

Metodologías y herramientas computacionales para el procesamiento y modelado de datos gravimétricos

Lic. Santiago Soler

Dr. Mario E. Gimenez
Director



Dr. Leonardo Uieda
Codirector



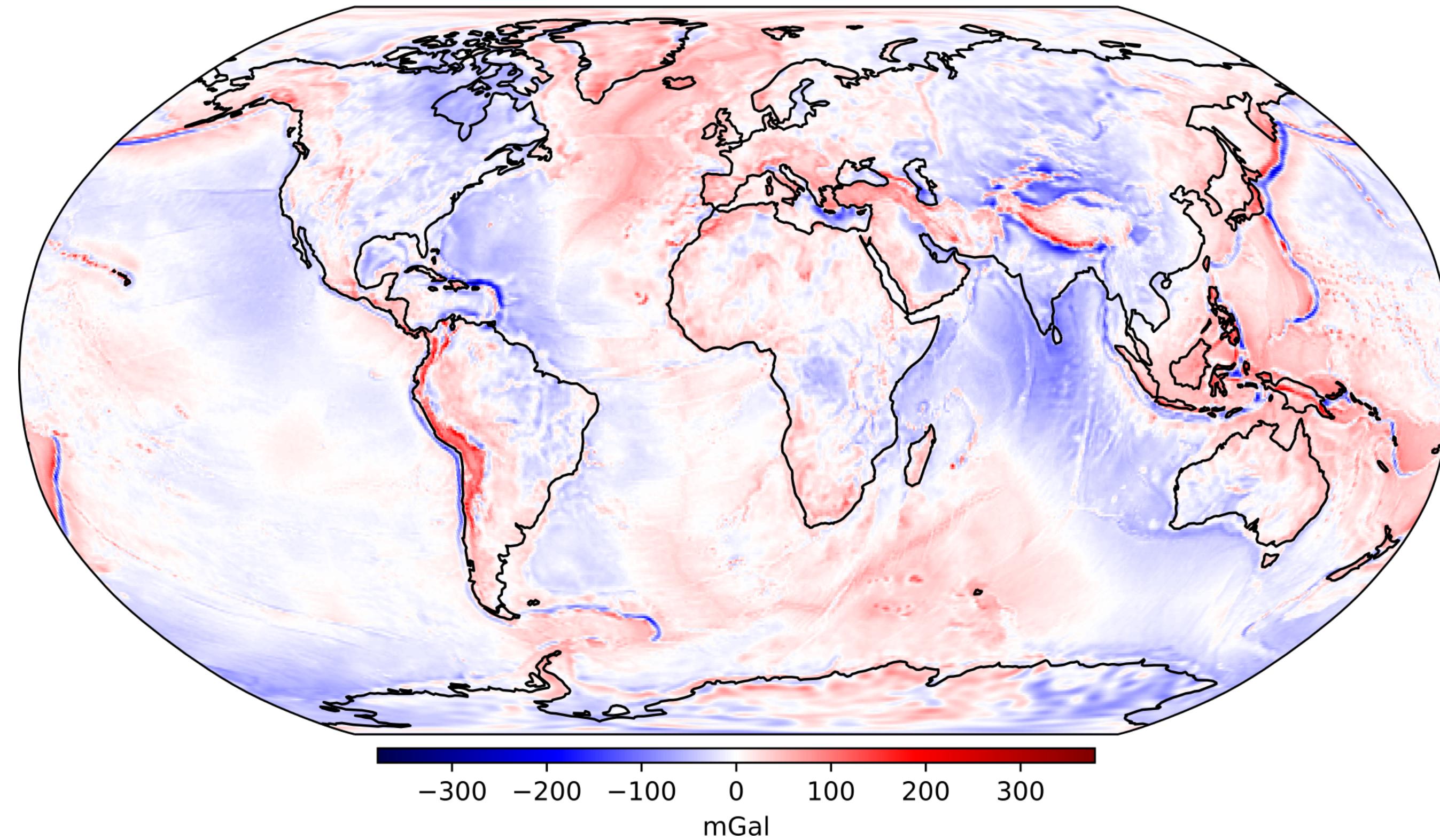
Contenido

1. Introducción
2. Teseroides densidad variable
3. Fuentes equivalentes potenciadas por gradiente
4. Fatiando a Terra
5. Conclusiones

Introducción

Gravedad terrestre

Disturbio de gravedad de la Tierra



Información sobre el interior de la Tierra

Problemas

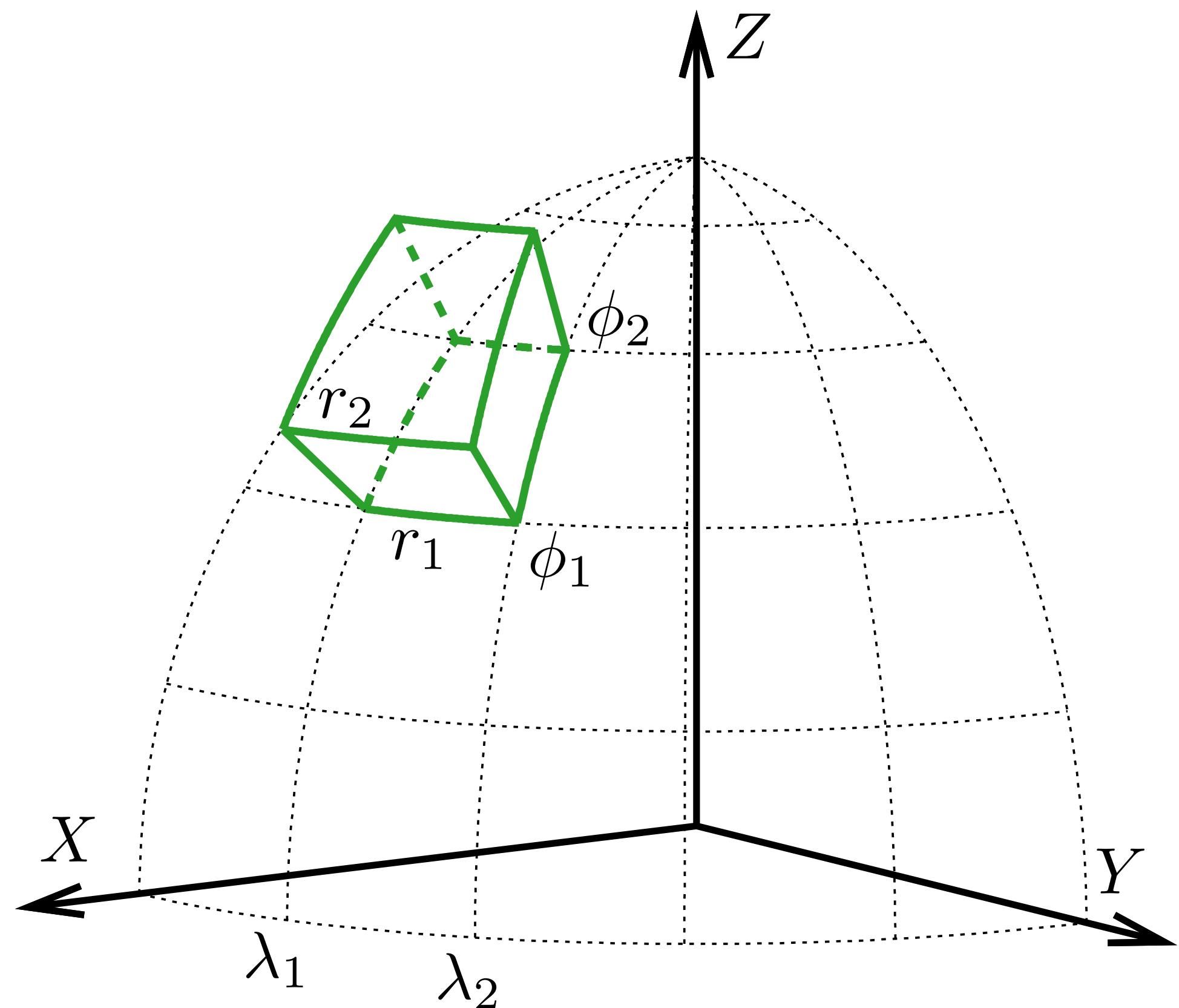
- Modelado de grandes estructuras: **curvatura de la Tierra**
- Procesamiento de **grandes cantidades de datos**
- Software disponible

Objetivos

- Desarrollo de nuevas metodologías
- Implementaciones mediante software de código abierto

Teseroides de densidad variable

¿Qué es un teseroide?



Problema

Campos gravitatorios

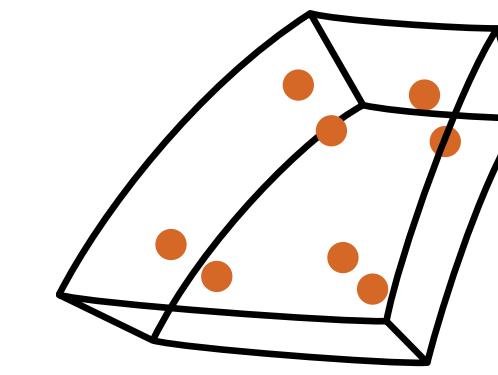
$$V(\mathbf{p}) = G\rho \int_{r_1}^{r_2} \int_{\lambda_1}^{\lambda_2} \int_{\phi_1}^{\phi_2} \frac{\kappa}{\|\mathbf{p} - \mathbf{q}\|} dr' d\lambda' d\phi',$$

No poseen solución analítica

Solución

Cuadratura de Gauss-Legendre

$$V(\mathbf{p}) \cong G\rho A \sum_{i=1}^{N_r} \sum_{j=1}^{N_\lambda} \sum_{k=1}^{N_\phi} W_{ijk} \frac{\kappa_{ik}}{\|\mathbf{p} - \mathbf{q}_{ijk}\|}$$

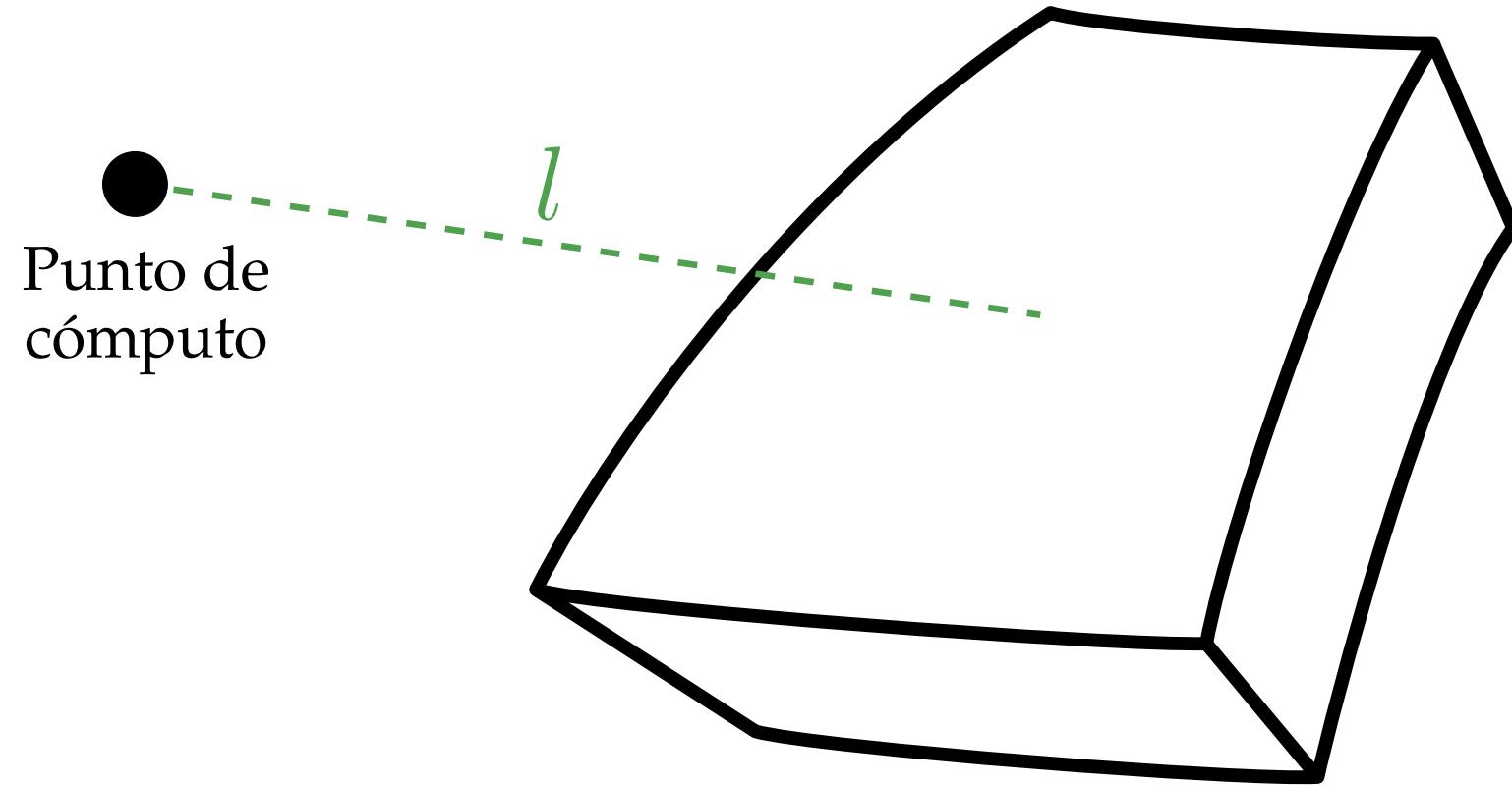


Aproxima el teseroide por masas puntuales

 Orden de la Cuadratura \Rightarrow  Precisión

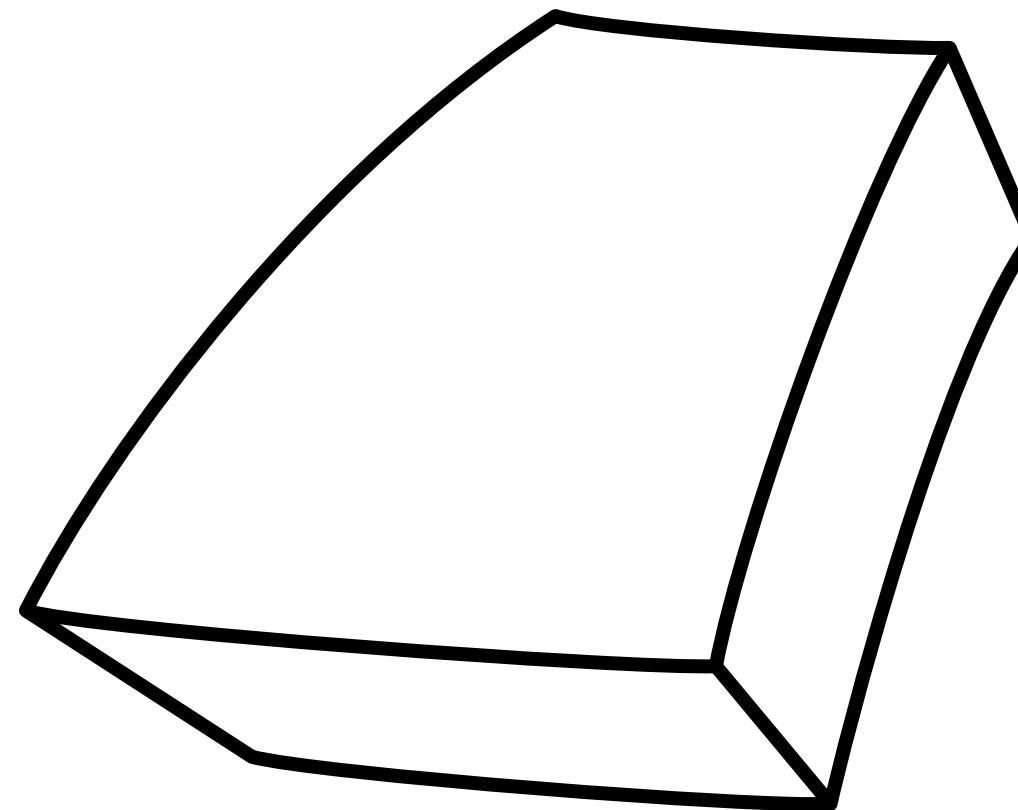
↑ Orden de la Cuadratura \Rightarrow ↑ Precisión

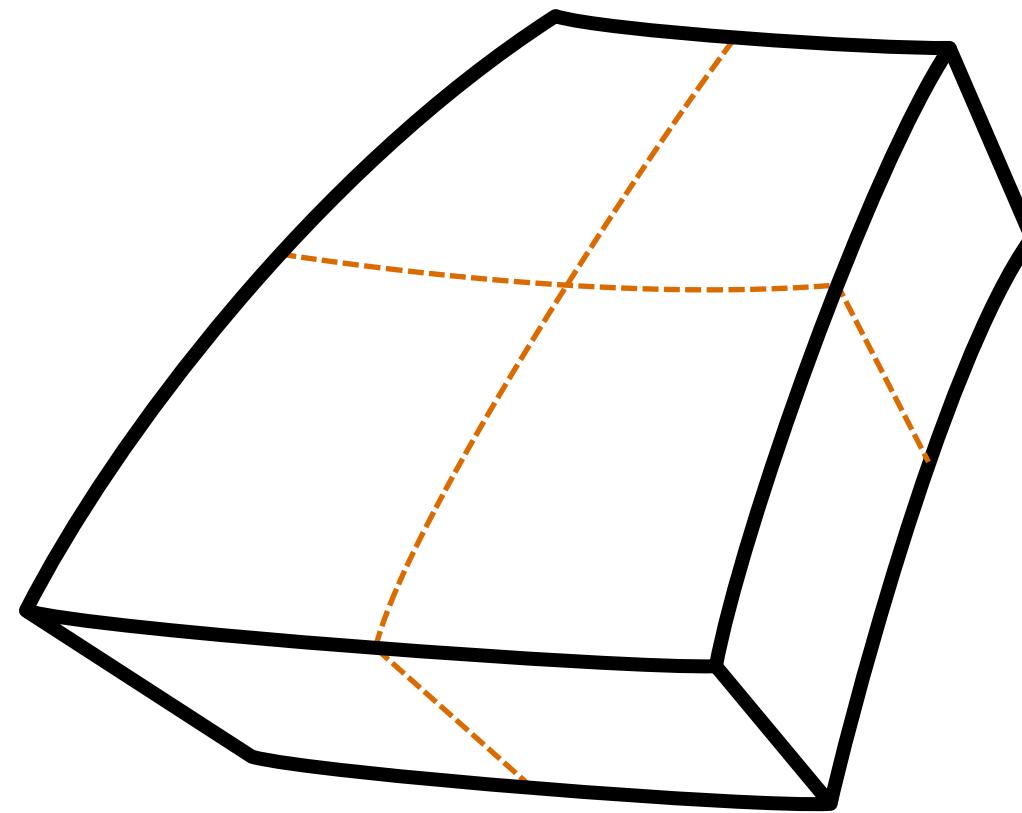
Pero... $\mathcal{O}(n^3)$ 😞

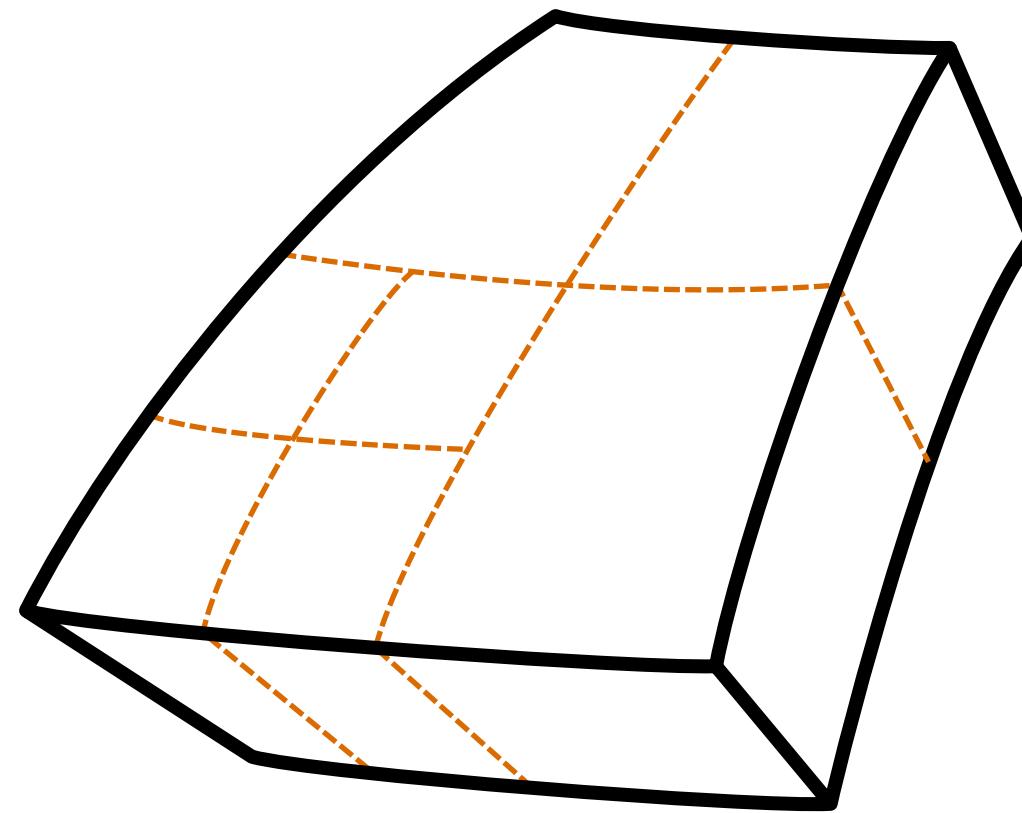


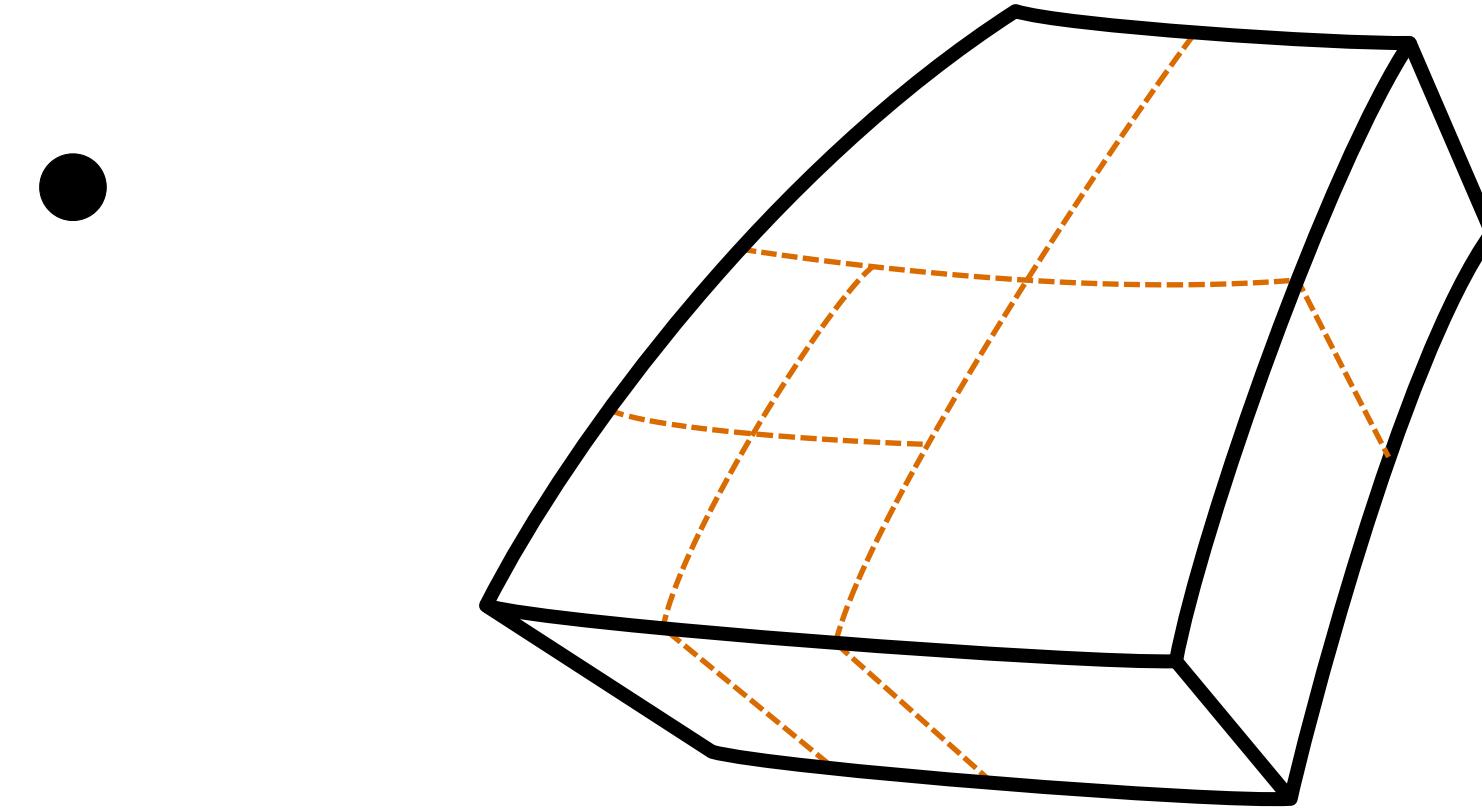
↑ Distancia a punto de cómputo \Rightarrow ↑ Precisión

Discretización adaptativa









Masas puntuales donde más se necesitan

Teseroides de densidad variable

$$V(\mathbf{p})=G \textcolor{red}{\rho} \int\limits_{r_1}^{r_2}\int\limits_{\lambda_1}^{\lambda_2}\int\limits_{\phi_1}^{\phi_2} \frac{\kappa}{\|\mathbf{p}-\mathbf{q}\|} \mathrm{d}r' \mathrm{d}\lambda' \mathrm{d}\phi',$$

$$V(\mathbf{p}) = G \rho \int_{r_1}^{r_2} \int_{\lambda_1}^{\lambda_2} \int_{\phi_1}^{\phi_2} \frac{\kappa}{\|\mathbf{p} - \mathbf{q}\|} dr' d\lambda' d\phi',$$

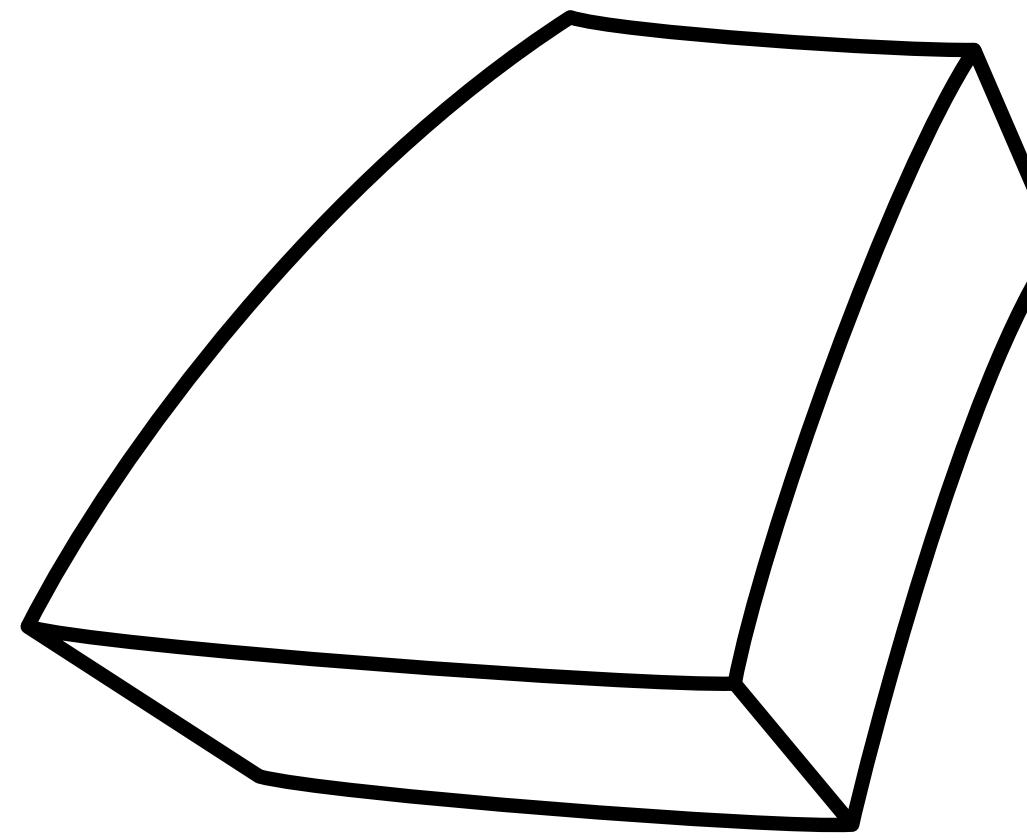
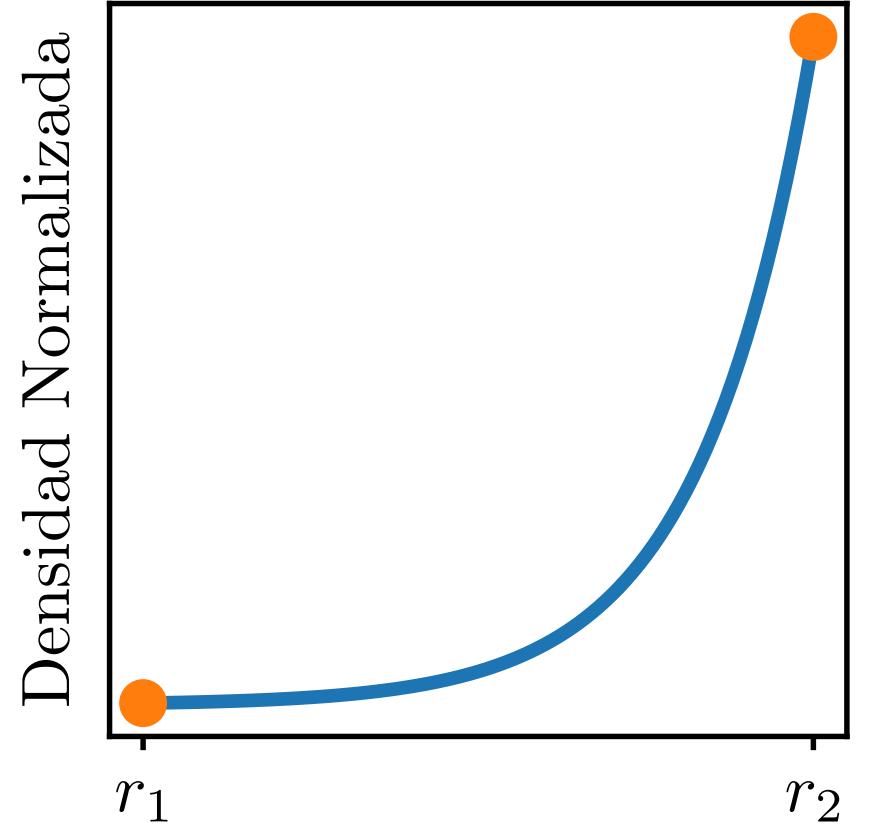


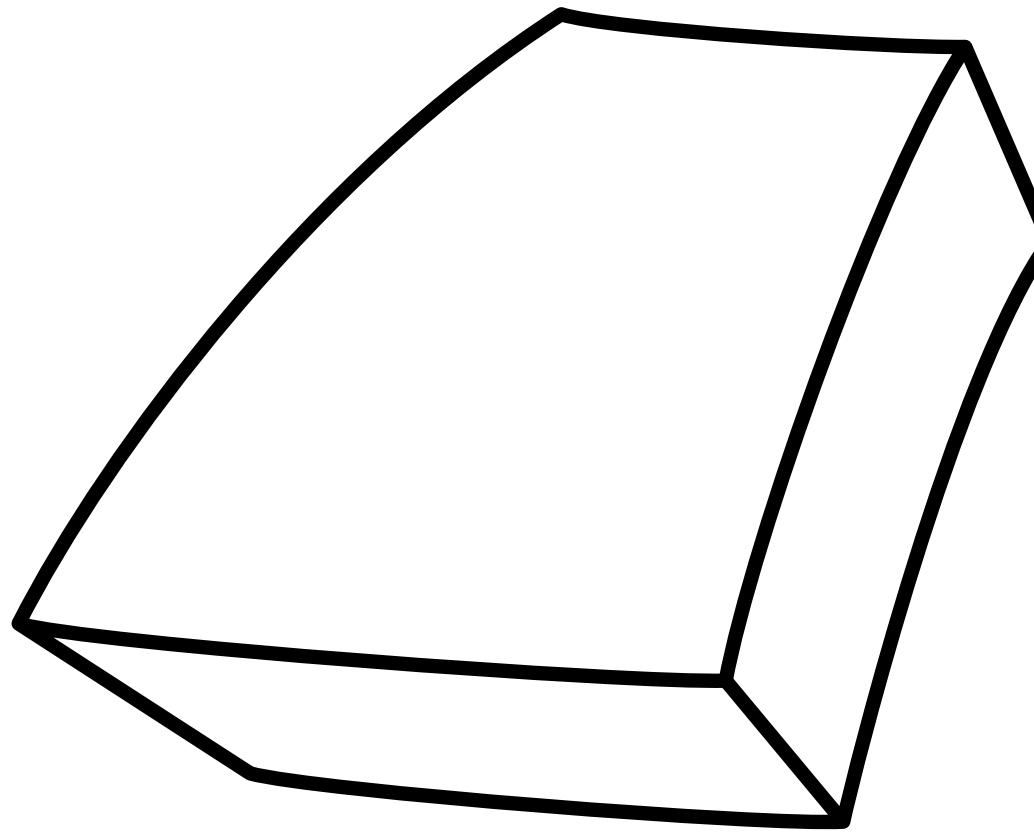
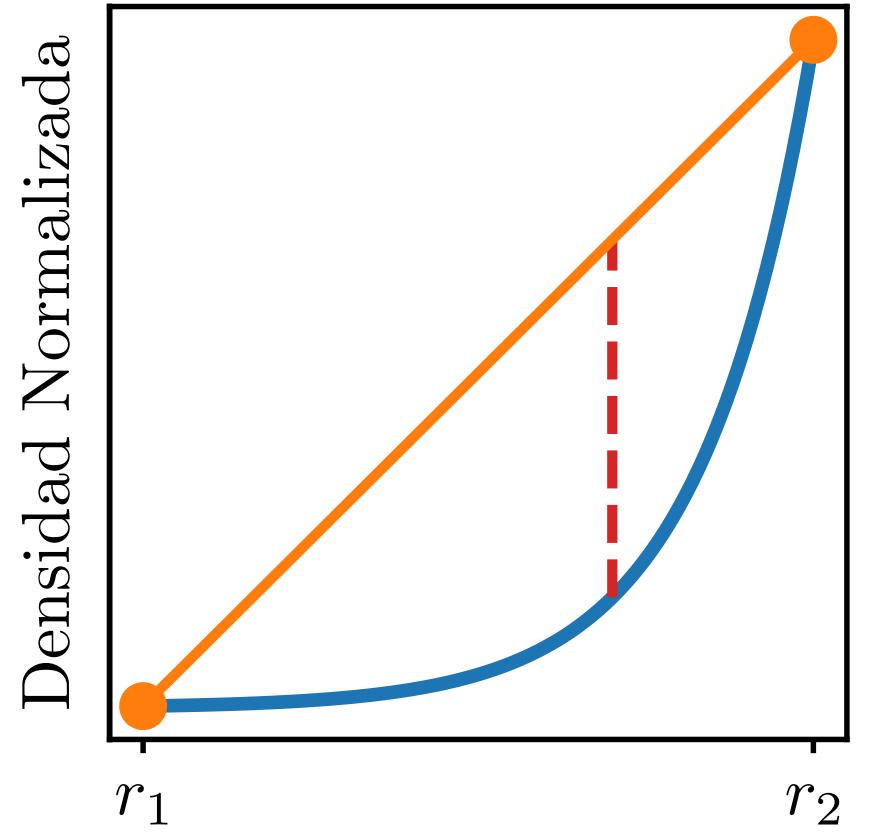
$$V(\mathbf{p}) = G \int_{r_1}^{r_2} \int_{\lambda_1}^{\lambda_2} \int_{\phi_1}^{\phi_2} \rho(r') \frac{\kappa}{\|\mathbf{p} - \mathbf{q}\|} dr' d\lambda' d\phi',$$

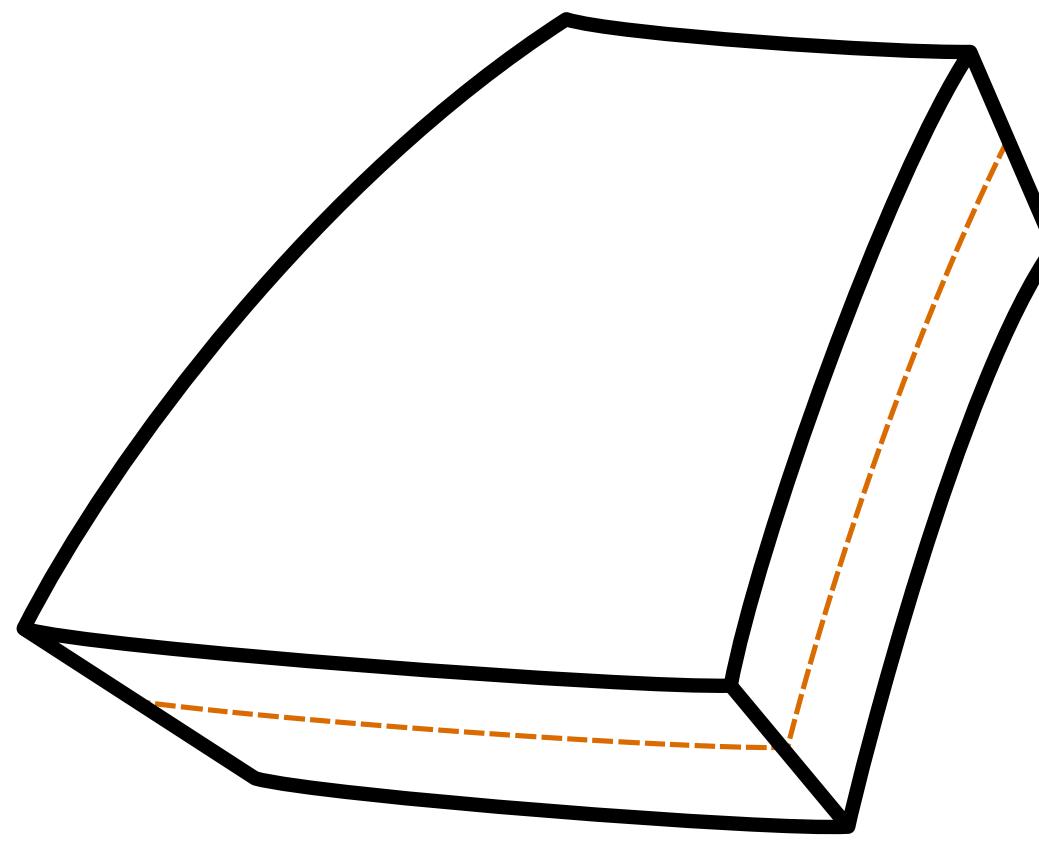
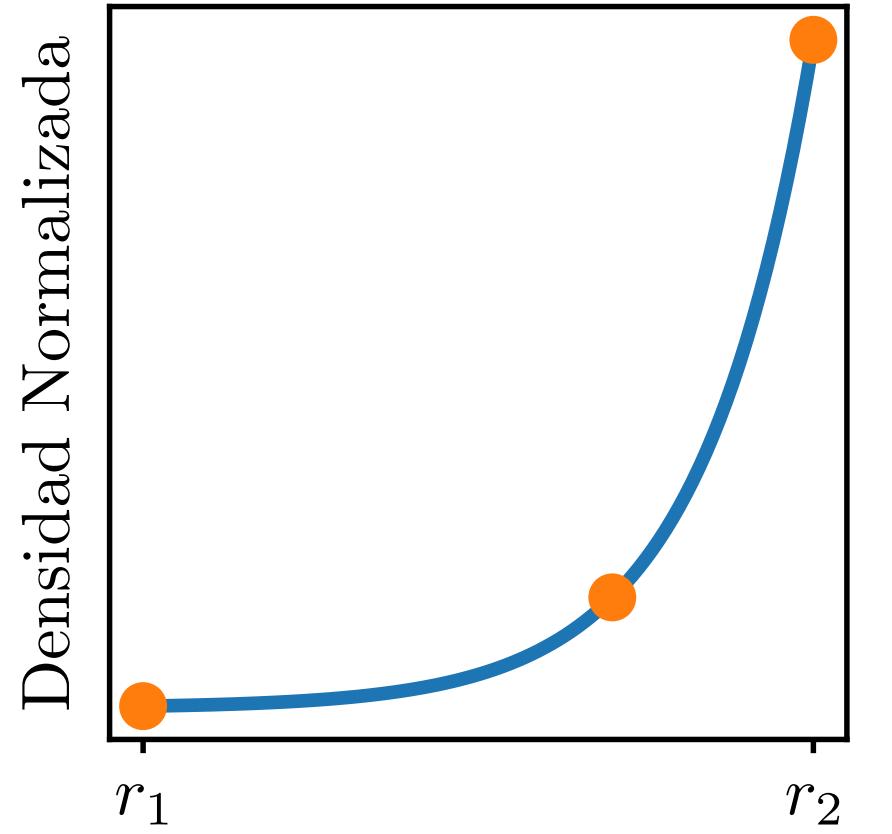
Desafío

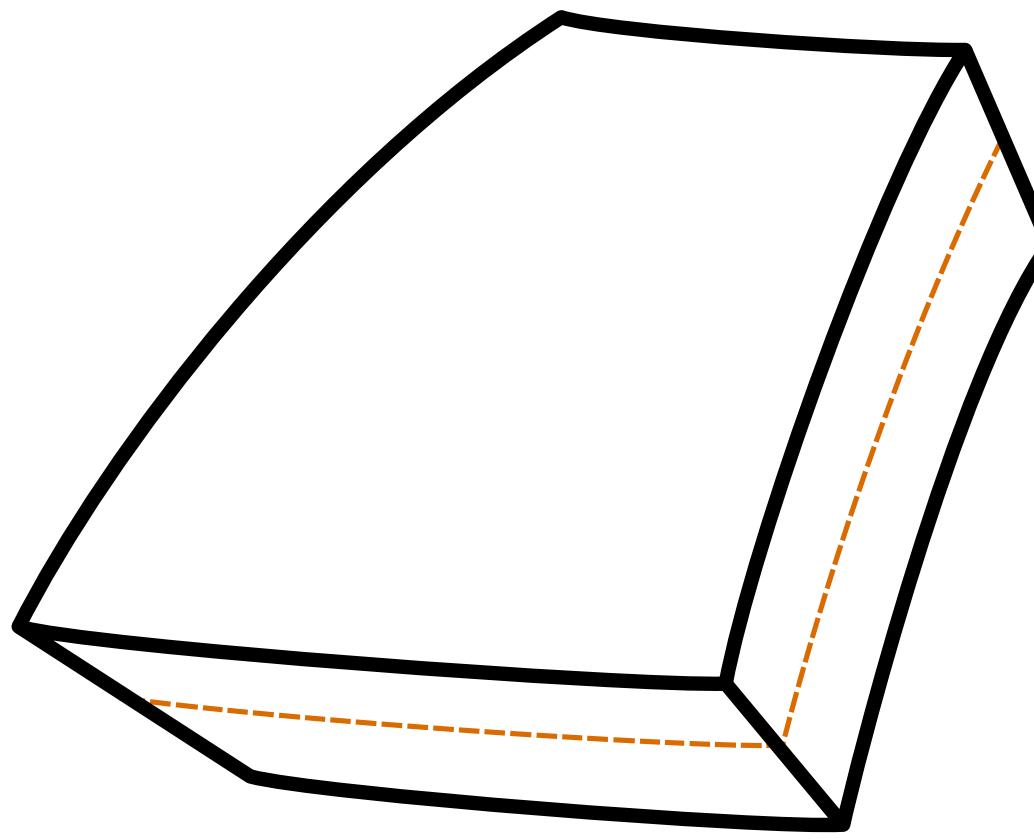
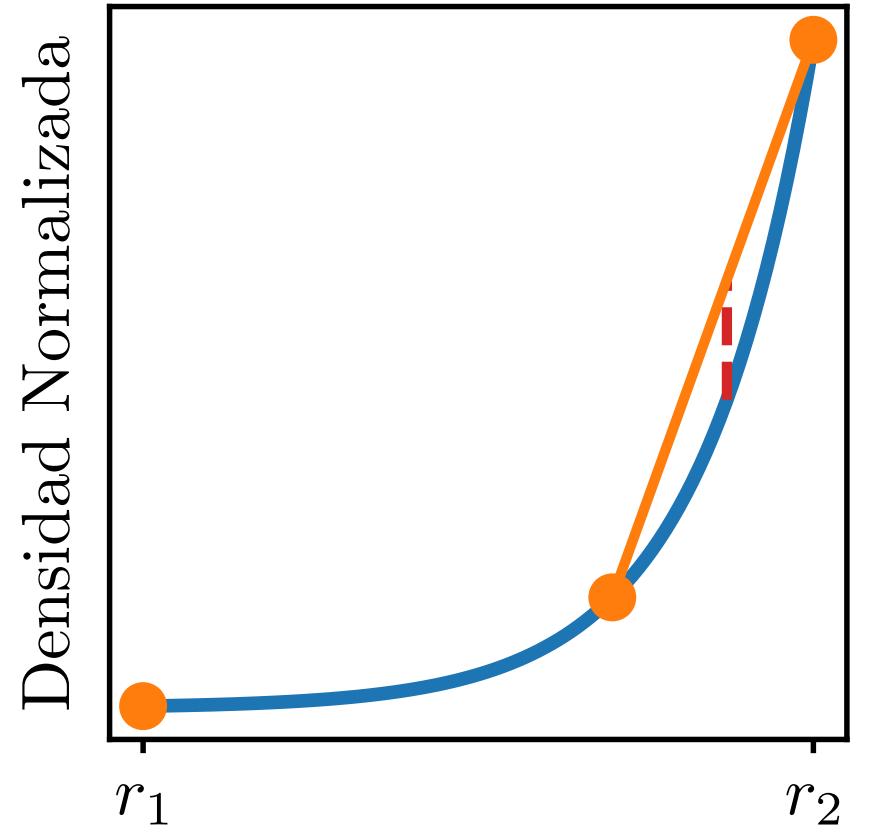
Función densidad en la aproximación numérica

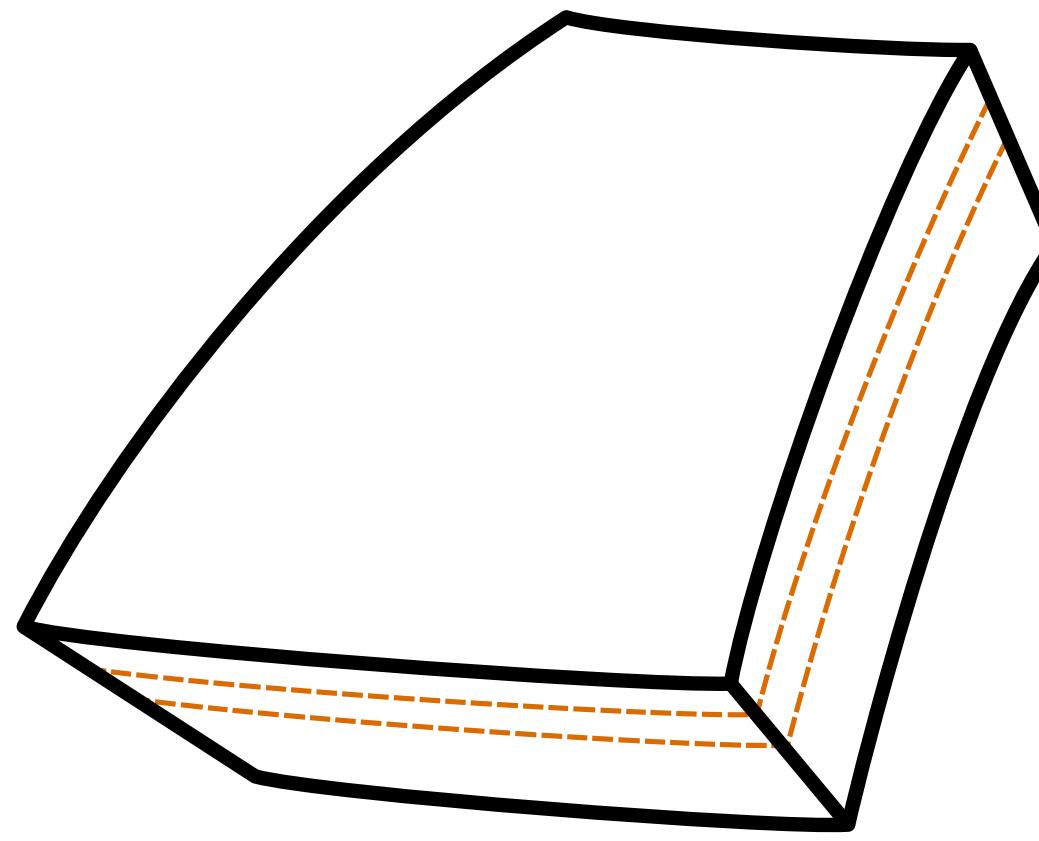
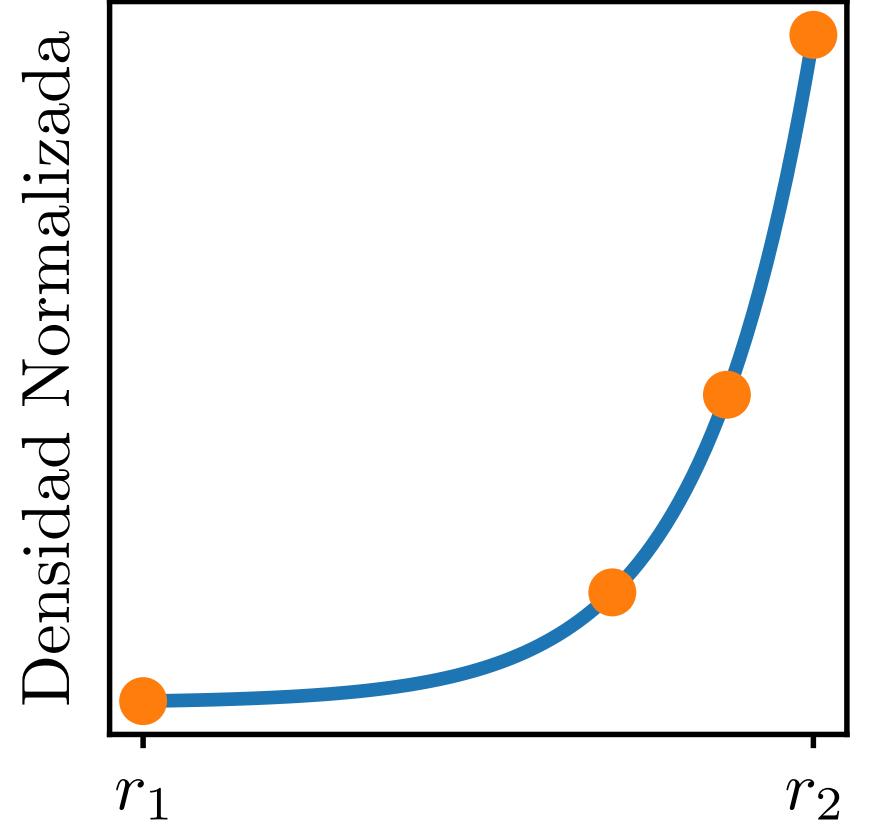
Discretización basada en densidad

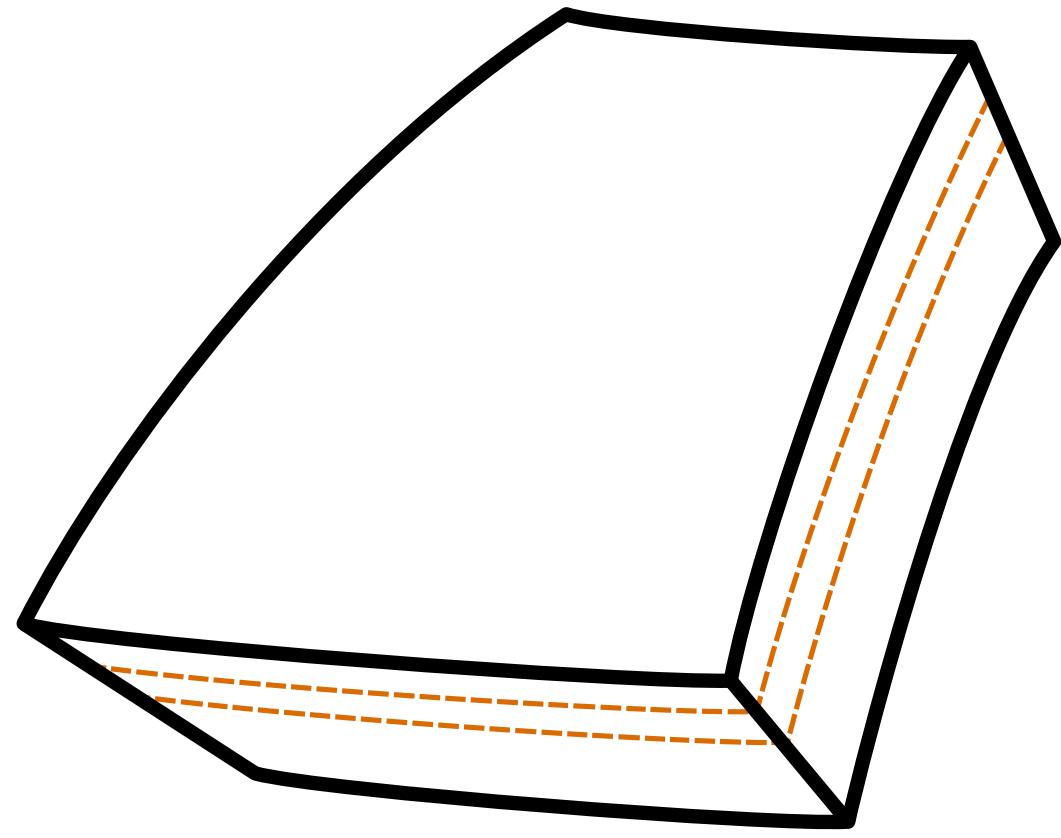
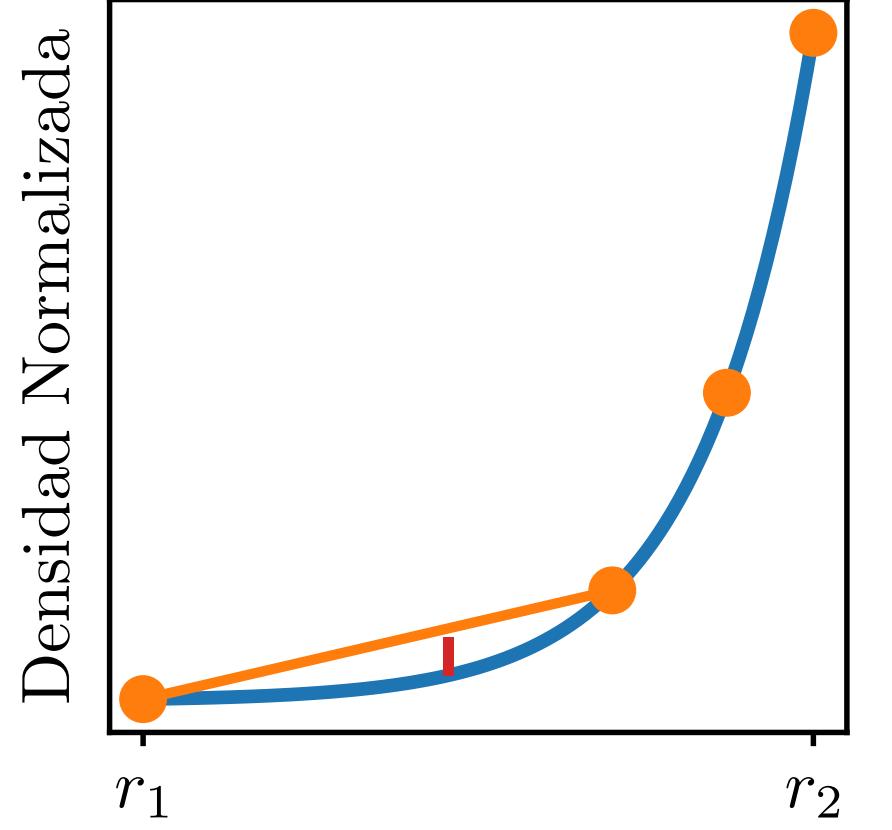


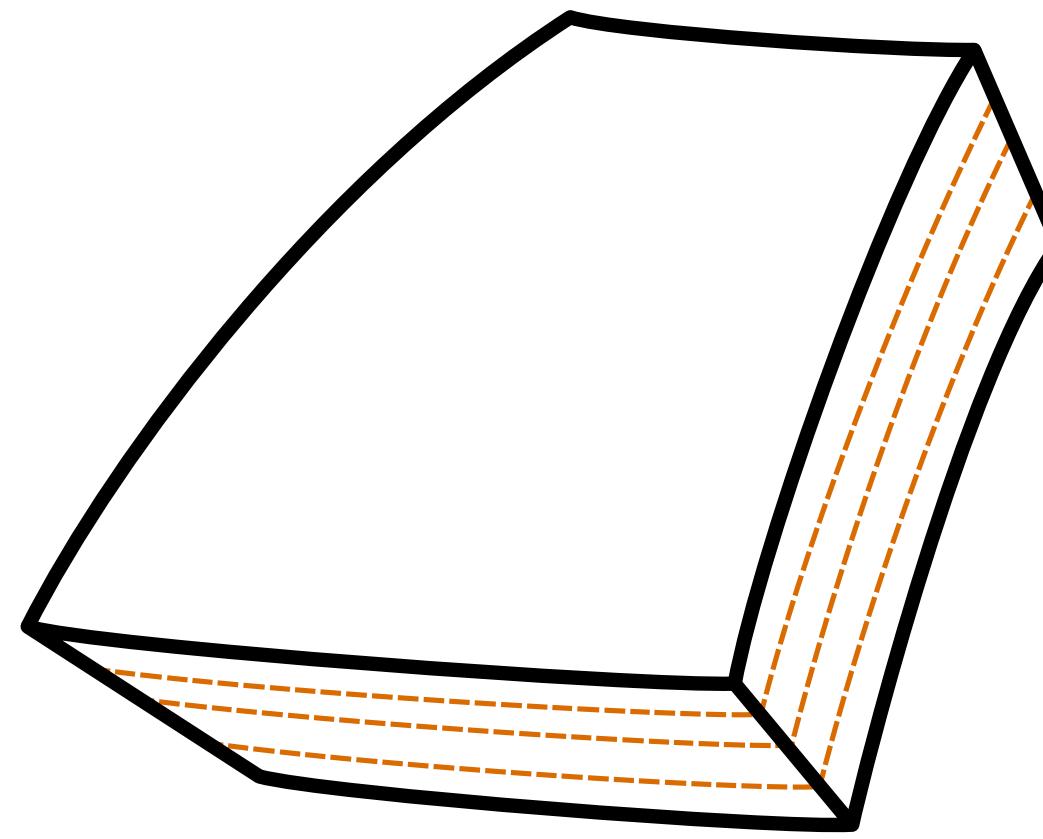
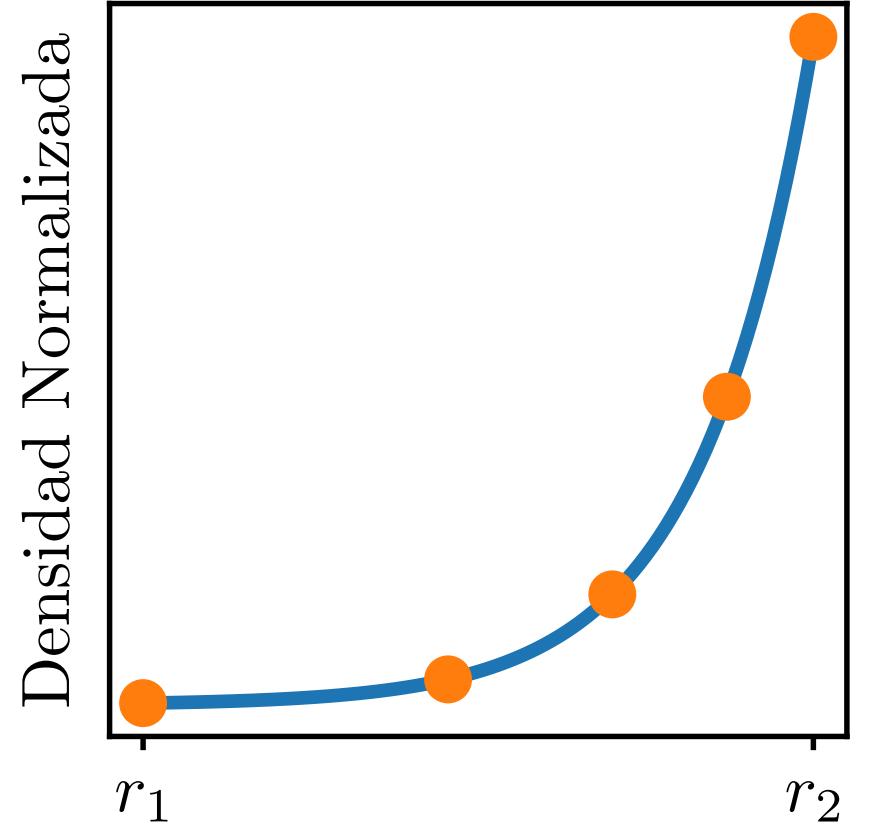




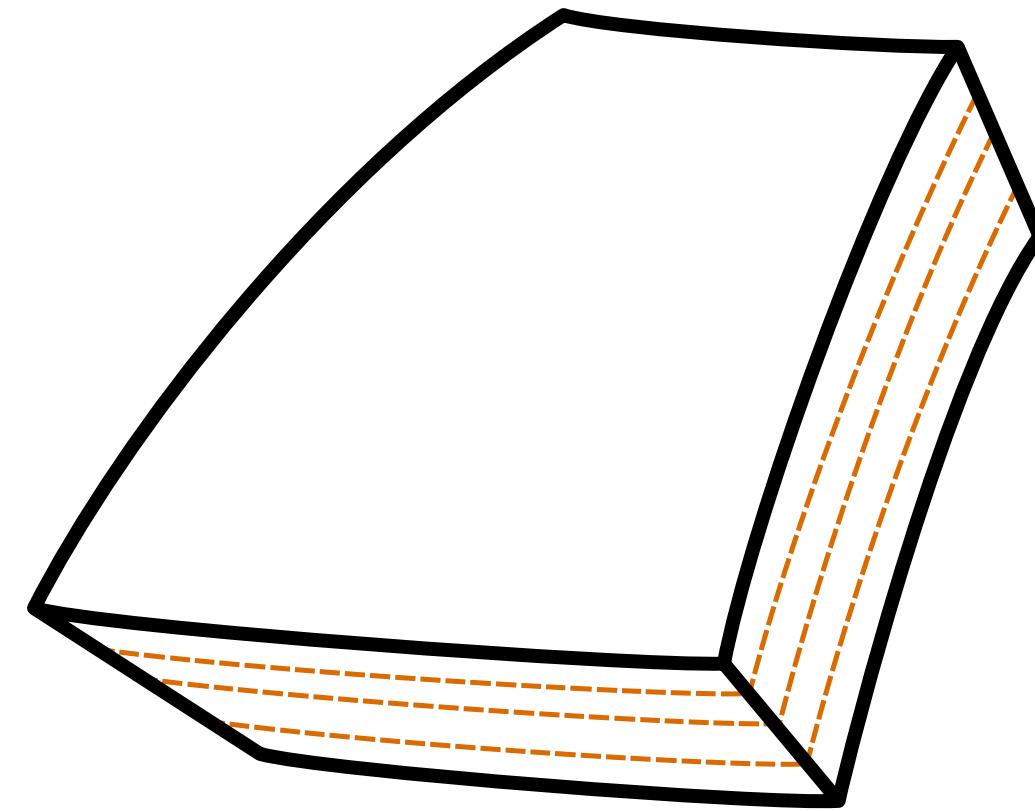


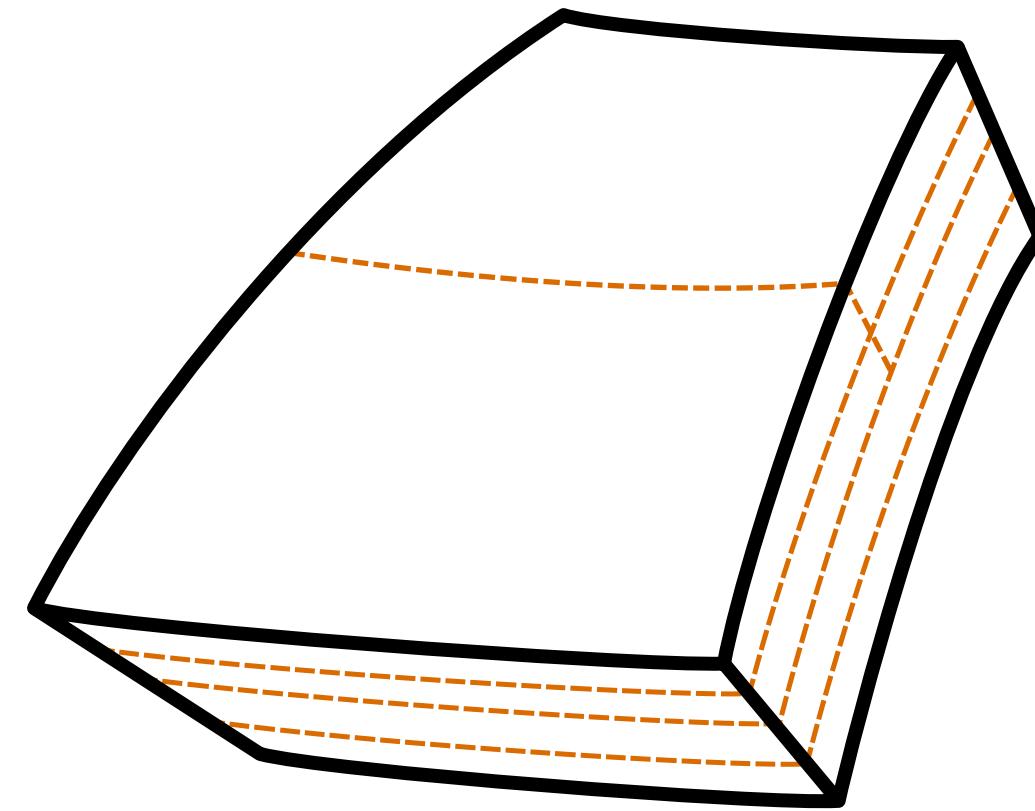


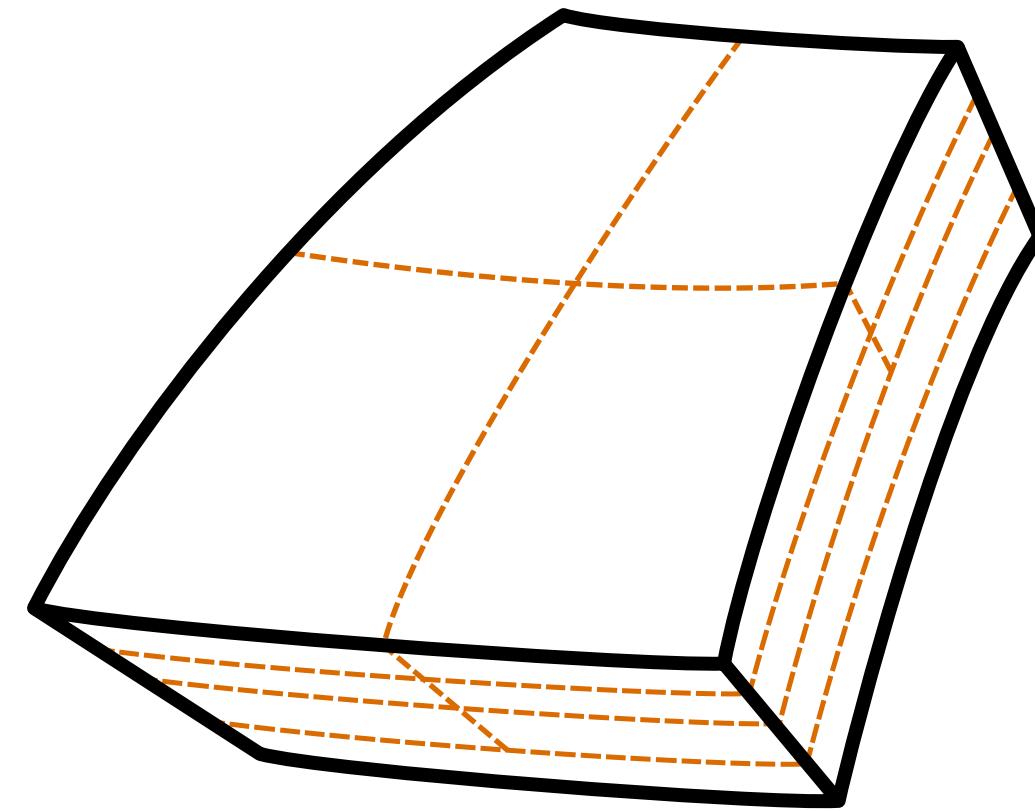


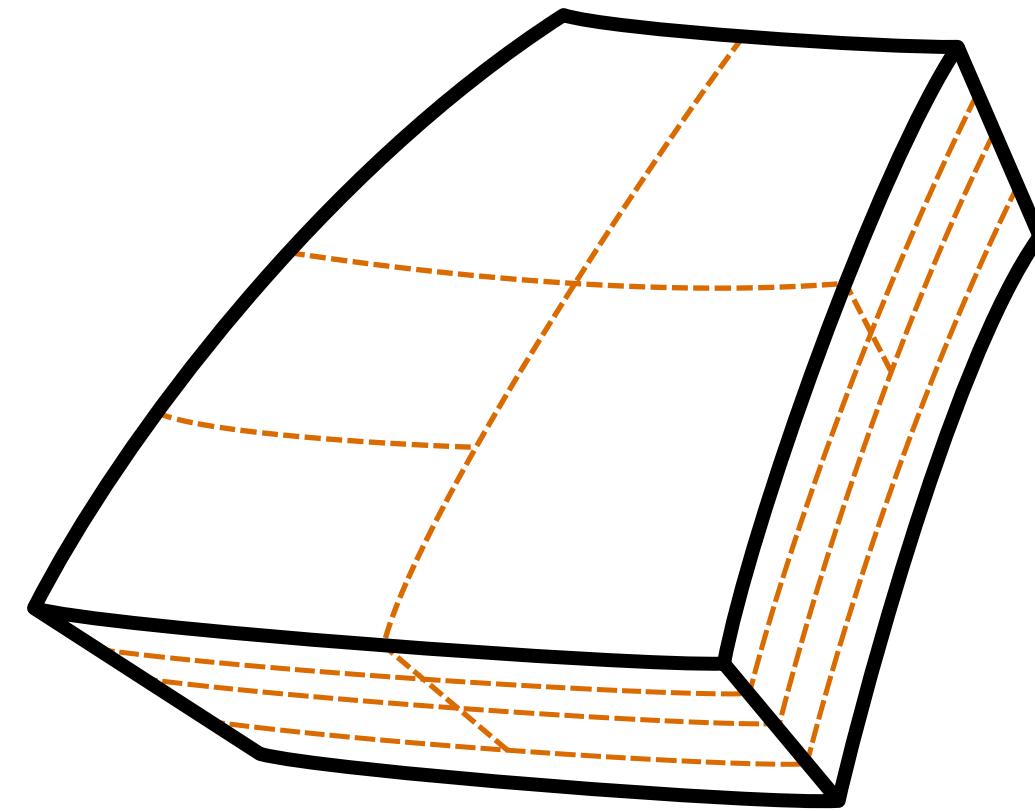


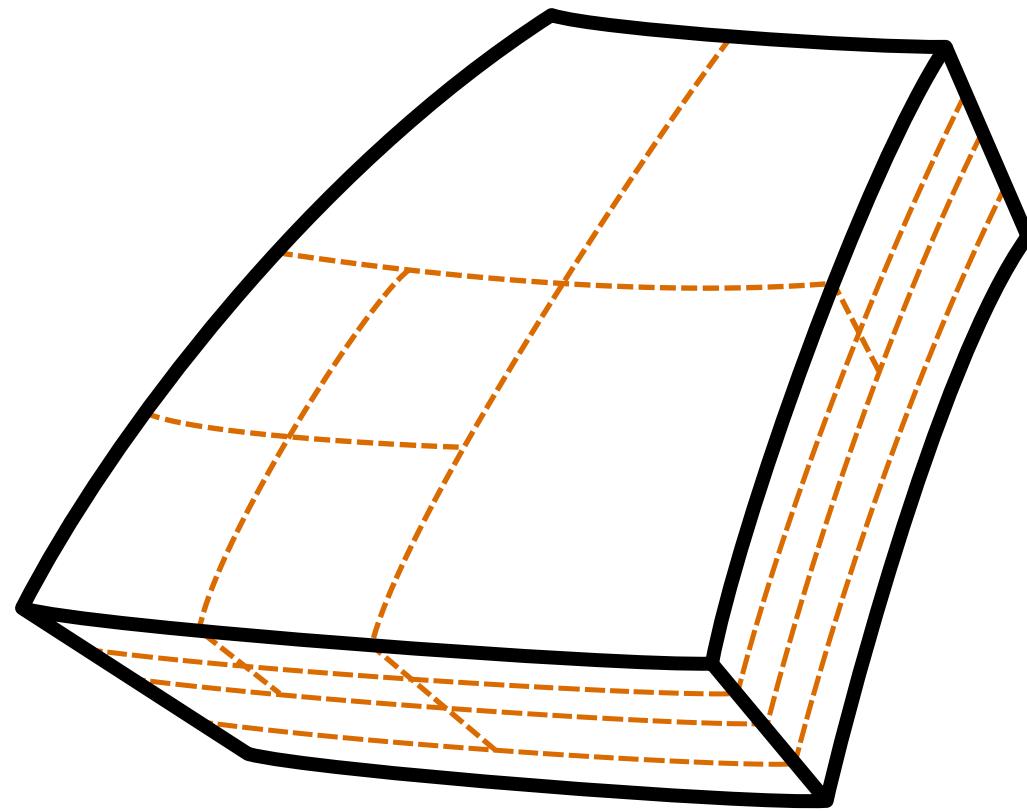
Discretización adaptativa bidimensional





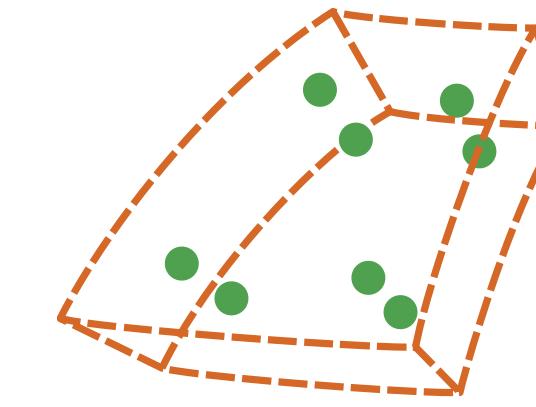






Cuadratura de Gauss-Legendre

$$V(\mathbf{p}) \cong G_A \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 W_{ijk} \frac{\rho(r_i) \kappa_{ik}}{\|\mathbf{p} - \mathbf{q}_{ijk}\|}$$



Aplicada sobre cada teseroide pequeño

Resumen del nuevo método

1. Discretización basada en densidad
2. Discretización adaptativa bidimensional
3. Cuadratura Gauss-Legendre

Precisión y tiempo de cómputo

Discretización basada en
densidad



δ

Discretización adaptativa
bidimensional



D

D y δ controlan la cantidad de subdivisiones



Subdivisiones



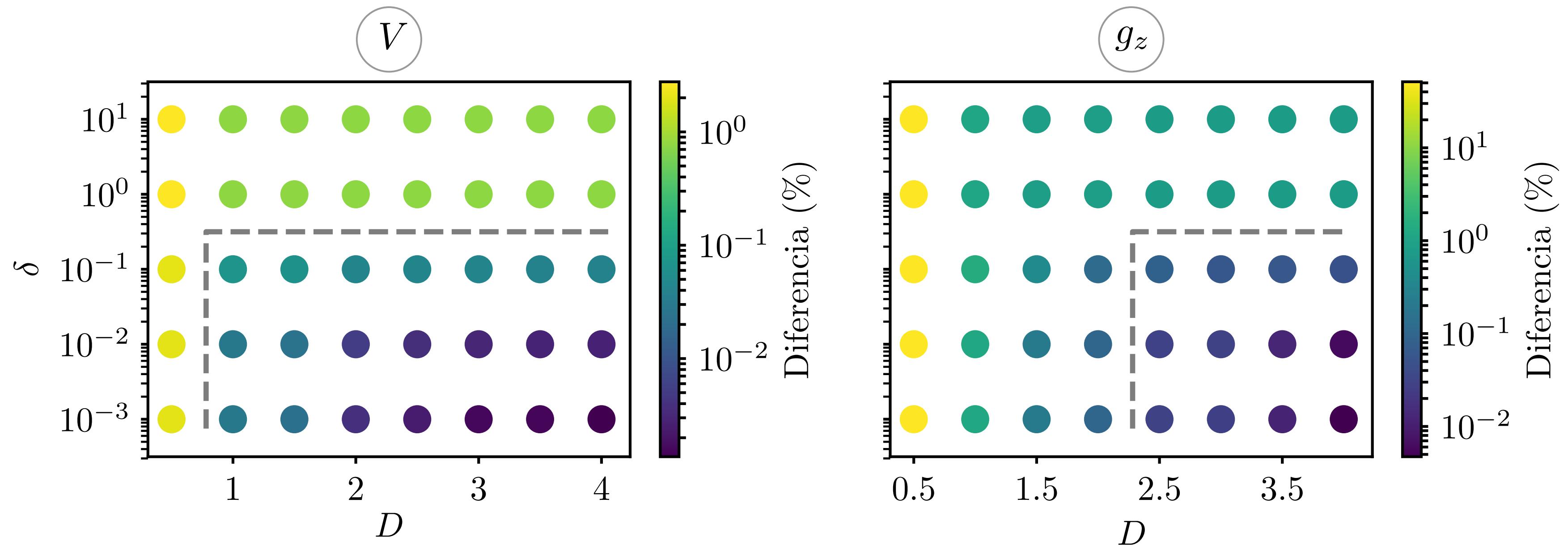
Precisión



Tiempo de cómputo

Determinación de D y δ

Comparaciones con soluciones analíticas para cascarones esféricos



Valores óptimos para D y δ :

- $D = 1$ para potencial V
- $D = 2.5$ para aceleración \mathbf{g}
- $\delta = 0.1$

 Garantizan errores por debajo del 1% 

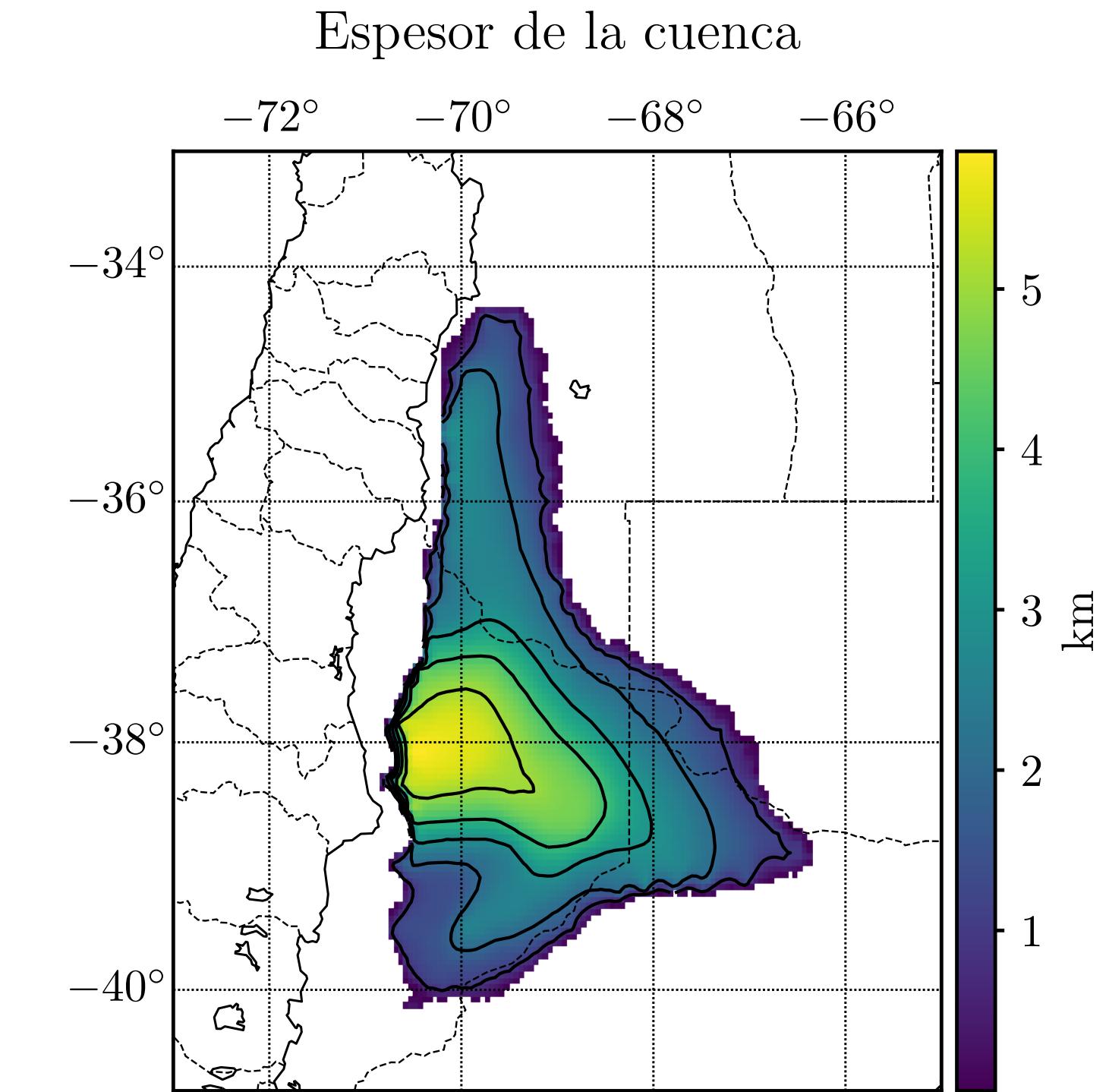
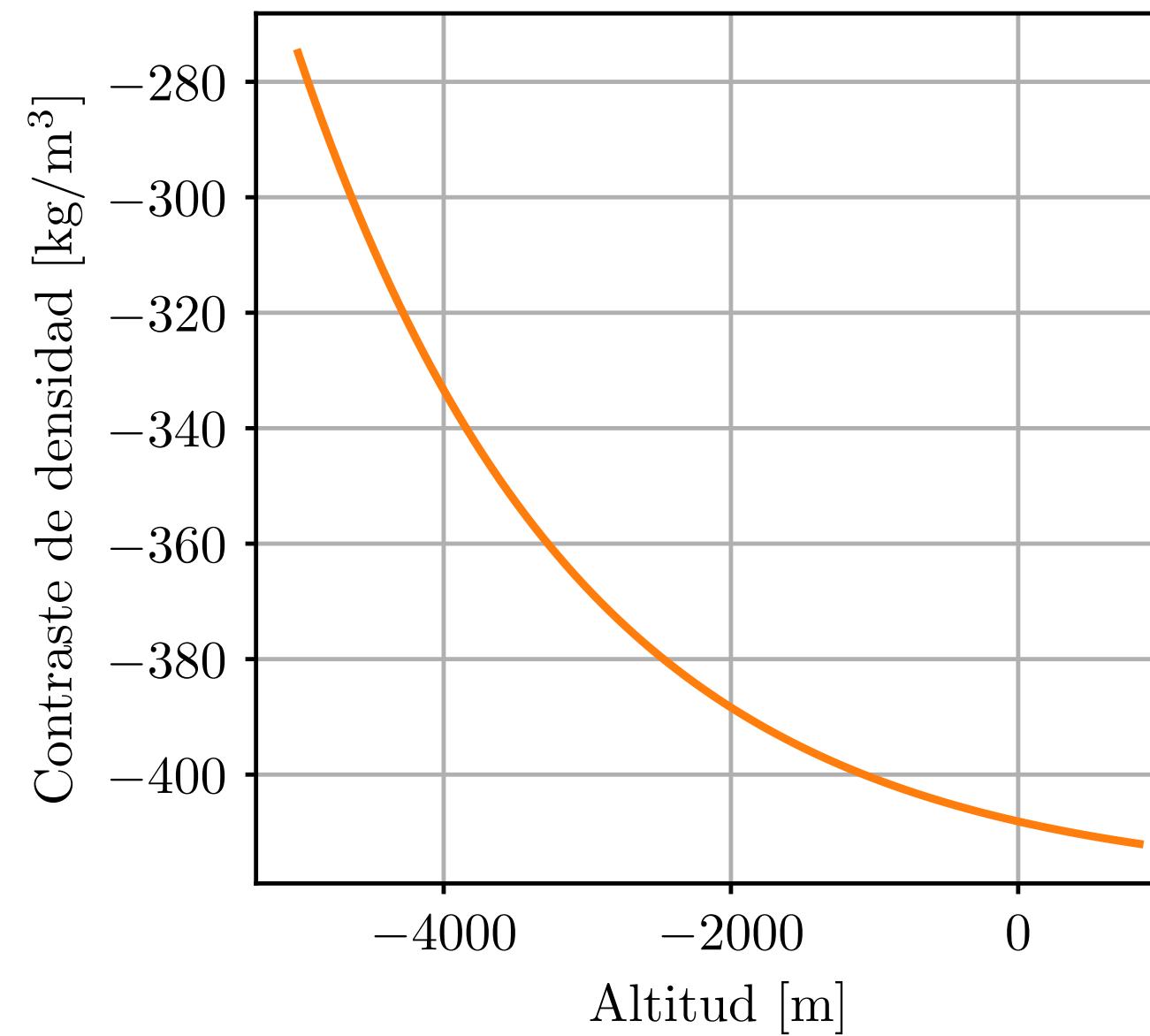
Posibles aplicaciones

Cuencas sedimentarias de gran tamaño

- Sedimentos con **compactación**
- **Curvatura del planeta** por su gran extensión

Modelo directo de la cuenca Neuquina

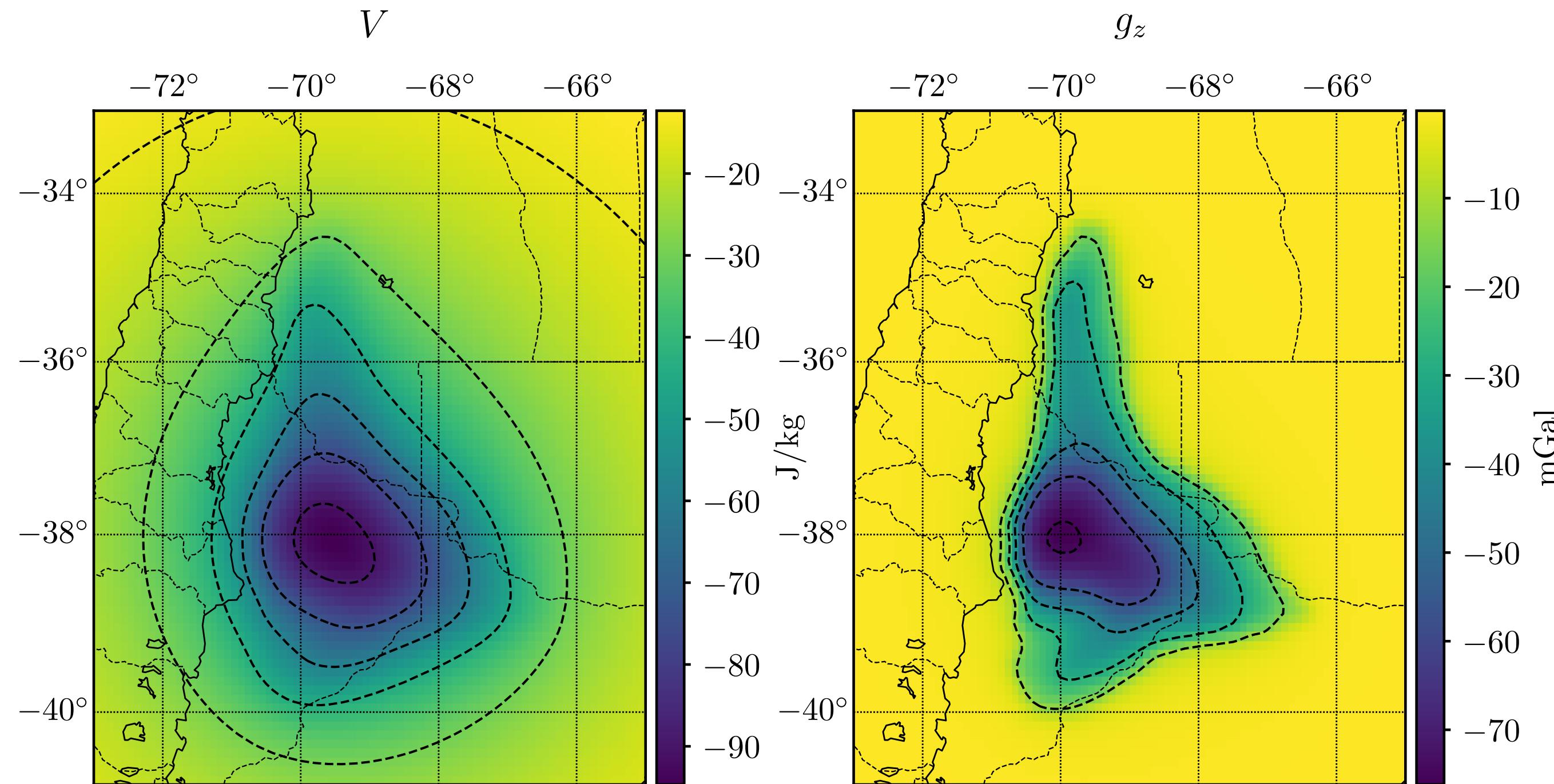
Modelo de la cuenca



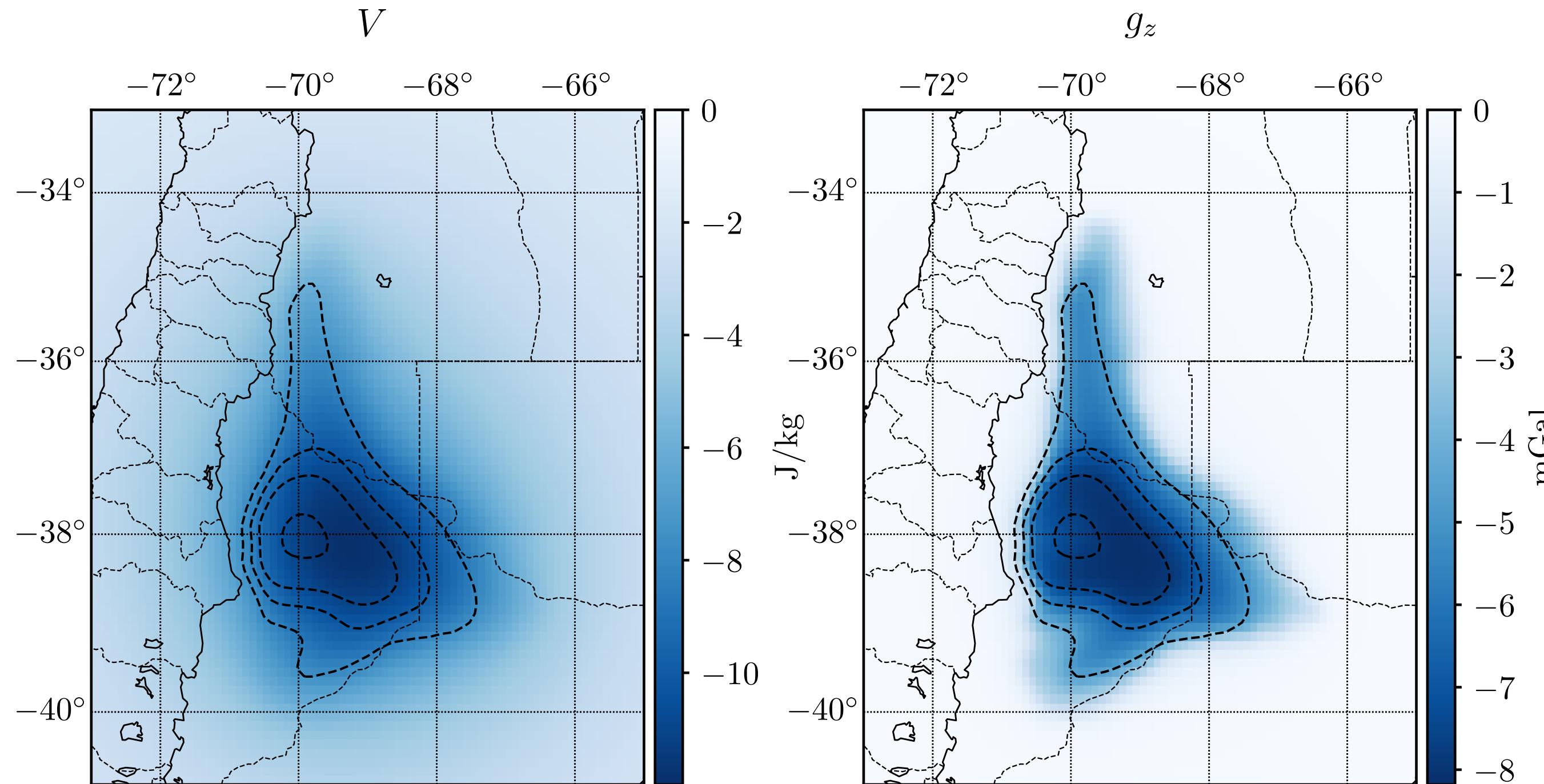
Heine, C. (2007). Formation and evolution of intracontinental basins. School of Geosciences, The University of Sydney, Australia.

Sigismondi, M. E. (2012). Estudio de la deformación litosférica de la cuenca Neuquina: estructura termal, datos de gravedad y sísmica de reflexión. FCEN, UBA.

Campos gravitatorios del modelo



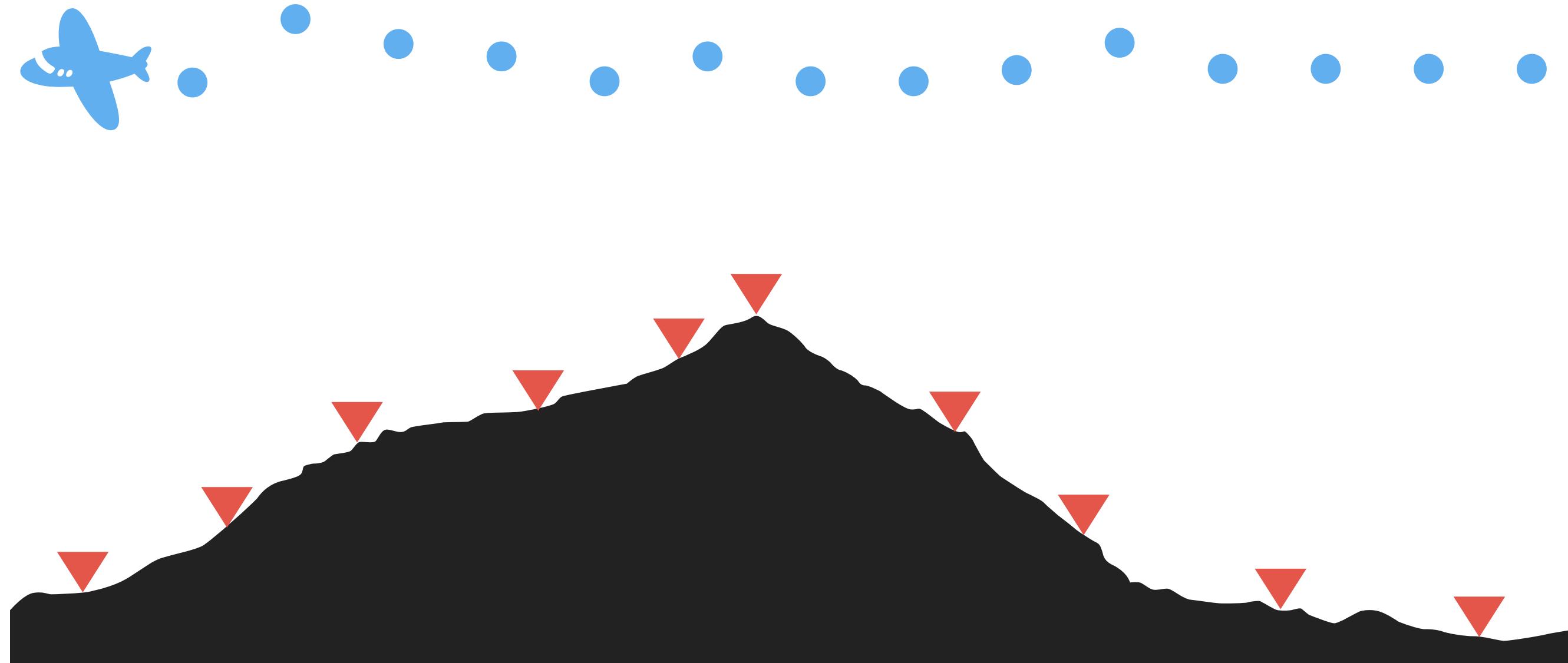
Diferencias con densidad constante



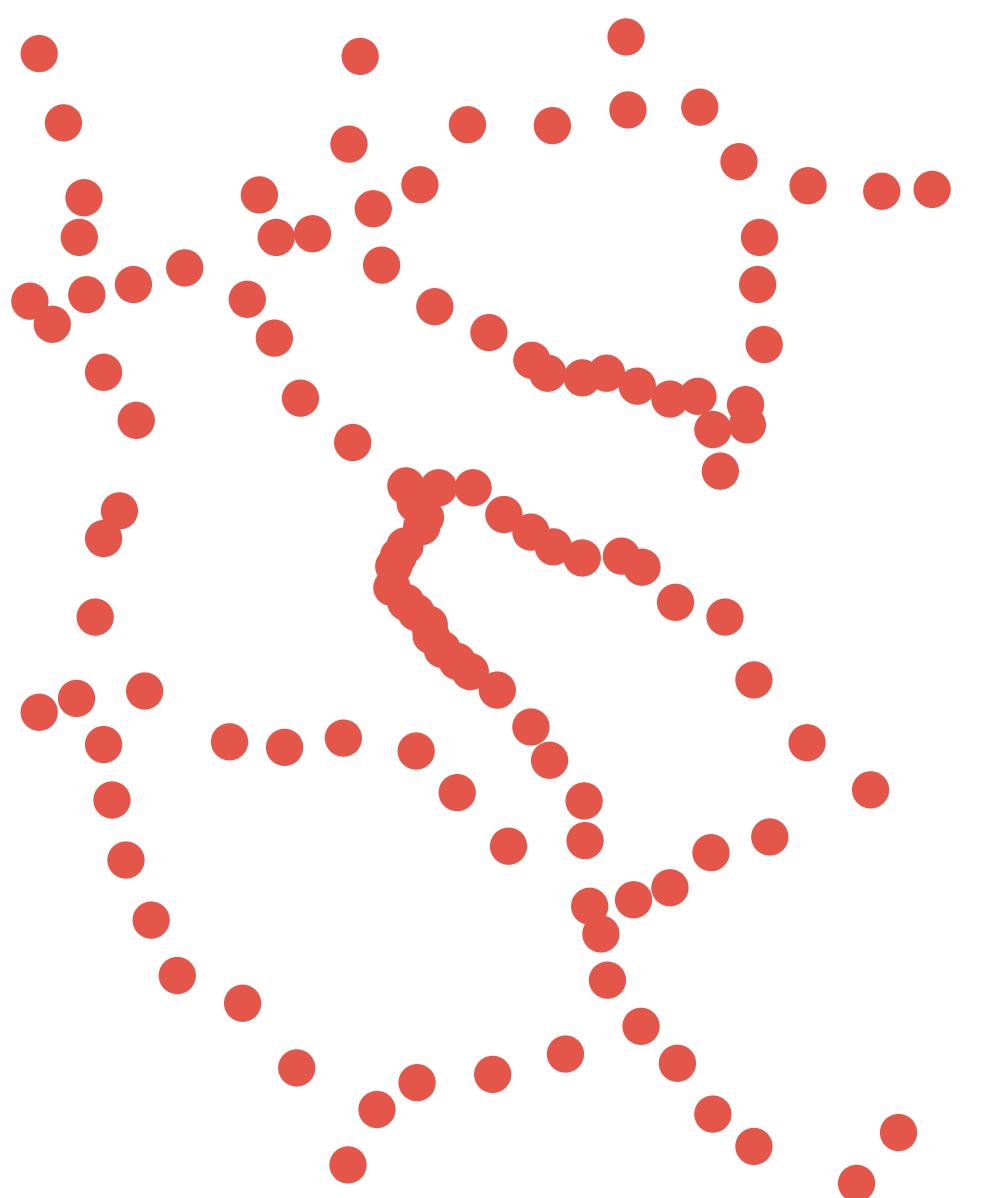
Fuentes equivalentes potenciadas por gradiente

¿Qué son las fuentes equivalentes?

Datos gravimétricos



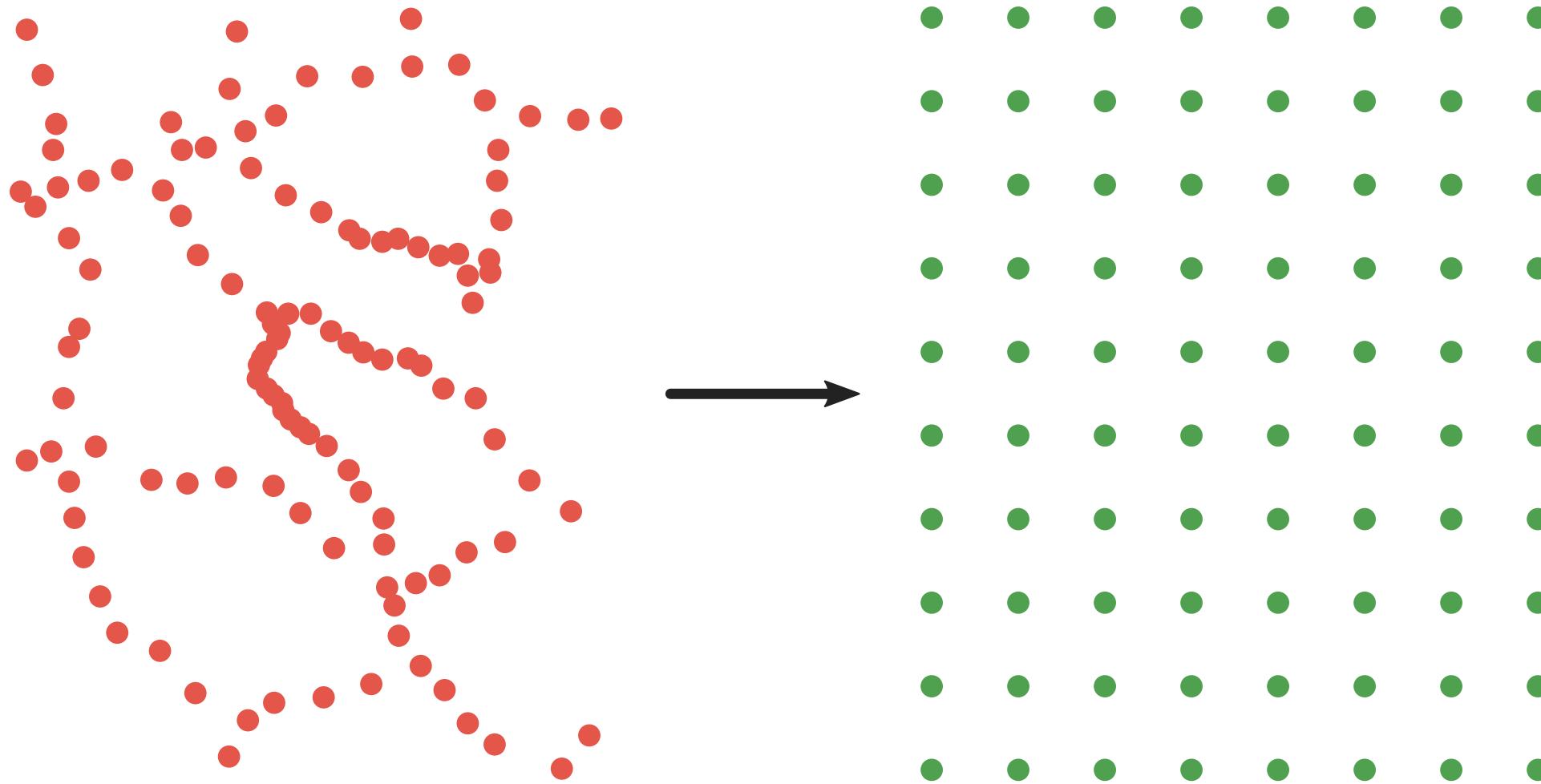
Muestras sobre terreno



Muestras aéreas

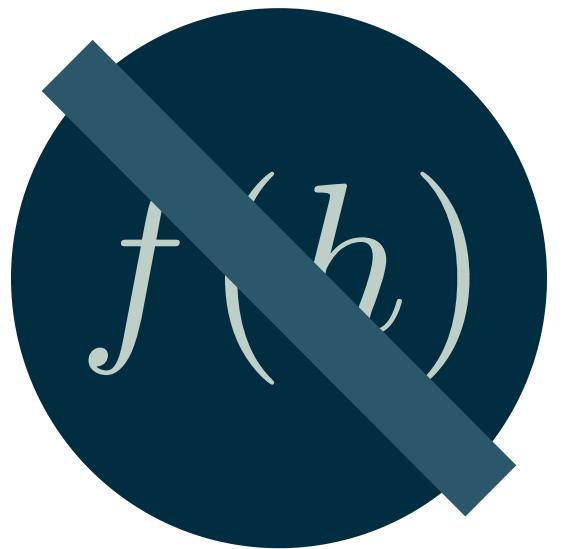


Interpolación

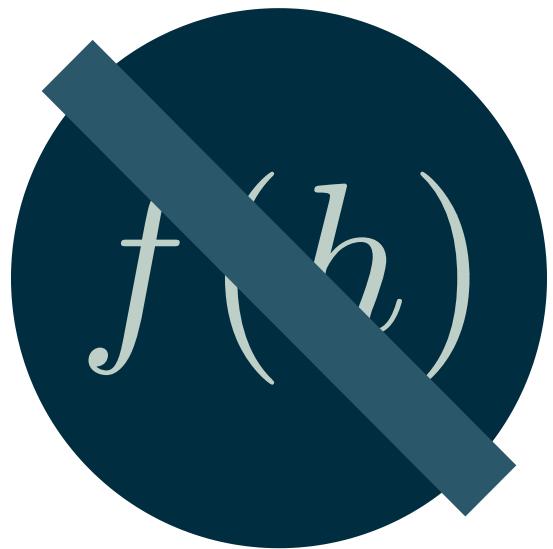


Interpoladores 2D de propósito general

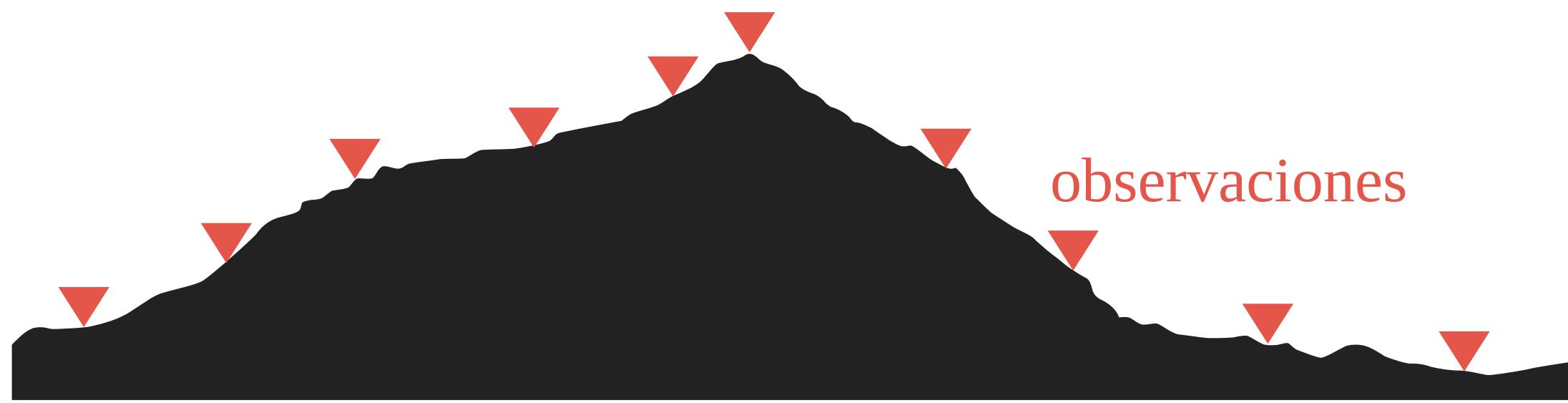
Interpoladores 2D de propósito general

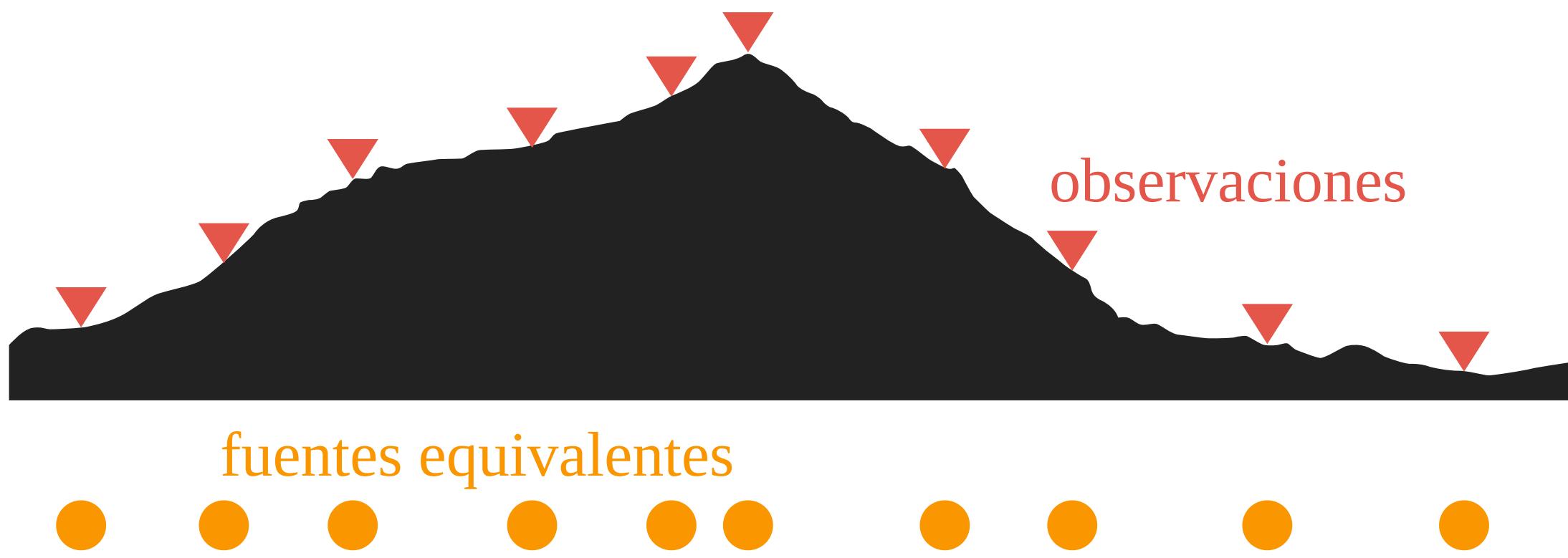


Interpoladores 2D de propósito general



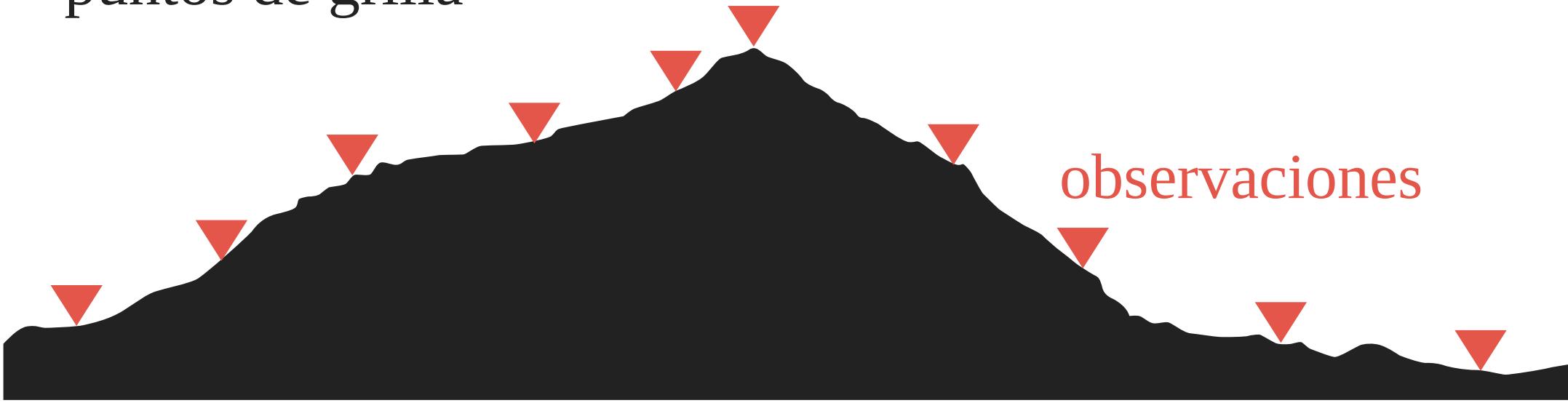
Fuentes equivalentes







puntos de grilla



observaciones

fuentes equivalentes

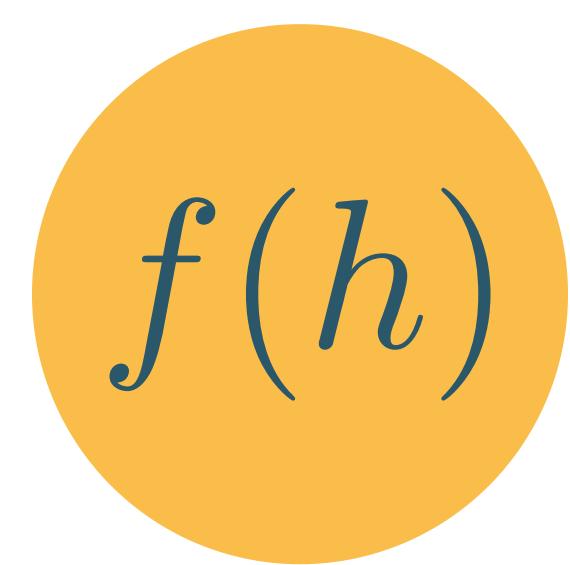


Ventajas

Ventajas

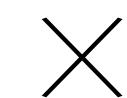
$$f(h)$$

Ventajas



Problema

200 000 datos

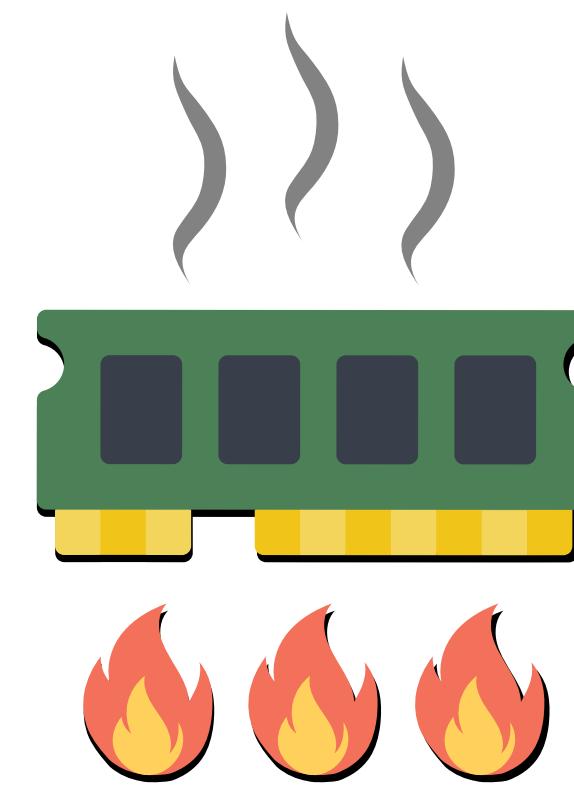


200 000 fuentes

Problema

$$\begin{array}{r} 200\,000 \text{ datos} \\ \times \\ 200\,000 \text{ fuentes} \\ \hline \end{array}$$

300 GB RAM



Solución

Fuentes equivalentes potenciadas por gradiente

Potenciación del gradiente

$$\mathbf{d} = \mathbf{A}\mathbf{c}$$

Potenciación del gradiente

$$\mathbf{d} = \mathbf{A}\mathbf{c} \quad \rightarrow \quad \mathbf{d} = \sum_{k=1}^K \mathbf{A}_k \mathbf{c}_k$$

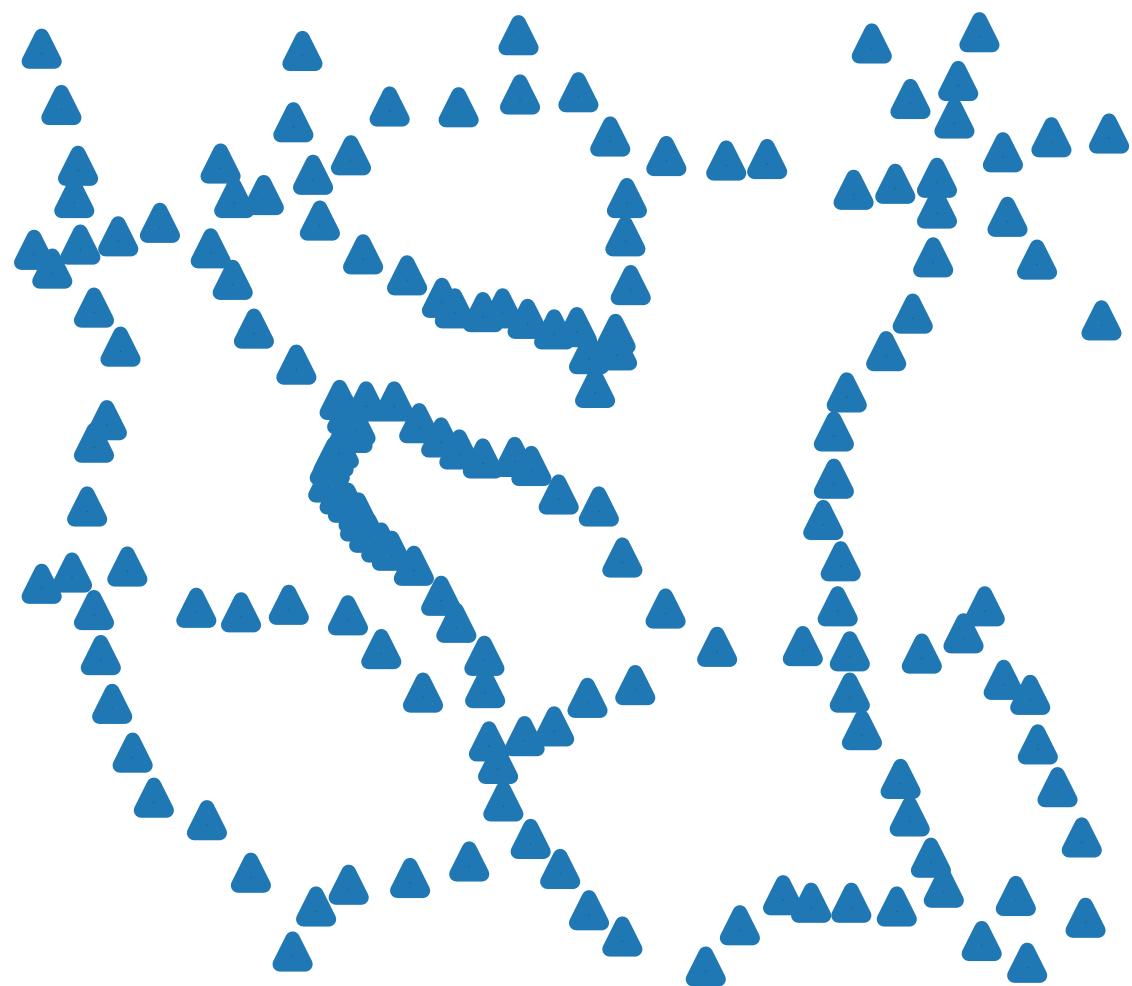
Potenciación del gradiente

$$\mathbf{d} = \mathbf{A}\mathbf{c} \quad \rightarrow \quad \mathbf{d} = \sum_{k=1}^K \mathbf{A}_k \mathbf{c}_k$$

Resolver ensambles de *predictores débiles* de forma aditiva

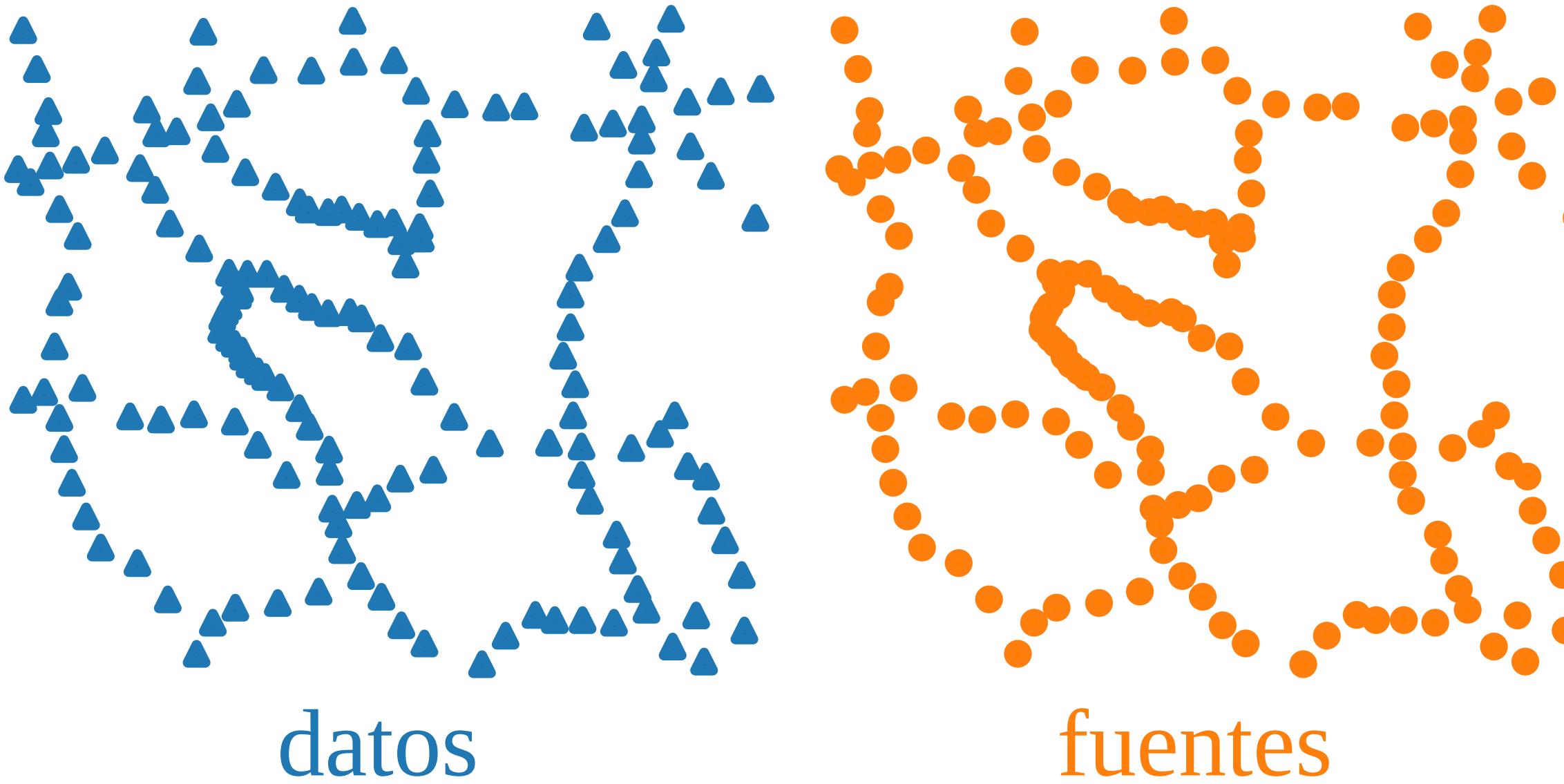
Fuentes equivalentes potenciadas por gradiente

Datos observados

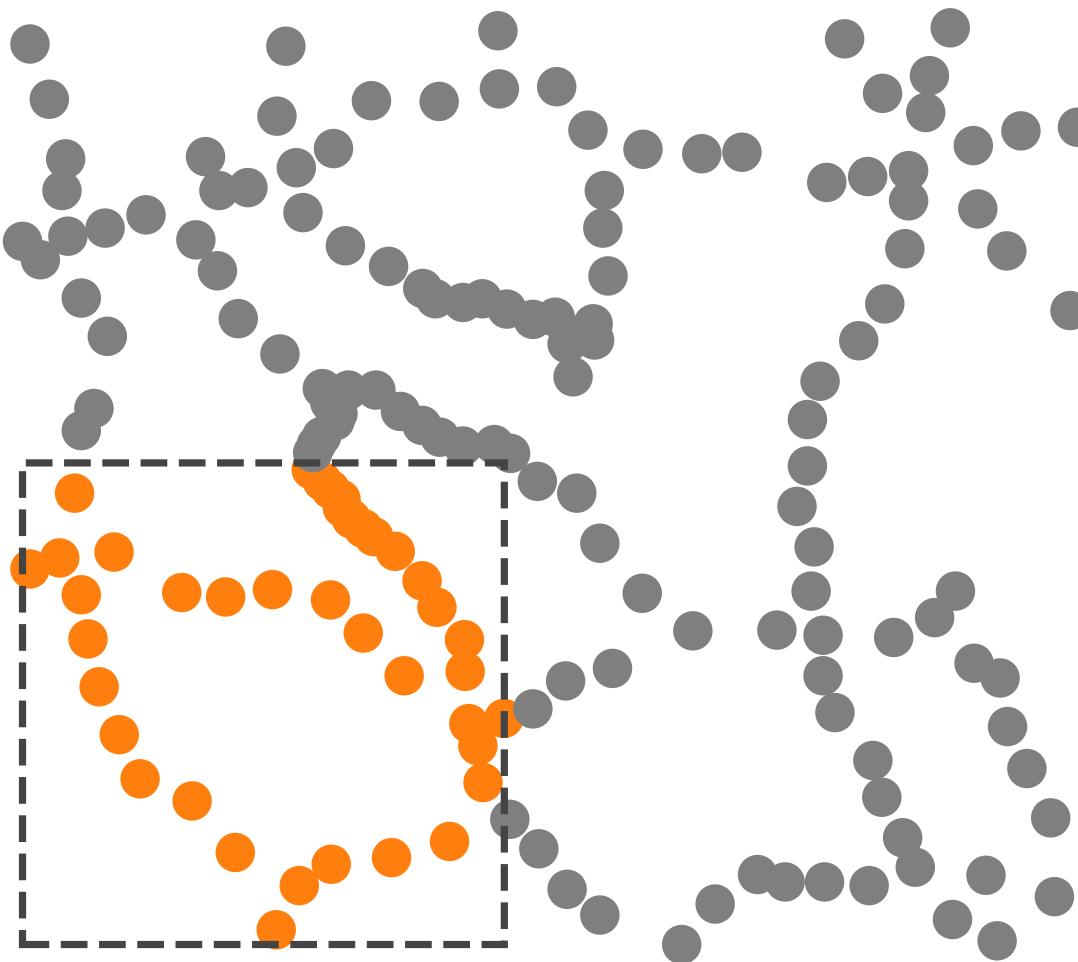


datos

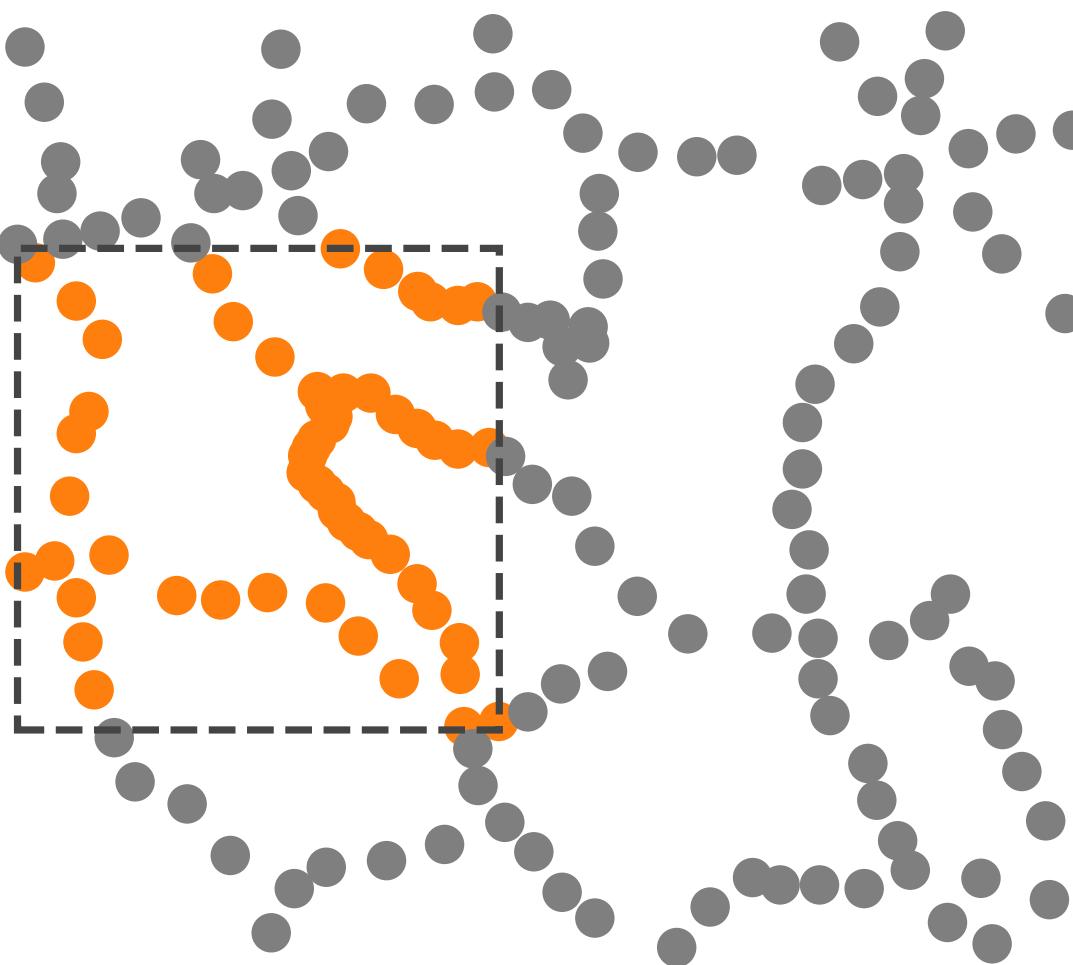
Defino fuentes



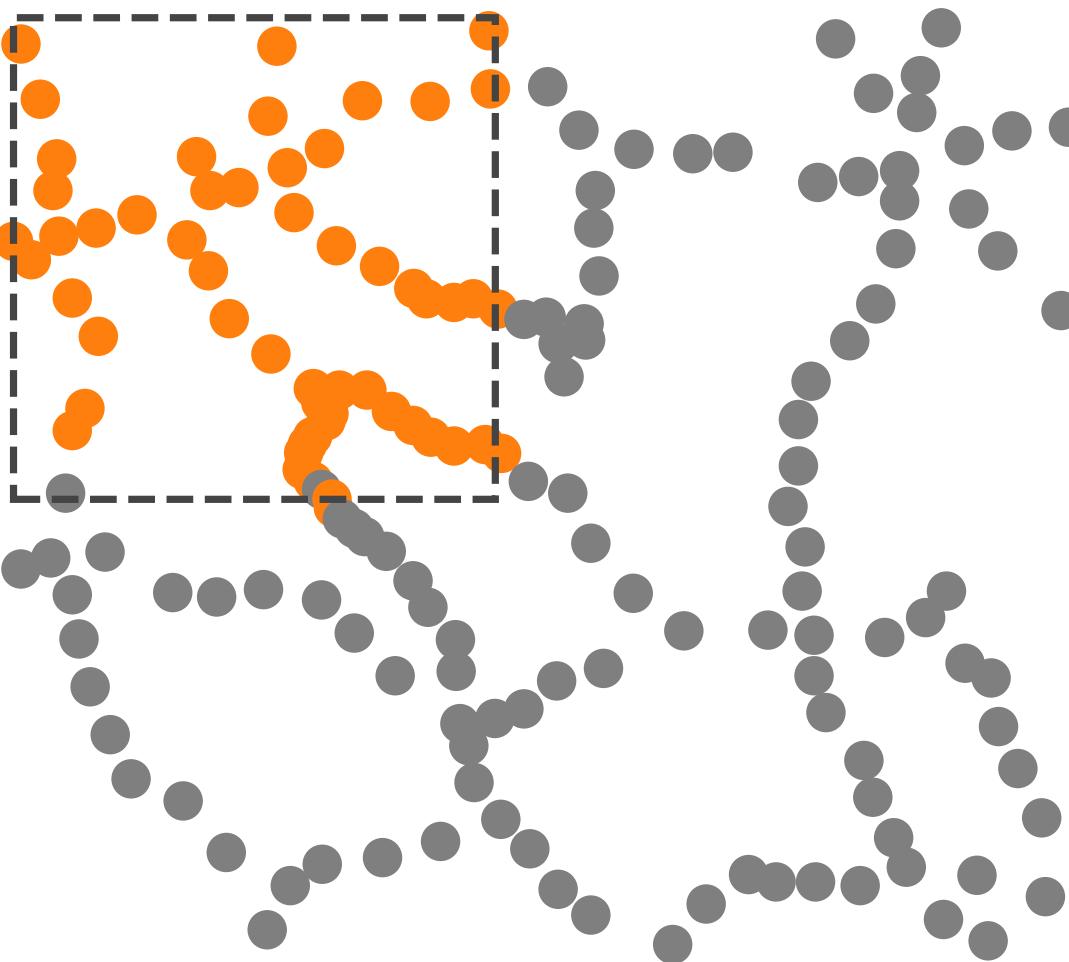
Ventanas solapadas



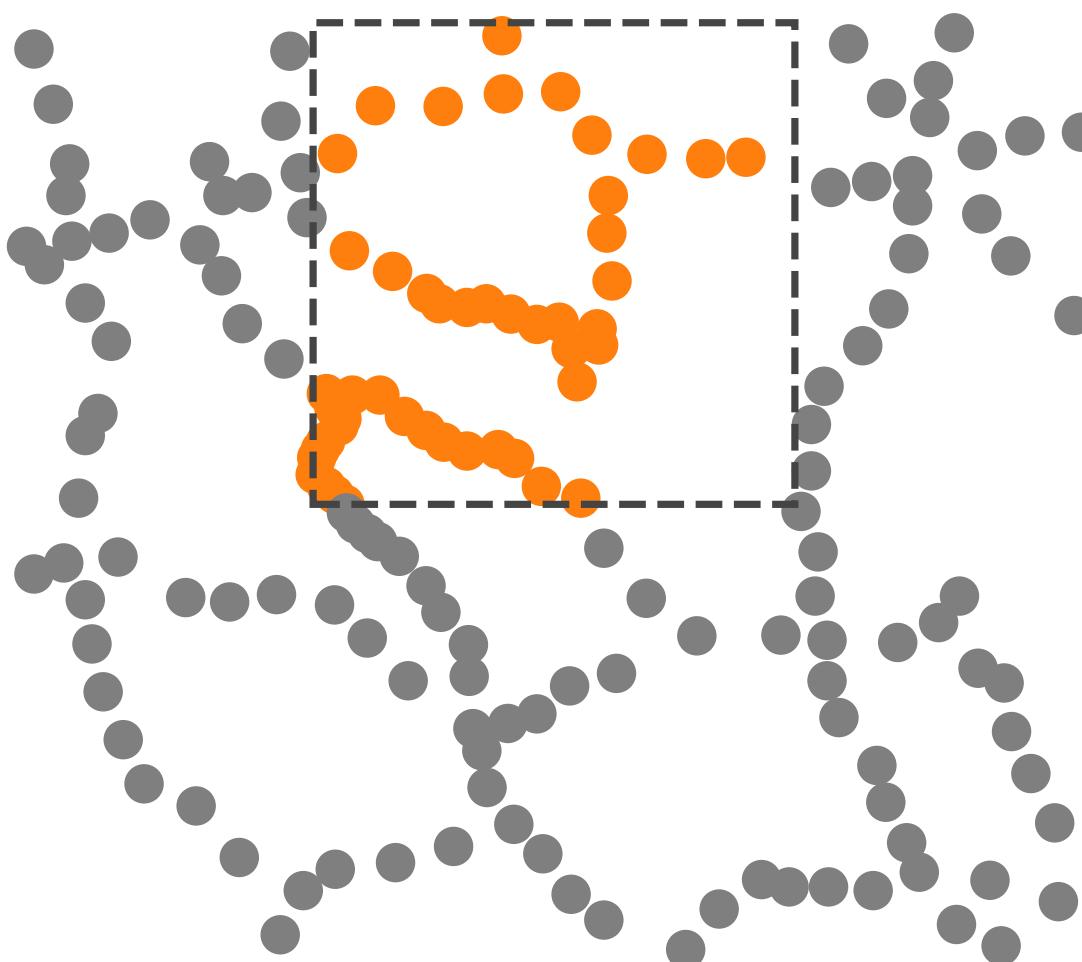
Ventanas solapadas



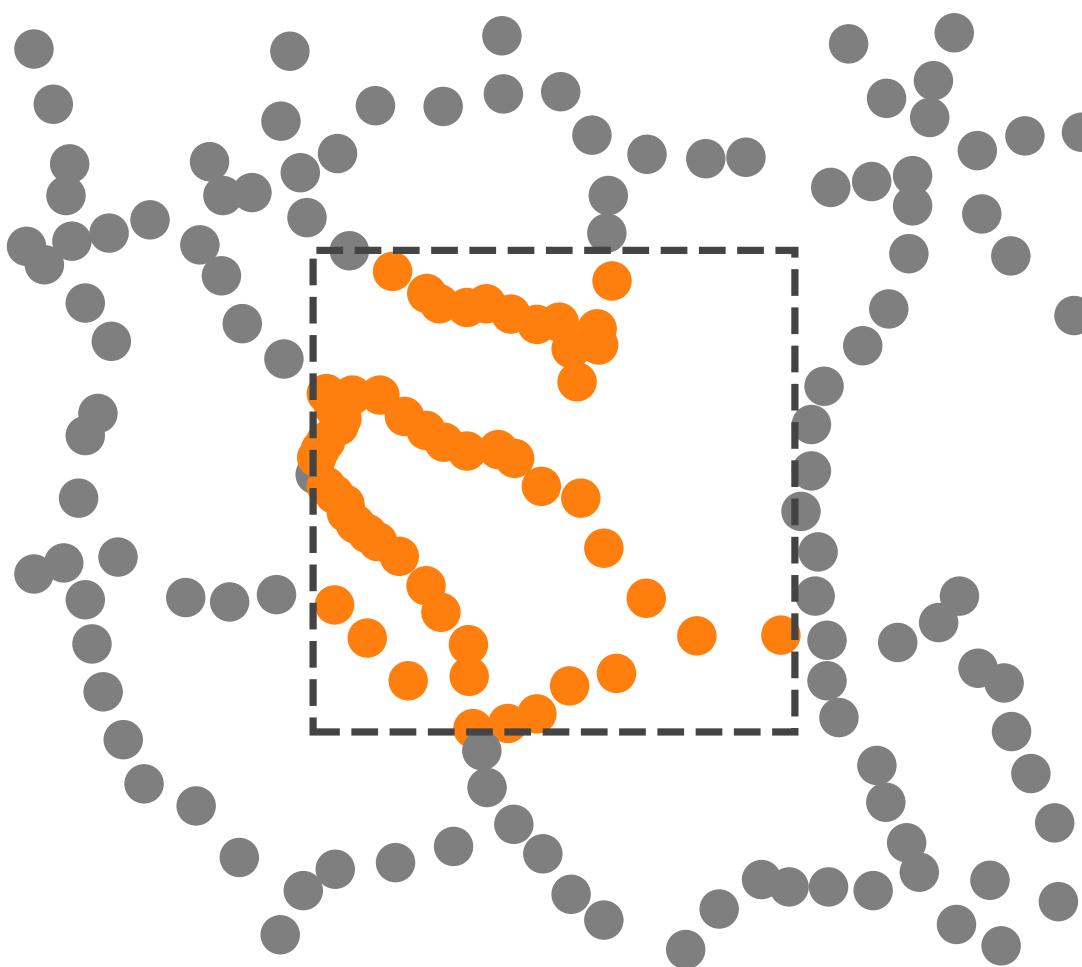
Ventanas solapadas



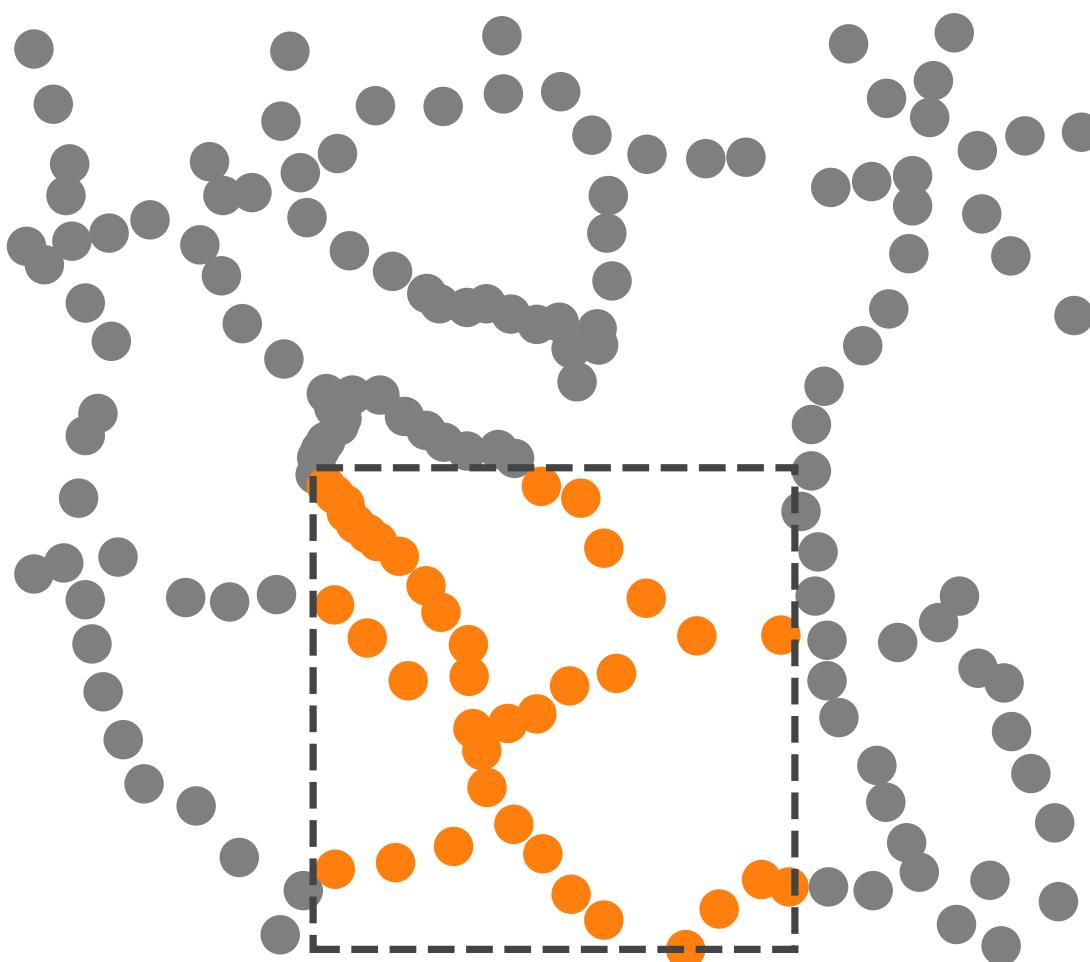
Ventanas solapadas



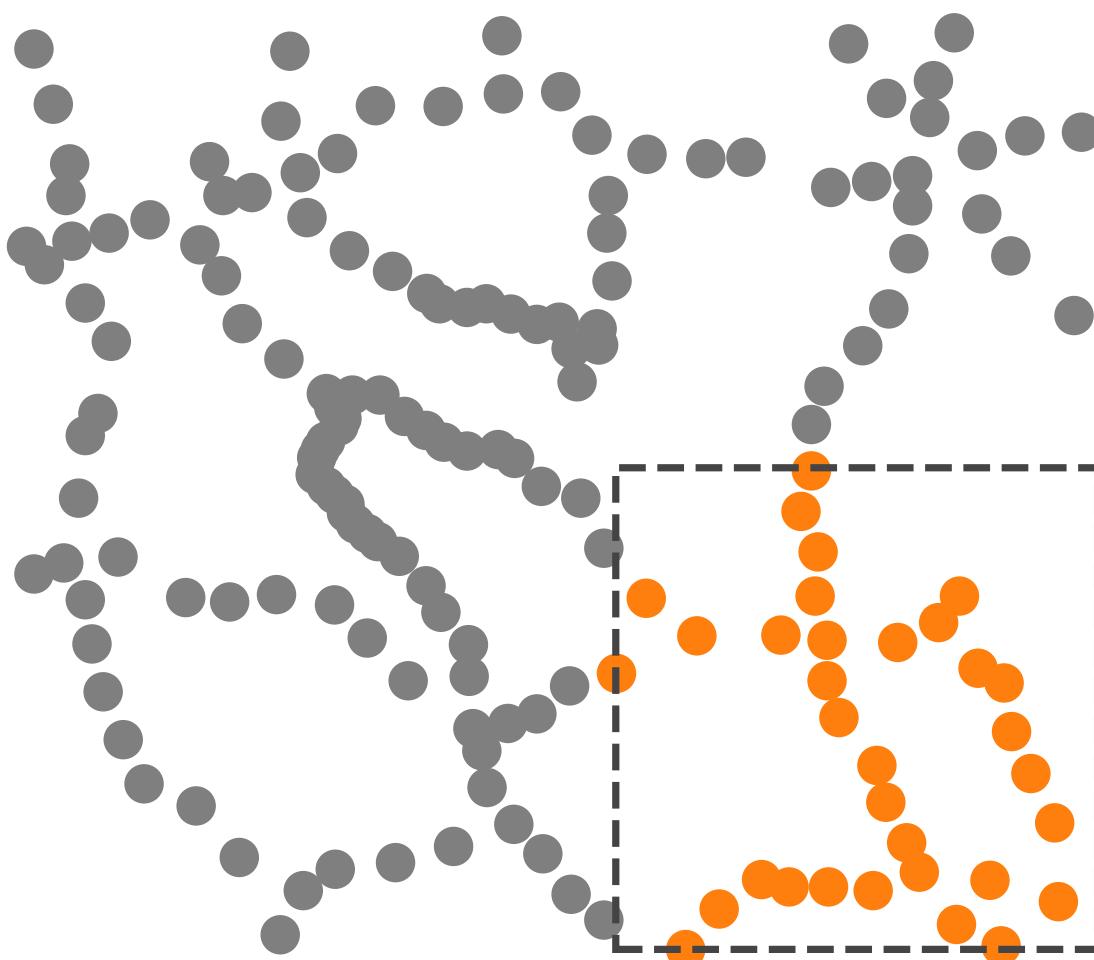
Ventanas solapadas



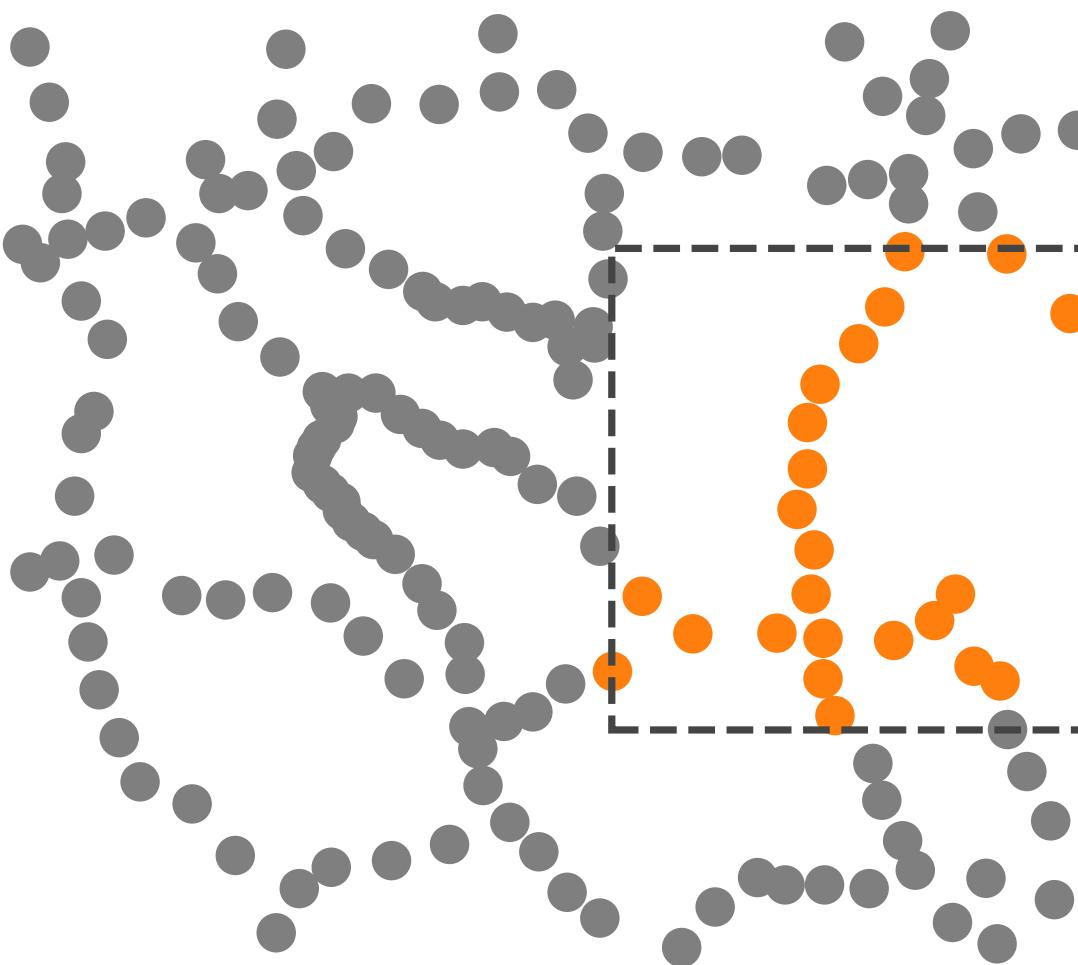
Ventanas solapadas



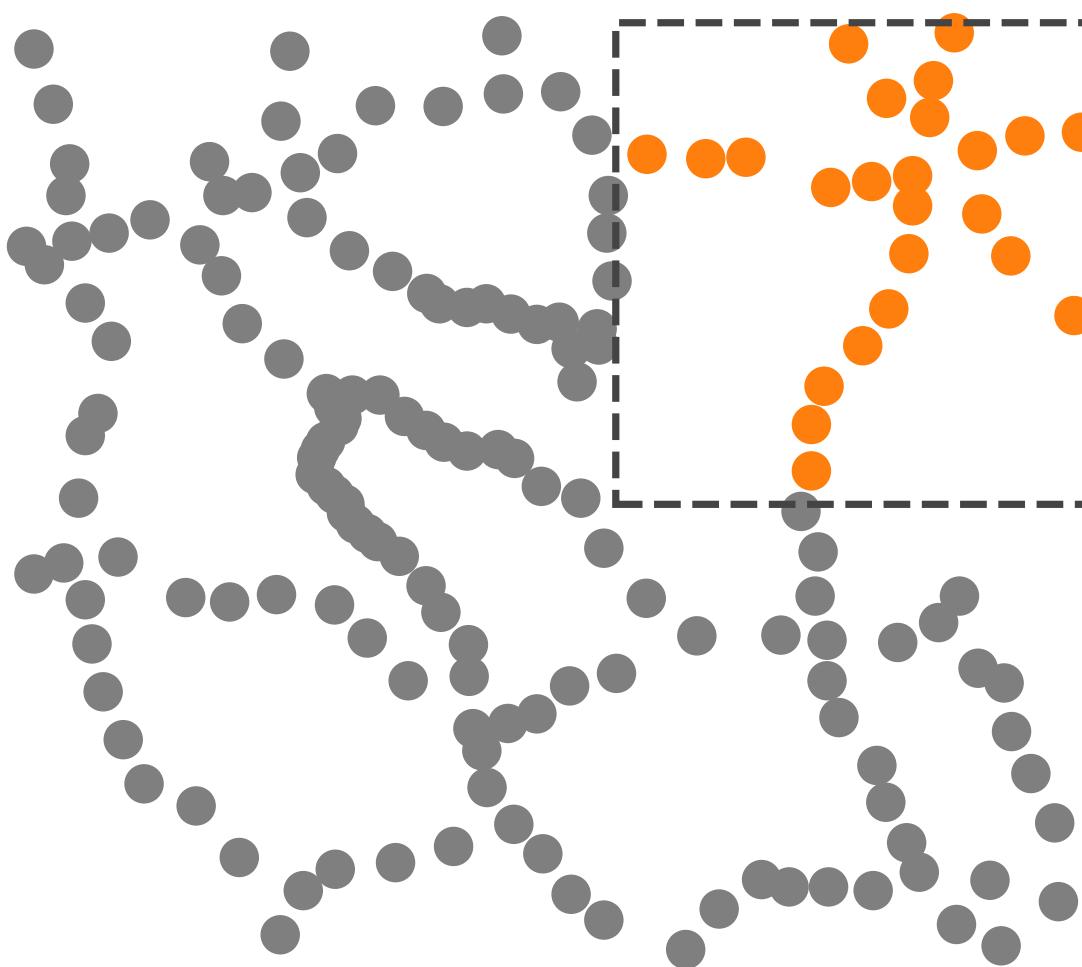
Ventanas solapadas



Ventanas solapadas

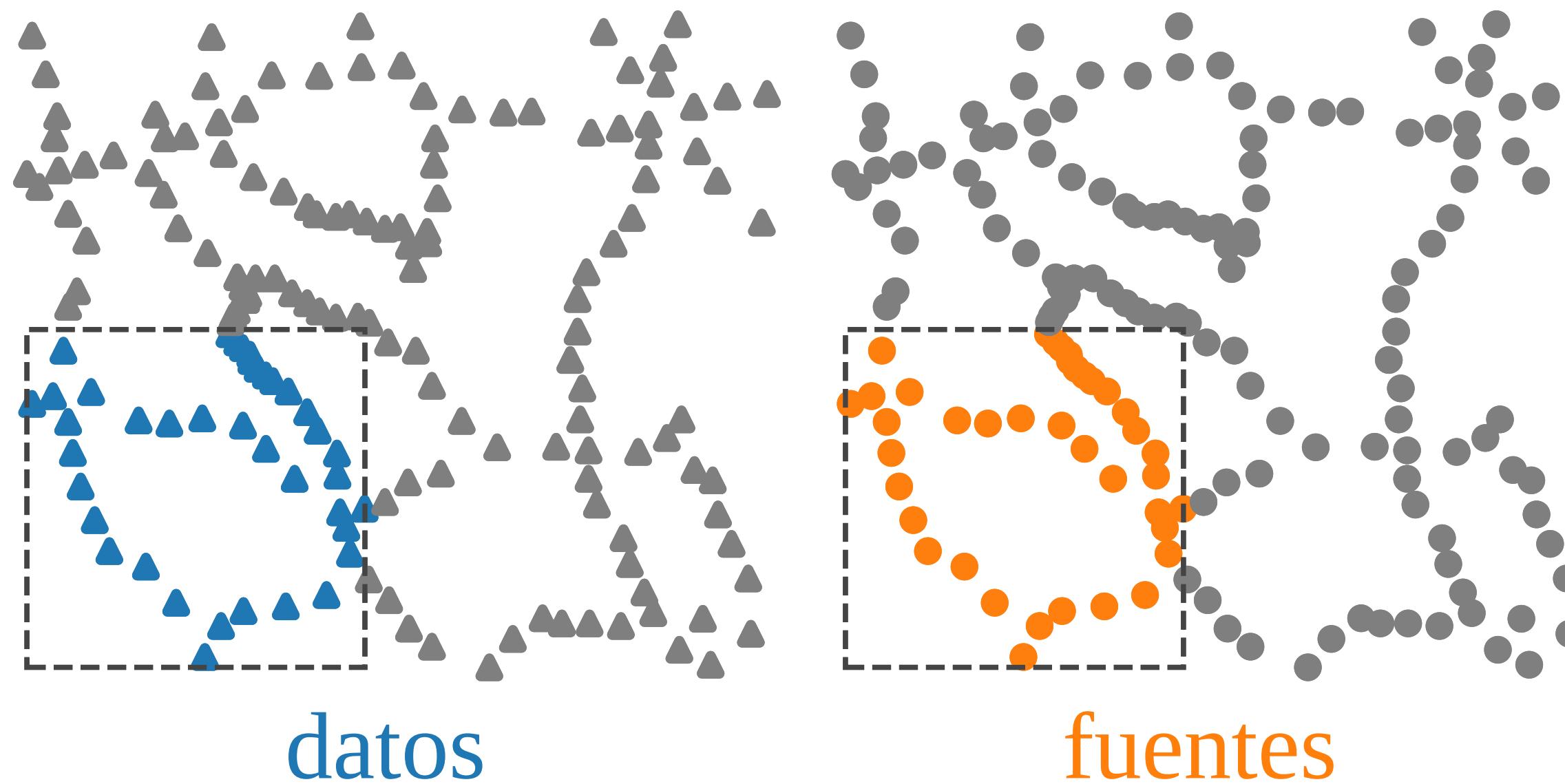


Ventanas solapadas

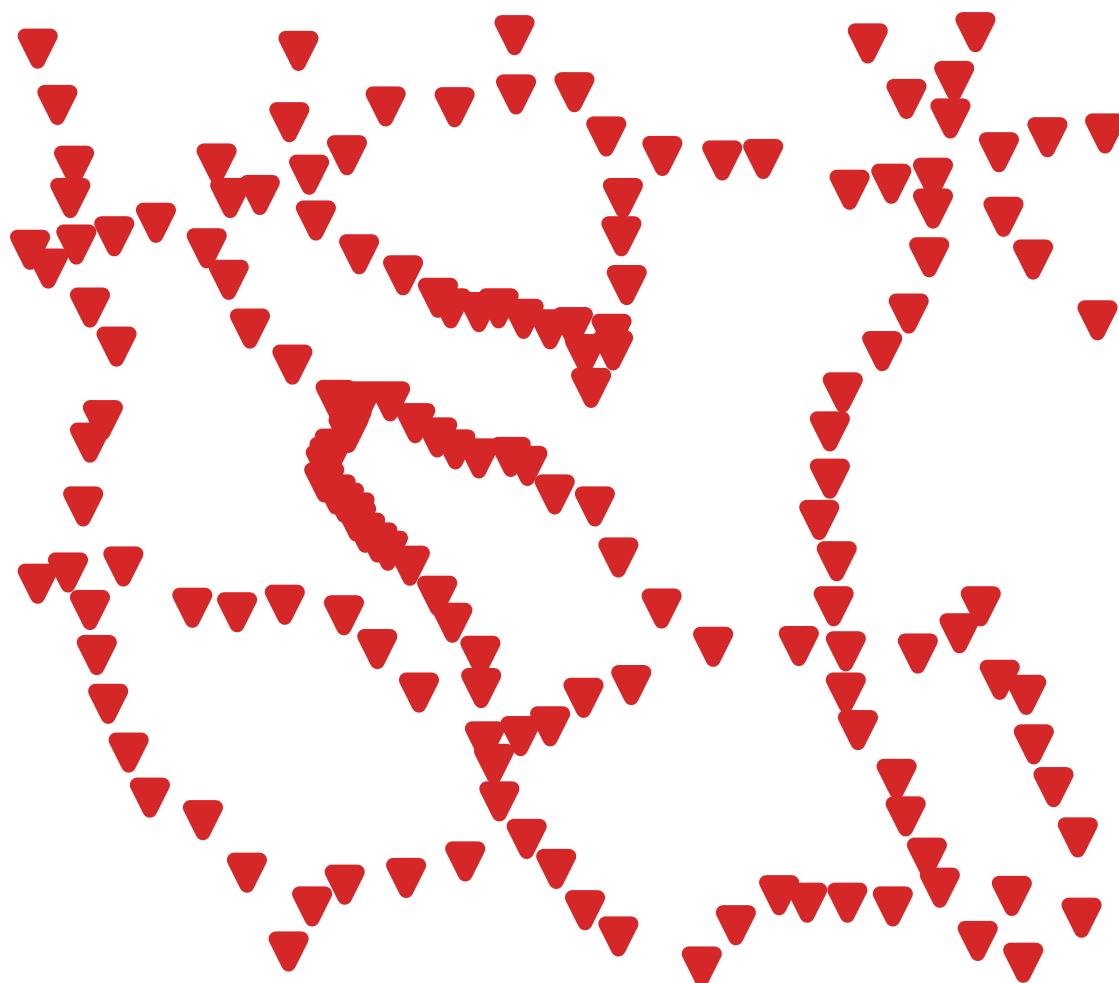


Ajustamos fuentes equivalentes iterativamente

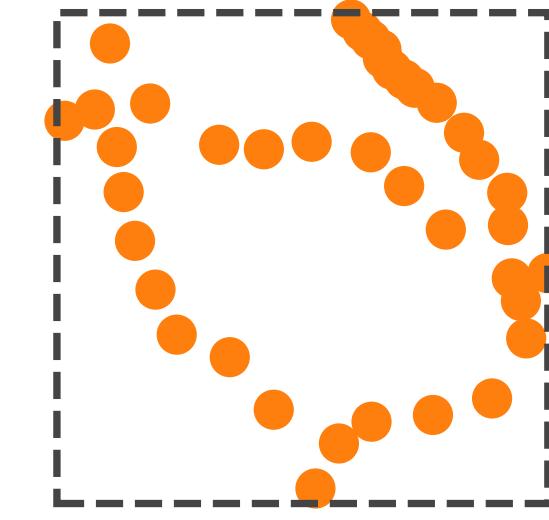
Ajuste de fuentes de la primer ventana



Cálculo del efecto de las fuentes y definición de residuos

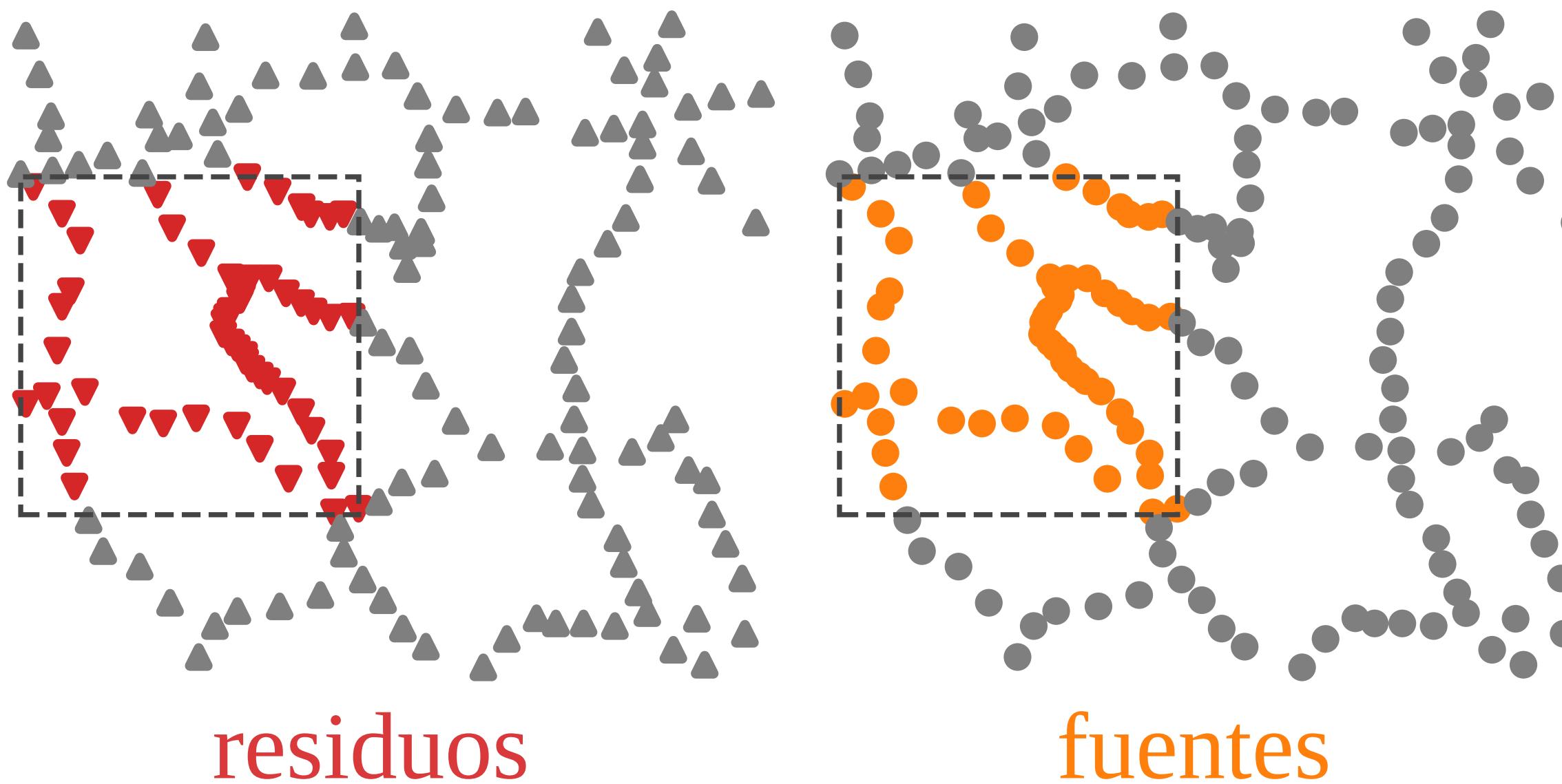


residuos

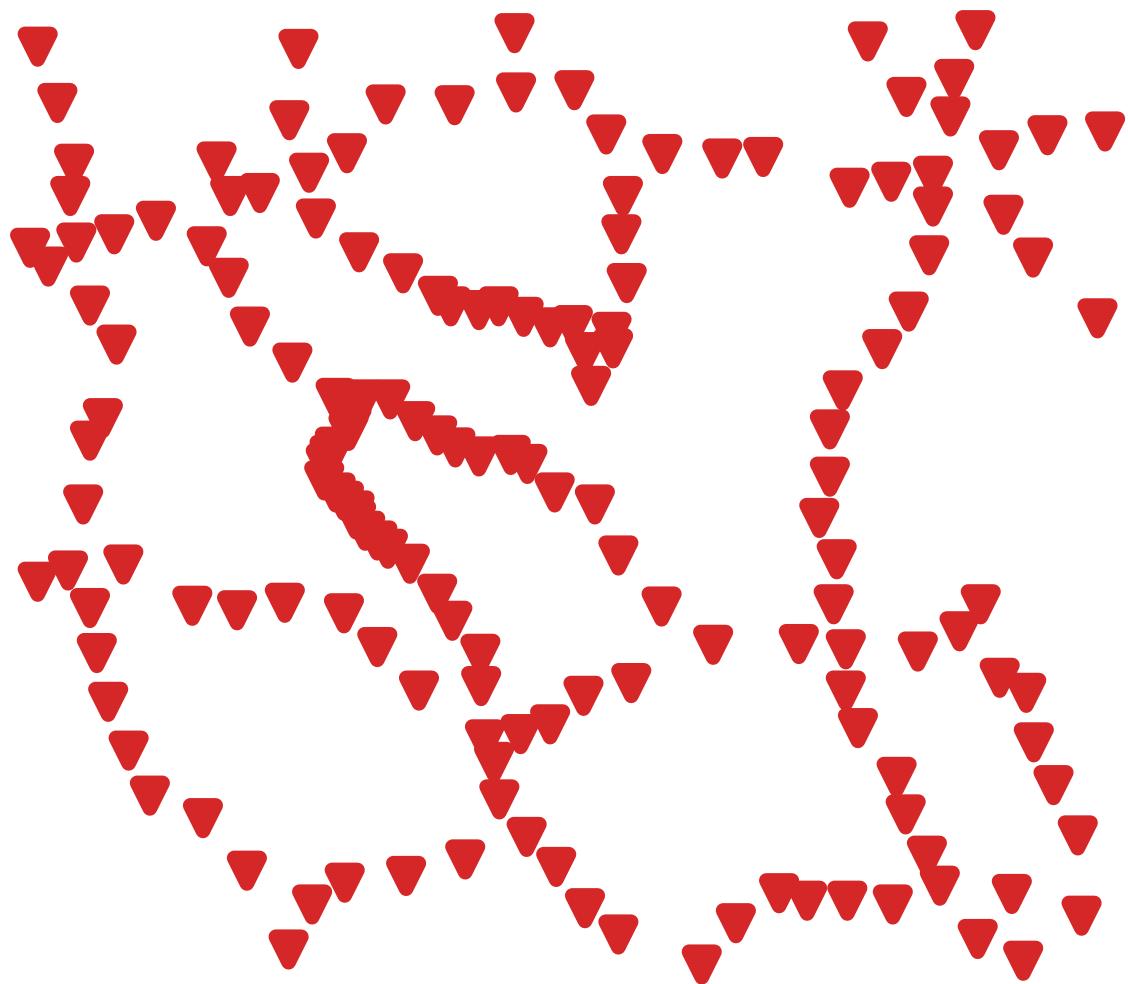


fuentes

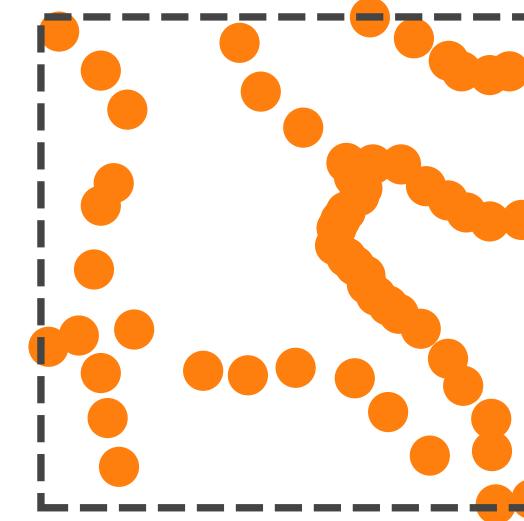
Ajuste de fuentes de la segunda ventana



Cálculo del efecto de las fuentes y actualización de residuos



residuos



fuentes

Repetimos para todas las ventanas

Características

Características

- Superposición de ventanas: **autocorrección**

Características

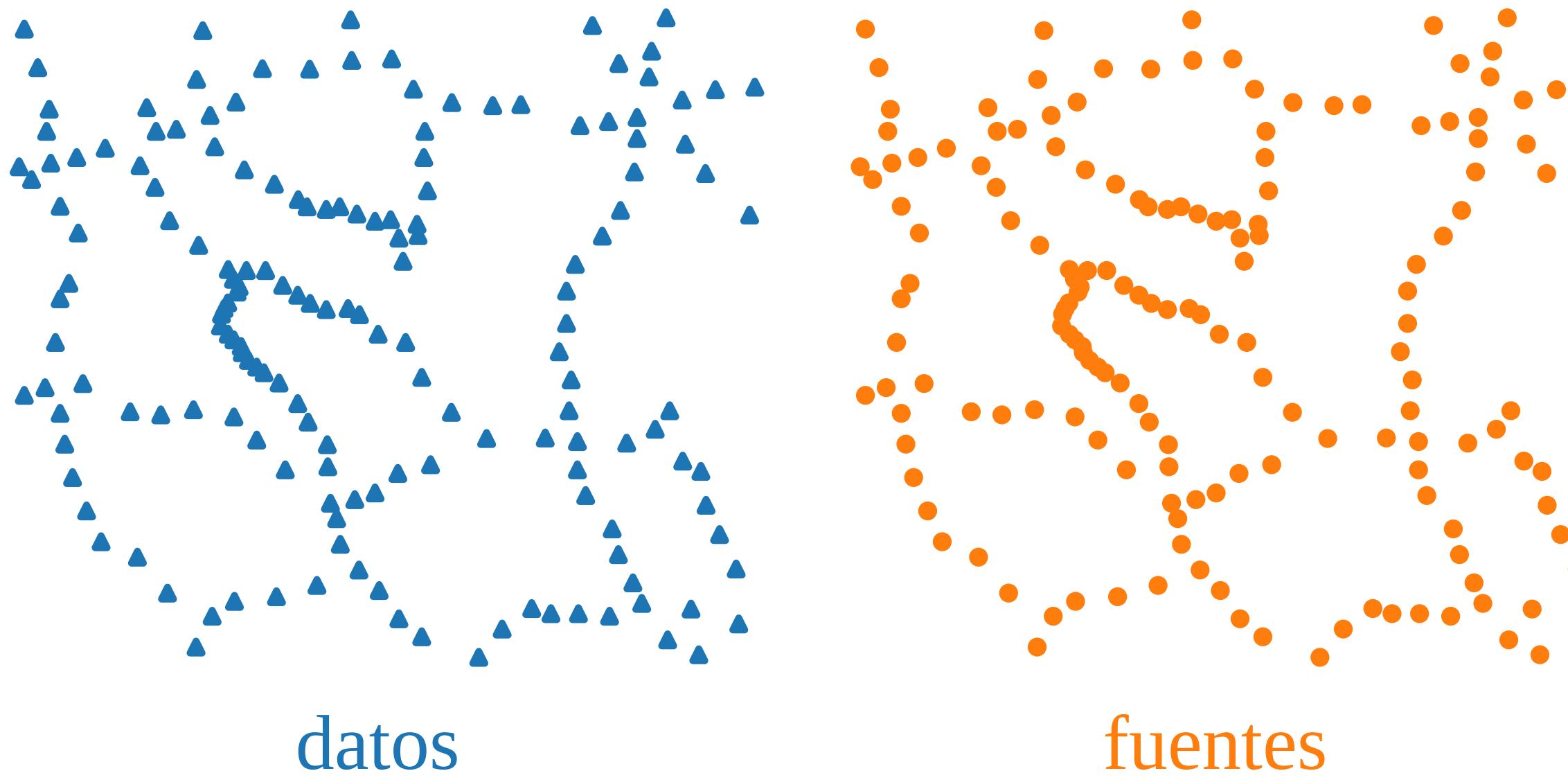
- Superposición de ventanas: **autocorrección**
- Ajuste sobre **datos en ventana**

Características

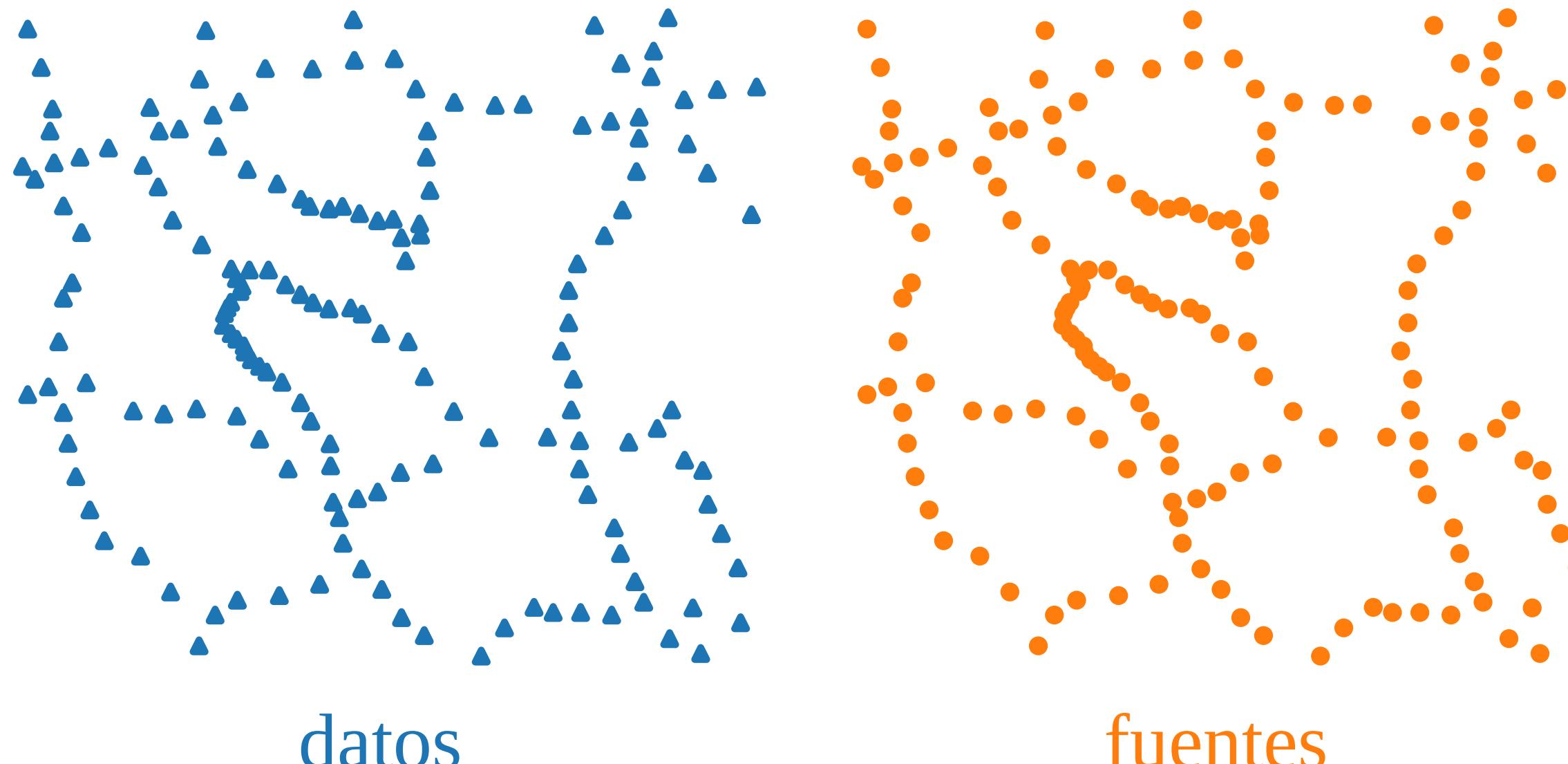
- Superposición de ventanas: **autocorrección**
- Ajuste sobre **datos en ventana**
- **Predicciones libres**

Ubicación de las fuentes

Fuentes debajo de datos

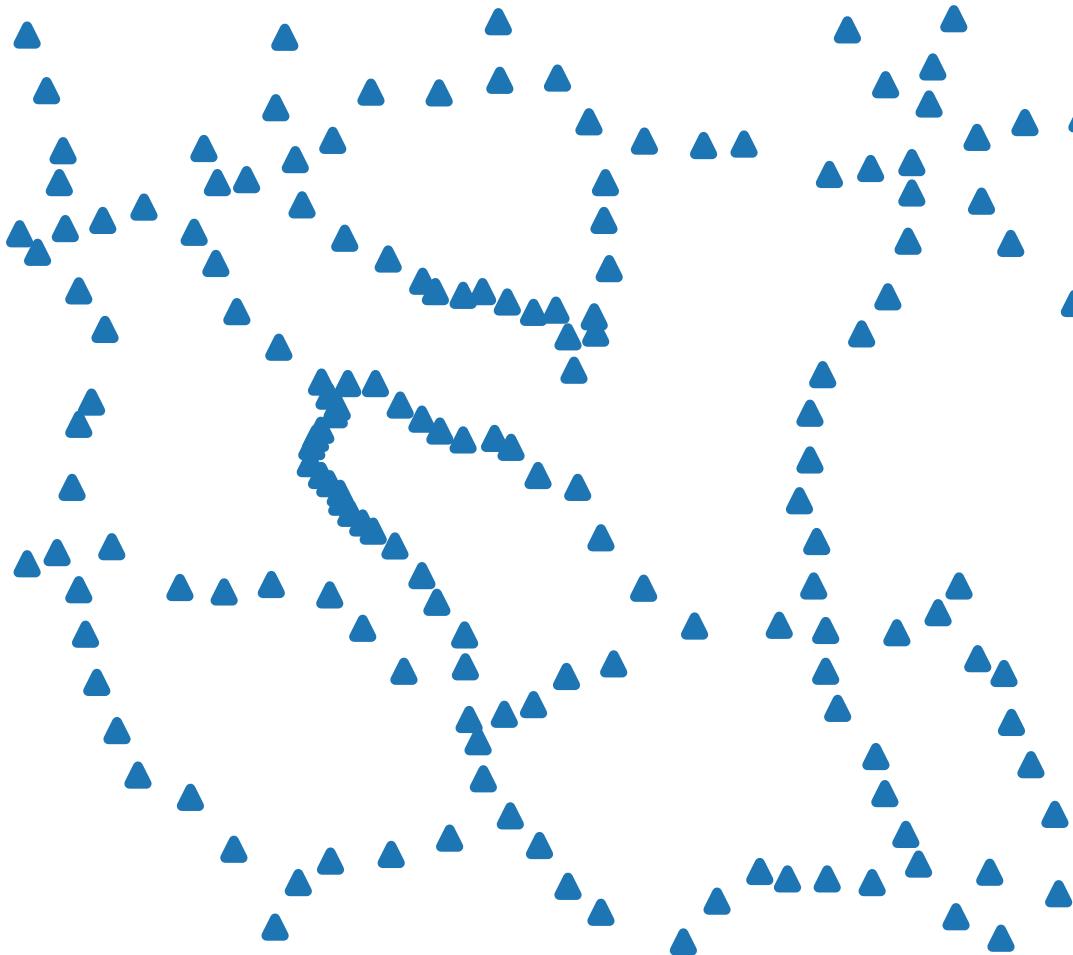


Fuentes debajo de datos

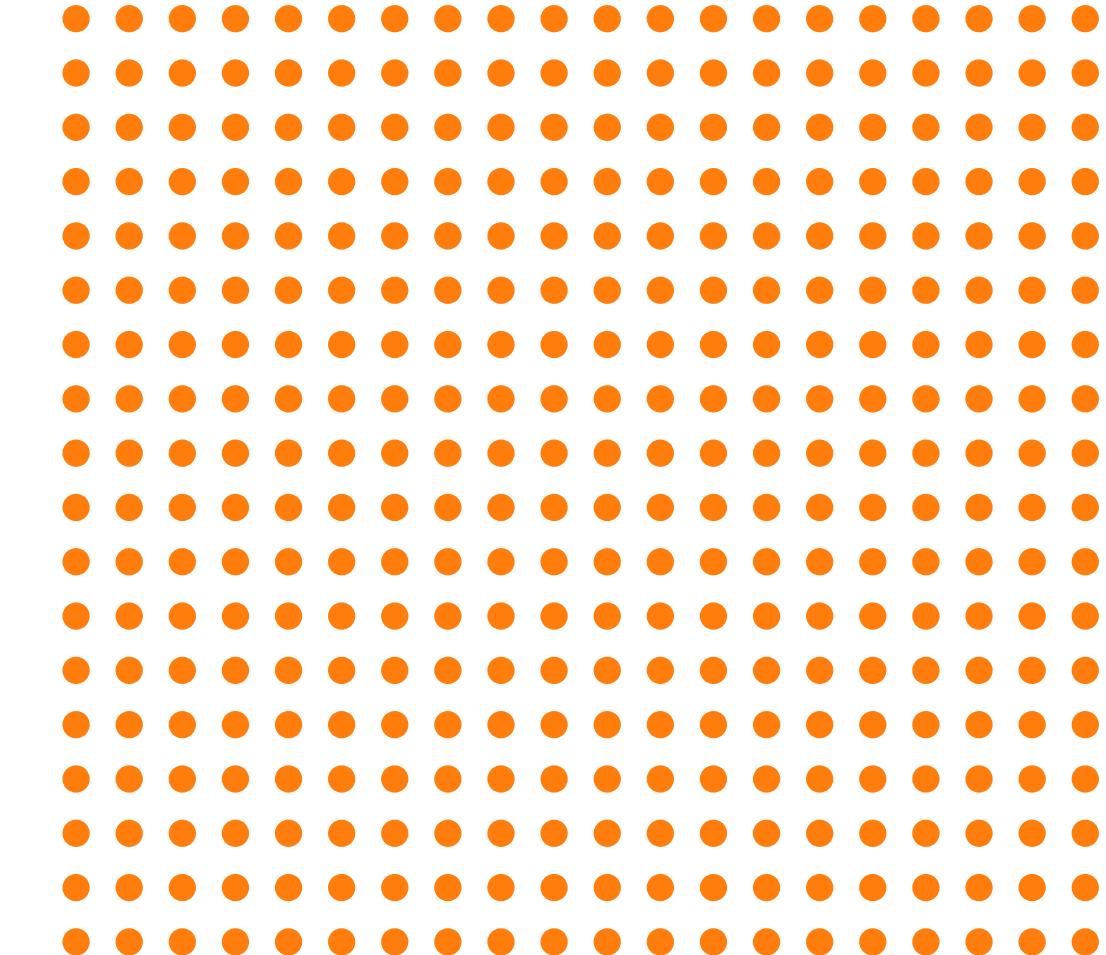


Possible anisotropía \Rightarrow aliasing

Fuentes en grilla regular

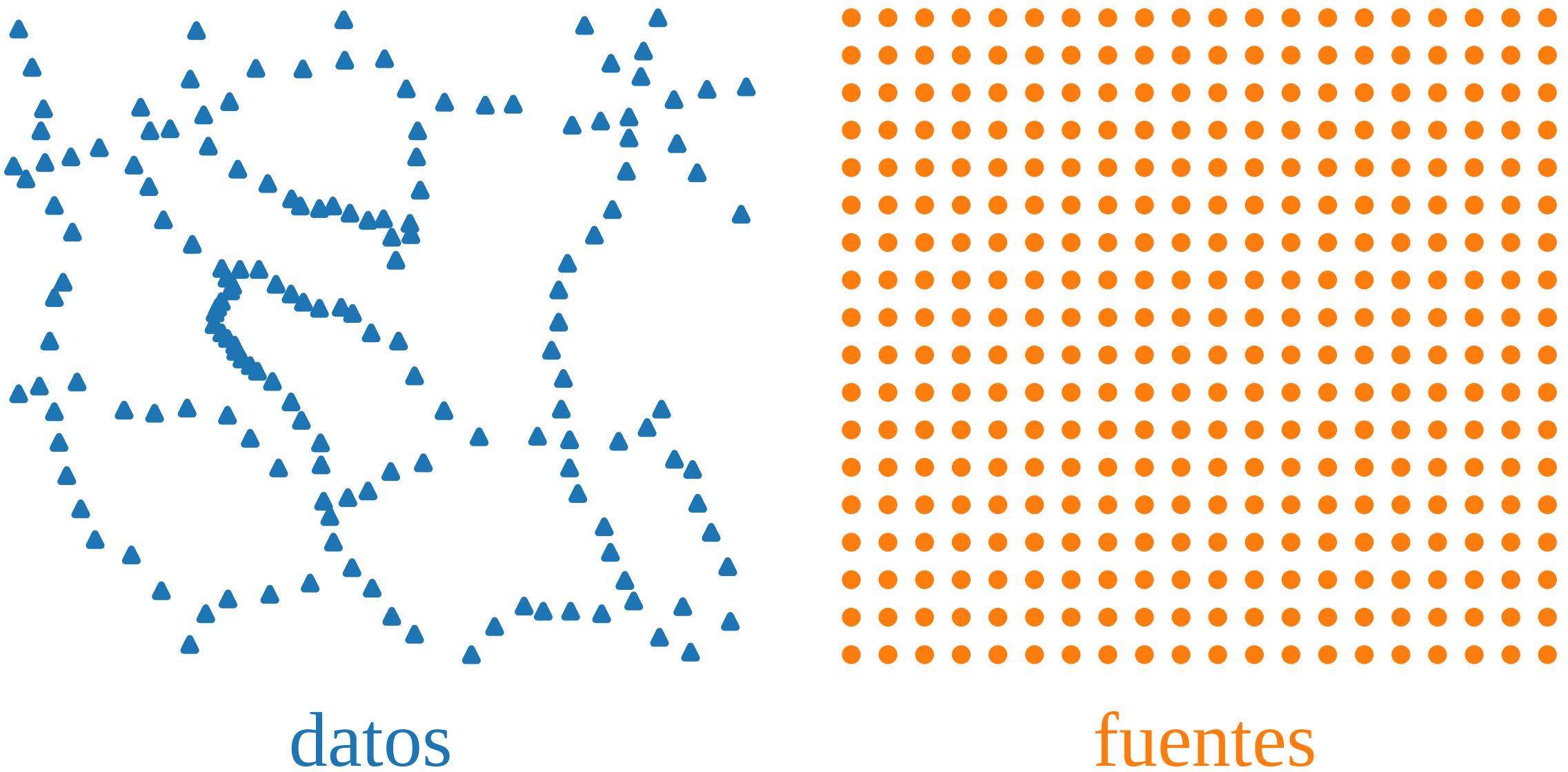


datos



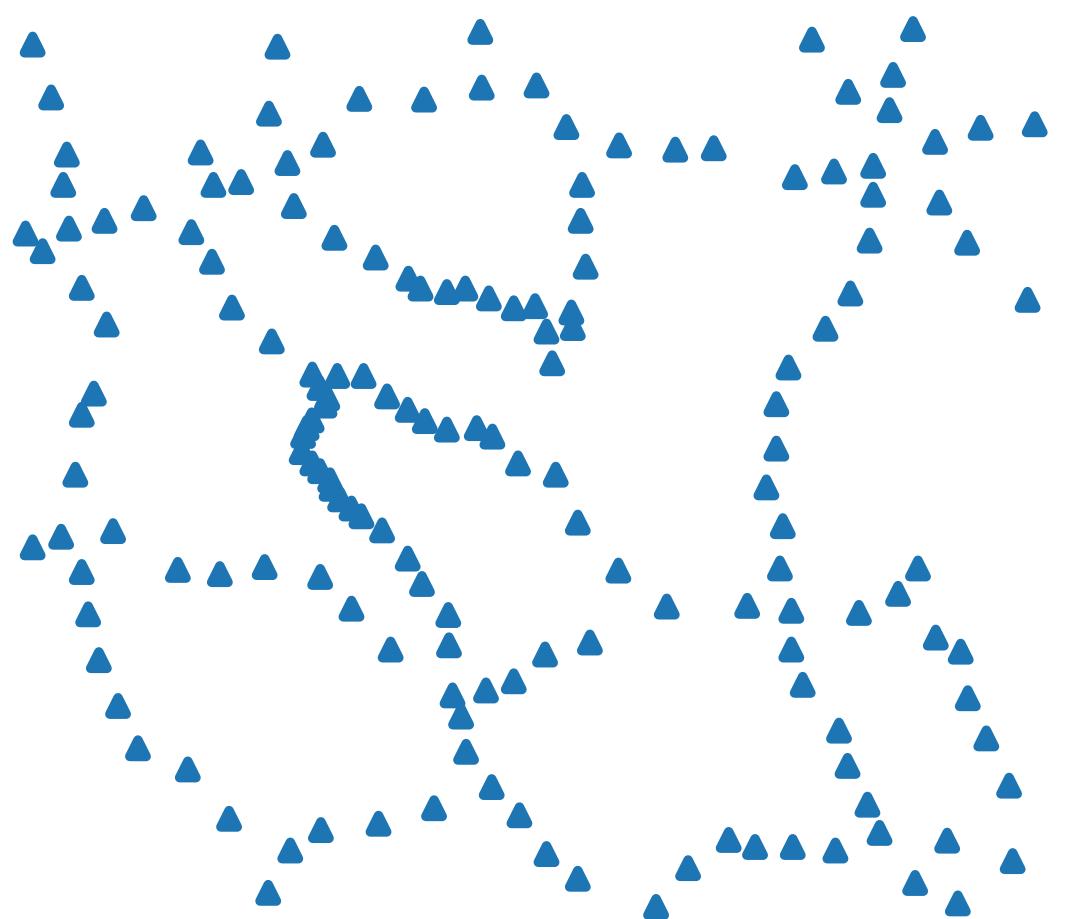
fuentes

Fuentes en grilla regular

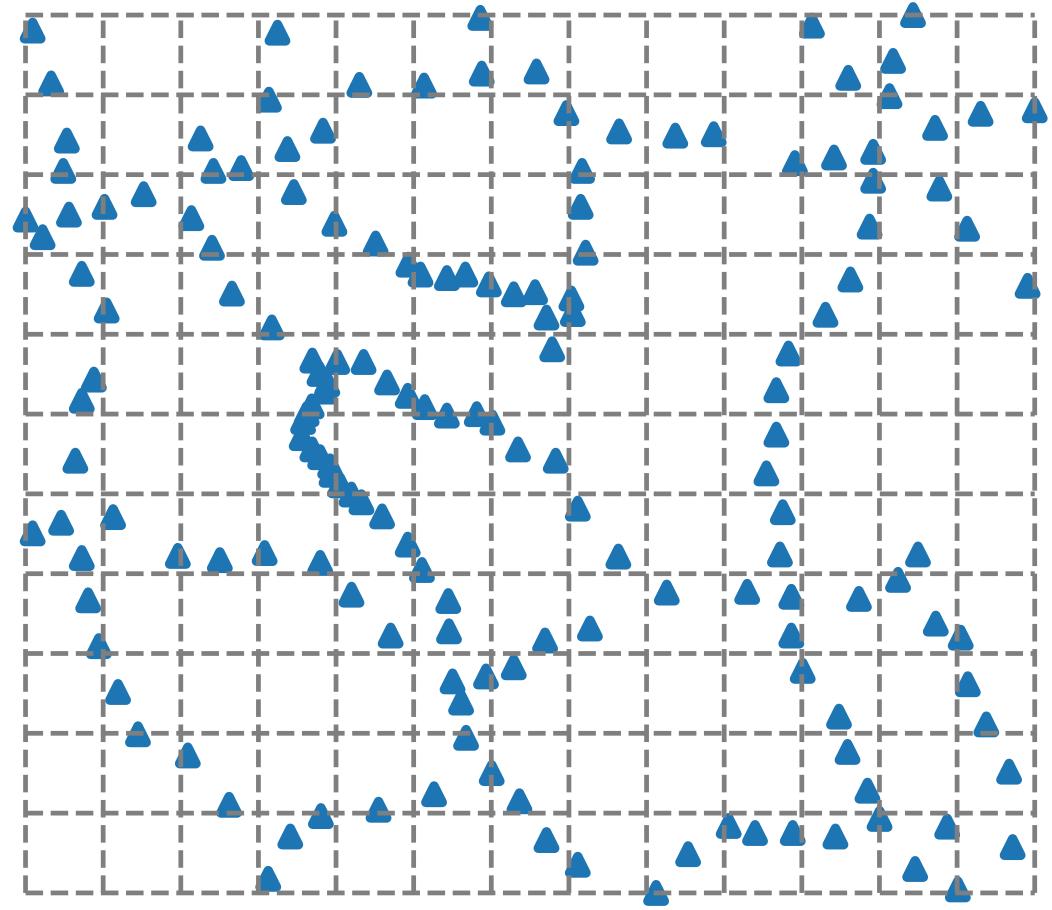


M fuentes $>$ N datos \Rightarrow problema subdeterminado

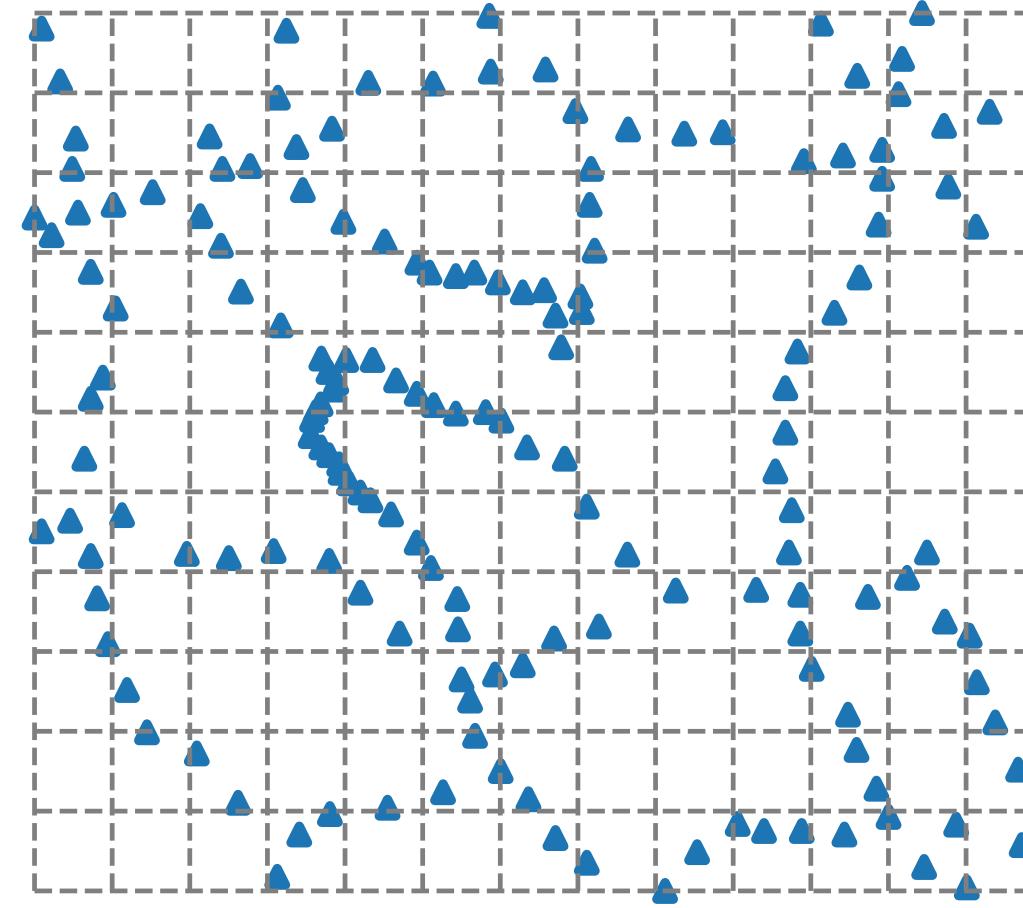
Fuentes promediadas por bloque



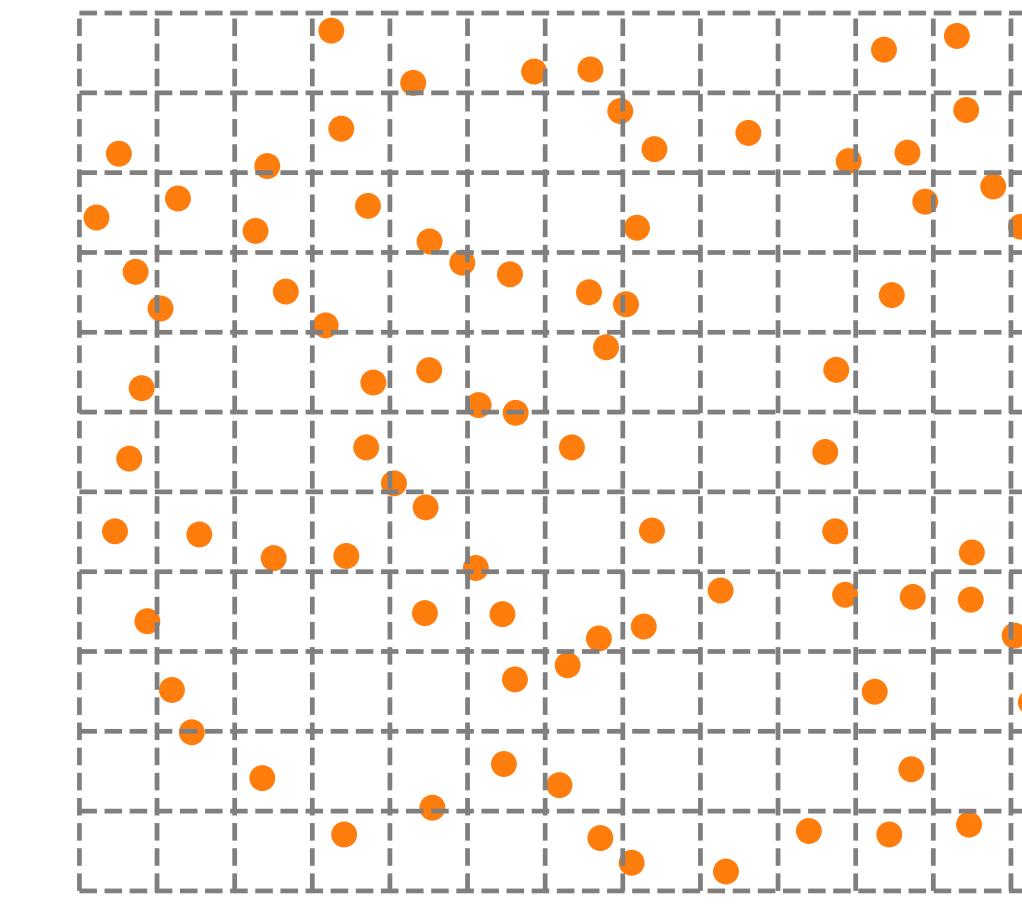
datos



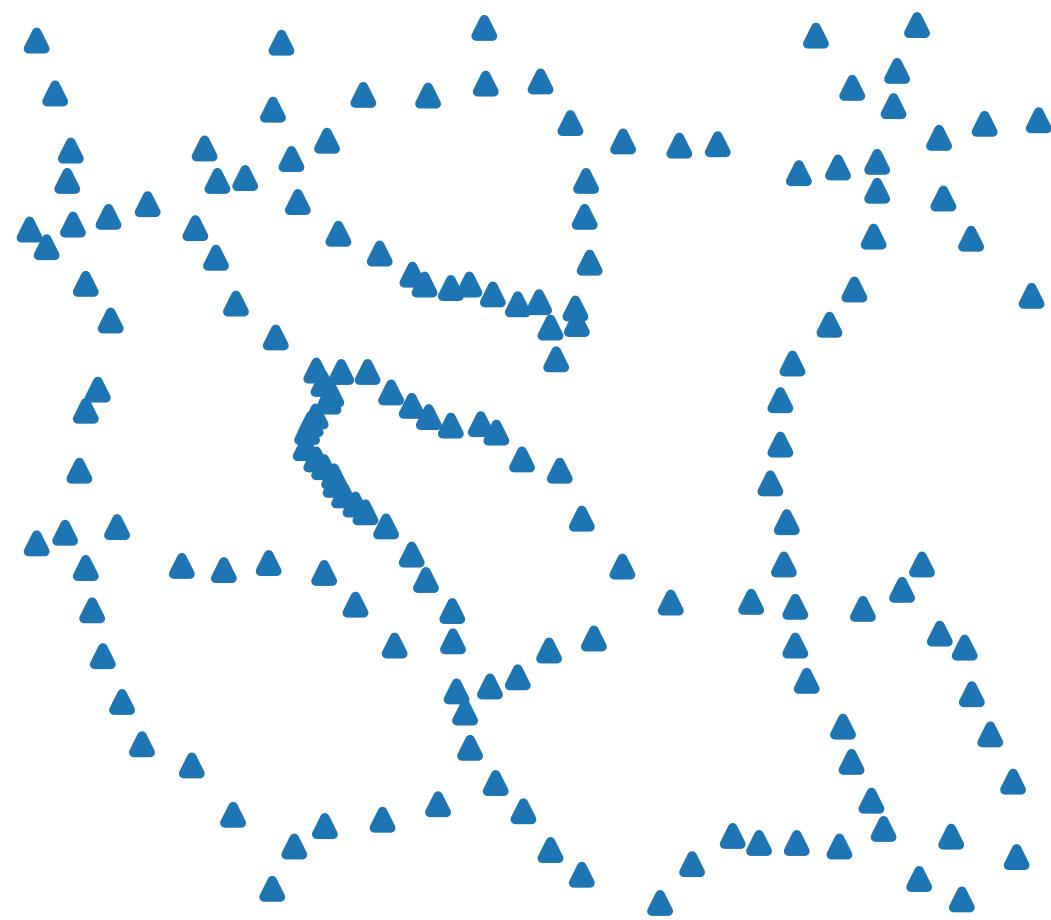
datos



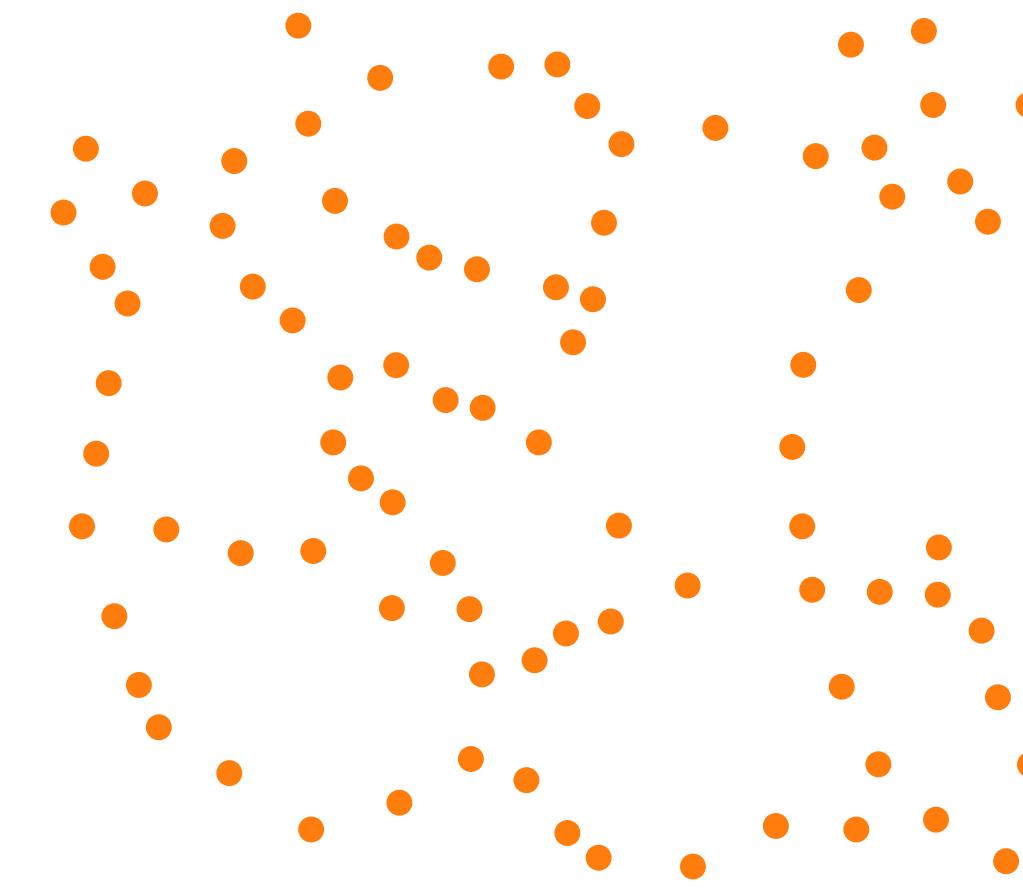
datos



fuentes



datos



fuentes

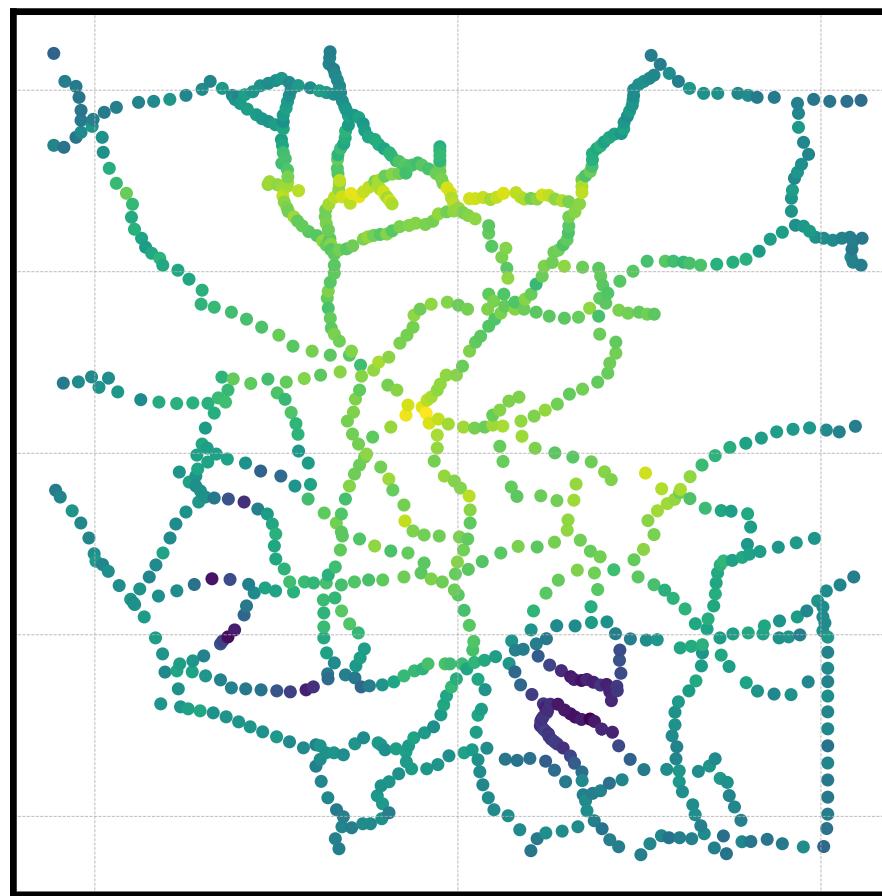
Características

- Evita anisotropías
- N datos $> M$ fuentes
 - Problema sobredeterminado
 - Reduce uso de memoria
- Tamaño de bloques controla cantidad de fuentes

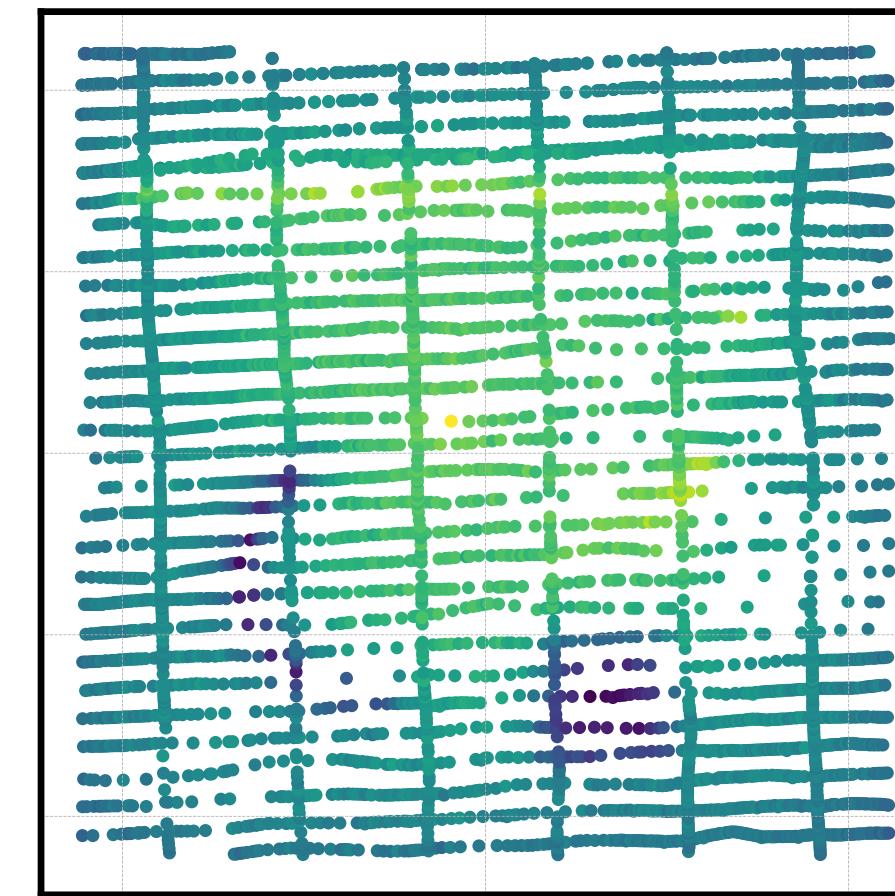
Desempeño de los nuevos métodos

Prueba con datos sintéticos

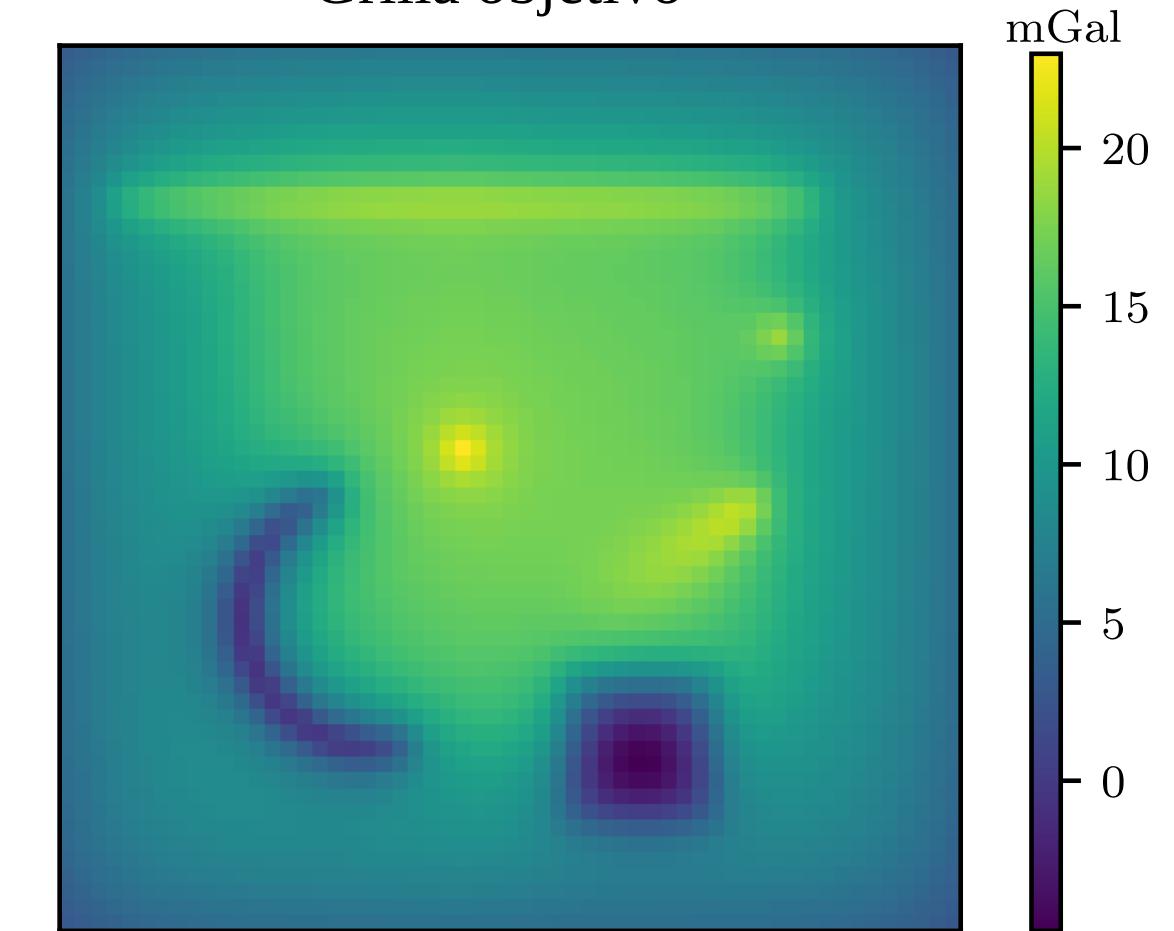
Muestra sobre terreno



Muestra aérea



Grilla objetivo



Resultados

Fuentes promediadas por bloque

Fuentes promediadas por bloque

- Precisión equivalente

Fuentes promediadas por bloque

- Precisión equivalente
- Menor cantidad de fuentes: reduce uso de memoria

Fuentes equivalentes potenciadas por gradiente

Fuentes equivalentes potenciadas por gradiente

- Reducción significativa de uso de memoria

Fuentes equivalentes potenciadas por gradiente

- Reducción significativa de uso de memoria
- Ligera menor precisión

Fuentes equivalentes potenciadas por gradiente

- Reducción significativa de uso de memoria
- Ligera menor precisión
- Menor tiempo de cómputo

Fuentes equivalentes potenciadas por gradiente

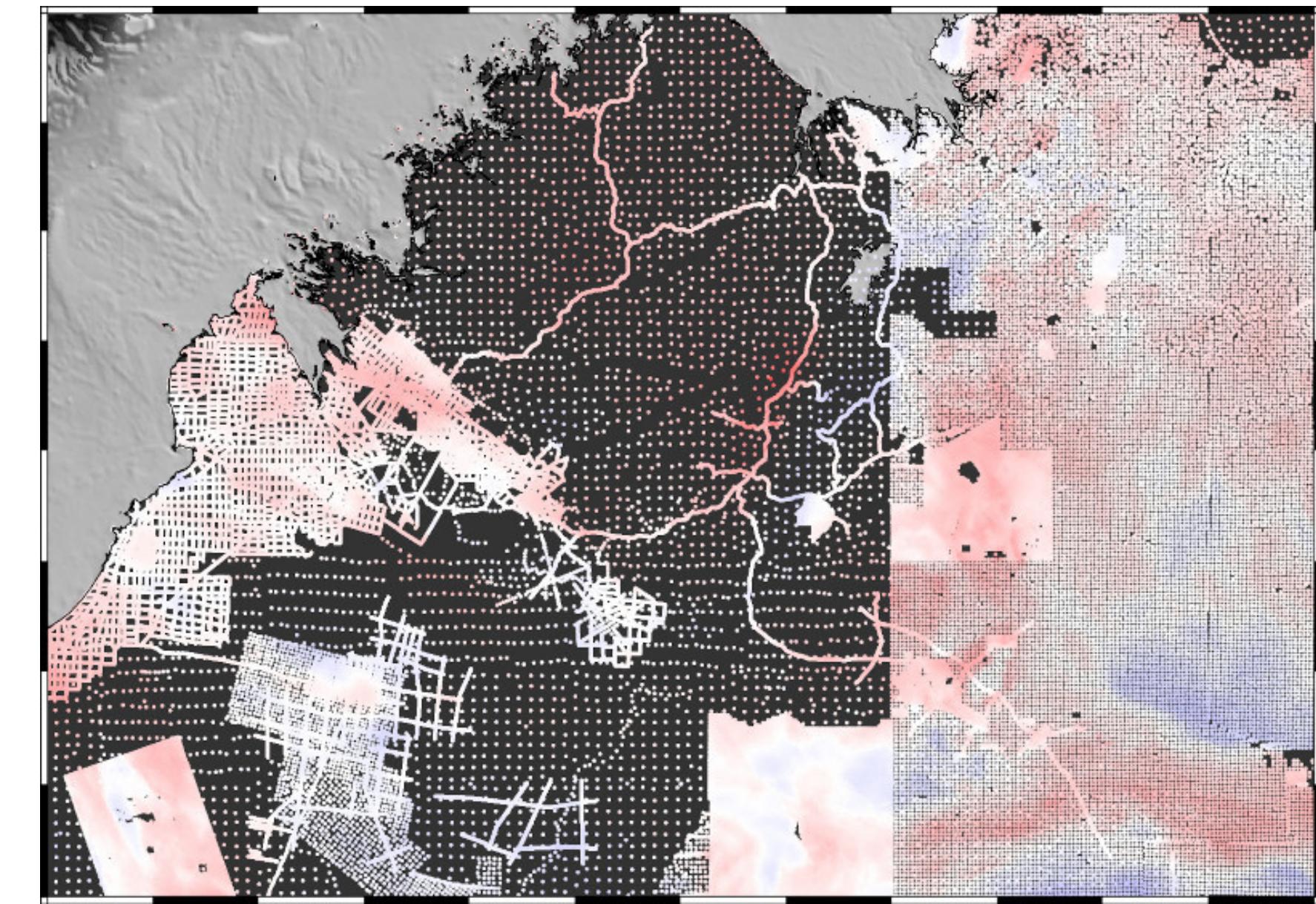
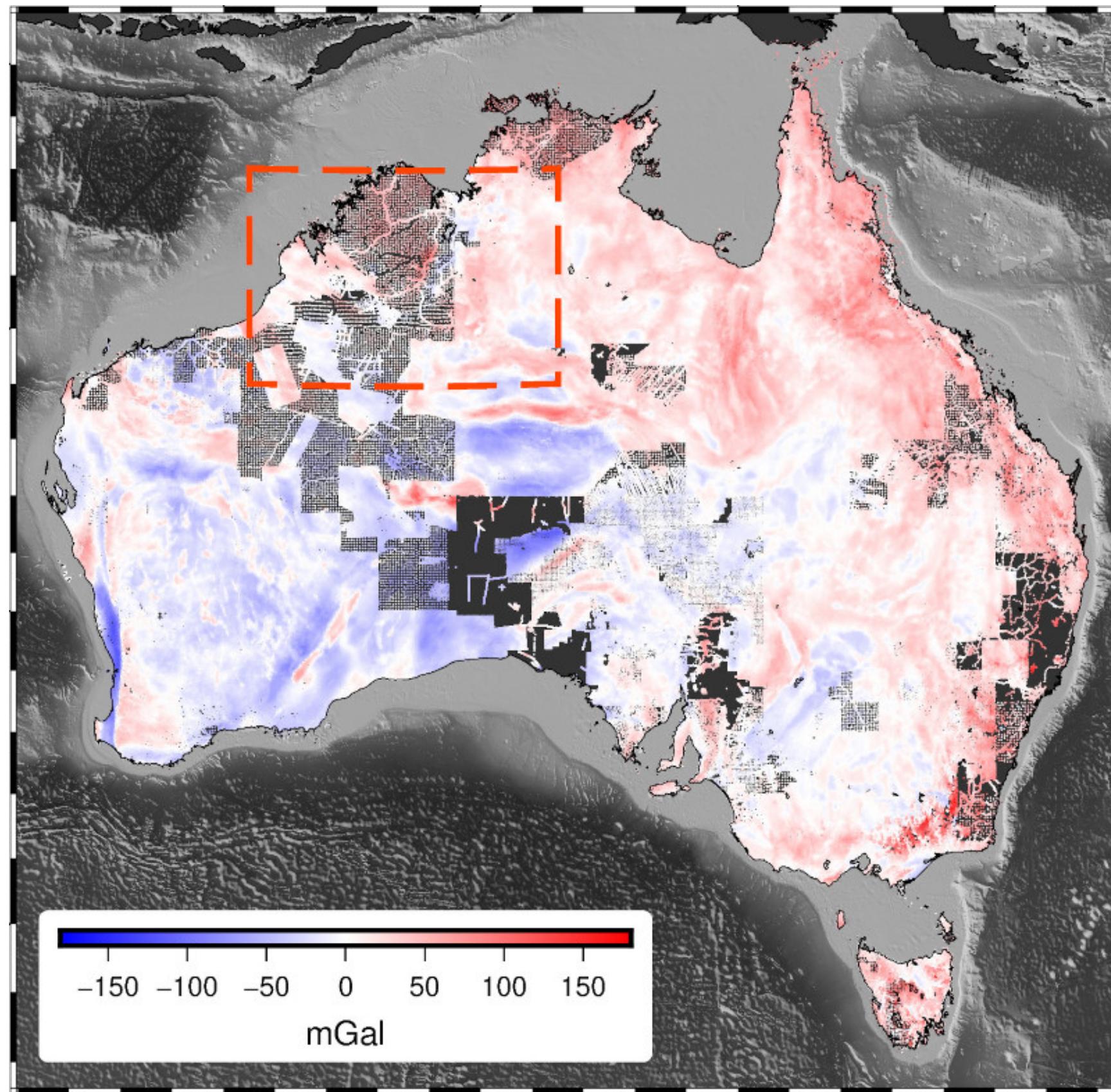
- Reducción significativa de uso de memoria
- Ligera menor precisión
- Menor tiempo de cómputo
-  Tamaño de ventana  Precisión

Fuentes equivalentes potenciadas por gradiente

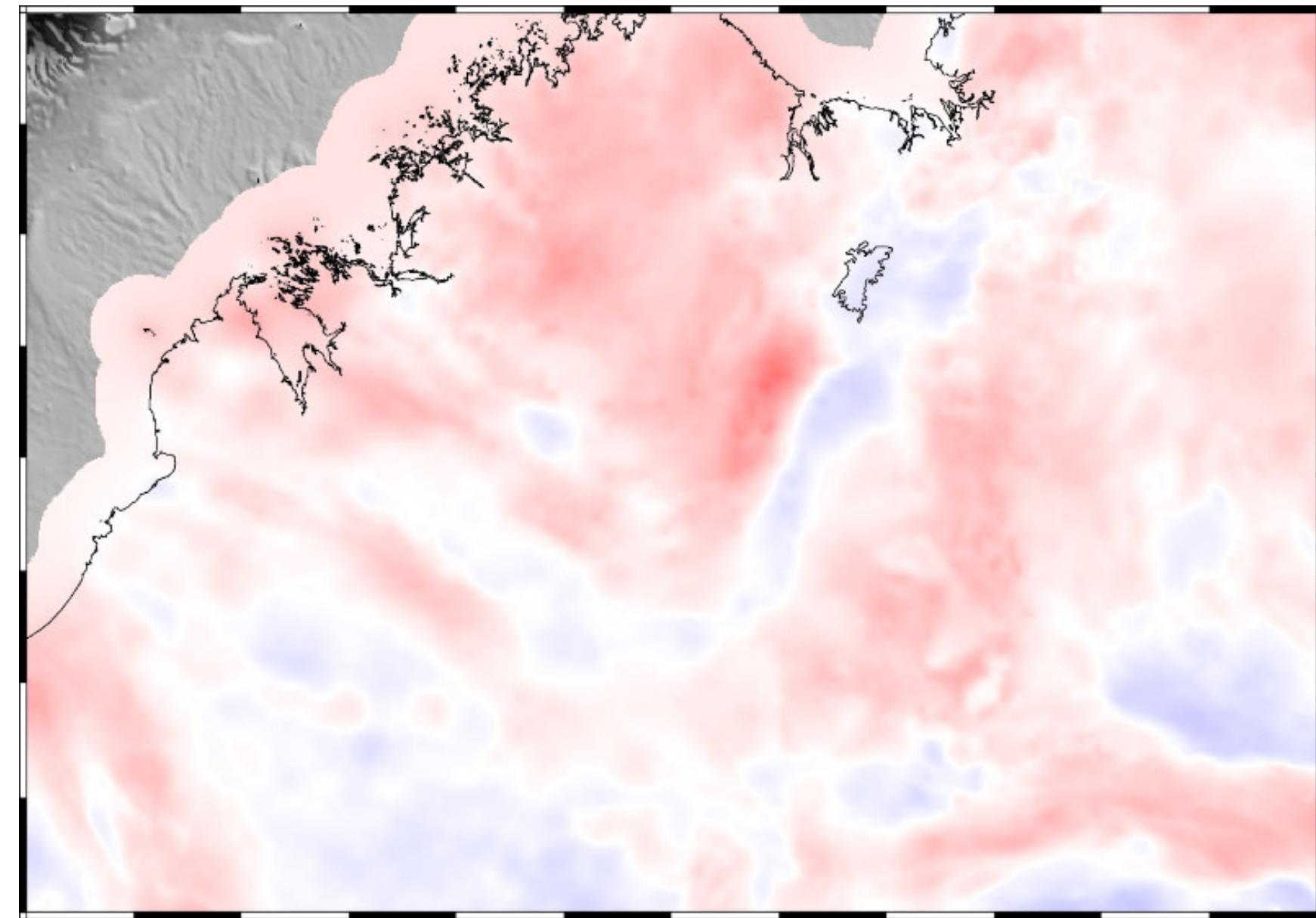
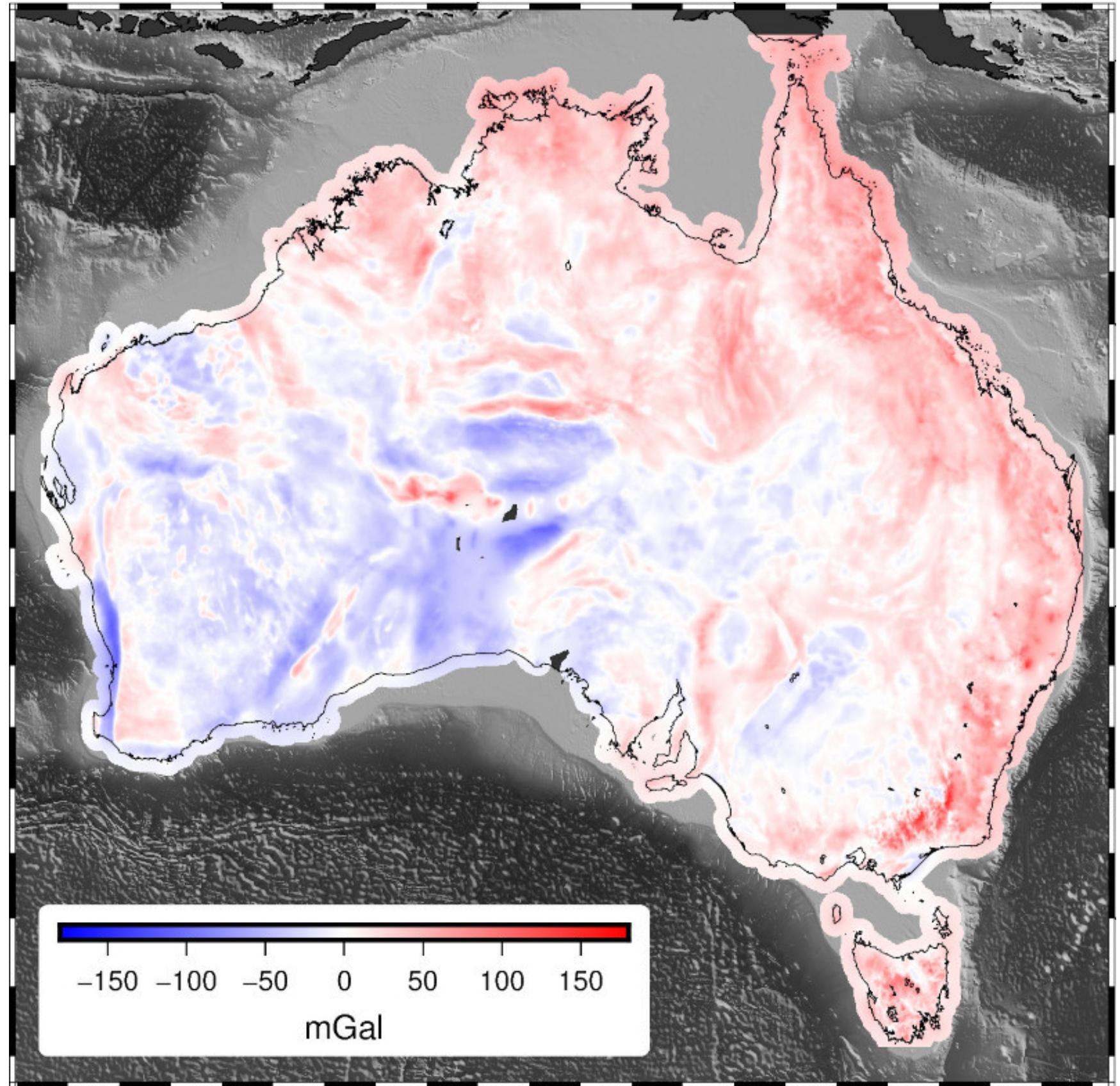
- Reducción significativa de uso de memoria
- Ligera menor precisión
- Menor tiempo de cómputo
- \uparrow Tamaño de ventana \uparrow Precisión
- Ventanas aleatorias: mejores predicciones

Grillado de datos gravimétricos de Australia

+1.7 millones de puntos



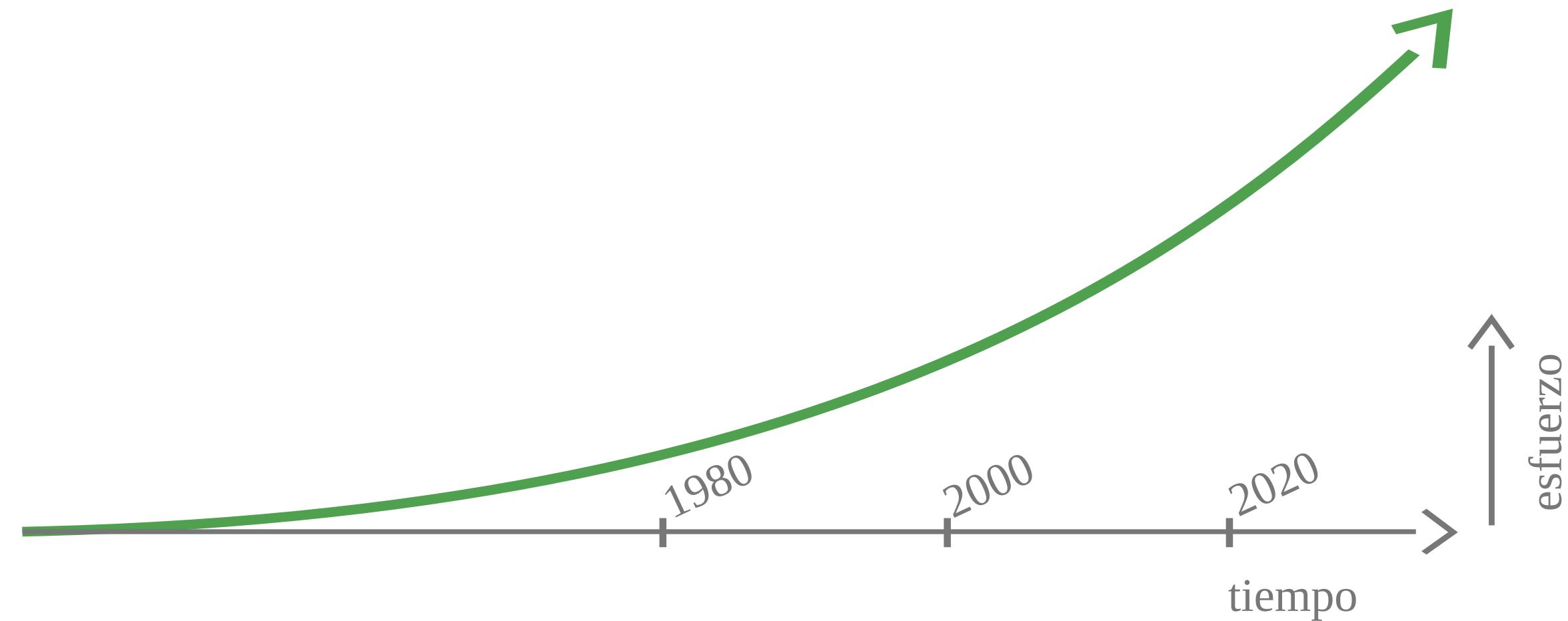
Grillados con fuentes equivalentes potenciadas con gradiente



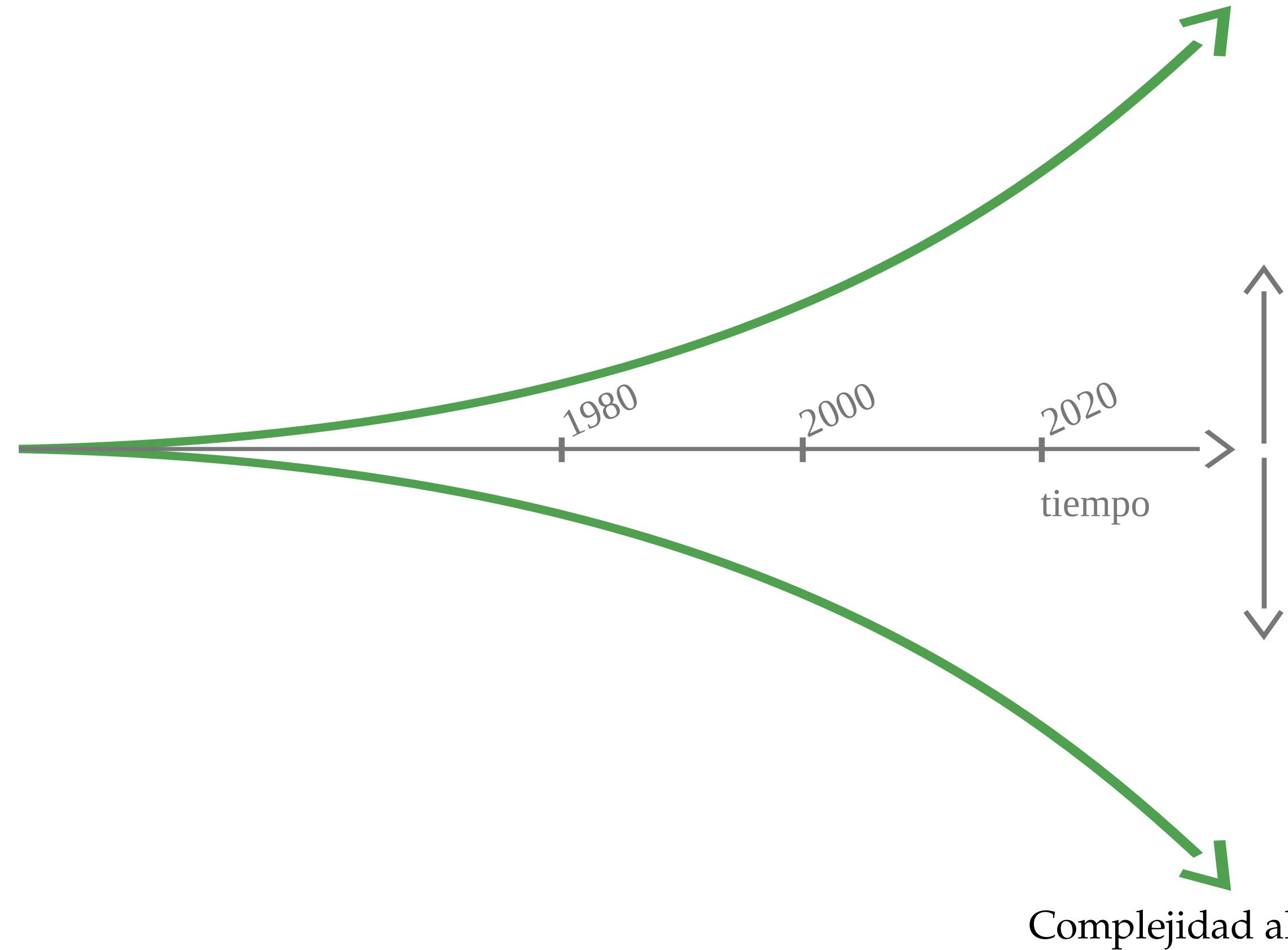
Fatiando a Terra

Complejización del problema científico

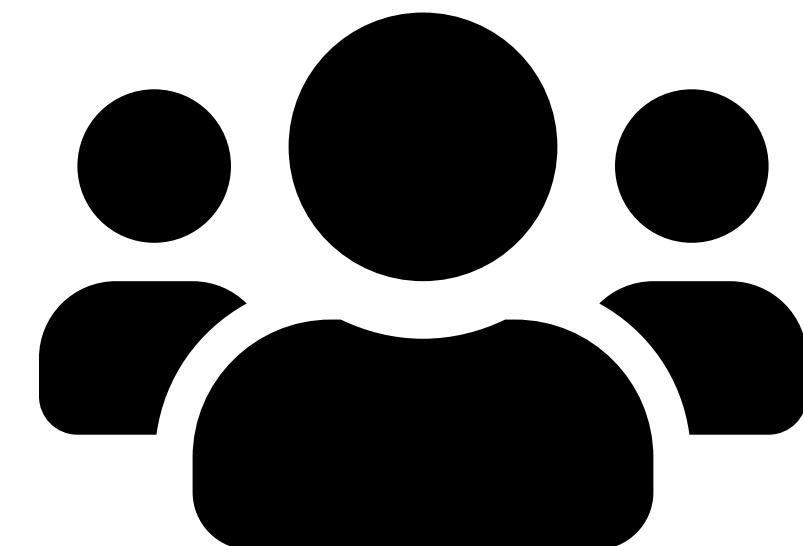
Complejidad del problema Geocientífico



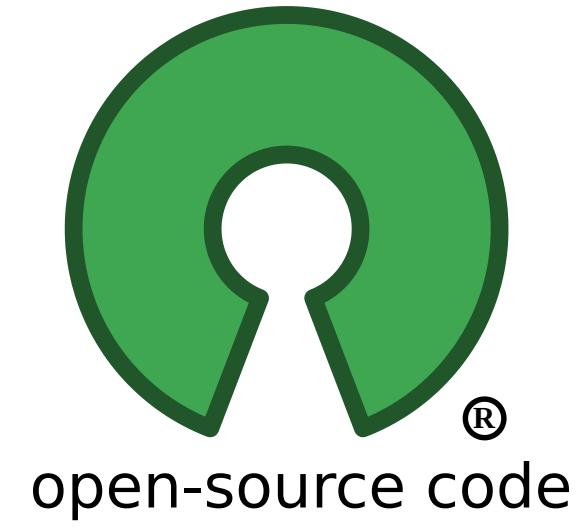
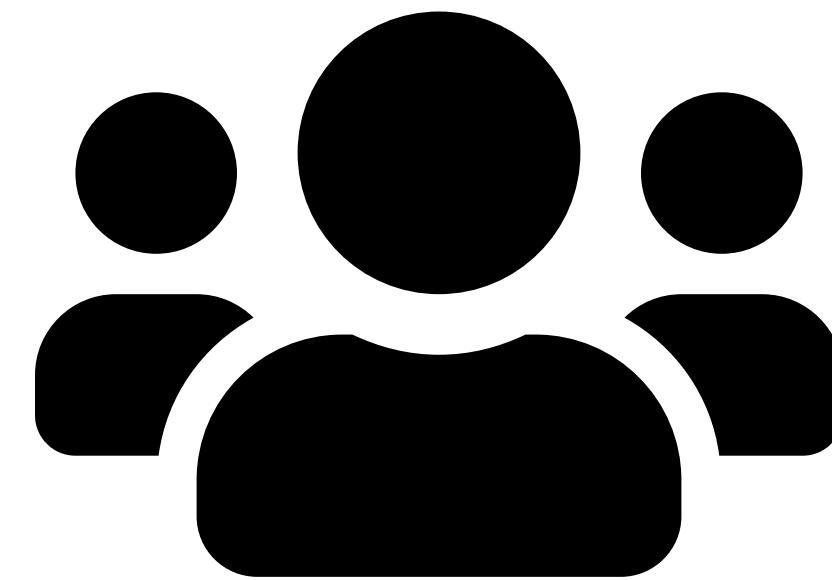
Complejidad del problema Geocientífico



Soluciones comunitarias



Soluciones comunitarias





Fatiando a Terra.

Open-source Python tools for Geophysics

www.fatiando.org

En portugués significa cortando la Tierra



Pooch

**Descarga y almacena
datos científicos**



[fatiando/pooch](#)



[doi: 10.21105/joss.01943](#)

VERDE

**Procesado y grillado
de datos espaciales**



[fatiando/verde](#)



[doi: 10.21105/joss.00957](#)

Boule

**Elipsoide de referencia y
gravedad normal**



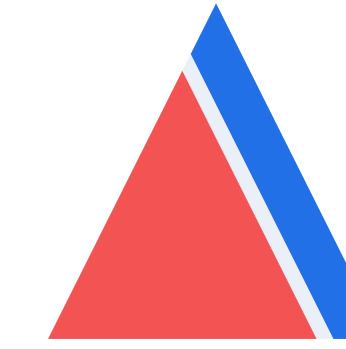
[fatiando/boule](#)

Δ harmonica

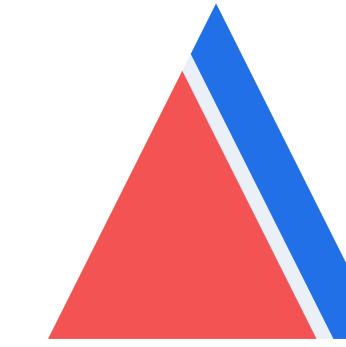
**Procesado y modelado de datos
gravimétricos y magnéticos**



[fatiando/harmonica](#)

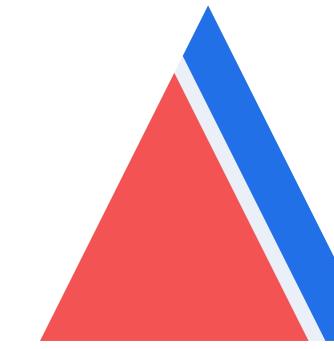


harmonica



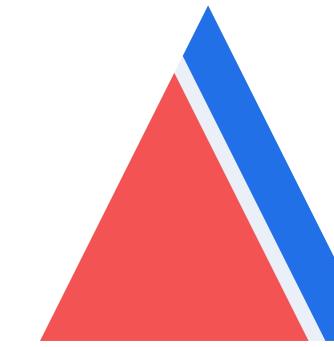
harmonica

- Lidero el proyecto



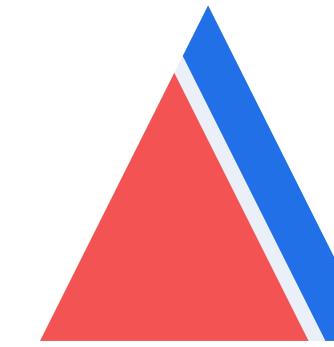
harmonica

- Lidero el proyecto
- Herramientas necesarias para el desarrollo de la Tesis



harmonica

- Lidero el proyecto
- Herramientas necesarias para el desarrollo de la Tesis
- Teseroides de densidad variable



harmonica

- Lidero el proyecto
- Herramientas necesarias para el desarrollo de la Tesis
- Teseroides de densidad variable
- Fuentes equivalentes potenciadas por gradiente

Teseroides de densidad variable

```
1 import boule as bl
2 import harmonica as hm
3
4 radio_medio = bl.WGS84.mean_radius
5 teseroide = [-70, -60, -40, -30, radio_medio - 10e3, radio_medio]
6
7 # Definimos una densidad lineal para este teseroide.
8 @njit
9 def densidad_lineal(radio):
10     """Funcion de densidad lineal"""
11     return a * radio + b
12
13 gravedad = hm.tesseroid_gravity(
14     coordenadas, teseroide, densidad_lineal, field="g_z"
15 )
```

Teseroides de densidad variable

```
1 import boule as bl
2 import harmonica as hm
3
4 radio_medio = bl.WGS84.mean_radius
5 teseroide = [-70, -60, -40, -30, radio_medio - 10e3, radio_medio]
6
7 # Definimos una densidad lineal para este teseroide.
8 @njit
9 def densidad_lineal(radio):
10     """Funcion de densidad lineal"""
11     return a * radio + b
12
13 gravedad = hm.tesseroid_gravity(
14     coordenadas, teseroide, densidad_lineal, field="g_z"
15 )
```

Teseroides de densidad variable

```
1 import boule as bl
2 import harmonica as hm
3
4 radio_medio = bl.WGS84.mean_radius
5 teseroide = [-70, -60, -40, -30, radio_medio - 10e3, radio_medio]
6
7 # Definimos una densidad lineal para este teseroide.
8 @njit
9 def densidad_lineal(radio):
10     """Funcion de densidad lineal"""
11     return a * radio + b
12
13 gravedad = hm.tesseroid_gravity(
14     coordenadas, teseroide, densidad_lineal, field="g_z"
15 )
```

Teseroides de densidad variable

```
1 import boule as bl
2 import harmonica as hm
3
4 radio_medio = bl.WGS84.mean_radius
5 teseroide = [-70, -60, -40, -30, radio_medio - 10e3, radio_medio]
6
7 # Definimos una densidad lineal para este teseroide.
8 @njit
9 def densidad_lineal(radio):
10     """Funcion de densidad lineal"""
11     return a * radio + b
12
13 gravedad = hm.tesseroid_gravity(
14     coordenadas, teseroide, densidad_lineal, field="g_z"
15 )
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Fuentes equivalentes potenciadas por gradiente

```
1 import harmonica as hm
2
3 fuentes_eq = hm.EquivalentSourcesGB(
4     depth=9e3,
5     damping=10,
6     window_size=100e3,
7     block_size=2e3,
8 )
9
10 fuentes_eq.fit(coordenadas, datos)
11
12 grilla = fuentes_eq.grid(upward=1e3, spacing=2e3)
```

Mejores prácticas

Documentación



Pooch v1.6.0

Search the docs ...

Pooch Documentation

GETTING STARTED

- Why use Pooch?
- Installing
- [Retrieving a single data file](#)
- Fetching files from a registry
- Manage a package's sample data

TRAINING YOUR POOCH

- Hashes: Calculating and bypassing
- User-defined cache location
- Registry files
- Multiple download URLs
- Download protocols
- Logging and verbosity

- Downloaders: Customizing the download

- Processors: Post-download actions

- Authentication

- Printing progress bars

- Unpacking archives

- Decompressing

REFERENCE



Retrieving a single data file

Basic usage

If you only want to download one or two data files, use the `pooch.retrieve` function:

```
import pooch

file_path = pooch.retrieve(
    # URL to one of Pooch's test files
    url="https://github.com/fatiando/pooch/raw/v1.0.0/data/tiny-data.txt",
    known_hash="md5:70e2af3fd7e336ae478b1e740a5f08e",
)
```

The code above will:

1. Check if the file from this URL already exists in Pooch's default cache folder (see `pooch.os_cache`).
2. If it doesn't, the file is downloaded and saved to the cache folder.
3. The MD5 `hash` is compared against the `known_hash` to make sure the file isn't corrupted.
4. The function returns the absolute path to the file on your computer.

If the file already existed on your machine, Pooch will check if its MD5 hash matches the `known_hash`:

- If it does, no download happens and the file path is returned.
- If it doesn't, the file is downloaded once more to get an updated version on your computer.

Since the download happens only once, you can place this function call at the start of your script or Jupyter notebook without having to worry about repeat downloads. Anyone getting a copy of your code should also get the correct data file the first time they run it.

See also

Pooch can handle multiple download protocols like HTTP, FTP, SFTP, and even download from repositories like [figshare](#) and [Zenodo](#) by using the DOI instead of a URL. See [Download protocols](#).

See also

You can use **different hashes** by specifying different algorithm names: `sha256:XXXXXX`, `sha1:XXXXXX`, etc. See [Hashes: Calculating and bypassing](#).

Pruebas de software

Name	Stmts	Miss	Branch	BrPart	Cover	Missing
/home/santi/git/harmonica/constants.py	2	0	0	0	100%	
/home/santi/git/harmonica/datasets/sample_data.py	42	0	0	0	100%	
/home/santi/git/harmonica/equivalent_sources/cartesian.py	83	0	20	0	100%	
/home/santi/git/harmonica/equivalent_sources/gradient_boosted.py	79	0	30	0	100%	
/home/santi/git/harmonica/equivalent_sources/spherical.py	61	0	8	0	100%	
/home/santi/git/harmonica/equivalent_sources/utils.py	29	0	14	0	100%	
/home/santi/git/harmonica/forward/_tesseroid_utils.py	162	0	66	0	100%	
/home/santi/git/harmonica/forward/_tesseroid_variable_density.py	67	0	16	0	100%	
/home/santi/git/harmonica/forward/point.py	80	0	20	0	100%	
/home/santi/git/harmonica/forward/prism.py	84	0	42	0	100%	
/home/santi/git/harmonica/forward/prism_layer.py	101	0	22	0	100%	
/home/santi/git/harmonica/forward/tesseroid.py	68	0	26	0	100%	
/home/santi/git/harmonica/forward/utils.py	57	0	8	0	100%	
/home/santi/git/harmonica/gravity_corrections.py	11	0	0	0	100%	
/home/santi/git/harmonica/io.py	70	0	42	0	100%	
/home/santi/git/harmonica/isostasy.py	18	0	2	0	100%	
/home/santi/git/harmonica/synthetic/surveys.py	29	0	6	0	100%	
/home/santi/git/harmonica/version.py	3	0	0	0	100%	
TOTAL	1046	0	322	0	100%	

===== 198 passed, 111 skipped, 85 warnings in 11.64s =====

Revisión por pares

The screenshot shows a GitHub pull request interface. At the top, it says "examples/equivalent_sources/gradient_boosted.py Outdated". Below that is a code snippet:

```
67 +  
68 + # Let's estimate the memory required to store the largest Jacobian when using  
69 + # these values for the window_size and the block_size.  
70 + jacobian_req_memory = hm.EquivalentSourcesGB.estimate_required_memory()
```

A comment by user **leouieda** on Nov 13, 2021, suggests making the method regular instead of a class method. The response by user **santisoler** on Nov 15, 2021, provides an example of how to do this.

leouieda on Nov 13, 2021 (Member): How about making this a regular method instead of a class method? Creating instances is cheap since the constructors don't do any computation. Then we could:

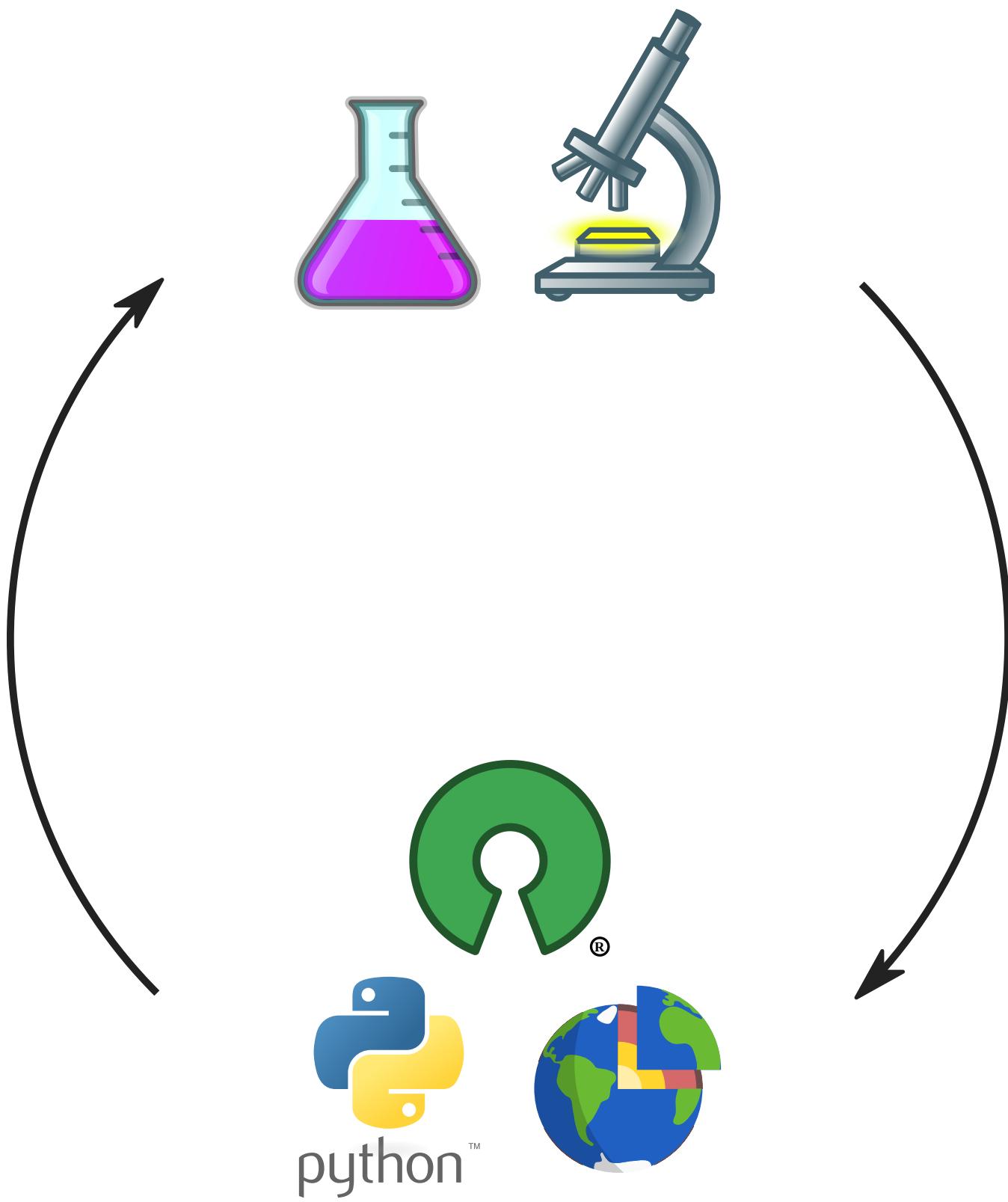
```
eqs = hm.EquivalentSourcesGB(window_size=10, block_size=1)  
print(eqs.estimate_required_memory(coordinates))  
# Happy = carry on  
# Too much = reduce window size and run again
```

This way you don't need to pass the arguments twice.

santisoler on Nov 15, 2021 (Member, Author): Now that you put it in an example, it actually looks better as a regular method. I'll change it!

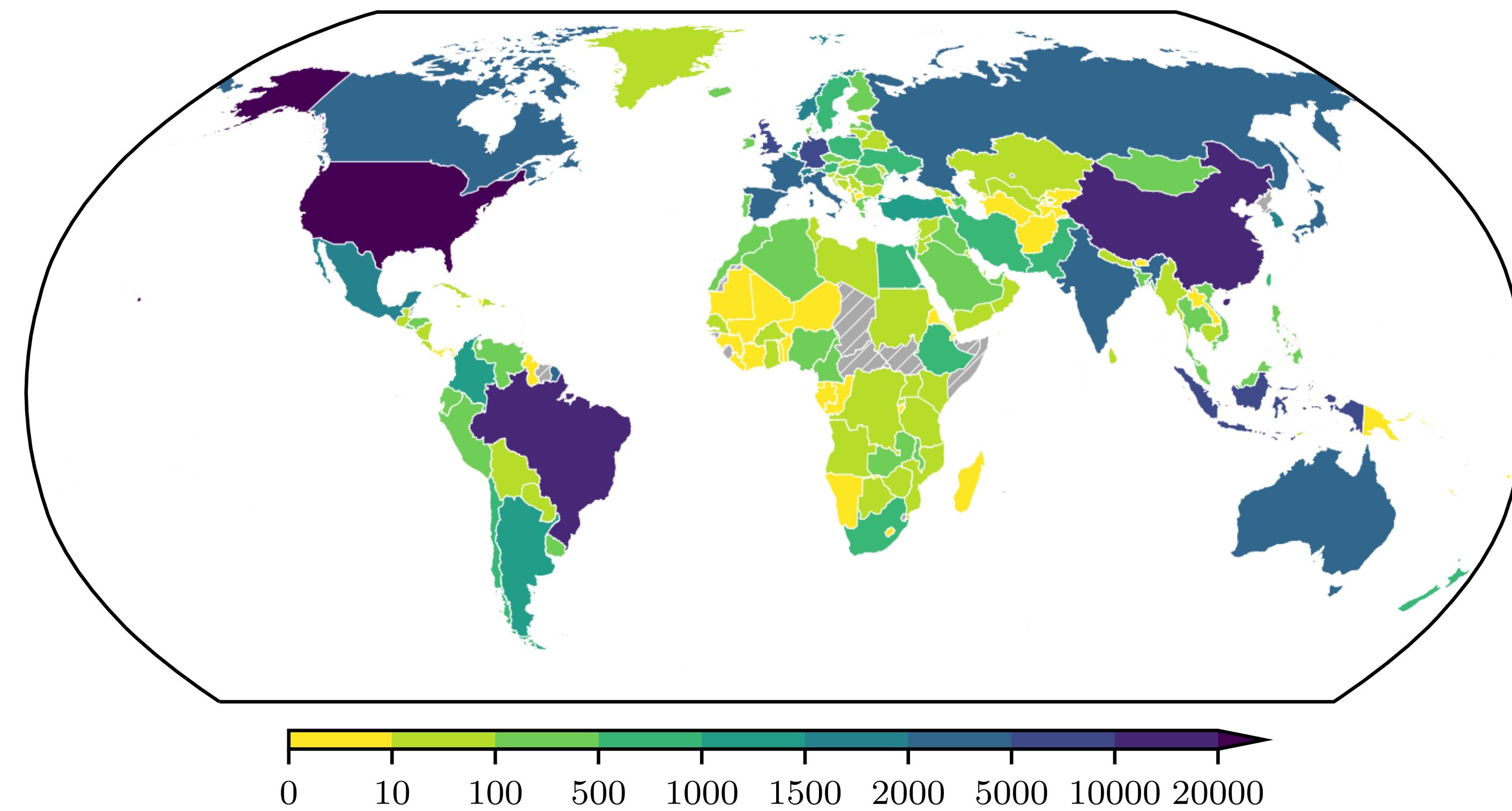
At the bottom, there is a "Reply..." button and a note: "Unresolve conversation santisoler marked this conversation as resolved."

Retroalimentación

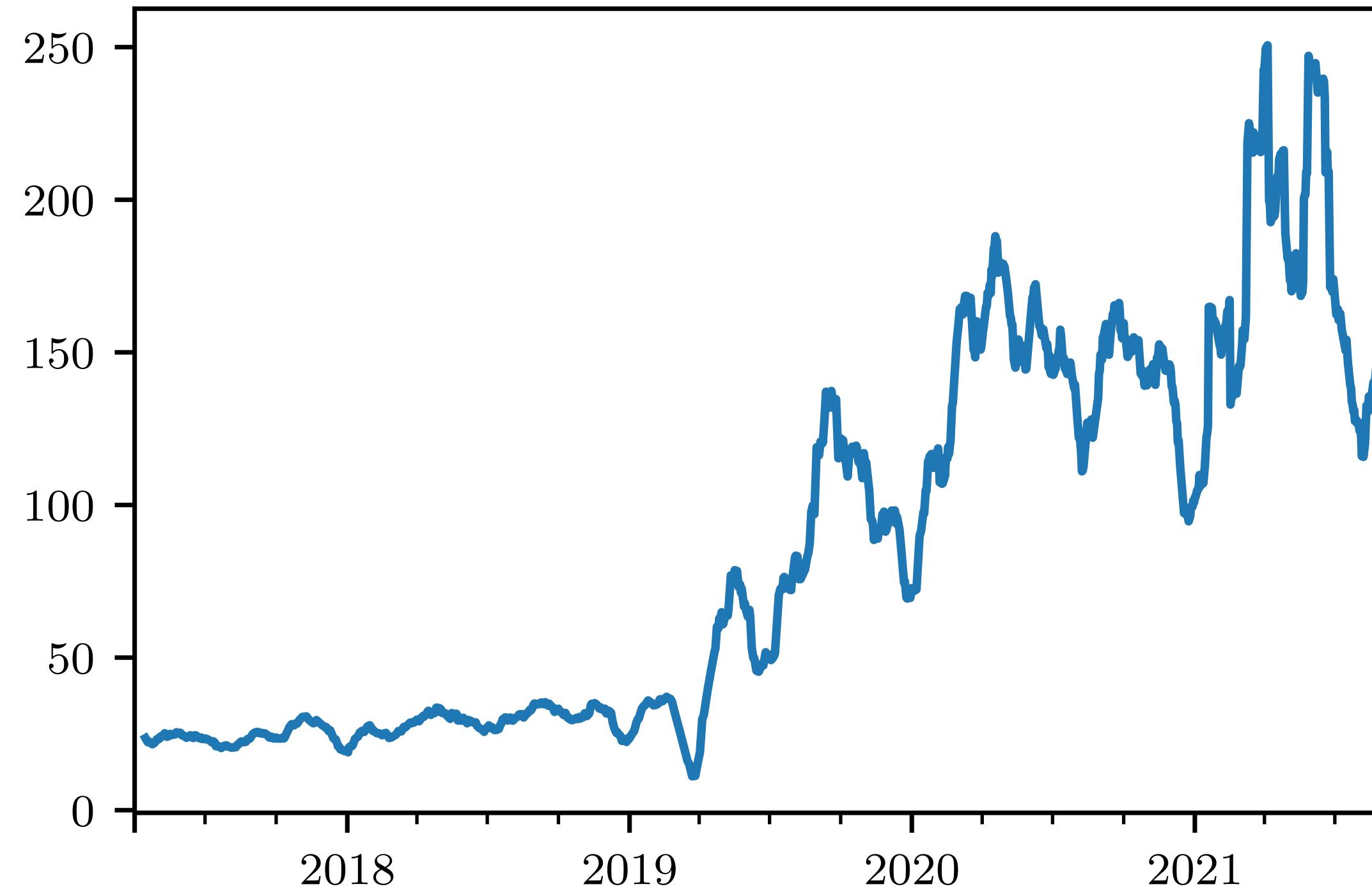


¿Quiénes lo usan?

Cantidad de visitas totales (entre 01/04/2017 y 29/09/2021)

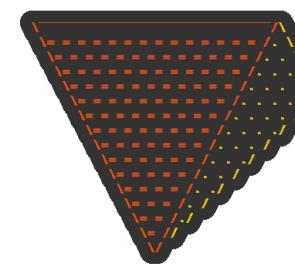


Cantidad de visitas diarias
(promediadas en ventana móvil de 28 días)

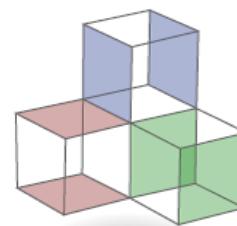


Ecosistema open-source en Geociencias

PyVista



subsurface



simpeg



pyGIMLi

Geophysical Inversion & Modelling Library



GemPy



PyGMT

Conclusiones

Teseroides con densidad variable

Teseroides con densidad variable

- Metodología de modelado directo

Teseroides con densidad variable

- Metodología de modelado directo
- Función de densidad arbitrarias

Teseroides con densidad variable

- Metodología de modelado directo
- Función de densidad arbitrarias
- Parámetros D y δ : errores < 1%

Teseroides con densidad variable

- Metodología de modelado directo
- Función de densidad arbitrarias
- Parámetros D y δ : errores < 1%
- Aplicación: modelado directo de la cuenca Neuquina

Fuentes equivalentes potenciadas por gradiente

Fuentes equivalentes potenciadas por gradiente

- Metodología para interpolación de grandes cantidades de datos

Fuentes equivalentes potenciadas por gradiente

- Metodología para interpolación de grandes cantidades de datos
- Disminuye significativamente uso de memoria

Fuentes equivalentes potenciadas por gradiente

- Metodología para interpolación de grandes cantidades de datos
- Disminuye significativamente uso de memoria
- Alcanza mejores tiempos de cómputo

Fuentes equivalentes potenciadas por gradiente

- Metodología para interpolación de grandes cantidades de datos
- Disminuye significativamente uso de memoria
- Alcanza mejores tiempos de cómputo
- Fuentes promediadas por bloque:

Fuentes equivalentes potenciadas por gradiente

- Metodología para interpolación de grandes cantidades de datos
- Disminuye significativamente uso de memoria
- Alcanza mejores tiempos de cómputo
- Fuentes promediadas por bloque:
 - Previenen anisotropías
 - Requieren menor uso de memoria

Fuentes equivalentes potenciadas por gradiente

- Metodología para interpolación de grandes cantidades de datos
- Disminuye significativamente uso de memoria
- Alcanza mejores tiempos de cómputo
- Fuentes promediadas por bloque:
 - Previenen anisotropías
 - Requieren menor uso de memoria
- Aplicación: grillado de +1.7M de datos sobre Australia

Software

Software

- Fatiando a Terra: proyecto open-source para geofísica

Software

- Fatiando a Terra: proyecto open-source para geofísica
- Implementaciones de las nuevas metodologías

Software

- Fatiando a Terra: proyecto open-source para geofísica
- Implementaciones de las nuevas metodologías
- Mejores prácticas: alta calidad

Software

- Fatiando a Terra: proyecto open-source para geofísica
- Implementaciones de las nuevas metodologías
- Mejores prácticas: alta calidad
- Retroalimentación con investigaciones científicas

Software

- Fatiando a Terra: proyecto open-source para geofísica
- Implementaciones de las nuevas metodologías
- Mejores prácticas: alta calidad
- Retroalimentación con investigaciones científicas
- Integración al ecosistema open-source en Geociencias

En síntesis

- Modelado de grandes estructuras con densidades variables
- Procesamiento de grandes cantidades de datos
- Herramientas de código abierto

Muchas gracias