

RDM in practice:

Best practices for (small) computational projects

Oliver Melchert^{1,2}

¹Leibniz Universität Hannover, Institute of Quantum Optics

²Leibniz Universität Hannover, Cluster of Excellence PhoenixD

Introduction / Outline

■ Introduction

- ▶ 15y of experience with computational projects
- ▶ Researcher in nonlinear optics
- ▶ Modelling + code development + simulation & analysis

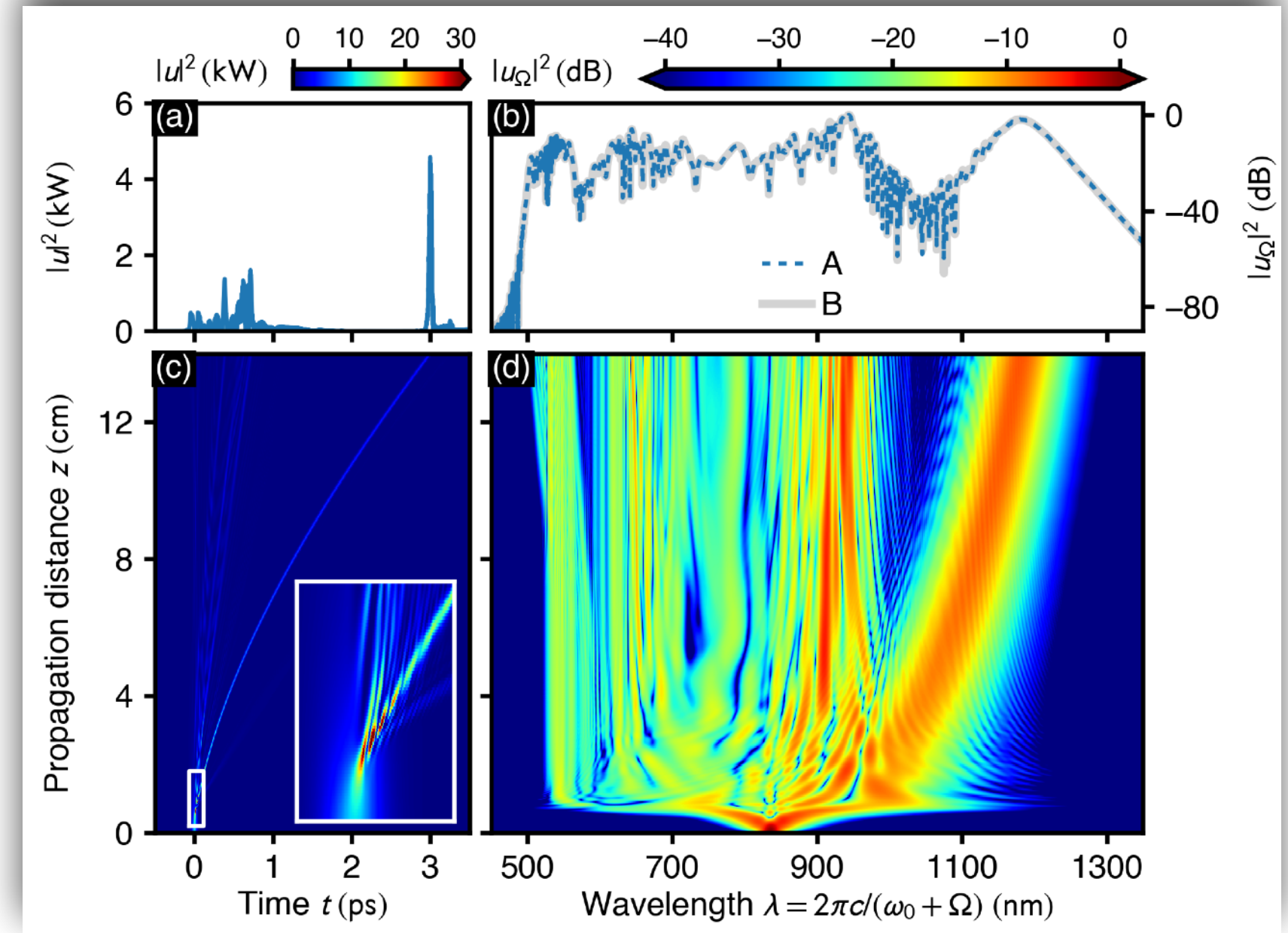
$$i\partial_z \mathcal{E}_\omega + \beta(\omega) \mathcal{E}_\omega + n_2 \frac{\omega}{c} \left((1 - f_R) |\mathcal{E}|^2 \mathcal{E} + f_R \mathcal{E} \mathcal{I}_R \right)_{\omega > 0} = 0$$

$$\mathcal{I}_R = \sum_{\omega} h(\omega) (|\mathcal{E}|^2)_{\omega} e^{-i\omega t}, \quad h(\omega) = \frac{\tau_1^{-2} + \tau_2^{-2}}{\tau_1^{-2} - (\omega + i\tau_2^{-1})^2}$$

■ Outline

- ▶ Day-to-day organisational challenges
- ▶ Further data-management activities
- ➔ 13 best practices + other coping mechanisms
- ➔ Illustrated by means of a project completed in 2022

[OM, A. Demircan; *SoftwareX* 20 (2022) 101232]



Contents lists available at [ScienceDirect](#)

SoftwareX

journal homepage: www.elsevier.com/locate/softx

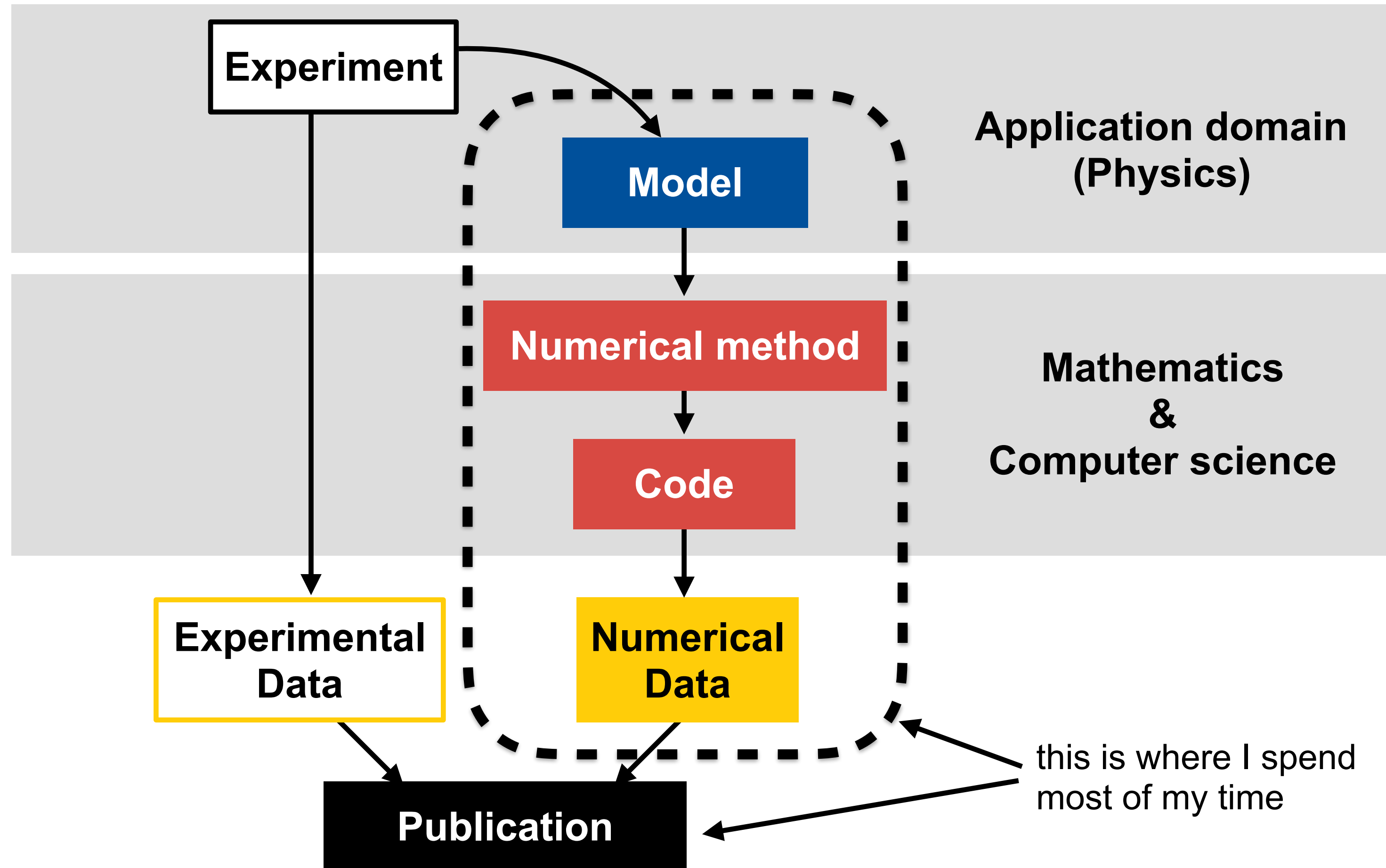
Original software publication

GNLStools.py: A generalized nonlinear Schrödinger Python module implementing different models of input pulse quantum noise

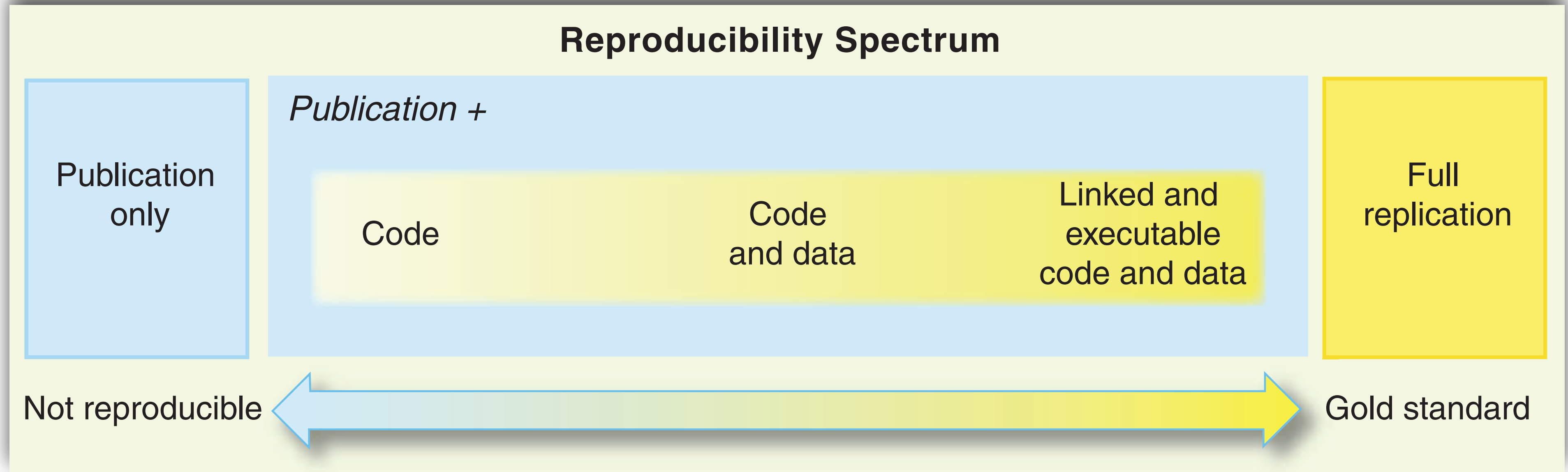
Oliver Melchert*, Ayhan Demircan

Leibniz Universität Hannover, Institute of Quantum Optics (IQO), 30167 Hannover, Germany
Cluster of Excellence PhoenixD (Photonics, Optics, and Engineering – Innovation Across Disciplines), Hannover, Germany

Where do I get my data - typical project life-cycle



What motivates data-management for me?



[D. Peng; *Reproducible Research in Computational Science*; *Science* 334 (2011) 1226]

- **Reproducibility** is key!

- ▶ Standard by which scientific claims are judged
- ▶ Helps others to build upon my work
- ➔ Heightens the **impact** of my work

- **Research data-management (RDM)**

- ▶ The actions I take to make it *easier* for others to *reproduce* my work
- ➔ Good scientific practice

User query on GitHub provided opportunity for concise project

Adding noise and calculating the coherence #3

🔒 Closed

██████████ opened this issue on Nov 17, 2021 · 4 comments



██████████ commented on Nov 17, 2021

First I would to thank the authors for their remarkable effort and for making it publicly accessible

My question is the following:

Considering simulating supercontinuum generation in optical fibers. How to add noise to the initial pulse and calculate the first order coherence. It will very nice if you add a small example of that.

Thank you again



omelchert commented on Nov 22, 2021

Owner

Thank you for this suggestion. A small extension module allowing to add noise to the initial condition and calculate coherence properties is already underway! I will keep you posted on the progress.



<https://github.com>

- Requirements
 - ▶ Modelling
 - ▶ Software engineering
 - ▶ Data analysis
 - ▶ Code & data project

Day-to-day organisational challenges

Poor organisational choices can result in slow progress

- Similar folder layout for all projects
 - ▶ Data is easily searchable / findable
 - ▶ Helps others to navigate your project
 - ➔ Agree on standard of how to do this
- **Best practice 01** - Project root folder:
 - ▶ Chronological order within "results" directories
 - ➔ Reveals chronological order of project
- **Best practice 02** - Readme file:
 - ▶ Overview of project
 - ▶ Amended throughout project life-cycle
 - ➔ Valuable when you collaborate with others
 - ➔ Basis for data management plan (DMP)
- **Best practice 03** - Project subfolders:
 - ▶ Logical order within project subfolders

The screenshot displays a file explorer window with a project directory structure. The left pane shows the root directory with files LICENSE.md and README.md, and folders results, results_EXAMPLES_TALK, and src. The middle pane shows a series of numbered experiment folders (numExp01 to numExp07) with subfolders for scripts, SCgeneration, noise models, and autocorrelation. The right pane shows a folder named pp_fig_FIG03 containing data folders (data_delG1e...M50.000000 to 150.000000), a main_fmas_SC_noise.py file, and another pp_fig_FIG03 folder. A preview window shows a figure named fig03.png, which is a 3x3 grid of plots. The top row shows Intensity I(t) (kW) vs Time t (ps) for three different pulse durations: (a) $t_0 = 28.4$ fs, (b) $t_0 = 56.7$ fs, and (c) $t_0 = 85.0$ fs. The middle row shows Spectrum I_ω (dB) vs Wavelength λ (nm). The bottom row shows Coherence g_2 vs Wavelength λ (nm).

[W. Noble; *A Quick Guide to Organizing Computational Biology Projects*; PLoS Comp. Biol. 5 (2009) e1000424]

Day-to-day organisational challenges

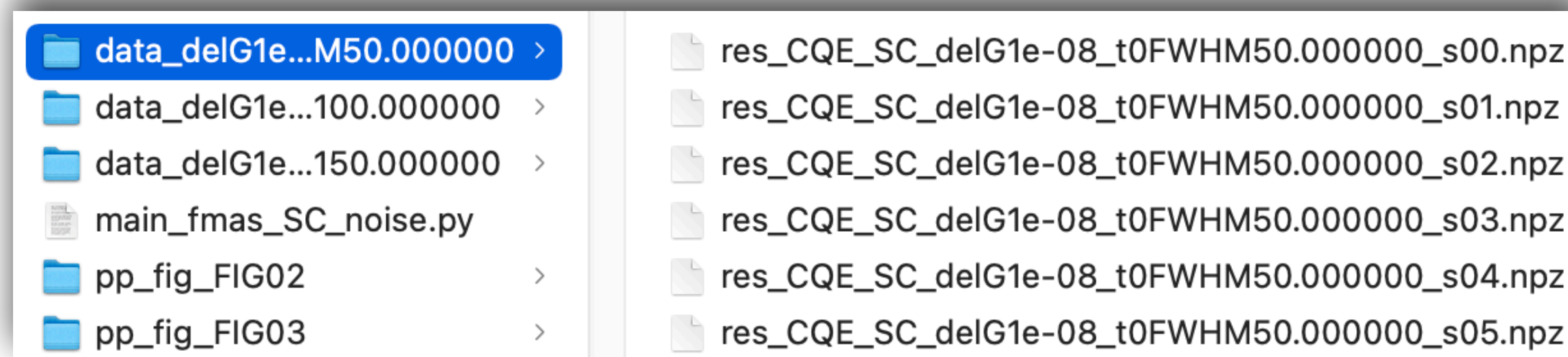
- **Best practice 04** - Keep it “tidy”:
 - ▶ Well documented
 - ▶ Modular
 - ▶ Easy to read and use
 - ➔ Agree on standards of how to do this

```
Returns: (du)
      du (1D numpy-array, cplx floats): instance of time-domain noise
"""
np.random.seed(s0)
N01 = np.random.normal
dt = t[1]-t[0]

# -- REPRESENTATIVE ENERGY OF PHOTON IN BIN
e0 = hBar*w0 # (J)
```

[B. Lee; *Ten simple rules for documenting scientific software*; PLoS Comp. Biol. 14 (2018) e1006561]

- **Best practice 05** - Filenames:
 - ▶ Choose meaningful filenames
 - ➔ Prevents accidental overwriting of files
 - ➔ Reveals what data is contained in file



- **Best practice 06** - Data format:
 - ▶ *Private* data: Choose any data-format you fancy!
 - ▶ *Public* data: Choose format that can be read by many programming languages
 - ➔ Good general choice for structured data: HDF5 <https://www.hdfgroup.org>

Day-to-day organisational challenges

■ Best practice 07 - Metadata:

- ▶ Embed metadata *within* your files
- ▶ Include:
 - Information about software environment
 - "Natural language" description of data
- ▶ Ensure it can be read by humans and machines!
- ➔ Establishes data provenance

```
1 import sys, numpy
2
3 path = sys.argv[1]
4 info = numpy.load(path)['info']
5
6 print(info)
```

```
[00 -- I:INFO, D:DATA
I01 OS-USER: /Users/melchert
I02 OS-ENV: ('Darwin', 'Olivers-iMac', '18.7.0', 'Darwin Kernel Version 18.7.0: Tue Aug 20 16:57:14 PDT 2
I03 OS-PID: 28769
I04 FILE: main_NLPM750_propagationDynamics_qNoise.py
I05 VERSION: 1.1
I06 DATE: 2020-05-13 18:38:40.627047
I07 FNAME: GNLSE_BlowWood_tMax8000.000000_Nt16384_zMax1000000.000000_Nz10000_nSkip50_w02.092929_t0200.000
D01 z (numpy-array, ndim=1): z-axis, i.e. propagation direction axis
D02 t (numpy-array, ndim=1): time axis
D03 w (numpy-array, ndim=1): angular frequency axis
D04 Aw (numpy-array, ndim=2): frequency components of field
```


How to disseminate results prior to peer review?

- **Best practice 08** - Post a preprint
 - ▶ Establishes priority
 - ▶ Broadcasts results early-on
 - ▶ Allows for community feedback
 - ➔ Permanent part of scientific record

The screenshot shows the arXiv preprint interface. At the top left is the Cornell University logo. On the right, a text block reads: "We gratefully acknowledge support from the Simons Foundation and member institutions." Below this is a search bar with "Search...", a dropdown menu set to "All fields", and a "Search" button. The breadcrumb path is "arXiv > physics > arXiv:2206.07526". The main content area shows the title "A generalized nonlinear Schrödinger Python module implementing different models of input pulse quantum noise" by "O. Melchert, A. Demircan", with a submission date of "[Submitted on 13 Jun 2022]". On the right side, there is a "Download:" section with links for "PDF" and "Other formats", and a Creative Commons Attribution (CC BY) license icon. Below the license, it says "Current browse context: physics.comp-ph" with navigation links "< prev" and "next >".

<https://arxiv.org/abs/2206.07526>

How to make code & data citable?

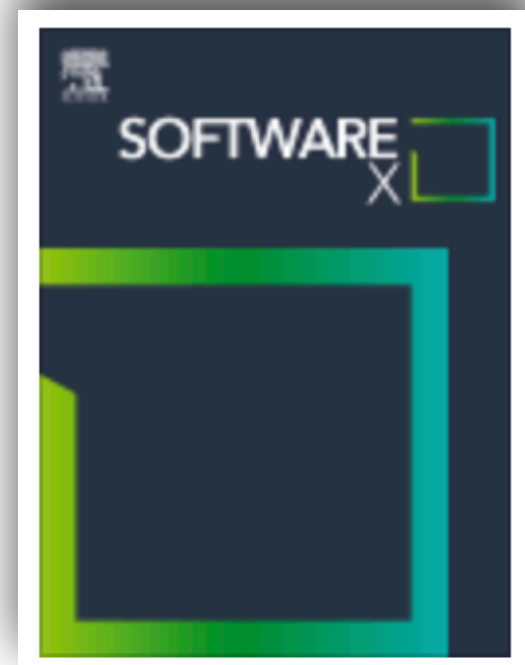
- **Scientific content citation problem**

There **used to be** no standards for content other than articles!

- Journals that support the **FAIR** principle:

- ▶ **F**indable, **A**ccessible, **I**nteroperable, **R**eusable

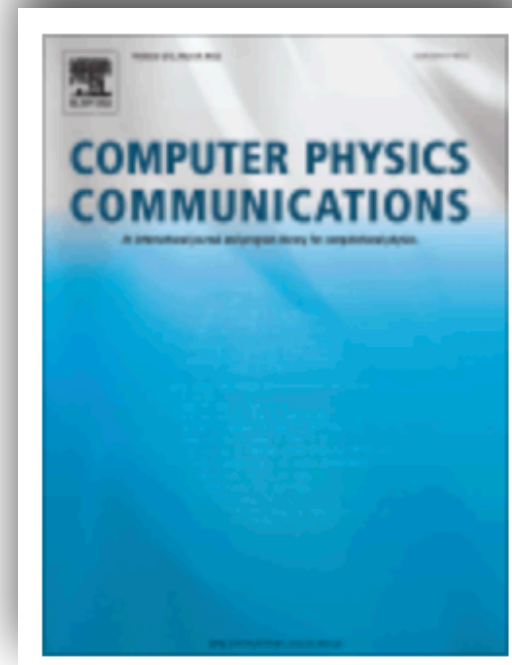
- ➔ Provides literature references for research artefacts



(Open Access)



(Open Access)



[OM, B. Roth, U. Morgner, A. Demircan; *SoftwareX* 10 (2019) 100275]

[OM, A. Demircan; *SoftwareX* 15 (2021) 100741]

[OM, A. Demircan; *Comp. Phys. Commun.* 273 (2022) 108257]

[OM, A. Demircan; *SoftwareX* 20 (2022) 101232]

The screenshot shows the article page for 'GNLStools.py: A generalized nonlinear Schrödinger Python module implementing different models of input pulse quantum noise' in the journal SoftwareX. The page includes the Elsevier logo, ScienceDirect link, article title, authors (Oliver Melchert and Ayhan Demircan), their affiliations (Leibniz Universität Hannover, Institute of Quantum Optics (IQO)), and the article's abstract. It also features an 'ARTICLE INFO' section with article history (received, revised, and accepted dates), keywords (Generalized nonlinear Schrödinger equation, Quantum noise, Spectral coherence, Python), and a 'Code metadata' table. The table lists the current code version (1.0.0), a permanent link to the GitHub repository, the MIT license, and the software code languages, tools, and services used (Python, GitHub). The article is published under a CC BY license.

ARTICLE INFO	
Article history:	
Received 14 June 2022	
Received in revised form 15 September 2022	
Accepted 11 October 2022	
Keywords:	
Generalized nonlinear Schrödinger equation	
Quantum noise	
Spectral coherence	
Python	

Code metadata	
Current code version	1.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00165
Legal Code License	MIT License
Code versioning system used	none
Software code languages, tools, and services used	Python, GitHub
Compilation requirements, operating environments & dependencies	The provided software requires Python, numpy and scipy. The provided examples need Python's matplotlib for figure generation.
If available Link to developer documentation/manual	Documentation provided within code
Support email for questions	melchert@iqo.uni-hannover.de

1. Introduction

The propagation of laser pulses in nonlinear waveguides supports the generation of supercontinuum spectra [1–3]. Starting from a spectrally narrow input pulse, the interplay of linear and nonlinear effects induces tremendous spectral broadening, yielding flat spectra that can extend from the violet to the infrared [4]. Such effects can be achieved, e.g., in photonic crystal fibers (PCFs) [5,6], wherein supercontinuum spectra can be produced using ~100 fs-duration pulses, peak powers ~10 kW and propagation lengths on the order of 1 m [4]. The resulting broad, flat spectra with high spectral density find application, e.g., in optical frequency metrology [7], and optical technologies [2].

A flexible theoretical framework for studying the complex physical processes associated with the generation of supercontinuum spectra is provided by the generalized nonlinear Schrödinger equation (GNLS) [1]. In order to model the propagation dynamics of laser pulses it combines the effects of linear dispersion, pulse self-steepening [8,9], and the Raman effect [10]. This accounts for various processes that support the generation of widely

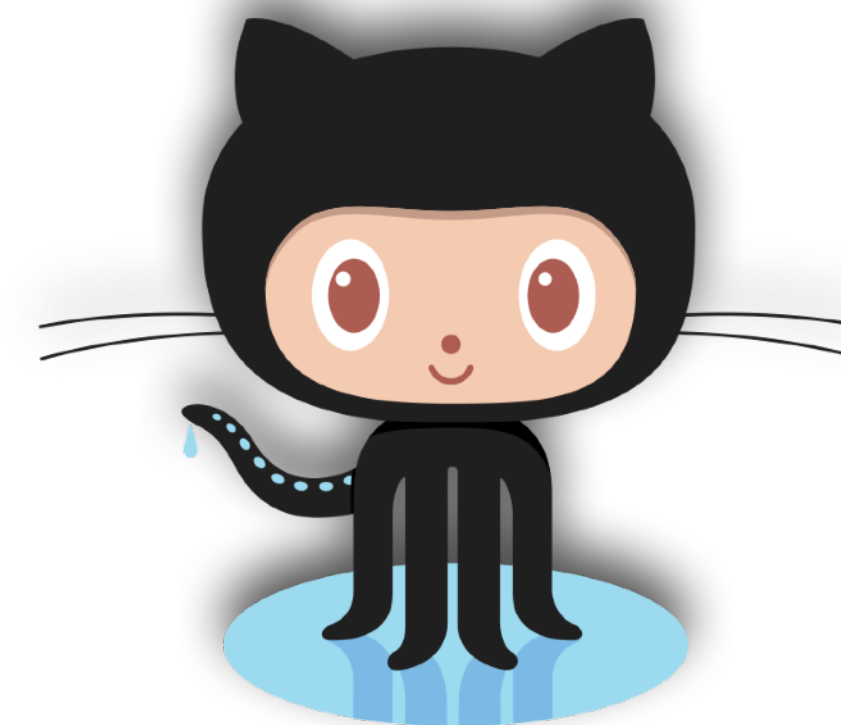
How to make code accessible?

- **Code availability** problem

There are no standards that clarify how to make code available for others!

- **Best practice 09 - Git[Hub/Lab]:**

- ▶ Version control platform allowing to develop/share code
- ➔ Helps making code externally available (repositories can also be kept private!)
- ➔ File size limitations: 100 Mb/file (Git), 2 GB/file (Git-LFS; Large File Service)



<https://github.com>

A screenshot of a GitHub repository page for 'omelchert / GNLStools'. The repository is public and has 1 branch and 0 tags. The commit history shows a commit by Oliver Melchert and Oliver Melchert on Nov 2, 2022, with 20 commits. The commit message is 'updated references in readme'. The file list includes 'results' (added figure 04, 7 months ago), 'src' (added version number, 7 months ago), 'LICENSE.md' (first commit, 8 months ago), and 'README.md' (updated references in readme, 3 months ago).

File	Commit Message	Time Ago
results	added figure 04	7 months ago
src	added version number	7 months ago
LICENSE.md	first commit	8 months ago
README.md	updated references in readme	3 months ago

A graphic titled 'Where the world builds software' showing statistics for GitHub. The text states: 'Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.' The statistics are: 65+ million Developers, 3+ million Organizations, 200+ million Repositories, and 72% Fortune 50.

Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

65+ million Developers	3+ million Organizations	200+ million Repositories	72% Fortune 50
---------------------------	-----------------------------	------------------------------	-------------------

[Y. Perez-Riverol *et al.*; *Ten Simple Rules for Taking Advantage of Git and GitHub*; PLoS Comp. Biol. 12 (2016) e1004948]

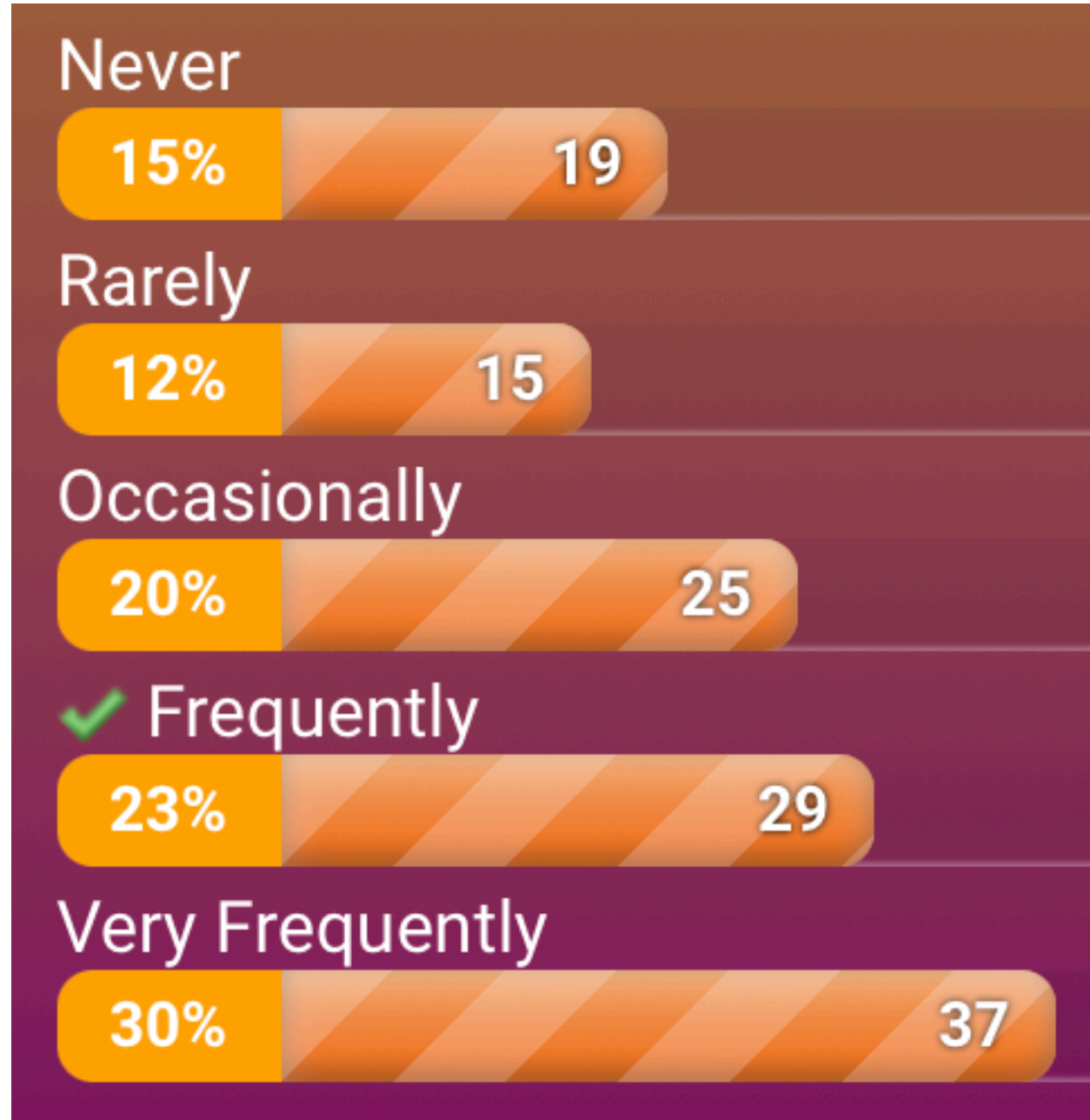
Do readers really want access to code?

- PLOS Computational Biology survey (conducted in 2021)

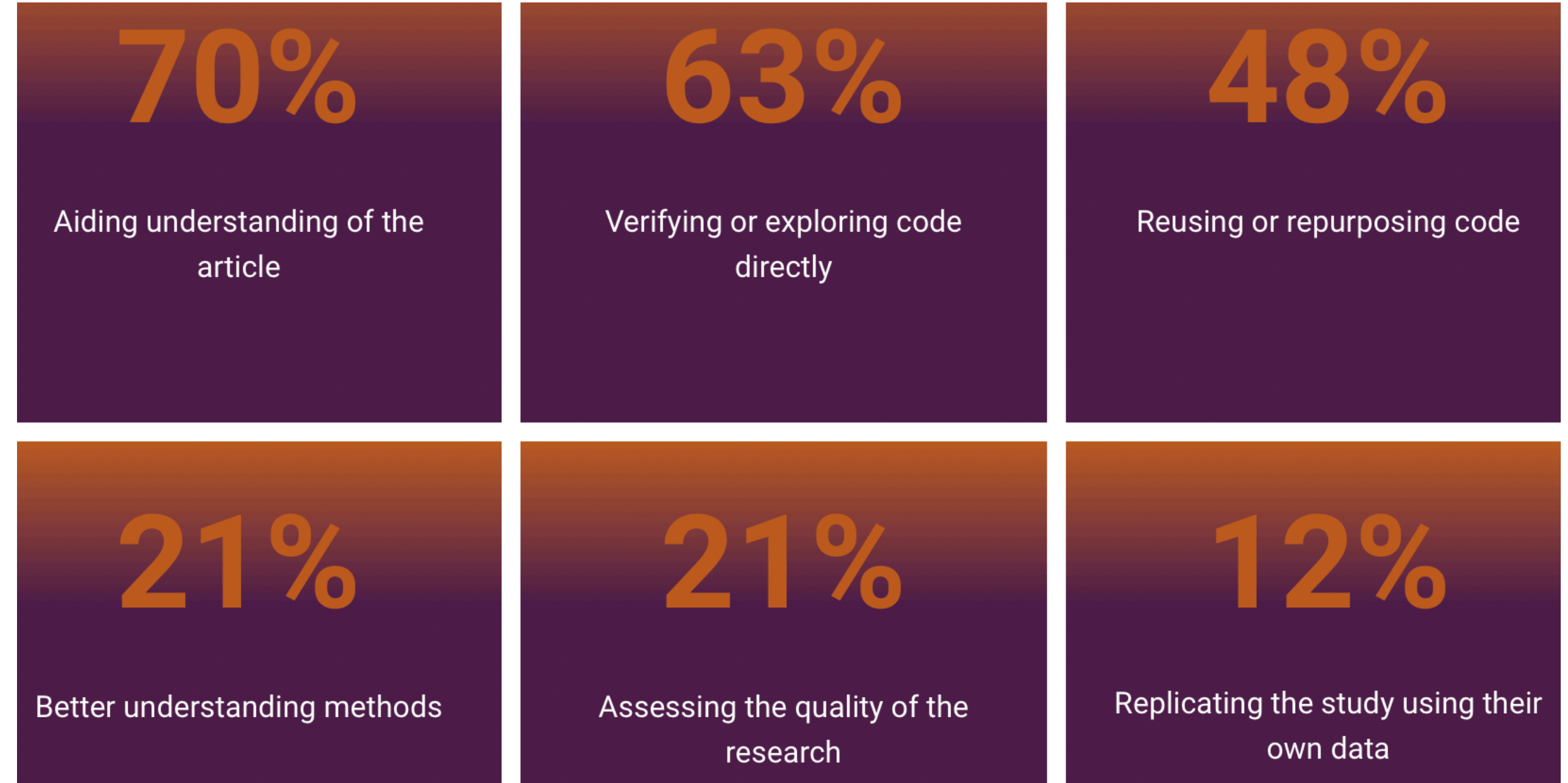
<https://plos.org/open-science/open-code/>



- ▶ Do you consult public code?



- ▶ Aims when consulting public code

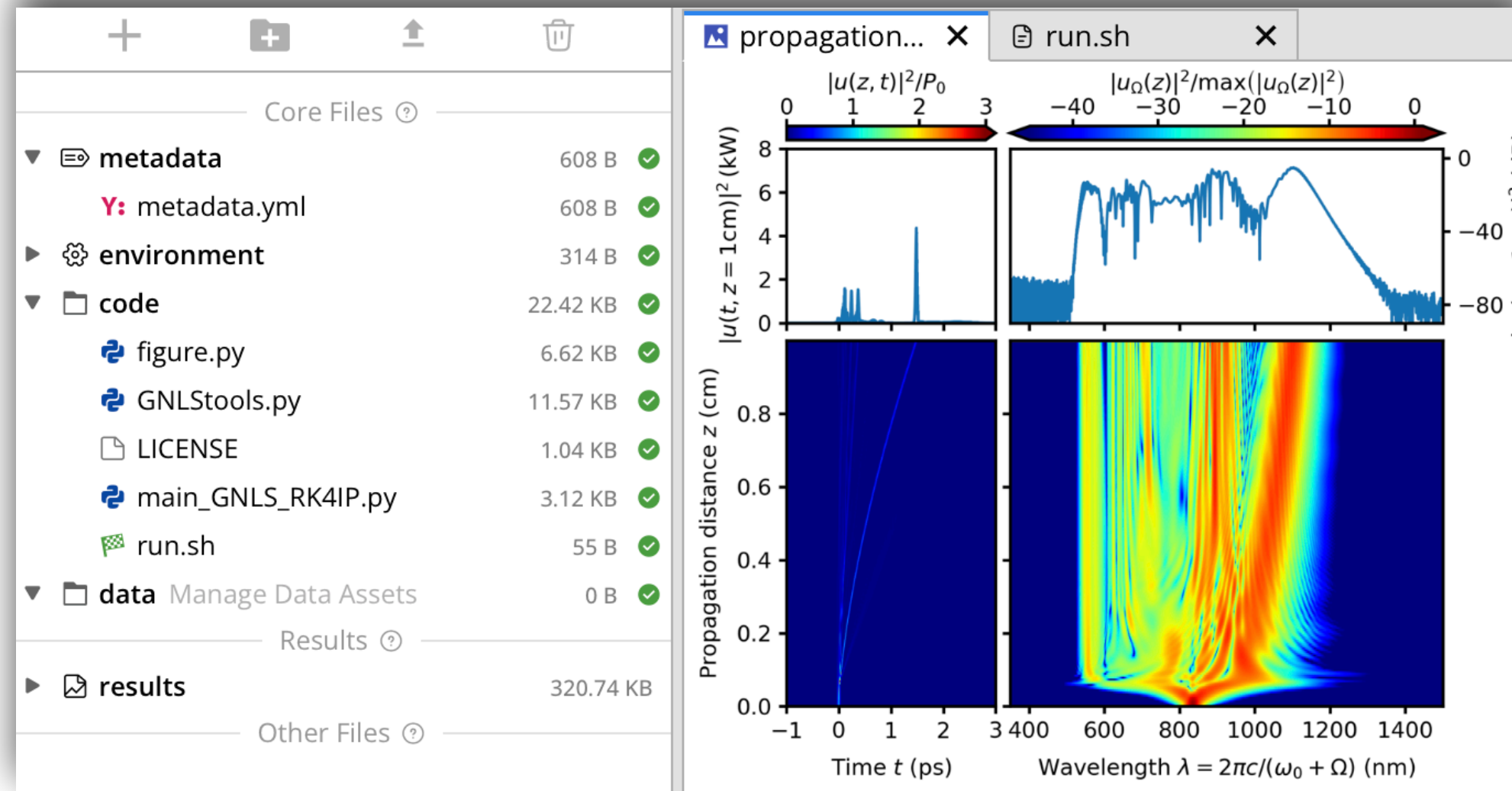


How to make code easily interoperable?

- **Interoperability** problem
- **Best practice 10** - Share on custom environment:
 - ▶ SoftwareX partners with Code Ocean
 - ➔ Enables collaborative computational research
 - ➔ Lets a user test your software without installing it



<https://codeocean.com>

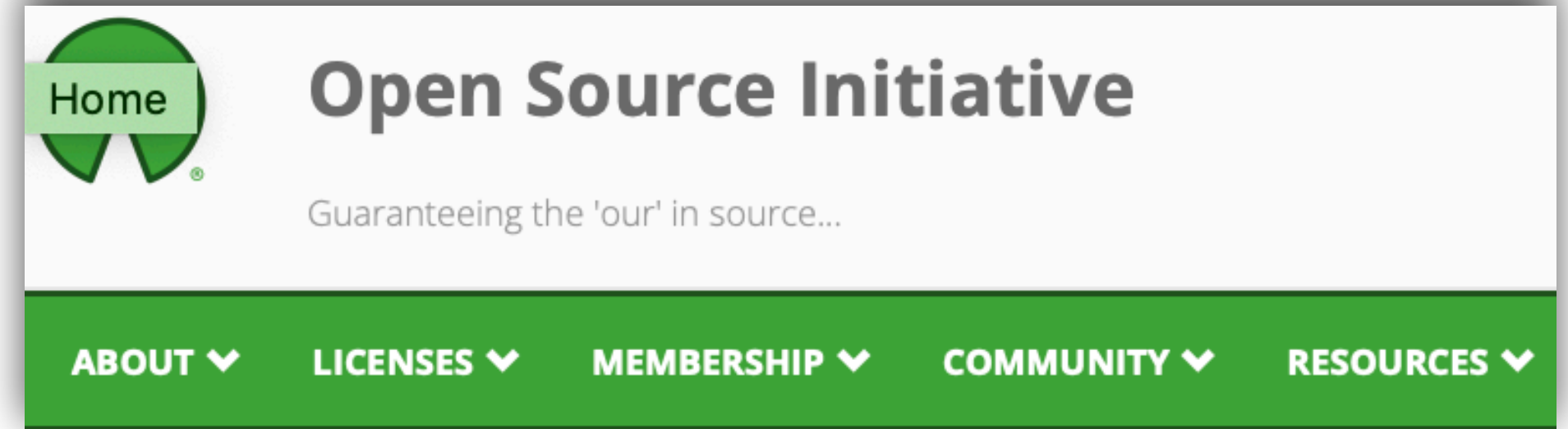


```
propagation... x run.sh x main_GNLS_... x GNLStools.py x figure.py x
36 import sys
37 import numpy as np
38 import matplotlib.pyplot as plt
39 from GNLStools import GNLS, noise_model_01, noise_model_02, noise_model_03
40 from figure import plot_propagation_dynamics
41
42 # -- SET COMPUTATIONAL GRID
43 z, dz = np.linspace(0, 0.1e6, 10000, retstep=True)
44 t = np.linspace(-3500, 3500, 2**13, endpoint=False)
45 w = np.fft.fftfreq(t.size, d=t[1]-t[0])*2*np.pi
46 # -- INSTANTIATE GENERALIZED NONLINEAR SCHRÖDINGER EQUATION
47 gnls = GNLS(
48     w, # (rad/fs)
49     beta_n = [
50         -1.1830e-2, # (fs^2/micron) beta_2
51         8.1038e-2, # (fs^3/micron) beta_3
52         -0.95205e-1, # (fs^4/micron) beta_4
53         2.0737e-1, # (fs^5/micron) beta_5
54         -5.3943e-1, # (fs^6/micron) beta_6
55         1.3486, # (fs^7/micron) beta_7
56         -2.5495, # (fs^8/micron) beta_8
57         3.0524, # (fs^9/micron) beta_9
58         -1.7140, # (fs^10/micron) beta_10
59     ],
60     gamma=0.11e-6, # (1/W/micron)
61     w0 = 2.2559, # (rad/fs)
62     fR = 0.18, # (-)
63     tau1 = 12.2, # (fs)
64     tau2 = 32.0 # (fs)
65 )
66
67 # -- SPECIFY INITIAL PULSE
```

<https://codeocean.com/capsule/4658074>

Set terms on which software might be used!

- **Best practice 11** - Provide a license:
 - ▶ Allows others to reuse your code
 - ▶ Clarify who owns the *intellectual property* (IP) rights
 - ➔ Facilitates access to software (as well as restricts it)



<https://opensource.org>

- Example of how I did this for the discussed project
 - ▶ IP owned by PhoenixD and me
 - ▶ I use the MIT-license
 - ▶ Licence header:

```
Copyright    2022 PhoenixD
Contributor: Oliver Melchert
Affiliation: Theoretical Optics and Computational Photonics Group,
              Institute of Quantum Optics,
              Leibniz Universität Hannover
```

[A. Morin et al.; *A Quick Guide to Software Licensing for the Scientist-Programmer*; PLoS Comp. Biol. 8 (2012) e1002598]

How to make auxiliary data/media citable?

Auxiliary data/media availability problem

There are no standards that clarify how to make data available!

Best practice 12 - DOIs:

- Establish provenance by using Digital Object Identifier (DOI) provided by data hosting platforms like Zenodo or figshare

<https://zenodo.org> <https://figshare.com>

- Allows to share code & data and make it citable via DOIs
- Provides storage and preservation strategy for your data

zenodo Search Upload Communities

January 27, 2023 Poster Open Access

Input Pulse Quantum Noise Models for the Generalized Nonlinear Schrödinger Equation

(max. 50 GB per data set)

Input Pulse Quantum Noise Models for the Generalized Nonlinear Schrödinger Equation
O. Melchert*, A. Demircan
Leibniz Universität Hannover, IQO, Welfengarten 1, 30167 Hannover, Germany

The Generalized Nonlinear Schrödinger Equation (GNLS) [3,4]

$$\partial_z u = i \sum_{n \geq 2} \frac{\beta_n}{n!} (i\partial_t)^n u + i\gamma \left(1 + \frac{i\partial_t}{\omega_0}\right) \left[u(z,t) \int R(t') |u(z,t-t')|^2 dt' \right]$$

Forward and inverse Fourier transforms:
 $F\{u(z,t)\} \equiv \frac{1}{T} \int_{-T/2}^{T/2} u(z,t) e^{i\Omega t} dt = u_\Omega(z)$ $\Omega = (\omega - \omega_0) \frac{2\pi}{T} Z$
 $F^{-1}\{u_\Omega(z)\} \equiv \sum_{\Omega} u_\Omega(z) e^{-i\Omega t} = u(z,t)$ (Angular frequency detuning)

Propagation constant: Raman response: $f_R = 0.18$
 $\beta(\omega) = \sum_{n=2}^{N_{\max}} \frac{\beta_n}{n!} (\omega - \omega_0)^n$ $R(t) = (1 - f_R) \delta(t) + f_R h_R(t)$ $\tau_1 = 12.2$ fs
 $h_R(t) = \frac{\tau_1^2 + \tau_2^2}{\tau_1 \tau_2} e^{-t/\tau_2} \sin(t/\tau_1) \Theta(t)$ $\tau_2 = 32$ fs (Silica fibers)

Pulse energy:
 $E(z) = \int_{-T/2}^{T/2} |u(z,t)|^2 dt = T \sum_{\Omega} |u_\Omega(z)|^2$
 $E(z) = h \sum_{\Omega} n_\Omega (\Omega + \omega_0)$ $n_\Omega \equiv T \frac{|u_\Omega(z)|^2}{h(\Omega + \omega_0)}$ (Number of photons)

Conserves total number of photons [3]:
 $C_{ph}(z) \equiv \sum_{\Omega} n_\Omega = \frac{2\pi}{h\Delta\Omega} \sum_{\Omega} \frac{|u_\Omega(z)|^2}{\omega_0 + \Omega} = \text{const.}$

Semiclassical Models for Input Pulse Quantum Noise

- Include weak classical noise through initial condition
 $u(z=0, t) = u_0(t) + \Delta u(t)$
 $\Delta u(t) = \text{stochastic noise field}$
- Noise field properties:
 - Zero mean: $\langle \Delta u(t) \rangle = 0$
 - Uncorrelated: $\langle \Delta u(t) \Delta u^*(0) \rangle = \sigma^2 \delta(t)$
 - Varies fast in comparison to u_0 : $\sigma^2 = \text{noise variance}$
 - $(z, t)|_{t=t_m} = u_m(z)$ $\Omega_m = m\Delta\Omega$, $\Delta\Omega = 2\pi/T$, $u_\Omega(z)|_{\Omega=\Omega_m} = u_{\Omega_m}(z)$

Model 3

- Samples Fourier representation
- Normally distributed spectral amplitudes
 $\Delta u_{\Omega_m} \equiv \sqrt{\frac{h(\omega_0 + \Omega_m)}{T}} \sqrt{T} e^{-i\Phi}$
- RVs: $\Phi \sim U(0, 2\pi)$ $I \sim \text{Exp}(2)$

photon per mode (exactly): $\frac{1}{2}$ photon per mode (on average)
 $(T|\Delta u_{\Omega_m}|^2) = h(\omega_0 + \Omega) \langle I \rangle = \frac{h(\omega_0 + \Omega)}{2}$

Coherence Properties of Simulated Spectra

Carrier frequency:
 $\omega_0 = 2.2559$ rad/fs
 $\lambda_0 = 835$ nm

Number of photons:
 $n_{ph} \approx 2.4 \times 10^9$

Interpulse coherence [4]:
 $\frac{\langle |u_{\Omega_1, m} u_{\Omega_2, k}^*| \rangle_{m \neq k}}{\sqrt{\langle |u_{\Omega_1, m}|^2 \rangle \langle |u_{\Omega_2, k}|^2 \rangle}}$

Intrapulse coherence [5]:
 $\frac{\langle |u_{\Omega_1, m} u_{\Omega_2, m}^*| \rangle}{\langle |u_{\Omega_1, m}|^2 \rangle \langle |u_{\Omega_2, m}|^2 \rangle}$

Intensity $I(t)$ (kW) vs Time t (ps) for $\tau_0 = 28.4$ fs, $\tau_0 = 56.7$ fs, $\tau_0 = 85.0$ fs.

Spectrum $I(\omega)$ (dB) vs Wavelength λ (nm) for $\tau_0 = 28.4$ fs, $\tau_0 = 56.7$ fs, $\tau_0 = 85.0$ fs.

Coherence ρ vs Wavelength λ (nm) for $\tau_0 = 28.4$ fs, $\tau_0 = 56.7$ fs, $\tau_0 = 85.0$ fs.

SCAN ME

Hosted on **GitHub** Funded by **DFG** Deutsche Forschungsgemeinschaft

Research Strategy within the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453).

Publication date:
January 27, 2023

DOI:
DOI [10.5281/zenodo.7575577](https://doi.org/10.5281/zenodo.7575577)

Keyword(s):
Poster Computational Physics
Generalized nonlinear Schrödinger equation
Quantum noise Spectral coherence Python

Related identifiers:
 Derived from
[10.1016/j.softx.2022.101232](https://doi.org/10.1016/j.softx.2022.101232) (Journal article)
<https://github.com/ElsevierSoftwareX/SOFTX-D-22-00165> (Software)
[10.48550/arXiv.2206.07526](https://arxiv.org/abs/2206.07526) (Preprint)

How to claim ownership of your publications?

- **Accurate attribution of scholarly research output** problem
- **Best practice 13 - ORCID:**
 - ▶ Open Researcher and Contributor ID
 - ➔ Provides unique, persistent identifier to researchers
 - ➔ Creates permanent record of research (not limited to publications)



<https://orcid.org>

GNLStools.py: A generalized nonlinear Schrödinger Python module implementing different models of input pulse quantum noise



SoftwareX
2022-12 | Journal article
DOI: [10.1016/j.softx.2022.101232](https://doi.org/10.1016/j.softx.2022.101232)
Part of ISSN: [2352-7110](https://www.issn.org/issn/2352-7110)
CONTRIBUTORS: Oliver Melchert; Ayhan Demircan

[Show more detail](#)

Do your very best — and talk [tweet] about it!

<https://twitter.com>



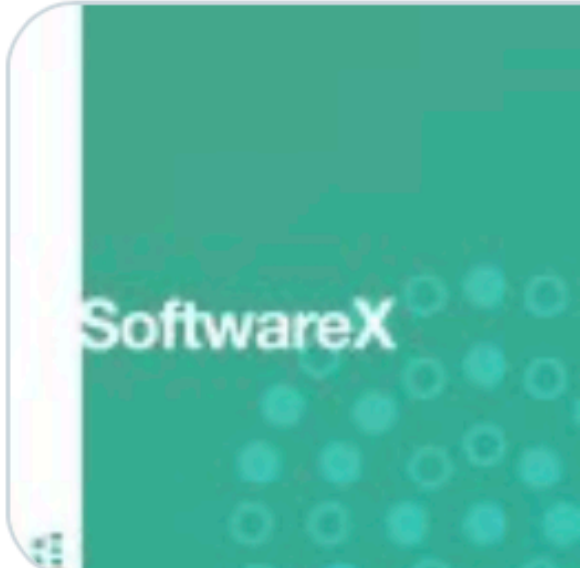
Oliver Melchert @OliverMelchert · Nov 2, 2022

Especially happy about this one, originating from a user-query on gitHub!

[GNLStools.py](#): A generalized nonlinear Schrödinger Python module implementing different models of input pulse quantum noise

doi.org/10.1016/j.soft...

@SoftXJournal @github #Python #Physics



sciencedirect.com
GNLStools.py: A generalized nonlinear Schröding...
We provide Python tools enabling numerical simulation and analysis of the propagation ...



Oliver Melchert
@OliverMelchert

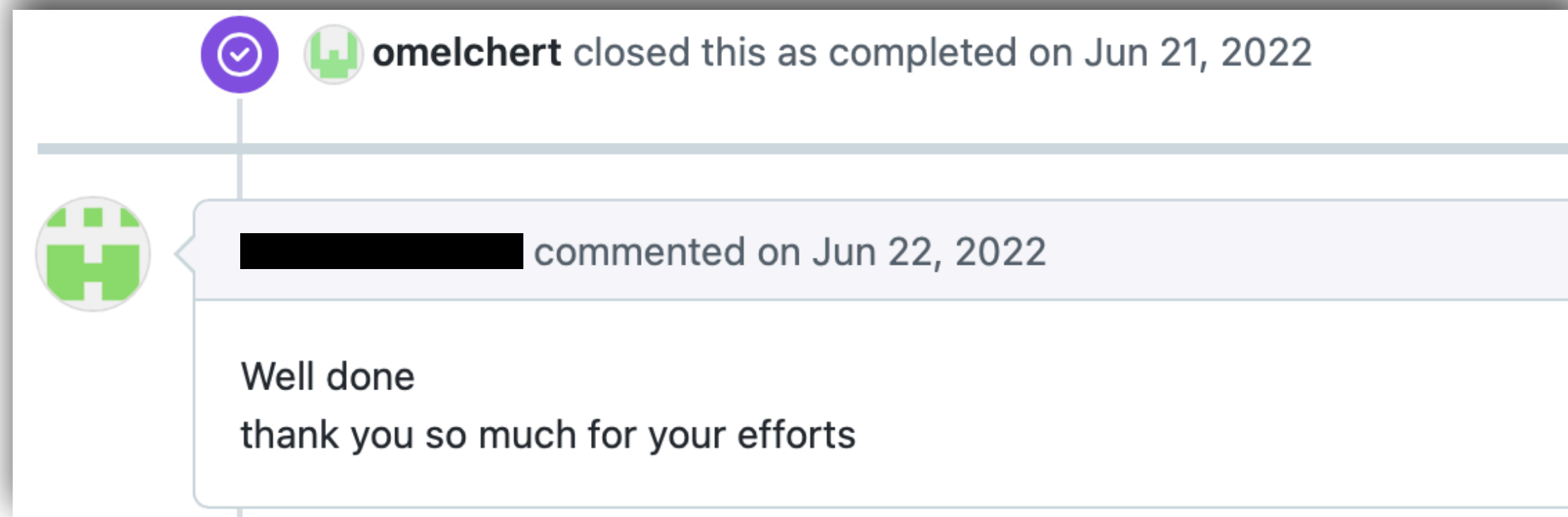


RDM 4 LUH

Leibniz University Research Data Management
@RDM4LUH Follows you

Summary

- RDM strategies helping to reproduce your work with ease
 - ▶ Best practices related to
 - Day-to-day organisational challenges
 - Further data-management activities
 - Quality control (not discussed)
 - ▶ Agree on "how to do things" (when there are no RDM guidelines)
 - ▶ Implementing these strategies takes time and effort
 - ➔ Develop a culture that values RDM



DFG Deutsche
Forschungsgemeinschaft
(EXC 2122, projectID 390833453)

Backup 01 — Questions asked in previous talks

■ Is Github a viable way to store data and what about self-hosted Git solutions (Gitlab etc.)?

- ▶ In principle yes, but it depends on the type of data
- ▶ I use it to host code, only!
- ▶ GitHub has file-size limits:
 - it blocks pushes that exceed 100 MB
 - Large file storage solution
- ▶ Zenodo (max. 50 GB per data set)
<https://zenodo.org/>
- ▶ Consider using seafile (quota depends on project)
<https://seafile.projekt.uni-hannover.de/>
- ▶ Large data: consider High-seas (file sizes up to TB)
<https://high-seas.projekt.uni-hannover.de/>

Using Git LFS, you can store files up to:

Product	Maximum file size
GitHub Free	2 GB
GitHub Pro	2 GB
GitHub Team	4 GB
GitHub Enterprise Cloud	5 GB

<https://docs.github.com/en/repositories/working-with-files/managing-large-files/about-git-large-file-storage>

[Y. Perez-Riverol *et al.*; *Ten Simple Rules for Taking Advantage of Git and GitHub*; PLoS Comp. Biol. 12 (2016) e1004948]

Backup 02 — Questions asked in previous talks

- **Should the source code be made available to everyone and should it be licensed under open source licenses?**
 - ▶ Depends on the data management guidelines
 - ▶ Why you should do it (in my opinion):
 - Freely provided code, whatever the quality, enables others to engage with your work
 - Making it available allows you to also cite it!
 - ▶ If your data management guidelines allow to do so, use an open source license. Code is meant to be used exactly as it is written, so licenses help to avoid plagiarism issues!

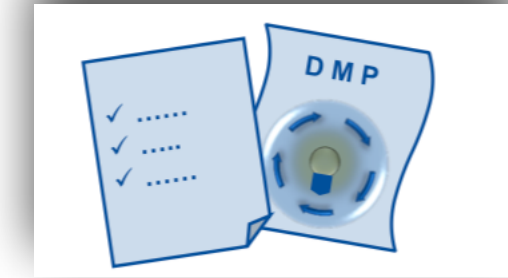
[N. Barnes; *Publish your computer code: it is good enough*; Nature 467 (2010)753]

[A. Morin *et al.*; *A Quick Guide to Software Licensing for the Scientist-Programmer*; PLoS Comp. Biol. 8 (2012) e1002598]

Backup 03 — Questions asked in previous talks

- **In what way should research data (raw data, code, I/O-parameters) be stored?**
 - ▶ Store data in a way so that your results can be reproduced with ease!
 - ▶ Find compromise between
 - *Cost in time*: time it takes to reproduce data when given the code + parameters
 - *Cost in disk space*: amount of space required to store *all* research data
 - ▶ If you can afford to pay some *cost in time*, you can keep the *cost in disk space* pretty small
 - ➔ Applies best to computer simulation studies
 - ➔ In laboratory experiments: *high cost in time* often equals *high cost in person-power*; then you don't want to repeat experiments and keep all the data ... even if the whole experiment went wrong!
 - ▶ Bugs are also research data!
 - ➔ Document them very well and keep track of which program versions were prone to it!

Backup 05 — Maintain a data management plan (DMP)



■ What is a DMP?

A DMP comprises all your data management activities ... in written form!

www.fdm.uni-hannover.de/en
(Tools for developing a DMP)

■ Best practice - Maintain a DMP:

- ▶ Describes how you treat data during the project
- ▶ Describes the roles and responsibilities of collaborators
- ▶ Covers entire project life-cycle
 - Data collection
 - Data organisation
 - Quality control (we skipped this!)
 - Data storage and backup (we also skipped this!)
 - Data documentation
 - Data preservation
 - Sharing with others
- ▶ Might be used to evaluate a projects merit
- ▶ Basis for the DMP is usually the project Readme file!



<https://dmponline.dcc.ac.uk>

[W. Michener; *Ten Simple Rules for Creating a Good Data Management Plan*; PLoS Comp. Biol. 11 (2015) e1004525]