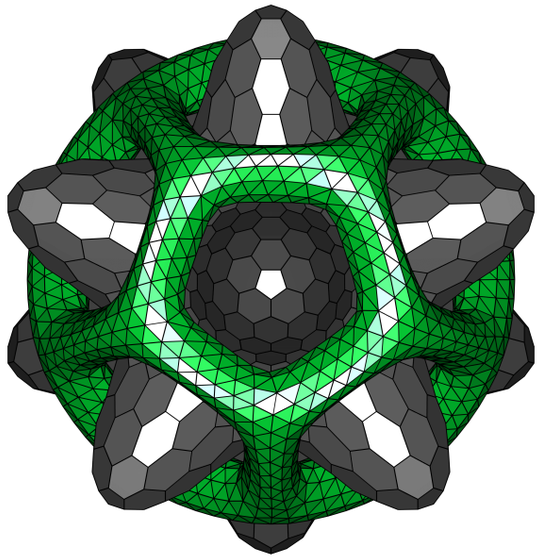
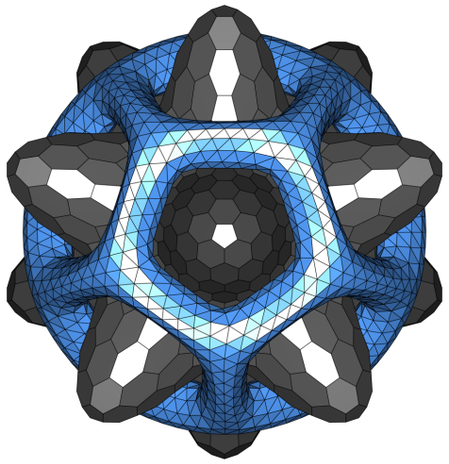
Introducing JOSE: Journal of Open Source Education



Kyle Niemeyer
Oregon State University
https://niemeyer-research-group.github.io



Quick JOSS update

- Year 1 paper published at *PeerJ CS* https://peerj.com/articles/cs-147/

PeerJ Computer Science

Journal of Open Source Software (JOSS): design and first-year review

Arfon M. Smith[1], Kyle E. Niemeyer[2], Daniel S. Katz[3], Lorena A. Barba[4], George Githinji[5], Melissa Gymrek[6], Kathryn D. Huff[7], Christopher R. Madan[8], Abigail Cabunoc Mayes[9], Kevin M. Moerman[10,11], Pjotr Prins[12,13], Karthik Ram[14], Ariel Rokem[15], Tracy K. Teal[16], Roman Valls Guimera[17] and Jacob T. Vanderplas[15]

- As of today: 322 published papers (!) and 80 under review
- 20 members of editorial board
- Fiscally sponsored by NumFOCUS
- Governance of Open Journals (https://github.com/openjournals/governance): BDFL + six steering council members

Overview

- Part of the Open Journals, sibling journal to JOSS (literally forked from it!). Also NumFOCUS-affiliated.

- JOSE publishes two types of articles that describe:
  - ✳ open educational software tools
  - ✳ open-source learning modules

- Motivation: credit efforts to develop software for assisting teaching/learning and open-source educational content

- Submissions are peer-reviewed, with the intent of *improving the quality of the software or content submitted*

(Remaining) Editorial Board

Lorena Barba
George Washington Univ.

Katy Huff
UIUC

Jason Moore
UC Davis

Anthony Scopatz
USC

Charles Severance
Univ. Michigan

Robert Talbert
Grand Valley State U.

Tracy Teal
Univ. Michigan

Carol Willing
Cal Poly San Luis Obispo

Submit software for review

Before you submit
Please make sure you've read the submission instructions before submitting. In particular please make sure there is a `paper.md` present in your repository that is structured like this. We promise this will make things go much more quickly during the review process 🚀

Title

What's the title of this paper?

Repository address

What's the URL of your software?

Software version

e.g. v1.0.0

View editors here »

Suggested editor

Type

✓ software
learning module

Description

Please give short (1-2 line) description of your software.

I certify that I am submitting software for which I am a primary author

I confirm that I read and will adhere to the JOSE code of conduct

Submit paper

Make software available in repository with OSI-approved license 🍺: `https://opensource.org/licenses`

Author short Markdown paper: `paper.md`

Submit to JOSE by filling out short form

JOSE Submitted

Editor assigns ≥1 reviewers, who review submission 🔎

JOSE Under review

Reviewer(s) raise comments and issues following guidelines 👀: `http://jose.theoj.org/about#reviewer_guidelines`

Authors fix issues 👩‍💻

Editor accepts paper, authors archive software ✔

Paper published & receives JOSE DOI ⚡

JOSE 10.21105/jose.#####

Accepted papers (2) Active papers (4) All papers (6)

barbagroup / CFDPython

JOSE Under Review

CFD Python, a.k.a. the 12 steps to Navier-Stokes, is a practical module for learning the foundations of Computational Fluid Dynamics (CFD) by coding solutions to the basic partial differential...

DOI pending

drvinceknight / Nashpy

JOSE Submitted

Nashpy is a python library for the computation of Nash equilibria. The main statement of need is portability as it only requires the standard python scientific stats as dependencies. I previously...

DOI pending

arm61 / pylj

JOSE Under Review

pylj is a Python package designed for use within the Jupyter notebooks framework to allow for the easy visualisation of a 2-dimensional, classical simulation. This software is designed to allow...

DOI pending

gboeing / pynamical

JOSE 10.21105/jose.00015

Pynamical is an educational Python package for introducing the modeling, simulation, and visualization of discrete nonlinear dynamical systems and chaos.

10.5281/zenodo.1294299

riffomonas / reproducible_research

JOSE Under Review

The Riffomonas Reproducible Research tutorial series is a collection of tutorials that focuses on the improvement of reproducible data analysis for those doing microbial ecology research.

DOI pending

First papers published!

A short course about fitting models with the `scipy.optimize` module

Article details

- View review »
- Download paper »
- Software repository »
- Software archive »

Submitted: 26 April 2018
Accepted: 04 July 2018

Cite as:

Rokem, (2018). A short course about fitting models with the scipy.optimize module . Journal of Open Source Education, 1(2), 16, https://doi.org/10.21105/jose.00016

Status badge

JOSE 10.21105/jose.00016

License

Authors of JOSE papers retain copyright.

This work is licensed under a Creative Commons Attribution 4.0 International License.

JOSE
The Journal of Open Source Education

A short course about fitting models with the `scipy.optimize` module

Ariel Rokem[1]

1 The University of Washington eScience Institute

DOI: 10.21105/jose.00016

Software

- Review
- Repository
- Archive

Submitted: 26 April 2018
Published: 04 July 2018

Summary

Fitting models and testing the match of the models to the measured data is a fundamental activity in many fields of science. This short (approximately 3-hour) course (available at: https://github.com/arokem/scipy-optimize) aims to teach participants to use the Scipy library's `optimize` module to fit models to data (Jones et al. 2001). Using data from a psychology experiment (Rokem and Landau 2016) as an example, the course motivates the use of explicit mathematical models to explain and predict data and compares linear models and non-linear models. The core of the lesson focuses on fitting a curve with the `curve_fit` function. The course also introduces the idea of model comparison with cross-validation for evaluation and selection between non-nested non-linear models.

Statement of need

Model fitting is useful in many different fields of research, but optimization for model fitting is not a topic that is usually covered in introductory statistics or computing classes in many fields (e.g., psychology). This course fills an existing need for hands-on curriculum that goes beyond the topics taught in introductory computing workshops, such as Software Carpentry, providing material for follow-up workshops on advanced/intermediate topics. The target audience for this course is researchers or students with some programming knowledge (e.g., having participated in a Software Carpentry workshop beforehand).

Pynamical: Model and visualize discrete nonlinear dynamical systems, chaos, and fractals

Article details

- View review »
- Download paper »
- Software repository »
- Software archive »

Submitted: 15 May 2018
Accepted: 21 June 2018

Cite as:

Boeing, (2018). Pynamical: Model and visualize discrete nonlinear dynamical systems, chaos, and fractals . Journal of Open Source Education, 1(1), 15, https://doi.org/10.21105/jose.00015

Status badge

JOSE 10.21105/jose.00015

JOSE
The Journal of Open Source Education

Pynamical: Model and visualize discrete nonlinear dynamical systems, chaos, and fractals

Geoff Boeing[1]

1 University of California, Berkeley

DOI: 10.21105/jose.00015

Software

- Review
- Repository
- Archive

Submitted: 15 May 2018
Published: 21 June 2018

Summary

Pynamical is an educational Python package for introducing the modeling, simulation, and visualization of discrete nonlinear dynamical systems and chaos, focusing on one-dimensional maps (such as the logistic map and the cubic map). Pynamical facilitates defining discrete one-dimensional nonlinear models as Python functions with just-in-time compilation for fast simulation. It comes packaged with the logistic map, the Singer map, and the cubic map predefined. The models may be run with a range of parameter values over a set of time steps, and the resulting numerical output is returned as a pandas DataFrame. Pynamical can then visualize this output in various ways, including with bifurcation diagrams (May 1976), two-dimensional phase diagrams (Packard et al. 1980), three-dimensional phase diagrams, and cobweb plots (Hofstadter 1985). These visualizations enable simple qualitative assessments of system behavior including phase transitions (Feigenbaum 1983), bifurcation points (Sander and Yorke 2015), attractors and limit cycles (Grebogi, Ott, and Yorke 1987), basins of attraction (Sprott and Xiong 2015), and fractals (Mandelbrot 1967, 1983).

Although most real-world systems are nonlinear dynamical systems, their mathematical analysis is notoriously difficult because they cannot be simply broken down into individual parts then recombined linearly (Strogatz 2014). Instead, researchers have long relied on visualization techniques to make system behavior comprehensible (Alpigini 2004; Layek

Submissions welcome!

http://jose.theoj.org