



The Journal of Open Source Economics

Journal Charter

This version: June 2019 (draft)

This document outlines the foundational principals and the objectives of The Journal of Opens Source Economics (JOSEcon), describes its organizational structure and operational strategy. The governing rules and practices are periodically updated to reflect the development of the journal.

Table of contents

1	Executive summary	1
2	Background and Motivation	1
	2.1. Numerical methods and software development in economics. 2.2. Right incentives for developing better code. 2.3. Success of the Journal of Statistical Software	
3	Vision and Mission of JOSEcon	4
	3.1. Foundational principles. 3.2. Types of JOSEcon papers. 3.3. Audience and role models. 3.4. Collaboration with other organizations	
4	JOSEcon Operations	8
	4.1. Organizational structure and governance. 4.2. Submission and review process	

1 Executive summary

The Journal of Open Source Economics (JOSEcon) publishes “code papers” written by, and useful for the code developers in economics.

The immediate goal of publishing coding work is to enable indexing and collection of citations by its authors, thus allowing for the quality coding work to be credited similarly to traditional publications. More broadly, JOSEcon promotes best software engineering practices in economics, building the computational economics community. It is also working together with traditional journals to help their review process of the papers that rely heavily on computations.

The field that has already build some mass of people aware and interested in better software development is computational macro, which has the majority of existing coding projects like QuantEcon or Econ-ARK. But JOSEcon covers all fields of economics and econometrics that require computational methods, such as microeconometrics and structural econometrics, empirical IO, life cycle modeling, Bayesian econometrics, computational game theory, etc.

The journal is the economics field specific version of The Journal of Open Source Software (JOSS), and inherits its main principles: open publication of open source code with an open review process.

2 Background and Motivation

2.1 Numerical methods and software development in economics

The use of numerical methods is becoming essential in many fields of modern economic research. As theoretical models in both macro- and microeconomics are gaining realism and moving away from analytically tractable specification, numerical solutions provide the necessary tools for analyzing the implications of these models. The development of computing technology itself has enabled the growth of computationally intensive methods in econometrics. Similarly, the development of computational methods is enabling the estimation of the deep structural parameters of the advanced theoretical models using real life data. Counterfactual simulations based on these calibrated and/or estimated models are invaluable for our understanding of the underlying mechanisms and the analysis of policy reforms in all fields of economics.

However, at large, the code development practices in economics lack the level of sophistication typical for other sciences such as physics or biology. Even those economists who are actively working with numerical models rarely have any software engineering training or use the standard toolchains. The common practice is copying the code base from supervisor to the student, and from project to project – resulting in repeated and long surviving errors.

Moreover, even though journals typically do require submission of the written code for the papers that rely on numerical methods, in the absence of common standards for such code supplements, the deposited code is rarely checked. It is not uncommon that the code archived at the publishers’ websites is hard to re-use simply because of the lack of

clear description of the operating environment it should be ran with. The code made available through personal websites may be updated more regularly, but may be as hard to use.

Computational methods and solution algorithms are typically published “on paper” in isolation from the programming implementations, and therefore are often coded by from scratch. In the absence of canonical implementations, the same methods are then programmed multiple times by different researchers, resulting in the practice of “reinventing the wheel” and sometimes introducing critical errors into the well established methods.

Although computational economics has not (yet) experienced a replicability crisis of the scale that has crippled some other fields, any sensible person who has done computationally intensive research will have sometimes had the queasy feeling that many results (in other people’s computational work) are probably fragile, and more than a few may even be wrong because of bugs buried in the sea of vast incomprehensible code that produced the results.

We are in this bad equilibrium because there are few incentives for economists to produce high quality, robust, well-documented code that is usable and testable by others.

2.2 Right incentives for developing better code

The current situation can be improved greatly by learning from other disciplines, first of all computer science and software engineering. In the brighter future the field of computational economics will possess a growing library of well architected and documented code that will cover the majority of known computational models. Through efficient sharing and repeated scrutiny within independent projects, the code base will gain reliability. Eventually, a collection of trusted canonical implementations will be formed. The growing community of economic code developers will contribute to the spreading of good coding practices and provide a role model for the students who wish to join the field.

The purpose of The Journal of Open Source Economics (JOSEcon) is to fuel the development of the computational economics community by providing the right incentives for writing high quality reusable code – publications and collection of citations.

JOSEcon publishes *code papers* which are small descriptions of the software package aimed at solving a particular well-defined problem in economics, with the focus of the editorial process being at the code itself. A typical JOSEcon publication contains a well documented software package deposited to a public repository, a short script or a notebook that demonstrates the use of the package, and a few page description of the are of application of the developed software as well as the provided example. Under separate rubric JOSEcon also publishes the code developed for and accompanying specific traditional papers, as well as replications of the existing published results (details below).

As a first step in this direction, the Econ-ARK project has been working to develop Jupyter notebooks that illustrate the use of the tools we are creating. It is easy to see how tools like Jupyter notebooks could revolutionize teaching; but if a mechanism like the JSS

Overall, if JOSEcon succeeds in creating a mechanism by which a contribution to computational economics community can be indexed and its importance measured by citation

counts and other traditional metrics that count, for example, for tenure, we could have a whole new paradigm for academic publication of computational tools for economists (and, maybe eventually for the rest of the field).

2.3 Success of the Journal of Statistical Software

The academic statistics community, in particular, confronted basically the same set of problems in the mid-90s, which prompted a few statistics professors to form the Journal of Statistical Software (JSS). (The history of the JSS is recounted quite well in the look-back review¹ presentation by the founding editor, with many valuable lessons learned.)

To illustrate the idea of the JSS, think of a case where an academic statistician published a paper in a regular statistics journal with some mathematical/statistical/conceptual contribution. Associated with that paper was some computer code implementing or illustrating the idea/algorithm/method. The peer review process would vet the mathematical/statistical/conceptual contribution, but there was no real quality check on the code. A social norm or journal policy may have required that the author share a zip file of the code with anybody who asked for it, or such a zip file may have been available from the journal as supplemental material for the article, but the author had no obligation to help anybody else verify or use the code. It was nobody's job really to ensure that the code was usable by anybody else or on any other machine or at any future time.

The JSS solved this problem (for statistics) by inventing a mechanism by which the production of high quality code could be professionally rewarded in the usual coin of the academic realm: A code contribution published by the JSS can be listed on the author's CV as an "article" that is eligible to receive citations that are treated by the profession the same way as citations to regular articles in more traditional journals (e.g., they show up in Google Scholar and other citation counting tools).

Of course, this would not be sufficient if the JSS were viewed as a "junk journal." But, using the same "impact factor" methodology that produces the conventional conclusion that the "top 5" economics journals are AER, Econometrica, JPE, QJE, ReStud (I've excluded finance journals), the JSS is now the fifth-highest "impact factor" journal in Statistics. This is astounding for a journal founded only 20 years ago. (The youngest of the other leading "top 5" journals in statistics is about 90 years old, leaving a side a "reviews" journal).

Elaborating: An "article" in the JSS is basically a corpus of computer code with some illustrations of the use of the code. Standards of review for a submission include whether the code is well documented, has useful illustrations, has adequate quality tests, etc.

With the tools that exist now, I think it would be possible to do an even better job for computational economics than JSS has done for statistics. Github, Python, Jupyter notebooks, and a host of other developments have created a whole ecosystem that could make such a project work much better than was possible in statistics when JSS was founded.

¹http://gifi.stat.ucla.edu/janspubs/2014/notes/deleeuw_mullen_U_14.pdf

3 Vision and Mission of JOSEcon

3.1 Foundational principles

JOSEcon is a developer-friendly, free and open-access, peer-reviewed journal for economic software packages.

JOSEcon accepts submissions of *code* written in conjunction with economic research (implementations of the original numerical methods, estimation algorithms, replications of the existing results and methods), and publishes *code papers* which are short description of the purpose of the submitted software along with an example of use.

Requirements for the code paper are minimal to minimize additional effort needed to submit to JOSEcon.

The review process has two stages. First stage aims at assessing the contribution of the submitted code in moving forward the open source economic code base. This assessment is performed by one or several editors, possibly with additional anonymous referee or referees.

The second stage of the review process is public and focuses on code review. Before acceptance decision is made, the assigned referee or referees make sure that the submitted software satisfies the list of known JOSEcon requirements.

The accepted code papers are published on [JOSEcon website](#) along with a link to the public repository containing the accepted code itself. The published code paper is assigned a digital object identifier (DOI) with CrossRef, which can be displayed in the repository with a JOSEcon badge.

JOSEcon accepts for publication only open source code as per the [OSI definition](#).

There are no restrictions on the programming languages at JOSEcon: submissions of code written in any languages are welcome, and the popularity of the language does not affect the outcome of the submission (apart, perhaps, from how fast the reviewers can be appointed).

There are no restrictions on the technology and hardware required for running the submitted code (apart from how fast the reviewers can be appointed).

The software should be feature complete (no half-baked solutions) and designed for maintainable extension (not one-off modifications). Minor ‘utility’ packages, including ‘thin’ API clients, are not acceptable.

The software should be available through public repository (GitHub, GitLab, etc), namely:

- Be stored in a repository that can be cloned without registration
- Be stored in a repository that is browsable online without registration
- Have an issue tracker that is readable without registration
- Permit individuals to create issues/file tickets against your repository

JOSEcon fosters the development of a community of computational economists, who may be involved with a journal to a varying degree (from developers to users of the economic code, from editors to reviewers to simply campaigners), but who share similar views about the future of computations in economics, and are responsive and welcoming to the new

members of the community.

Authors, editors and reviewers, as well as any other members of JOSEcon community, adhere to a code of conduct adapted from the [Contributor Covenant](#) code of conduct.

Authors, editors and reviewers of JOSEcon are expected and required to adhere to ethical standards, in particular:

- All authors are required to make the JOSEcon editors and the community aware of any mistakes in their published work which they become aware.
- Plagiarism for both code and code papers is not tolerated.
- Repeated submissions of the same work is not allowed.
- Submissions of the work which is under revision in a separate software journal is not allowed. (It is, however, perfectly fine to submit the “big” paper to a traditional journal, while the software developed for it being submitted to JOSEcon, perhaps with a smaller scale example of use.)
- Listed authors on the code papers must include all people who contributed towards the submission, and not include anybody who has not. For the replication papers JOSEcon encourages but does not require the inclusion of the original authors.
- Code reviews should be accurate and non-fraudulent.
- Reviewers and editors must disclose conflicts of interest.

JOSS is an open access journal. Copyright of JOSS papers is retained by submitting authors and accepted papers are subject to a Creative Commons Attribution 4.0 International License.

Any code snippets included in JOSS papers are subject to the MIT license regardless of the license of the submitted software package under review.

Any use of the JOSS logo is licensed CC BY 4.0.

3.2 Types of JOSEcon papers

JOSEcon publishes three types of code papers:

1. Full featured code library or package that implements a particular model or solution/estimation method used in economics. It is expected that a package has a well documented API, and is accompanied by an independent run script that implements an example (or multiple examples) of use of the package. The code paper describes the purpose of the software, and presents the computed example. Generally there is no single traditional paper associated with this type of submission, but it is understood that the research applications of the software would follow and built on the provided examples.
2. Code supplement for a particular traditional paper that relies heavily on computations. The scenario under the second type of submission is that the code written during the work on a bigger paper is of a sufficient quality to constitute a publication at JOSEcon. The primary criterion here is whether the code can be used in a similar application, or serve as the bases for development of the next generation software in the area. It is expected that the code is organized into a re-usable library/package, and a run script that calls the routines in the library to produce the results in the paper. Ideally, the whole set of results should be reproduced by

the run script, but in case where the required run time would hinder the review process, simpler examples are sufficient.

3. Replications of the existing methods and published results, written in new programming languages, or with significant refactoring and improvements in code architecture. JOSEcon welcomes replications of existing papers, but has special rules requiring the authors of the replicated paper be contacted and invited to participate in the replication. It is not necessary that they do, but the record of communication should be submitted together with the code paper. The replication code should also be organized into library/package and a run script structure, and it is expected that the run script replicates the results in the same form as in the original paper for easier comparison.

Many other types and kinds of software written by or for economists are not currently considered for publication at JOSEcon. In particular, the journal does not accept:

- Simple technical code such as thin library wrappers, converters, and similar;
- Text markup and style files such as latex or html;
- Code written exclusively for teaching purposes;

Jupyter notebooks and similar document format that allows for combination of executable code cells with explanatory comments, is an excellent way to present the API of a library and set up examples. JOSEcon encourages the use of this format for the run scripts that can simultaneously present the example.

Fedor:
refer-
eeing
con-
tribu-
tion of
teach-
ing ma-
teri-
als is
all dif-
ferent
matter

3.3 Audience and role models

JOSEcon offers several role models for a few target audiences.

The primary audience (JOSEcon community) are the economists who already rely on computational methods or plan to use them in the near future: faculty and students writing code in their research projects.

Fedor:
static
pdf ver-
sion of
Jupyter
note-
books
can be
used for
code
paper?

Members of the JOSEcon community fill the roles on the editorial board and referees, and are active readers and user of published code.

There is a good number of open source projects in economics producing excellent code already, as well as several projects aimed at teaching computational economics. JOSEcon strives to unite these developers into the bigger scale community.

1. [Econ-ARK](#) by Christopher Carroll and the team;
2. [QuantEcon](#) by Thomas Sargent, John Stachurski and the team;
3. [OpenSourceEconomics](#) by Philipp Eisenhauer and the team;
4. [skillmodels](#) by Janos Gabler and the team;
5. [VFI toolkit](#) by Robert Kirkby;
6. [EconForge and Dolo library](#) by Pablo Winant;
7. [Hetsol toolkit](#) by Michael Rieter;
8. [Dynare](#) by Michel Juillard and the team;
9. [Macroeconomic Model Data Base](#) by Volker Wielands;
10. [pyblp package](#) by Christopher Conlon and Jeff Gortmaker;

There are several ways that new members are recruited into the JOSEcon community. Existing faculty who are interested in computational methods would benefit from the lower

barrier of entry thanks to JOSEcon collection of tested and reliable code. In addition, the code published at JOSEcon would provide great resource in teaching. Students working in the fields which have substantial computational component would have a chance to start their careers by publishing in JOSEcon relatively quickly. The younger students interested in the field could use their assignments and term papers to replicate the results of published traditional papers, and also publish well ahead of their job market.

Important other role models include:

- Referee for a different journal wishing to verify the computational part of the paper provided that it has been submitted to JOSEcon;
- Member of editorial board of another journal to suggest independent code review in their editorial process;

3.4 Collaboration with other organizations

JOSEcon is actively seeking collaborations with other organizations to popularize better practices in the use of numerical methods throughout the economic profession.

Societies: Endorsements from various societies in economics help JOSEcon gain respect in the profession.

- [Society of Computational Economics](#)

Existing software journals: The JOSEcon editorial team carefully monitors other software journals both to absorb better journal practices, and to monitor competition.

1. [Journal of Open Source Software \(JOSS\)](#)
 - Founded in May 2016
 - Developer-friendly, very short structured “papers”, focus of the code
 - Indexed in CrossRef, DOI identifiers
 - The founder and current editor-in-chief is [Arfon Smith](#)
2. [Journal of Open Source Education \(JOSE\)](#)
3. [Journal of Statistical Software \(JSS\)](#)
4. [ReScience C journal](#)
5. [International Journal of Open Source Software and Processes](#)
6. [rOpenSci](#)
7. [Run My Code community](#)
8. [Software Impacts](#)

JOSEcon builds on the principles and infrastructure developed by [The Open Journals](#), which includes JOSS and JOSE.

Traditional journals: JOSEcon is actively seeking relationships with “traditional” journals, and hopes that when it is reputable enough, such relationships would develop without outside force.

JOSEcon vision is that in the long term traditional journals would take into consideration during their editorial process the publication at JOSEcon for the papers that rely heavily

on computation. Such collaboration would benefit the traditional journals, taking code review part off their shoulders.

Publishing houses: It is unclear how JOSEcon would coordinate with the existing publishing houses, being absolutely open (both the publications, and the editorial process) to the public. Yet, involvement of a respected brand would be invaluable for the development of the journal.

Open Source Initiative: JOSEcon is an affiliate of the Open Source Initiative. As such JOSEcon is committed to public support for open source software and the role OSI plays therein.

Fedor:
formal
affili-
ation
pending

4 JOSEcon Operations

4.1 Organizational structure and governance

The governance structure at JOSEcon is primarily modeled from JOSS and Julia project, and consists of an *Editorial Board* running the every day operations at the journal, and an *Advisory Board* overseeing these operations.

The editorial board consists of an *Editor-in-Chief* and a team of *Co-Editors*. The members of the editorial board share the responsibilities of handling submitted papers, in particular they make the initial rejection decisions (individually or in private discussion with the other editors, or by appointing external anonymous referees), and if the submission is not rejected right away, facilitate the open review process that follows, and take the final decision.

The advisory board consists of prominent senior economists working in the broadly defined field of computational economics. Currently the advisory board members are:

- Christopher Carroll
- John Stachurski
- John Rust
- Felix Kubler
- Sergey Malin

4.2 Submission and review process

JOSEcon operates similarly to the Journal of Open Source Software (JOSS), and together with the core principles inherits the overall structure of the editorial process which is described at <https://joss.readthedocs.io>

In particular, JOSEcon had clearly written guidelines for both authors, reviewers and editors.

The one difference between JOSEcon and JOSS is that the former has multiple submission types (above), and opts for a two stage review process:

1. Private review by editorial board (in some cases with anonymous referees) assessing the impact of the submission for the open source economics code;
2. Public review similar to JOSS focusing on code review.

Once the submission passes through the first stage, it is typically never rejected. Instead, the reviewers help to ensure that the code satisfies the standards specified in the guidelines, and even if the original authors take time to respond to the comments and suggestions, the code is eventually accepted to JOSEcon.

The accepted submissions are issued a JOSEcon badge that the authors are welcome to display at the code repository. The code paper is published at JOSEcon website, and the accepted version of software is archived through the public archive services.

Accepted and published code can be developed further in the same repository, to the point when the updates are sufficient to grant a new publication.

The web site of the journal is located at <http://josecon.org>. It contains the archive of all published papers with a topic index, and provides a full text search facility.