PHOENIX – Paderborn highly optimized solver for two-dimensional nonlinear Schrödinger equations: applications in polariton systems

J. Wingenbach[1,2,*], D. Bauch [1,2], X. Ma[1], R. Schade[3], C. Plessl[3,4], S. Schumacher[1,2,5]

[1]Department of Physics and CeOPP, Paderborn University, Germany
[2]Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany
[3]Paderborn Center for Parallel Computing (PC2), Paderborn University, Warburger Strasse 100, 33098 Paderborn, Germany
[4]Department of Computer Science, Paderborn University, Warburger Strasse 100, 33098 Paderborn, Germany
[5]Wyant College of Optical Sciences, University of Arizona, Tucson, AZ 85721

Variations of the Gross-Pitaevskii equation, a form of the nonlinear Schrödinger equation (NLSE), are central to modeling exciton-polariton condensates, quantum fluids of light-matter quasiparticles in semiconductor microcavities. Numerically efficiently solving these equations with various extensions is key to advancing theoretical insights and enabling new explorations of complex and large nonlinear systems.

Here, we introduce PHOENIX (**P**aderborn **H**ighly **O**ptimized and **E**nergy-efficient solver for **N**onl**I**near Schrödinger equations with e**X**tensions), an open-source, C++-based solver tailored for two-dimensional nonlinear Schrödinger and Gross-Pitaevskii equations, with direct relevance to exciton-polariton systems. PHOENIX is designed for GPU-assisted simulation and supports execution via MATLAB and Python, enabling seamless integration with pre-existing codebases and familiar post-processing workflows. Targeted at efficient single-node CPU and single-GPU simulations, PHOENIX delivers up to three orders of magnitude speedup and energy savings up to 99.8% compared to standard MATLAB implementations of fourth-order Runge-Kutta methods. It supports both single and double precision and performs close to theoretical cache bandwidth limits, especially on high-end consumer GPUs, demonstrating a cache-efficient implementation. Moreover, PHOENIX provides more than 20 pre-implemented time-stepping integrators including Dormand-Prince and Fehlberg-method and is designed for straightforward code extensions for integrators and NLSE extensions [1].

Beyond benchmarking for a wide range of hardware, we showcase the practical utility of PHOENIX through simulations of exciton-polariton condensates. These applications incorporate extensions such as non-Hermiticity, spin-orbit coupling, and stochastic noise, facilitating ensemble evolutions and quantum state tomography. With provided example codes and detailed documentation, PHOENIX serves as a powerful, accessible tool for the OECS community to explore nonlinear wave dynamics and polariton condensates with unprecedented performance and flexibility (list of all applications of PHOENIX published to date in GitHub release [2]).

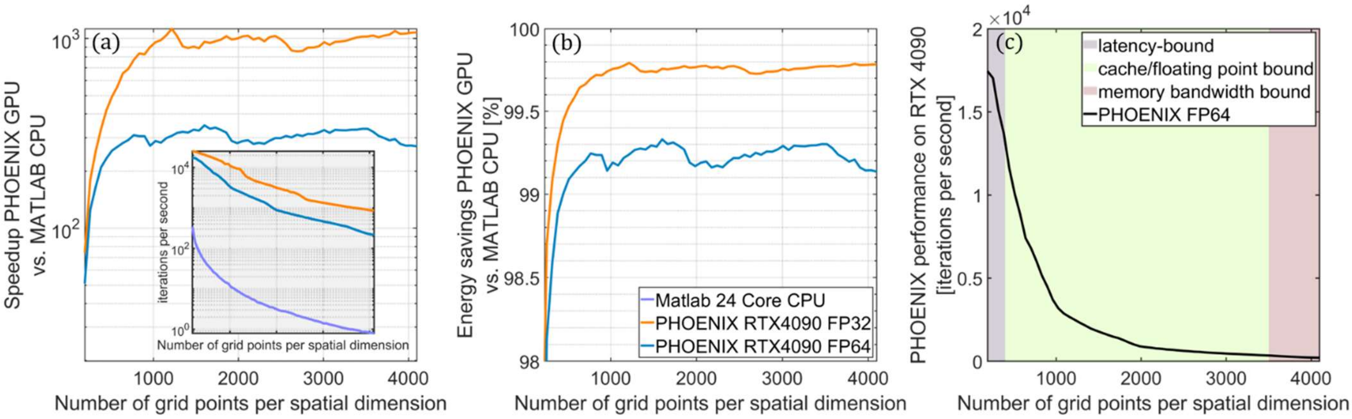

Figure 1: Performance comparison on high-end consumer hardware. (a) Speedup and iterations per second and (b) energy savings of PHOENIX compared to a conventional Runge-Kutta CPU implementation in MATLAB for various grid sizes of a square grid. Simulations use MATLAB R2024a, and PHOENIX compiled with GCC and CUDA. PHOENIX achieves microsecond iteration times even for large grids. (c) PHOENIX performance on RTX 4090 in double precision float and computational bounds.

References

[1] J. Wingenbach, D. Bauch, X. Ma, R. Schade, C. Plessl, and S. Schumacher, *Computer Physics Communications* **315**, 109689 (2025).

[2] Code available at: J. Wingenbach, D. Bauch, X. Ma, R. Schade, C. Plessl, and S. Schumacher, GitHub, (June 2025), https://github.com/Schumacher-Group-UPB/PHOENIX

*Email: jawi1@campus.uni-paderborn.de