism
modelo = [
    Prism(0, 1000, 0, 2000, 1500, 2500,
          props={'magnetization': [4,-1,3]})]
from fatiando.vis import myv
myv.figure()
myv.prisms(modelo)
myv.show()
```

Out:



In:

```python
from fatiando import gridder

area = (-5000, 5000, -5000, 5000)
shape = (50, 50)
x, y, z = gridder.regular(area, shape, z=0)

print(x)
print(x.size)
```

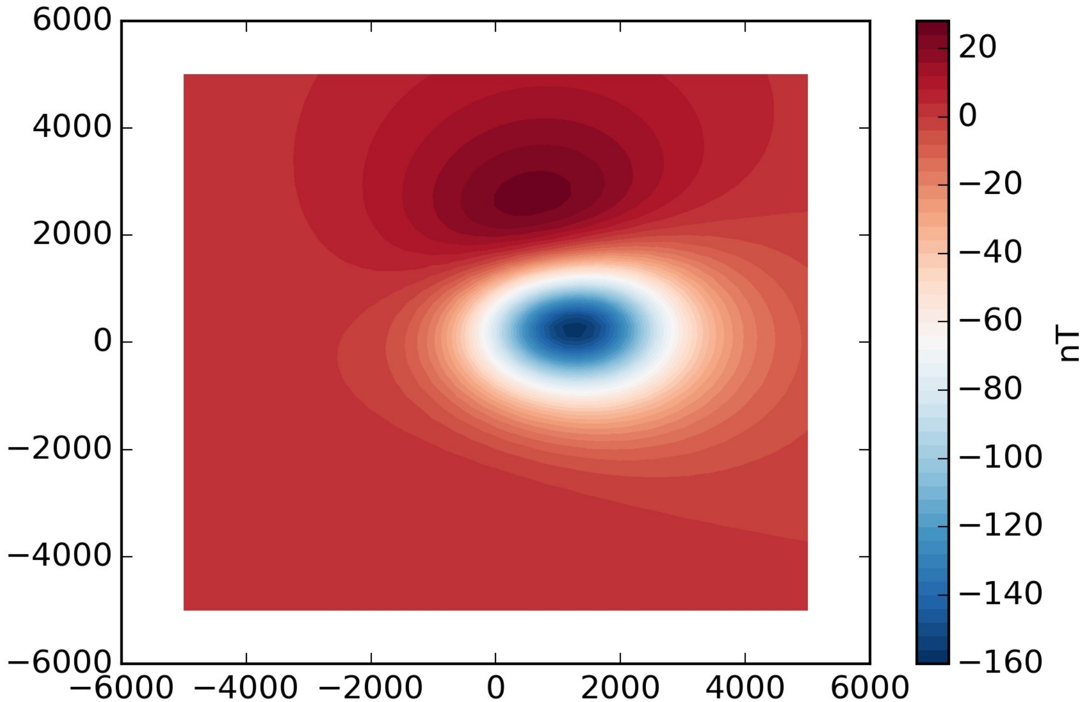
Out:

```
[-5000. -5000. -5000. ... 5000. 5000. 5000.]
2500
```

In:

```python
from fatiando.gravmag import prism
mag = prism.tf(x, y, z, modelo,
               inc=-60, dec=20)

import matplotlib.pyplot as plt
plt.tricontourf(y, x, mag, 50, cmap="RdBu_r")
plt.colorbar().set_label('nT')
```

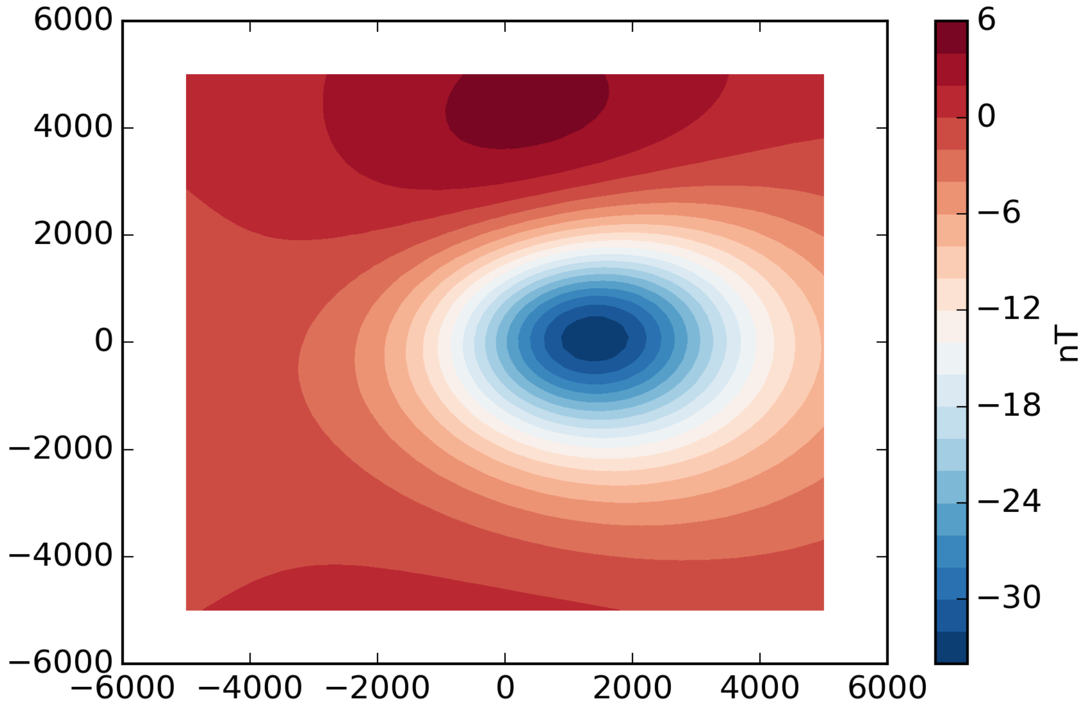
Out:



In:

```
from fatiando.gravmag import transform
up = transform.upcontinue(x, y, mag, shape,
                          1500)

plt.tricontourf(y, x, up, 20, cmap="RdBu_r")
plt.colorbar().set_label('nT')
```

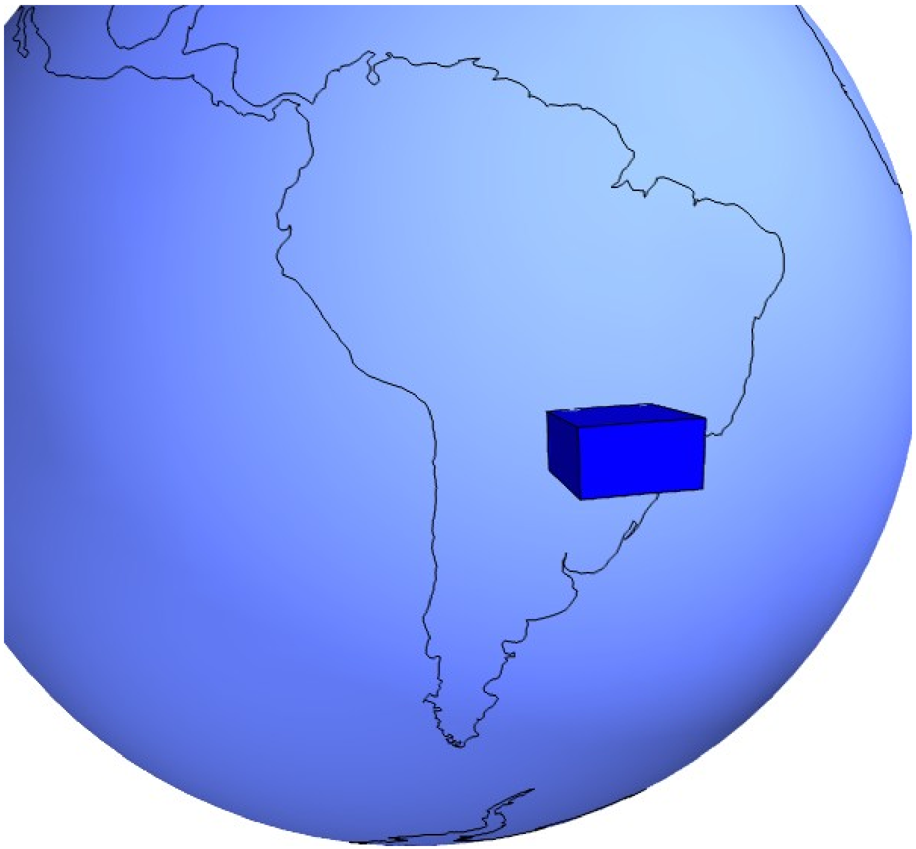
Out:



In:

```python
from fatiando.mesher import Tesseroid
mod = [Tesseroid(-60, -50, -25, -20, 2000, 0,
                 props={'density': 2670})]
myv.figure()
myv.tesseroids(mod, scale=(1, 1, 500))
myv.earth()
myv.continents()
```

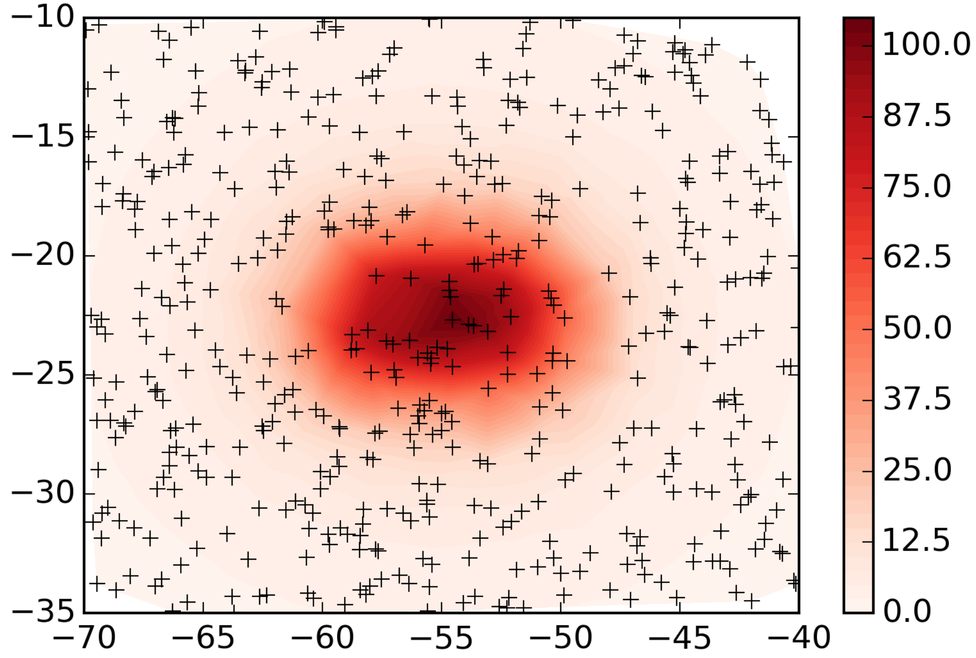
Out:



In:

```
lon, lat, h = gridder.scatter(
    (-70, -40, -35, -10), 500, z=250e3)

from fatiando.gravmag import tesseroid
g = tesseroid.gz(lon, lat, h, mod)

plt.tricontourf(lon, lat, g, 50, cmap="Reds")
plt.colorbar()
```

Out:



Módulos

fatiando.gridder
mesher
utils
vis
gravmag
seismic
inversion

fatiando.gridder

mesher

utils

vis

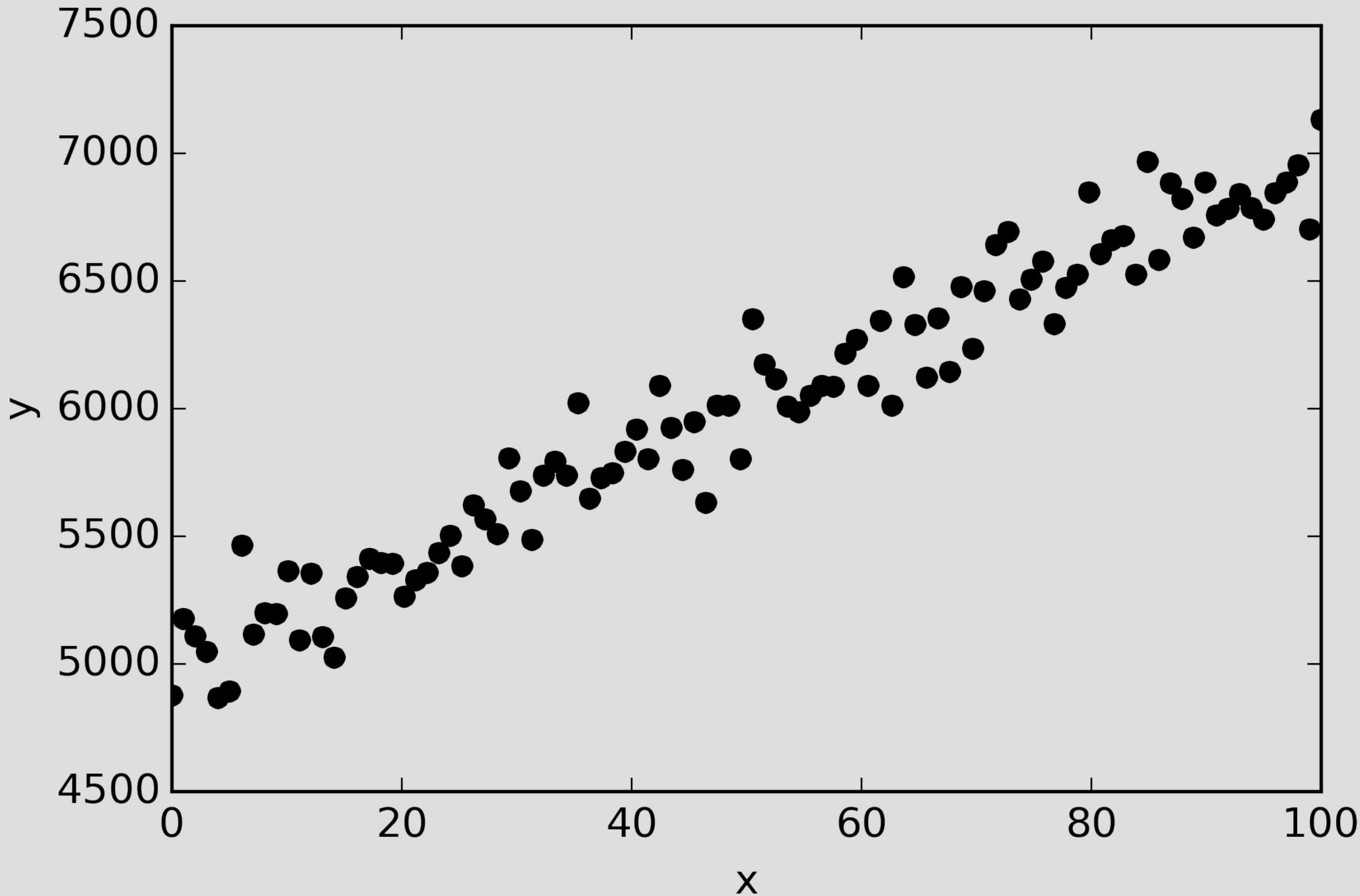
gravmag

seismic

inversion

Módulos

fatiando.inversion



$$y_i = a x_i + b$$

$$y_1 = a x_1 + b$$

$$y_2 = a x_2 + b$$

$$\vdots$$

$$y_N = a x_N + b$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\bar{d} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\bar{d} = \bar{\bar{A}} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\bar{d} = \bar{\bar{A}} \; \bar{p}$$

$$\bar{d} = \bar{\bar{A}} \bar{p}$$

dados preditos

matriz Jacobiana

vetor de parâmetros

$\bar{d}^{o}$

$$\bar{d}^{o}$$

$$\Downarrow$$

$$\hat{\bar{p}}$$

$$\varphi = \| \bar{d}^o - \bar{d} \|^2$$

$$\Downarrow$$

$$\hat{\bar{p}}$$

$$\varphi = \|\bar{d}^{o} - \bar{d}\|^{2}$$

minimizar

$$\hat{\bar{p}}$$

$$\varphi = \| \bar{d}^{o} - \bar{d} \|^{2}$$

função
do ajuste
(misfit)

minimizar

$$\hat{\bar{p}}$$

<u>específico</u>

<u>genérico</u>

especifico

genérico

$$y_i = a x_i + b$$

$$\bar{\bar{A}}$$

específico

$$y_i = a\,x_i + b$$

$$\overline{\overline{A}}$$

genérico

$$\varphi = \left\| \overline{d}^{o} - \overline{d} \right\|^2$$

minimizar

específico

$$y_i = a x_i + b$$

$$\overline{\overline{A}}$$

usuário implementa

fatiando.inversion

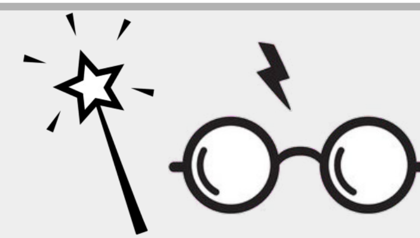
genérico

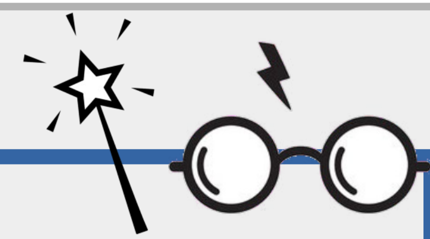
$$\varphi = \| \bar{d}^o - \bar{d} \|^2$$

minimizar

```
from fatiando.inversion import Misfit

class Regressao(Misfit):
    def __init__(self, x, y):
        Misfit.__init__(self, data=y, nparams=2,
                        islinear=True)
        self.x = x

    def predicted(self, p):
        a, b = p
        return a*self.x + b

    def jacobian(self, p):
        A = np.empty((self.ndata,
                      self.nparams))
        A[:, 0] = self.x
        A[:, 1] = 1
        return A
```

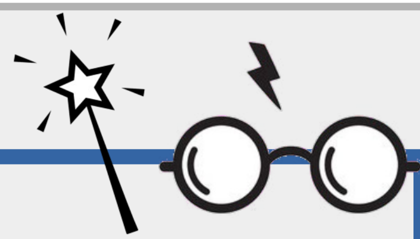
```python
from fatiando.inversion import Misfit

class Regressao(Misfit):
    def __init__(self, x, y):
        Misfit.__init__(self, data=y, nparams=2,
                        islinear=True)
        self.x = x

    def predicted(self, p):
        a, b = p
        return a*self.x + b

    def jacobian(self, p):
        A = np.empty((self.ndata,
                      self.nparams))
        A[:, 0] = self.x
        A[:, 1] = 1
        return A
```





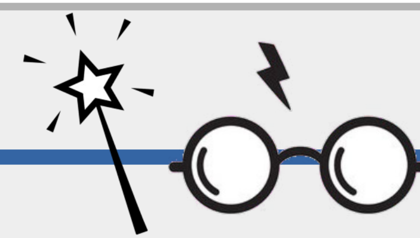
```python
from fatiando.inversion import Misfit

class Regressao(Misfit):
    def __init__(self, x, y):
        Misfit.__init__(self, data=y, nparams=2,
                        islinear=True)
        self.x = x

    def predicted(self, p):
        a, b = p
        return a*self.x + b

    def jacobian(self, p):
        A = np.empty((self.ndata,
                      self.nparams))
        A[:, 0] = self.x
        A[:, 1] = 1
        return A
```

```
from fatiando.inversion import Misfit

class Regressao(Misfit):
    def __init__(self, x, y):
        Misfit.__init__(self, data=y, nparams=2,
                        islinear=True)
        self.x = x

    def predicted(self, p):
        a, b = p
        return a*self.x + b

    def jacobian(self, p):
        A = np.empty((self.ndata,
                      self.nparams))
        A[:, 0] = self.x
        A[:, 1] = 1
        return A
```



$$y_i = a x_i + b$$

```python
from fatiando.inversion import Misfit
```



```python
class Regressao(Misfit):
    def __init__(self, x, y):
        Misfit.__init__(self, data=y, nparams=2,
                        islinear=True)
        self.x = x
```

```python
    def predicted(self, p):
        a, b = p
        return a*self.x + b
```

$$y_i = a x_i + b$$

```python
    def jacobian(self, p):
        A = np.empty((self.ndata,
                      self.nparams))
        A[:, 0] = self.x
        A[:, 1] = 1
        return A
```

$$\bar{\bar{A}}$$

<u>específico</u>

$$y_i = a x_i + b$$

$$\bar{\bar{A}}$$

usuário implementa

fatiando.inversion

genérico

$$\varphi = \| \bar{d}^o - \bar{d} \|^2$$

minimizar

In:

```
reg = Regressao(x, yo)
reg.fit()
print(reg.estimate_)
```

Out:

```
[   20.395431    4980.22844991]
```

In:

```python
reg = Regressao(x, yo)
reg.fit()
print(reg.estimate_)
```

Out: [20.395431   4980.22844991]

a b

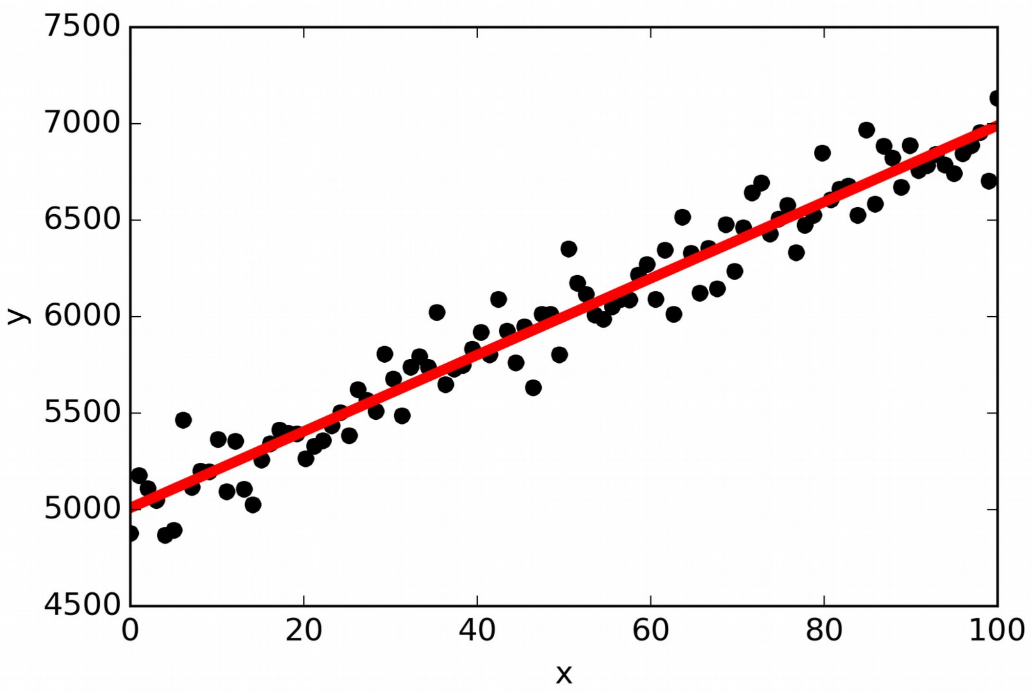
In:
```
reg = Regressao(x, yo)
reg.fit()
print(reg.estimate_)
```

Out: [20.395431   4980.22844991]

a b

In:
```
plt.plot(x, yo, 'ok')
plt.plot(x, reg.predicted(), '-r')
```

Out:



In:

```
reg.config('newton', initial=[1, 1]).fit()
print(reg.estimate_)
```

Out:

```
[   20.395431    4980.22844991]
```

In:
```
reg.config('newton', initial=[1, 1]).fit()
print(reg.estimate_)
```

Out:
```
[   20.395431    4980.22844991]
```

In:
```
reg.config('levmarq', initial=[0, 0]).fit()
print(reg.estimate_)
```

Out:
```
[   20.39562568  4980.21718193]
```

In:
```
reg.config('newton', initial=[1, 1]).fit()
print(reg.estimate_)
```
Out:
```
[   20.395431    4980.22844991]
```

In:
```
reg.config('levmarq', initial=[0, 0]).fit()
print(reg.estimate_)
```
Out:
```
[   20.39562568  4980.21718193]
```

In:
```
reg.config('acor',
           bounds=[0, 1000, 0, 10e5]).fit()
print(reg.estimate_)
```
Out:
```
[   20.39543101  4980.22844959]
```

minimizar $\varphi(\bar{p}) = \|\bar{d}^{o} - \bar{d}\|^{2}$

minimizar $\|\bar{d}^o - \bar{d}\|^2$

instável

minimizar $\bar{d}^{o} - \bar{d}\,\|^{2}$

instável

minimizar

$$\Gamma(\bar{p}) = \varphi(\bar{p}) + \mu\,\theta(\bar{p})$$

minimizar **instável** $\bar{d}^o - \bar{d}\,\|^2$

minimizar

função regularizadora

$$\Gamma(\bar{p}) = \varphi(\bar{p}) + \mu\,\theta(\bar{p})$$

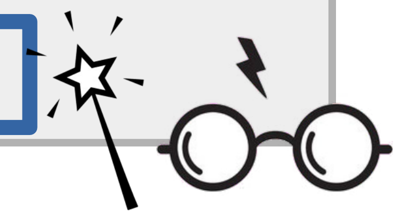
escalar

In:

```
from fatiando.inversion import Damping
phi = Regressao(x, yo)
gamma = phi + 10e-7*Damping(nparams=2)
```

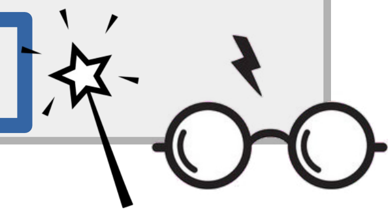
In:

```
from fatiando.inversion import Damping
phi = Regressao(x, yo)
gamma = phi + 10e-7*Damping(nparams=2)
```



$$\Gamma(\bar{p}) = \varphi(\bar{p}) + \mu\,\theta(\bar{p})$$

In:
```
from fatiando.inversion import Damping
phi = Regressao(x, yo)
gamma = phi + 10e-7*Damping(nparams=2)
```



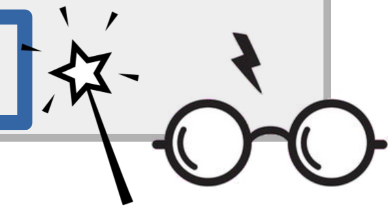
$$\Gamma(\bar{p}) = \varphi(\bar{p}) + \mu\theta(\bar{p})$$

In:
```
gamma.fit()
print(gamma.estimate_)
```

Out:
```
[    20.395431     4980.22844991]
```

In:

```
from fatiando.inversion import Damping
phi = Regressao(x, yo)
gamma = phi + 10e-7*Damping(nparams=2)
```



$$\Gamma(\bar{p}) = \varphi(\bar{p}) + \mu\,\theta(\bar{p})$$

In:

```
gamma.fit()
print(gamma.estimate_)
```

Out:

```
[   20.395431     4980.22844991]
```

In:

```
gamma.config('acor',
             bounds=[0, 1000, 0, 10e5]).fit()
print(gamma.estimate_)
```

Out:

```
[   20.14654754   4992.15040076]
```

conclusão

fatiando a terra

biblioteca: funções e classes

fatiando a terra

- biblioteca: funções e classes
  - modelagem, processamento, visualização

fatiando a terra

- biblioteca: funções e classes
  - modelagem, processamento, visualização
    - inversão: fatiando.inversion

fatiando a terra

- biblioteca: funções e classes
  - modelagem, processamento, visualização
    - inversão: fatiando.inversion
      - Simples ➡ Complexos (reais)

fatiando a terra

- biblioteca: funções e classes
  - modelagem, processamento, visualização
    - inversão: fatiando.inversion
      - Simples ➡ Complexos (reais)

        Caching

        Matrizes esparsas

        BLAS, LAPACK, MKL

Usuários: downloads, citações

- Usuários: downloads, citações
- Colaboradores: locais e externos

Usuários: downloads, citações

Colaboradores: locais e externos

Open-source: BSD license

Usuários: downloads, citações

Colaboradores: locais e externos

Open-source: BSD license

fatiando.org

Usuários: downloads, citações

Colaboradores: locais e externos

Open-source: BSD license

fatiando.org

github.com/fatiando

fatiando.org

Tesseroids

modelagem direta
aprox. esférica

fatiando a terra

modelagem direta
otimização
regularização, etc

Novo método

Introdução

Tesseroids

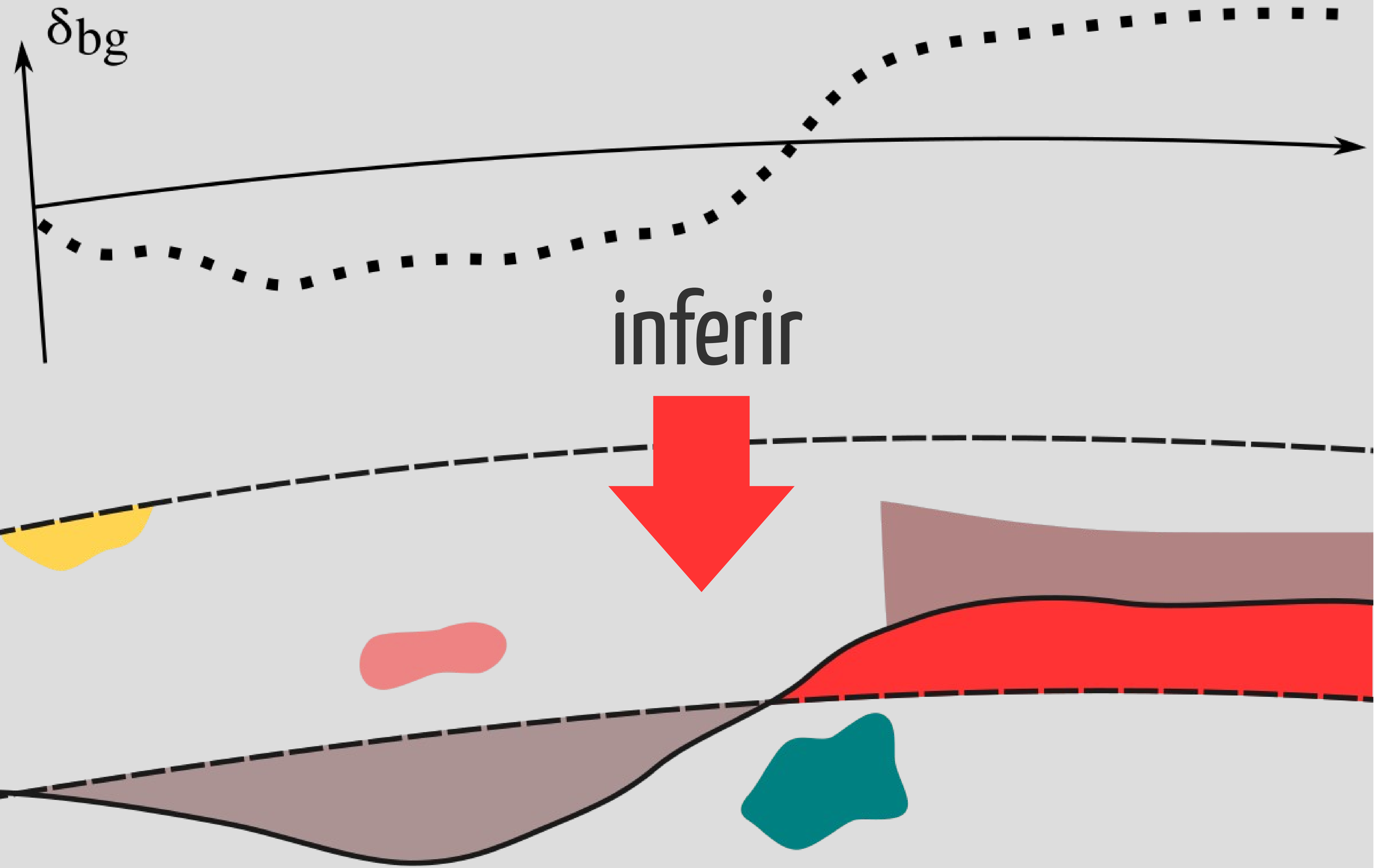
Fatiando a Terra

Inversão Moho

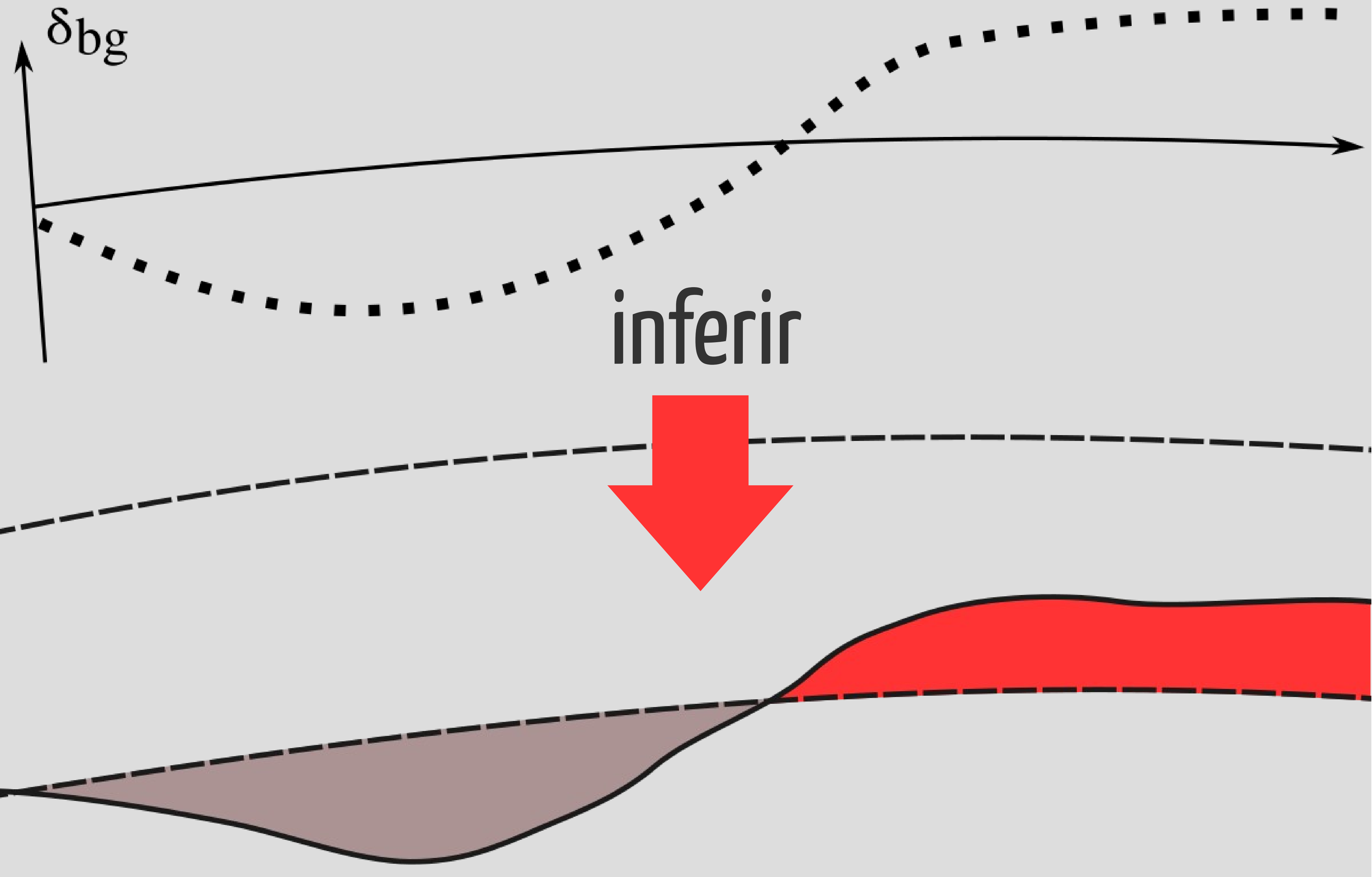
Conclusão

Inversão não-linear rápida em coordenadas esféricas

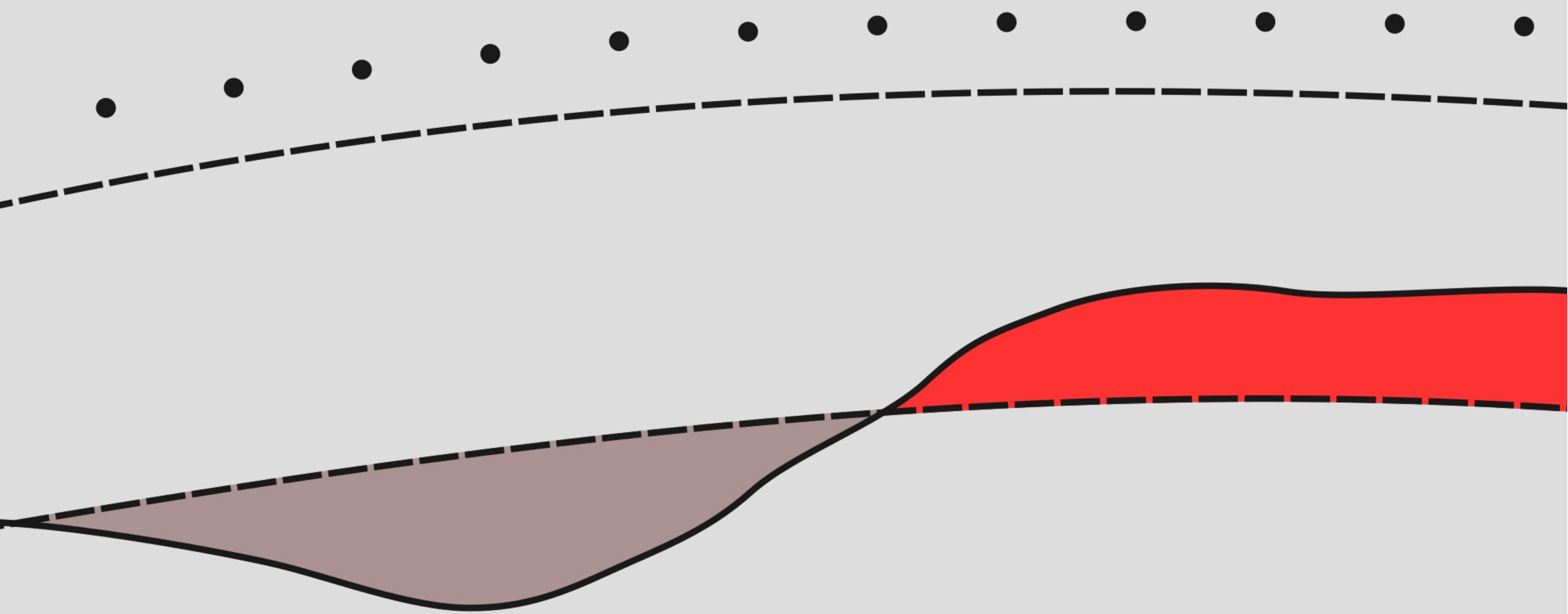
com aplicação na Moho da América do Sul

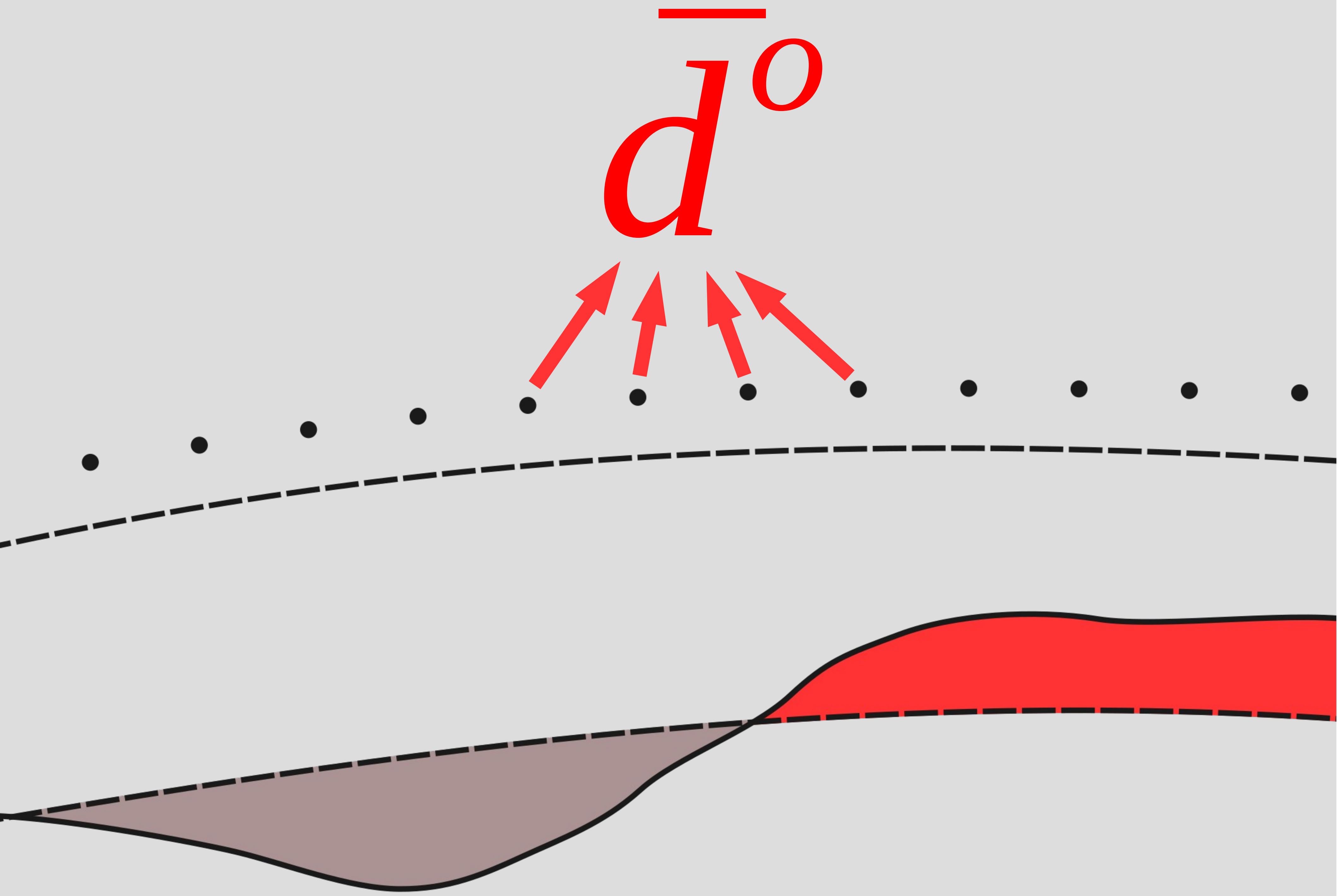


δ_{bg}
inferir

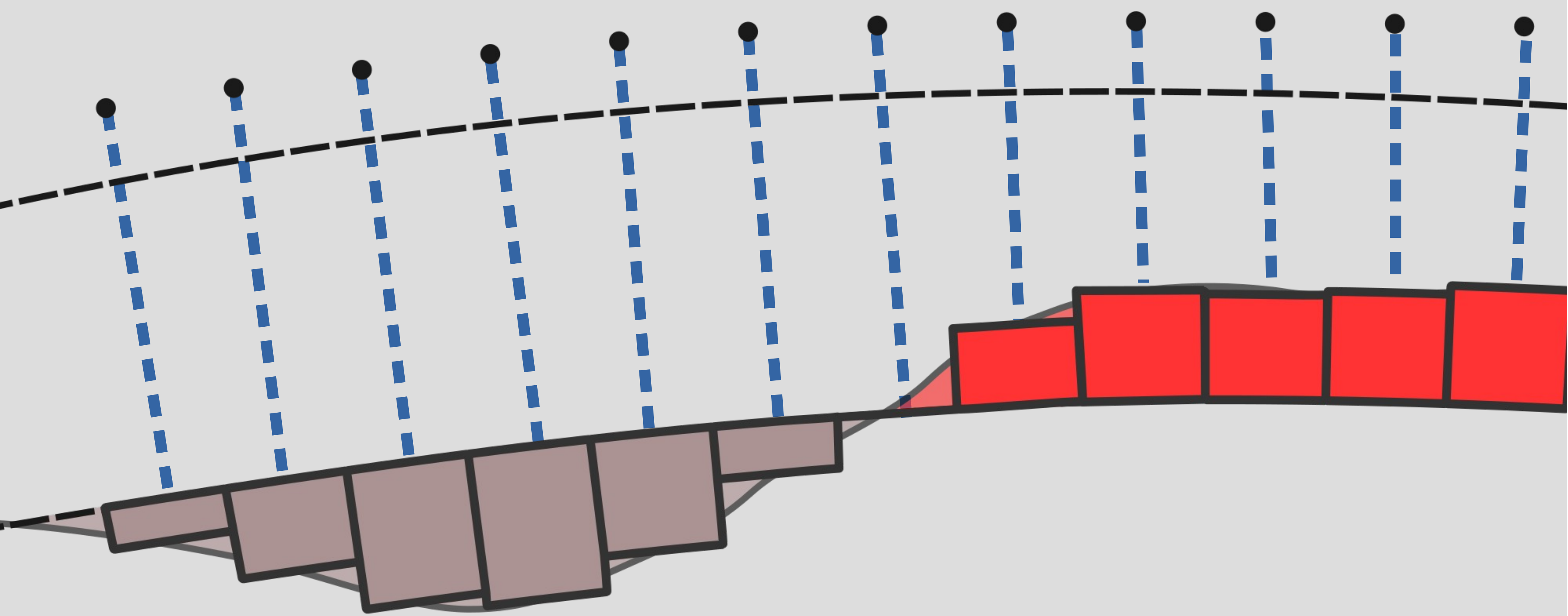

δ_{bg}
inferir

Grid de observações

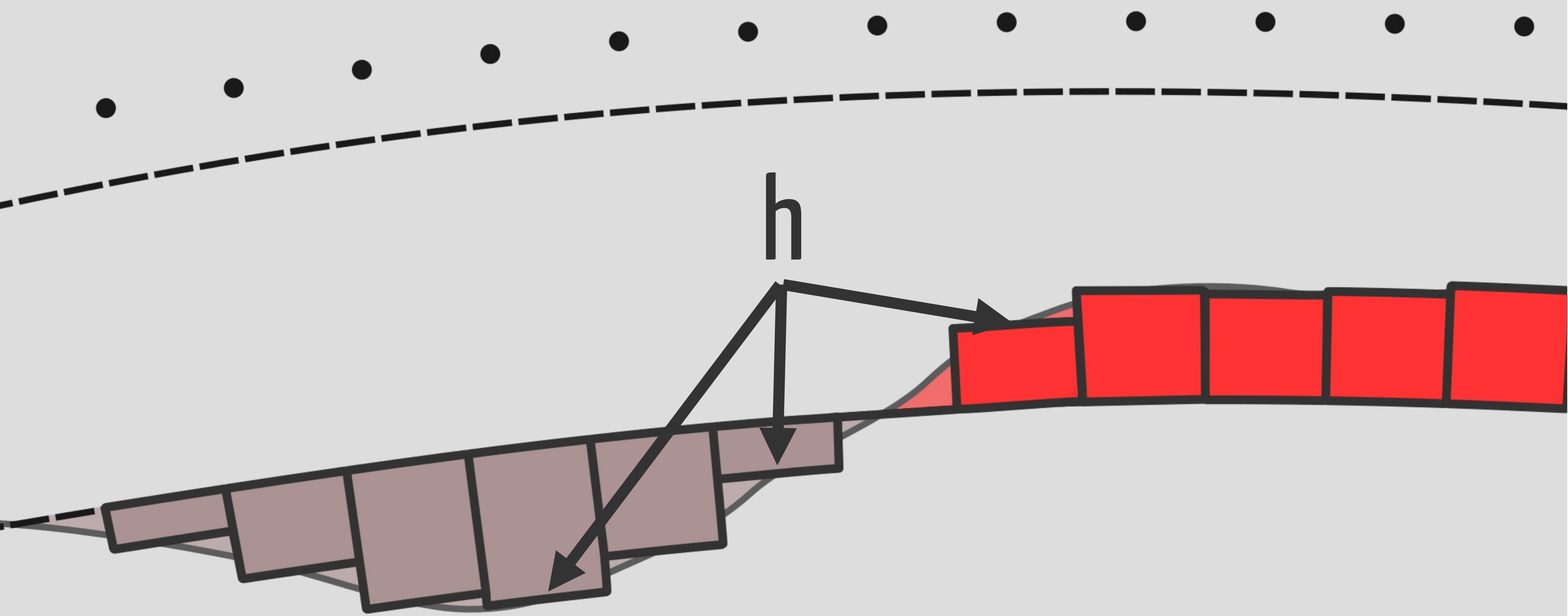


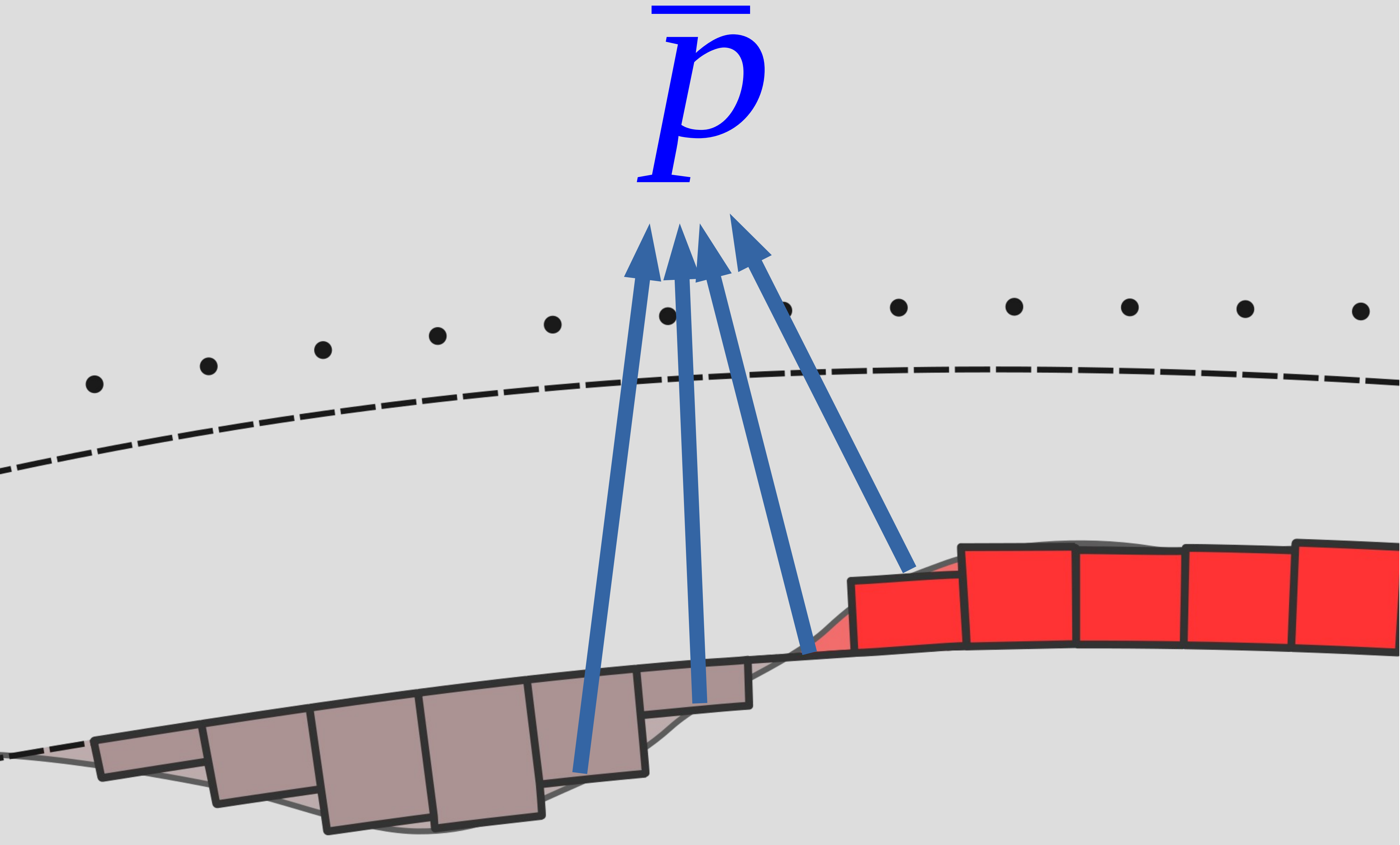

$\bar{d}^o$

1 tesseroide para cada



parâmetros = h




$\bar{p}$

Estimar $\bar{p}$

a partir de $\bar{d}^{o}$

inversão
não-linear

$$d_i = f_i(\bar{p})$$

$$d_i = f_i(\bar{p})$$

modelagem direta
(tesseroides)

Resíduos

$$\bar{r} = \bar{d}^{o} - \bar{d}$$

Minimizar

$$\phi(\bar{p}) = \|\bar{r}\|^2$$

Gauss-Newton

$\overline{p}^{\,0}$

$$\bar{p}^0 \longrightarrow \bar{p}^1 = \bar{p}^0 + \Delta\bar{p}^0$$

$$\bar{p}^{0} \rightarrow \bar{p}^{1} = \bar{p}^{0} + \Delta\bar{p}^{0}$$

$$\left(\bar{\bar{A}}^{kT}\,\bar{\bar{A}}^{k}\right)\Delta\bar{p}^{k} = \bar{\bar{A}}^{kT}\left[\bar{d}^{o} - \bar{d}\left(\bar{p}^{k}\right)\right]$$

$$\bar{p}^{0} \longrightarrow \bar{p}^{1} = \bar{p}^{0} + \Delta\bar{p}^{0}$$

$$\left(\bar{\bar{A}}^{kT}\,\bar{\bar{A}}^{k}\right)\Delta\bar{p}^{k} = \bar{\bar{A}}^{kT}\left[\bar{d}^{o} - \bar{d}\left(\bar{p}^{k}\right)\right]$$

$$\boxed{A_{ij} = \frac{\partial f_i}{\partial p_j} \quad \underline{\text{Jacobiana}}}$$

$$\bar{p}^0 \longrightarrow \bar{p}^1 = \bar{p}^0 + \Delta\bar{p}^0$$

mal posto

$$(\bar{\bar{A}}^{kT}\bar{\bar{A}}^{k})\,\Delta\bar{p}^{k} = \bar{\bar{A}}^{kT}[\bar{d}^{0} - \bar{g}(\bar{p}^{k})]$$

$$\boxed{A_{ij} = \frac{\partial f_i}{\partial p_j} \quad \underline{\text{Jacobiana}}}$$

Regularização

Suavidade

$$\theta(\bar{p}) = \left\| \bar{\bar{R}}\, \bar{p} \right\|^2$$

Função objetivo

$$\Gamma\left(\bar{p}\right) = \phi + \mu\,\theta$$

Função objetivo

ajuste

$$\Gamma\left(\bar{p}\right) = \phi + \mu\,\theta$$

Função objetivo

ajuste

$$\Gamma(\bar{p}) = \phi + \mu\theta$$

regularização

Função objetivo

ajuste

balanço

$$\Gamma(\bar{p}) = \phi + \mu\,\theta$$

regularização

Gauss-Newton

$$\Delta\bar{p} = \left(\bar{\bar{A}}^T \bar{\bar{A}} + \mu \bar{\bar{R}}^T \bar{\bar{R}}\right)^{-1} \left[\bar{\bar{A}}^T \bar{r}^k - \mu \bar{\bar{R}}^T \bar{\bar{R}} \bar{p}^k \right]$$

Gauss-Newton

$$\bar{\Delta p} = (A^T A + \mu R^T R)^{-1} [A^T r^k - \mu R^T R p^k]$$

custoso

(computacionalmente)

1. Construir $\bar{\bar{A}}$

1. Construir $\bar{\bar{A}}$

2. Sistema linear

1. Construir $\bar{\bar{A}}$

2. Sistema linear

3. Calcular $\bar{r}$

Bott

(1960)

$$\Delta \bar{p} = \frac{\bar{r}}{2\pi G \Delta \rho}$$

$$\Delta \bar{p} = \frac{\bar{r}}{\boxed{2\pi G \Delta\rho}}$$

$$\frac{\partial \, \textit{platô de Bouguer}}{\partial h}$$

1. Construir $\bar{\bar{A}}$

2. Sistema linear

3. Calcular $\bar{r}$

1. ~~Construir~~ $\overline{\overline{A}}$

2. ~~Sistema linear~~

3. Calcular $\overline{r}$

rápido

pouca memória

converge

instável

regularização

Silva et al.

(2014)

Bott

Bott

caso particular

Gauss-Newton

$$\Delta \bar{p} = \frac{\bar{r}}{2 \pi G \Delta \rho}$$

caso
particular

$$\Delta \bar{p} = \left(\bar{\bar{A}}^{T} \bar{\bar{A}} \right)^{-1} \bar{\bar{A}}^{T} \bar{r}$$

$$\Delta \bar{p} = \frac{\bar{r}}{\boxed{2\pi G \Delta \rho}}$$

caso
particular

$$\Delta \bar{p} = \boxed{\left(\bar{\bar{A}}^{T} \bar{\bar{A}}\right)^{-1} \bar{\bar{A}}^{T}} \bar{r}$$

$$\Delta\bar{p} = \frac{\bar{r}}{\boxed{2\pi G \Delta\rho}}$$

$$\boxed{\begin{array}{l} A_{ii} = 2\pi G \Delta\rho \\ A_{ij} = 0 \;\; para \;\; i \neq j \end{array}}$$

caso particular

$$\Delta\bar{p} = \boxed{\left(\bar{\bar{A}}^T \bar{\bar{A}}\right)^{-1} \bar{\bar{A}}^T}\,\bar{r}$$

$$\tilde{A}$$

$$\Downarrow$$

$$\begin{pmatrix} 2\pi G\Delta\rho & 0 & \cdots & 0 \\ 0 & 2\pi G\Delta\rho & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2\pi G\Delta\rho \end{pmatrix}$$

nesse

trabalho

Gauss-Newton

$$\Delta\bar{p} = (\bar{\bar{A}}^T \bar{\bar{A}})^{-1} \bar{\bar{A}}^T [\bar{d}^o - \bar{d}(\bar{p}^k)]$$

Gauss-Newton

$$\Delta\bar{p} = (\bar{\bar{A}}^T \bar{\bar{A}})^{-1} \bar{\bar{A}}^T [\bar{d}^o - \bar{d}(\bar{p}^k)]$$

tesseroides

Gauss-Newton

$$\Delta\bar{p} = (\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T [\bar{d}^o - \bar{d}(\bar{p}^k)]$$

$$\tilde{A}_{ii} = 2\pi G \Delta\rho_i$$

tesseroides

regularização

suavidade

$$\Delta\bar{p} = \left(\widetilde{A}^T \widetilde{A} + \mu \bar{\bar{R}}^T \bar{\bar{R}}\right)^{-1} \left[\widetilde{A}^T \bar{r}^k - \mu \bar{\bar{R}}^T \bar{\bar{R}} \bar{p}^k \right]$$

esparsas

$$\Delta\bar{p} = \left(\boxed{\tilde{A}^{T}\tilde{A}} + \mu\,\boxed{\bar{\bar{R}}^{T}\bar{\bar{R}}}\right)^{-1}\left[\tilde{A}^{T}\bar{r}^{k} - \mu\,\bar{\bar{R}}^{T}\bar{\bar{R}}\,\bar{p}^{k}\right]$$

esparsas

$$\Delta \bar{p} = \left(\boxed{\tilde{A}^T \tilde{A}} + \mu \boxed{\bar{\bar{R}}^T \bar{\bar{R}}} \right)^{-1} \left[\tilde{A}^T \bar{r}^k - \mu \bar{\bar{R}}^T \bar{\bar{R}} \bar{p}^k \right]$$

~99.8% tempo de computação

1. Construir $\bar{\bar{A}}$

2. Sistema linear

3. Calcular $\bar{r}$

1. ~~Construir~~ $\bar{\bar{A}}$

Bott

2. Sistema linear

3. Calcular $\bar{r}$

1. ~~Construir~~ $\bar{\bar{A}}$

Bott

2. ~~Sistema linear~~

matrizes esparsas

3. Calcular $\bar{r}$

rápido

pouca memória

converge

instável

regularização

~~instável~~

~~regularização~~

implementação

python™

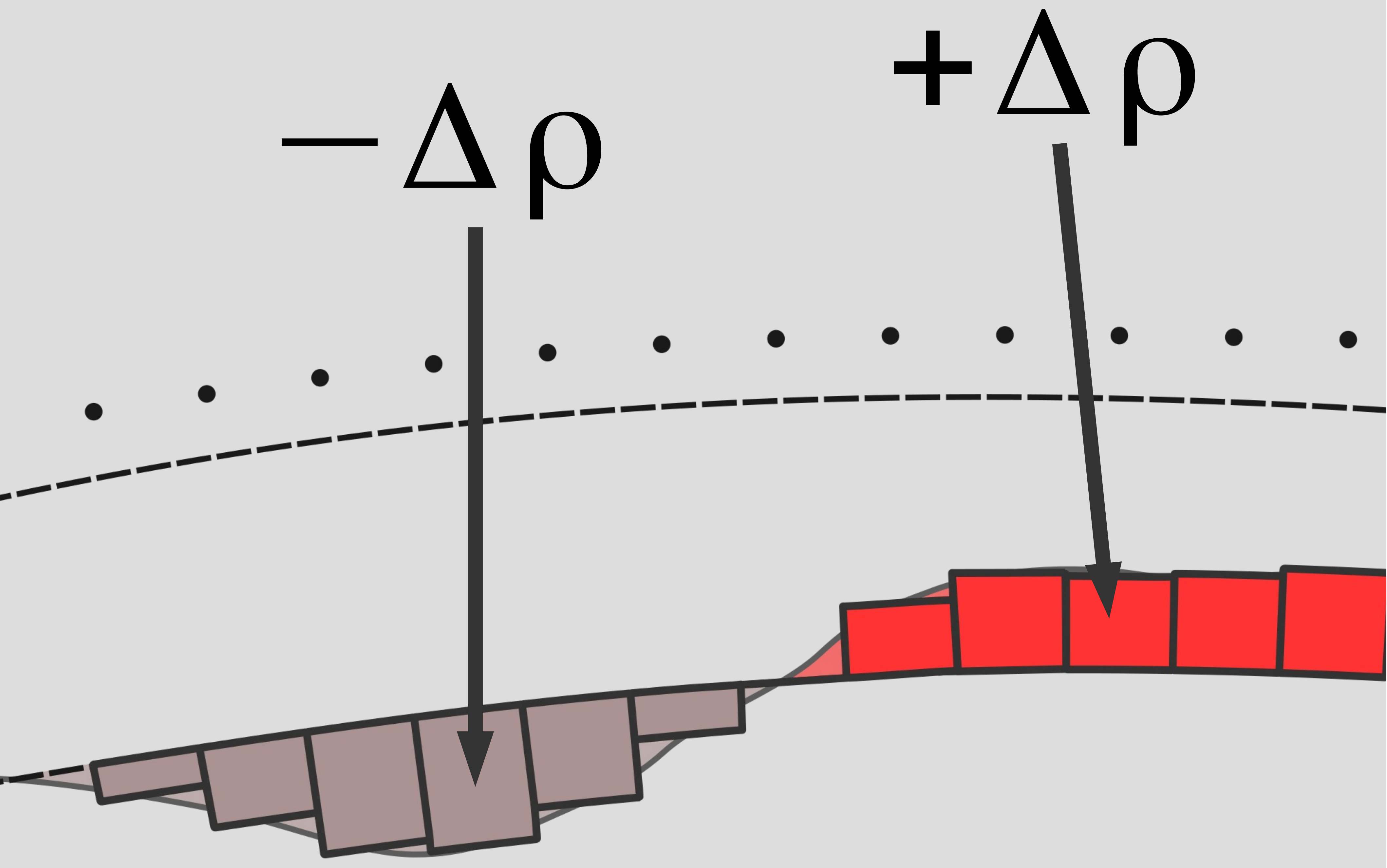
fatiando a terra

modeling and inversion in geophysics

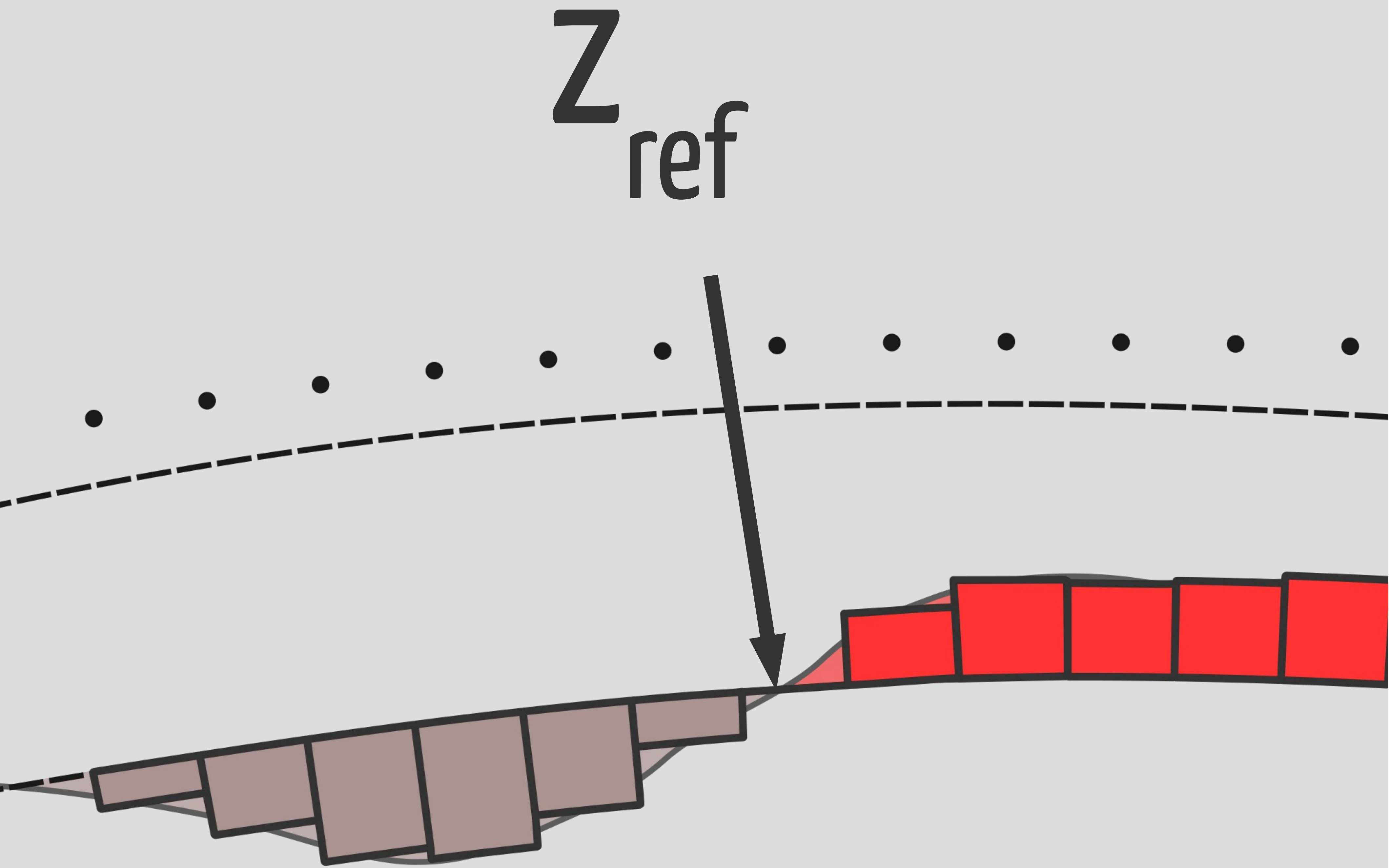
scipy (matrizes esparsas)

estimar

hiperparâmetros



$-\Delta\rho$
$+\Delta\rho$


Z_{ref}

Função objetivo

ajuste

balanço

$$\Gamma(\bar{p}) = \phi + \mu\,\theta$$

regularização

μ

$\Delta\rho$

z_{ref}

validação cruzada

validação | cruzada

μ

validação cruzada

μ

$\Delta\rho$

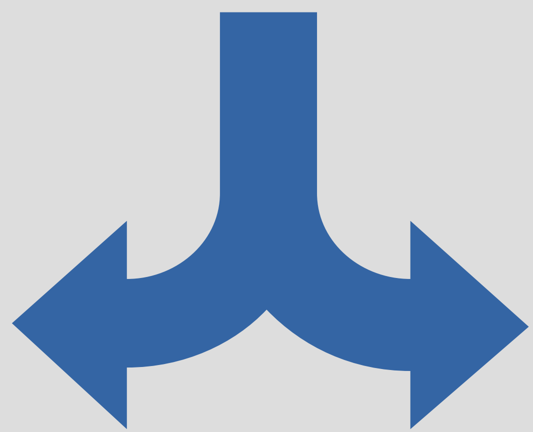
z_{ref}

validação

μ

Separar os dados

$$\overline{d}^{o}$$



Separar os dados

$$\bar{d}^{o}$$

$$\bar{d}^{o}_{inv}$$

Separar os dados

$$\bar{d}^o$$

$$\bar{d}^o_{inv} \qquad \bar{d}^o_{test}$$

para μ_n em $[\mu_{1}, \ldots, \mu_N]$:

para μ_n em $[\mu_1, \ldots, \mu_N]$:

inversão: $\bar{d}^o_{inv} \longrightarrow \hat{\bar{p}}^n$

para μ_n em $[\mu_1, \ldots, \mu_N]$:

inversão: $\bar{d}^{o}_{inv} \longrightarrow \hat{\bar{p}}^{n}$

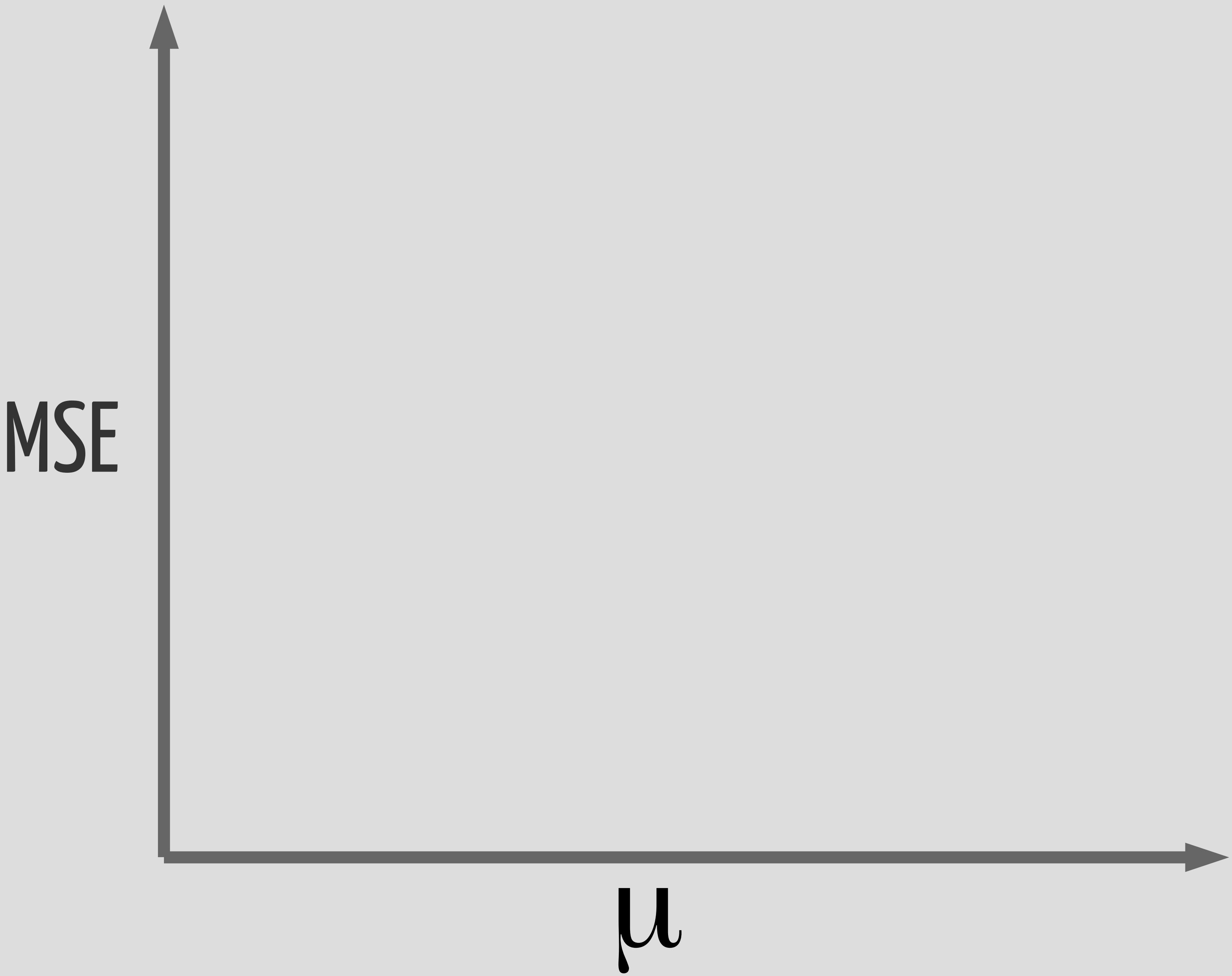
$\hat{\bar{p}}^{n} \longrightarrow$ prever $\bar{d}_{test}^{\;n}$

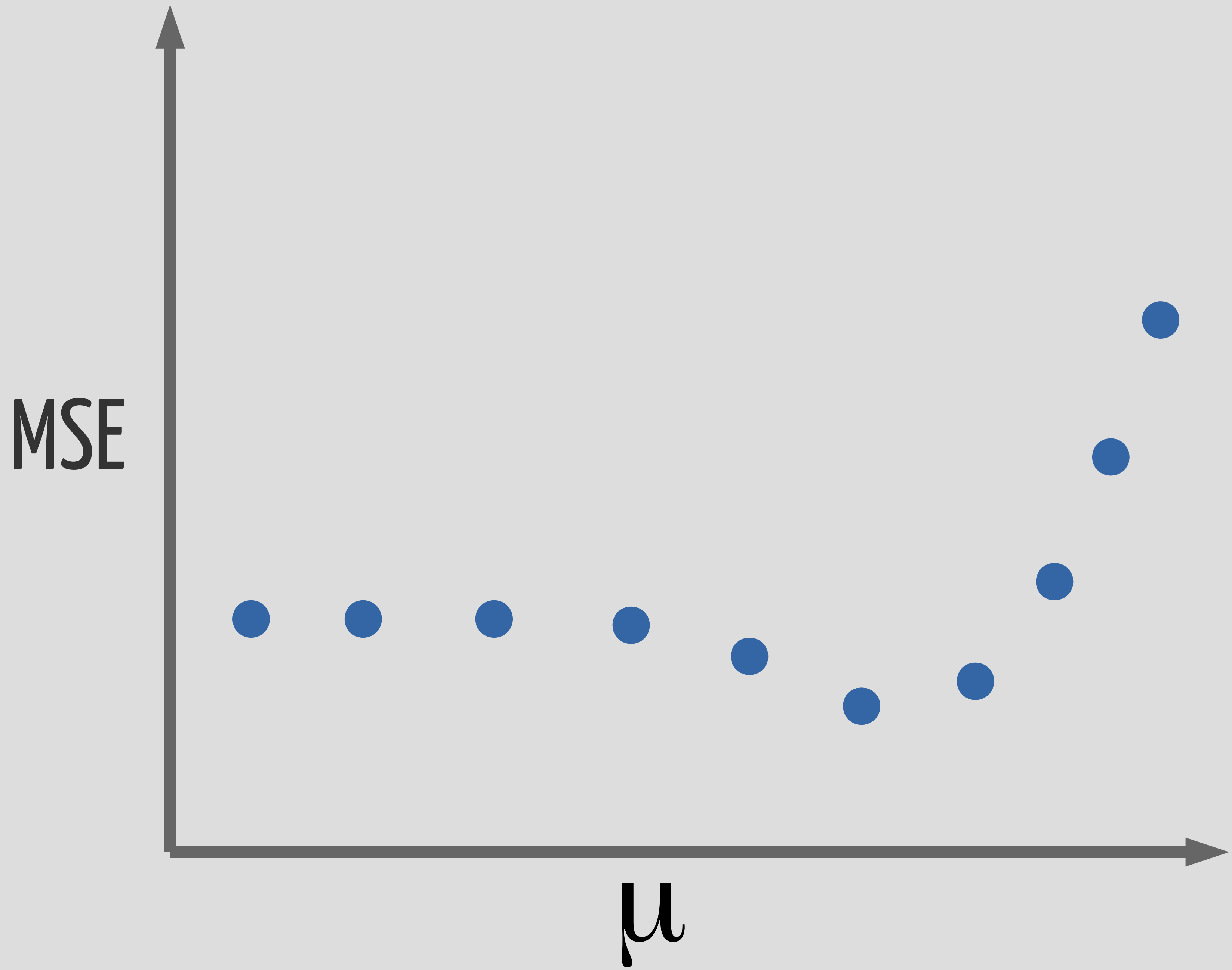
para μ_n em $[\mu_1, \ldots, \mu_N]$:

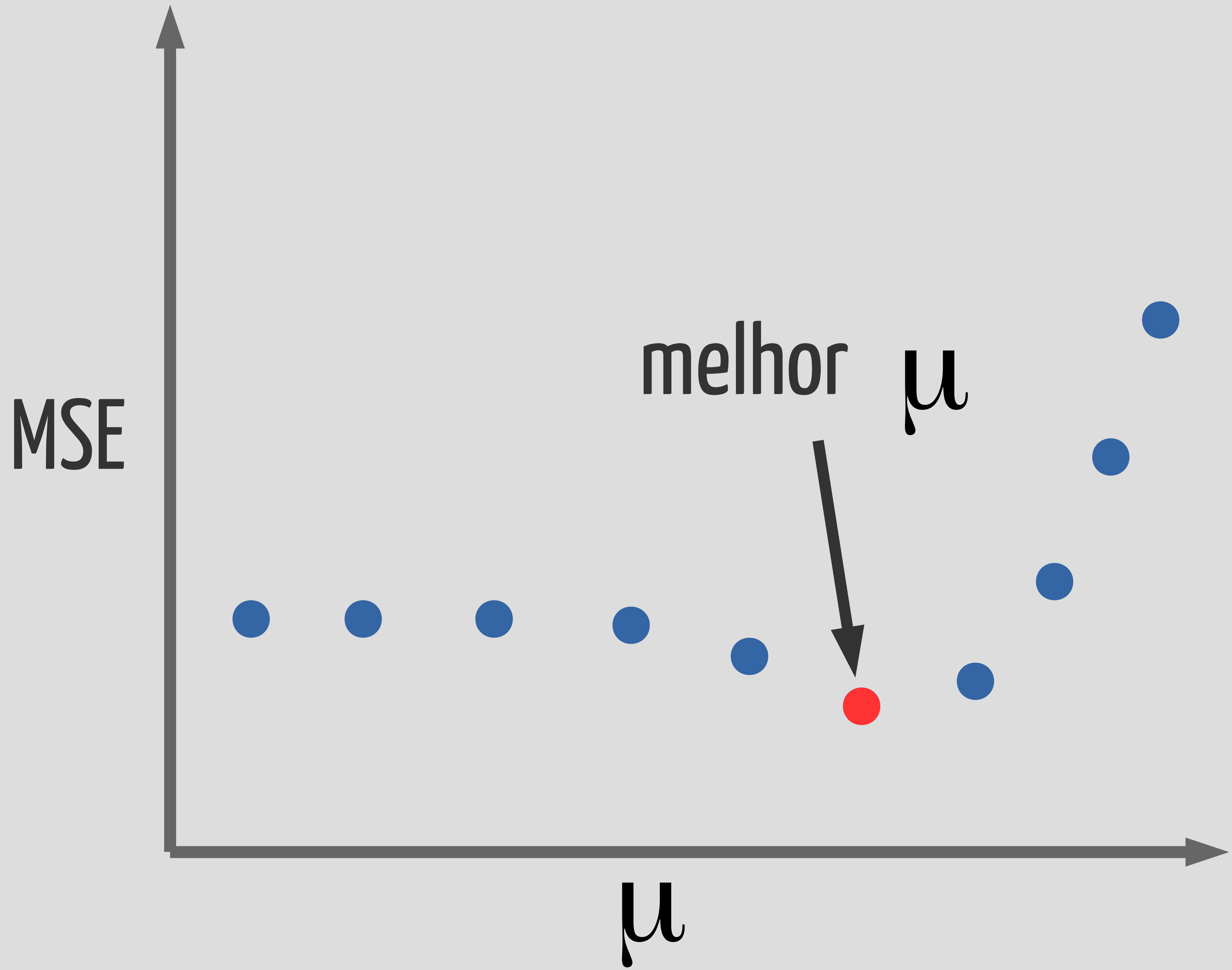
inversão: $\bar{d}^o_{inv} \longrightarrow \hat{\bar{p}}^n$

$\hat{\bar{p}}^n \longrightarrow$ prever $\bar{d}_{test}^n$

$$MSE_n = \frac{\|\bar{d}^o_{test} - \bar{d}_{test}^n\|^2}{N_{test}}$$


MSE
μ


MSE
μ



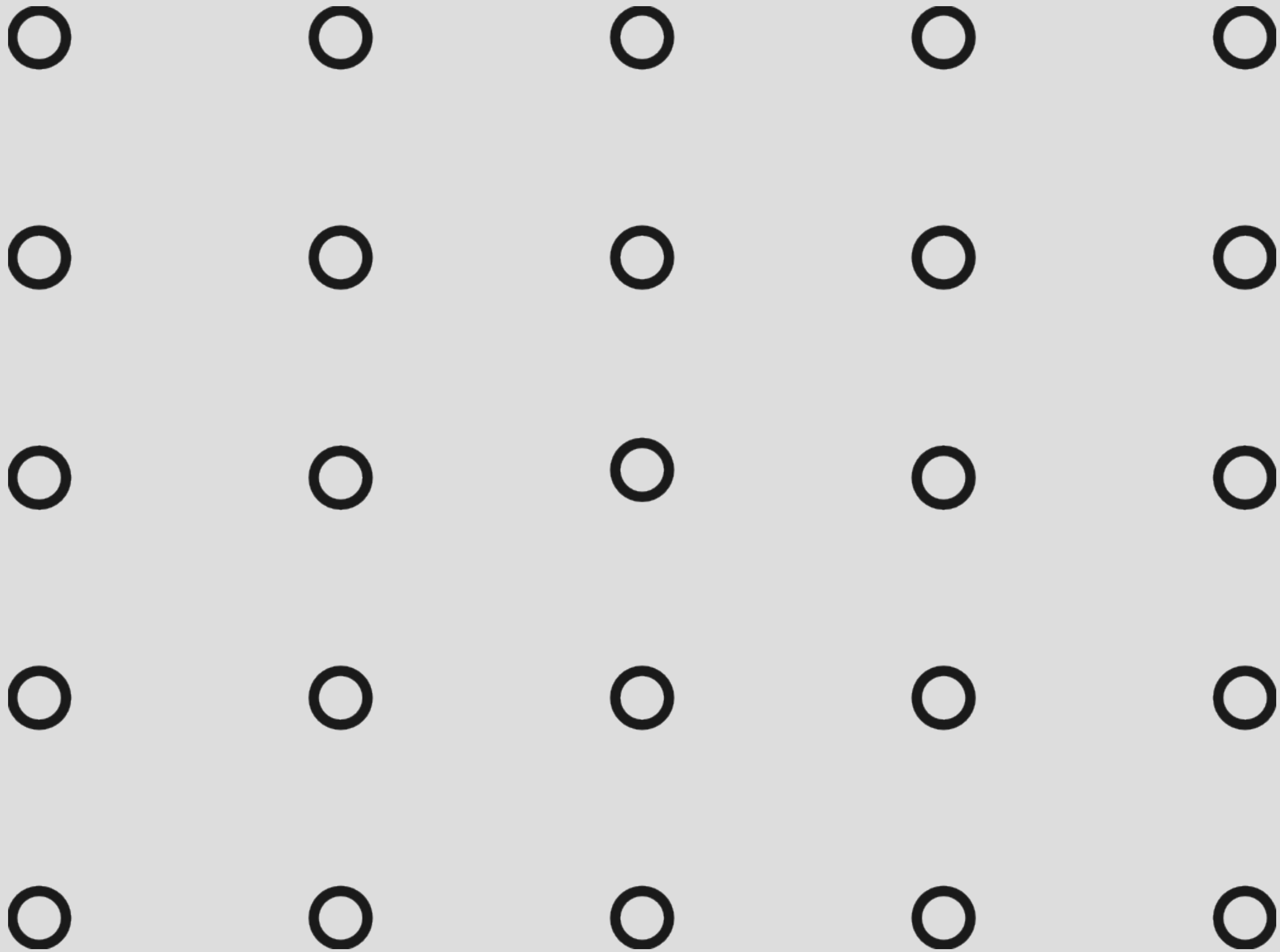
MSE
melhor μ
μ

Separar os dados

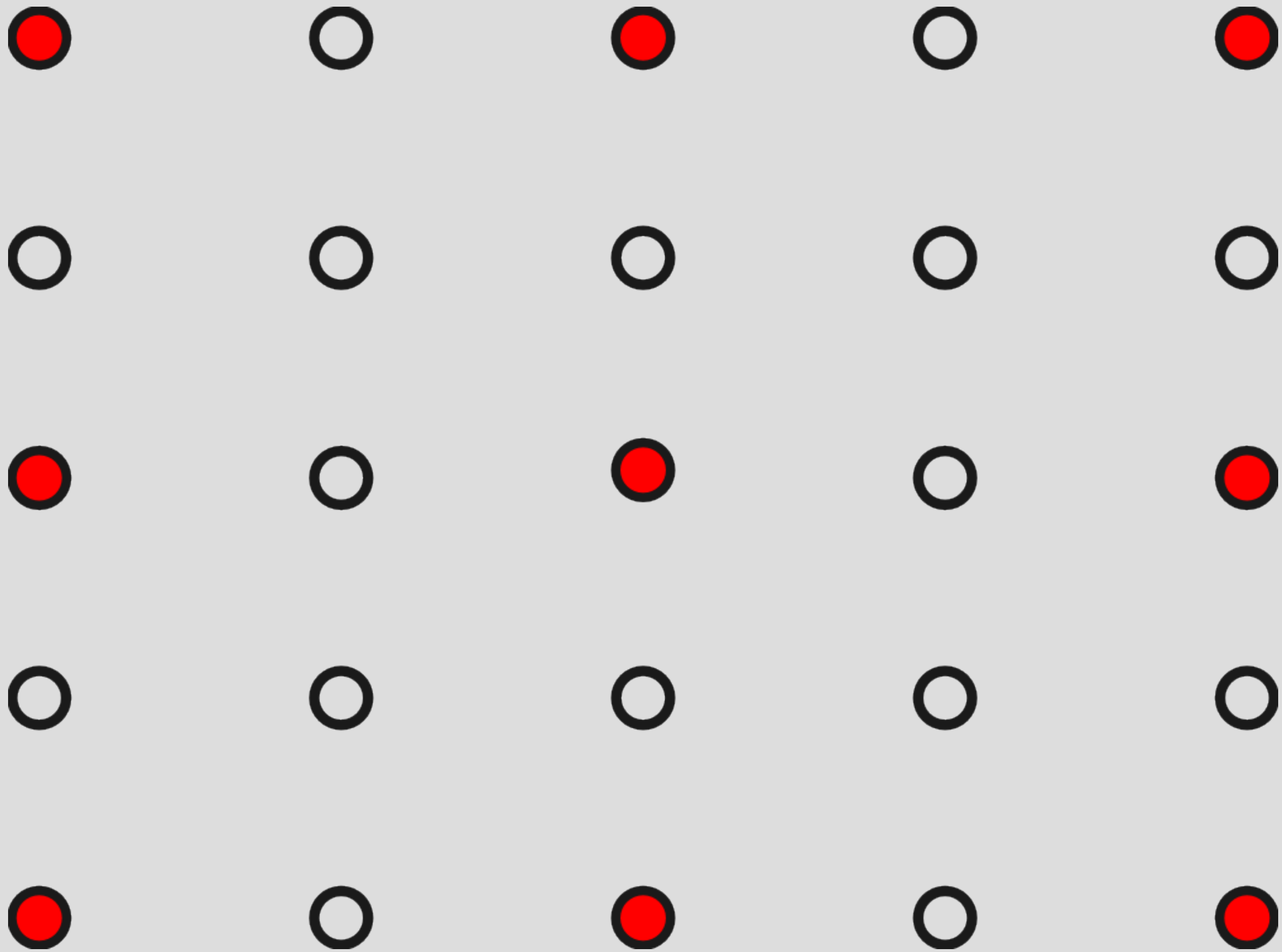
$$\bar{d}^{o}$$

$$\bar{d}^{o}_{inv} \qquad \bar{d}^{o}_{test}$$

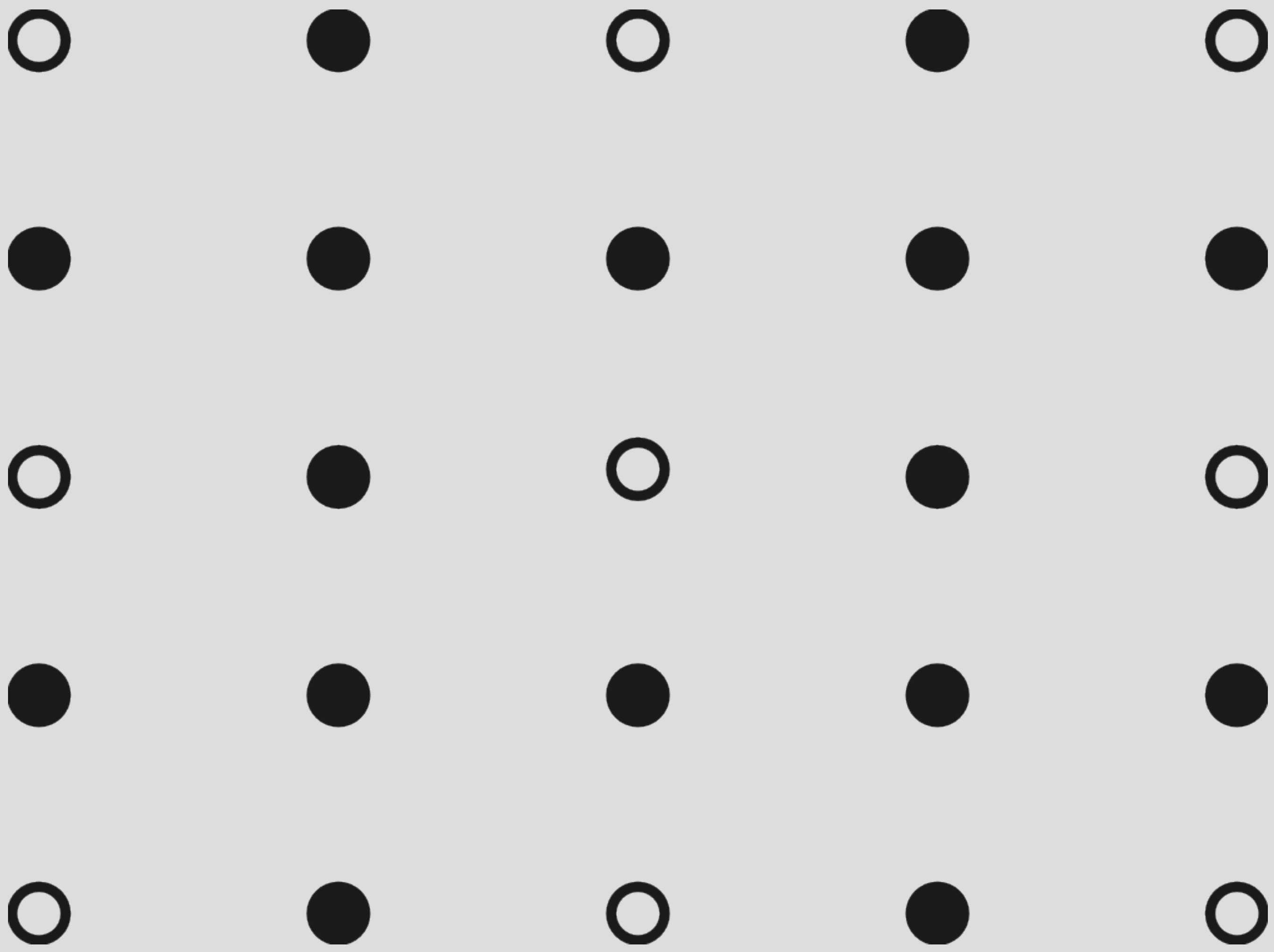
dado completo



dado inversão

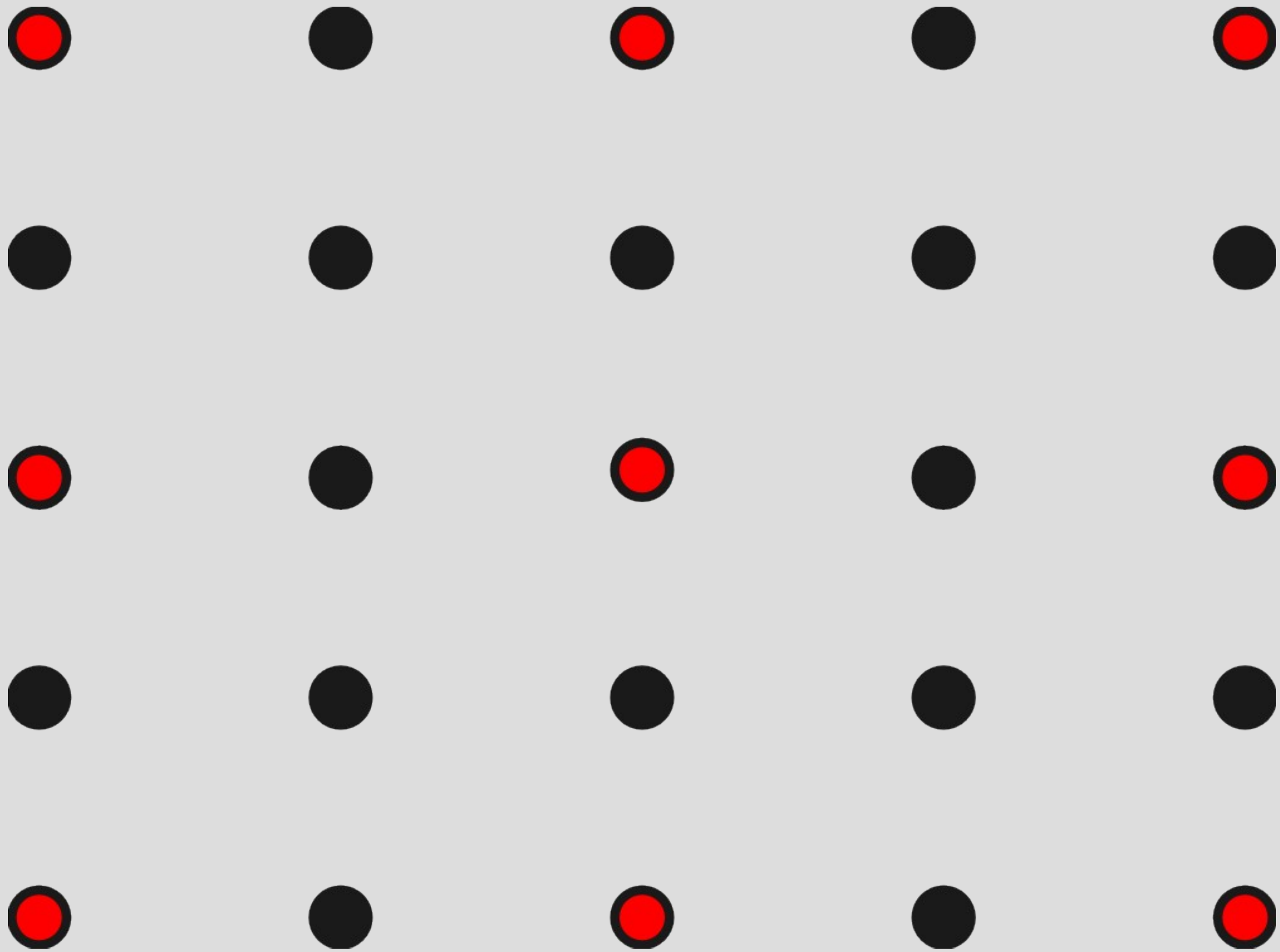


dado teste



dado teste

dado inversão



validação | cruzada

μ | $\Delta\rho$

z_{ref}

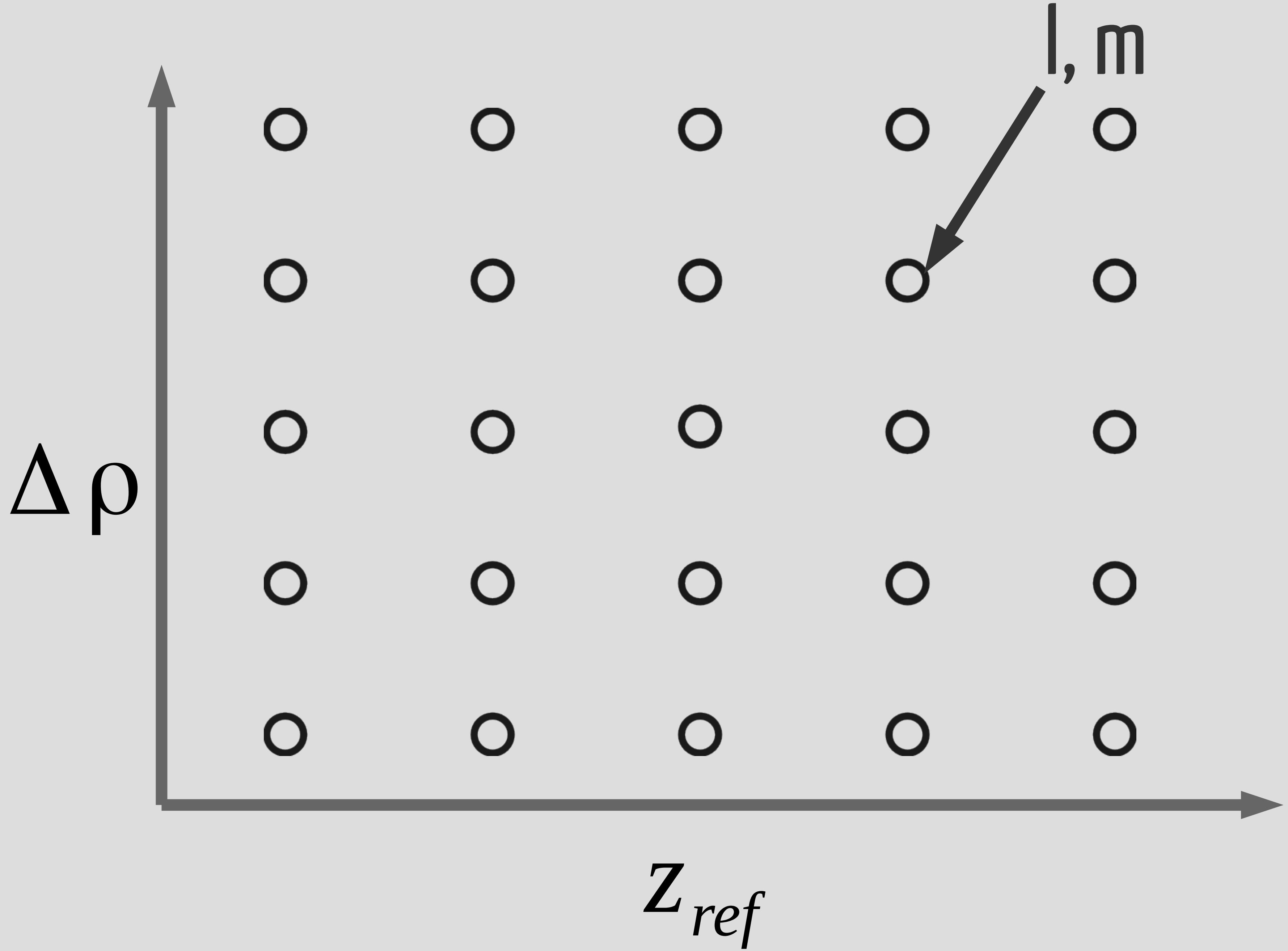
cruzada

$$\Delta\rho$$

$$z_{ref}$$

estimativas pontuais

$$\overline{Z}_s^o$$


l, m
Δ ρ
z_{ref}

para cada $z_{ref,l}$ e $\Delta\rho_m$:

para cada $z_{ref,l}$ e $\Delta\rho_m$:

inversão: $\bar{d}^{o}_{inv} \rightarrow \hat{\bar{p}}^{l,m}$

para cada $z_{ref,l}$ e $\Delta\rho_m$:

inversão: $\bar{d}^{o}_{inv} \rightarrow \hat{\bar{p}}^{l,m}$

interpolar $\hat{\bar{p}}^{l,m} \rightarrow \bar{z}_{s}^{l,m}$

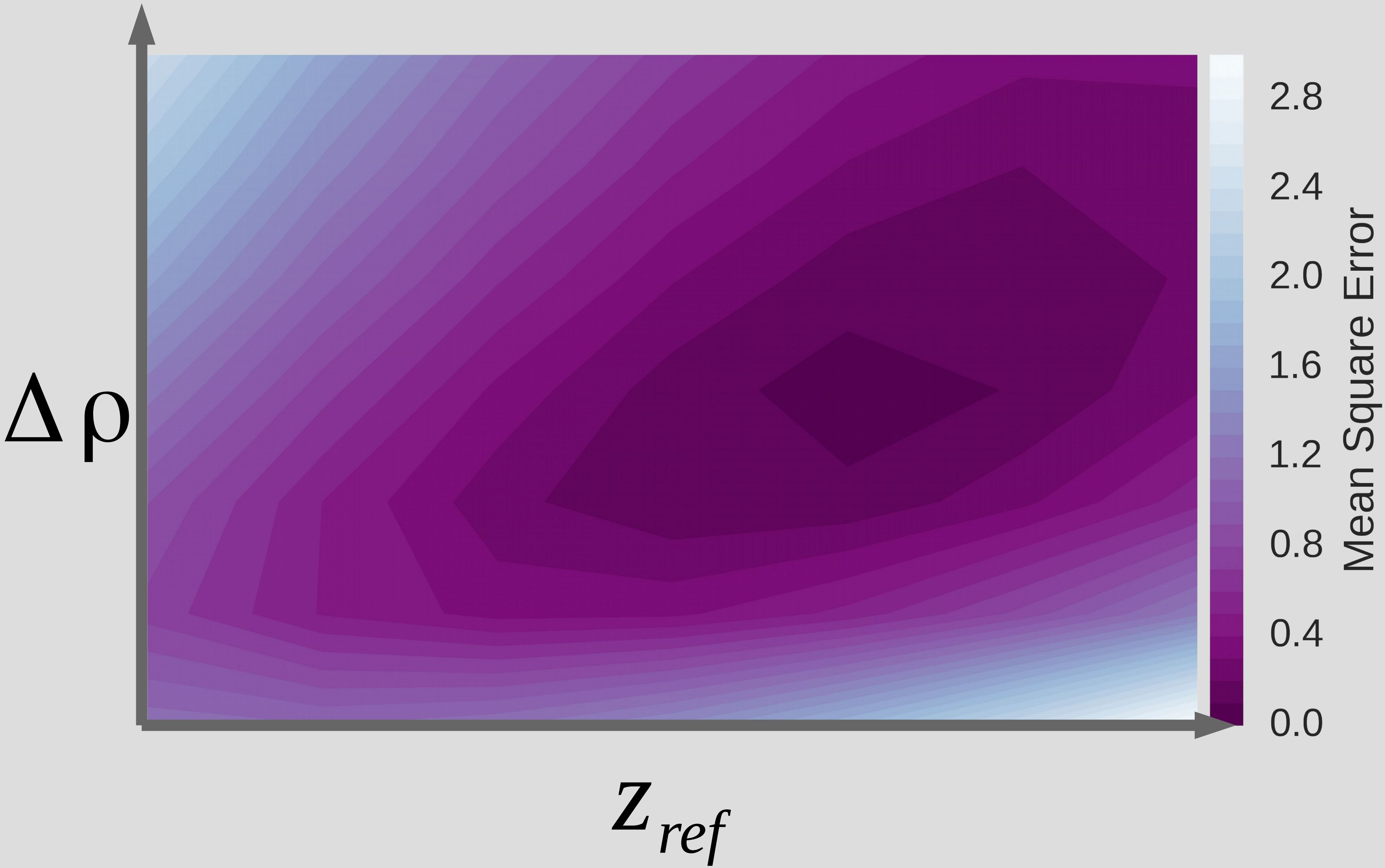
para cada $z_{ref,l}$ e $\Delta\rho_m$:

inversão: $\bar{d}^{o}_{inv} \rightarrow \hat{\bar{p}}^{l,m}$

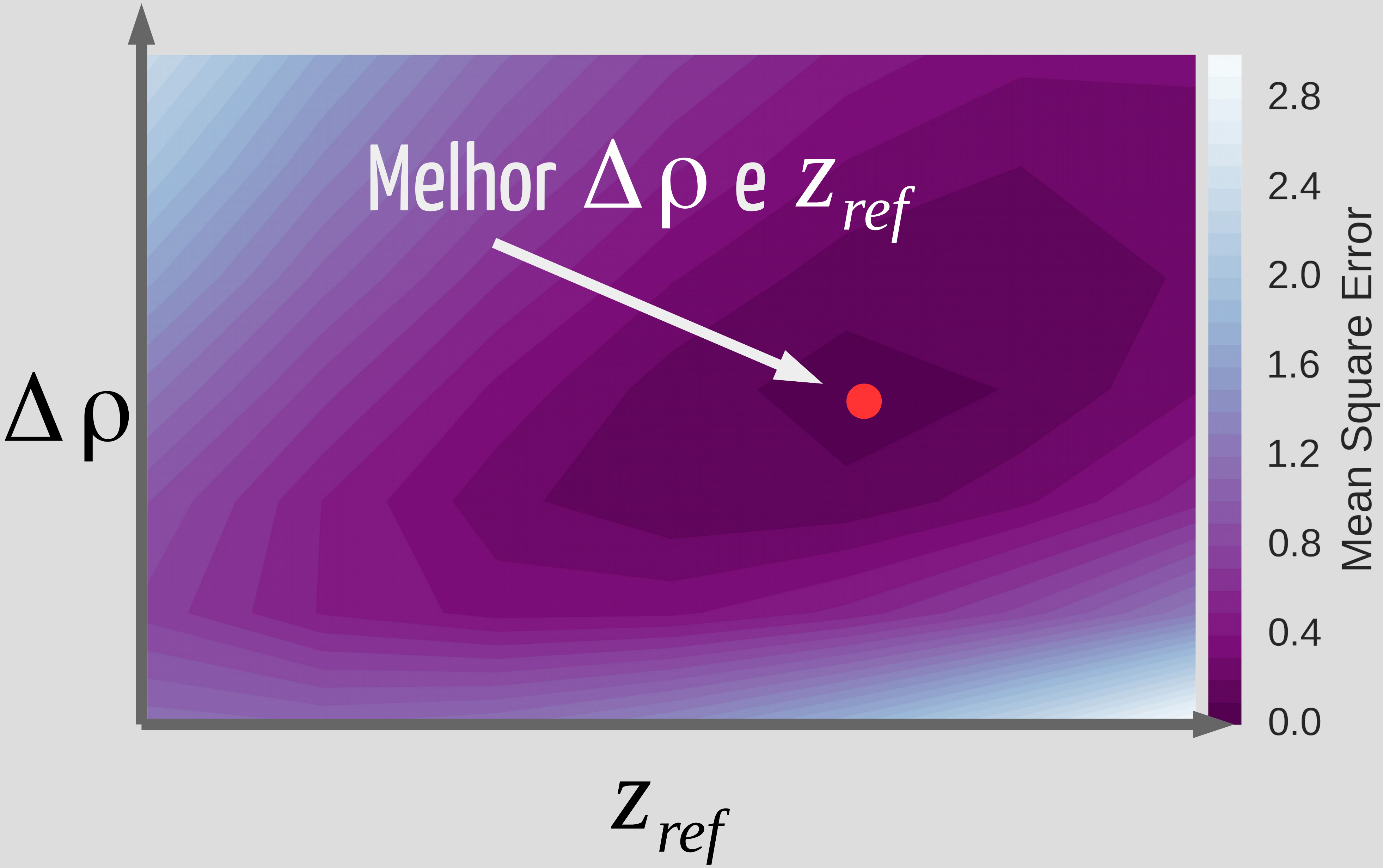
interpolar $\hat{\bar{p}}^{l,m} \rightarrow \bar{z}_s^{l,m}$

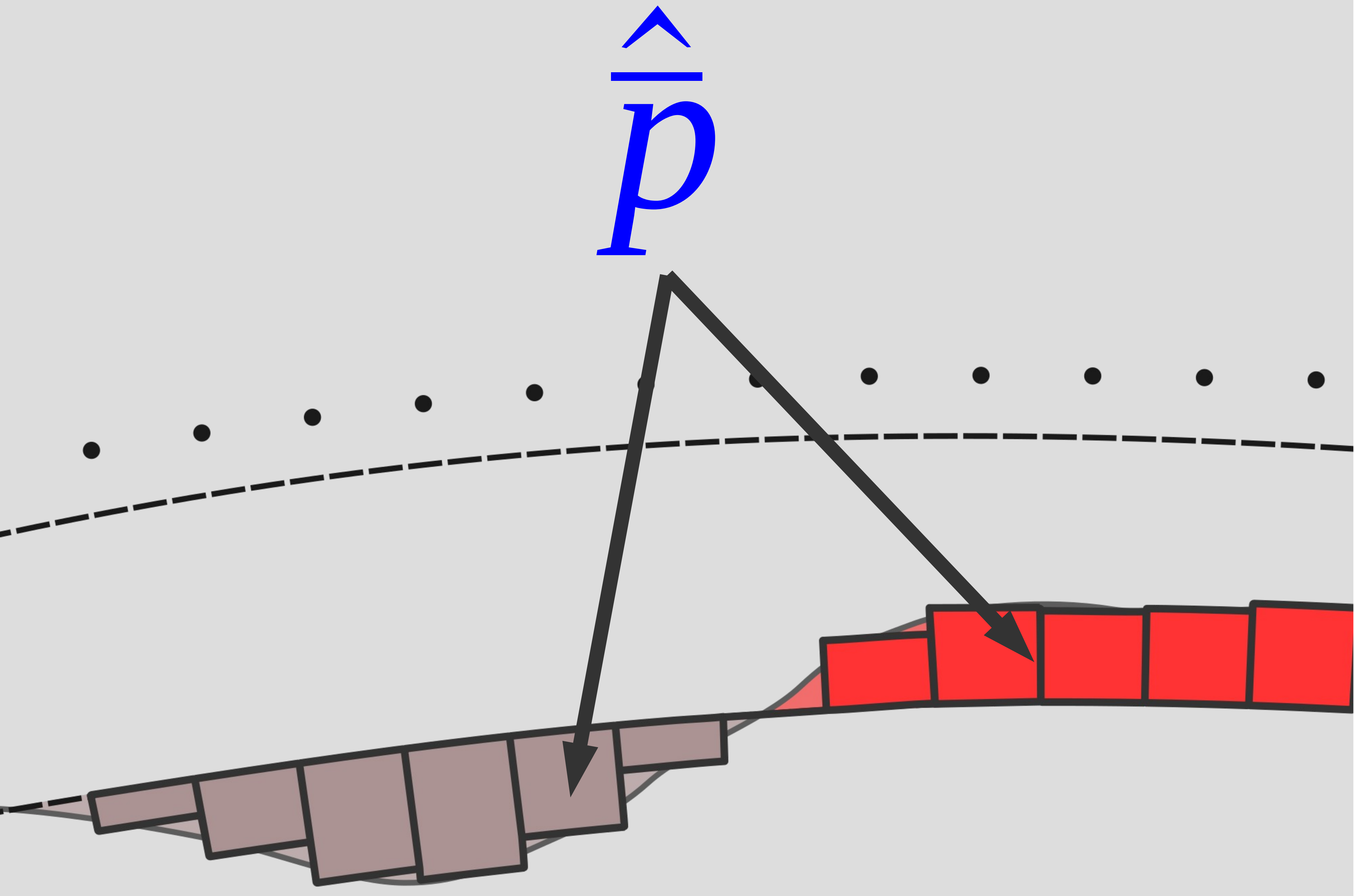
$$MSE_{l,m} = \frac{\|\bar{z}_s^{o} - \bar{z}_s^{l,m}\|^2}{N_s}$$


Δ ρ
z ref

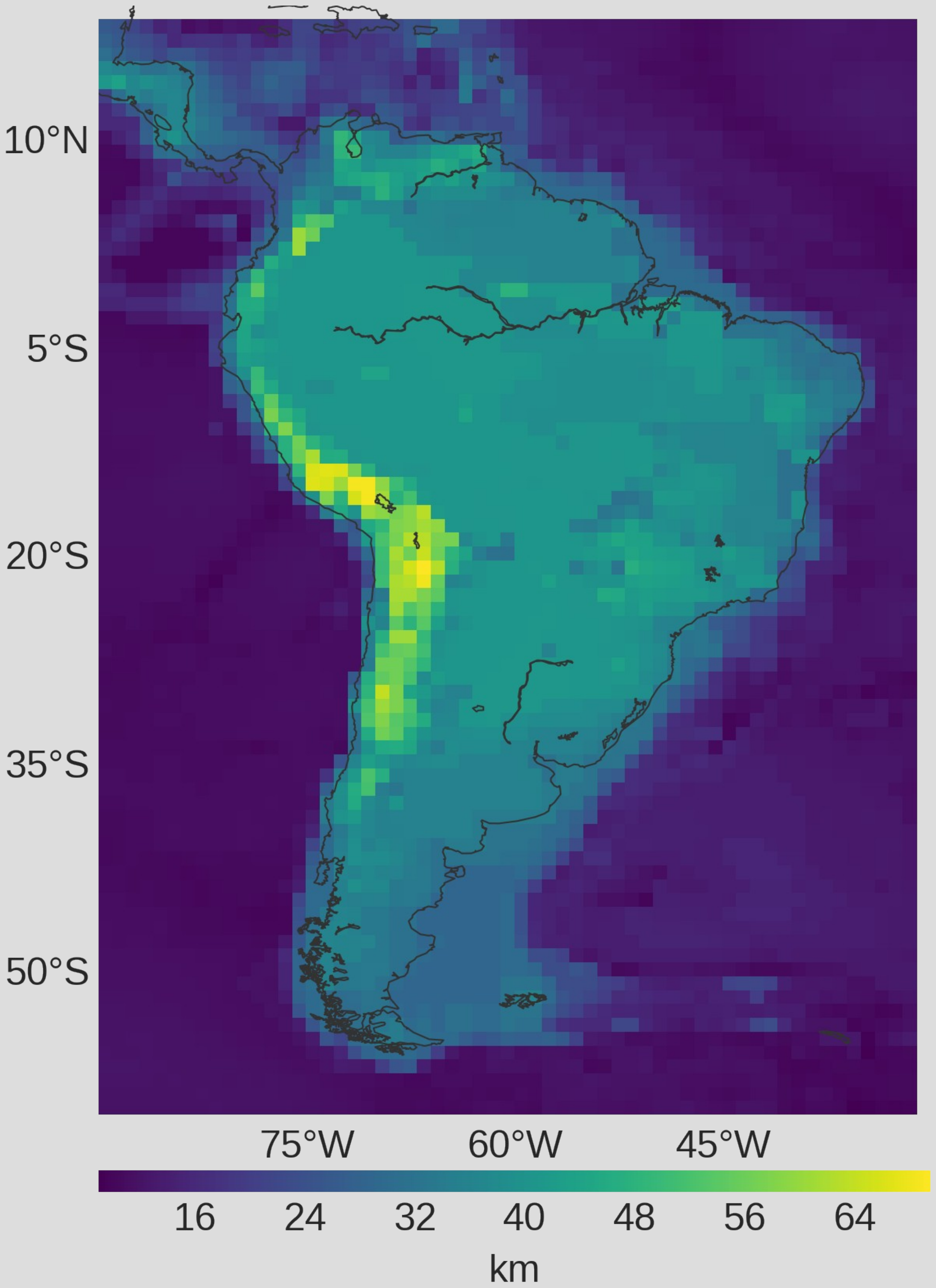


Δ ρ
z ref
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0.0
Mean Square Error


Melhor Δ ρ e z_{ref}
Δ ρ
z_{ref}
Mean Square Error
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0.0


$\hat{\bar{p}}$

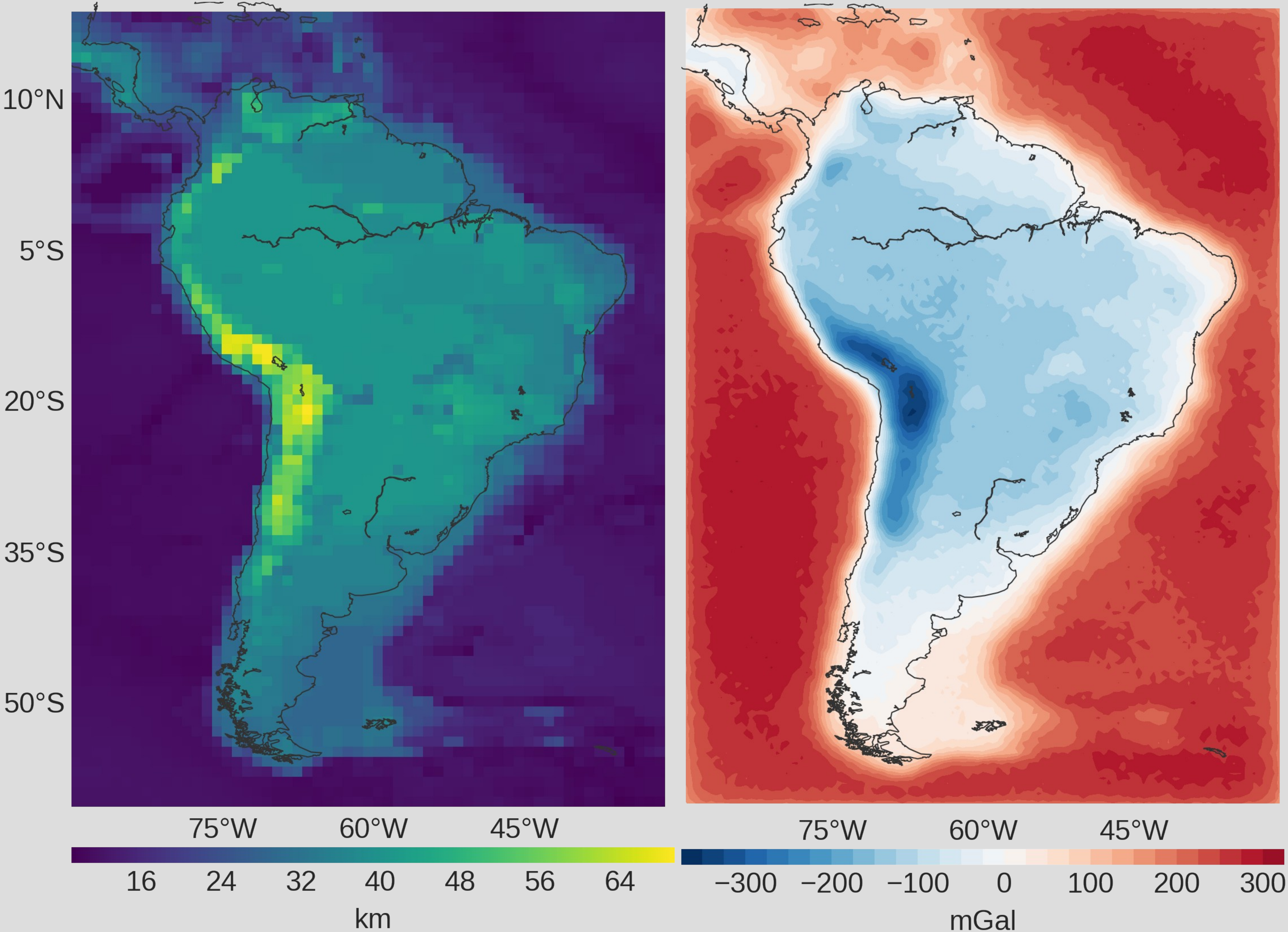
sintético



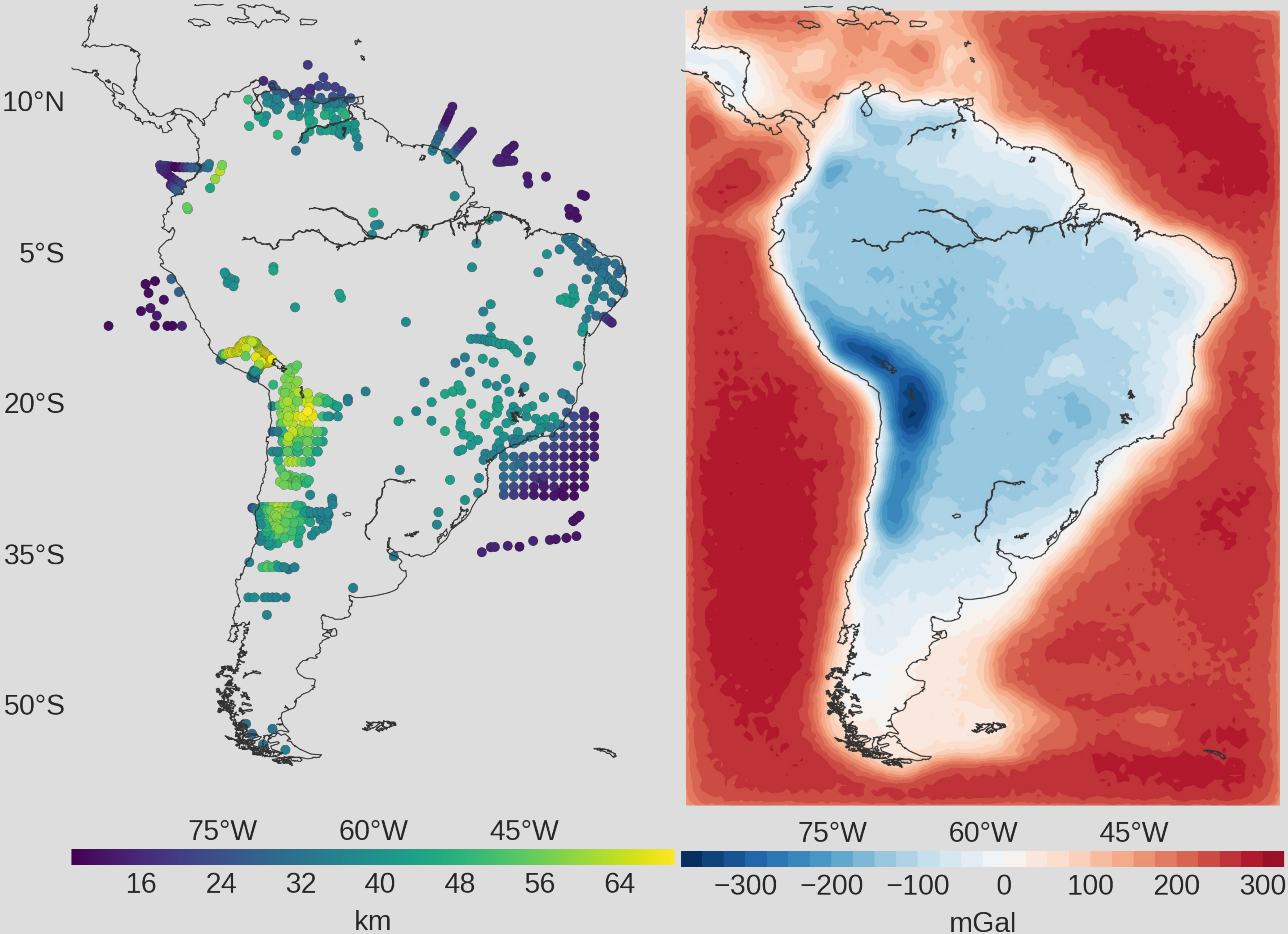
Moho CRUST1.0

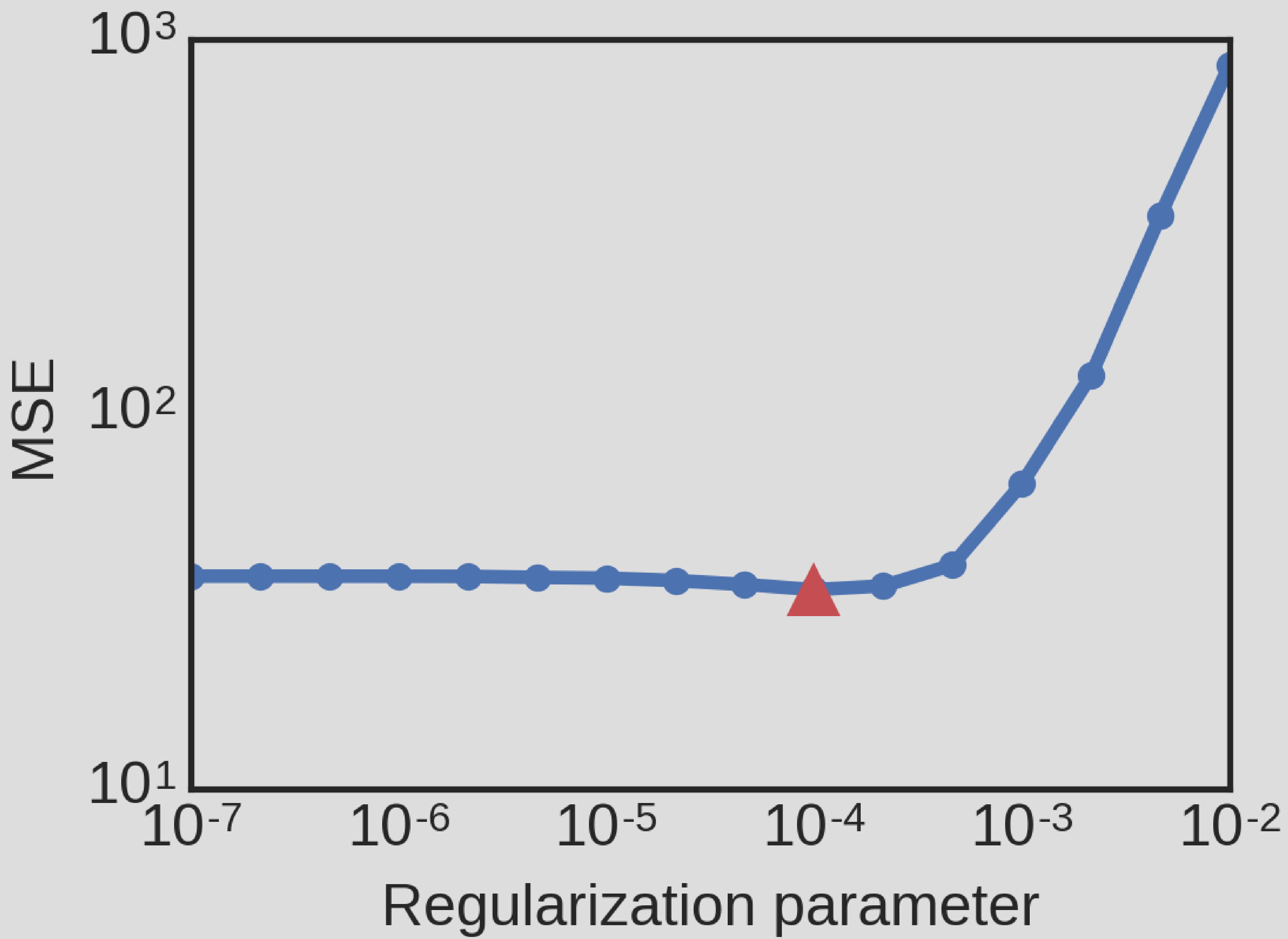
$$\Delta\rho = 350\ kg.m^{-3}$$

$$z_{ref} = 30\ km$$

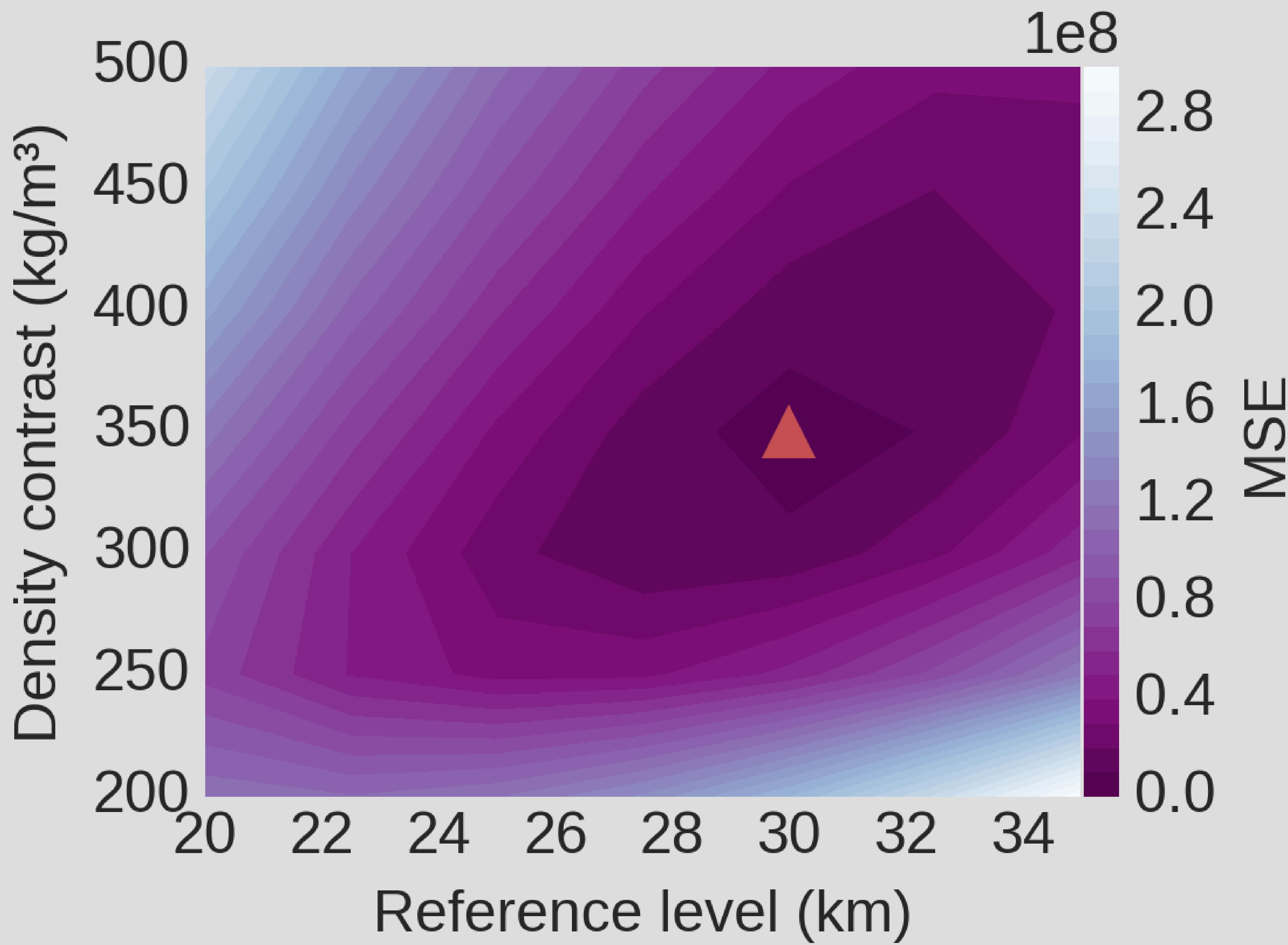


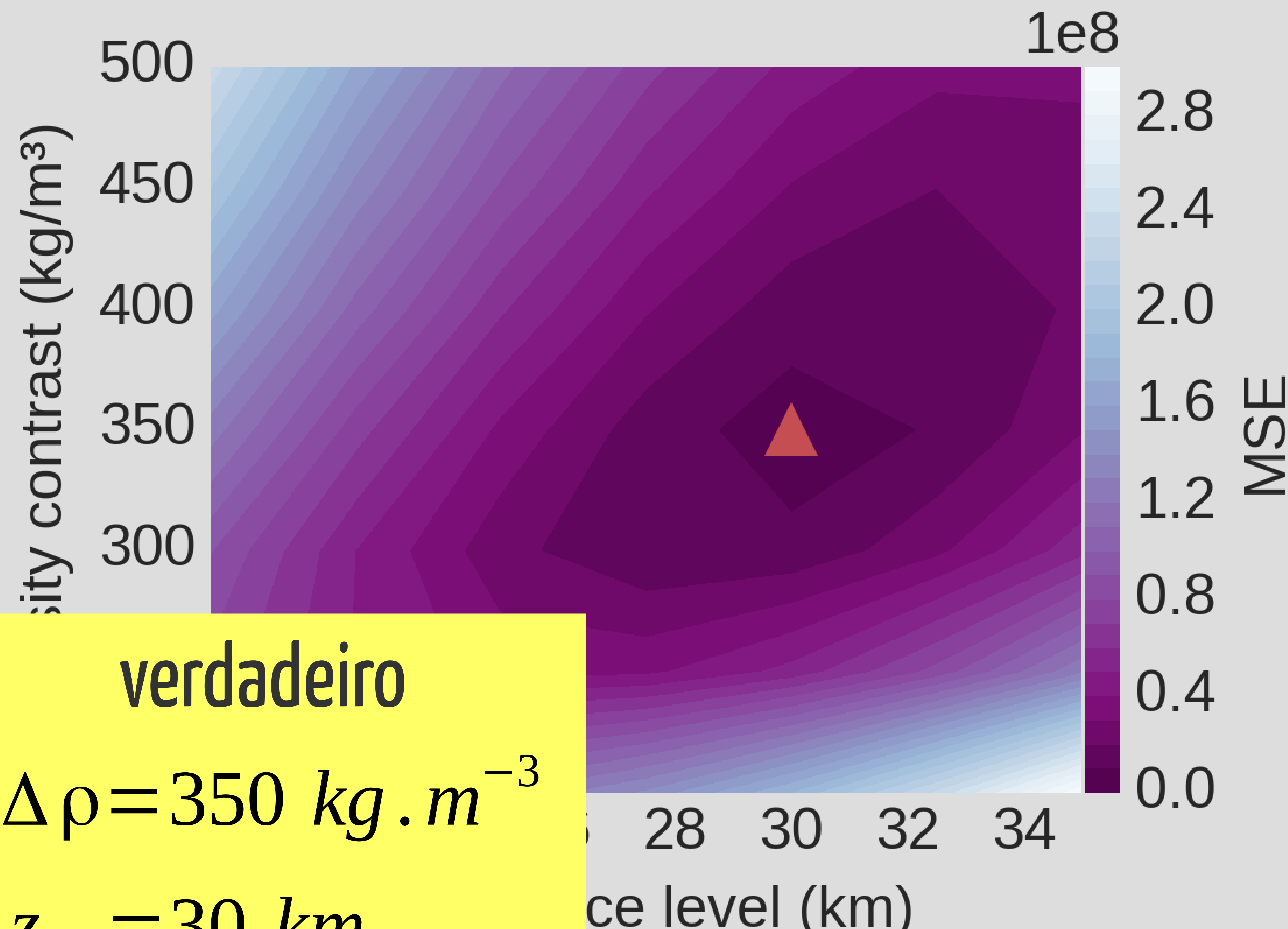
10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
16
24
32
40
48
56
64
km
75°W
60°W
45°W
−300
−200
−100
0
100
200
300
mGal


10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
16
24
32
40
48
56
64
km
75°W
60°W
45°W
−300
−200
−100
0
100
200
300
mGal



MSE
10^3
10^2
10^1
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
Regularization parameter

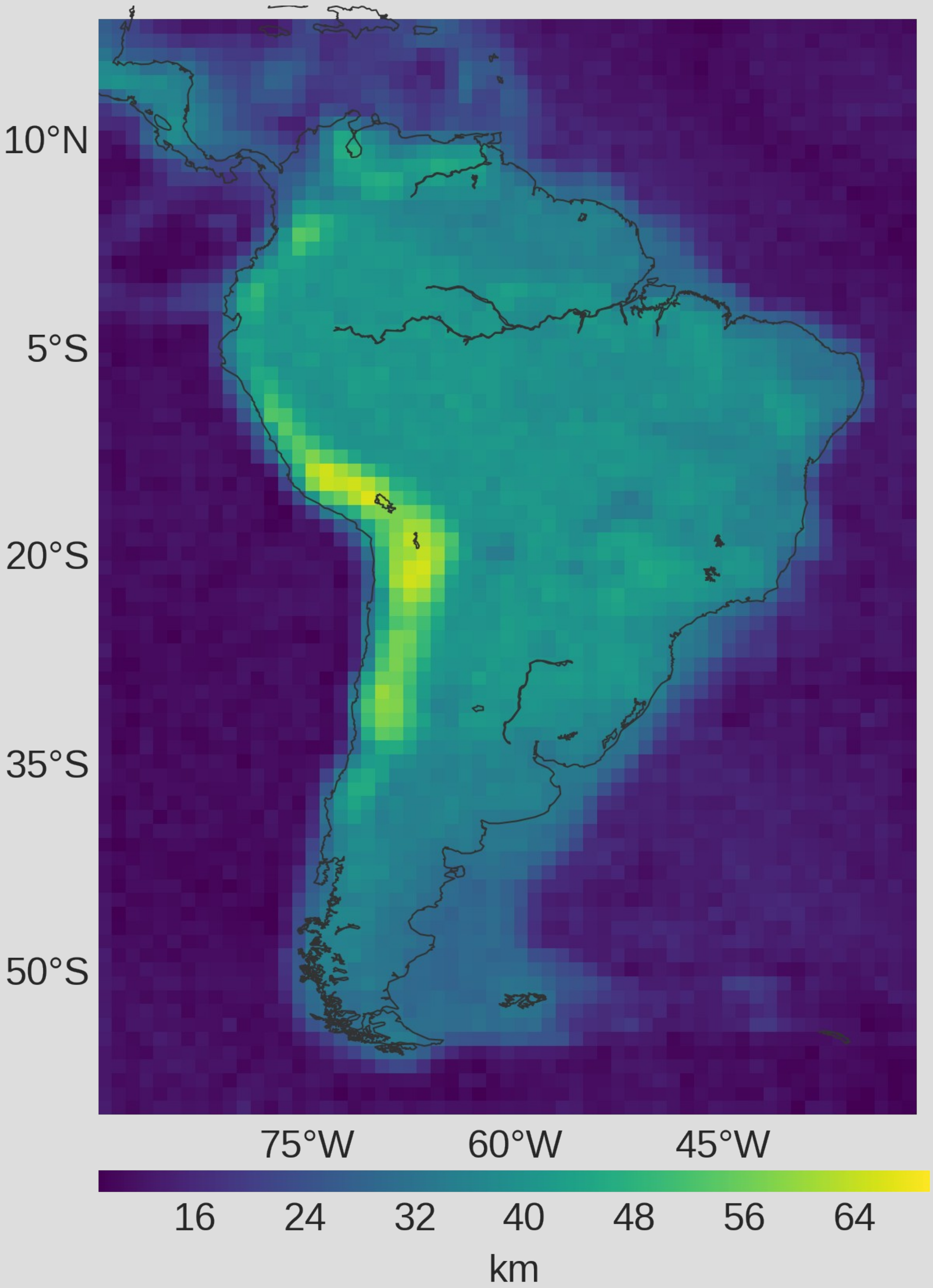

1e8
500
450
400
350
300
250
200
20
22
24
26
28
30
32
34
Density contrast (kg/m³)
Reference level (km)
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0.0
MSE



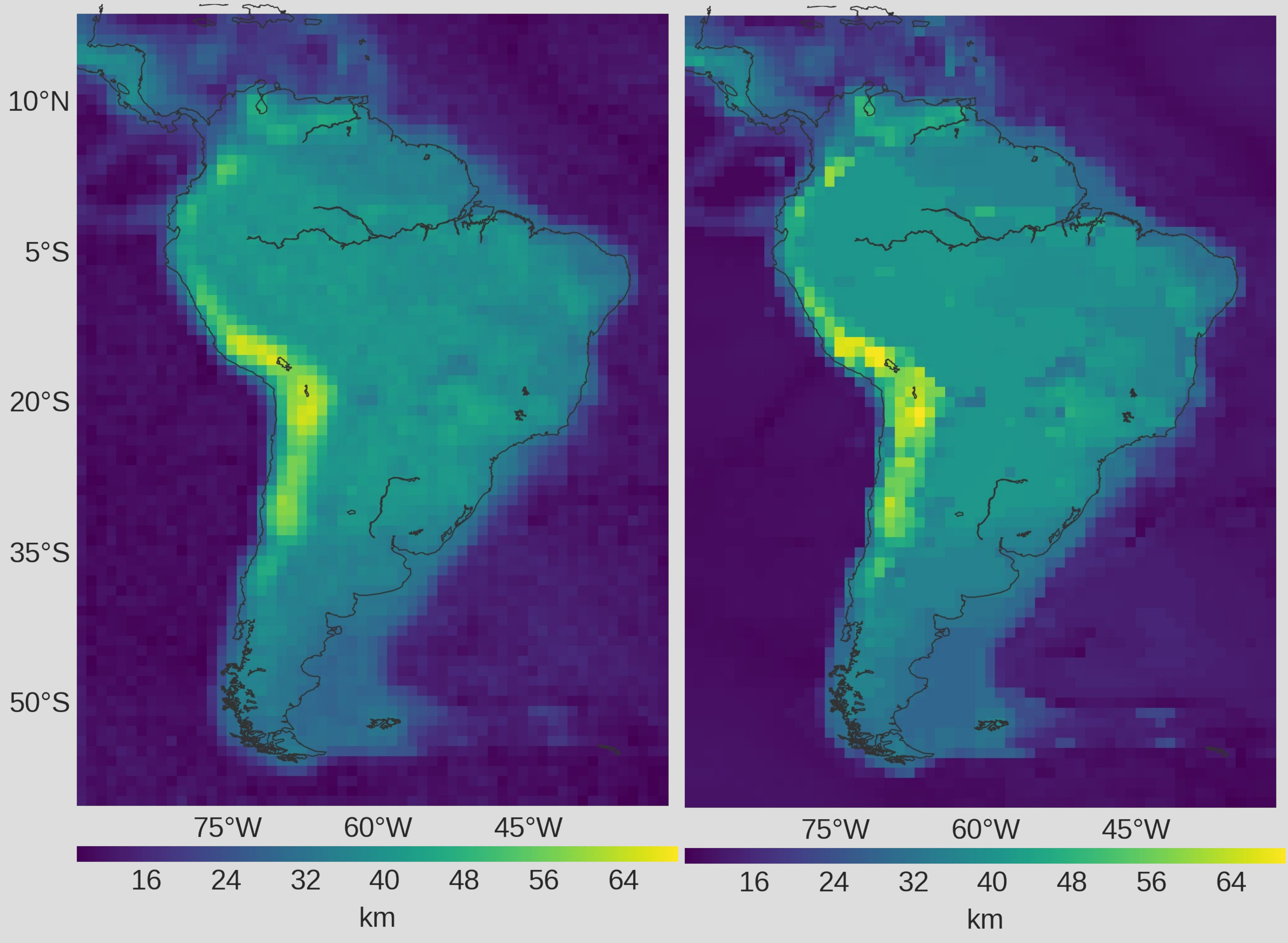
verdadeiro

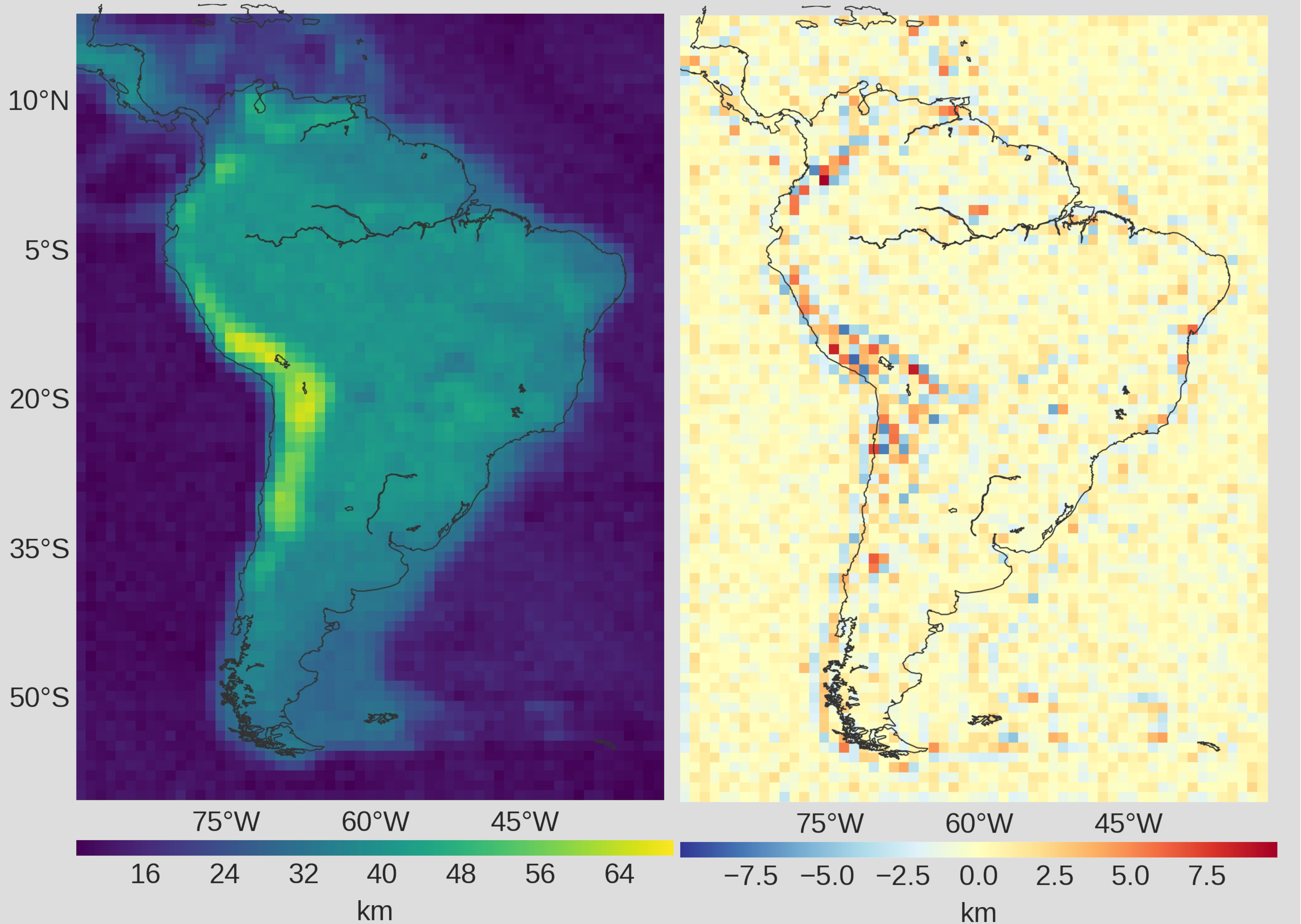
$$\Delta\rho = 350\ kg.m^{-3}$$

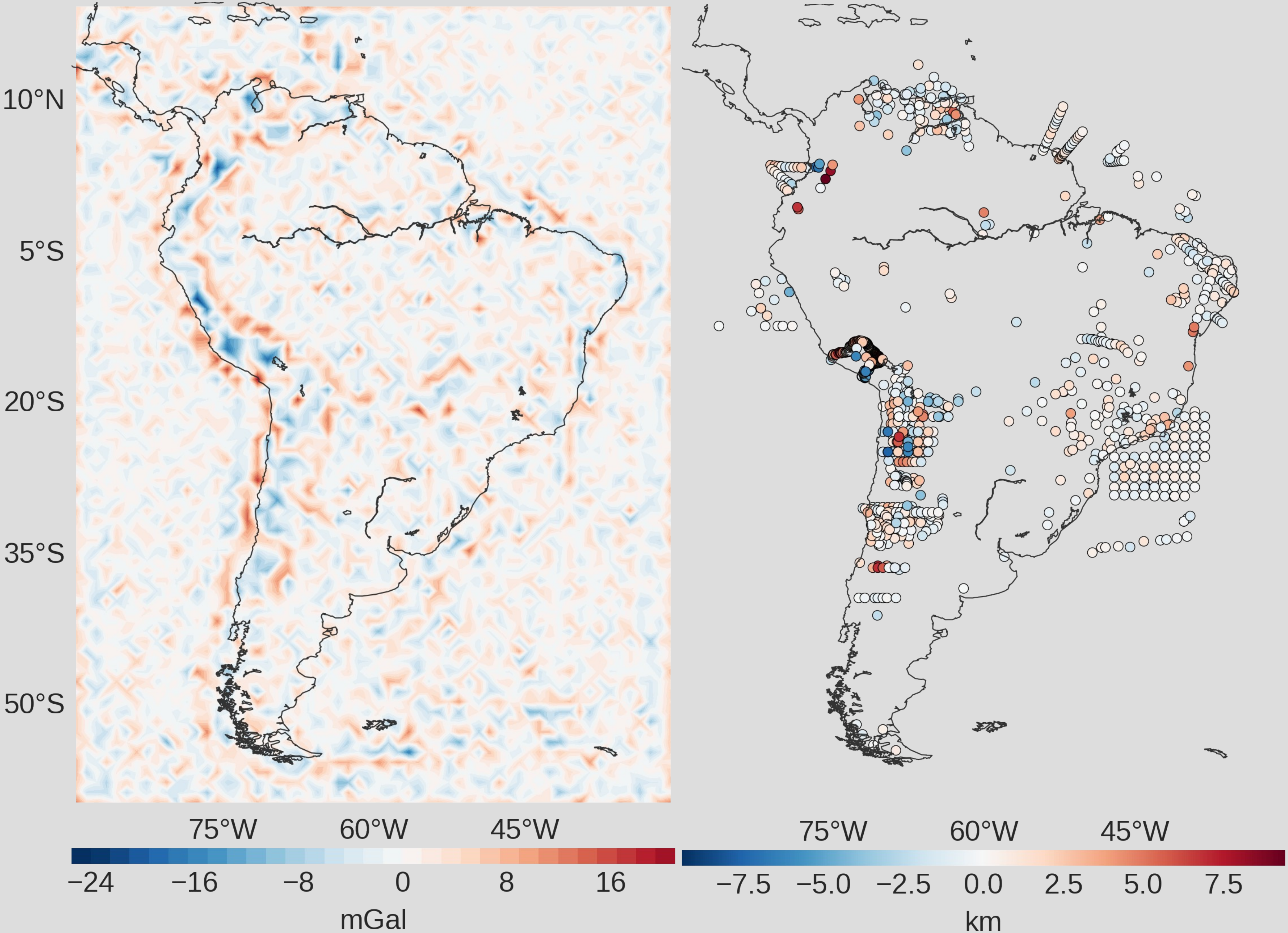
$$z_{ref} = 30\ km$$



10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
16
24
32
40
48
56
64
km

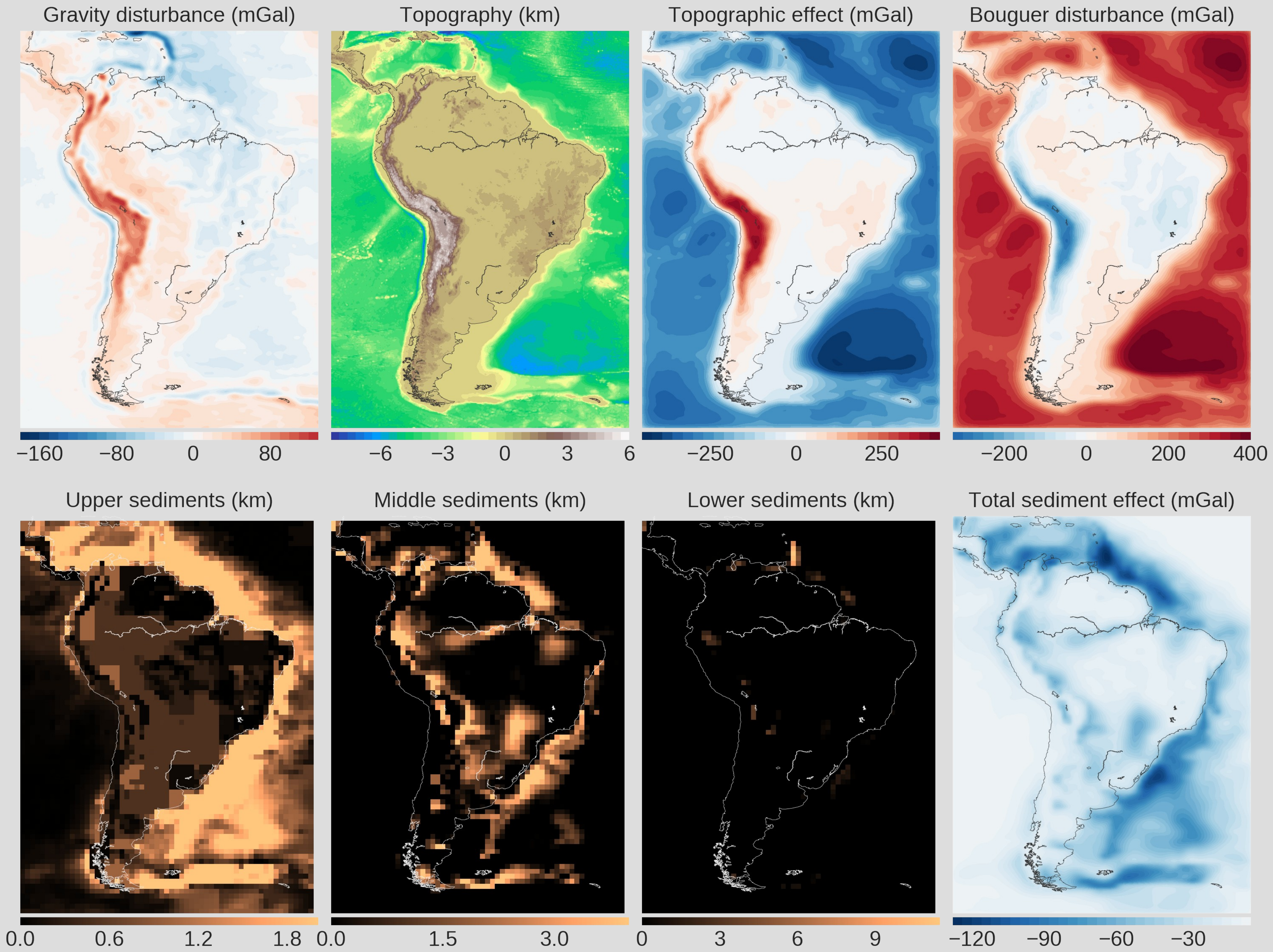

10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
16
24
32
40
48
56
64
km
75°W
60°W
45°W
16
24
32
40
48
56
64
km


10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
16
24
32
40
48
56
64
km
75°W
60°W
45°W
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
km

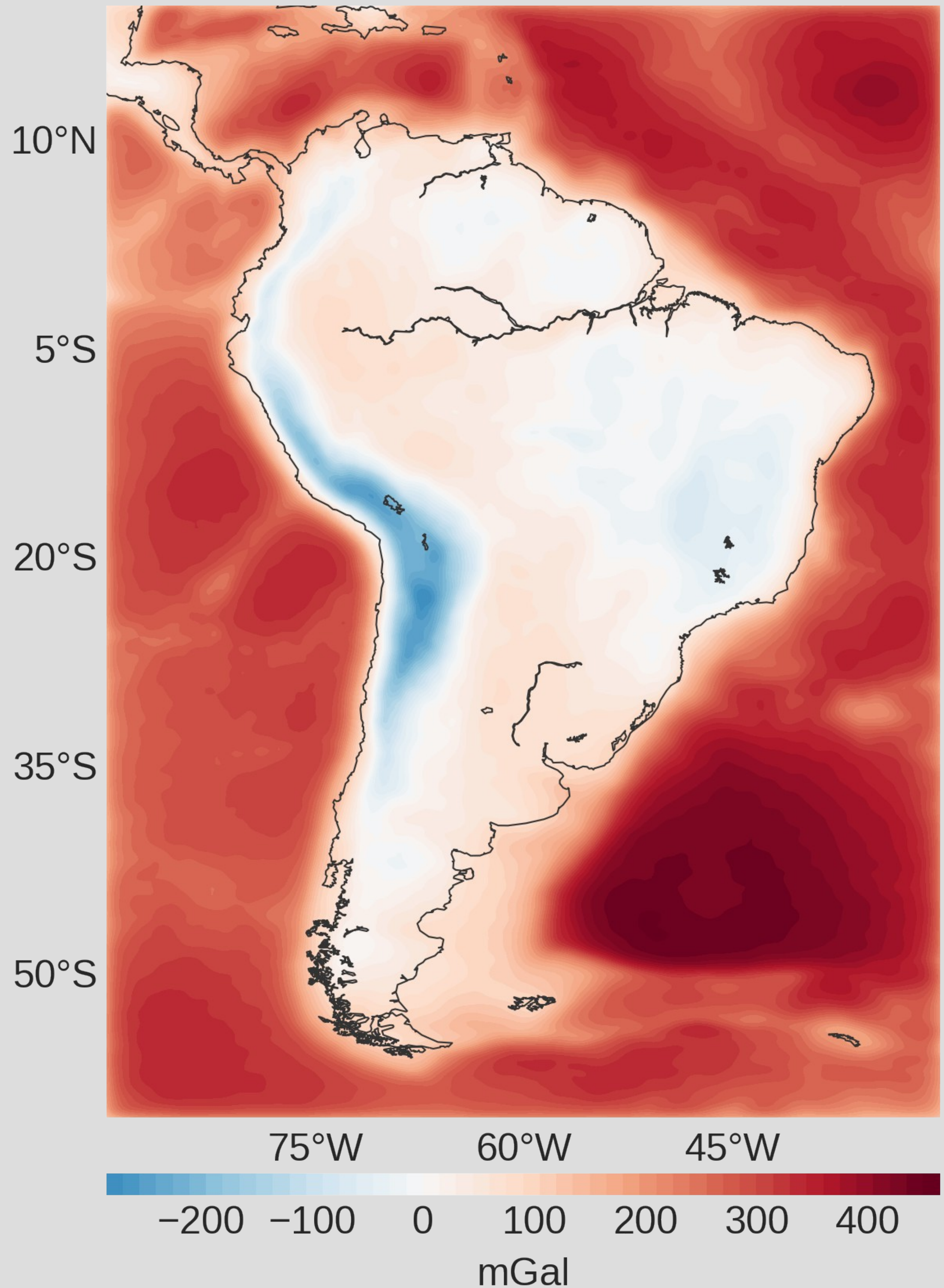


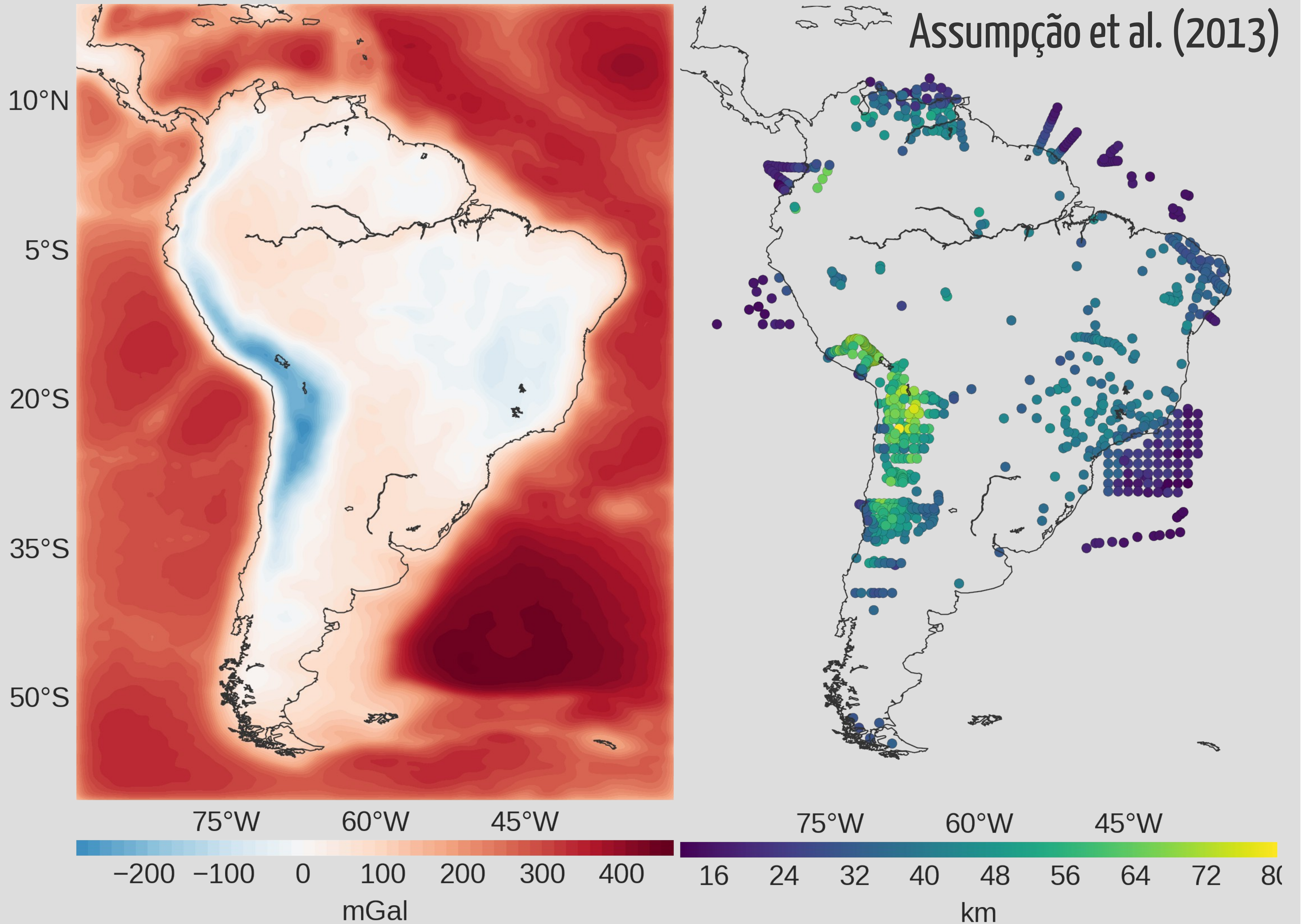
10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−24
−16
−8
0
8
16
mGal
75°W
60°W
45°W
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
km

América do Sul

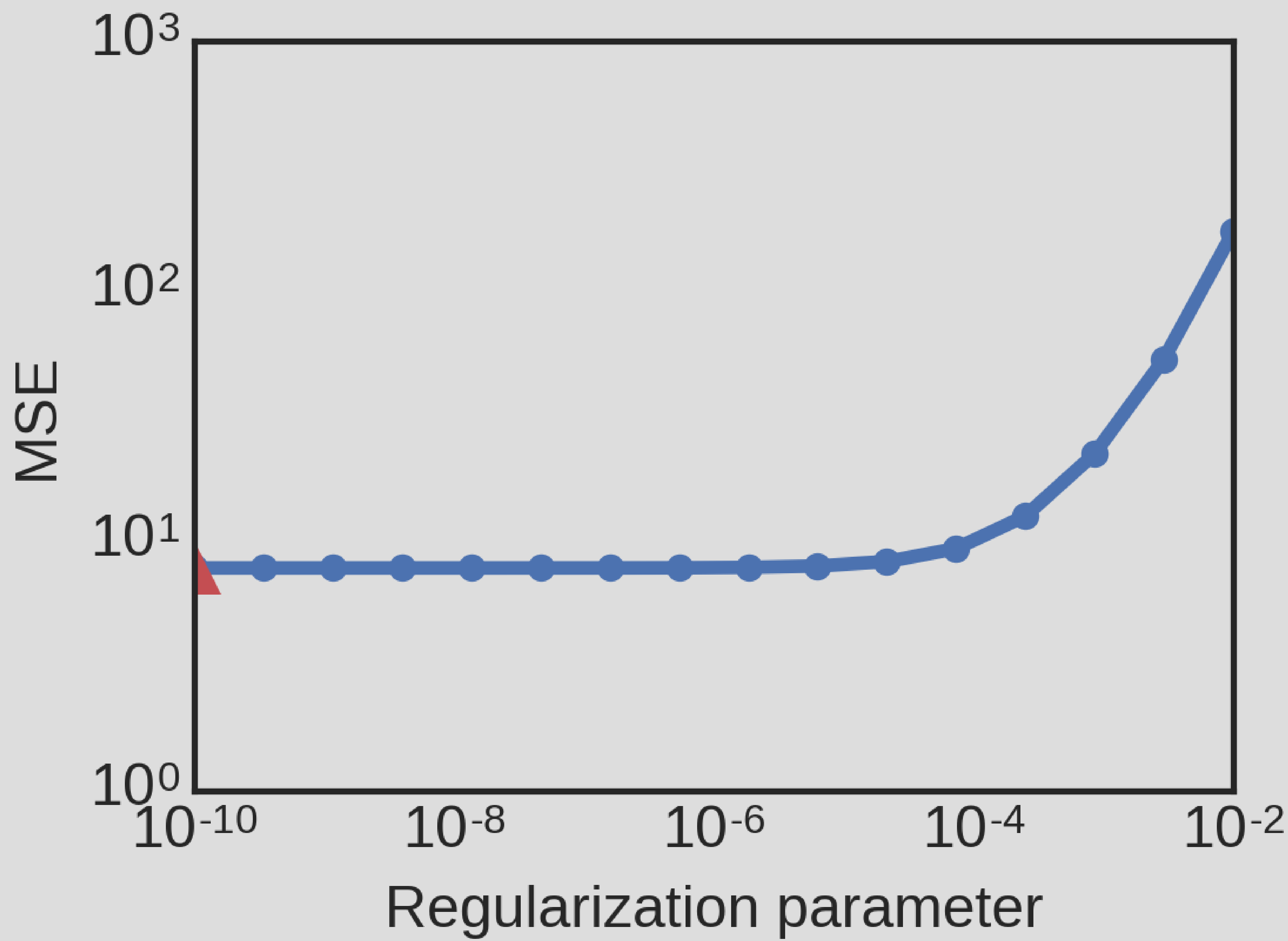


Gravity disturbance (mGal)
Topography (km)
Topographic effect (mGal)
Bouguer disturbance (mGal)
−160 −80 0 80
−6 −3 0 3 6
−250 0 250
−200 0 200 400
Upper sediments (km)
Middle sediments (km)
Lower sediments (km)
Total sediment effect (mGal)
0.0 0.6 1.2 1.8
0.0 1.5 3.0
0 3 6 9
−120 −90 −60 −30

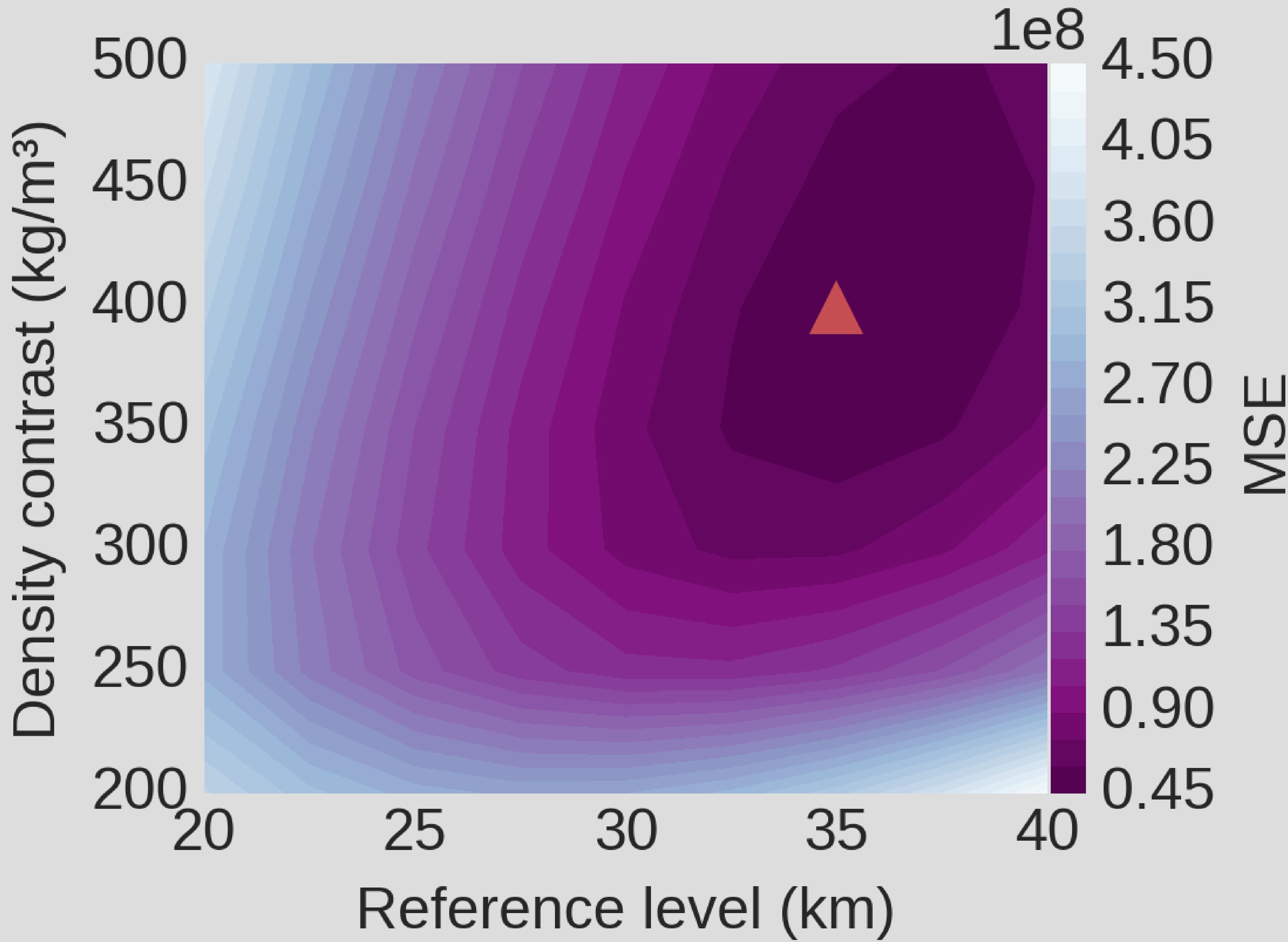

10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−200
−100
0
100
200
300
400
mGal



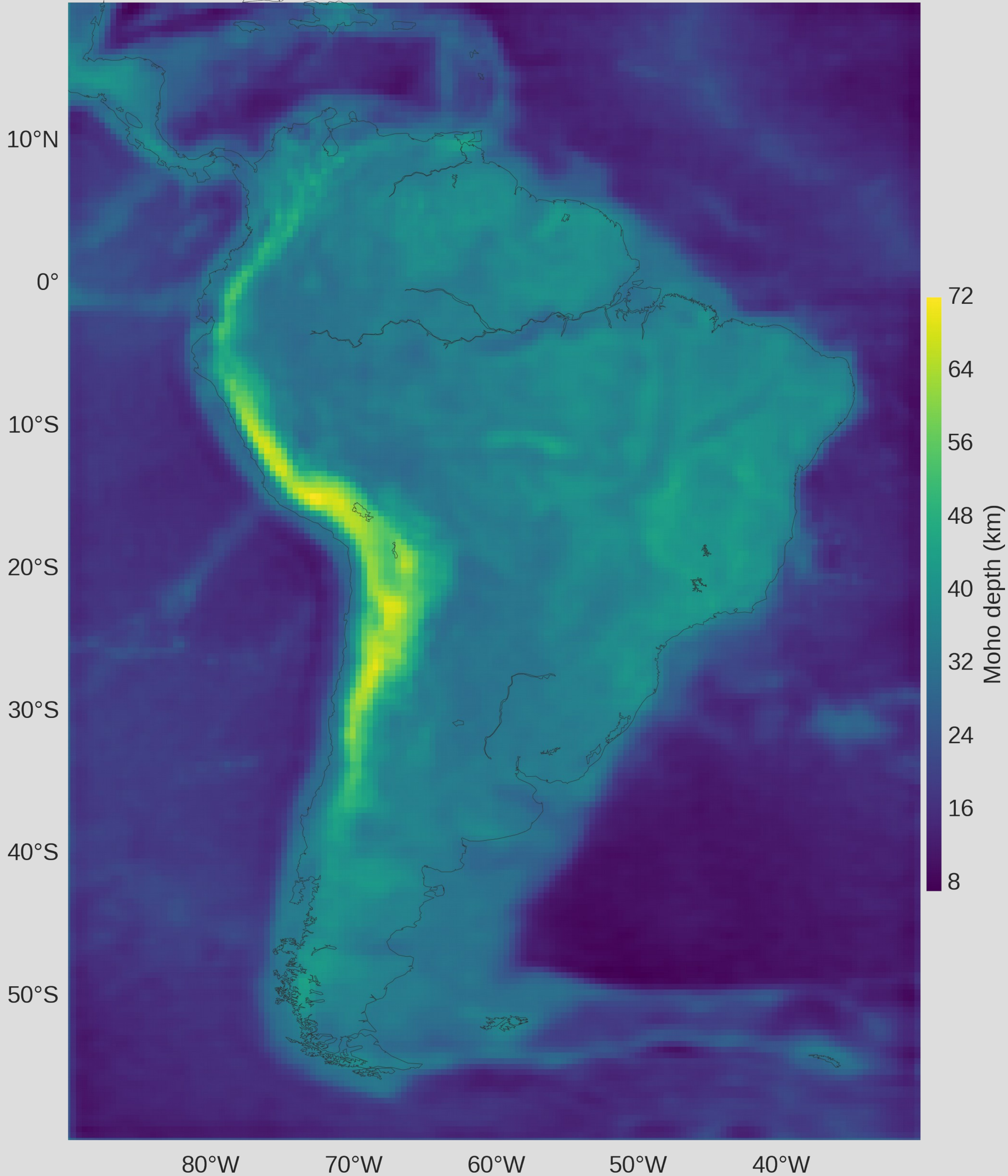
Assumpção et al. (2013)
10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
75°W
60°W
45°W
−200
−100
0
100
200
300
400
mGal
16
24
32
40
48
56
64
72
km



MSE
10^3
10^2
10^1
10^0
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
Regularization parameter

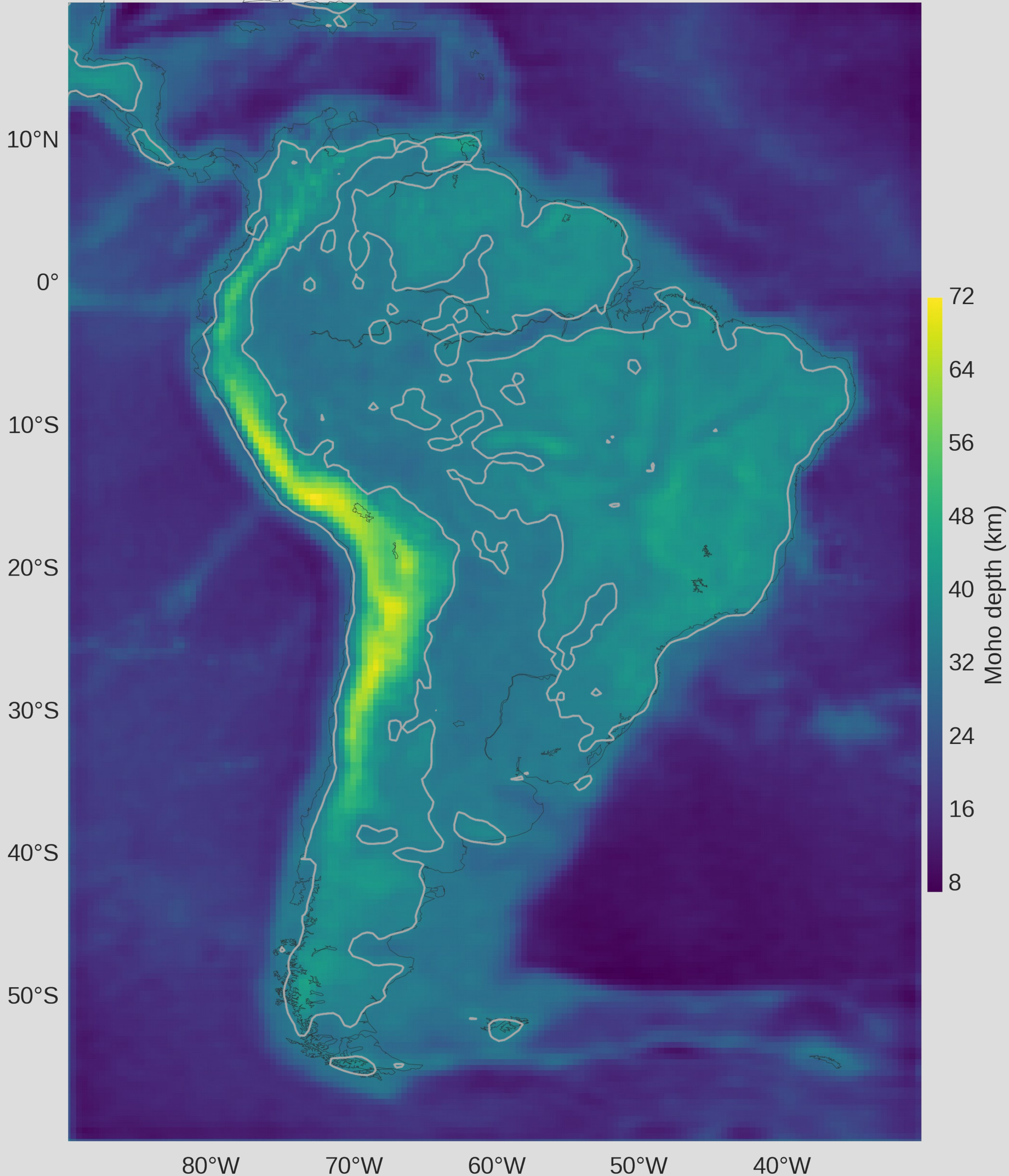


1e8
4.50
4.05
3.60
3.15
2.70
2.25
1.80
1.35
0.90
0.45
MSE
500
450
400
350
300
250
200
Density contrast (kg/m³)
20
25
30
35
40
Reference level (km)



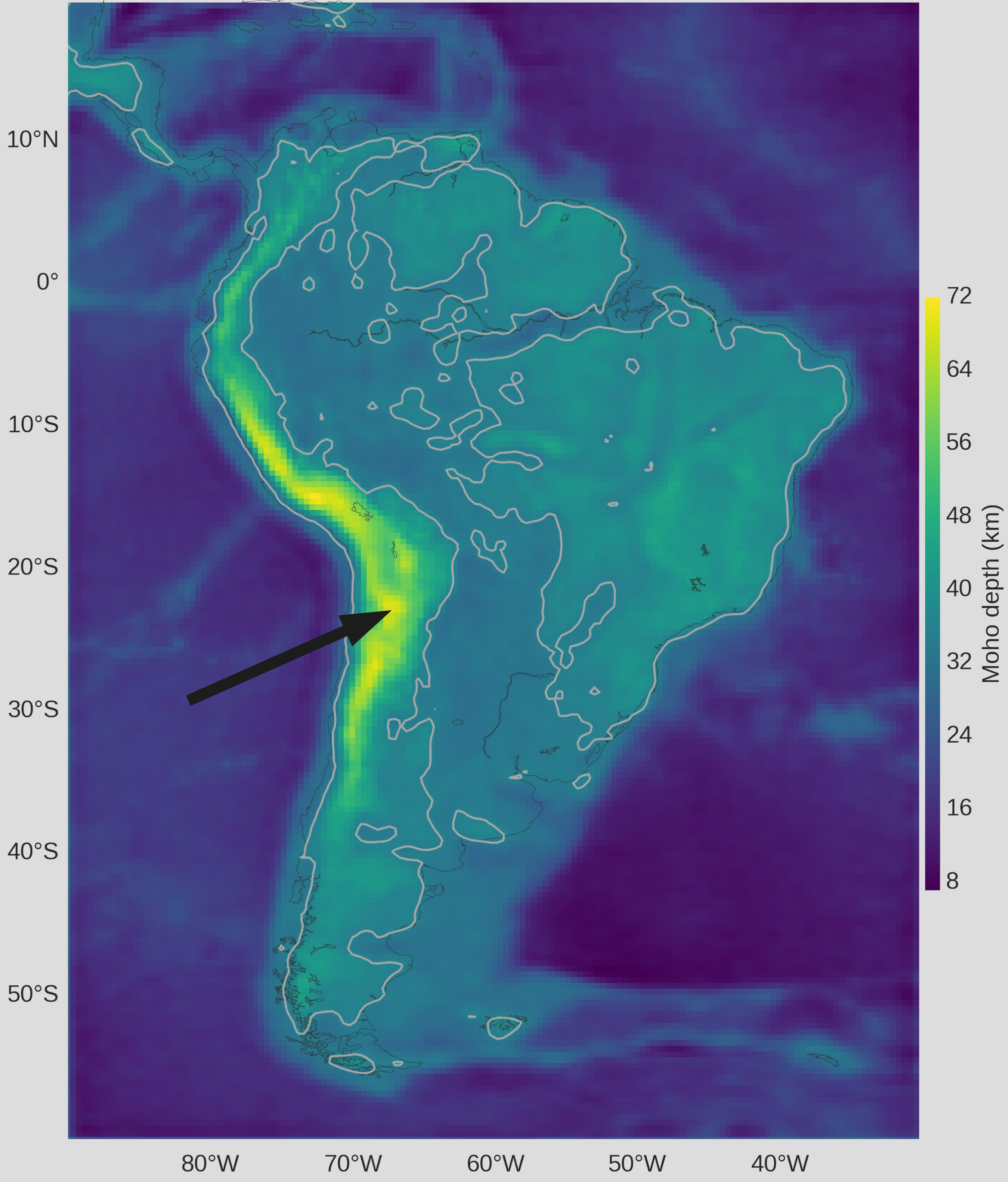
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
72
64
56
48
40
32
24
16
8
Moho depth (km)

35 km

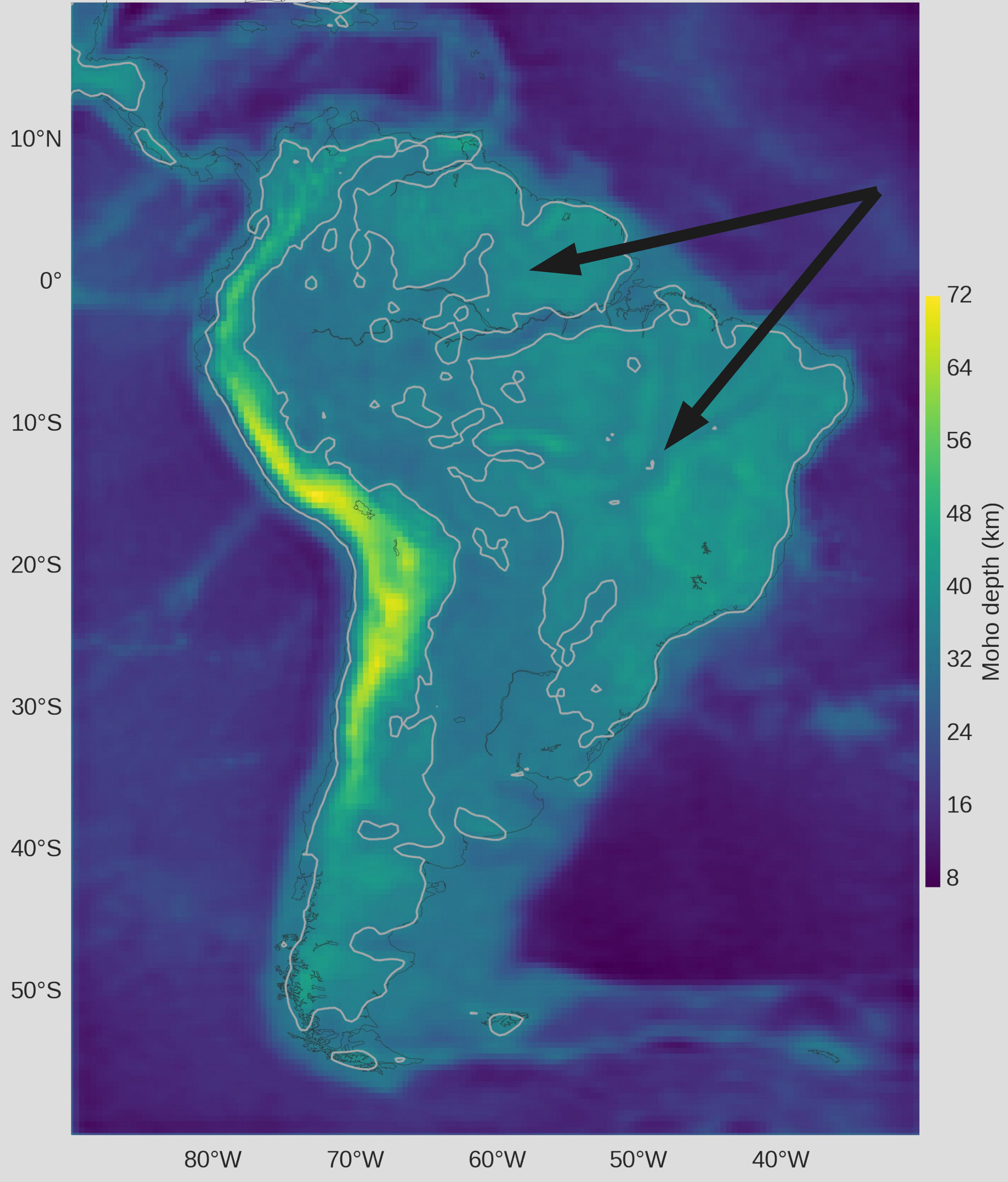


10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
Moho depth (km)
72
64
56
48
40
32
24
16
8

35 km

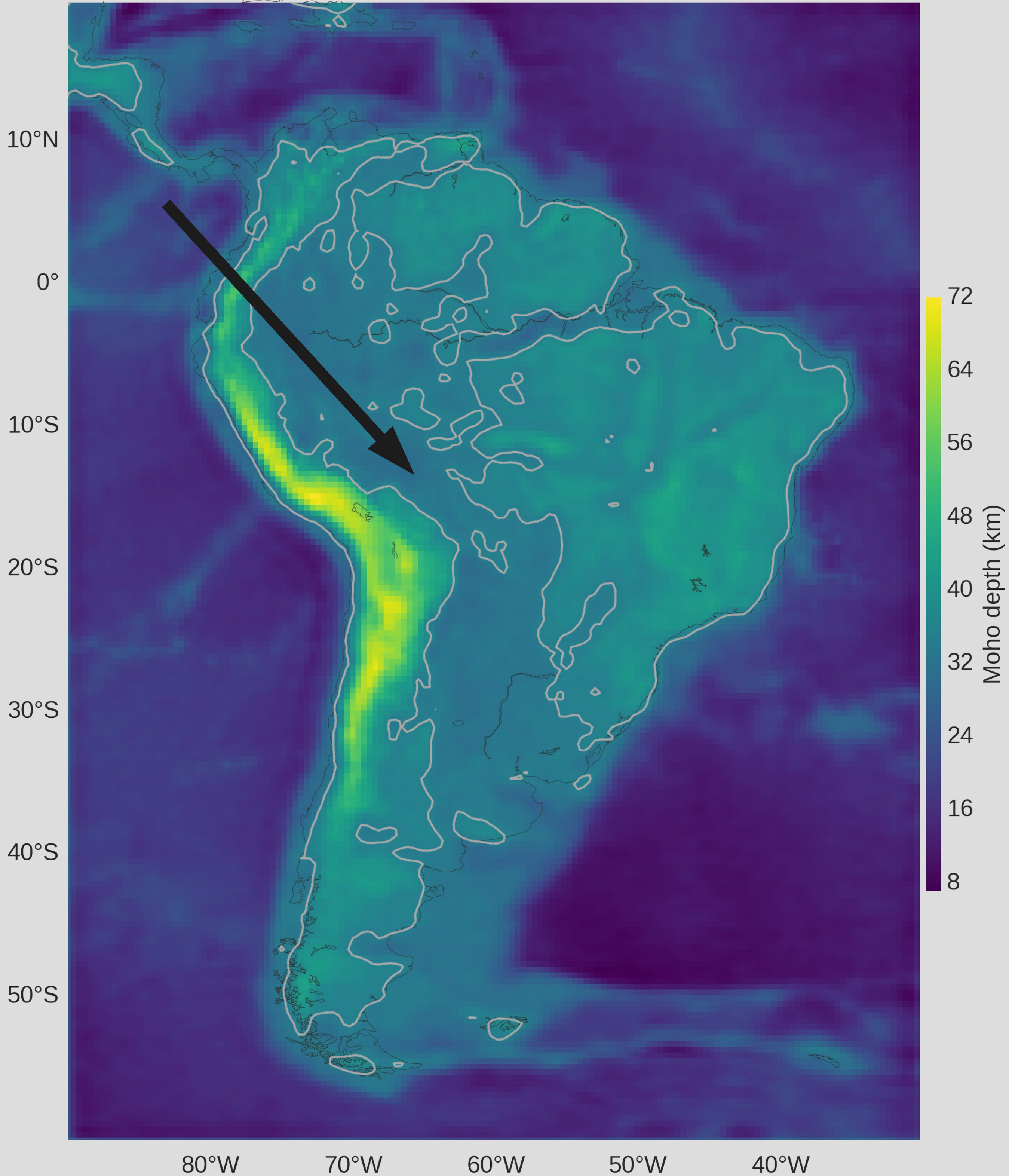

10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
Moho depth (km)
72
64
56
48
40
32
24
16
8

35 km



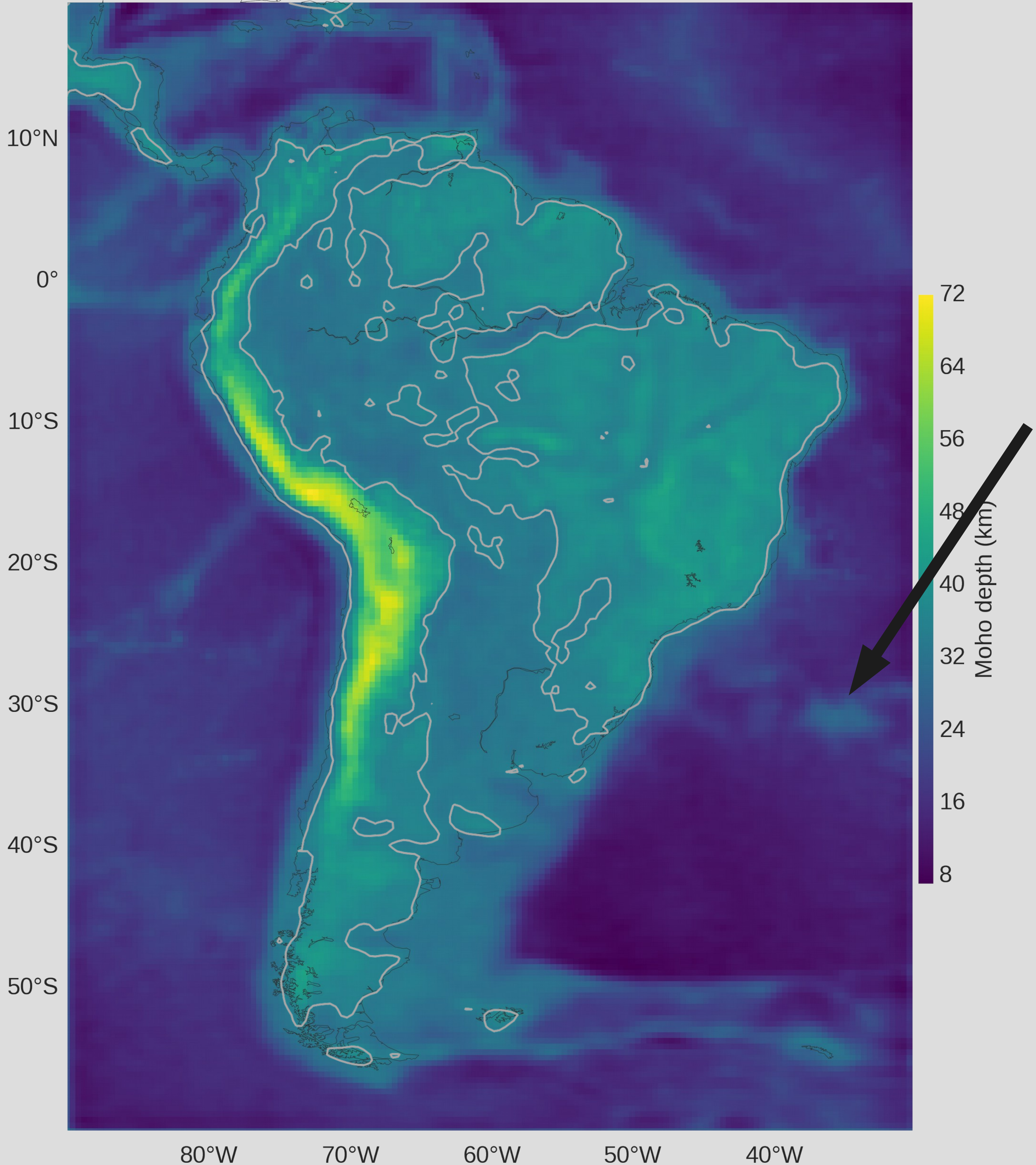
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
Moho depth (km)
72
64
56
48
40
32
24
16
8

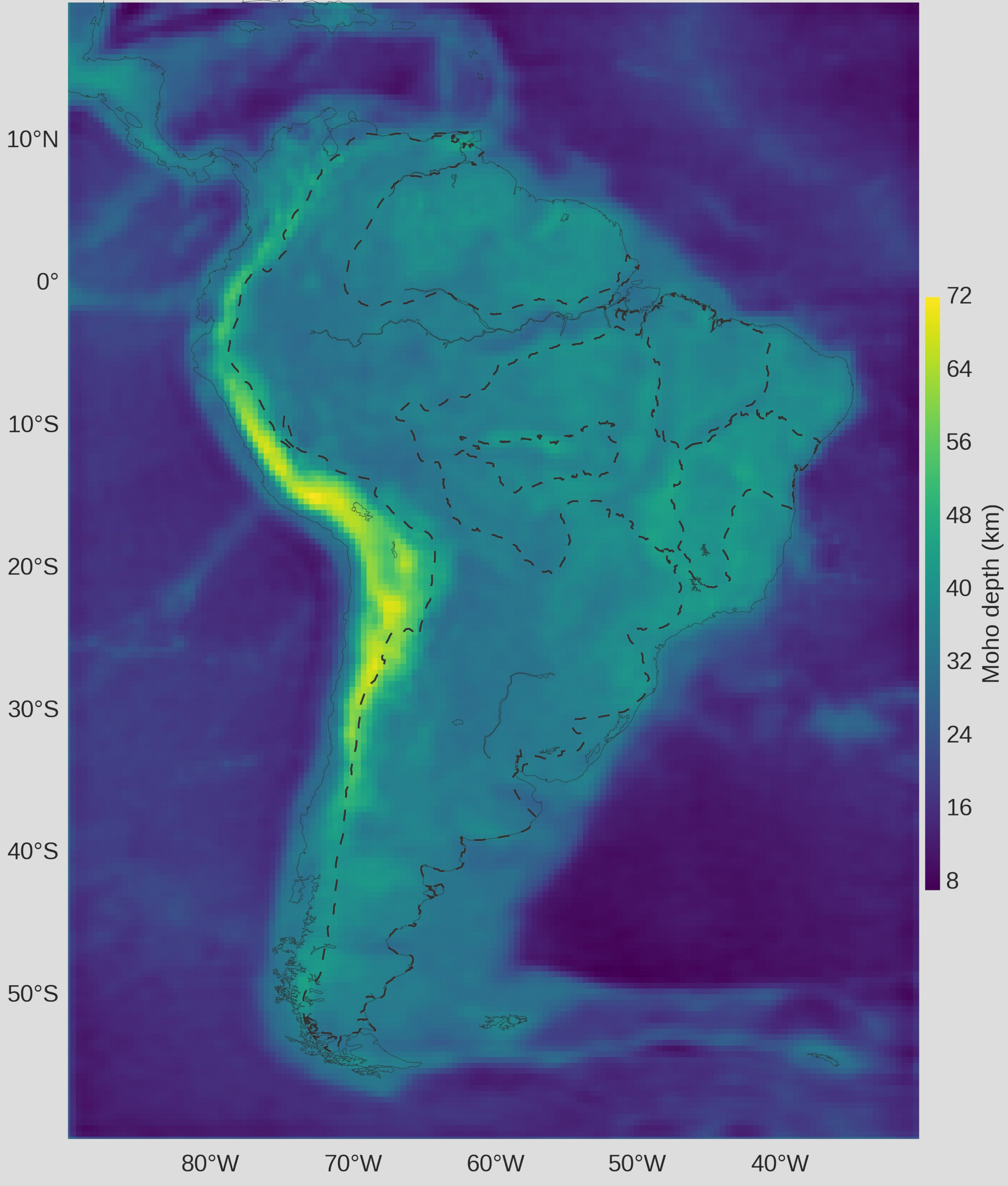
35 km



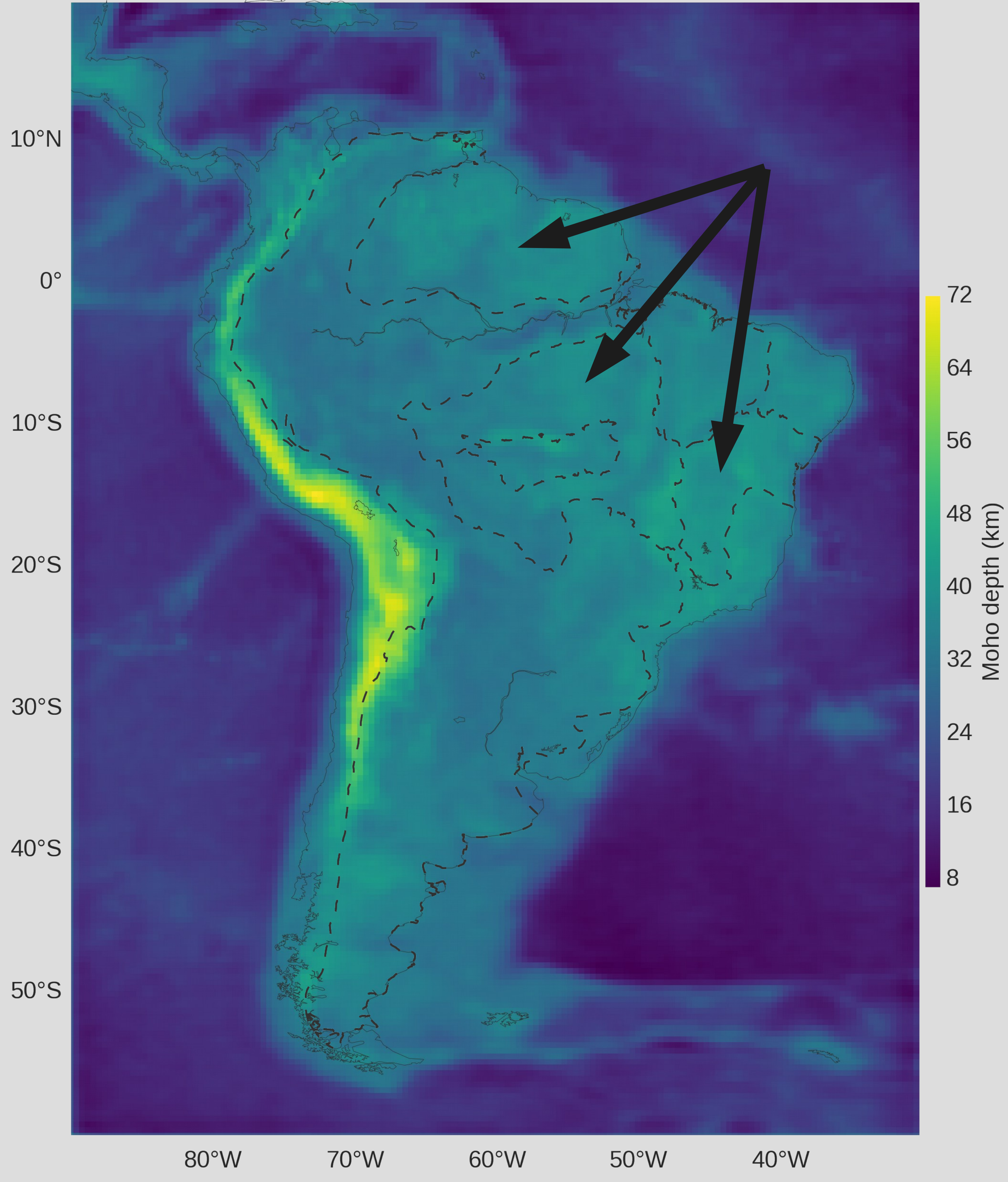
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
72
64
56
48
40
32
24
16
8
Moho depth (km)

35 km

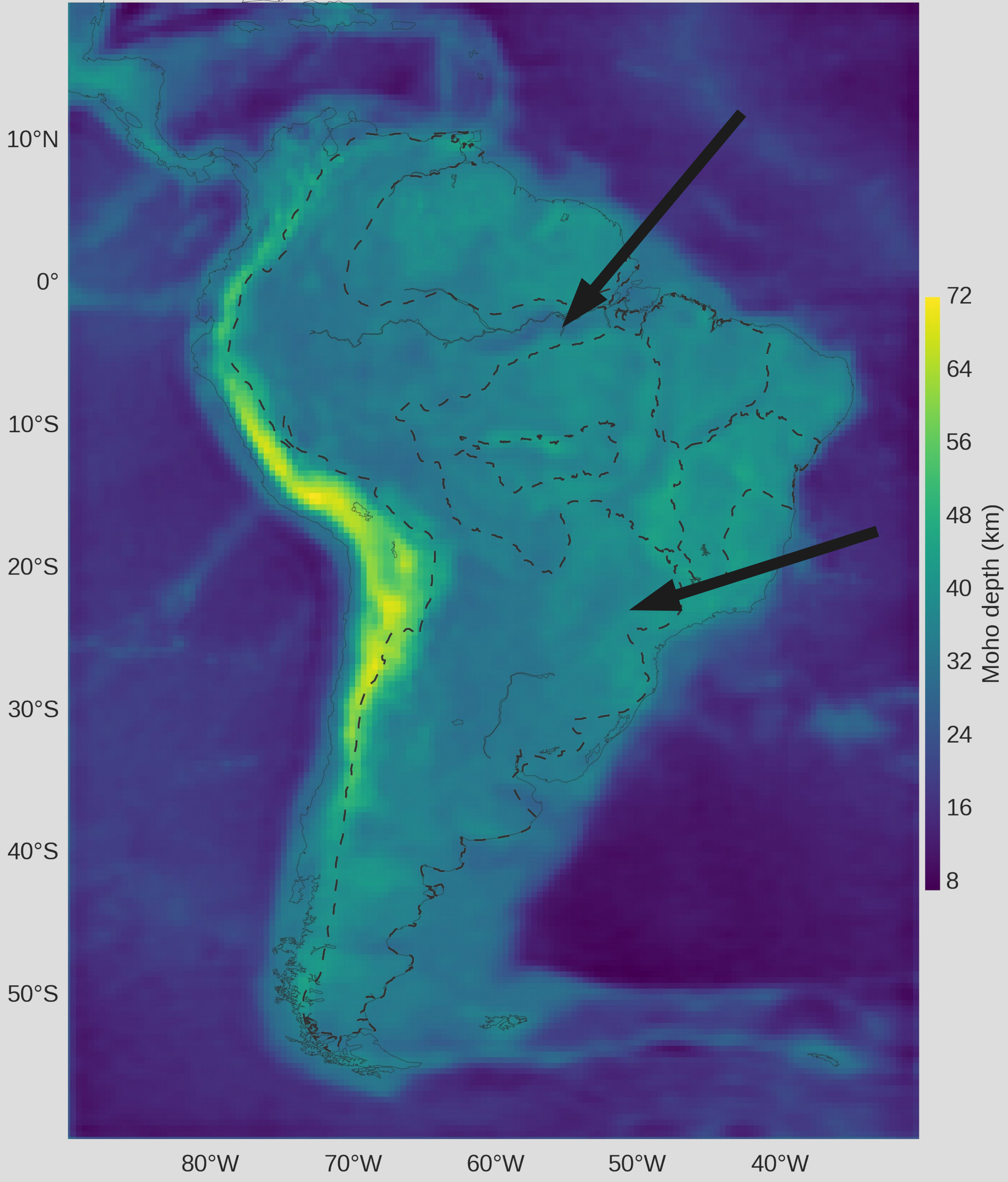

10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
Moho depth (km)
72
64
56
48
40
32
24
16
8



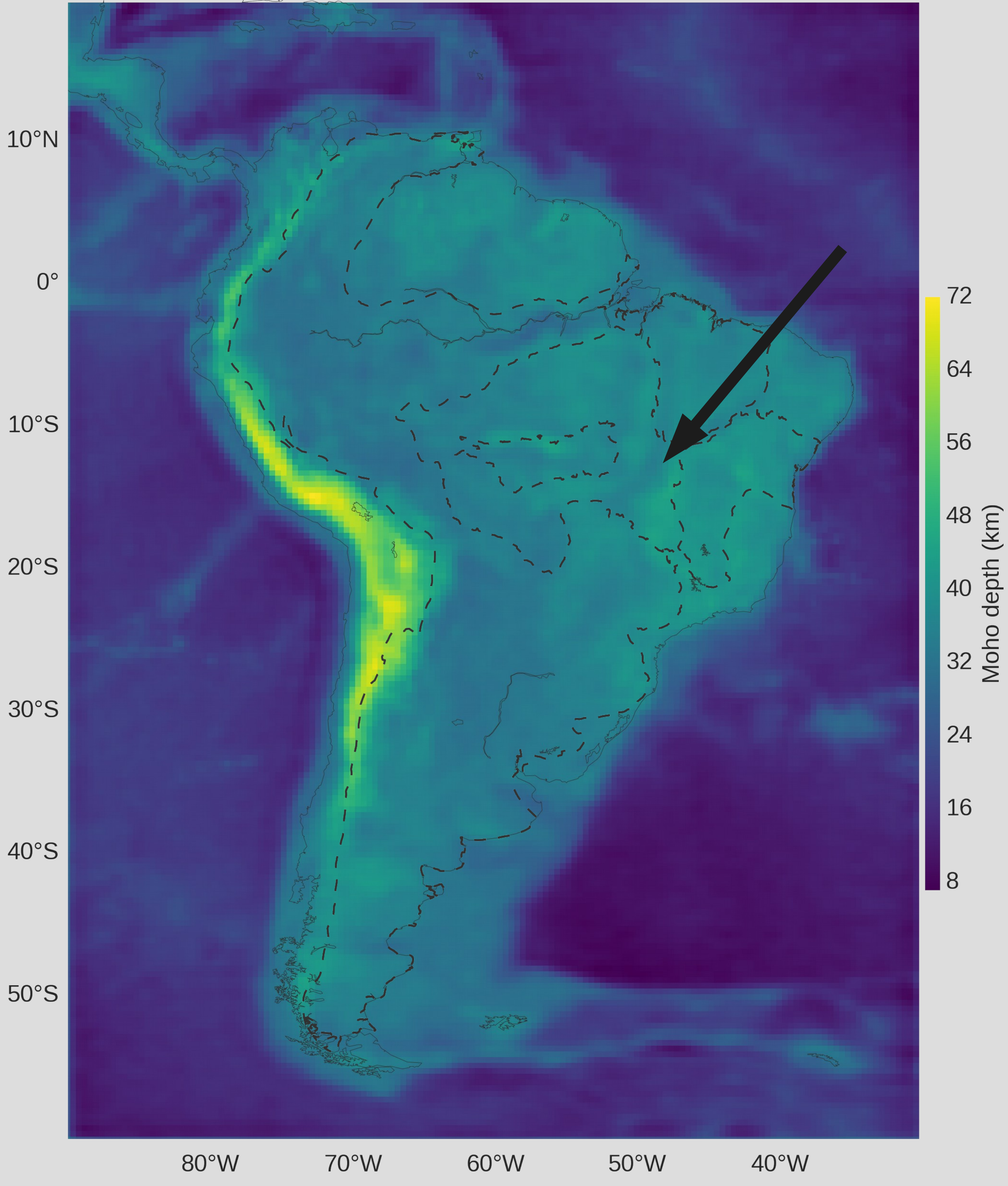
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
72
64
56
48
40
32
24
16
8
Moho depth (km)



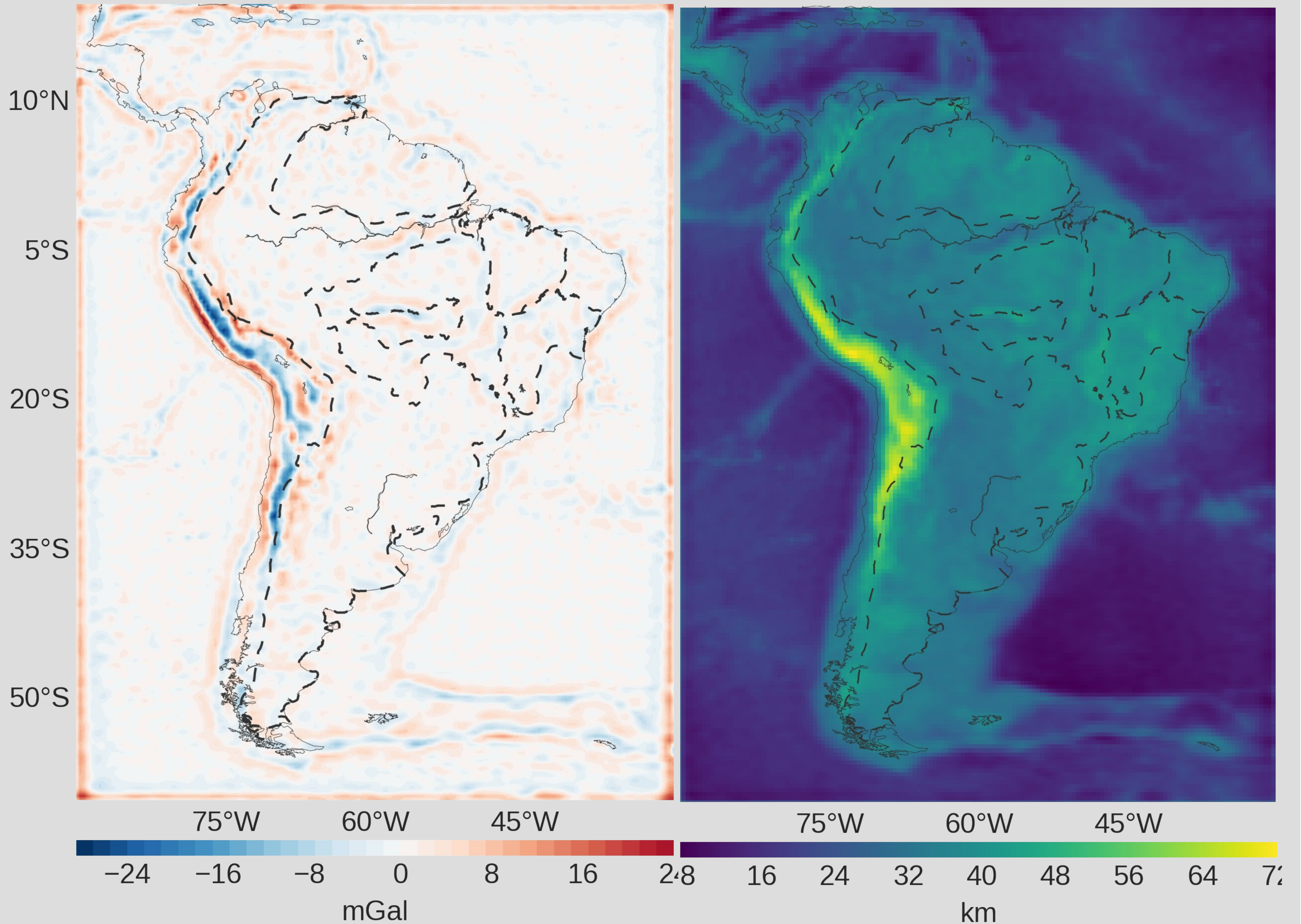
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
72
64
56
48
40
32
24
16
8
Moho depth (km)



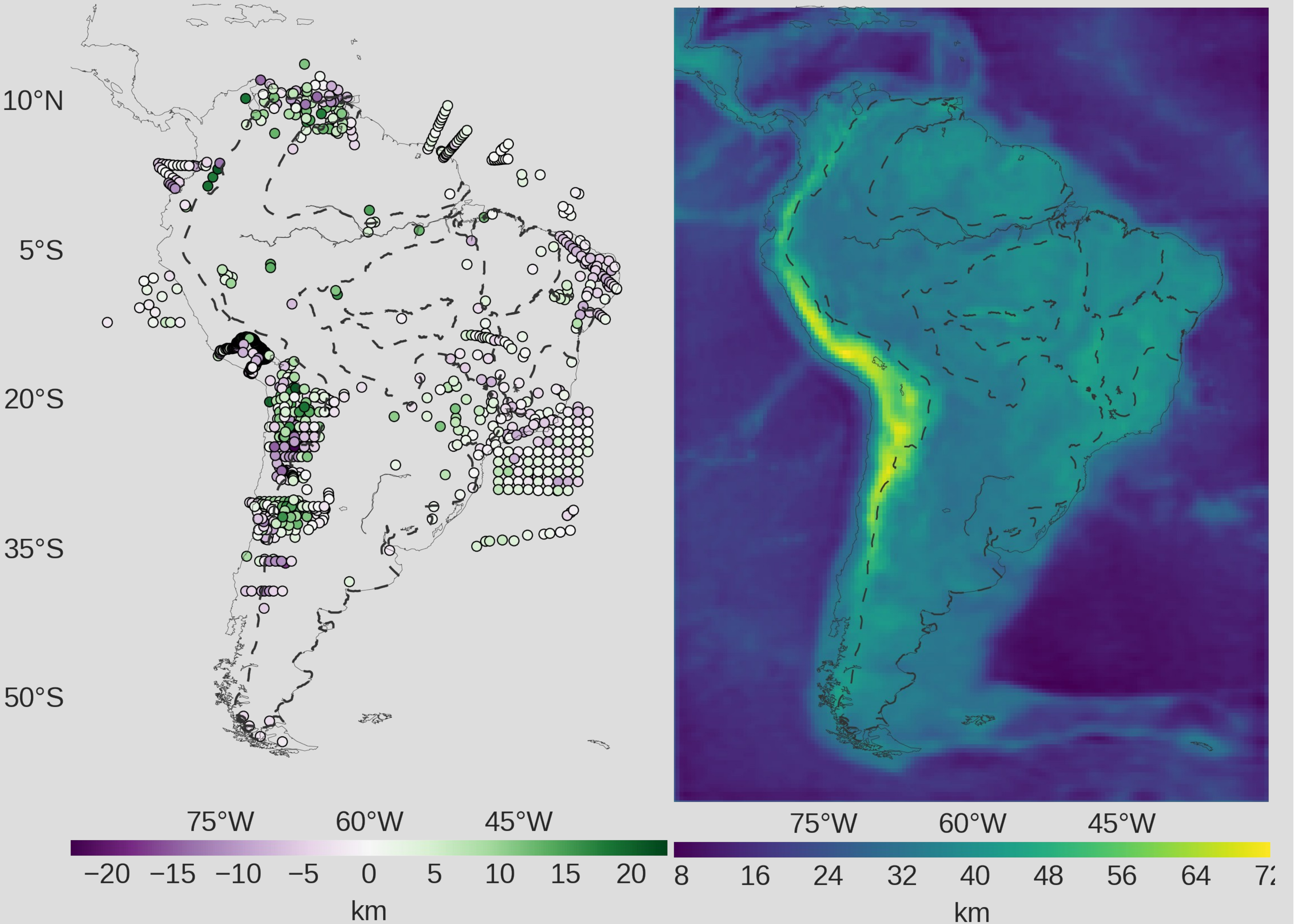
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
72
64
56
48
40
32
24
16
8
Moho depth (km)

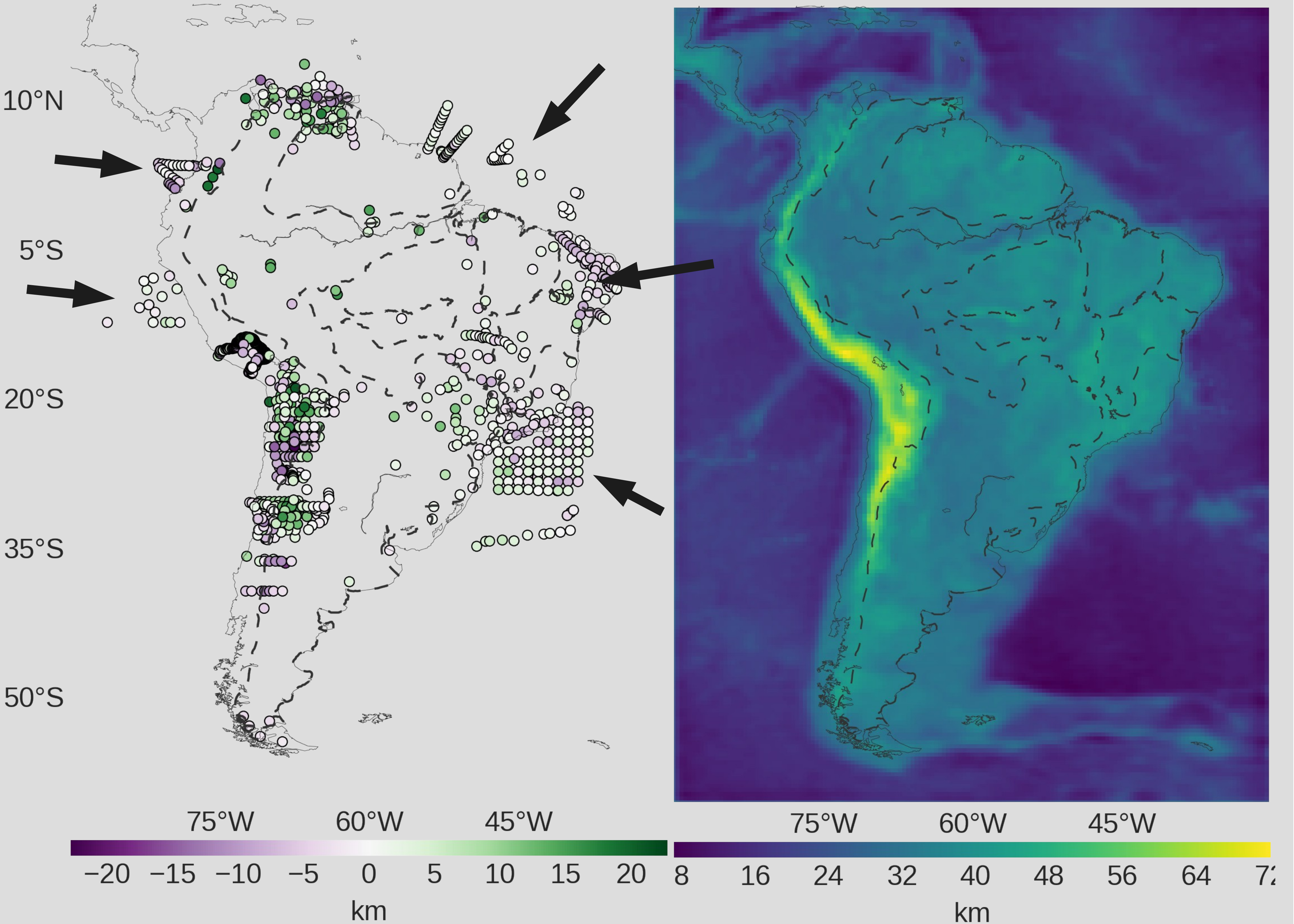


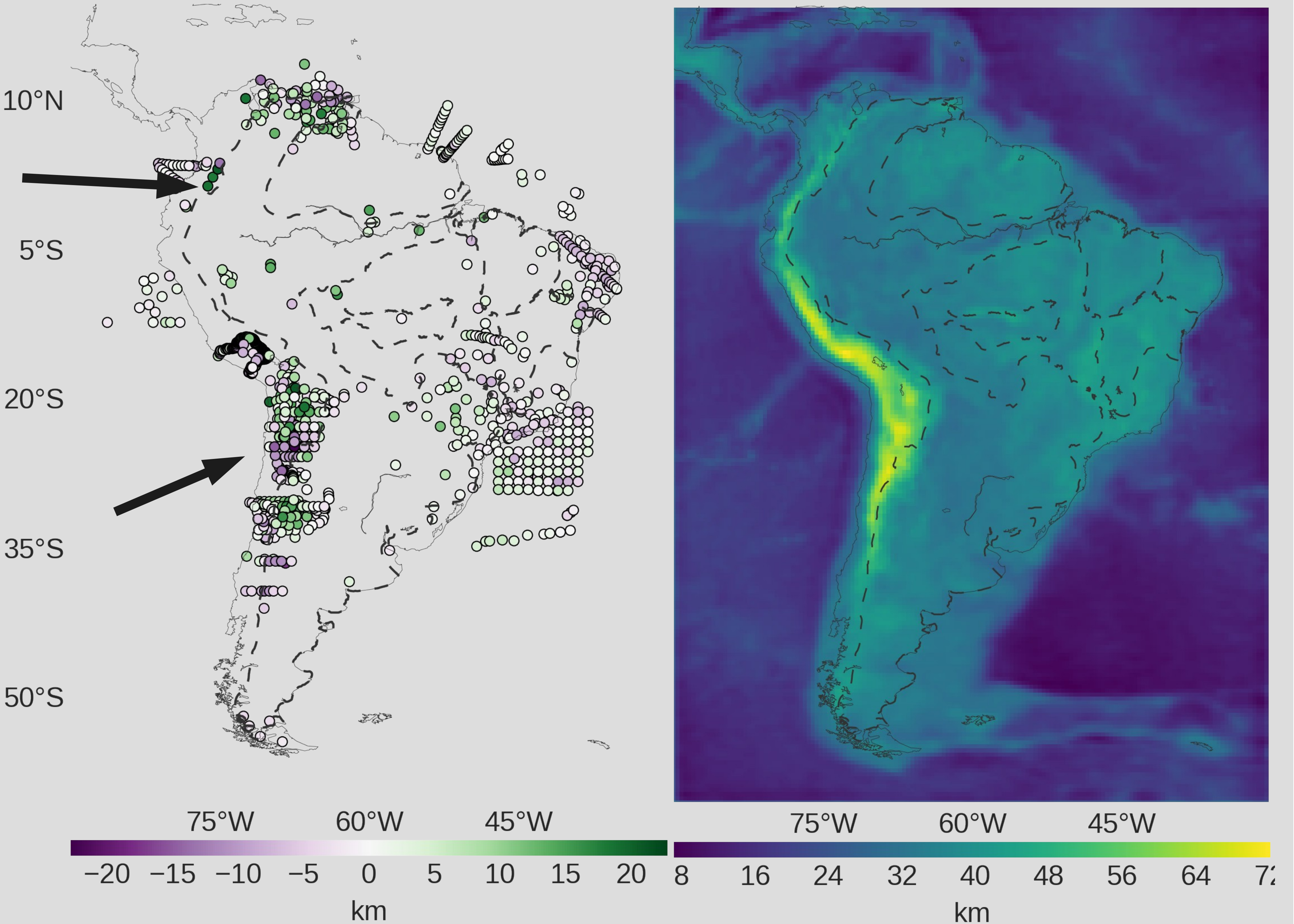
10°N
0°
10°S
20°S
30°S
40°S
50°S
80°W
70°W
60°W
50°W
40°W
72
64
56
48
40
32
24
16
8
Moho depth (km)

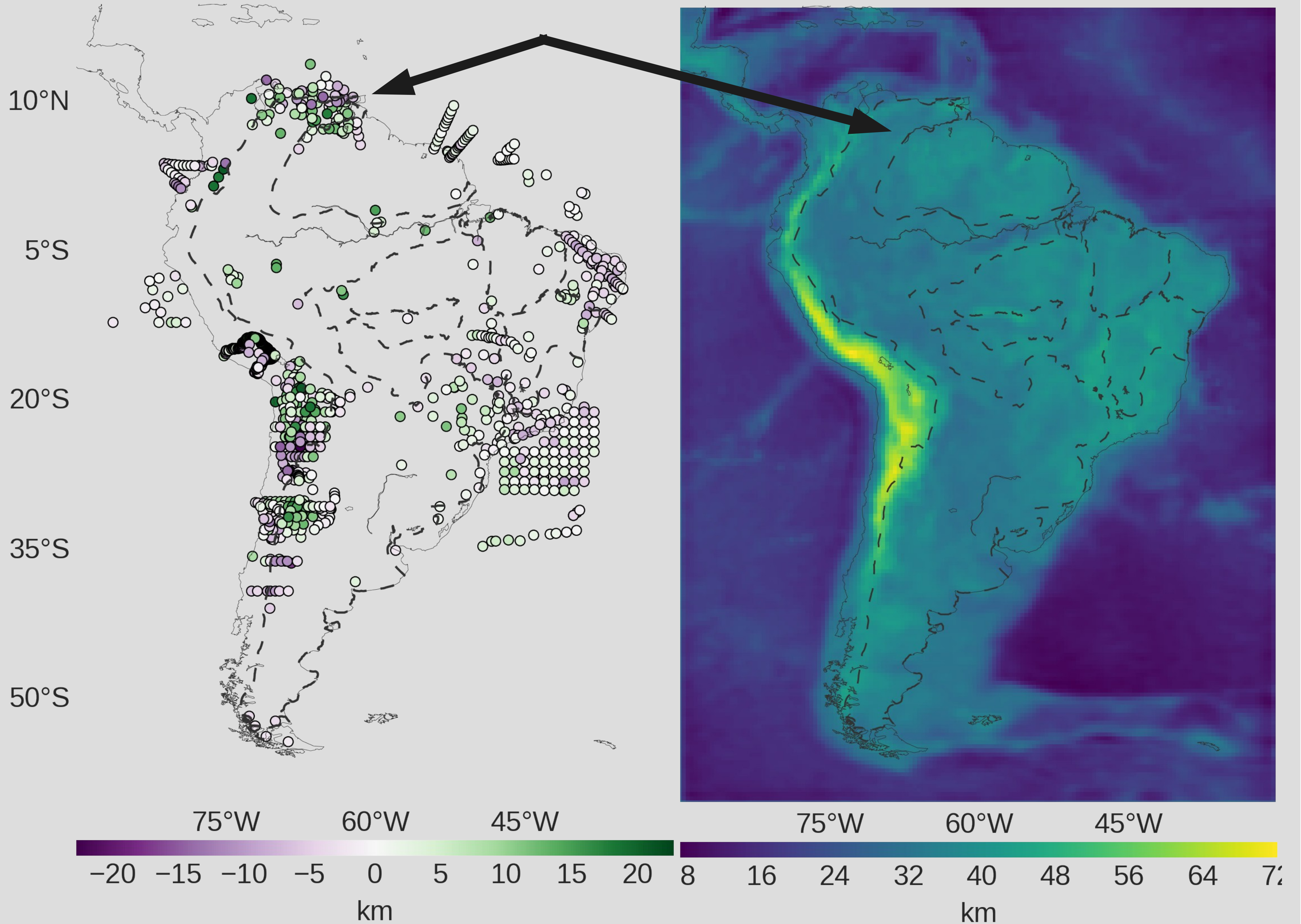


10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−24
−16
−8
0
8
16
mGal
75°W
60°W
45°W
16
24
32
40
48
56
64
km

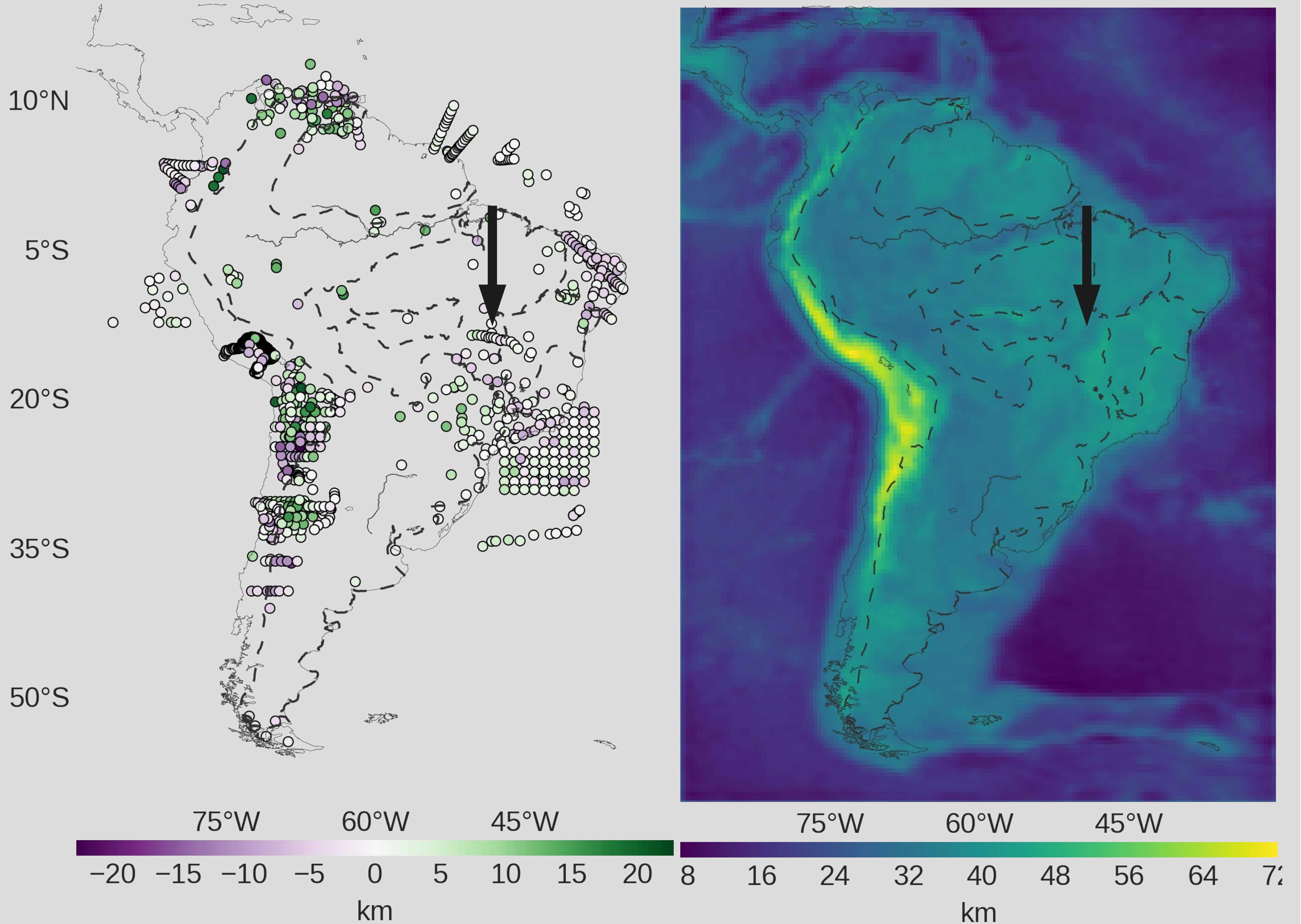

10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
km


10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
km

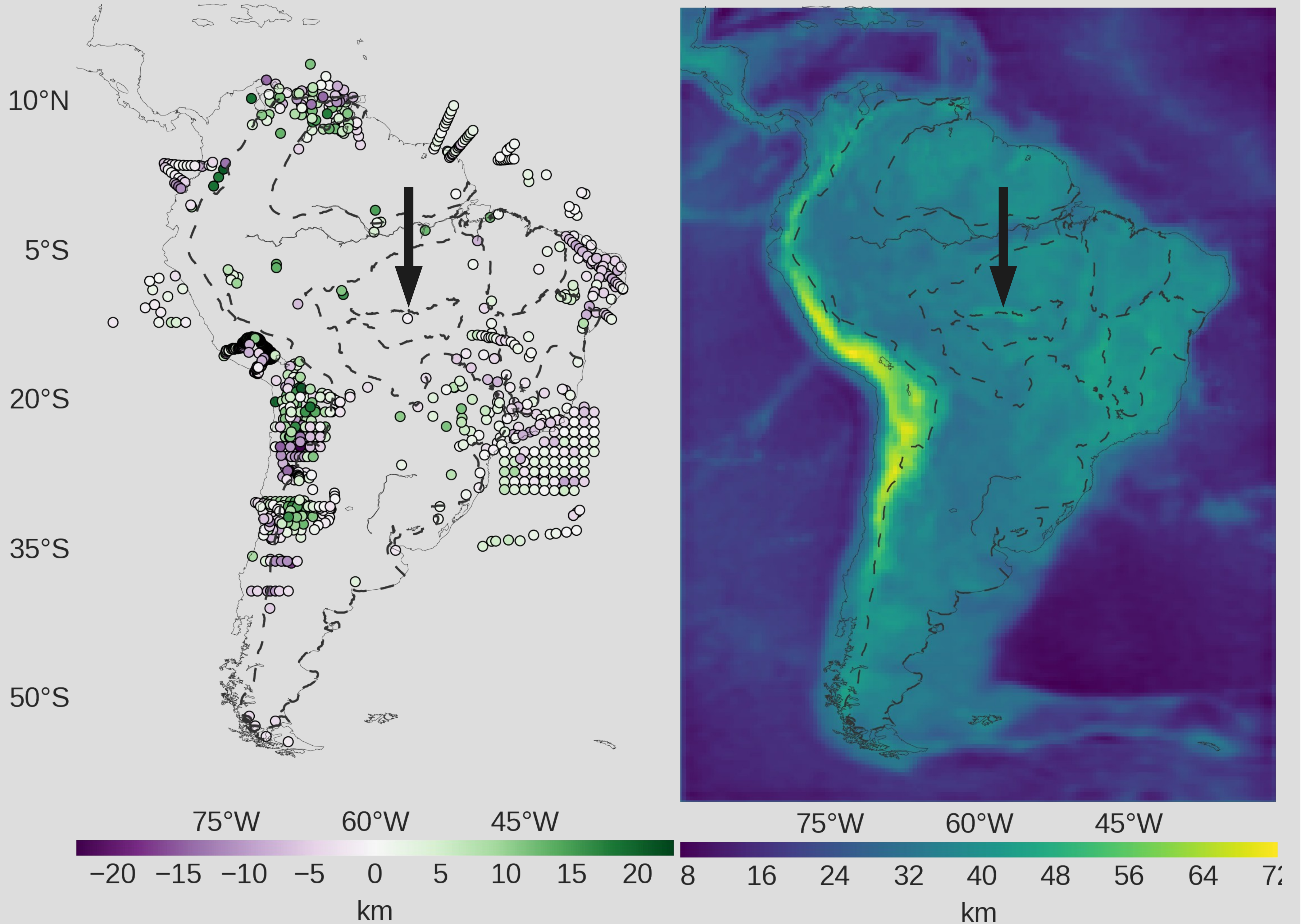

10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
km

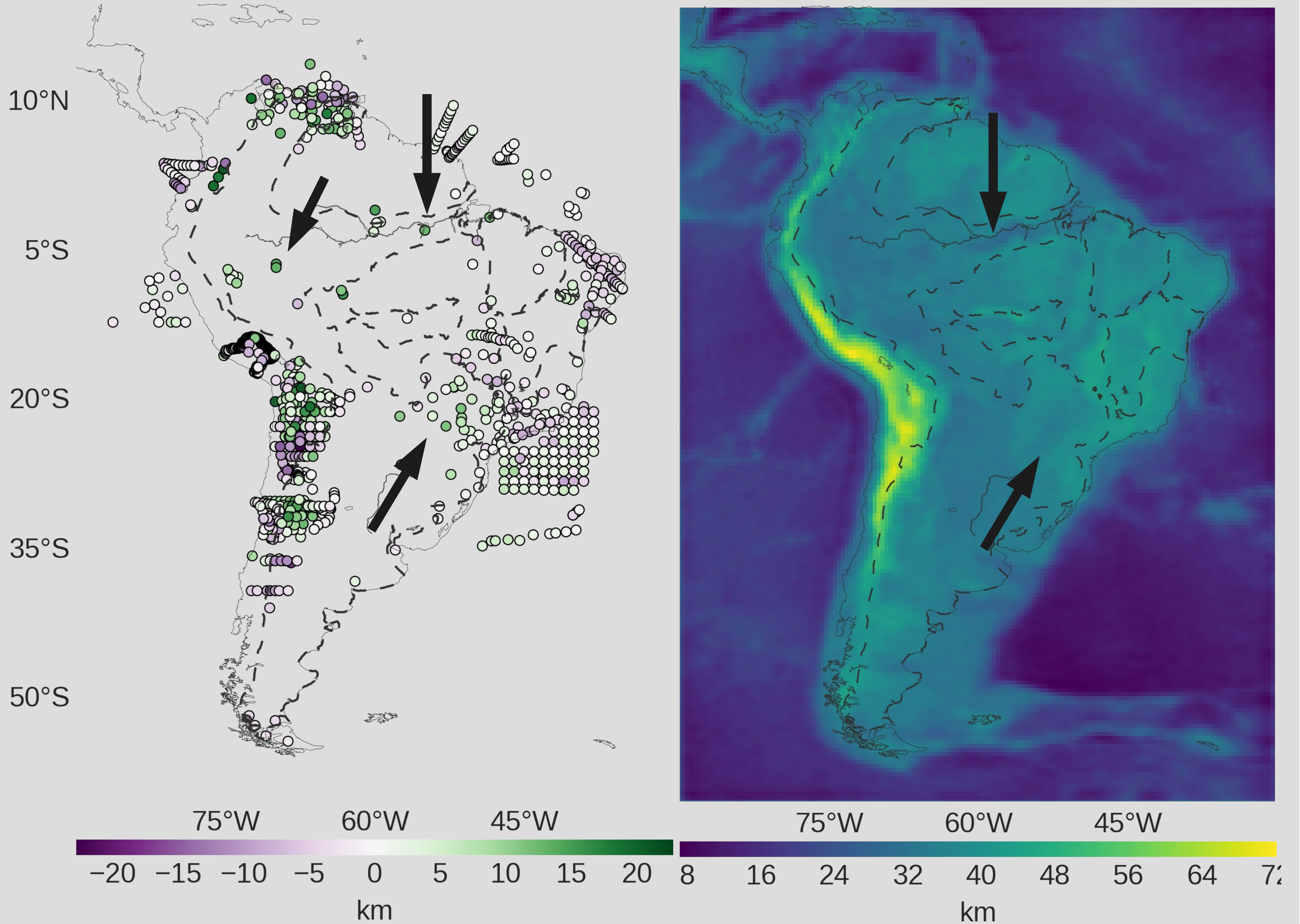


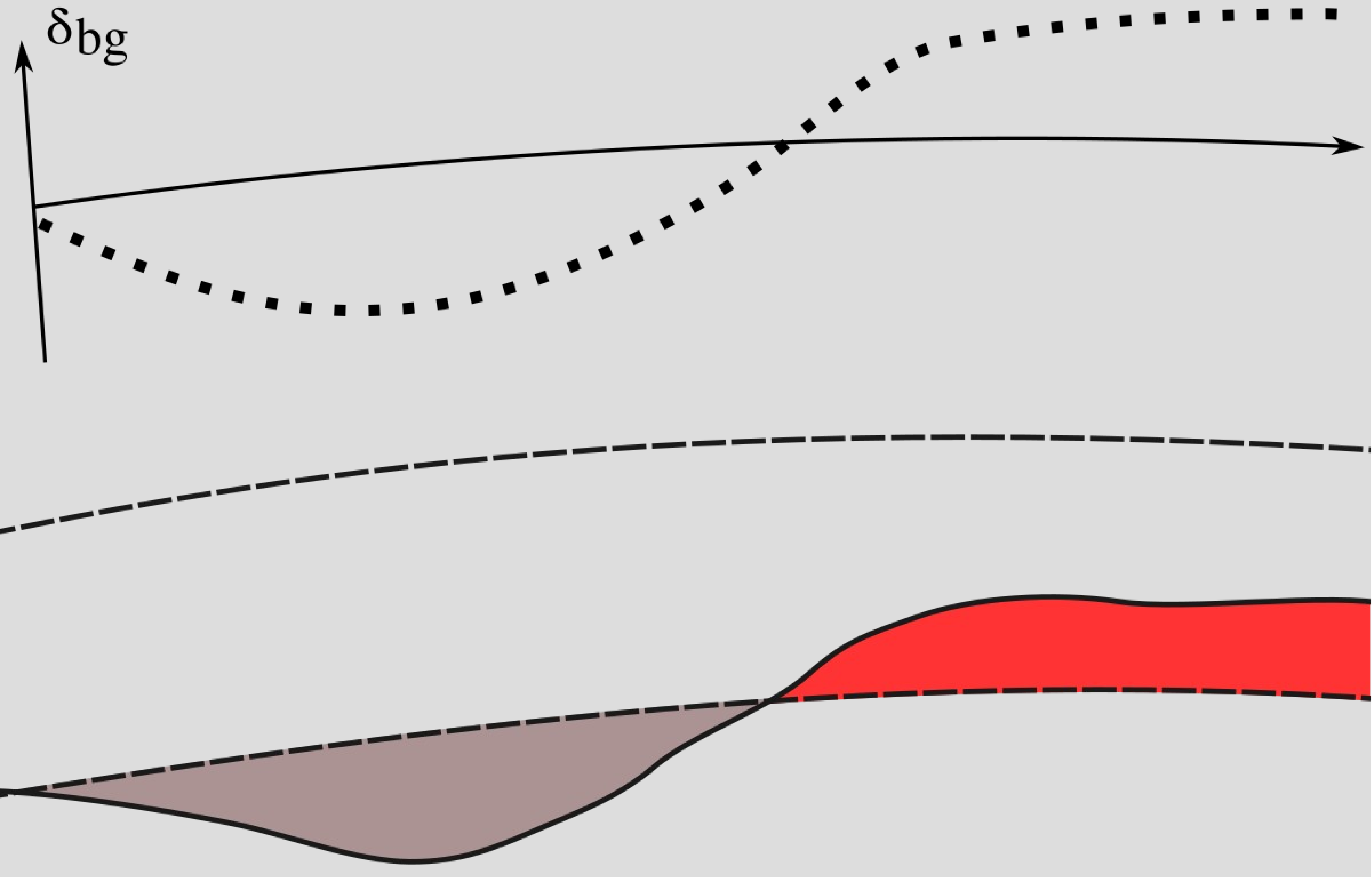
10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
km

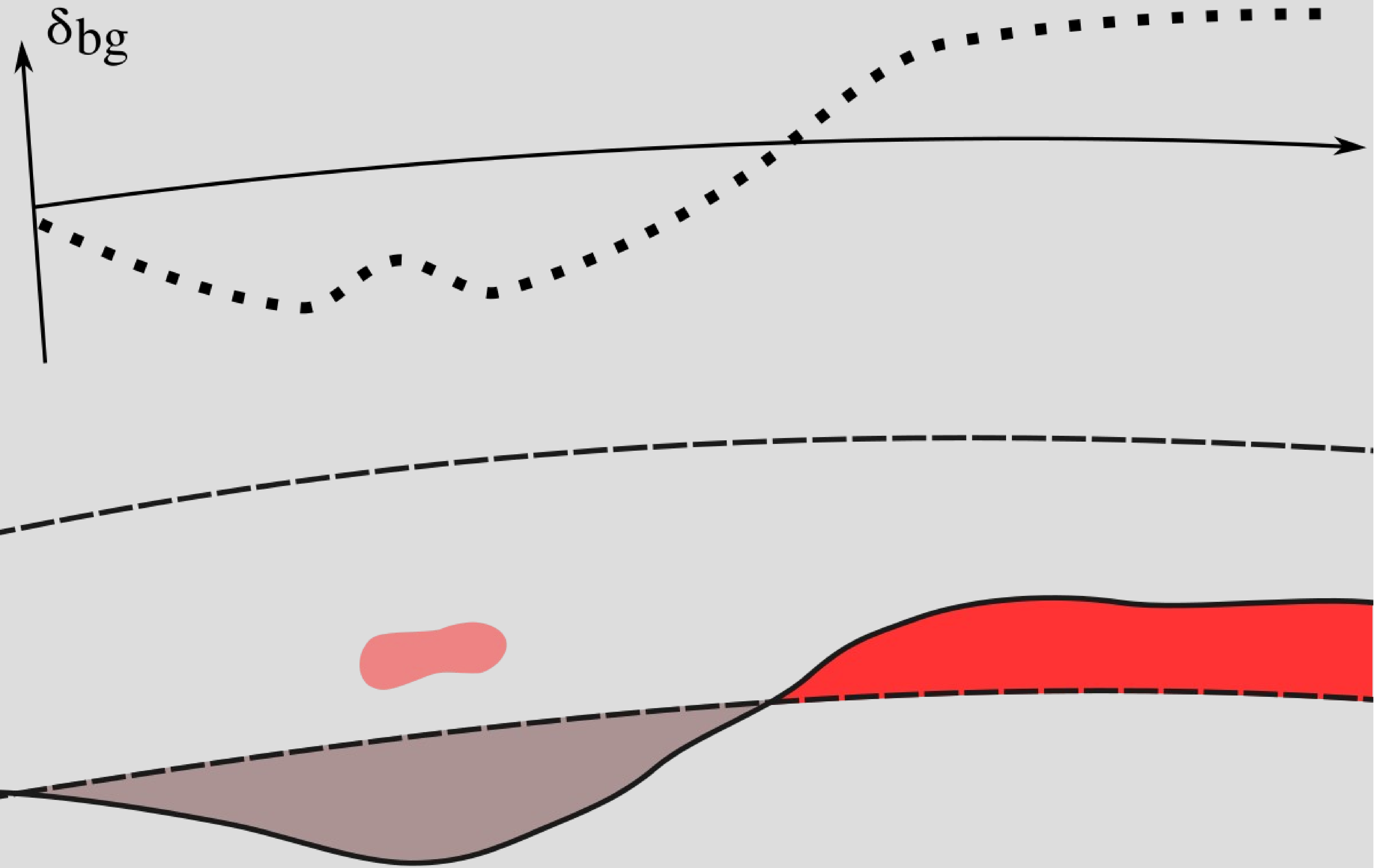


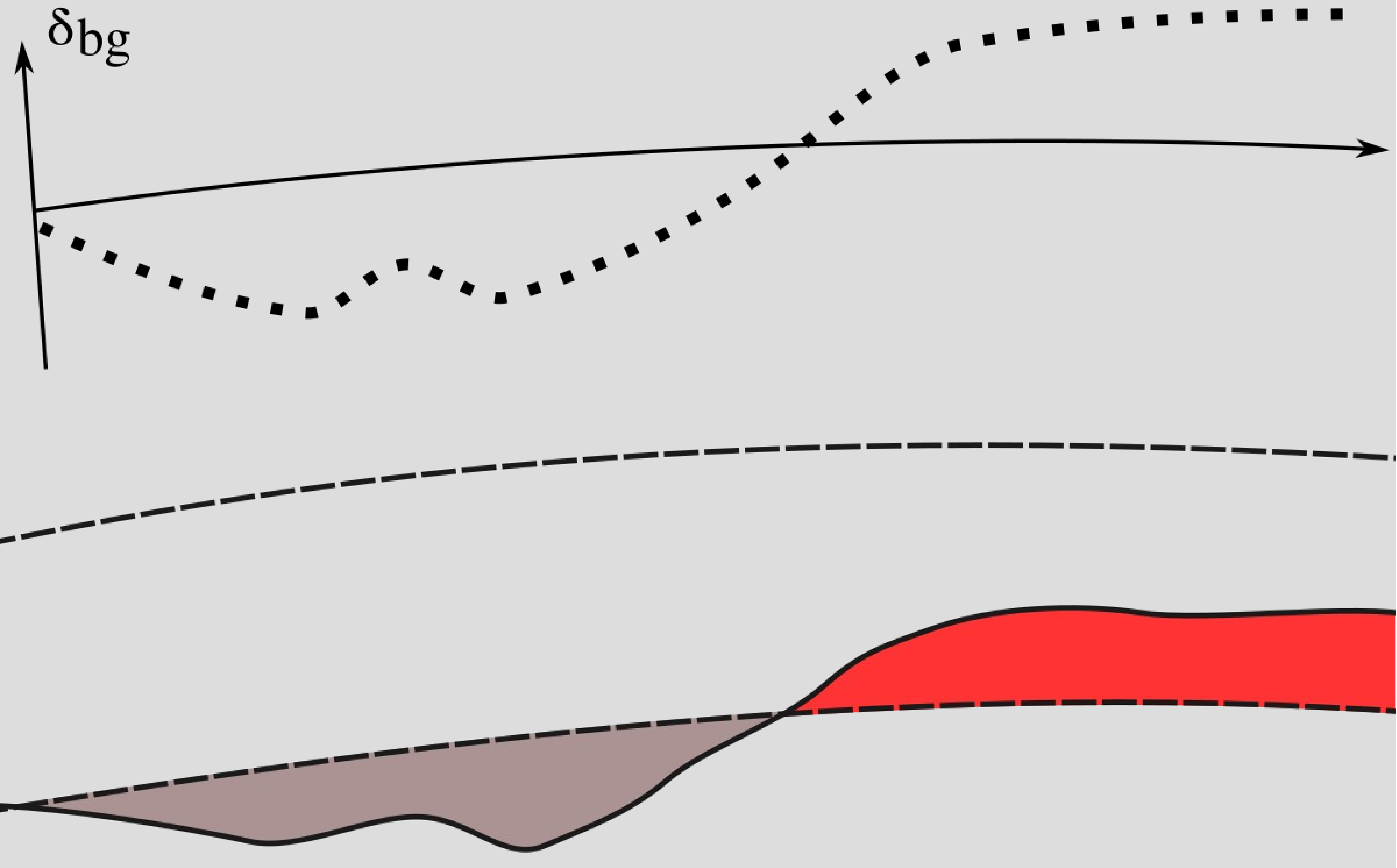
10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
km


10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
km


10°N
5°S
20°S
35°S
50°S
75°W
60°W
45°W
−20
−15
−10
−5
0
5
10
15
20
km
75°W
60°W
45°W
8
16
24
32
40
48
56
64
72
km


δ_{bg}


δ_{bg}


δ_{bg}

conclusão

Novo método

Baseado em Bott (1960) e Silva et al. (2014)

Tesseroides

Gauss-Newton + Regularização

Matrizes esparsas

Validação cruzada $\longrightarrow$ μ $\Delta\rho$ z_{ref}

Moho América do Sul

Compatível com soluções anteriores (grav e sismo)

Correções (topo e sedimentos) apropriadas

~6 km desvio padrão com sísmica

Diferença grande concentrada nos Andes

Moho América do Sul

Compatível com soluções anteriores (grav e sismo)

Correções (topo e sedimentos) apropriadas

~6 km desvio padrão com sísmica

Diferença grande concentrada nos Andes

Resolução pode ser falsa

Depende de correções corretas

Difícil estimar o erro

Introdução

Tesseroids

Fatiando a Terra

Inversão Moho

Conclusão

conclusões finais

programa A

programa B

Novo método

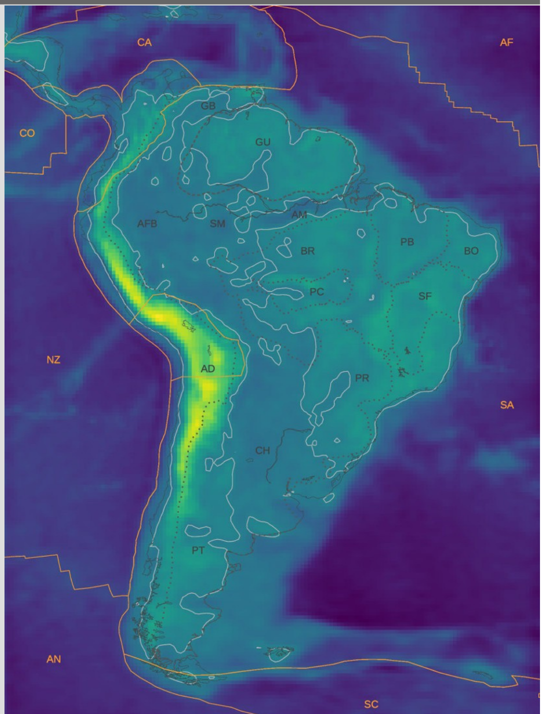
Tesseroids

modelagem direta
aprox. esférica

fatiando a terra

modelagem direta
otimização
regularização, etc

método: inversão não-linear rápida
aplicação: Moho da América do Sul



Tesseroids

Algoritmo aprimorado

Quantificação do erro

Usuários

Tesseroids

Algoritmo aprimorado

Quantificação do erro

Usuários

Desenvolvedores

Desenvolvimento estagnado

fatiando a terra

- Biblioteca em Python
- Vários módulos
- Pesquisa + educação
- Usuários + desenvolvedores
- Além dessa tese

fatiando a terra

Facilitar a participação:

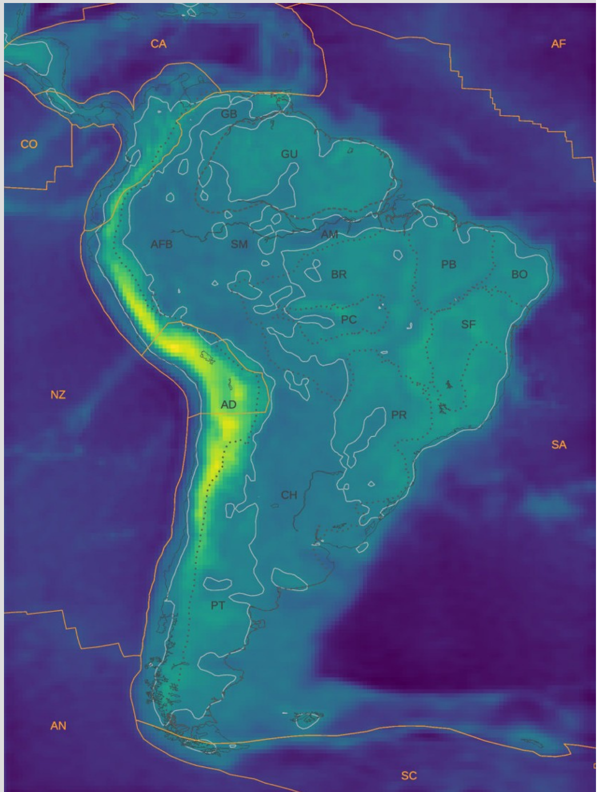
- Guia do desenvolvedor
- Guia do usuário
- Workshop
- Videos?

Inversão rápida + Moho Am. do Sul

Método novo + eficiente

Approx. esférica

Python + Fatiando a Terra



Inversão rápida + Moho Am. do Sul

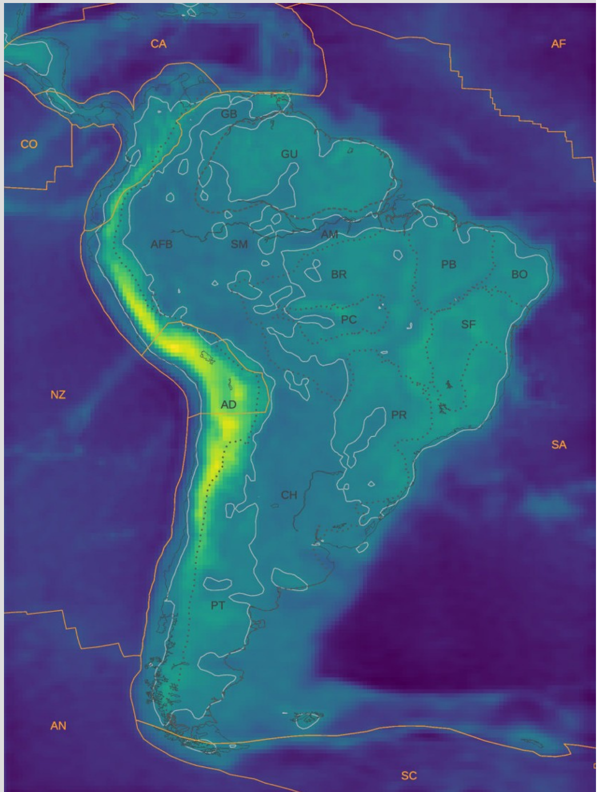
Método novo + eficiente

Approx. esférica

Python + Fatiando a Terra

Depende das correções

Incerteza



PINGA

Grupo de problemas inversos em geofísica

open science

código + dados + figuras

pinga-lab.org

leouieda.com

THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

https://xkcd.com/1403