



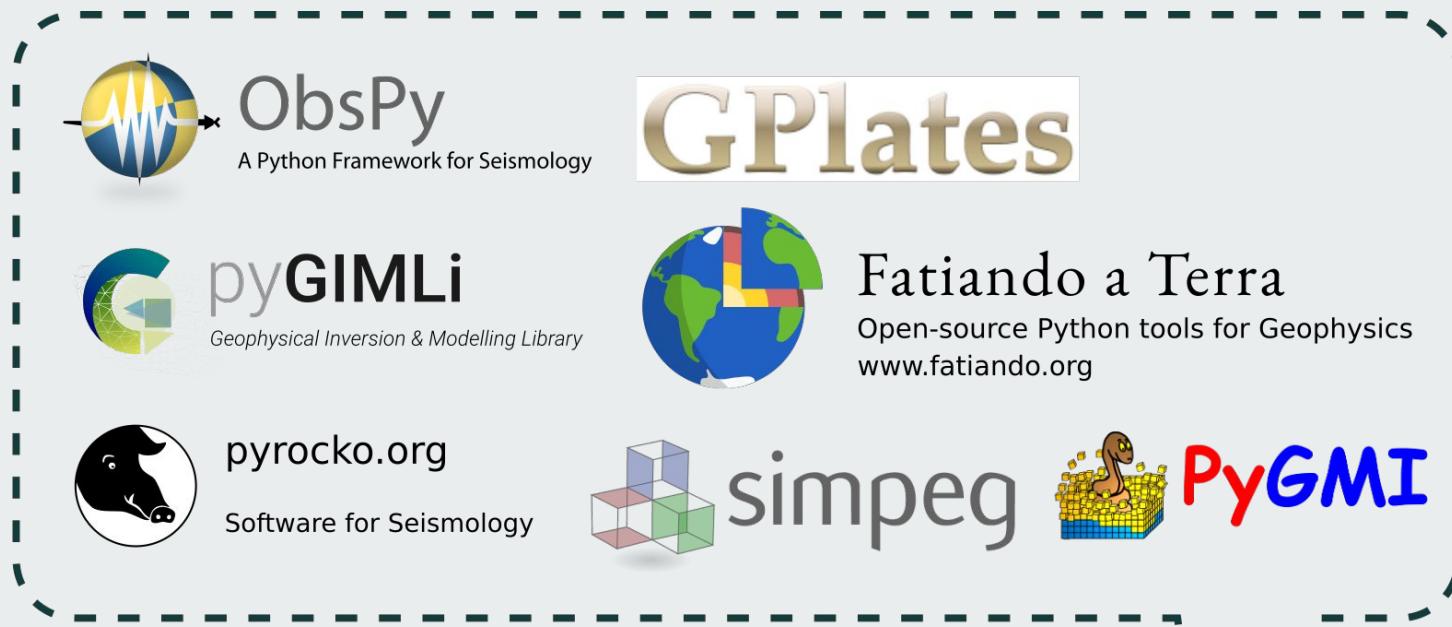
Experiencias en el desarrollo de Fatiando a Terra

Santiago Soler

Computer-Oriented Geoscience Lab
IGSV, UNSJ
CONICET



Librerías de código abierto en Geofísica



Seismic Unix



TauP





Creada por
Leonardo Uieda



Fatiando a Terra.

Open-source Python tools for Geophysics

www.fatiando.org

VERDE

Spatial data processing and interpolation (**gridding**) using Green's functions (or radial basis functions) with a machine learning inspired interface.

🔗 [fatiando/verde](#)

🌐 [www.fatiando.org/verde](#)

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

✓ Stable and ready for use

harmonica

Processing and modeling **gravity** and **magnetic** data, like terrain correction, upward continuation, equivalent layers, 3D inversion, and more.

🔗 [fatiando/harmonica](#)

🌐 [www.fatiando.org/harmonica/dev](#)

⚠ Early development and design

Pooch

Manages the download of sample data files over HTTP from a server and storing them in a local directory. Used by our other libraries.

🔗 [fatiando/pooch](#)

🌐 [www.fatiando.org/pooch](#)

⟳ Ready for use but still changing

RockHound

Download geophysical models and datasets (PREM, CRUST1.0, ETOPO1) and load them into Python. Relies on Pooch to manage the downloads.

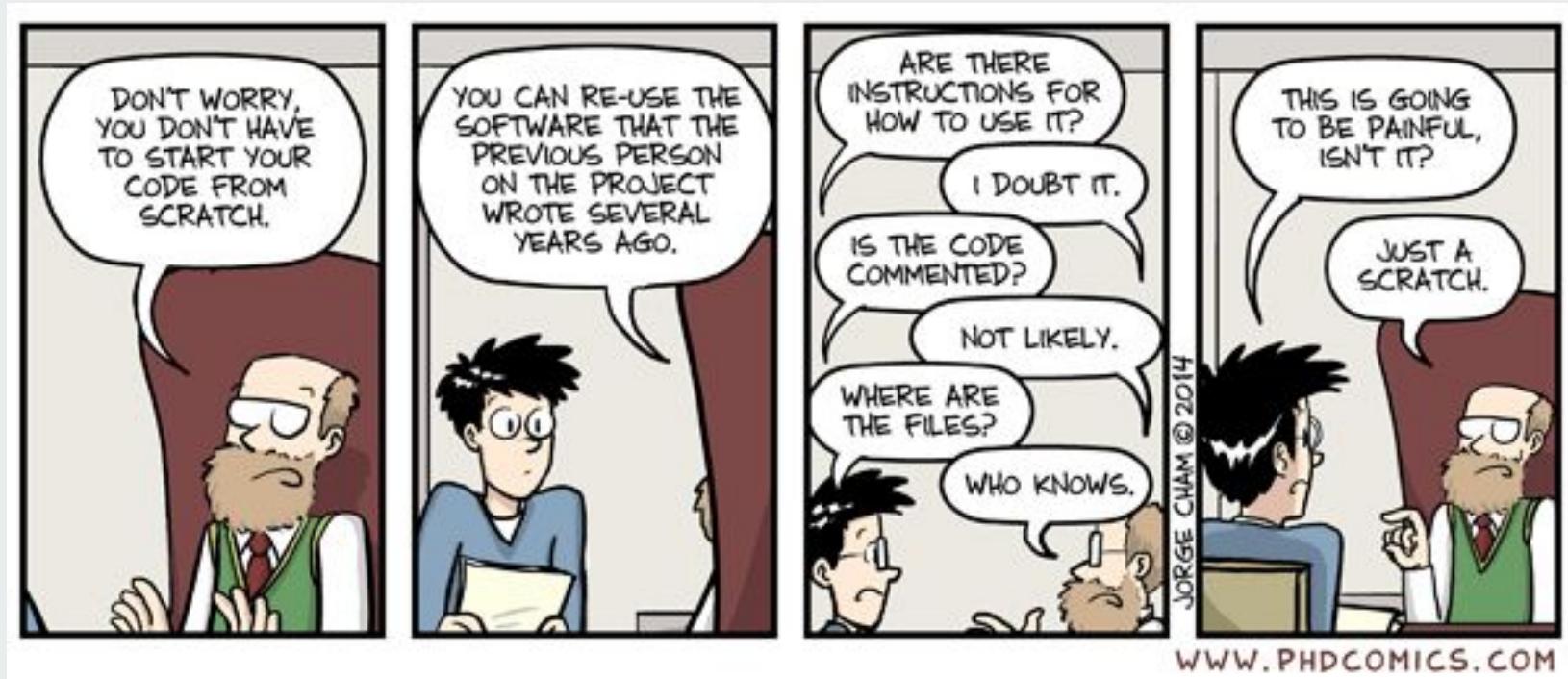
🔗 [fatiando/rockhound](#)

🌐 [www.fatiando.org/rockhound](#)

⟳ Ready for use but still changing

¿Por qué desarrollar una
librería de código abierto?

“Scratch” by Jorge Cham,
<http://www.phdcomics.com>



No te preocunes, no tendrás que escribir tu código desde cero.

Podrás reutilizar el software que escribió otra persona del proyecto hace muchos años.

¿Hay instrucciones de uso?

Lo dudo.

¿El código está comentado?

Probablemente no.

¿Dónde están los archivos?

Quién sabe.

Esto va a ser doloroso, no?

Solo un razguño.

- Ciencia más abierta
- Reproducibilidad
- Colaboración
- Eficiencia

Objetivos de la Charla

Objetivos de la Charla

- Desarrollar una librería

Objetivos de la Charla

- Desarrollar una librería
- Colaborar

Objetivos de la Charla

- Desarrollar una librería
- Colaborar
- Buenas prácticas

Objetivos de la Charla

- Desarrollar una librería
- Colaborar
- Buenas prácticas
- Herramientas útiles

Objetivos de la Charla

- Desarrollar una librería
- Colaborar
- Buenas prácticas
- Herramientas útiles
- Intercambiar conocimientos

Contenido

- Control de versiones
- Tres patas para un buen desarrollo
- Automatización
- Construir una comunidad
- ¿Puedo publicar mi librería?

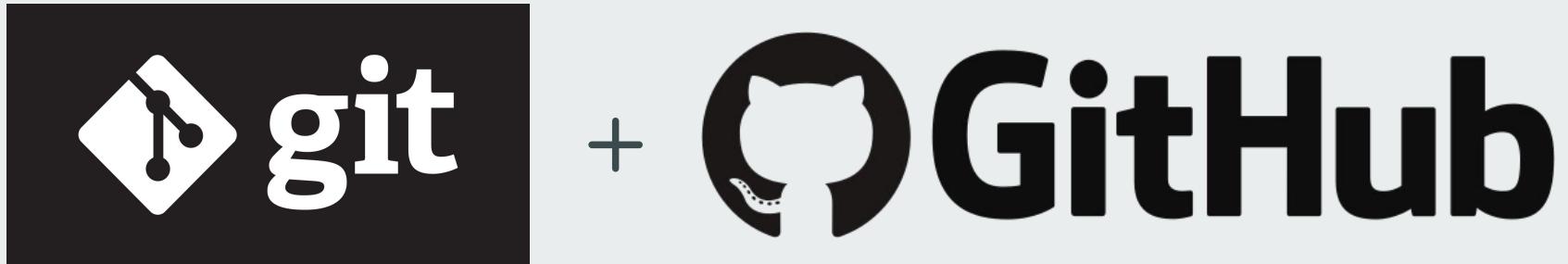
Contenido

- **Control de versiones**
- Tres patas para un buen desarrollo
- Automatización
- Construir una comunidad
- ¿Puedo publicar mi librería?

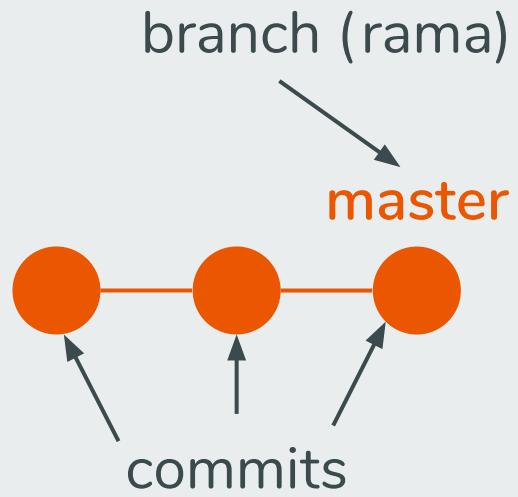
Control de Versiones

Control de Versiones
=
Registro de cambios

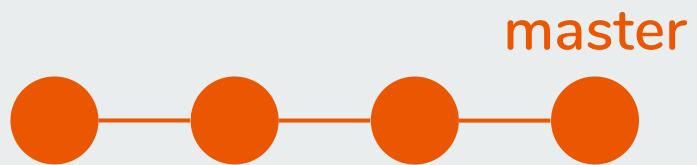
Control de Versiones



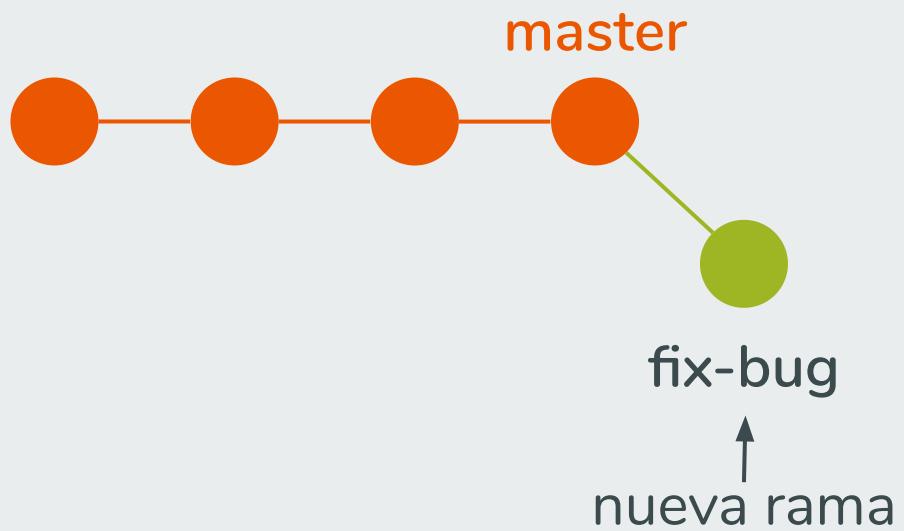
Git



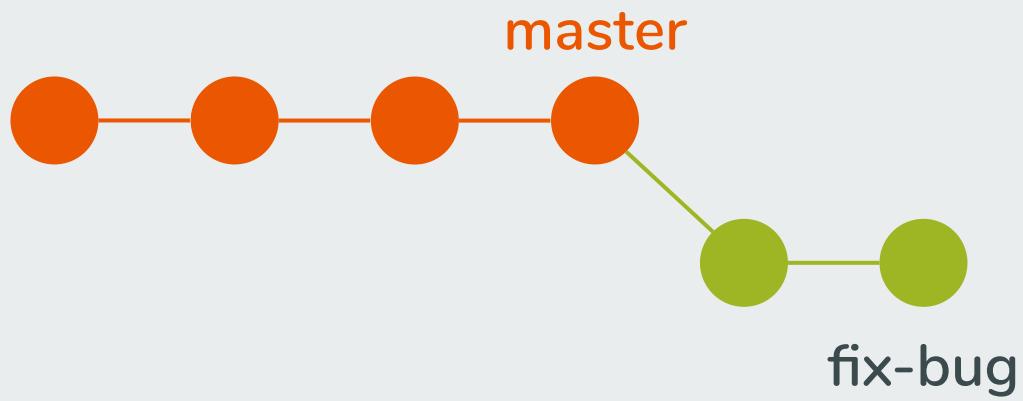
Git



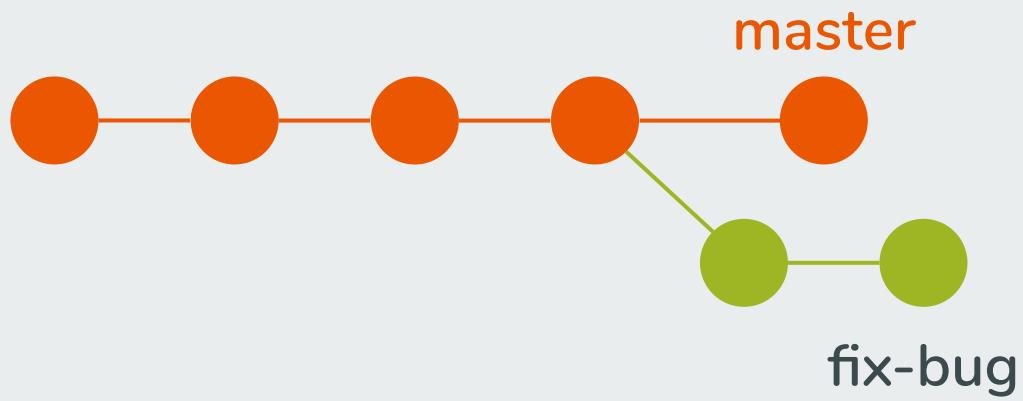
Git



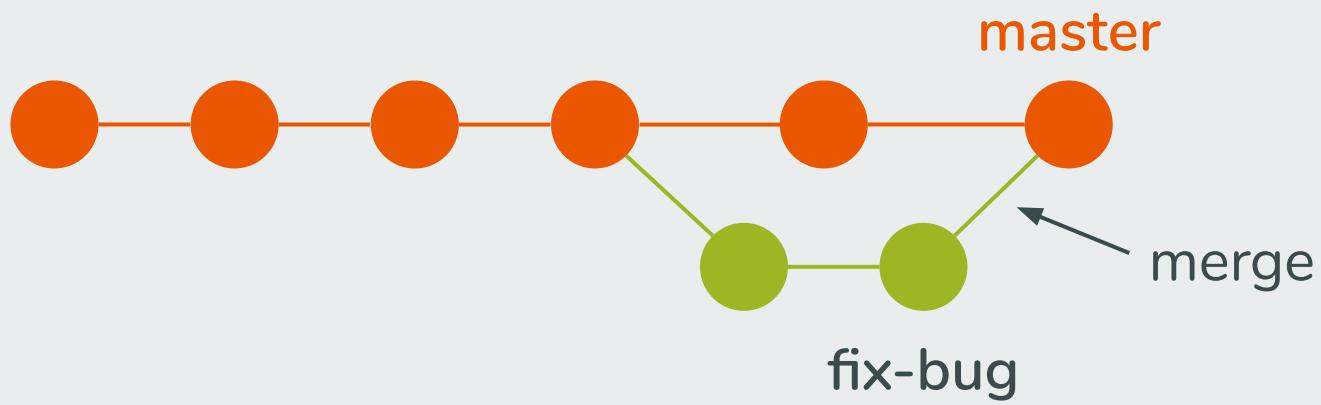
Git



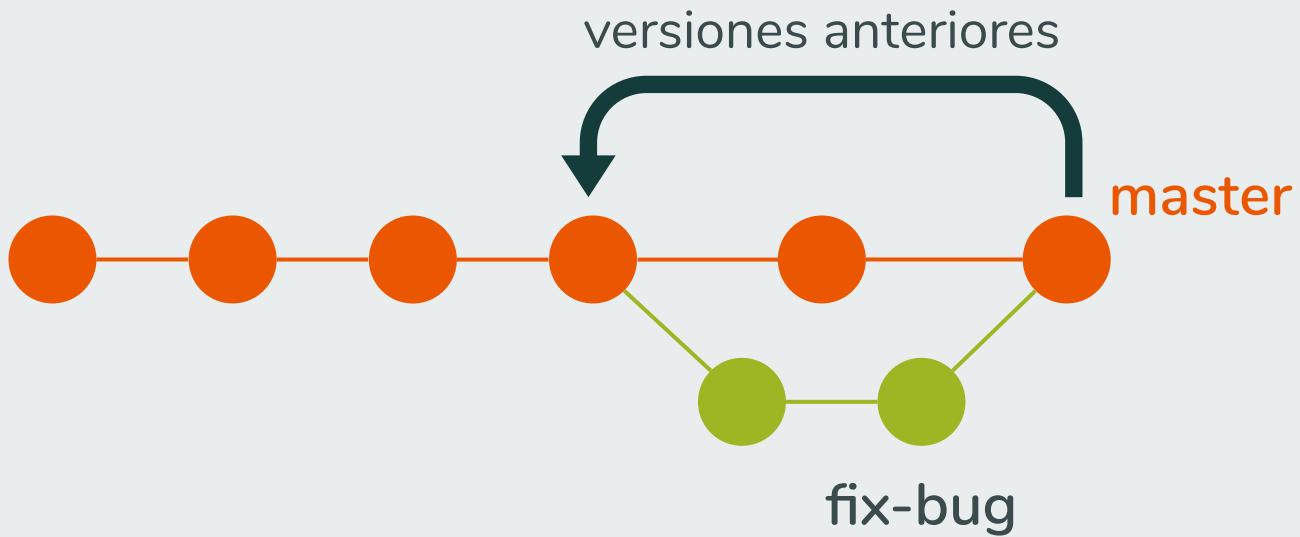
Git



Git



Git



Controlador de versiones

=

Controlador de versiones

=



+

Controlador de versiones



=

+



Commit Messages

Commit Messages

Commit messages

- ¿Qué se hizo?
- ¿Cómo se hizo?
- ¿Por qué se hizo?

Commit Messages

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

<https://xkcd.com/1296/>

Commit Messages

```
commit 2bec134ae96e20d3c61d98332d18d6d6219ff2e9
```

Author: Leonardo Uieda

Date: Mon Nov 5 16:52:22 2018 -1000

Use the xarray pcolormesh to plot grids (#151)

It automatically adjusts the coordinates to match the center of each pixel. Using matplotlib's pcolormesh would result in pixels being off by $0.5 * \text{spacing}$ unless we have pixel registered grids (which we don't).

Fixes #147

<https://github.com/fatiando/verde/commit/2bec134ae96e20d3c61d98332d18d6d6219ff2e9>

Commit Messages

```
commit 2bec134ae96e20d3c61d98332d18d6d6219ff2e9
```

Author: Leonardo Uieda

Date: Mon Nov 5 16:52:22 2018 -1000

Use the xarray pcolormesh to plot grids (#151)

It automatically adjusts the coordinates to match the center of each pixel. Using matplotlib's pcolormesh would result in pixels being off by $0.5 * \text{spacing}$ unless we have pixel registered grids (which we don't).

Fixes #147

Commit Message

Commit Messages

```
commit 2bec134ae96e20d3c61d98332d18d6d6219ff2e9
```

Author: Leonardo Uieda

Date: Mon Nov 5 16:52:22 2018 -1000

Use the xarray pcolormesh to plot grids (#151)

Línea principal

It automatically adjusts the coordinates to match the center of each pixel. Using matplotlib's pcolormesh would result in pixels being off by $0.5 * \text{spacing}$ unless we have pixel registered grids (which we don't).

Fixes #147

<https://github.com/fatiando/verde/commit/2bec134ae96e20d3c61d98332d18d6d6219ff2e9>

Commit Messages

```
commit 2bec134ae96e20d3c61d98332d18d6d6219ff2e9
```

Author: Leonardo Uieda

Date: Mon Nov 5 16:52:22 2018 -1000

Use the xarray pcolormesh to plot grids (#151)

It automatically adjusts the coordinates to match the center of each pixel. Using matplotlib's pcolormesh would result in pixels being off by $0.5 * \text{spacing}$ unless we have pixel registered grids (which we don't).

Fixes #147

Cuerpo

Cómo escribir commit messages:

<https://chris.beams.io/posts/git-commit/>



GitHub



fatiando / harmonica

Used by 1

Unwatch 3

Star 29

Fork 15

Code

Issues 13

Pull requests 4

Projects 0

Wiki

Security

Insights

Settings

Forward modeling, inversion, and processing gravity and magnetic data <https://www.fatiando.org/harmonica/dev>

Edit

geophysics

earth-science

python

scipy

fatiando-a-terra

gravity

magnetic-fields

inverse-problems

Manage topics

73 commits

7 branches

1 release

1 environment

5 contributors

BSD-3-Clause

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



leouieda Update contact information to point to Slack (#115) ...

Latest commit 43ba585 6 days ago

.github

Update the repository layout with current Verde (#2)

last year

data

Add gravity station data for South Africa (#99)

last month

doc

Update contact information to point to Slack (#115)

6 days ago

examples

Make gz effects more obvious in point mass example (#113)

9 days ago

harmonica

Avoid copy of input arrays in forward modelling (#109)

12 days ago

.azure-pipelines.yml

Take ownership of conda on Azure Mac builds (#102)

last month

.codacy.yml

Add sample airborne magnetic data from Rio (#24)

last year

GitHub



fatiando / **harmonica**

Used by 1 Unwatch 3 Star 29 Fork 15

Code Issues 13 Pull requests 4 Projects 0 Wiki Security Insights Settings

Forward modeling, inversion, and processing gravity and magnetic data <https://www.fatiando.org/harmonica/dev> Edit

geophysics earth-science python scipy fatiando-a-terra gravity magnetic-fields inverse-problems Manage topics

73 commits 7 branches 1 release 1 environment 5 contributors BSD-3-Clause

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

leouieda Update contact information to point to Slack (#115) ... Latest commit 43ba585 6 days ago

.github Update the repository layout with current Verde (#2) last year

data Add gravity station data for South Africa (#99) last month

doc Update contact information to point to Slack (#115) 6 days ago

examples Make gz effects more obvious in point mass example (#113) 9 days ago

harmonica Avoid copy of input arrays in forward modelling (#109) 12 days ago

.azure-pipelines.yml Take ownership of conda on Azure Mac builds (#102) last month

.codacy.yml Add sample airborne magnetic data from Rio (#24) last year

Issues

GitHub Issues



fatiando / harmonica

Used by 1

Unwatch 3

Star 29

Fork 15

Code

Issues 13

Pull requests 4

Projects 0

Wiki

Security

Insights

Settings

Filters

is:issue is:open

Labels 10

Milestones 1

New issue

13 Open ✓ 31 Closed

Author ▾

Labels ▾

Projects ▾

Milestones ▾

Assignee ▾

Sort ▾

! Catch numba errors about mixing int and float in forward modelling enhancement

2

hacktoberfest help wanted

#112 opened 12 days ago by leouieda

! Gallery example of gravity disturbance calculation (local) documentation help wanted

#103 opened on 19 Sep by leouieda v0.1.0

! Replace the Rio magnetic data with Great Britain open dataset enhancement

3

help wanted

#101 opened on 19 Sep by leouieda v0.1.0

! Implement the fast iterative EQL interpolator enhancement help wanted

3

#95 opened on 5 Sep by santisoler

! Enhance EQL gridded to interpolate data in spherical coordinates enhancement

help wanted

GitHub Issues



fatiando / harmonica

Used by 1

Unwatch 3

Star 29

Fork 15

Code

Issues 13

Pull requests 4

Projects 0

Wiki

Security

Insights

Settings

Filters

is:issue is:open

Labels 10

Milestones 1

New issue

13 Open ✓ 31 Closed

Author ▾

Labels ▾

Projects ▾

Milestones ▾

Assignee ▾

Sort ▾

! Catch numba errors about mixing int and float in forward modelling enhancement

2

hacktoberfest help wanted

#112 opened 12 days ago by leouieda

! Gallery example of gravity disturbance calculation (local) documentation help wanted

#103 opened on 19 Sep by leouieda v0.1.0

! Replace the Rio magnetic data with Great Britain open dataset enhancement

3

help wanted

#101 opened on 19 Sep by leouieda v0.1.0

! Implement the fast iterative EQL interpolator enhancement help wanted

3

#95 opened on 5 Sep by santisoler

! Enhance EQL gridded to interpolate data in spherical coordinates enhancement

help wanted



GitHub Issues

fatiando / harmonica

Unwatch 3

Star 29

Fork 15

Code

Issues 13

Pull requests 5

Projects 0

Wiki

Security

Insights

Settings

Implement the fast iterative EQL interpolator #95

Edit

New issue

Open

santisoler opened this issue on 5 Sep · 3 comments



santisoler commented on 5 Sep

Member



...

Description of the desired feature

Would be nice to implement the fast iterative equivalent layer technique described on [Siqueira, Oliveira Jr. and Barbosa \(2017\)](#). It could be a fast alternative to the `EQLHarmonic` gridded from [#78](#) that could be very useful when working with a very high number of data points.

The new class should inherit from `verde.base.BaseGridded` and follow its protocols. See the `EQLHarmonic` class defined on [#78](#) to get inspiration.

Are you willing to help implement and maintain this feature? Yes

Assignees

No one—assign yourself

Labels

enhancement

help wanted

Projects

None yet

Milestone

No milestone



santisoler added **enhancement** **help wanted** labels on 5 Sep



GitHub Issues

fatiando / harmonica

Code Issues 13 Pull requests 5 Projects 0 Wiki Security Insights Settings

Implement the fast iterative EQL interpolator #95

Open santisoler opened this issue on 5 Sep · 3 comments

santisoler commented on 5 Sep

Description of the desired feature

Would be nice to implement the fast iterative equivalent layer technique described on [Siqueira, Oliveira Jr. and Barbosa \(2017\)](#). It could be a fast alternative to the `EQLHarmonic` gridded from [#78](#) that could be very useful when working with a very high number of data points.

The new class should inherit from `verde.base.BaseGridded` and follow its protocols. See the `EQLHarmonic` class defined on [#78](#) to get inspiration.

Are you willing to help implement and maintain this feature? Yes

Assignees
No one—assign yourself

Labels
enhancement
help wanted

Projects
None yet

Milestone
No milestone

santisoler added **enhancement** **help wanted** labels on 5 Sep



GitHub Issues

fatiando / harmonica

Code Issues 13 Pull requests 5 Projects 0 Wiki Security Insights Settings

Implement the fast iterative EQL interpolator #95

Open santisoler opened this issue on 5 Sep · 3 comments

santisoler commented on 5 Sep

Description of the desired feature

Would be nice to implement the fast iterative equivalent layer technique described on [Siqueira, Oliveira Jr. and Barbosa \(2017\)](#). It could be a fast alternative to the `EQLHarmonic` gridded from [#78](#) that could be very useful when working with a very high number of data points.

The new class should inherit from `verde.base.BaseGridded` and follow its protocols. See the `EQLHarmonic` class defined on [#78](#) to get inspiration.

Are you willing to help implement and maintain this feature? Yes

santisoler added **enhancement** **help wanted** labels on 5 Sep

Assignees
No one—assign yourself

Labels
enhancement **help wanted**

Projects
None yet

Milestone
No milestone



GitHub Issues

Would be nice to implement the fast iterative equivalent layer technique described on [Siqueira, Oliveira Jr. and Barbosa \(2017\)](#). It could be a fast alternative to the `EQLHarmonic` gridder from [#78](#) that could be very useful when working with a very high number of data points.

The new class should inherit from `verde.base.BaseGridder` and follow its protocols. See the `EQLHarmonic` class defined on [#78](#) to get inspiration.

Are you willing to help implement and maintain this feature? Yes

santisoler added **enhancement** **help wanted** labels on 5 Sep



santisoler commented on 5 Sep

Author

Member



...

@birocoles Would you like to tackle this down? And/or maybe get one of your students involved. I can help you implementing this!



birocoles commented on 6 Sep • edited

Member



...

Hi @santisoler, I can handle this issue. I am currently trying to implement the horizontal components of point masses. After that I will address the present issue.

Labels

enhancement

help wanted

Projects

None yet

Milestone

No milestone

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants



Lock conversation

Pin issue



GitHub Issues

Would be nice to implement the fast iterative equivalent layer technique described on [Siqueira, Oliveira Jr. and Barbosa \(2017\)](#). It could be a fast alternative to the `EQLHarmonic` gridder from [#78](#) that could be very useful when working with a very high number of data points.

The new class should inherit from `verde.base.BaseGridder` and follow its protocols. See the `EQLHarmonic` class defined on [#78](#) to get inspiration.

Are you willing to help implement and maintain this feature? Yes

santisoler added **enhancement** **help wanted** labels on 5 Sep



santisoler commented on 5 Sep

Author

Member



...

@birocoles Would you like to tackle this down? And/or maybe get one of your students involved. I can help you implementing this!



birocoles commented on 6 Sep • edited

Member



...

H @santisoler, I can handle this issue. I am currently trying to implement the horizontal components of point masses. After that I will address the present issue.

Labels

enhancement

help wanted

Projects

None yet

Milestone

No milestone

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants



Lock conversation

Pin issue ⓘ

Pull Requests



GitHub Pull Requests

fatiando / harmonica

Unwatch 3 Star 29 Fork 15

Code Issues 13 Pull requests 4 Projects 0 Wiki Security Insights Settings

Filters ▾ is:pr is:open Labels 10 Milestones 1 New pull request

<input type="checkbox"/> 4 Open	<input checked="" type="checkbox"/> 67 Closed	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
<input type="checkbox"/> Adding Great Britain open mag dataset	29	#111 opened 16 days ago by nshea3 • Changes requested						
<input type="checkbox"/> WIP Add option to build a sparse Jacobian matrix on EQL	29	#106 opened 22 days ago by santisoler						
<input type="checkbox"/> WIP Add horizontal Cartesian components for point masses	74	#96 opened on 7 Sep by birocoles • Changes requested						
<input type="checkbox"/> WIP Add tesseroids layer	11	#65 opened on 30 May by santisoler • Changes requested						

💡 **ProTip!** Add `no:assignee` to see everything that's not assigned.



GitHub Pull Requests

Add forward model for prisms #86

[Edit](#)

Merged santisoler merged 58 commits into master from prism_gravity 28 days ago

[Conversation 58](#)[Commits 58](#)[Checks 7](#)[Files changed 5](#)

+594 -0



santisoler commented on 30 Jul • edited

[Member](#)[...](#)

Add forward model for prisms in Cartesian coordinates following Nagy et al. (2000) and Nagy et al. (2002). Compute the potential and the downward component of the gravitational acceleration generated by prisms on computation points. Perform non-jitted checks for valid prisms boundaries. Define a safe_atan2 function following Fukushima (2019) in order to guarantee that the generated field satisfies Poisson's equation. Use a safe_log function to allow the computation of gravitational fields on singular points of the analytical solution. Add test functions for symmetry and result comparison with an infinite slab (Bouger plate).

Fixes [#49](#)

Reminders

- Run `make format` and `make check` to make sure the code follows the style guide.
- Add tests for new features or tests that would have caught the bug that you're fixing.
-

Reviewers

stickler-ci

leouieda

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone



GitHub Pull Requests

Add forward model for prisms #86

[Edit](#)

Merged santisoler merged 58 commits into master from prism_gravity 28 days ago

[Conversation 58](#)[Commits 58](#)[Checks 7](#)[Files changed 5](#)

+594 -0



santisoler commented on 30 Jul • edited

[Member](#)[...](#)

Add forward model for prisms in Cartesian coordinates following Nagy et al. (2000) and Nagy et al. (2002). Compute the potential and the downward component of the gravitational acceleration generated by prisms on computation points. Perform non-jitted checks for valid prisms boundaries. Define a safe_atan2 function following Fukushima (2019) in order to guarantee that the generated field satisfies Poisson's equation. Use a safe_log function to allow the computation of gravitational fields on singular points of the analytical solution. Add test functions for symmetry and result comparison with an infinite slab (Bouger plate).

Fixes [#49](#)

Reminders

- Run `make format` and `make check` to make sure the code follows the style guide.
- Add tests for new features or tests that would have caught the bug that you're fixing.

Reviewers

stickler-ci

leouieda

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone



GitHub Pull Requests

Add forward model for prisms #86

[Edit](#)

Merged santisoler merged 58 commits into master from prism_gravity 28 days ago

[Conversation 58](#)[Commits 58](#)[Checks 7](#)[Files changed 5](#)

+594 -0



santisoler commented on 30 Jul • edited

[Member](#)[...](#)

Add forward model for prisms in Cartesian coordinates following Nagy et al. (2000) and Nagy et al. (2002). Compute the potential and the downward component of the gravitational acceleration generated by prisms on computation points. Perform non-jitted checks for valid prisms boundaries. Define a safe_atan2 function following Fukushima (2019) in order to guarantee that the generated field satisfies Poisson's equation. Use a safe_log function to allow the computation of gravitational fields on singular points of the analytical solution. Add test functions for symmetry and result comparison with an infinite slab (Bouger plate).

Fixes [#49](#)

Reminders

- Run `make format` and `make check` to make sure the code follows the style guide.
- Add tests for new features or tests that would have caught the bug that you're fixing.

Reviewers

stickler-ci

leouieda

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone



GitHub Pull Requests

Add forward model for prisms #86

[Edit](#)

Merged santisoler merged 58 commits into master from prism_gravity 28 days ago

[Conversation 58](#)[Commits 58](#)[Checks 7](#)[Files changed 5](#)

+594 -0



santisoler commented on 30 Jul • edited

[Member](#)[...](#)

Add forward model for prisms in Cartesian coordinates following Nagy et al. (2000) and Nagy et al. (2002). Compute the potential and the downward component of the gravitational acceleration generated by prisms on computation points. Perform non-jitted checks for valid prisms boundaries. Define a safe_atan2 function following Fukushima (2019) in order to guarantee that the generated field satisfies Poisson's equation. Use a safe_log function to allow the computation of gravitational fields on singular points of the analytical solution. Add test functions for symmetry and result comparison with an infinite slab (Bouger plate).

Fixes #49

Reminders

- Run `make format` and `make check` to make sure the code follows the style guide.
- Add tests for new features or tests that would have caught the bug that you're fixing.

Reviewers

stickler-ci

leouieda

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone



GitHub Pull Requests

Add forward model for prisms #86

[Edit](#)

Merged santisoler merged 58 commits into master from prism_gravity 28 days ago

[Conversation 58](#)[Commits 58](#)[Checks 7](#)[Files changed 5](#)

+594 -0



santisoler commented on 30 Jul • edited

[Member](#)[...](#)

Add forward model for prisms in Cartesian coordinates following Nagy et al. (2000) and Nagy et al. (2002). Compute the potential and the downward component of the gravitational acceleration generated by prisms on computation points. Perform non-jitted checks for valid prisms boundaries. Define a safe_atan2 function following Fukushima (2019) in order to guarantee that the generated field satisfies Poisson's equation. Use a safe_log function to allow the computation of gravitational fields on singular points of the analytical solution. Add test functions for symmetry and result comparison with an infinite slab (Bouger plate).

Fixes [#49](#)

Reminders

- Run `make format` and `make check` to make sure the code follows the style guide.
- Add tests for new features or tests that would have caught the bug that you're fixing.

Reviewers

- stickler-ci
- leouieda



Assignees

No one—assign yourself



Labels

None yet



Projects

None yet



Milestone

No milestone



GitHub Pull Requests



All checks have passed
11 successful checks [Show all checks](#)

This branch has no conflicts with the base branch
Merging can be performed automatically.

[Squash and merge](#) or view [command line instructions](#).

Create a merge commit
All commits from this branch will be added to the base branch via a merge commit.
Not enabled for this repository

Squash and merge
The 2 commits from this branch will be combined into one commit in the base branch.

Rebase and merge
The 2 commits from this branch will be rebased and added to the base branch.
Not enabled for this repository

ing them. [M](#)

[Close pull request](#) [Comment](#)

- Abrir Issues
- Crear Pull-Requests
- Solucionar un problema a la vez
- Solicitar y realizar reviews
- Squash and Merge

Releases



GitHub Releases

Releases Tags

Draft a new release

Latest release

v0.6.0

leouieda released this 2 days ago · 2 commits to master since this release

e1df9d4

Verified

⚠ Pooch v0.6.0 is the last release to support Python 2.7 ⚠

DOI: <https://doi.org/10.5281/zenodo.3515031>

New features:

- Add optional download progress bar to `pooch.HTTPDownloader` (#97)

Maintenance:

- Warn that 0.6.0 is the last version to support Python 2.7 (#108)

Documentation:

- Update contact information to point to our Slack channel (#107)
- Add icepack to list of projects using Pooch (#98)

This release contains contributions from:

- Daniel Shapero
- Leonardo Uieda

Edit



GitHub Releases

Releases Tags

Draft a new release

Latest release v0.6.0 Edit

leouieda released this 2 days ago · 2 commits to master since this release

⚠ Pooch v0.6.0 is the last release to support Python 2.7 ⚠

DOI: <https://doi.org/10.5281/zenodo.3515031>

New features:

- Add optional download progress bar to `pooch.HTTPDownloader` (#97)

Maintenance:

- Warn that 0.6.0 is the last version to support Python 2.7 (#108)

Documentation:

- Update contact information to point to our Slack channel (#107)
- Add icepack to list of projects using Pooch (#98)

This release contains contributions from:

- Daniel Shapero
- Leonardo Uieda



GitHub Releases

Releases Tags

v0.6.0 v0.6.0

Latest release

leouieda released this 2 days ago · 2 commits to master since this release

e1df9d4

Verified

Pooch v0.6.0 is the last release to support Python 2.7

DOI: <https://doi.org/10.5281/zenodo.3515031>

New features:

- Add optional download progress bar to `pooch.HTTPDownloader` (#97)

Maintenance:

- Warn that 0.6.0 is the last version to support Python 2.7 (#108)

Documentation:

- Update contact information to point to our Slack channel (#107)
- Add icepack to list of projects using Pooch (#98)

This release contains contributions from:

- Daniel Shapero
- Leonardo Uieda

Draft a new release

Edit

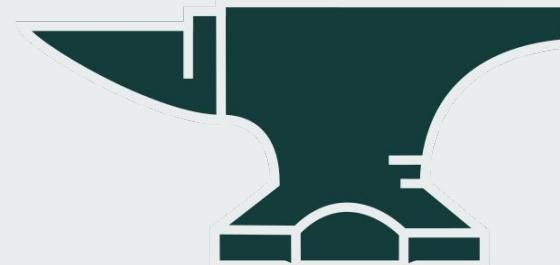
Versionado semántico
<https://semver.org/>

Releases

Descargar librería con un gestor de paquetes



```
$ pip install verde
```



```
$ conda install -c conda-forge verde
```

Releases

Obtener un DOI

DOI: Digital Object Identifier





Search



Upload

Communities



Log in

Sign up

May 21, 2019

Software Open Access

Rockhound: Download geophysical models/datasets and load them in Python

Uieda, Leonardo; Soler, Santiago R.

RockHound is a Python library to download geophysical models and datasets (PREM, CRUST1.0, ETOPO1) and load them into Python data structures (pandas, numpy, xarray).

Many of these models use non-conventional file formats or can be tricky to find on the internet. RockHound knows how to download them if you don't already have them locally, read the file format, and return a nicely formatted data structure. Under the hood, it uses Pooch to manage the downloads.

Preview

rockhound-0.1.0.zip

File	Size
.azure-pipelines.yml	5.6 kB
.codacy.yml	108 Bytes
.codeclimate.yml	556 Bytes
.codecov.yml	217 Bytes
.coveragerc	78 Bytes
.github	
ISSUE_TEMPLATE	
bug_report.md	816 Bytes
feature_request.md	622 Bytes
PULL_REQUEST_TEMPLATE.md	688 Bytes
config.yml	1.9 kB
.gitignore	244 Bytes
.pylintrc	14.8 kB
.stickler.yml	287 Bytes
.travis.yml	4.8 kB

38

views

7

downloads

[See more details...](#)

Indexed in

Publication date:

May 21, 2019

DOI:

[DOI 10.5281/zenodo.3086002](#)

Communities:

Fatiando a Terra

License (for files):

BSD 3-Clause "New" or "Revised" License

Versions

Version v0.1.0

10.5281/zenodo.3086002

May 21, 2019



Search



Upload

Communities



Log in

Sign up

May 21, 2019

Software Open Access

Rockhound: Download geophysical models/datasets and load them in Python

Uieda, Leonardo; Soler, Santiago R.

RockHound is a Python library to download geophysical models and datasets (PREM, CRUST1.0, ETOPO1) and load them into Python data structures (pandas, numpy, xarray).

Many of these models use non-conventional file formats or can be tricky to find on the internet. RockHound knows how to download them if you don't already have them locally, read the file format, and return a nicely formatted data structure. Under the hood, it uses Pooch to manage the downloads.

Preview

rockhound-0.1.0.zip

rockhound-0.1.0		5.6 kB
.azure-pipelines.yml		108 Bytes
.codacy.yml		556 Bytes
.codeclimate.yml		217 Bytes
.codecov.yml		78 Bytes
.coveragerc		
.github		
ISSUE_TEMPLATE		
bug_report.md		816 Bytes
feature_request.md		622 Bytes
PULL_REQUEST_TEMPLATE.md		688 Bytes
config.yml		1.9 kB
.gitignore		244 Bytes
.pylintrc		14.8 kB
.stickler.yml		287 Bytes
.travis.yml		4.8 kB

38

views

7

downloads

[See more details...](#)

Indexed in

OpenAIRE

Publication date:

May 21, 2019

DOI:

[DOI 10.5281/zenodo.3086002](#)

Communities:

Fatiando a Terra

License (for files):

BSD 3-Clause "New" or "Revised" License

Versions

Version v0.1.0

10.5281/zenodo.3086002

May 21, 2019

Contenido

- Control de versiones
- Tres patas para un buen desarrollo
- Automatización
- Construir una comunidad
- ¿Puedo publicar mi librería?

Tres patas para un buen desarrollo

Tres patas para un buen desarrollo

Buen código

Documentación

Testing

Tres patas para un buen desarrollo

Buen código

Documentación

Testing

Buen código

Buen código

Diseño

+

Styling

Buen código

Diseño

+

Styling

- Modular
- Funciones cortas y objetivas
- ¿Podré seguir extendiendo mi API en el futuro?

The Art of Scientific Software Maintenance:

<https://jrleeman.github.io/ScientificSoftwareMaintenance/>

Buen código

Diseño

+

Styling

¿Cómo comenzar a escribir una nueva función?

1. Ejemplo de uso.
2. Documentación para nueva función.
3. Código de nueva función.

Buen código

Diseño

+

Styling

- Código legible
- No abbreviar variables
- Estilo estandarizado: **PEP8**

<https://www.python.org/dev/peps/pep-0008/>

Buen código

Diseño

+

Styling

Herramientas útiles

- flake8
- pylint
- black

Buen código

Diseño

+

Styling

Herramientas útiles

- flake8
 - pylint
 - black
-
- ```
graph LR; A[flake8] --> B[checkers]; C pylint --> B; D[black] --> E[autoformateador]
```

# Tres patas para un buen desarrollo

Buen código

Documentación

Testing

# Documentación

# Documentación

- Documentamos todas las funciones.
- Estilo estandarizado: **numpy style**

Numpy docstring style guide:

<https://numpydoc.readthedocs.io/en/latest/format.html>

- Referencias a artículos científicos

# Documentación

```
def geodetic_to_spherical(longitude, latitude, height):
 """
 Convert from geodetic to geocentric spherical coordinates.

 The geodetic datum is defined by the default :class:`harmonica.ReferenceEllipsoid`
 set by the :func:`harmonica.set_ellipsoid` function.
 The coordinates are converted following [Vermeille2002]_.

 Parameters

 longitude : array
 Longitude coordinates on geodetic coordinate system in degrees.
 latitude : array
 Latitude coordinates on geodetic coordinate system in degrees.
 height : array
 Ellipsoidal heights in meters.

 Returns

 longitude : array
 Longitude coordinates on geocentric spherical coordinate system in degrees.
 The longitude coordinates are not modified during this conversion.
 spherical_latitude : array
 Converted latitude coordinates on geocentric spherical coordinate system in
 degrees.
 radius : array
 Converted spherical radius coordinates in meters.

 See also

 spherical_to_geodetic : Convert from geocentric spherical to geodetic coordinates.
```

# Documentación

```
def geodetic_to_spherical(longitude, latitude, height):
 """
 Convert from geodetic to geocentric spherical coordinates.

 The geodetic datum is defined by the default :class:`harmonica.ReferenceEllipsoid`
 set by the :func:`harmonica.set_ellipsoid` function.
 The coordinates are converted following [Vermeille2002]_.

 Parameters

 longitude : array
 Longitude coordinates on geodetic coordinate system in degrees.
 latitude : array
 Latitude coordinates on geodetic coordinate system in degrees.
 height : array
 Ellipsoidal heights in meters.

 Returns

 longitude : array
 Longitude coordinates on geocentric spherical coordinate system in degrees.
 The longitude coordinates are not modified during this conversion.
 spherical_latitude : array
 Converted latitude coordinates on geocentric spherical coordinate system in
 degrees.
 radius : array
 Converted spherical radius coordinates in meters.

 See also

 spherical_to_geodetic : Convert from geocentric spherical to geodetic coordinates.
```

Breve resumen

# Documentación

```
def geodetic_to_spherical(longitude, latitude, height):
 """
 Convert from geodetic to geocentric spherical coordinates. Resumen extendido
 The geodetic datum is defined by the default :class:`harmonica.ReferenceEllipsoid` set by the :func:`harmonica.set_ellipsoid` function.
 The coordinates are converted following [Vermeille2002].
 """

 Parameters

 longitude : array
 Longitude coordinates on geodetic coordinate system in degrees.
 latitude : array
 Latitude coordinates on geodetic coordinate system in degrees.
 height : array
 Ellipsoidal heights in meters.

 Returns

 longitude : array
 Longitude coordinates on geocentric spherical coordinate system in degrees.
 The longitude coordinates are not modified during this conversion.
 spherical_latitude : array
 Converted latitude coordinates on geocentric spherical coordinate system in degrees.
 radius : array
 Converted spherical radius coordinates in meters.

 See also

 spherical_to_geodetic : Convert from geocentric spherical to geodetic coordinates.
```

# Documentación

```
def geodetic_to_spherical(longitude, latitude, height):
 """
 Convert from geodetic to geocentric spherical coordinates. Resumen extendido
 The geodetic datum is defined by the default :class:`harmonica.ReferenceEllipsoid`

 set by the :func:`harmonica.set_ellipsoid` function.

 The coordinates are converted following [Vermeille2002].
```

**Parameters**

-----

- longitude : array  
Longitude coordinates on geodetic coordinate system in degrees.
- latitude : array  
Latitude coordinates on geodetic coordinate system in degrees.
- height : array  
Ellipsoidal heights in meters.

**Returns**

-----

- longitude : array  
Longitude coordinates on geocentric spherical coordinate system in degrees.  
The longitude coordinates are not modified during this conversion.
- spherical\_latitude : array  
Converted latitude coordinates on geocentric spherical coordinate system in degrees.
- radius : array  
Converted spherical radius coordinates in meters.

**See also**

-----

- spherical\_to\_geodetic : Convert from geocentric spherical to geodetic coordinates.

# Documentación

```
def geodetic_to_spherical(longitude, latitude, height):
 """
 Convert from geodetic to geocentric spherical coordinates.

 The geodetic datum is defined by the default :class:`harmonica.ReferenceEllipsoid`
 set by the :func:`harmonica.set_ellipsoid` function.
 The coordinates are converted following [Vermeille2002].
 """

 Parameters

 longitude : array
 Longitude coordinates on geodetic coordinate system in degrees.
 latitude : array
 Latitude coordinates on geodetic coordinate system in degrees.
 height : array
 Ellipsoidal heights in meters.

 Returns

 longitude : array
 Longitude coordinates on geocentric spherical coordinate system in degrees.
 The longitude coordinates are not modified during this conversion.
 spherical_latitude : array
 Converted latitude coordinates on geocentric spherical coordinate system in
 degrees.
 radius : array
 Converted spherical radius coordinates in meters.

 See also

 spherical_to_geodetic : Convert from geocentric spherical to geodetic coordinates.
```

## Parámetros de entrada

# Documentación

```
def geodetic_to_spherical(longitude, latitude, height):
 """
 Convert from geodetic to geocentric spherical coordinates.

 The geodetic datum is defined by the default :class:`harmonica.ReferenceEllipsoid` set by the :func:`harmonica.set_ellipsoid` function.
 The coordinates are converted following [Vermeille2002]_.

 Parameters

 longitude : array
 Longitude coordinates on geodetic coordinate system in degrees.
 latitude : array
 Latitude coordinates on geodetic coordinate system in degrees.
 height : array
 Ellipsoidal heights in meters.

 Returns

 longitude : array
 Longitude coordinates on geocentric spherical coordinate system in degrees.
 The longitude coordinates are not modified during this conversion.
 spherical_latitude : array
 Converted latitude coordinates on geocentric spherical coordinate system in degrees.
 radius : array
 Converted spherical radius coordinates in meters.

 See also

 spherical_to_geodetic : Convert from geocentric spherical to geodetic coordinates.
```

## Parámetros de salida

# Documentación

See also

spherical\_to\_geodetic : Convert from geocentric spherical to geodetic coordinates.

Examples

## Ejemplos simples (opcional)

In the poles, the radius should be the reference ellipsoid's semi-minor axis:

```
>>> import harmonica as hm
>>> spherical = hm.geodetic_to_spherical(longitude=0, latitude=90, height=0)
>>> print(", ".join("{:.4f}".format(i) for i in spherical))
0.0000, 90.0000, 6356752.3142
>>> print("{:.4f}".format(hm.get_ellipsoid().semiminor_axis))
6356752.3142
```

In the equator, it should be the semi-major axis:

```
>>> spherical = hm.geodetic_to_spherical(longitude=0, latitude=0, height=0)
>>> print(", ".join("{:.4f}".format(i) for i in spherical))
0.0000, 0.0000, 6378137.0000
>>> print("{:.4f}".format(hm.get_ellipsoid().semimajor_axis))
6378137.0000
```

"""

```
Get ellipsoid
ellipsoid = get_ellipsoid()
Convert latitude to radians
latitude_rad = np.radians(latitude)
prime_vertical_radius = ellipsoid.semimajor_axis / np.sqrt(
 1 - ellipsoid.first_eccentricity ** 2 * np.sin(latitude_rad) ** 2)
```

# Documentación

See also

spherical\_to\_geodetic : Convert from geocentric spherical to geodetic coordinates.

Examples

In the poles, the radius should be the reference ellipsoid's semi-minor axis:

```
>>> import harmonica as hm
>>> spherical = hm.geodetic_to_spherical(longitude=0, latitude=90, height=0)
>>> print(", ".join("{:.4f}".format(i) for i in spherical))
0.0000, 90.0000, 6356752.3142
>>> print("{:.4f}".format(hm.get_ellipsoid().semiminor_axis))
6356752.3142
```

In the equator, it should be the semi-major axis:

```
>>> spherical = hm.geodetic_to_spherical(longitude=0, latitude=0, height=0)
>>> print(", ".join("{:.4f}".format(i) for i in spherical))
0.0000, 0.0000, 6378137.0000
>>> print("{:.4f}".format(hm.get_ellipsoid().semimajor_axis))
6378137.0000
```

"""

```
Get ellipsoid
ellipsoid = get_ellipsoid()
Convert latitude to radians
latitude_rad = np.radians(latitude)
prime_vertical_radius = ellipsoid.semimajor_axis / np.sqrt(
 1 - ellipsoid.first_eccentricity ** 2 * np.sin(latitude_rad) ** 2)
```

Código

# Compilar la documentación



+

The GitHub Pages logo, which consists of the GitHub octocat icon followed by the text "GitHub Pages".

# Compilar la documentación

VERDE  
v1.2.0

Docs » Home

Improve this page

**GETTING STARTED**

- Overview
- Installing
- Citing Verde
- Gallery

**USER GUIDE**

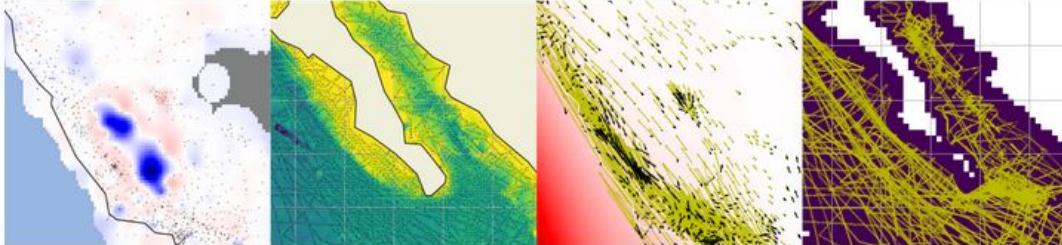
- Sample Data
- Grid Coordinates
- Trend Estimation
- Data Decimation
- Geographic Coordinates
- Chaining Operations
- Evaluating Performance
- Model Selection
- Using Weights
- Vector Data

**REFERENCE DOCUMENTATION**

- API Reference
- Changelog
- References

**GETTING HELP AND CONTRIBUTING**

- Fatiando a Terra
- Contributing
- Code of Conduct



## About

Verde is a Python library for processing spatial data (bathymetry, geophysics surveys, etc) and interpolating it on regular grids (i.e., *gridding*).

Most gridding methods in Verde use a Green's functions approach. A linear model is estimated based on the input data and then used to predict data on a regular grid (or in a scatter, a profile, as derivatives). The models are Green's functions from (mostly) elastic deformation theory. This approach is very similar to *machine learning* so we implement gridded classes that are similar to [scikit-learn](#) regression classes. The API is not 100% compatible but it should look familiar to those with some scikit-learn experience.

<https://www.fatiando.org/verde>

# Compilar la documentación

- Model Selection
- Using Weights
- Vector Data
- REFERENCE DOCUMENTATION**
- API Reference**
  - Interpolators
  - Data Processing
  - Composite Estimators
  - Model Selection
- Coordinate Manipulation**
  - verde.grid\_coordinates**
    - Examples using verde.grid\_coordinates
  - verde.scatter\_points
  - verde.profile\_coordinates
  - verde.get\_region
  - verde.pad\_region
  - verde.project\_region
  - verde.inside
  - verde.block\_split
- Utilities
- Input/Output
- Datasets
- Base Classes and Functions

Changelog

References

**GETTING HELP AND CONTRIBUTING**

- Fatiando a Terra
- Contributing
- Code of Conduct

[Docs](#) » [API Reference](#) » `verde.grid_coordinates`

[!\[\]\(35e0b177ebd89a5d4fef67a6ac01e20b\_img.jpg\) Improve this page](#)

## verde.grid\_coordinates

```
verde.grid_coordinates(region, shape=None, spacing=None, adjust='spacing', pixel_register=False, extra_coords=None) [source]
```

Generate the coordinates for each point on a regular grid.

The grid can be specified by either the number of points in each dimension (the `shape`) or by the grid node spacing.

If the given region is not divisible by the desired spacing, either the region or the spacing will have to be adjusted. By default, the spacing will be rounded to the nearest multiple. Optionally, the East and North boundaries of the region can be adjusted to fit the exact spacing given. See the examples below.

**Parameters:** `region : list = [W, E, S, N]`

The boundaries of a given region in Cartesian or geographic coordinates.

`shape : tuple = (n_north, n_east) or None`

The number of points in the South-North and West-East directions, respectively.

`spacing : float, tuple = (s_north, s_east), or None`

The grid spacing in the South-North and West-East directions, respectively. A single value means that the spacing is equal in both directions.

`adjust : {'spacing', 'region'}`

Whether to adjust the spacing or the region if required. Ignored if `shape` is given instead of `spacing`. Defaults to adjusting the spacing.

`pixel_register : bool`

If True, the coordinates will refer to the center of each grid pixel instead of the grid lines. In practice, this means that there will be one less element per dimension of the grid when compared to grid line registered (only if given `spacing` and not `shape`). Default is False.

`extra_coords : None, scalar, or list`

If not None, then value(s) of extra coordinate arrays to be generated. These extra arrays will have the same `shape` as the others but will contain a constant value. Will generate an extra array per value given in `extra_coords`. Use this to generate arrays of constant heights or times, for example, that might be needed to evaluate a gridded.

- Documentar todas las funciones
- Estilo estandarizado
- Compilar documentación

# Tres patas para un buen desarrollo

Buen código

Documentación

Testing

# Testing

# Testing

```
def range_overlap(ranges):
 '''Return common overlap among a set of [left, right] ranges.'''
 max_left = 0.0
 min_right = 1.0
 for (left, right) in ranges:
 max_left = max(max_left, left)
 min_right = min(min_right, right)
 return (max_left, min_right)

def test_range_overlap():
 assert range_overlap([(0.0, 1.0), (5.0, 6.0)]) == None
 assert range_overlap([(0.0, 1.0), (1.0, 2.0)]) == None
 assert range_overlap([(0.0, 1.0)]) == (0.0, 1.0)
 assert range_overlap([(2.0, 3.0), (2.0, 4.0)]) == (2.0, 3.0)
 assert range_overlap([(0.0, 1.0), (0.0, 2.0), (-1.0, 1.0)]) == (0.0, 1.0)
 assert range_overlap([]) == None
```

# ¿Qué testear?

Funcionamiento

Resultados

# ¿Qué testear?

Funcionamiento

Resultados

- ¿Funciona bajo parámetros válidos?
- ¿Levanta error bajo parámetros inválidos?

# ¿Qué testear?

Funcionamiento

Resultados

- **Conocemos resultado esperado**
  - ¿Reproduce resultado esperado?
- **No conocemos resultado esperado**
  - ¿Satisface propiedades conocidas?
  - ¿Aproxima a solución analítica?

# ¿Cómo ejecutar los testeos?

pytest

+

coverage

- Ejecutamos todos las funciones de testeo
- Reporte de fallos
- Reporte de cobertura de testeo

# Pytest

```
[santi@eevaa harmonica] harmonica master $ make test
Run a tmp folder to make sure the tests are run on the installed version
mkdir -p tmp-test-dir-with-unique-name
cd tmp-test-dir-with-unique-name; NUMBA_DISABLE_JIT=1 MPLBACKEND='agg' pytest --cov-config=../.coveragerc --cov-report=term-missing --cov=harmonica --doctest-modules -v --pyargs harmonica
===== test session starts =====
platform linux -- Python 3.7.3, pytest-5.2.2, py-1.8.0, pluggy-0.13.0 -- /home/santi/.miniconda3/envs/harmonica/bin/python
cachedir: .pytest_cache
rootdir: /home/santi/git/harmonica, ini file: pytest.ini
plugins: cov-2.8.1
collected 97 items

./coordinates.py::harmonica.coordinates.geodetic_to_spherical PASSED [1%]
./coordinates.py::harmonica.coordinates.spherical_to_geodetic PASSED [2%]
./ellipsoid.py::harmonica.ellipsoid.ReferenceEllipsoid PASSED [3%]
./ellipsoid.py::harmonica.ellipsoid.get_ellipsoid PASSED [4%]
./ellipsoid.py::harmonica.ellipsoid.print_ellipsoids PASSED [5%]
./ellipsoid.py::harmonica.ellipsoid.set_ellipsoid PASSED [6%]
./forward/prism.py::harmonica.forward.prism.prism_gravity PASSED [7%]
./forward/tesseroid.py::harmonica.forward.tesseroid.tesseroid_gravity PASSED [8%]
./tests/test_coordinates.py::test_geodetic_to_spherical_with_spherical_ellipsoid PASSED [9%]
./tests/test_coordinates.py::test_geodetic_to_spherical_on_equator PASSED [10%]
./tests/test_coordinates.py::test_geodetic_to_spherical_on_poles PASSED [11%]
./tests/test_coordinates.py::test_spherical_to_geodetic_with_spherical_ellipsoid PASSED [12%]
./tests/test_coordinates.py::test_spherical_to_geodetic_on_equator PASSED [13%]
./tests/test_coordinates.py::test_spherical_to_geodetic_on_poles PASSED [14%]
./tests/test_coordinates.py::test_spherical_to_geodetic_identity PASSED [15%]
```

# Coverage

| coverage: platform: linux, python: 3.7.3-final-0                                      |       |      |        |        |       |         |
|---------------------------------------------------------------------------------------|-------|------|--------|--------|-------|---------|
| Name                                                                                  | Stmts | Miss | Branch | BrPart | Cover | Missing |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/constants.py                 | 1     | 0    | 0      | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/coordinates.py               | 32    | 0    | 0      | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/datasets/sample_data.py      | 38    | 0    | 0      | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/ellipsoid.py                 | 68    | 0    | 12     | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/equivalent_layer/harmonic.py | 51    | 0    | 12     | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/forward/point_mass.py        | 50    | 0    | 18     | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/forward/prism.py             | 74    | 0    | 42     | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/forward/tesseroid.py         | 210   | 0    | 86     | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/forward/utils.py             | 33    | 0    | 6      | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/gravity_corrections.py       | 32    | 0    | 0      | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/io.py                        | 65    | 0    | 40     | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/isostasy.py                  | 17    | 0    | 2      | 0      | 100%  |         |
| /home/santi/.local/lib/python3.7/site-packages/harmonica/version.py                   | 4     | 0    | 0      | 0      | 100%  |         |
| TOTAL                                                                                 | 675   | 0    | 218    | 0      | 100%  |         |

- Testteamos todas las funciones
- Coverage 100%
- Testeos robustos

# Contenido

- Control de versiones
- Tres patas para un buen desarrollo
- Automatización
- Construir una comunidad
- ¿Puedo publicar mi librería?

# Automatización

# Automatización

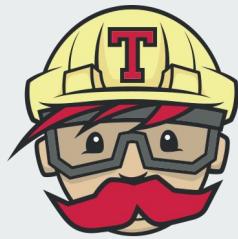
=

# Tiempo Libre

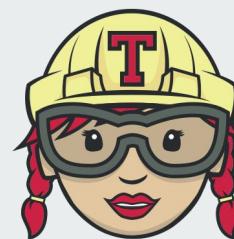
# Continuous Integration



+



Travis CI



Azure Pipelines

# ¿Qué automatizamos?

- Ejecutar testeos



# ¿Qué automatizamos?

- Ejecutar testeos



- Chequear styling



# CI: Testeos y styling en Pull Requests

The screenshot shows a GitHub pull request interface with the following details:

- Review requested**: A green circle with a checkmark icon. Text: "Review has been requested on this pull request. It is not required to merge. [Learn more](#)".
- 1 pending reviewer**: A user icon followed by the text "1 pending reviewer".
- All checks have passed**: A green circle with a checkmark icon. Text: "All checks have passed" and "14 successful checks".
- Continuous Integration Checks (boxed)**:
  - continuous-integration/travis-ci/pr — The Travis CI build passed Required [Details](#)
  - fatiando.verde — #20191125.6 succeeded Required [Details](#)
  - fatiando.verde (Mac Python36) — Successful in 3m — Mac Python36 succeed... [Details](#)
  - fatiando.verde (Mac Python36-Optional) — Successful in 3m — Mac Python... [Details](#)
  - fatiando.verde (Mac Python37) — Successful in 3m — Mac Python37 succeed... [Details](#)
  - fatiando.verde (Mac Python37-Optional) — Successful in 3m — Mac Python... [Details](#)
- This branch has no conflicts with the base branch**: A green circle with a checkmark icon. Text: "Merging can be performed automatically."
- Action Buttons**:
  - Squash and merge** (green button with dropdown arrow)
  - or view [command line instructions](#).

# ¿Qué automatizamos?

- Ejecutar testeos



- Chequear styling



- Compilar documentación



# ¿Qué automatizamos?

- Ejecutar testeos



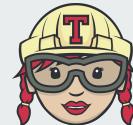
- Chequear styling



- Compilar documentación



- Publicar nueva documentación



# ¿Qué automatizamos?

- Ejecutar testeos



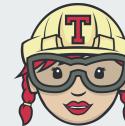
- Chequear styling



- Compilar documentación



- Publicar nueva documentación



- Publicar nuevo release en Pypi



# Scripts para Continuous Integration:



[fatiando/continuous-integration](https://github.com/fatiando/continuous-integration)

# Contenido

- Control de versiones
- Tres patas para un buen desarrollo
- Automatización
- Construir una comunidad
- ¿Puedo publicar mi librería?

# Construir una comunidad

# Incentivar a otros a colaborar

- Código de conducta
- Guías para contribuir
- Solicitar ayuda en Issues
- Dar crédito

<https://github.com/fatiando/contributing/>

# Comunicación



Sitio web



Reuniones

# Contenido

- Control de versiones
- Tres patas para un buen desarrollo
- Automatización
- Construir una comunidad
- ¿Puedo publicar mi librería?

¿Puedo publicar mi librería?



<https://joss.theoj.org>



The Journal of  
Open Source Software

About

Papers

Docs

Bloc

Submit



The Journal of Open Source Software is a developer friendly, open access journal for research software packages.

Committed to publishing quality research software with zero article processing charges or subscription fees.

[Submit a paper to JOSS](#)

 Volunteer to review

 Explore Papers

## ☐ Documentation

 Learn More

## Recently Published Papers 764

PUBLISHED

Published about 2 hours ago

Virtual Scanner: MRI on a Browser

JavaScript Python



@imr-framework

DOI: 10.21105/ioss.01637

# JOSS: The Journal of Open Source Software

- Peer review abierto
- Review de código
- GitHub
- Publicación rápida
- Instrucciones y requerimientos claros



Adina Wagner  
@AdinaKrik

It's been an uplifting experience, and the review process at [@JOSS\\_TheOJ](#) has been **pure joy.** I can't believe anyone (incl. me) ever trusted me doing this - but I'm very glad and thankful for all the encouragement and help! Whoever needs to hear it: If I can do this, you can too! 😎



Tom Faulkenberry  
@tomfaulkenberry

I've said it before, but it bears repeated frequently. Reviewing for [@JOSS\\_TheOJ](#) is a **really cool** experience.



Ethan White  
@ethanwhite

Reviewing for [@JOSS\\_TheOJ](#) is definitely **the most satisfying review work** I do. Sign up to help out!

# Conclusiones

- Controlador de versiones
- GitHub: Issues y Pull Requests
- Buen código
- Documentación
- Testing
- Automatización
- Comunidad

¡Muchas Gracias!

# Descargá las diapositivas



<https://doi.org/10.6084/m9.figshare.10013006>



Contenido bajo  
Creative Commons Attribution 4.0