Performance Evaluation of Numerical Methods for the Maxwell-Liouville-von Neumann Equations

Michael Riesch (ID) · **Nikola Tchipev** (ID) · **Sebastian Senninger** (ID) · **Hans-Joachim Bungartz** · **Christian Jirauschek** (ID)

Abstract The Maxwell-Liouville-von Neumann (MLN) equations are a valuable tool in nonlinear optics in general and to model quantum cascade lasers (QCLs) in particular. Several numerical methods to solve these equations with different accuracy and computational complexity have been proposed in related literature. We present an open-source framework for solving the MLN equations and parallel implementations of three numerical methods using OpenMP. The performance measurements demonstrate the efficiency of the parallelization.

1 Introduction

The Maxwell-Liouville-von Neumann (MLN) equations are essential in the field of nonlinear optics in general and a valuable tool to model quantum cascade lasers (QCLs) in particular. In our current research we describe the light-matter interaction in the active region of a QCL with Maxwell's equations in 1D for the electric and magnetic field components $E_z(x,t)$ and $H_y(x,t)$,

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

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

Michael Riesch · Sebastian Senninger · Christian Jirauschek
Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Arcisstr. 21, 80333 Munich, Germany
E-mail: michael.riesch@tum.de

Nikola Tchipev · Hans-Joachim Bungartz
Department of Informatics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany

where x is the propagation coordinate, y and z are the transversal directions, and t is time. Furthermore, z denotes the quantum well growth direction. In Maxwell's equations we consider the conductivity σ, permittivity ε, and permeability μ of the active region material as well as the polarization term $P_z(x,t)$.

The medium is represented by a (sufficiently large) number of quantum mechanical multilevel systems in propagation direction (Jirauschek and Kubis 2014). The behavior of each system (described by the $n \times n$ density matrix $\hat{\rho}$ for n discrete energy levels) is governed by the Liouville-von Neumann equation

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

where $\hbar$ is the reduced Planck constant, $\hat{\rho}_{\mathrm{phen}}$ is a phenomenological term that includes scattering processes, and the Hamiltonian $\hat{H} = \hat{H}_0 + \hat{H}_I(E_z)$ consists of a time-independent part $\hat{H}_0$ and a time-dependent interaction part $\hat{H}_I = -\hat{\mu} E_z$. In the latter term, $\hat{\mu}$ denotes the dipole moment operator.

Both equations are coupled by the interaction term $\hat{H}_I$ in Eq. (2) (which depends on the electric field E_z) and the polarization term P_z in Eq. (1) (or its derivative, respectively) which is calculated as

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

where N is the density of quantum mechanical particles in the system.

In general, the MLN equations cannot be solved analytically due to their nonlinearity and numerical methods are required. Naturally, these methods have to be accurate and stable. In particular, the properties of the density matrix have to be maintained even for large simulation end times. Depending on the numerical method the discretization step size in time and space must be chosen very small, leading to an increase of the computational workload. Most approaches use the rotating wave approximation (RWA) to allow larger discretization step sizes (Wang et al. 2007). However, Ziolkowski et al. (1995) demonstrated that this approximation omits certain features which may be crucial for understanding the dynamics of the simulated device. In particular, this will be the case for QCL broadband frequency combs, which are one of the main topics of our research.

For some QCL simulations (e.g. Tzenov et al. 2016, 2017) it is important to include more than two energy levels, since various effects are only visible if at least three levels are considered. The number of energy levels has a large impact on the computational workload, since n^2 density matrix entries have to be calculated for every discretization step. As a consequence, the numerical methods must be implemented efficiently using parallelization techniques in order to handle the computational workload. This is particularly the case if more dimensions or additional effects are considered in the simulations.

In order to determine the most efficient numerical method, Saut and Bourgeade (2006) implemented three methods for the bidimensional MLN equations and compared their performance. We aim to extend this work and shift the focus towards parallelization. In this paper we present an open-source framework that serves as common base for different numerical methods. On top of this framework we implement three methods using the OpenMP standard for parallelization and discuss their performance and parallel efficiency.

The rest of the paper is organized as follows: Section 2 gives an overview of the numerical methods found in literature. The common framework and the implementation of three of the most promising methods are presented in Sec. 3. The results of the performance comparison are given and discussed in Sec. 4. Finally, we conclude with our findings and give an outlook on our future work.

2 Numerical Methods for the Maxwell-Liouville-von Neumann Equations

Several numerical methods with different degrees of accuracy and computational complexity have been proposed in literature over the last two decades. The pioneering work was done by Ziolkowski et al. (1995), who combined the finite-difference time-domain (FDTD, see Taflove and Hagness 2005) method for Maxwell's equations with the Crank-Nicolson scheme for the Liouville-von Neumann equation for two energy levels (i.e. the optical Bloch equations). The FDTD method uses a staggered grid in time and space (Yee grid, see Yee 1966) and is straightforward to implement (update equations, sources and boundary conditions). The main drawback is that small discretization step sizes are required to keep the numerical dispersion to a minimum. Slavcheva et al. (2002) extended this approach to two dimensions and an arbitrary number of energy levels. The implicit nature of the Crank-Nicolson scheme is resolved by multiple predictor-corrector (PC) steps.

Bidégaray et al. (2001) performed a rigorous stability analysis of the FDTD-PC approach and found that the Crank-Nicolson scheme may yield unrealistic results for a system with more than two energy levels. An alternative was proposed in Bidégaray (2003), where an operator splitting (OS) technique was employed to update the Liouville-von Neumann equation. This OS technique bases on the calculation of matrix exponentials, which is computationally expensive.

Therefore, Saut and Bourgeade (2006) as well as Marskar and Österberg (2011) used the pseudo-spectral time-domain (PSTD, see Liu 1997) method, since the spatial discretization step size may be increased without impairing accuracy. Using this method the same operator splitting technique can be employed for fewer spatial grid points, which yields a performance boost. Additionally, the PSTD method is superior to the FDTD method in terms of numerical dispersion. The main drawback, however, is that the implementation of sources or sharp material parameter changes becomes increasingly difficult. Also, absorbing boundaries must be included in order to prevent the wrap-around effect (Liu 1997).

Finally, several research groups (Sukharev and Nitzan 2011; Deinega and Seideman 2014; Cartar et al. 2017) used the FDTD method combined with the fourth-order Runge-Kutta scheme for the Liouville-von Neumann equation. This approach has – to the best of our knowledge – not been subject to a rigorous stability analysis yet, but is a very promising candidate with respect to performance.

3 Implementations

Each numerical method in the previous section was implemented and tested by the respective authors. However, the simulation source codes are not publicly available which renders the extension of an existing solution difficult. Therefore, we decided to start our own project featuring a common framework for all numerical methods to be implemented. A schematic of the mbsolve project (Riesch and Jirauschek 2017) is depicted in Fig. 1.

Based on the literature research results in the previous section we selected three candidates, namely the FDTD-predictor-corrector (FDTD-PC) approach by Ziolkowski et al. (1995), the FDTD-operator splitting (FDTD-OS) method by Bidégaray (2003), and a candidate similar to the FDTD-Runge-Kutta (FDTD-RK) approaches reported in literature. We implemented all three methods for the simplest case, i.e. for a system with two energy levels.

In all approaches the updates of the Liouville-von Neumann equation as well as the updates of the electric and magnetic field are spatially independent and can be performed in parallel. However, there exists a spatial dependency between the electric and the magnetic
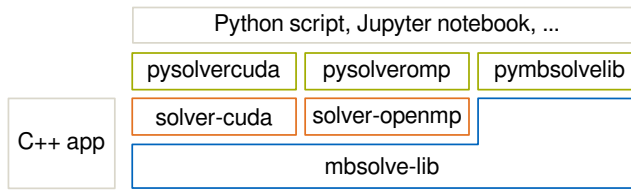
Fig. 1 Schematic of the mbsolve project. The mbsolve-lib library contains the base classes (e.g. solver, device, scenario). Various solver libraries which specialize the solver base class and implement numerical methods may be added. It is possible to use different parallelization techniques (e.g. OpenMP or NVIDIA CUDA). However, hybrid approaches that combine two techniques are not supported at the moment. All libraries can be loaded into a C++ application or a Python script.
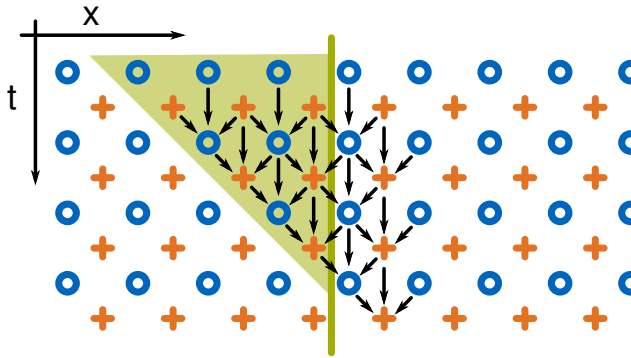

Fig. 2 Schematic overview of the FDTD-PC method. The circles and crosses represent the spatiotemporal discretization of the magnetic field and electric field, respectively. Additionally, the crosses denote the positions of the quantum mechanical systems. The data dependencies are marked with arrows. The grid is partitioned into tiles (the green bar denotes a tile border) which can be subsequently assigned to different threads. Two costly synchronizations between the threads are required during each time step. The synchronizations can be traded against redundant calculations (marked with the green triangle).

field. Therefore, synchronizations are required between the updates of the electric field and the updates of the magnetic field. With the help of the OpenMP standard it was quite straightforward to implement the calculations in parallel. However, after the first performance tests it became apparent that the straightforward implementation is suboptimal due to the number of synchronization calls. Following the recommendations from Krishnamoorthy et al. (2007), we traded synchronization calls versus redundant calculations and were thus able to exploit the parallelization potential more efficiently. Figure 2 describes this approach shortly. The implementation details as well as the causes for the performance improvements compared to the straightforward version will be discussed thoroughly in a separate publication.

The operator splitting method requires efficient calculations of matrix exponentials. These calculations are performed using the Eigen library (Guennebaud et al. 2010), which offers an implementation of the Padé approximation algorithm.

Finally, the update mechanism of the FDTD-RK approach is outlined shortly. In this description the time $t^n = n\Delta t$ corresponds to the time step n, where Δt is the time step size. First, the electric field value of the next time step $E_z^{n+1} = E_z(t^{n+1})$ is calculated using the FDTD method. Both the old and the updated value of the field are stored. Then, the density

matrix is updated with the rule for the fourth-order Runge-Kutta method (for a thorough description, see Hairer et al. 1993)

$$\hat{\rho}^{n+1} = \hat{\rho}^n + \Delta t \left(k_1 + 2k_2 + 2k_3 + k_4 \right) / 6, \tag{4}$$

where $k_1 = f(t^n, \hat{\rho}^n)$, $k_2 = f(t^{n+1/2}, \hat{\rho}^n + \Delta t k_1/2)$, $k_3 = f(t^{n+1/2}, \hat{\rho}^n + \Delta t k_2/2)$, and $k_4 = f(t^{n+1}, \hat{\rho}^n + \Delta t k_3)$. The function f denotes the right hand side of the Liouville-von Neumann equation (2), which is time-dependent due to the electric field. The required electric field values E_z^n and E_z^{n+1} have already been calculated and stored, the missing value $E_z^{n+1/2} \approx \left(E_z^n + E_z^{n+1} \right)/2$ is approximated by the average between the old and the updated field value.

4 Performance Measurements

We tested the implementations of the methods mentioned in the previous section using the setup from Ziolkowski et al. (1995) and verified the correctness of our implementations. In this setup, the density of quantum mechanical particles in the medium is $N = 10^{24} \, \mathrm{m}^{-3}$. The two-level systems are described with the Hamiltonian

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \hbar \omega_{12} \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 0 & \mu_{12} \\ \mu_{12} & 0 \end{bmatrix} E_z, \tag{5}$$

where $\omega_{12} = 4\pi \times 10^{14} \, \mathrm{s}^{-1}$ is the transition frequency and $\mu_{12} = 10^{-29} \, \mathrm{A\,s\,m}$ is the dipole moment between the two levels, and the phenomenological term

$$\hat{\rho}_{\mathrm{phen}} = \begin{bmatrix} T_1^{-1} \rho_{22} & -T_2^{-1} \rho_{12} \\ -T_2^{-1} \rho_{21} & -T_1^{-1} \rho_{22} \end{bmatrix}, \tag{6}$$

where $T_1 = 10^{-10} \, \mathrm{s}$ and $T_2 = 10^{-10} \, \mathrm{s}$ are upper level lifetime and dephasing time, respectively. Initially, the lower level is fully populated. Furthermore, the medium has vacuum permittivity $\varepsilon = \varepsilon_0$, vacuum permeability $\mu = \mu_0$, and zero conductivity $\sigma = 0$.

For the first performance measurements, the grid contained 32 768 spatial grid points, which corresponds to a spatial discretization size of 4.578 nm. By setting the Courant number $C = 0.5$, we chose a time step size of 7.635×10^{-18} s. In later measurements, we varied the number of grid points and modified the time step size accordingly using the same Courant number.

We measured the execution time on a quad-socket Intel Xeon Processor E7-4870 with 40 physical cores in total as function of the number of used threads and the number of grid points, respectively. The measurements were executed five times in order to ensure reproducibility. Also, variations (e.g. caused the operating system) were reduced by choosing the simulation end time sufficiently long. In all measurements the execution time exceeded 25 s. It should be noted that we used the FDTD version with redundant calculations in all measurements.

The performance is calculated as number of grid point updates per time unit, i.e. $P = N_x N_t / t_{\mathrm{exec}}$, where N_x and N_t are the number of spatial and temporal grid points, respectively, and t_{exec} is the measured execution time. Hence, the metric can be used to compare the performance of problems with different sizes. The parallel efficiency
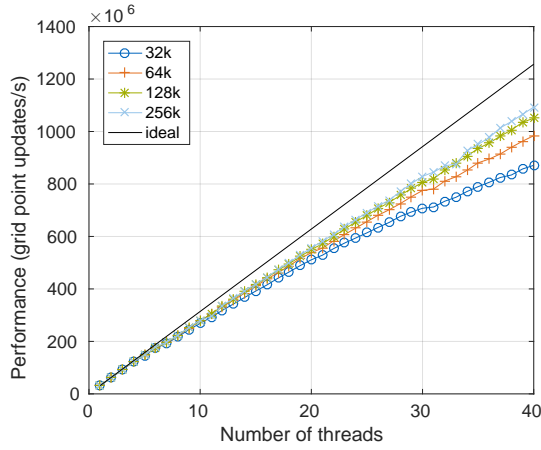
$$E_T = \frac{S_T}{T} \tag{7}$$

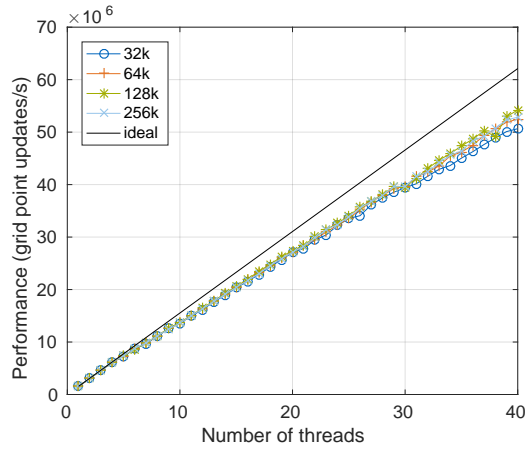**Fig. 3** Performance scalability of the FDTD-PC implementation for different problem sizes.

Fig. 4 Performance scalability of the FDTD-OS implementation for different problem sizes.

is used as metric for the scalability. In this equation, T is the number of used threads and $S_T = P_T/P_1$ denotes the achieved speedup.

Figure 3 demonstrates that the FDTD-PC implementation scales well with the number of used threads and achieves a parallel efficiency of $E_{40} = 69.6\%$ for 32k spatial grid points. The efficiency increases with the problem size and reaches 87 % for 256k grid points.

We performed the same measurement for the FDTD-OS implementation and found that the parallel efficiency increased to $E_{40} = 81.6\%$ for 32k spatial grid points (see Fig. 4). However, the absolute performance decreased by factor ≈ 17 (up to ≈ 20 for larger problems) compared to the FDTD-PC implementation. This is caused by the higher computational
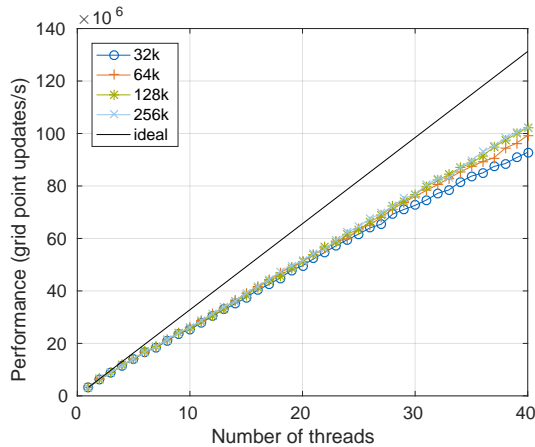
Fig. 5 Performance scalability of the FDTD-RK implementation for different problem sizes.

workload due to the matrix exponential calculations and in agreement with our expectations.

Finally, the results for the same measurement with the FDTD-RK method are depicted in Fig. 5. The parallel efficiency is $E_{40} = 71.2\%$ for this method (for 32k spatial grid points), while the absolute performance is ≈ 9 to ≈ 10 times lower compared to the FDTD-PC method.

5 Conclusion and Outlook

In this paper we presented a framework for solvers for the Maxwell-Liouville-von Neumann equations and three implementations of different numerical methods that base on it. Since the source code is publicly available, it may serve as base not only for our future extensions but also for those of other research groups.

We measured the performance of all three implementations and verified that the parallelization is efficient on current multi-processor systems. With regard to absolute performance, the FDTD-OS is computationally more expensive but will preserve the properties of the density matrix when applied to more than two energy levels. The FDTD-RK approach performs better than the FDTD-OS method (by factor ≈ 2), but is still ≈ 10 slower than FDTD-PC. An analysis of this method with respect to the preservation of physical properties has not been published yet.

In order to reduce the computational workload of the FDTD-OS approach, the FDTD method could be replaced with the PSTD method. This will be one of the next steps of our research. Additionally, we will consider further alternatives and investigate their stability and performance.

Furthermore, it would be of interest to evaluate the performance on different hardware architectures (e.g. the Intel Xeon Phi or graphics processing units (GPUs)). In future work, we will compare the performance of the most promising numerical method on those architectures.

Supplementary Material

See Riesch and Jirauschek (2017) for source code, build instructions, and basic documentation. In this paper we used the development branch `riesch2017b`, the Eigen library version 3.3.4, and the Intel C++ compiler 17.0. The build was configured with the arguments

`-DCMAKE_BUILD_TYPE=RelWithDebInfo -DARCH=SSE3 -DMAT_EXP_METHOD=PADE`

using CMake and built with GNU make. The scripts in `tools/loadleveler/riesch2017b` provide the necessary information to run the simulations.

References

Bidégaray B (2003) Time discretizations for Maxwell-Bloch equations. Numer Methods Partial Differ Equ 19(3):284–300

Bidégaray B, Bourgeade A, Reignier D (2001) Introducing physical relaxation terms in Bloch equations. J Comput Phys 170(2):603–613

Cartar W, Mørk J, Hughes S (2017) Self-consistent Maxwell-Bloch model of quantum-dot photonic-crystal-cavity lasers. Phys Rev A 96:023,859

Deinega A, Seideman T (2014) Self-interaction-free approaches for self-consistent solution of the Maxwell-Liouville equations. Phys Rev A 89:022,501

Guennebaud G, Jacob B, et al. (2010) Eigen v3. http://eigen.tuxfamily.org

Hairer E, Nørsett SP, Wanner G (1993) Solving Ordinary Differential Equations I, 2nd edn. Springer-Verlag Berlin Heidelberg

Jirauschek C, Kubis T (2014) Modeling techniques for quantum cascade lasers. Appl Phys Rev 1(1):011,307

Krishnamoorthy S, Baskaran M, Bondhugula U, Ramanujam J, Rountev A, Sadayappan P (2007) Effective automatic parallelization of stencil computations. SIGPLAN Not 42(6):235–244

Liu QH (1997) The PSTD algorithm: A time-domain method requiring only two cells per wavelength. Microw Opt Technol Lett 15(3):158–165

Marskar R, Österberg U (2011) Multilevel Maxwell-Bloch simulations in inhomogeneously broadened media. Opt Express 19(18):16,784–16,796

Riesch M, Jirauschek C (2017) mbsolve: An open-source solver tool for the Maxwell-Bloch equations. https://github.com/mriesch-tum/mbsolve

Saut O, Bourgeade A (2006) Numerical methods for the bidimensional Maxwell–Bloch equations in nonlinear crystals. J Comput Phys 213(2):823–843

Slavcheva G, Arnold JM, Wallace I, Ziolkowski RW (2002) 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 66(6):63,418

Sukharev M, Nitzan A (2011) Numerical studies of the interaction of an atomic sample with the electromagnetic field in two dimensions. Phys Rev A 84:043,802

Taflove A, Hagness SC (2005) Computational Electrodynamics: The Finite-Difference Time-Domain Method. Artech House

Tzenov P, Burghoff D, Hu Q, Jirauschek C (2016) Time domain modeling of terahertz quantum cascade lasers for frequency comb generation. Opt Express 24(20):23,232–23,247

Tzenov P, Burghoff D, Hu Q, Jirauschek C (2017) Analysis of operating regimes of terahertz quantum cascade laser frequency combs. IEEE Trans THz Sci Technol 7(4):351–359

Wang CY, Diehl L, Gordon A, Jirauschek C, Kärtner FX, Belyanin A, Bour D, Corzine S, Höfler G, Troccoli M, Faist J, Capasso F (2007) Coherent instabilities in a semiconductor laser with fast gain recovery. Phys Rev A 75:031,802

Yee K (1966) Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans Antennas Propag 14(3):302–307

Ziolkowski RW, Arnold JM, Gogny DM (1995) Ultrafast pulse interactions with two-level atoms. Phys Rev A 52:3082–3094