The Journal of Open Source Software

NLSE: A Python package to solve the nonlinear Schrödinger equation

Tangui Aladjidi [1¶], **Clara Piekarski** [1], and **Quentin Glorieux** [1]

1 Laboratoire Kastler Brossel, Sorbonne University, CNRS, ENS-PSL University, Collège de France; 4 Place Jussieu, 75005 Paris, France ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 19 March 2024
Published: unpublished

Summary

The non-linear Schrödinger equation (NLSE) is a general non-linear equation allowing to model the propagation of light in non-linear media. This equation is mathematically isomorphic to the Gross-Pitaevskii equation (GPE) describing the evolution of cold atomic ensembles. Providing a flexible, modern and performant framework to solve these equations is a crucial need to model realistic experimental scenarii.

Statement of need

`NLSE` harnesses the power of pseudo-spectral schemes in order to solve efficiently the following general type of equation:

$$i\partial_t\psi = -\frac{1}{2m}\nabla^2\psi + V\psi + g|\psi|^2\psi.$$

In order to take advantage of the computing power of modern Graphical Processing Units (GPU) for Fast Fourier Transforms (FFT), the main workhorse of this code is the cupy package that maps numpy functionalities onto the GPU using NVIDIA's `CUDA` API. It also heavily uses just-in-time compilation using numba in order to optimize performance while having an easily maintainable Python codebase. Compared to naive Numpy based CPU implementations, this package provides a 100 to 10000 times speedup for typical sizes Figure 2. While optimized for the use with GPU, it also provides a performant CPU fallback layer.

The goal of this package is to provide a natural framework for all physicists wishing to model the propagation of light in non-linear media or the temporal evolution of Bose gases. It can also be used to model the propagation of light in general. It supports lossy, non-linear and non-local media.

It provides several classes to model 1D, 2D propagation, and leverages the array functionalities of numpy like broadcasting in order to allow easy scans of physical parameters to most closely replicate experimental setups.

This code has been developped during the author's PhD thesis (Aladjidi, 2023) and used as the main simulation tool for several publications like (Glorieux et al., 2023) and (Baker-Rasooli et al., 2023).

The Journal of Open Source Software

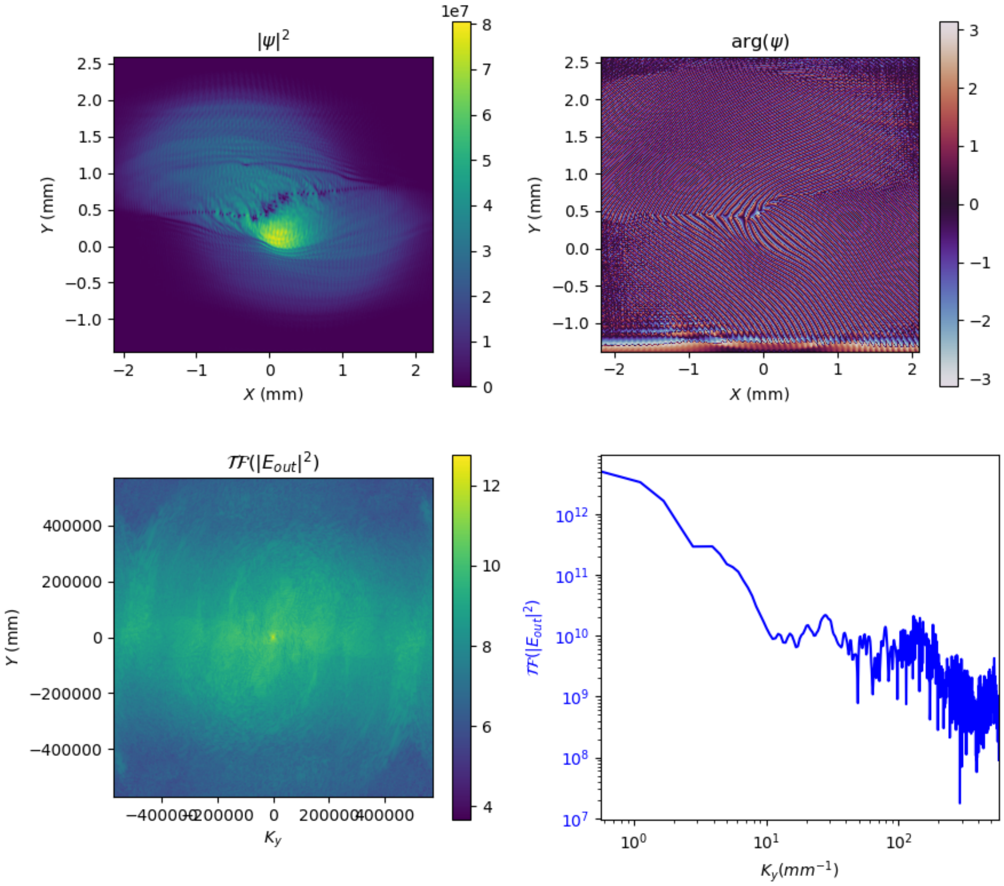

Figure 1: Example of an output of the solver. A shearing layer is observed nucleating vortices, that are attracted towards the center due to an attractive potential. The density and phase of the field are represented as well as the momentum distribution in order to get a quick overview of the state of the field.

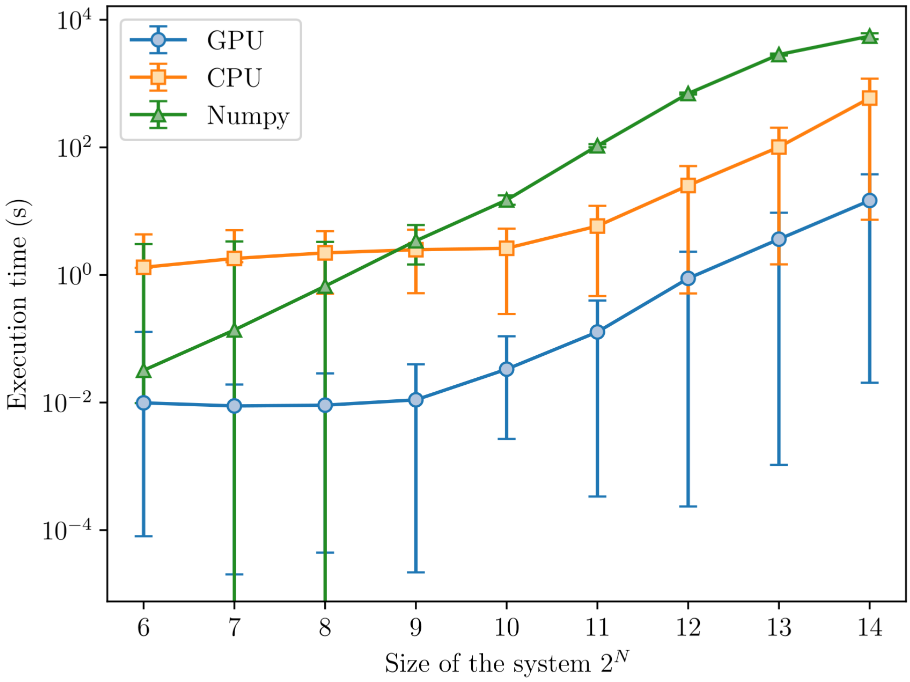

Figure 2: CPU vs GPU benchmark for 1 cm of propagation (200 evolution steps).

Acknowledgements

We acknowledge contributions from Myrann Baker-Rasooli as our most faithful beta tester.

Authors contribution

T.A wrote the original code and is the main maintainer, C.P extended the functionalities to include coupled systems. Q.G supervised the project.

References

Aladjidi, T. (2023). *Full optical control of quantum fluids of light in hot atomic vapors* [PhD thesis]. Sorbonne Université.

Baker-Rasooli, M., Liu, W., Aladjidi, T., Bramati, A., & Glorieux, Q. (2023). Turbulent dynamics in a two-dimensional paraxial fluid of light. *Physical Review A*, *108*(6), 063512. https://doi.org/10.1103/PhysRevA.108.063512

Glorieux, Q., Aladjidi, T., Lett, P. D., & Kaiser, R. (2023). Hot atomic vapors for nonlinear and quantum optics. *New Journal of Physics*, *25*(5), 051201. https://doi.org/10.1088/1367-2630/acce5a