2nd URSI AT-RASC, Gran Canaria, 28 May – 1 June 2018

Numerical Methods for the Maxwell-Liouville-von Neumann (MLN) Equations

Michael Riesch*[(1)], Nikola Tchipev[(2)], Hans-Joachim Bungartz[(2)], and Christian Jirauschek[(1)]
(1) Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Arcisstr. 21, 80333 Munich, Germany
(2) Department of Informatics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany

Abstract

In this contribution we evaluate the performance of three numerical methods to solve the Maxwell-Liouville-von Neumann equations (a generalization of the Maxwell-Bloch equations). First, we verify that our implementations of the methods (using the OpenMP standard for parallelization) exhibit good performance scalability. Then, we compare the absolute performance values of the methods and discuss their advantages and drawbacks.

1 Introduction

The Maxwell-Liouville-von Neumann (MLN) equations combine the classical treatment of Maxwell's equations with the quantum mechanical Liouville-von Neumann equations. This semi-classical method is a valuable tool for the investigation of light-matter interaction on small scales and in particular for the modeling of quantum cascade lasers (QCLs).

Due to their nonlinearity, numerical methods are usually required to solve the MLN equations. Often, the rotating wave approximation (RWA) and the slowly-varying envelope approximation (SVEA) are applied in order to reduce the computational workload. These approximations, however, omit substantial features of the solution [1] and will fail when used for the simulation of broadband QCL frequency combs. Hence, we concentrate on numerical methods which do not employ the RWA/SVEA.

Of course, the increased computational workload of numerical methods beyond the RWA/SVEA must be treated appropriately. Several research groups have presented their approaches using different parallelization techniques. However, performance data and/or optimization details have rarely been in the focus of the corresponding publications, the comparison in [2] being a notable exception. We aim to perform a similar comparison for the one-dimensional MLN equations. Therefore, we have extended our previous research [3] by implementing additional methods on top of our common open-source framework and evaluated the performance of the implementations.

Naturally, the accuracy of numerical methods is important and is discussed in this contribution. In particular, the question whether a certain method preserves the properties of the density matrix is crucial [4] and has to be answered.

The rest of this paper is structured as follows: The MLN equations are introduced in the following section. Subsequently, various methods to treat Maxwell's equations and the Liouville-von Neumann equation numerically are discussed. Finally, we present the implementation of the most promising candidates and compare their parallel performance.

2 Description of the Maxwell-Liouville-von Neumann (MLN) Equations

In this contribution we consider Maxwell's equations in one dimension for the field components $E_z(x,t)$ and $H_y(x,t)$,

$$\partial_t E_z = \varepsilon^{-1}\left(-\sigma E_z - \partial_t P_z + \partial_x H_y\right), \tag{1a}$$

$$\partial_t H_y = \mu^{-1}\partial_x E_z, \tag{1b}$$

where x is the propagation direction, y and z are the lateral coordinates, and t is time. The equations include the conductivity σ, permittivity ε, and permeability μ of the active region material as well as the polarization term $P_z(x,t)$.

Furthermore, we consider a sufficiently large number of quantum mechanical systems, which are distributed along the propagation direction of the electromagnetic wave. Each system is described by a $n \times n$ density matrix $\hat{\rho}$ (for n energy levels). The time evolution of each density matrix is given by the Liouville-von Neumann equation

$$\partial_t \hat{\rho} = -\mathrm{i}\hbar^{-1}\left[\hat{H}, \hat{\rho}\right] + \hat{\rho}_{\text{phen}}, \tag{2}$$

where $\hbar$ is the reduced Planck constant, $\hat{H}$ is the Hamiltonian, and $\hat{\rho}_{\text{phen}}$ is a phenomenological term introducing dissipation. It should be noted that for $n = 2$ energy levels the term "Maxwell-Bloch equations" is more common.

Equations (1) and (2) are coupled by the Hamiltonian $\hat{H}(E_z)$, which depends on the electric field, and the polarization

$$P_z = N\,\mathrm{Tr}\left\{\hat{\mu}\hat{\rho}\right\}, \tag{3}$$

where $\hat{\mu}$ is the dipole moment operator and N is the density of quantum mechanical particles in the system.

3 Numerical Treatment of the MLN Equations

There is a multitude of numerical methods that solve Maxwell's equations with different strengths and drawbacks. In the context of coupling Maxwell's equations to the Liouville-von Neumann or Bloch equations, the finite-difference time-domain (FDTD) method [5] is widely used, e.g., in the pioneering work by Ziolkowski et al. [1] and publications that base on it [4, 6, 7, 8, 9]. While FDTD is relatively straightforward to implement and parallelize, its main drawback is the numerical dispersion it introduces. In order to avoid artifacts in the simulation results, the discretization must be chosen very fine.

Therefore, the approaches in [2, 10] use the pseudo-spectral time-domain (PSTD) method [11]. This method minimizes the numerical dispersion, allowing for coarser discretization and thereby reducing the computational workload. However, the implementation of sources, sharp material parameter changes, and boundary conditions is more elaborate [11].

Both FDTD and PSTD have been coupled to the operator splitting (OS) technique introduced in [4, 6], that solves the Liouville-von Neumann equation. This technique is computationally expensive, but guarantees to preserve the positive semidefiniteness of the density matrix. In [4] it was demonstrated that this does not hold for the predictor-corrector (PC) method used in [1, 7], at least when more than two energy levels are considered. Finally, different approaches using the fourth-order Runge-Kutta (RK) method have been presented in related literature (e.g., in [8, 9, 12]). This method is promising with respect to performance, but has not been analyzed in terms of the preservation of the density matrix properties yet.

4 Implementation, Verification and Performance

We implemented three of the most promising approaches (FDTD-PC, FDTD-OS and FDTD-RK) on top of our open-source framework mbsolve [13]. Our implementations were based on the adjoint representation (also known as pseudospin or Bloch vector representation) [14], which has already been used in [1, 7]. This representation exploits the redundancy of information in the density matrix and minimizes the quantities which must be updated at every time step.

The code was parallelized using the OpenMP standard. Following the recommendations from [15], we traded costly synchronization operations for redundant calculations and were able to achieve good performance scalability on a quad-socket Intel Xeon Processor E7-4870 with 40 physical cores in total. The value of the parallel efficiency $E_{40} = S/40$, where S denotes the speedup using all 40 cores, varied between 70 % and 90 % (depending on the numerical method and the problem size).

Prior to the performance measurements, we verified our implementations with the help of the simulation examples from [1] and [16]. Then, we measured the execution times of our implementations for the test setup in [1] and calculated the number of grid point updates per second $P = N_x N_t / t_{\text{exec}}$, where N_x and N_t are the number of spatial and temporal grid points, respectively, and t_{exec} is the measured execution time.

Figure 1 shows the performance comparison of the different methods for varying problem sizes. The predictor-corrector method is significantly faster than both other methods, while the Runge-Kutta approach is about two times faster than the operator splitting technique. The inferior performance of the latter is due to the costly evaluation of matrix exponentials. In the current implementation, the Padé approximation provided by the Eigen library [17] is used. However, we found that the matrix exponentials can be evaluated more efficiently when using the adjoint representation [18]. This optimization will be incorporated in future performance comparisons.

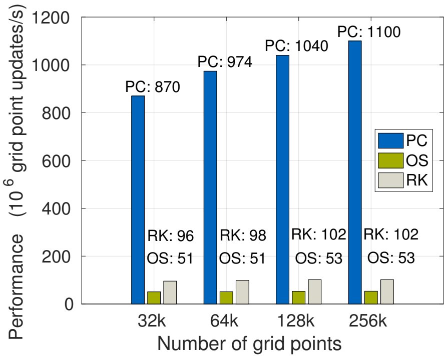

Figure 1. Results of performance measurements of the FDTD method coupled with the predictor-corrector (PC), the operator splitting (OS), and the Runge-Kutta (RK) approach, respectively.

5 Conclusion and Outlook

In summary, we present parallelized implementations of different methods for solving the Maxwell-Liouville-von Neumann equations based on an open-source platform and compare their performance. Although the operator splitting method is the computationally most intensive method and shows the least performance, it guarantees physically realistic results. The Runge-Kutta method performs better, but it must be clarified whether this method preserves the density matrix properties. This analysis will be one of our next steps.

In order to improve accuracy (and maybe even to reduce the execution time of the simulation), the PSTD method can be employed. Therefore, future work will concentrate on the implementation and evaluation of the PSTD method.

As already mentioned, there are many numerical methods to solve Maxwell's equations apart from FDTD and PSTD. Hence, further investigations should consider the inclusion of a different method in order to improve accuracy and performance of the simulations.

6 Acknowledgments

This work was supported by the German Research Foundation (DFG) within the Heisenberg program (JI 115/4-2) and under DFG Grant No. JI 115/9-1. Nikola Tchipev acknowledges the funding provided by Intel as part of the Intel Parallel Computing Center ExScaMIC-KNL. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (www.lrz.de). Finally, the authors thank Mariem Kthiri and Sebastian Senninger for their help in the development of the mbsolve project.

References

[1] R. W. Ziolkowski, J. M. Arnold, and D. M. Gogny, "Ultrafast pulse interactions with two-level atoms," *Phys. Rev. A*, vol. 52, no. 4, pp. 3082–3094, 1995.

[2] O. Saut and A. Bourgeade, "Numerical methods for the bidimensional Maxwell–Bloch equations in nonlinear crystals," *J. Comput. Phys.*, vol. 213, no. 2, pp. 823–843, 2006.

[3] M. Riesch, C. Jirauschek, N. Tchipev, and H.-J. Bungartz, "Performance evaluation of numerical methods for the Maxwell-Liouville equations," in *Numerical Simulation of Optoelectronic Devices (NUSOD), 2017 International Conference on*. IEEE, 2017, pp. 223–224.

[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] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, 2005.

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

[7] G. Slavcheva, J. M. Arnold, I. Wallace, and R. W. Ziolkowski, "Coupled Maxwell-pseudospin equations for investigation of self-induced transparency effects in a degenerate three-level quantum system in two dimensions: Finite-difference time-domain study," *Phys. Rev. A*, vol. 66, no. 6, p. 63418, 2002.

[8] M. Sukharev and A. Nitzan, "Numerical studies of the interaction of an atomic sample with the electromagnetic field in two dimensions," *Phys. Rev. A*, vol. 84, no. 4, p. 043802, 2011.

[9] W. Cartar, J. Mørk, and S. Hughes, "Self-consistent Maxwell-Bloch model of quantum-dot photonic-crystal-cavity lasers," *Phys. Rev. A*, vol. 96, no. 2, p. 023859, 2017.

[10] 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.

[11] Q. H. Liu, "The PSTD algorithm: A time-domain method requiring only two cells per wavelength," *Microw. Opt. Technol. Lett.*, vol. 15, no. 3, pp. 158–165, 1997.

[12] A. Deinega and T. Seideman, "Self-interaction-free approaches for self-consistent solution of the Maxwell-Liouville equations," *Phys. Rev. A*, vol. 89, no. 2, p. 022501, 2014.

[13] M. Riesch and C. Jirauschek, "mbsolve: An open-source solver tool for the Maxwell-Bloch equations," https://github.com/mriesch-tum/mbsolve, 2017.

[14] F. T. Hioe and J. H. Eberly, "*N*-level coherence vector and higher conservation laws in quantum optics and quantum mechanics," *Phys. Rev. Lett.*, vol. 47, no. 12, pp. 838–841, 1981.

[15] S. Krishnamoorthy, M. Baskaran, U. Bondhugula, J. Ramanujam, A. Rountev, and P. Sadayappan, "Effective automatic parallelization of stencil computations," *SIGPLAN Not.*, vol. 42, no. 6, pp. 235–244, 2007.

[16] X. Song, S. Gong, and Z. Xu, "Propagation of a few-cycle laser pulse in a V-type three-level system," *Opt. Spectrosc.*, vol. 99, no. 4, pp. 517–521, 2005.

[17] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[18] M. Riesch and C. Jirauschek, "Numerical method for the Maxwell-Liouville-von Neumann equations using efficient matrix exponential computations," *https://arxiv.org/abs/1710.09799*, 2017.