Numerical Method for the Maxwell-Liouville-von Neumann Equations using Efficient Matrix Exponential Computations

MICHAEL RIESCH* AND CHRISTIAN JIRAUSCHEK

Department of Electrical and Computer Engineering, Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany
*michael.riesch@tum.de

Abstract: We present a novel method to solve the Maxwell-Liouville-von Neumann (MLN) equations in an accurate and efficient way without invoking the rotating wave approximation (RWA). The method is a combination of two established concepts, namely the operator splitting method as well as the adjoint representation of the Lie algebra SU(N) (or pseudospin representation). The former concept ensures the accuracy of the approach, but is computationally expensive. The latter concept provides an efficient representation of the problem and two optimization possibilities. We have implemented and verified both optimization approaches and demonstrate that substantial speedup can be achieved.

OCIS codes: (000.3860) Mathematical methods in physics; (000.4430) Numerical approximation and analysis; (020.1670) Coherent optical effects; (190.7110) Ultrafast nonlinear optics.

References and links
1. C. Jirauschek and T. Kubis, “Modeling techniques for quantum cascade lasers,” Appl. Phys. Rev. 1, 011307 (2014).
2. D. Dietze, A. Benz, G. Strasser, K. Unterrainer, and J. Darmo, “Terahertz meta-atoms coupled to a quantum well intersubband transition,” Opt. Express 19, 13700–13706 (2011).
3. R. W. Ziolkowski, J. M. Arnold, and D. M. Gogny, “Ultrafast pulse interactions with two-level atoms,” Phys. Rev. A 52, 3082–3094 (1995).
4. 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 66, 043418 (2002).
5. G. Slavcheva, J. M. Arnold, and R. W. Ziolkowski, “Ultrashort pulse lossless propagation through a degenerate three-level medium in nonlinear optical waveguides and semiconductor microcavities,” IEEE J. Sel. Top. Quant. 9, 929–938 (2003).
6. B. Bidégaray, A. Bourgeade, and D. Reignier, “Introducing physical relaxation terms in Bloch equations,” J. Comput. Phys. 170, 603–613 (2001).
7. B. Bidégaray, “Time discretizations for Maxwell-Bloch equations,” Numer. Methods Partial Differ. Equ. 19, 284–300 (2003).
8. O. Saut and A. Bourgeade, “Numerical methods for the bidimensional Maxwell–Bloch equations in nonlinear crystals,” J. Comput. Phys. 213, 823–843 (2006).
9. R. B. Sidje, “Expokit: a software package for computing matrix exponentials,” ACM T. Math. Software 24, 103–105 (1998).
10. R. Marskar and U. Österberg, “Multilevel Maxwell-Bloch simulations in inhomogeneously broadened media,” Opt. Express 19, 16784–16796 (2011).
11. D. H. Hall, “Solution of two level system using 3rd order Magnus expansion,” https://arxiv.org/abs/1610.05951 (2016).
12. L. Guduff, A. J. Allami, C. van Heijenoort, J.-N. Dumez, and I. Kuprov, “Efficient simulation of ultrafast magnetic resonance experiments,” Phys. Chem. Chem. Phys. 19, 17577–17586 (2017).
13. C. Weninger and N. Rühringer, “Stimulated resonant x-ray Raman scattering with incoherent radiation,” Phys. Rev. A 88, 053420 (2013).
14. T. Mathisen and J. Larson, “A view on the open STIRAP problem,” https://arxiv.org/abs/1609.09673 (2016).
15. F. T. Hioe and J. H. Eberly, “N-level coherence vector and higher conservation laws in quantum optics and quantum mechanics,” Phys. Rev. Lett. 47, 838–841 (1981).
16. A. Taflove and S. C. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method (Artech House, 2005).
17. G. Strang, “On the construction and comparison of difference schemes,” SIAM J. Numer. Anal. 5, 506–517 (1968).
1. Introduction

The Maxwell-Liouville-von Neumann (MLN) equations describe the interaction of electromagnetic fields with quantum mechanical systems and are an important tool in nonlinear optics, for example to model the dynamics of quantum cascade lasers (QCLs) [1] or quantum well structures [2]. Usually, numerical methods are required to solve the MLN equations. The rotating wave approximation (RWA) is commonly used in order to save computation time but is avoided in this work since it omits certain features of the solution.

Several numerical methods to solve the MLN equations without invoking the RWA have been published in the last decades, starting with the pioneering work by Ziolkowski et al. [3]. This work uses the finite-difference time-domain (FDTD) method for Maxwell’s equations as well as the Crank-Nicholson scheme and a predictor-corrector (PC) method for the Liouville-von Neumann equations. In the latter, only two energy levels are considered (in this case, the equations are commonly called Maxwell-Bloch equations).

In the work by Slavcheva et al. [4,5], this approach was extended to cope with more than two energy levels. The methods in [3–5] use the adjoint representation of the Lie algebra SU(N) for N energy levels (also called pseudospin representation) and hence eliminate redundant computations and memory requirements. Quantum mechanical operators such as the density matrix are Hermitian, therefore it is sufficient to calculate and store only the half of the off-diagonal elements. Additionally, the trace of the density matrix must equal 1, so the diagonal elements can be expressed using only N − 1 real quantities. Regarding the numerical method, the predictor-corrector method is also computationally efficient.

However, Bidégaray et al. [6] proved that this method may produce unrealistic results when applied to multi-level systems. As an alternative that provides long-term stability in the simulations, an operator splitting approach was suggested [7, 8]. The major drawback of the operator splitting technique is the required computational effort to calculate the matrix exponentials at every time step. Different ways to perform this calculation efficiently can be found in literature: The approximation used in [6–8], employing the Expokit [9] software package and replacing the FDTD with the pseudo-spectral time-domain (PSTD) method [10], the Magnus expansion via Sylvester’s formula [11], the scaling and squaring method as well as a Krylov subspace method [12], and diagonalization of the matrix [13].

Although several research groups used the adjoint representation [3–5,11,14] or the operator splitting technique [6–8, 10, 13], there is – to the best of our knowledge – no approach that combines both concepts. In the work at hand we present such a combination and describe two optimization possibilities for the matrix exponential calculations.

In the following, we describe the Maxwell-Liouville-von Neumann equations (Section 2) and the Liouville-von Neumann equation in the adjoint representation (Section 3). In Section 4 we focus on the numerical treatment of the Liouville-von Neumann equation. The resulting method can be used in combination with both FDTD and PSTD, therefore the treatment of Maxwell’s equations shall receive less attention in the scope of this work. The verification of the presented
methods and their computational performance are discussed in Section 5 using two test cases. Finally, we summarize the performance improvements of the presented methods and give an outlook on future work.

2. The Maxwell-Liouville-von Neumann Equations

In the work at hand, we consider one-dimensional Maxwell’s equations for the electric and magnetic field components \( E_z (x, t) \) and \( H_y (x, t) \).

\[
\begin{align*}
\partial_t E_z &= \epsilon^{-1} ( -\sigma E_z - \partial_x P_z + \partial_z H_y ), \\
\partial_t H_y &= \mu^{-1} \partial_x E_z,
\end{align*}
\]

where \( x \) is the propagation direction, \( y \) and \( z \) denote the transversal coordinates, and \( t \) is time. In Maxwell’s equations the conductivity \( \sigma \), permittivity \( \epsilon \), and permeability \( \mu \) of the active region material as well as the polarization term \( P_z (x, t) \) are taken into account.

The second part of the MLN equations is the Liouville-von Neumann equation. It is used to determine the behavior of the quantum mechanical multilevel systems along the propagation direction (described by the density operator \( \hat{\rho} (x, t) \), which can be written as \( N \times N \) matrix for \( N \) discrete energy levels). The equation reads

\[
\partial_t \hat{\rho} = \mathcal{L} (\hat{\rho}) + \mathcal{G} (\hat{\rho}) = -i\hbar^{-1} \left[ \hat{H}, \hat{\rho} \right] + \mathcal{G} (\hat{\rho}),
\]

where the right hand side consists of the relaxation superoperator \( \mathcal{G} (\hat{\rho}) \) (that may include e.g. scattering processes) and the Liouvillian \( \mathcal{L} (\hat{\rho}) \). Here, \( \hbar \) denotes the reduced Planck constant. The Hamiltonian \( \hat{H} (x, t) = \hat{H}_0 + \hat{H}_1 (x, t) \) consists of a time-independent part \( \hat{H}_0 \) and a time-dependent interaction part \( \hat{H}_1 (x, t) = - \hat{\mu} E_z (x, t) \), where \( \hat{\mu} \) is the dipole moment operator. It is practical to define a corresponding pair of Liouvillian superoperators \( \mathcal{L}_0 (\hat{\rho}) \) and \( \mathcal{L}_1 (\hat{\rho}) \). We assume that only the electric field component \( E_z (x, t) \) is relevant for the interaction process.

Finally, the polarization term \( P_z (x, t) \) in Eq. (1) (or its derivative, respectively) is calculated as

\[
\partial_t P_z = N_a \text{Tr} \{ \hat{\mu} \partial_x \hat{\rho} \} = N_a \text{Tr} \{ \hat{\mu} \mathcal{L} (\hat{\rho}) + \hat{\rho} \mathcal{G} (\hat{\rho}) \},
\]

where \( N_a \) is the density of quantum mechanical particles in the system. Since

\[
\text{Tr} \{ \hat{\rho} [\hat{\mu}, \hat{\phi}] \} = \text{Tr} \{ \hat{\mu} \hat{\phi} \hat{\rho} \} - \text{Tr} \{ \hat{\mu} \hat{\rho} \hat{\phi} \} = \text{Tr} \{ \hat{\phi} \hat{\rho} \hat{\mu} \} - \text{Tr} \{ \hat{\rho} \hat{\mu} \hat{\phi} \} = 0,
\]

one can write Eq. (3) as

\[
\partial_t P_z = N_a \text{Tr} \{ \hat{\mu} \mathcal{L}_0 (\hat{\rho}) + \hat{\rho} \mathcal{G} (\hat{\rho}) \} = N_a \text{Tr} \{ \mathcal{L}_0 (\hat{\rho}) \hat{\mu} + \mathcal{G} (\hat{\rho}) \hat{\mu} \},
\]

where \( \mathcal{L}_0 (\hat{\rho}) \) represents the time-independent part of the Liouvilian.

3. The Liouville-von Neumann Equation in Adjoint Representation

The adjoint representation (also called pseudospin representation) is one of the most efficient ways to describe the density matrix, since only the non-redundant \( N^2 - 1 \) real elements are considered [15]. It is practical to write them as a vector \( \tilde{\rho} \). The density matrix \( \hat{\rho} \) can be composed as

\[
\hat{\rho} = N^{-1} \mathbb{I} + \frac{1}{2} \sum_{j=1}^{N^2 - 1} d_j \hat{s}_j,
\]

where \( \mathbb{I} \) is the \( N \times N \) identity matrix, \( \hat{s}_j \) are generators of the Lie algebra of SU\((N)\), and the vector elements are defined as \( d_j := \text{Tr} \{ \hat{\rho} \hat{s}_j \} \). The generators are traceless Hermitian \( N \times N \) matrices.
and fulfill the condition \( \text{Tr} \{ \delta_j \delta_k \} = 2 \delta_{jk} \), where \( \delta_{jk} \) denotes the Kronecker delta. One possible choice \[15\] for the generators \( \delta = \{ \hat{u}_{12}, \ldots, \hat{v}_{12}, \ldots, \hat{w}_N \} \) consists of \( N(N-1)/2 \) generator pairs \( \hat{u}_{jk} := \hat{u}_{jk} + \hat{v}_{kj} \), \( \hat{v}_{jk} := -i \hat{u}_{jk} - \hat{v}_{kj} \), and \( N-1 \) generators

\[
\hat{w}_l := -\frac{2}{I(I+1)} (\hat{t}_{11} + \hat{t}_{22} + \cdots + \hat{t}_{ll} - I_l \hat{t}_{l+1,l+1}),
\]

(7)

where the indices satisfy \( 1 \leq j < k \leq N \) and \( 1 \leq I \leq N-1 \) and the transition-projection operators are defined as \( \hat{t}_{jk} := |j \rangle \langle k| \). For \( N = 2 \) and \( N = 3 \) these generators produce the Pauli matrices and the Gell-Mann matrices, respectively.

In order to transform the Liouville-von Neumann equation to the adjoint representation, Eq. (6) is inserted into Eq. (2). Subsequently, the Frobenius inner product\[1\] \( \langle \cdot, \cdot \rangle_F \) is applied to the result. The left hand side of Eq. (2) then reads

\[
\text{Tr} \{ \partial_t \hat{\rho} \hat{\delta}_k \} = \text{Tr} \left\{ \frac{1}{2} \sum_{j=1}^{N^2-1} \partial_t d_j \delta_j \right\} = \frac{1}{2} \sum_{j=1}^{N^2-1} \partial_t d_j \text{Tr} \{ \delta_j \delta_k \} = \partial_t d_k.
\]

(8)

Since both superoperators are linear, we can write

\[
\text{Tr} \{ \mathcal{L} (\hat{\rho}) \hat{\delta}_k + \mathcal{G} (\hat{\rho}) \hat{\delta}_k \} = \text{Tr} \{ \mathcal{L} (\hat{\rho}) \hat{\delta}_k \} + \text{Tr} \{ \mathcal{G} (\hat{\rho}) \hat{\delta}_k \},
\]

(9a)

\[
\text{Tr} \{ \mathcal{L} (\hat{\rho}) \hat{\delta}_k \} = \text{Tr} \left\{ N^{-1} \mathcal{L} \left( \hat{\mu} \right) \hat{\delta}_k \right\} + \sum_{j=1}^{N^2-1} \frac{1}{2} \text{Tr} \{ \mathcal{L} \{ \hat{\delta}_j \} \hat{\delta}_k \} d_j,
\]

(9b)

\[
\text{Tr} \{ \mathcal{G} (\hat{\rho}) \hat{\delta}_k \} = \text{Tr} \left\{ N^{-1} \mathcal{G} \left( \hat{\mu} \right) \hat{\delta}_k \right\} + \sum_{j=1}^{N^2-1} \frac{1}{2} \text{Tr} \{ \mathcal{G} \{ \hat{\delta}_j \} \hat{\delta}_k \} d_j
\]

(9c)

for the right hand side. As a result, the Liouville-von Neumann equation can be transformed to

\[
\partial_t \hat{d} = (\mathcal{L} + \mathcal{G}) \hat{d} + \hat{d}^\text{eq},
\]

(10)

where the elements of the \( N \times N \) matrices \( \mathcal{L} \) and \( \mathcal{G} \) are

\[
l_{jk} := \frac{1}{2} \text{Tr} \{ \mathcal{L} \{ \hat{\delta}_k \} \hat{\delta}_j \} = -\frac{1}{2\hbar} \text{Tr} \{ [\hat{\mu}, \hat{\delta}_k] \hat{\delta}_j \} = \frac{1}{2\hbar} \text{Tr} \{ \hat{\mu} [\hat{\delta}_j, \hat{\delta}_k] \},
\]

(11a)

\[
g_{jk} := \frac{1}{2} \text{Tr} \{ \mathcal{G} \{ \hat{\delta}_k \} \hat{\delta}_j \},
\]

(11b)

respectively, and the equilibrium vector \( \hat{d}^\text{eq} \) has the elements

\[
d_j^{\text{eq}} := N^{-1} \text{Tr} \{ \mathcal{G} \{ \hat{\mu} \} \hat{\delta}_j \}.
\]

(12)

With the separation of the Liouvillean in mind, we can split the matrix \( \mathcal{L} + \mathcal{G} \) into a time-independent matrix \( \mathcal{M} \) and a time-dependent part \( \mathcal{U}_E \), whose elements are

\[
m_{jk} := \frac{i}{2\hbar} \text{Tr} \{ \hat{\mu}_0 [\hat{\delta}_j, \hat{\delta}_k] \} + g_{jk},
\]

(13a)

\[
u_{jk} := \frac{i}{2\hbar} \text{Tr} \{ -\hat{\mu} [\hat{\delta}_j, \hat{\delta}_k] \} = -\frac{i}{2\hbar} \text{Tr} \{ \hat{\mu} [\hat{\delta}_j, \hat{\delta}_k] \},
\]

(13b)

\[\text{The Frobenius inner product is defined as } \langle \hat{a}, \hat{b} \rangle_F = \text{Tr} \{ \hat{a} \hat{b}^\dagger \}. \text{ Note that } \langle \cdot, \cdot \rangle_F = \text{Tr} \{ \cdot \}, \text{ since the generators are Hermitian.}\]
respectively, and write Eq. (10) as
\[ \partial_t \vec{d} = (\mathcal{M} + U \mathcal{E}_z) \vec{d} + \vec{d}^{eq}, \]  
which is the Liouville-von Neumann equation in the adjoint representation.

Analogously, the derivative of the polarization must be expressed as function of the vector \( \vec{d} \).
In order to do so, we transform the dipole moment operator to a vector
\[ \hat{\mu} = \frac{1}{2} \sum_{k=1}^{N^2-1} v_k \hat{s}_k := \frac{1}{2} \sum_{k=1}^{N^2-1} \text{Tr} \{ \hat{\mu} \hat{s}_k \} \hat{s}_k \]  
using the generators as basis\(^2\) and insert Eqs. (6) and (15) into Eq. (5). The argument of the trace function in Eq. (5) then reads
\[ [\mathcal{L}_0 (\hat{\rho}) + \mathcal{G} (\hat{\rho})] \hat{\mu} = \left[ \frac{1}{2} \sum_{j=1}^{N^2-1} d_j \mathcal{L}_0 (\hat{s}_j) + N^{-1} \mathcal{G} (\hat{I}) \right] \hat{s}_k + \frac{1}{2} \sum_{j=1}^{N^2-1} d_j \hat{s}_k \]  
and can be simplified using the definitions in Eqs. (11) and (13) once the trace function is applied. Then, the derivative of the polarization is calculated as
\[ \partial_t P_z = N_a \text{Tr} \{ [\mathcal{L}_0 (\hat{\rho}) + \mathcal{G} (\hat{\rho})] \hat{\mu} \} = N_a \frac{1}{2} \sum_{k=1}^{N^2-1} v_k \left( \sum_{j=1}^{N^2-1} m_{kj} d_j + d_k^{eq} \right) = \frac{1}{2} N_a \vec{v}^T \left( \mathcal{M} \vec{d} + \vec{d}^{eq} \right). \]  

4. Numerical Treatment

The achievements by Bidégaray et al. [6, 7] serve as basis for the numerical treatment of the equations described in the previous sections. The method in [7] uses the finite-difference time-domain (FDTD) method [16] to solve Maxwell’s equations and an operator splitting approach for the Liouville-von Neumann equation. Figure 1 provides a schematic overview. As already mentioned, we focus on the efficient numerical treatment of the Liouville-von Neumann equation in this study. Our efforts can be reused in an approach that uses a different method for Maxwell’s equations, e.g. the pseudo-spectral time-domain (PSTD) method used in [10]. Since the adjoint representation was not used in [6, 7], we revisit the derivation of the operator splitting approach in the following. Subsequently, we describe two methods to compute the required matrix exponentials efficiently.

4.1. Operator Splitting Approach

The right hand side in Eq. (14) can be split up in two operators \( L_1 \) (time-independent) and \( L_2 \) (time-dependent), yielding
\[ \partial_t \vec{d} = L_1 (\vec{d}) + L_2 (\vec{d}). \]  
\(^2\)This is possible since the generators span all traceless Hermitian \( N \times N \) matrices. Due to the invariance of the Liouvillian superoperator, the main diagonal of every Hamiltonian (and consequently of the dipole moment operator) can be shifted so that the matrix becomes traceless.
Subsequently, the differential equation is solved individually for each operator. For operator $L_1$, the solution is

$$
\partial_t \tilde{d} = L_1 \left( \tilde{d} \right) = M \tilde{d} + \tilde{d}^\text{eq} \rightarrow \tilde{d} = \exp \left[ M \left( t - t_0 \right) \right] \left[ \tilde{d} \left( t_0 \right) + M^{-1} \tilde{d}^\text{eq} \right] - M^{-1} \tilde{d}^\text{eq},
$$

(19)

where $\tilde{d}^\text{in} := M^{-1} \tilde{d}^\text{eq}$ is the inhomogeneous part of the solution. The solution for operator $L_2$ can be determined as

$$
\partial_t \tilde{d} = L_2 \left( \tilde{d} \right) = U E_z \tilde{d} \rightarrow \tilde{d} = \exp \left[ U \int_{t_0}^{t} E_z \left( \tau \right) d\tau \right] \tilde{d} \left( t_0 \right).
$$

(20)

The complete differential equation can be solved using the symmetric Strang splitting [17], i.e., by updating $\tilde{d}$ first with the solution of operator $L_1$ for half a time step

$$
\tilde{d} \left( t_0 + \Delta t / 2 \right) = \exp \left( M \Delta t / 2 \right) \left[ \tilde{d} \left( t_0 \right) + \tilde{d}^\text{in} \right] - \tilde{d}^\text{in},
$$

(21)

then with the solution of operator $L_2$ for a full time step

$$
\tilde{d} \left( t_0 + \Delta t \right) = \exp \left[ U \int_{t_0}^{t_0 + \Delta t} E_z \left( \tau \right) d\tau \right] \tilde{d} \left( t_0 \right) \approx \exp \left[ U E_z \left( t_0 + \Delta t / 2 \right) \Delta t \right] \tilde{d} \left( t_0 \right),
$$

(22)

where the midpoint rule is applied as second order approximation for the integral, and finally with the solution of operator $L_1$ for half a time step again. The resulting operator splitting approach features second order accuracy as well and has the update rules

$$
\tilde{d}' \leftarrow \exp \left( M \Delta t / 2 \right) \left( \tilde{d}^{n-1/2} + \tilde{d}^n \right) - \tilde{d}^n,
$$

(23a)

$$
\tilde{d}'' \leftarrow \exp \left( U E_z^N \Delta t \right) \tilde{d}',
$$

(23b)

$$
\tilde{d}^{n+1/2} \leftarrow \exp \left( M \Delta t / 2 \right) \left( \tilde{d}'' + \tilde{d}^n \right) - \tilde{d}^n,
$$

(23c)

$^3$Note that the Magnus expansion is not required since $U$ is time-independent.
where $\hat{d}^n = \hat{d}((n\Delta t) \text{ and } E_z^n = E_z(n\Delta t)$ represents the discretization at a given time step $n$.

In related work [7, 8, 18] the calculation of the matrix exponentials has been identified as the computational bottleneck. While the expression $A_\theta := \exp(M\Delta t/2)$ is constant and can be calculated once (e.g. using the Padé approximation), the interaction term $A_\theta^n := \exp (UE_z^n\Delta t)$ must be updated every time step. Therefore, the latter calculation should be optimized first.

### 4.2. Efficient Computation of Matrix Exponentials

By inspection of the elements $u_{jk}$ we can determine two properties of the matrix $U$. The commutator $\hat{C} := [\hat{\sigma}_j, \hat{\sigma}_k]$ in Eq. (13b) is antisymmetric, as a result the same holds for the elements $u_{jk}$. Furthermore, $\hat{C}$ is skew-Hermitian (which holds for every commutator of two Hermitian matrices). Then, the trace

$$
\text{Tr} \{ \hat{\mu} \hat{C} \} = \sum_{i=1}^{N} \mu_{ii}e^{i\hat{\mu}t} + \sum_{1 \leq j < k \leq N} \left( \mu_{jk}^* e^{i\hat{\mu}t} - \mu_{jk} e^{i\hat{\mu}t} \right)
$$

with $\mu_{jk} = \langle j | \hat{\mu} | k \rangle$ and $c_{jk} = \langle j | \hat{C} | k \rangle$ is purely imaginary, since $\mu_{ii}$ and $c_{ii}$ are real and purely imaginary, respectively, and $\mu_{jk}^* e^{i\hat{\mu}t} - \mu_{jk} e^{i\hat{\mu}t} = 2i \left[ \text{Re} \{ \mu_{jk} \} \text{ Im} \{ c_{jk} \} - \text{Im} \{ \mu_{jk} \} \text{ Re} \{ c_{jk} \} \right]$ is purely imaginary. As a result, the elements $u_{jk}$ in Eq. (13b) are always real.

This fact leads to two optimization possibilities. First, a real antisymmetric matrix $U = RR^\dagger$ can be diagonalized so that $A$ is a diagonal matrix containing the (purely imaginary and pairwise complex conjugated) eigenvalues $\lambda_i$ and $R$ is a unitary matrix consisting of the eigenvectors. The interaction term now reads

$$
A_\theta^n = R \exp (\Delta E_z^n\Delta t) R^\dagger = R \begin{bmatrix} \exp (\lambda_1 E_z^n\Delta t) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \exp (\lambda_N E_z^n\Delta t) \end{bmatrix} R^\dagger
$$

and the cost of calculating the matrix exponential is reduced to two (complex) matrix multiplications and $N^2 - 1$ calls to the scalar exponential function. The matrix $R$ as well as the eigenvalues remain constant and can be precalculated. This is one of the most accurate and efficient techniques for normal matrices [19] (real antisymmetric matrices are always normal).

Second, this special case of matrix exponential allows analytic solutions. For two-level systems the exponential of $3 \times 3$ matrices has to be evaluated, which can be accomplished using Rodrigues’ formula. This formula was generalized to arbitrary matrix dimensions [20]. As prerequisites, one has to determine the eigenvalues $\lambda_{2i-1,2i} = \pm i\theta_1$ of $U$ and set up a matrix $\Sigma = T\Theta T^T$ for each pair of eigenvalues. The matrix

$$
T = \frac{1}{\sqrt{2}} \begin{bmatrix} (\bar{r}_1 + \bar{r}_2) & -i(\bar{r}_1 - \bar{r}_2) & (\bar{r}_3 + \bar{r}_4) & -i(\bar{r}_3 - \bar{r}_4) \end{bmatrix}
$$

is constructed using the eigenvectors $\bar{r}_k$ of $U$ that correspond to the eigenvalues $\lambda_k$; the matrix $\Theta$ is zero apart from the entries

$$
\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}
$$

in the $(2i-1)$-th and $(2i)$-th row and column, respectively. Then, the matrix exponential can be calculated as

$$
A_\theta^n = I + \sum_{i=1}^{q} \sin (\theta_i E_z^n\Delta t) \Sigma_i + \left[ 1 - \cos (\theta_i E_z^n\Delta t) \right] \Sigma_i^2,
$$

where $I$ is the identity matrix and $q$ is the number of eigenvalue pairs.

---

4Note that $U$ has an extra eigenvalue $\lambda_{N^{2-1}} = 0$ besides the purely imaginary pairs in the odd-dimensional case.

5In the odd-dimensional case, the matrices $\Theta_i$ are padded with zeros.
5. Verification and Performance Evaluation

We implemented the diagonalization technique as well as the method using the generalized Rodrigues formula based on the mbsolve project [21]. This project has recently served as base for our operator splitting approach using the Eigen library [22] and the Padé approximation for the matrix exponential calculations [18]. The Padé approximation approach serves as reference in terms of verification as well as performance in two test cases, which we describe in the following. Finally, we discuss the performance of each technique.

5.1. Two-level system (Maxwell-Bloch case)

The first test case is the self-induced transparency (SIT) simulation by Ziolkowski et al. [3]. In this simulation, a two-level system is described with the Hamiltonian

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

(28)

where \( \omega_{12} = 4\pi \times 10^{14} \text{ s}^{-1} \) is the transition frequency and \( \mu_{12} = 10^{-29} \text{ A s m} \) is the dipole moment between the two levels, and the relaxation superoperator

\[ \mathcal{G}(\hat{\rho}) = \begin{bmatrix} T_{11}^{-1} \rho_{22} & -T_{12}^{-1} \rho_{12} \\ -T_{21}^{-1} \rho_{21} & -T_{22}^{-1} \rho_{22} \end{bmatrix}, \]

(29)
where $T_1 = 10^{-10} \, \text{s}^{-1}$ and $T_2 = 10^{-10} \, \text{s}^{-1}$ are decay terms (upper level lifetime and dephasing time, respectively). These components are transformed to adjoint representation as described in Section 3. Additionally, a initial density matrix

$$\hat{\rho}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

(30)

can be transformed to an initial condition $\tilde{\rho}_0$.

The simulation was set up with 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 $7.635 \times 10^{-18} \, \text{s}$. Then, the simulation was executed and the populations of both levels were recorded. A snapshot after 187.5 fs is depicted in Fig. 2. It shows perfect agreement between the results of all methods and the data depicted in [3].

### 5.2. Three-level system

The second test case incorporates the three-level setup presented by Song et al. [23]. The Hamiltonian is

$$\hat{H} = \hat{H}_0 + \hat{H}_1 = \hbar \begin{bmatrix} 0 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} - \begin{bmatrix} 0 & \mu_{12} & \gamma \mu_{12} \\ \mu_{12} & 0 & 0 \\ \gamma \mu_{12} & 0 & 0 \end{bmatrix} E_z,$$

(31)

Note that a different sign convention is used for $\mu_{12}$ in [3]. However, the convention has no effect on the result if applied consistently.
where $\omega_2 = 2.372 \text{s}^{-1}$ and $\omega_3 = 2.417 \text{s}^{-1}$ are the eigenfrequencies, $\mu_{12} = 1.48 \times 10^{-29} \text{A s m}$ is the dipole moment between the levels 1 and 2, and $\gamma = \sqrt{2}$ is the ratio between the dipole moments. Furthermore, we define the relaxation superoperator as

$$
\mathcal{G}(\hat{\rho}) = T_1^{-1} \begin{bmatrix}
\frac{1}{g_1} \rho_{11} - \rho_{12} & \frac{1}{g_1} (\rho_{22} - \rho_{11}) & -\rho_{13} \\
-\rho_{21} & \frac{1}{g_1} (\rho_{33} - \rho_{11}) & -\rho_{23} \\
-\rho_{31} & -\rho_{32} & \frac{1}{g_1} (\rho_{33} - \rho_{11})
\end{bmatrix},
$$

(32)

where $g_1 = \rho_{33} + \rho_{22} - 2\rho_{11}$, and $T_1 = 10^{-10} \text{s}$ is the relaxation time. As initial condition, the density matrix

$$
\hat{\rho}_0 = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
$$

(33)

was transformed to the adjoint representation.

Using the choice of grid point size and time step size from the two-level test case, the results from [23] could be reproduced. Figure 3 shows the agreement of the different methods for this test case.

5.3. Performance

The simulations were executed on an Intel Xeon Processor E7-4870. Although the code is designed to run efficiently in parallel, only one thread was used since we wanted to measure the quality of a single-thread optimization. In order to ensure reproducibility, the measurements were repeated five times. The performance was determined as number of grid point updates per time unit, i.e. $P = N_s N_t / t_{\text{exec}}$, where $N_s$ and $N_t$ are the number of spatial and temporal grid points, respectively, and $t_{\text{exec}}$ is the measured execution time. This allows the metric to be used to compare the performance of problems with different sizes.

It should be noted that the execution time does not contain the time required to construct or delete data structures (which is negligible for the simulation setups in question). It does, however,
contain the time required to store the result data, in particular the time to convert the vector \( \vec{J} \) to the density matrix \( \hat{\rho} \).

The performance values of the different methods applied to the two test cases are shown in Fig. 4. Compared to our baseline (the Padé approximation method), the diagonalization approach performs 1.26x better for the two-level test case and 1.55x better for the three-level test case. Using the generalized Rodrigues formula, we were able to achieve a speedup of 3.48x and 7.35x, respectively.

6. Conclusion

Our method solves the Maxwell-Liouville-von Neumann equations in an efficient and accurate way. It is a novel combination of two established concepts. The adjoint representation is most efficient in terms of eliminating the redundancy inherent to quantum mechanical operators and additionally leads to interesting properties of the resulting description. These properties are exploited in order to provide two efficient implementations of the operator splitting technique, which is accurate and stable, yet computationally expensive.

Both implementations are tested with the help of a two-level and a three-level test case and can reproduce the results found in related literature. Regarding the performance, the diagonalization approach yields a small improvement and the method using the generalized Rodrigues formula provides substantial speedup, at least at test cases with small level count.

We expect that both methods provide even larger performance improvements compared to standard methods (e.g. the Padé approximation) in test cases with many energy levels. Also, the performance ratio between the diagonalization approach and the Rodrigues method may differ. However, the behavior of the methods applied to a many-level setup has to be investigated first.

Finally, the method is able to cope with a broad range of problems. However, there are two limitations, namely the restriction to one-dimensional models as well as the strict definition of the interaction Hamiltonian. In future work we will aim to generalize our method in order to overcome those limitations.

Supplementary Material

See [21] for source code, build instructions, and basic documentation. In this paper we used the development branch riesch2017c, the Eigen library version 3.3.4, and the Intel C++ compiler 17.0.

Funding

German Research Foundation (DFG) (JI 115/4-1, JI 115/9-1).

Acknowledgments

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). We thank Gabriela Slavcheva for the interesting discussion and her valuable input on the adjoint representation.