Performance Evaluation of Numerical Methods for the Maxwell-Liouville Equations

Michael Riesch and Christian Jirauschek
Institute for Nanoelectronics
Technical University of Munich
Arcisstraße 21, 80333 Munich, Germany
Email: michael.riesch@tum.de

Nikola Tchipev and Hans-Joachim Bungartz
Department of Informatics
Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany

***Abstract*—Several numerical methods for solving the Maxwell-Liouville equations have been published, featuring different accuracy, application range, and computational complexity. We implement the most established method on a multi-core central processing unit (CPU) as well as on a graphics processing unit (GPU) and demonstrate the efficiency of both implementations. The acquired performance values may serve as a reference for future performance measurements of alternative numerical methods.**

I. Introduction

The Maxwell-Liouville equations are a valuable tool for modeling light-matter interaction in nanoscale optical devices in general and quantum cascade lasers (QCLs) in particular [1]. While the electric and the magnetic field are described by Maxwell's equations, the carrier transport is modeled with a density matrix approach (i.e. the Liouville-von Neumann equation), that takes an arbitrary number of quantum mechanical energy levels N into account. For the simplest case $N = 2$ the equation set is equal to the Maxwell-(optical) Bloch equations.

Due to the nonlinearity of the Maxwell-Liouville equations, numerical methods are usually required to solve them. Naturally, the computational workload increases with finer spatial and temporal discretization. However, also the number of energy levels N has a large impact on the workload since N^2 density matrix elements have to be calculated for every discretization step.

In order to cope with the rising computational demands the energy level count is held low in simulations and/or approximations are applied (e.g. the rotating wave approximation (RWA) or the wave packet approximation [2]). These approximations inevitably omit certain features which may be crucial for understanding the dynamics of the simulated device. Therefore, the appropriate way to handle the computational workload is to exploit the parallel computing capabilities offered by modern high-performance computing (HPC) architectures.

In this contribution, we present two implementations of a selected numerical method on different hardware architectures. On the basis of performance measurements we demonstrate the efficiency of both implementations.

This work was funded by the Heisenberg program of the German Research Foundation (DFG, JI115/4-1) and under DFG Grant No. JI 115/9-1.

II. Numerical Methods for the Maxwell-Liouville Equations

Several numerical methods have been proposed in literature featuring different accuracy and computational complexity. The methods described in the following can be applied to the Maxwell-Liouville equations without the aforementioned approximations.

The pioneering work was done by Ziolkowski, Arnold and Gogny [3]. Their approach combines the Finite Difference Time Domain (FDTD) method for Maxwell's equations with the Crank-Nicholson scheme for the optical Bloch equations. While this method can be implemented very easily (the implicit nature of the Crank-Nicholson scheme is resolved by multiple predictor-corrector steps) and gives the correct results for a system with two energy levels, it fails when it is extended to a system with more energy levels [4].

Bidégaray proposed an alternative to the Crank-Nicholson scheme [5]. The alternative approach solves the Liouville-von Neumann equation with the help of operator splitting. Thus, the approach yields correct results for more than two energy levels. However, the required matrix exponential calculation has to be implemented efficiently, e.g. with the help of an external library.

The major drawback of the FDTD method is its numerical dispersion. Hence, the work of Marskar and Österberg [6] – also based on the operator splitting technique mentioned before – incorporates the Pseudo-Spectral Time Domain (PSTD) method for Maxwell's equations. By transforming the fields into a pseudo-spectral domain with respect to the spatial coordinate the spatial derivatives can be calculated easily. As long as the Nyquist-Shannon sampling theorem is not violated, this method is more accurate than FDTD. However, this comes at the cost of multiple Fourier transform calculations, which must be implemented efficiently as well.

We selected the FDTD-Crank-Nicholson method as starting point because despite its shortcomings it is the most established approach.

III. Performance Measurement

The first implementation of the FDTD-Crank-Nicholson method is based on OpenMP and is executed on a multi-core central processing unit (CPU). The second one makes use

978-1-5090-5323-0/17/$31.00 ©2017 IEEE

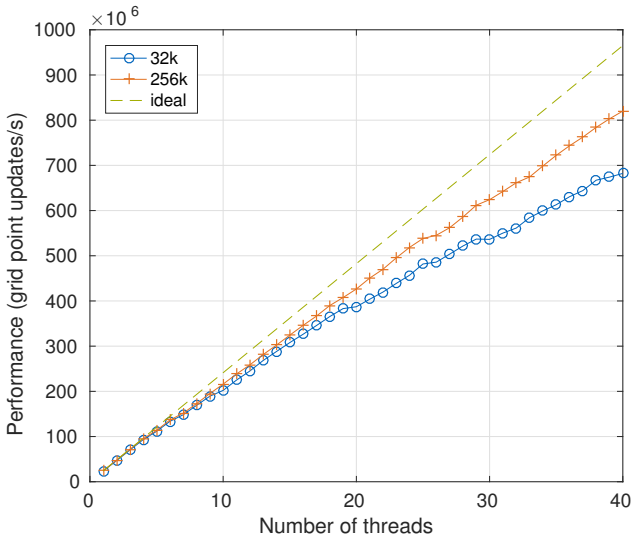

Fig. 1. Performance scaling of the CPU implementation of the FDTD-Crank-Nicholson method for different numbers of grid points.

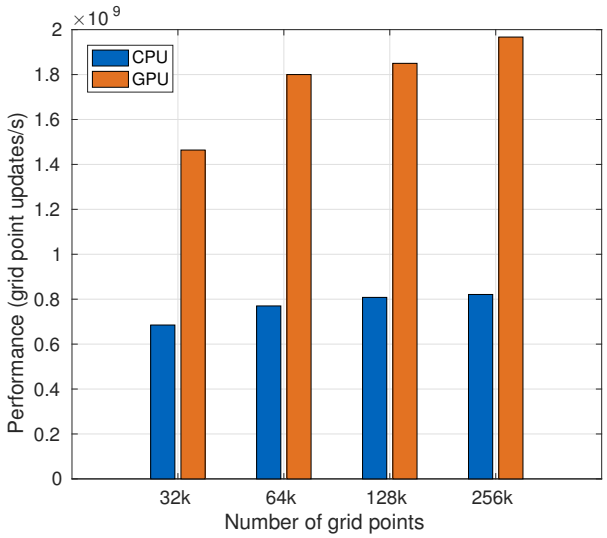

Fig. 2. Performance comparison of the CPU implementation with 40 threads vs. the GPU implementation. Both are implementations of the FDTD-Crank-Nicholson method.

of the NVIDIA CUDA framework in order to outsource the computations to a graphics processing unit (GPU). We used a quad-socket Intel Xeon Processor E7-4870 with 40 cores in total as test CPU, the test GPU is a NVIDIA Tesla K20c. The simulations are carried out at least 10 times to ensure reproducibility of the performance results. The simulation results of the two-level system from [3] serve as first benchmark, our simulation results are consistently checked against theirs.

It should be noted that the time required to set up data structures or transfer results is not included in our execution time measurement. Especially for the GPU this time is not negligible and produces fixed costs.

IV. Results and Discussion

We have measured the performance of both the CPU as well as the GPU implementation of the FDTD-Crank-Nicholson method and discuss the results in the following. First of all, the method clearly benefits from parallelization. Depending on the problem size (i.e. number of grid points) a speedup of 29 (or 34, respectively) is achieved using 40 cores (see Fig. 1). An increase in performance is observed for large problem sizes.

The performance results of the CPU implementation on 40 cores is subsequently compared to the GPU implementation performance. As depicted in Fig. 2, the GPU outperforms the CPU by factor 2.0 to 2.4. This speedup is reasonable if one considers the ratio of theoretical peak performance values between the GPU and the CPU models in question, which is approximately 3. Of course, this comes at the cost of higher programming efforts and the aforementioned setup time and data transfer time. Finally, the GPU implementation exploits the parallel potential better for large problem sizes.

V. Conclusion and Outlook

The results in the section before show that our FDTD-Crank-Nicholson implementations (both CPU and GPU) are already well-optimized and solve the Maxwell-Liouville equations for a two-level system efficiently.

As the next step, we plan to implement the other numerical methods we discussed above. Then, we simulate the same two-level system with the corresponding implementations and compare their performance values with those of the FDTD-Crank-Nicholson implementations. Such a comparison has not been performed yet (to our best knowledge). Also, since performance data have been published in only few cases (additionally, different hardware has been used) a comparison based on literature research is infeasible.

Subsequently, the number of energy levels is increased and the performance is measured. While the impact of the additional workload on the performance cannot be estimated yet, the present results indicate that the parallel computing capabilities of the architectures in question will be exploited better.

References

[1] C. Jirauschek and T. Kubis, "Modeling techniques for quantum cascade lasers," *Appl. Phys. Rev.*, vol. 1, no. 1, p. 011307, 2014.

[2] E. Charron and M. Sukharev, "Non-hermitian wave packet approximation of bloch optical equations," *J. Chem. Phys.*, vol. 138, no. 2, p. 024108, 2013.

[3] R. W. Ziolkowski, J. M. Arnold, and D. M. Gogny, "Ultrafast pulse interactions with two-level atoms," *Phys. Rev. A*, vol. 52, pp. 3082–3094, 1995.

[4] B. Bidégaray, A. Bourgeade, and D. Reignier, "Introducing physical relaxation terms in Bloch equations," *J. Comput. Phys.*, vol. 170, no. 2, pp. 603–613, 2001.

[5] B. Bidégaray, "Time discretizations for Maxwell-Bloch equations," *Numer. Methods Partial Differ. Equ.*, vol. 19, no. 3, pp. 284–300, 2003.

[6] R. Marskar and U. Österberg, "Multilevel Maxwell-Bloch simulations in inhomogeneously broadened media," *Opt. Express*, vol. 19, no. 18, pp. 16 784–16 796, 2011.