

Code and **don't** Perish?!

How about publishing your software?

menti.com
8766 433



Workshop held at MMS Days, April 10, 2024

by B. Schmidt, J. Ph. Thiele

Pressure to publish scientific work to succeed in an academic career

- Quantitative pressure: Number of publications (ResearchGate, GoogleScholar, h-index)
- Qualitative pressure: High rank journals (Nature, Science, impact factor)
- Institutional pressure: Strongest at research universities/institutions
- Individual pressure: Highest for **young researchers in non-permanent employment** (i.e., postdocs)

Advantages

- Motivation for researchers to produce cutting edge research
- Motivation for researchers to start publishing early in their career
- Motivation for researchers to focus on research advancement

Disadvantages

- Emphasis on massive publishing may reduce scientific creativity
- Obsession with quantity rather than quality, choice of questionable (predator?) journals
- Detraction from research, teaching (and other community) obligations
- Sometimes even scientific misconduct / fraud / psychological issues

menti.com (1)
8766 433

Developing software as part of scientific work

- Increasing importance of computations and simulations
- Software for specific scientific needs often not available „off the shelf“
- Close connection to numerical | computational algorithms
- One-purpose („quick & dirty“) versus general purpose, documented („slow & clean“) software

Advantages

- Investment in software is investment in science*
- Reproducibility and reusability of scientific results through open software (FAIR principles)
- Community idea: Give and take (collaborative platforms, open source)

Disadvantages

- Distraction from the original scientific task?
- Too many technicalities (versions, dependencies, incompatibilities, data formats, ...)
- Imbalanced community interaction: more „give“ than „take“
- Career obstacle: **Lacking recognition** of software as research output !?!

Manifesto: ...software has made an invaluable contribution to advancing research...

- We believe that the full benefits of software in research will only be realized when software is accepted as a valid research output.
- We call for all researchers to have access to basic software training to help them harness the power of software for their research.
- We say that there must be reward and recognition for *Research Software Engineers*: the people who contribute to research by developing software.
- Finally, we argue that good software practices create better software, and that better software improves the reproducibility and reusability of research.



... the principal record of scientific research is still the peer-reviewed publication. Given that software is a fundamental part of doing science in the digital age, the question we are often asked is: *where can I publish papers which are primarily focused on my scientific software?*

menti.com (3)
8766 433

General journals

- Computing in Science & Engineering
- Concurrency and Computation: Practice and Experience
- The Journal of Open Source Software (JOSS)
- Journal of Open Research Software (JORS)
- Journal of Software: Practice and Experience
- Nature Toolbox
- Research Ideas and Outcomes
- SoftwareX

menti.com (4a,b,c)
8766 433

Informatics, Mathematics and Statistics

- Transactions on Mathematical Software (TOMS)
- Computing in Science & Engineering
- Mathematical Programming Computation
- Numerical Algorithms
- The R Journal

Example: Journal of Open Source Software (JOSS)

The Journal ... is a **developer friendly**, open access journal for research software packages.

What *exactly* do you mean by 'journal'?

JOSS is an academic journal (ISSN 2475-9066) with a formal **peer review process that is designed to *improve the quality of the software submitted***.
Upon acceptance into JOSS, a Crossref DOI is minted ...

Don't we have enough journals already?

In a perfect world papers about software weren't necessary ...
But for most researchers, **papers and not software are the currency of academic research** and that citations are required for a good career.
We believe that after you've done the hard work of writing great software, **it shouldn't take weeks and months to write a paper** about your work.

You said *developer friendly*, what do you mean?

We ... help you prepare your submission. If your software is already well documented **then paper preparation should take no more than an hour (250-1000 words)**.

Example: Journal of Open Source Software (cont'd)

Software associated with your submission must:

- 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**; individuals may create issues

What we mean by research software

- solves complex modeling problems in a scientific context
(physics, mathematics, biology, medicine, social science, neuroscience, engineering)
- supports the functioning of research instruments or the execution of research experiments;
extracts knowledge from large data sets; offers a mathematical library
- **machine learning models and notebooks are not in-scope for JOSS**

Substantial scholarly effort

- Your software should be a **significant contribution** to the available open source software that either enables some new research challenges to be addressed or makes addressing research challenges significantly better (e.g., faster, easier, simpler).
- **JOSS' minimum allowable contribution should represent not less than three months of work for an individual.**
JOSS requires that software should be feature-complete (i.e., no half-baked solutions).

Considerations of authors

- Publication alongside other publication describing scientific application, algorithm, method assessment
- Submissions for which the **implementation of the software itself reflects a substantial scientific effort**
- Also submissions **on significant contributions to other existing packages**

Considerations of editors and reviewers

- Age of software / length of commit history / number of commits / number of authors
- Total lines of code: < 1000 will usually be flagged, < 300 will be desk rejected
- Whether the software has already been cited in academic papers.
- Whether the software is **sufficiently useful that it is *likely to be cited* by your peer group**

Typical paper submission work flow

- Make your software available in an open repository (GitHub, etc.) with an open source license
- Software should be **full-featured, well-documented, and contain checking procedures** (e.g. automated tests)
- Write a **short paper in Markdown format**: title, summary, authors, affiliations, key references
- (Optionally) create a metadata file describing your software using YAML, a human-friendly data language

Example: Journal of Open Research Software (JORS)

General features

- The **metajournal** JORS features peer reviewed software **metapapers** describing research **software with high reuse potential**
- Equally importantly, **the software and the papers will be citable, and reuse will be tracked.**
- **JORS also publishes full-length research papers** that cover different aspects of creating, maintaining and evaluating open source research software. The aim of the section is to **promote the dissemination of best practice and experience** related to the development and maintenance of reusable, sustainable research software.

Open access policy

- This journal provides immediate open access to its content on the principle that making research freely available to the public supports **a greater global exchange of knowledge.**
- Authors of articles published in *Journal of Open Research Software* **remain the copyright holders** and grant third parties the right to use, reproduce and share the article according to the CC license agreement.
- Authors are encouraged to **publish their data in recommended repositories**. For a list of generic and subject specific repositories that meet peer review criteria, see [wiki of the Open Access Directory](https://openresearchsoftware.metajnl.com/about)

Source code repositories hold many versions of the software

- Allow the deposit of software under the correct licence
- Provide a unique, persistent identifier for a particular version of the source code
- Has a published backup policy that **do not allow deletion without warning**

Long term preservation repositories hold a set of files

- Allow the deposit of software under the correct licence
- Provide a unique, persistent identifier (e.g. a DOI) for the deposited software
- Have a published preservation strategy that guarantees **long term preservation**

Requirements

- Software available in at least one repository
- Ideally both types of repositories
- **Recommendations:** Assembla, Codeplex, Figshare, GitHub, GitLab, Google code, LaunchPad, Savanna, Zenodo

Journal statistics for the 2023 volume

Submissions received ¹	43
Reviews requested ²	224
Reviews received ³	45
Total Rejections ⁴	19
...of which, Desk rejects ⁵	10
Acceptances ⁶	15
Acceptance rate ⁷	42%
Time from submission to publication ⁸	328 days

Example: Transactions on Mathematical Software (TOMS)

General features

- As a scientific journal, *Transactions on Mathematical Software (TOMS)* documents the theoretical underpinnings of numeric, symbolic, algebraic, and geometric computing applications.
- The purpose of *Transactions on Mathematical Software (TOMS)* is to communicate important research results addressing the development, evaluation and use of mathematical software.
- This journal is published on a quarterly basis.
- It focuses on analysis and construction of algorithms and programs, and the interaction of programs and architecture.
- Algorithms documented in *TOMS* are available as the Collected Algorithms of the ACM at calgo.acm.org.
- The software may be written in any programming language that is in widespread use, but the author must be able to make the case as to why the language chosen is the most appropriate given the goals of wide usability and applicability of research published in TOMS.

Bibliometrics	Publication Years	Publication counts	Citation count	Available for Download	Downloads (6 weeks)	Downloads (12 months)	Downloads (cumulative)	Average Downloads per Article	Average Citation per Article
	1975 - 2024	1,761	97,136	1,761	19,811	141,148	1,497,885	851	55

Research paper

- A paper that discusses **original research into mathematical software**. Where appropriate, reference will be made to publicly available software written by the author and/or by others.

Survey paper

- A paper that **summarizes and organizes recent mathematical software research results** in a novel way that contributes to advancing research in the field. The paper should have a narrow focus. (Very high standards for effective presentation will be applied.)

Algorithm paper

- A paper that describes the **implementation of a particular algorithm** in computer software. A machine readable implementation of the algorithm, in the form of a complete, and well-engineered software package, must be included.
This software is considered part of the submission.

Initiative and Policies

- TOMS accepts manuscripts for an additional, and presently optional, review of computational results.
- This Replicated Computational Results (RCR) review is focused solely on replicating any computational results that are included in a manuscript.
- If the results are successfully replicated, the manuscript receives a special RCR designation when published.

Review process

- The RCR reviewer has the sole responsibility to replicate manuscript computational results.
- The RCR reviewer will be known to the authors and work together with the authors during the RCR process.
- ... responsibility to declare whether or not computational results in the manuscript are replicated.
RCR reviewers will document the details of how results were replicated.
- There is some risk now and in the future that RCR efforts will fail.
In this case, the manuscript is not ready for publication with the presented results.
- A manuscript whose computational results are successfully replicated will be published with a special RCR designation as a text-only note on the bottom of the first page and a special graphic designation

Physical Chemistry / chemical physics / theoretical chemistry / molecular physics

- Computer Physics Communications (computer programs in physics)
- Journal of Chemical Theory and Computation
- Journal of Chemical Physics (chemical physics software)
- Journal of Computational Chemistry (special articles - software news and updates)
- SIAM Journal on Scientific Computing
 - Numerical Algorithms for Scientific Computing
 - Software, High-Performance Computing, and Computational Science and Engineering
 - Methods for Computational Science and Engineering

Advantages (over pure software journals)

- Better visibility within community?
- Better community integration?
- Journals with higher reputation?
- More rigorous peer review?

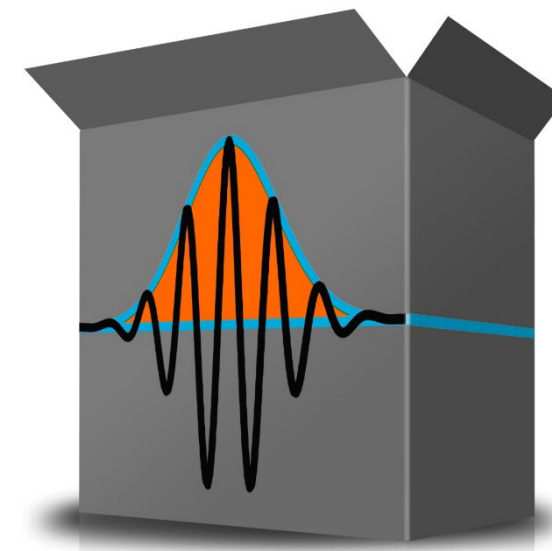
menti.com (5)
8766 433

Features

- Quantum dynamics: Schrödinger/Liouville/Lindblad equations
Control of quantum systems by external fields, also optimal control theory
- Fully classical and mixed quantum-classical dynamics:
Classical trajectories, surface hopping trajectories
- Animated graphics „on the fly“, also Wigner representations
- Object-oriented design: Easy customization

Hosted at SourceForge

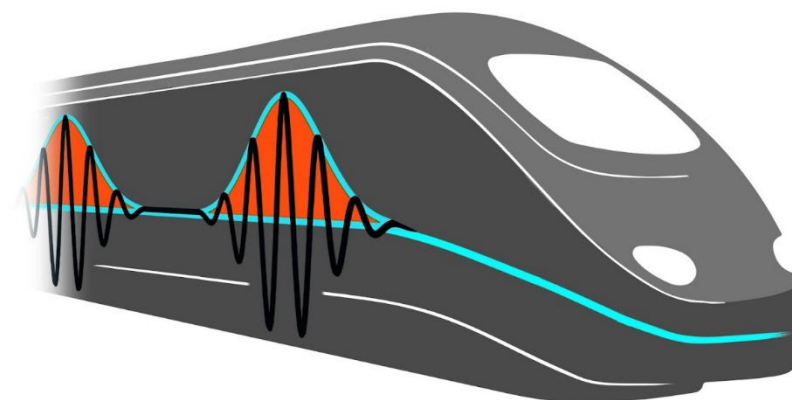
- Open source, free software (public at SF.net since 2008: cvs -> svn -> git)
- Matlab/Octave version (**mature version**): 1 long-term collaborator
- Python/C++ version (**early version**): 1 long-term collaborator
- Extensive **Wiki pages, 100+ demo examples** (~own papers)
- Ticket system, blog, FAQ, extensive technical documentation



menti.com (6a)
8766 433

Features

- Quantum dynamics: Time-independent, time-dependent Schrödinger equations
- Low-rank tensor decompositions (~DMRG):
Tensor trains, aka matrix product states
Beating the *curse of dimensionality*
- Chain-like quantum systems:
Nearest-neighbor interactions only
- Fully classical and mixed quantum-classical dynamics
- Animated graphics „on the fly“, also Wigner representations
- Object-oriented design: Easy customization



Hosted at Github

- Open source, free software (public at GitHub since 2022)
- Python version building on scikit_tt (tensor train toolbox): 1 long-term collaborator (P. Gelss @ ZIB)
- No Wiki pages (yet), but a few demo examples
- Ticket system, some technical documentation

menti.com (6b)
8766 433

WaveTrain: A Python package for numerical quantum mechanics of chain-like systems based on tensor trains

Journal of
COMPUTATIONAL
CHEMISTRY



ELSEVIER

WavePacket: A I dynamics. III. Q hopping trajectory

Burkhard Schmidt ^{a,*}

WavePacket: A Matlab package II: Open quantum systems, optimal control, and model reduction

Burkhard Schmidt ^{a,*}, Carsten Hartmann ^b

^a Institut für Mathematik, Freie Universität Berlin, Arnimallee 6, D-14195 Berlin, Germany

^b Institut für Mathematik, Brandenburgische Technische Universität, Konrad-Wachsmann-Allee 1, D-03046 Cottbus, Germany

WavePacket Closed quan

Burkhard Schmidt ^{a,*}, Ulf Lorenz ^b

^a Institut für Mathematik, Freie Universität Berlin, Arnimallee 6, D-14195 Berlin, Germany

^b Institut für Chemie, Universität Potsdam, Karl-Liebknecht-Straße 24-25, D-14476 Potsdam, Germany

Cite as: J. Chem. Phys. 158, 164801 (2023); doi: [10.1063/5.0147314](https://doi.org/10.1063/5.0147314)

Submitted: 21 February 2023 • Accepted: 10 April 2023 •

Published Online: 28 April 2023



View Online

Exp

Jerome Riedel, ^{1,a)} Patrick Gelß, ^{2,3,b)} Rupert Klein, ^{2,c)} and Burkhard Schmidt ^{2,4,d)}

AFFILIATIONS

¹ Institut für Chemie, Freie Universität Berlin, Altensteinstraße 23A, D-14195 Berlin, Germany

² Institut für Mathematik, Freie Universität Berlin, Arnimallee 3–9, D-14195 Berlin, Germany

³ Zuse-Institut Berlin, Takustraße 7, D-14195 Berlin, Germany

⁴ Weierstraß-Institut für Angewandte Analysis und Stochastik, Mohrenstraße 39, 10117 Berlin, Germany

Case study: Metadata in software publications of WavePacket here: Computer Physics Communications

Program summary

Program Title: WavePacket

Program Files doi: <http://dx.doi.org/10.17632/tc5mrh>

Licensing provisions: GPLv3

Programming language: MATLAB

Nature of problem:

Schrödinger's equations are of fundamental importance for distinguishable particles. The solutions of the time-dependent coordinate space, the absolute squares of which are independent equation (TDSE) describes the dynamics of a particle. The role for the simulation, understanding, and prediction of optical physics where systems are driven by temporal fields.

Solution method:

All numerical methods compiled in WavePacket are based on discrete variable representations. Currently implemented are Gauss-Hermite, Gauss-Legendre and FFT-based schemes. The TISE is solved either by direct diagonalization or by propagation in imaginary time. For the TDSE there is a choice between second order differencing, operator splitting and Chebychev polynomial methods.

Additional comments including Restrictions and Unusual features:

The WAVEPACKET program package is rather easy and intuitive to use, providing visualization of quantum dynamics 'on the fly'. It is mainly intended for low-dimensional systems, typically not exceeding three to

Program summary

Program Title: WAVEPACKET

Program Files doi: <http://dx.doi.org/10.17632/9g8b7jychy.1>

Licensing provisions: GPLv3

Programming language: MATLAB

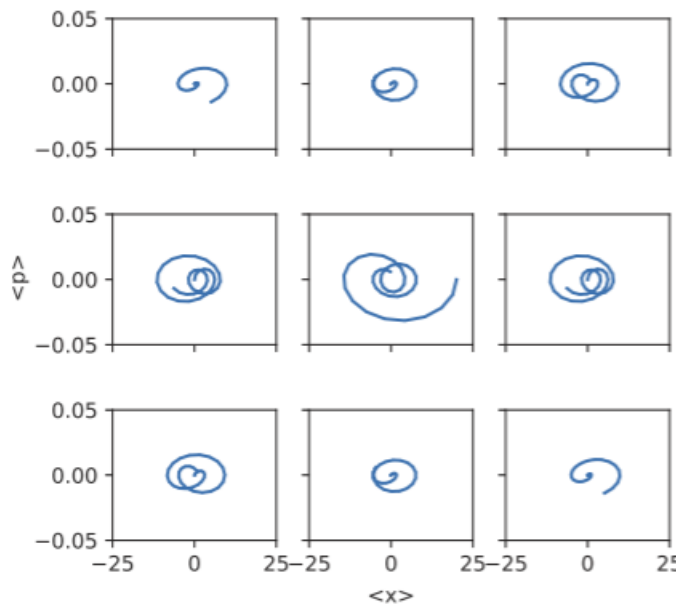
Journal reference of previous version: Comput. Phys. Comm. **213** (2017), 223.

Does the new version supersede the previous version?: The previous article focused on the treatment of closed quantum systems by discrete variable representations and implementation of various numerical algorithms for solving Schrödinger's equations. Complementary to that, the present second part is concerned with open quantum systems and optimal control by external fields. In addition, two approaches to dimension reduction useful in modeling of quantum control are described.

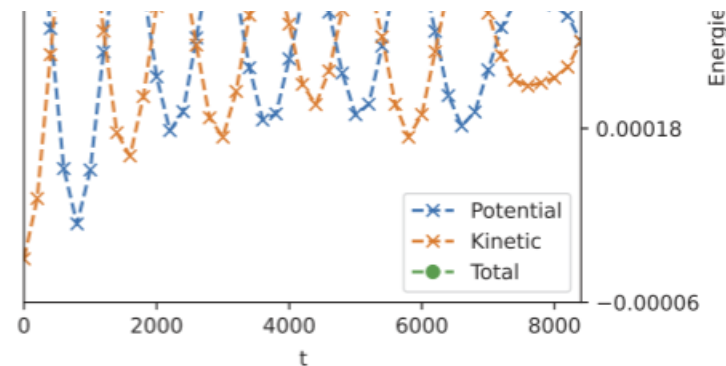
Reasons for the new version: The reason for having a second article on the WavePacket software package lies in the fact that a complete description of the package would have exceeded the scope of a regular article. Several significant features of the WAVEPACKET package are introduced here which could not be mentioned in the first article, due to length constraints.

Summary of revisions: Here we describe the numerical treatment of open quantum systems dynamics modeled by Lindblad master equations. Moreover, we explain the WAVEPACKET functions for optimal

Case study: Contents of software publication of WaveTrain here: Journal of Chemical Physics



```
from wave_train.dynamics.ceom import CEoM
dynamics = CEoM(hamilton = hamilton,
                 num_steps = 50, step_size = 20, sub_steps = 5,
                 solver = 'rk', normalize = 0)
dynamics.coherent(displace=[1.0 if i ==
                             hamilton.n_site//2 else 0.0 for i in
                             range(hamilton.n_site)])
dynamics.solve()
```



QuantumMechanics
hamilton
observe
bra_ket, expect, reduce, ...

TISE
n_levels
solver, eigen
ranks, repeats, conv_eps
e_est, ...
solve, start_solve, update_solve

TDSE
num_steps, step_size, sub_steps
solver, normalize
max_rank, repeats, threshold
fundamental, coherent
solve, start_solve, update_solve

QCMD
num_steps, step_size, sub_steps
solver, normalize
fundamental
solve, start_solve, update_solve
strang_marchuk, pickaback, ...

CEoM
num_steps, step_size, sub_steps
solver
coherent
solve, start_solve, update_solve
runge_kutta, quasi_exact, ...

How do I prepare a citation file for my software?

BibTeX format (.bib file)

- Plain text files with human- *and* machine-readable citation information
- Often found at developers' web pages and/or collaborative platforms
- **Direct implementation into LaTeX / BibTeX codes** of citing pages

Citation file format (.cff file)

- Plain text files, human- *and* machine-readable citation information for software and data sets
- **Support by GitHub | Zenodo | Zotero:**
Upload file "CITATION.cff" to default branch
Automatically linked from repo landing page
Also provided as BibTeX snippet

Codemeta (.json file)

- Similar to CFF,
- but more extensive

menti.com (7)
8766 433

```
cff-version: 1.2.0
message: "If you use this software, please cite it."
authors:
  - family-names: Smith
    given-names: John
    orcid: https://orcid.org/0000-0001-8250-4438
title: "My Research Project"
version: 2.0.4
identifiers:
  - type: doi
    value: 10.5281/zenodo.1234567
date-released: 2023-01-01
```

```
{
  "@context": "https://doi.org/10.5063/schema/codemeta-2.0",
  "@type": "SoftwareSourceCode",
  "license": "https://spdx.org/licenses/GPL-1.0+",
  "codeRepository": "https://sourceforge.net/p/wavepacket/matlab",
  "contIntegration": "https://sourceforge.net/p/wavepacket/matlab",
  "dateCreated": "2008-09-03",
  "dateModified": "2024-01-18",
  "downloadUrl": "https://sourceforge.net/projects/matlab.wavepacket",
  "issueTracker": "https://sourceforge.net/p/wavepacket/matlab/bugs",
  "name": "WavePacket",
  "version": "7.2.0",
  "identifier": "https://sourceforge.net/projects/matlab.wavepacket",
  "description": "WavePacket is a program package for numerical simulation of wave packets in quantum mechanics.",
  "applicationCategory": "Physics",
  "releaseNotes": "https://sourceforge.net/projects/matlab.wavepacket/files/7.2.0",
  "funding": " ",
  "developmentStatus": "active",
  "isPartOf": "https://sourceforge.net/projects/wavepacket/",
  "referencePublication": "https://doi.org/10.1002/JCC.26045",
  "funder": {
    "@type": "Organization",
    "name": " "
  },
  "keywords": [
    "Quantum Dynamics"
  ],
  "programmingLanguage": [
    "MATLAB"
  ],
  "runtimePlatform": [
    "MATLAB"
  ],
}
```

How do I cite software? (recommendations by JORS)

menti.com (8)
8766 433

Software purchased off-the-shelf

- *ProductName. Version. ReleaseDate. Publisher. Location.*
- SuperScience. 1.2. December 2012. ResearchSoftware. Edinburgh, UK.

Software downloaded from the web

- *ProductName. Version. ReleaseDate. Publisher. Location. DOI or URL. DownloadDate.*
- OGSA-DAI REST. 4.2.1. December 2012. OGSA-DAI Project.
<http://sourceforge.net/projects/ogsa-dai>. 27/04/2012.

Software checked-out from a public repository

- *ProductName. Publisher. URL. CheckoutDate. RepositorySpecificCheckoutInformation.*
- OGSA-DAI REST. OGSA-DAI Project. <http://sourceforge.net/projects/ogsa-dai>. 27/04/2012. Check-out: ogsa-dai/branch/ogsadai4.1/, revision 1657.

Software provided by a researcher

- *ProductName. Author. Location. ContactDetails. ReceivedDate.*
- BestFFTroutine ever file. Fred Bloggs, EPCC, The University of Edinburgh, UK. Fred.bloggs@epcc.ed.ac.uk. 27/04/2012.

Make your software citable through publication(s)

- Simplest way to gain recognition of research software as „research output“
- Cultural change from „publish or perish“ or „code and perish“: → code and publish

menti.com (9)
8766 433

What are the benefits of publishing software?*

- Allowing others to reuse your software is both of benefit to you, and to the research community → FAIR
Balancing of “take from” and “give back to” the research community
- By publishing your software, others can scrutinize your code and reproduce your results. They can build on this software to look at new kinds of studies. They can use this for other purposes such as teaching, etc.

Have your software publications refereed (peer review)

- Is the work (the software) new and significant? Is it a sufficient contribution?
- Is the quality of the presentation satisfactory (or can be made so)?
- What revisions and/or changes are necessary or desirable?
- → Improve the quality of the software and/or publication