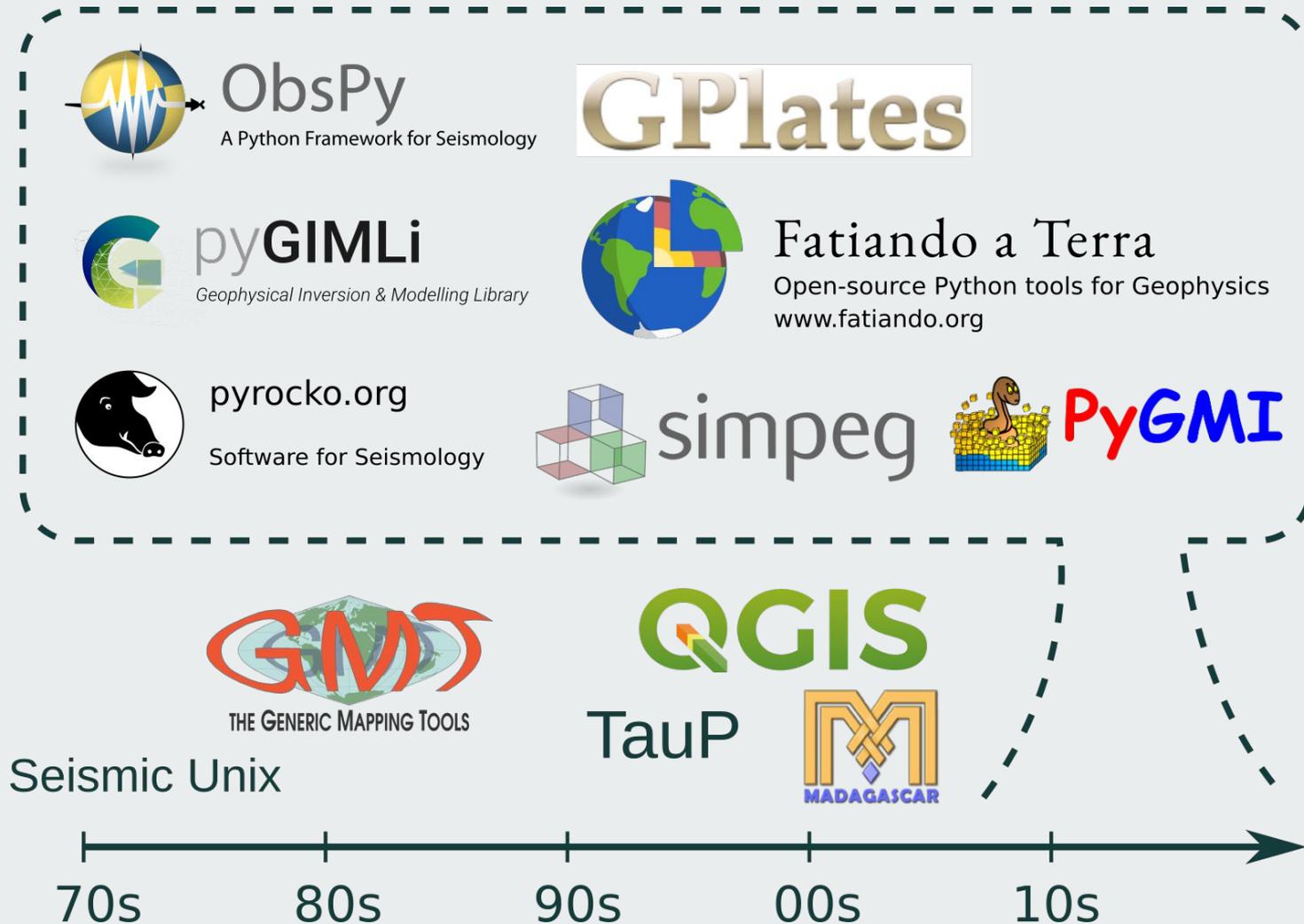# Experiencias en el desarrollo de Fatiando a Terra

Santiago Soler

Computer-Oriented Geoscience Lab
IGSV, UNSJ
CONICET

Librerías de código abierto en Geofísica

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: 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.
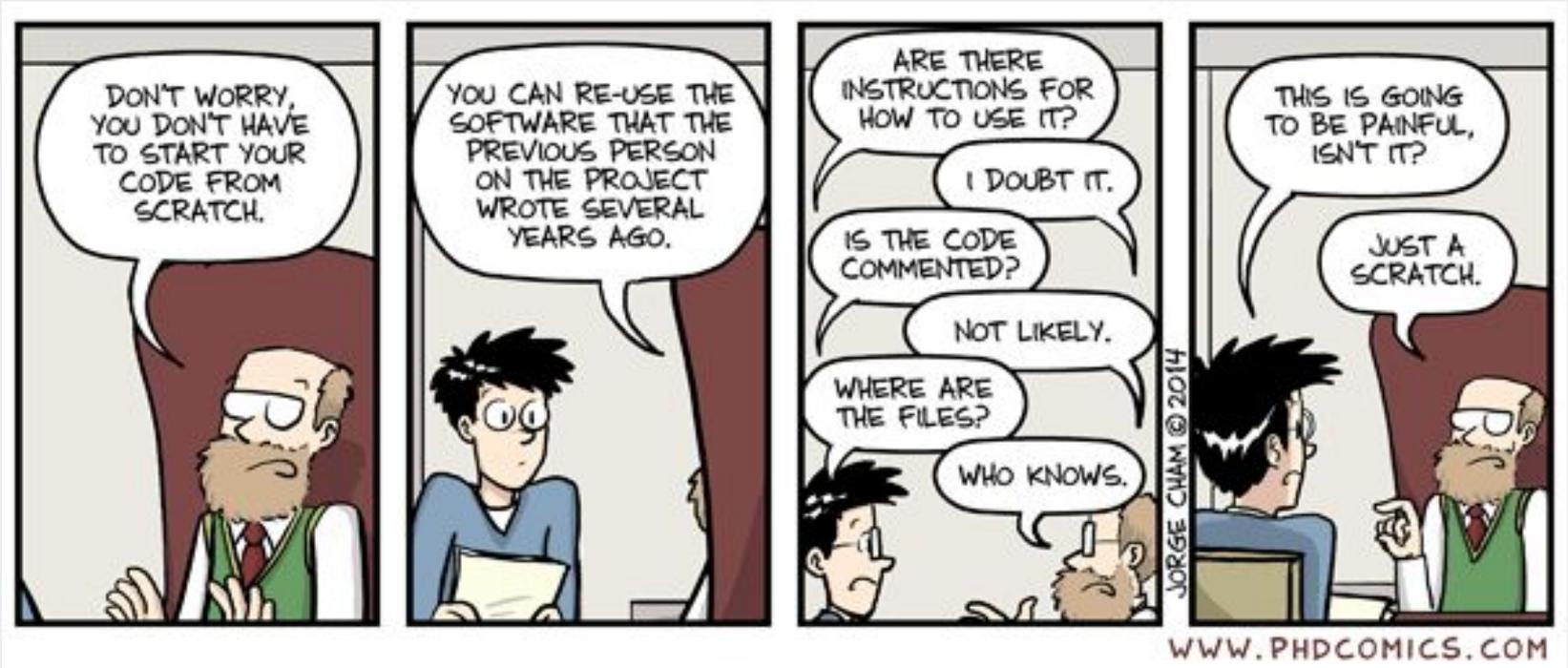
- fatiando/rockhound
- www.fatiando.org/rockhound
- ↻ Ready for use but still changing

¿Por qué desarrollar una librería de código abierto?

"Scratch" by Jorge Cham, http://www.phdcomics.com

No te preocupes, 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

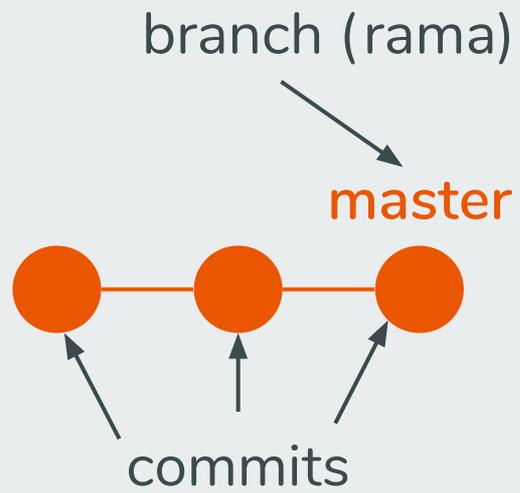- ¿Puedo publicar mi librería?

Contenido

Control de Versiones

Control de Versiones
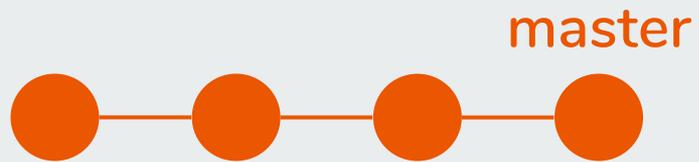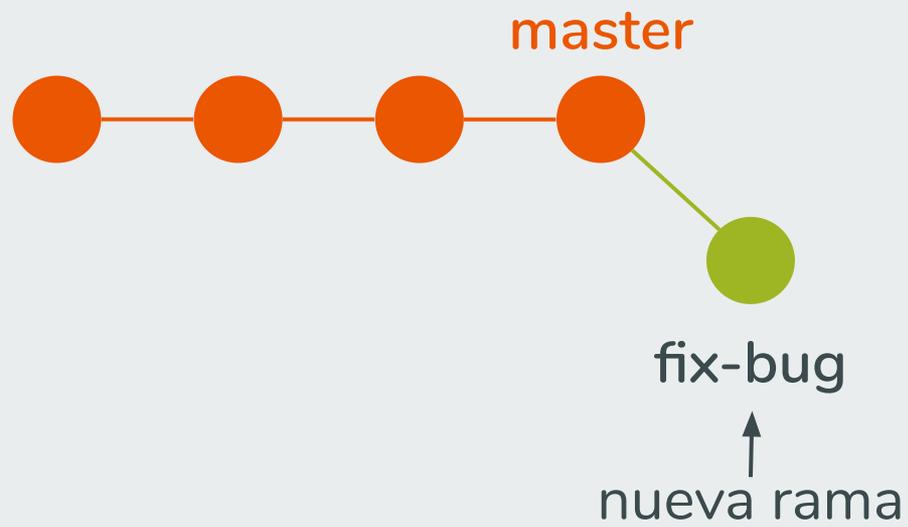# =
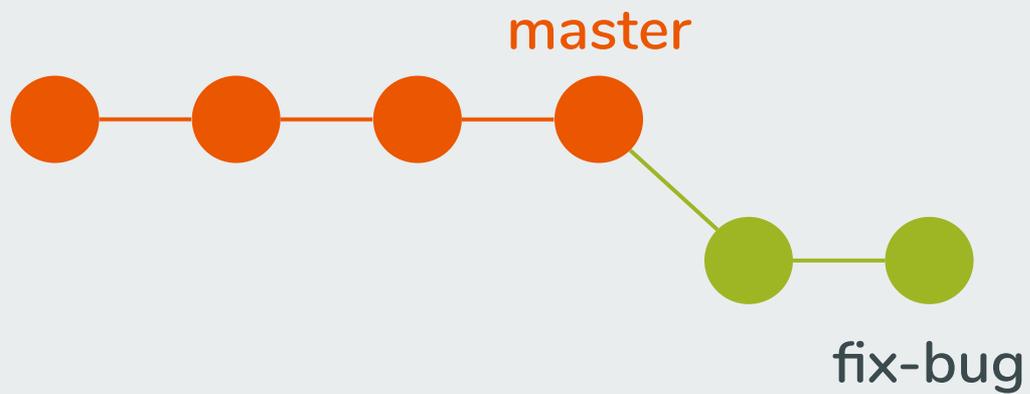# Registro de cambios

Control de Versiones

Git

branch (rama)

master

commits
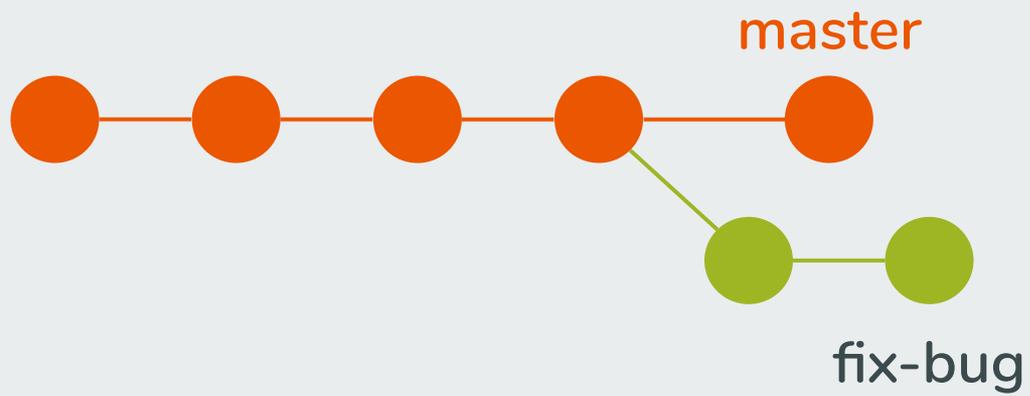
Git

Git

Git

Git
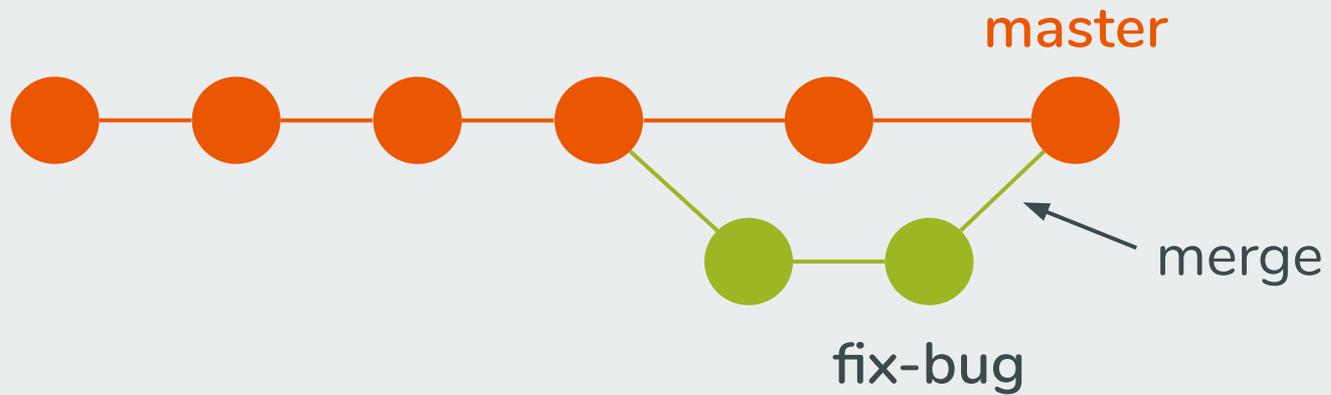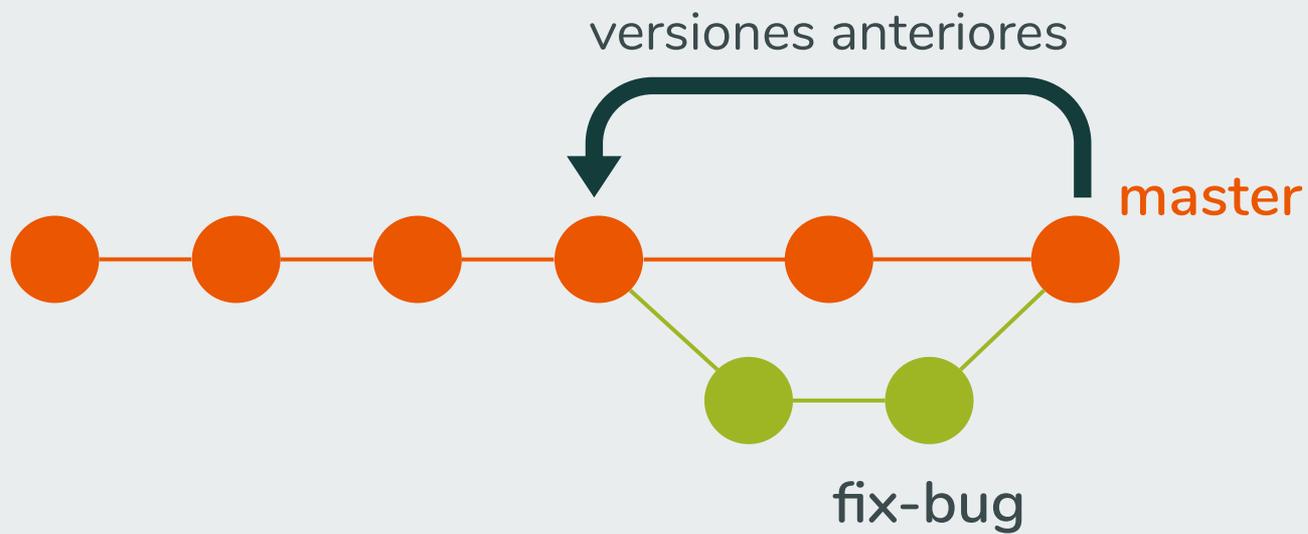
master

fix-bug

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
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	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*spacing unless we have pixel
registered grids (which we don't).

Fixes #147
```

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*spacing unless we have pixel
registered grids (which we don't).

Fixes #147
```

Commit Message

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)   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*spacing unless we have pixel
registered grids (which we don't).

Fixes #147
```

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*spacing unless we have pixel
registered grids (which we don't).

Fixes #147
```

Cuerpo

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

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

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 gridder 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 gridder 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` 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

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

👁 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` 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

Assignees ⚙
No one—assign yourself

Labels ⚙
enhancement
help wanted

Projects ⚙
None yet

Milestone ⚙
No milestone

GitHub Issues

fatiando / harmonica

Unwatch ▾ 3 | ★ Star 29 | ⑂ Fork 15

<> Code | ⊙ Issues **13** | ⋔ Pull requests **5** | ▥ Projects **0** | ▤ Wiki | 🛡 Security | �ⅠⅠⅠ 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` 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

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

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   �ⅼⅼ Insights   ⚙ Settings

Filters ▾   🔍 is:pr is:open   🏷 Labels **10**   ⯒ Milestones **1**   **New pull request**

☐   ⑂ **4 Open**   ✓ **67 Closed**   Author ▾   Labels ▾   Projects ▾   Milestones ▾   Reviews ▾   Assignee ▾   Sort ▾

☐   ⑂ **Adding Great Britain open mag dataset** ✗   💬 29
　　#111 opened 16 days ago by nshea3 · Changes requested   📝 5 of 5 ━━━━

☐   ⑂ **WIP Add option to build a sparse Jacobian matrix on EQL** •
　　#106 opened 22 days ago by santisoler   📝 0 of 5 ━━━━

☐   ⑂ **WIP Add horizontal Cartesian components for point masses** •   💬 74
　　#96 opened on 7 Sep by birocoles · Changes requested   📝 3 of 5 ━━━━

☐   ⑂ **WIP Add tesseroids layer** ✗   💬 11
　　#65 opened on 30 May by santisoler · Changes requested   📝 0 of 5 ━━━━

💡 **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

GitHub Pull Requests

GitHub Pull Requests
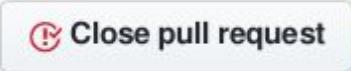
GitHub Pull Requests

GitHub Pull Requests

- Abrir Issues

- Crear Pull-Requests

- Issues y PRs descriptivos

- Solucionar un problema a la vez

- Solicitar y realizar reviews

- Squash and Merge

Releases

GitHub Releases

GitHub Releases

GitHub Releases

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

zenodo

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
  ○ 📄 .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

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   May 21, 2019

- **Releases periódicas**

- **Versionado semántico**

- **Gestor de paquetes**

- DOI

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 abreviar 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 $\longrightarrow$ checkers

- black $\longrightarrow$ 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

```python
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

```python
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

```python
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

```python
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                              Parámetros de entrada
    ----------
    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

```python
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
--------

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
```

Ejemplos simples (opcional)

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

+

Compilar la documentación

Compilar la documentación

- **Documentar __todas__ las funciones**

- **Estilo estandarizado**

- **Compilar documentación**

Tres patas para un buen desarrollo

Buen código

Documentación

Testing

Testing

Testing

```python
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

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%
```

- **Testeamos <u>todas</u> 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

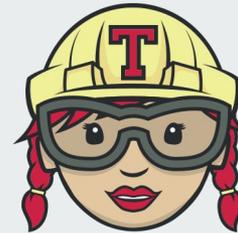# Automatización

=

Tiempo Libre
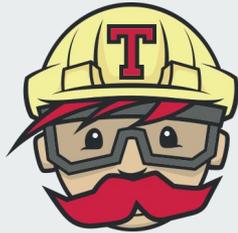
¿Qué automatizamos?

- Ejecutar testeos 👷🏾‍♂️

¿Qué automatizamos?

- Ejecutar testeos 👷🏿 🔵
- Chequear styling 👷🏼‍♀️ 🔵
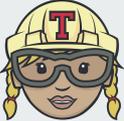
CI: Testeos y styling en Pull Requests

¿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

Contenido

- Control de versiones

- Tres patas para un buen desarrollo

- Automatización

- Construir una comunidad

- ¿Puedo publicar mi librería?

Construir una comunidad

Incentivar a otres a colaborar

- Código de conducta

- Guías para contribuir

- Solicitar ayuda en Issues

- Dar crédito

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

Comunicación

 Sitio web

 slack

 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   Blog   Submit   Log in with ORCID

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/joss.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