The Journal of Open Source Software

ConFluxPro: A toolkit for soil gas analysis

Valentin Gartiser[1, 2], **Verena Lang**[1], **and Martin Maier**[2]

1 Forest Research Institute, Freiburg, Germany **2** Soil Physics, Department of Crop Sciences, Göttingen, Germany

DOI:

Software

- Review
- Repository
- Archive

Submitted:
Published:

Summary

`ConFluxPro` is an R-package to model soil gas fluxes with the flux-gradient method (FGM). The FGM is a cost-effective method to measure the fluxes and production of gases in soils. It relies on the principle that gas exchange in soils is driven by molecular diffusion. Flux rates can therefore be derived by applying Fick's first law of diffusion. First, it requires to measure vertical concentration profiles of gases and estimate the diffusivity of the soil. In a second step, flux rates can be modeled by deriving the concentration gradient and diffusion coefficients of the soil. `ConFluxPro` was developed to facilitate any data handling and modeling related to the FGM. It (I) provides object classes for the preparation, combination and modification of soil gas and physical data, (II) implements different common FGM models, (III) introduces an inverse modeling approach, (IV) provides functions for post-hoc calibration and (V) uncertainty estimation of the model results.

Statement of need

Because the FGM is conceptually simple, it has been applied in numerous studies. However, codes or evaluation files have often not been shared publicly and small differences between implementations limits the comparability between studies. The goal of `ConFluxPro` was to make using the FGM easy, flexible and reproducible.

The measurement of gas fluxes in soils is important to understand subsurface processes, climate responses and the cycling of elements. The flux-gradient method (FGM) is uniquely equipped to estimate fluxes within the soil, is low-cost and ideal for long-term monitoring.

Template

The forces on stars, galaxies, and dark matter under external gravitational fields lead to the dynamical evolution of structures in the universe. The orbits of these bodies are therefore key to understanding the formation, history, and future state of galaxies. The field of "galactic dynamics," which aims to model the gravitating components of galaxies to study their structure and evolution, is now well-established, commonly taught, and frequently used in astronomy. Aside from toy problems and demonstrations, the majority of problems require efficient numerical tools, many of which require the same base code (e.g., for performing numerical orbit integration).

`Gala` is an Astropy-affiliated Python package for galactic dynamics. Python enables wrapping low-level languages (e.g., C) for speed without losing flexibility or ease-of-use in the user-interface. The API for `Gala` was designed to provide a class-based and user-friendly interface to fast (C or Cython-optimized) implementations of common operations such as gravitational potential and force evaluation, orbit integration, dynamical transformations, and chaos indicators for nonlinear dynamics. `Gala` also relies heavily on and interfaces well with the implementations of physical units and astronomical coordinate systems in the `Astropy` package (Astropy Collaboration, 2013) (`astropy.units` and `astropy.coordinates`).

`Gala` was designed to be used by both astronomical researchers and by students in courses on gravitational dynamics or astronomy. It has already been used in a number of scientific publications (Pearson, Price-Whelan, & Johnston, 2017) and has also been used in graduate courses on Galactic dynamics to, e.g., provide interactive visualizations of textbook material (Binney & Tremaine, 2008). The combination of speed, design, and support for Astropy functionality in `Gala` will enable exciting scientific explorations of forthcoming data releases from the *Gaia* mission (Gaia Collaboration, 2016) by students and experts alike.

Mathematics

Single dollars ($) are required for inline mathematics e.g. $f(x) = e^{\pi/x}$

Double dollars make self-standing equations:

$$\Theta(x) = \left\{\begin{array}{l} 0\textrm{ if } x < 0\cr 1\textrm{ else} \end{array}\right.$$

Citations

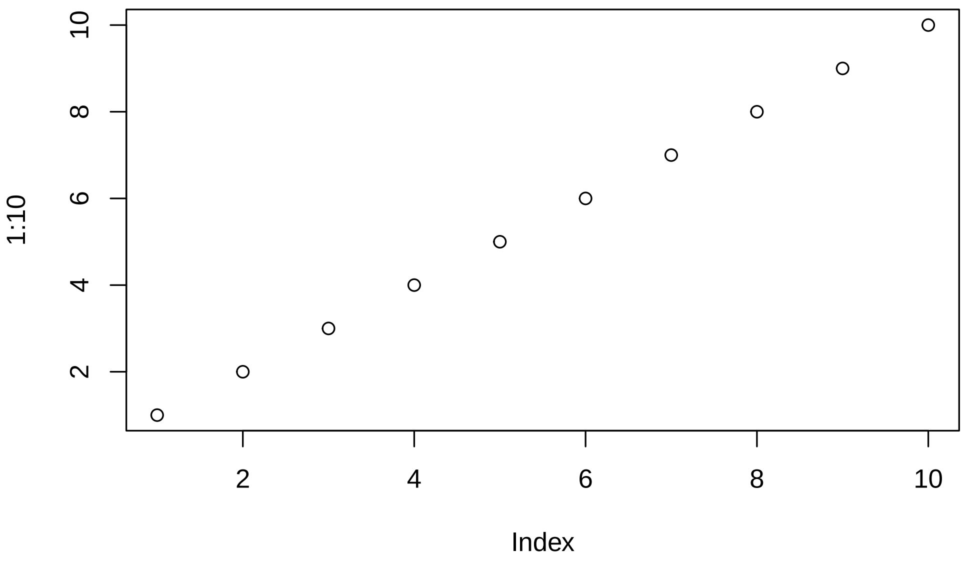
Citations to entries in paper.bib should be in rMarkdown format.

For a quick reference, the following citation commands can be used: - `@author:2001` -> "Author et al. (2001)" - `[@author:2001]` -> "(Author et al., 2001)" - `[@author1:2001; @author2:2001]` -> "(Author1 et al., 2001; Author2 et al., 2002)"

Rendered R Figures

Figures can be plotted like so:

```
plot(1:10)
```



Acknowledgements

We acknowledge contributions from Brigitta Sipocz, Syrtis Major, and Semyeong Oh, and support from Kathryn Johnston during the genesis of this project.

References

Astropy Collaboration. (2013). Astropy: A community Python package for astronomy, *558*. doi:10.1051/0004-6361/201322068

Binney, J., & Tremaine, S. (2008). *Galactic Dynamics: Second Edition*. Princeton University Press.

Gaia Collaboration. (2016). The Gaia mission, *595*. doi:10.1051/0004-6361/201629272

Pearson, S., Price-Whelan, A. M., & Johnston, K. V. (2017). Gaps in Globular Cluster Streams: Pal 5 and the Galactic Bar. *ArXiv e-prints*. Retrieved from https://arxiv.org/abs/1703.04627