Strang splitting schemes for N-level Bloch models

Marc E. Songolo[a]**and Brigitte Bidégaray-Fesquet[b]

[a]Department of Mathematics and Computer Science,
University of Lubumbashi, Lubumbashi, Republic Democratic of Congo;
ICAM School of Engineering-Nantes campus, 44470 Carquefou, France;
IETR, Polytech Nantes, University of Nantes, 44306 Nantes, France;
[b]Univ. Grenoble Alpes, Grenoble INP††, LJK, 38000 Grenoble, France

April 5, 2023

Abstract. We define a splitting scheme for the N-level Bloch model which makes use of exact numerical solutions of sub-equations. These exact solutions involve matrix exponentials which we want to avoid to calculate at each time-step. The resulting scheme is nonstandard and preserves qualitative properties of the Bloch equations. We explore and compare numerically multiple ways to implement it and in particular take into account the specific structure of the Bloch equations.

Keywords: Bloch equation, Matrix exponential, Exact finite difference schemes, Nonstandard finite difference schemes, Splitting method.

1 Introduction

Available laser sources allow studying radiation–matter interactions [8] because the intensities and the pulse durations achievable by lasers make it possible to reach the order of magnitude of the cohesion energy of electrons

**Corresponding author. Email: marc.songolo@gmail.com
††Institute of Engineering Univ. Grenoble Alpes

in atoms. Some phenomena such as spontaneous or stimulated Raman scattering, Brillouin or Rayleigh, laser effect, two-photon absorption, etc. require at least a semi-classical model where the field is modeled classically and is coupled with a quantum medium. Quantum models are more precise than classical optics models and allow to obtain information on the structure of the atoms, thus highlighting the whole phenomenon [4].

In this context, Maxwell–Bloch equations can be used. The electromagnetic field is then modeled by Maxwell equations and matter is described at the quantum mechanical level by the Bloch equations. The coupling between these two systems is done by the expression of the polarization. Bloch equations describe the evolution of the density matrix. They are derived from the Schrödinger equation or in the Heisenberg formalism. The density matrix is a quantum observable (unlike the wave function) and allows to describe the probability of the presence of electrons in the quantified energy levels (diagonal elements of the matrix) and the coherences between these levels (off–diagonal elements). Its size depends on the number of levels. In many references, the derivation of the Bloch equation is only presented in the case of two-level atoms.

Bloch equations can already raise problems both from a theoretical and a numerical point of view. They have been approached by various numerical methods such as the Crank–Nicolson method [1, 3, 12, 18, 26, 27, 28], a fourth-order Runge–Kutta method [23], the relaxation method [9], etc. But most of these numerical schemes do not conserve the qualitative properties of Bloch equations. To overcome this deficiency, Bidégaray *et al.* [5] have introduced a Strang splitting method which preserves some physical properties (Hermicity, positivity, trace conservation) of the continuous model.

Following this study, we have discussed in depth Strang splitting schemes for the two-level Bloch model in [20]. A special feature of these schemes is that the solutions of sub-equations are exact, of variable time-step size for the Liouville equation and conform to the Nonstandard Finite Difference (NSFD) methodology developed by Mickens [14]. These splitting schemes preserve physical properties of the Bloch equations. Moreover, they are explicit and retain the advantage of stability when coupled with Maxwell equations. In this paper, we want to extend this type of schemes to the N-level Bloch model, in order to improve the scheme presented in [5].

The paper is organized as follows: we introduce the Bloch model in Section 2. Section 3 provides the construction rules of NSFD schemes. In Section 4, we first introduce the decomposition of the Bloch equation, then

the exact schemes for the sub-equations, and finally the splitting scheme. In order to reduce the algorithmic complexity of the splitting scheme, we propose in Section 5 an equivalent formulation of the matrix exponentials that occur in the exact discretization of the Liouville equation. In Section 6, we compare numerically the Crank–Nicolson method and the reformulations of the exponential discussed in this paper. The study is followed by an appendix containing an alternative method for the derivation of the matrix exponential.

2 Bloch model

The derivation of the Bloch equation can be found in many textbooks (see for example [4, 6, 7, 13, 24]). In this study, we use dimensionless equations:

$$\partial_t \rho = -i[H_0, \rho] - i[V, \rho] + Q(\rho), \tag{1}$$

where $[A, B] = AB - BA$ is the commutator of the two operators A and B. The diagonal entries of the *density matrix* ρ are called *populations* and express the presence probabilities of electrons in the quantified energy levels. The off-diagonal entries are complex numbers called *coherences*, whose moduli can be interpreted as conditional probabilities of transition between the energy levels. In equation (1), H_0 is the free Hamiltonian of electrons and is a diagonal matrix $\mathrm{diag}(\omega_j)_{j=1,\dots,n}$. The potential $V(t)$ is a zero diagonal, Hermitian matrix and results from the interaction with an electromagnetic wave. A phenomenological relaxation matrix $Q(\rho)$ can be added to model many phenomena such as spontaneous emission, collisions, vibrations in crystal lattices, etc. It must be chosen so as to preserve over time some properties of the density matrix, in particular Hermicity, positiveness and trace (see [5] for details).

3 NSFD Schemes

The NSFD method was initiated by Mickens to overcome the defects of numerical instabilities presented by classical methods such as Euler and Runge Kutta methods. In fact, NSFD methods have the potential to preserve qualitative properties of the original system of differential equations.

To discuss NSFD schemes, we cast the Bloch equation as

$$\partial_t \rho = F(t, \rho), \tag{2}$$

which unknown is the matrix $\rho : [t_0, T] \to \mathbb{C}^{n\times n}$, initially equal to $\rho_0 \in \mathbb{C}^{n\times n}$, and $F : [t_0, T] \times \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is a given function.

For the numerical approximation of (2), we discretize the interval $[t_0, T]$ at the discrete times $t_n = t_0 + n\Delta t$, where the parameter $\Delta t > 0$ is the step size. We denote by ρ^n an approximation of the solution $\rho(t_n)$ at time t_n.

The finite difference equation reads

$$D_{\Delta t}(\rho^n) = F_{\Delta t}(t_n, \rho^n), \tag{3}$$

where $D_{\Delta t}(\rho^n)$ and $F_{\Delta t}(t_n, \rho^n)$ approximate $\partial_t \rho(t_n)$ and $F(t_n, \rho(t_n))$ respectively.

Definition 1. *The scheme* (3) *is called a nonstandard finite difference method if at least one of the following conditions is satisfied:*

- $D_{\Delta t}(\rho^n) = (\phi(\Delta t))^{-1}(\rho^{n+1} - \rho^n)$ *where* $\phi(\Delta t) = \Delta t I + \mathcal{O}(\Delta t^2)$ *is a positive diagonal matrix;*
- $F_{\Delta t}(t_n, \rho^n) = g(\rho^n, \rho^{n+1}, \Delta t)$ *where* $g(\rho^n, \rho^{n+1}, \Delta t)$ *is a nonlocal approximation of the right-hand side of System* (2)*.*

These notions are discussed in detail in [2, 14, 15, 16]. Moreover, Mickens has introduced in [15] a rule for the construction of NSFD schemes for complex equations.

Rule for complex equations. *For differential equations having* N (≥ 3) *terms, it is generally useful to construct finite difference schemes for various sub-equations composed of* M *terms, where* $M < N$*, and then combine all the schemes together in an overall consistent finite difference model.*

By this last rule, it is necessary to split Equation (1) into two sub-equations, then solve sub-equations by exact methods, and finally, connect solutions of sub-equations through a single consistent solution. To this end, we explore how to construct consistent finite difference models using Strang splitting method.

4 Splitting method

We rewrite Equation (1) as

$$\partial_t \rho_{jk} = -i\omega_{jk}\rho_{jk} - i[V,\rho]_{jk} + Q(\rho)_{jk},$$

where $\omega_{jk} = \omega_j - \omega_k$ is the frequency associated with the transition from level k to level j. This equation is decomposed into the relaxation–nutation evolution

$$\partial_t \rho = L\rho, \tag{4}$$

where $(L\rho)_{jk} = -i\omega_{jk}\rho_{jk} + Q(\rho)_{jk}$ and the interaction with the electromagnetic field

$$\partial_t \rho_{jk} = -i[V,\rho]_{jk}. \tag{5}$$

We have seen in [20] that this splitting yields the best approximation for the Bloch equation, and even is better than no splitting for a Self-Induced Transparency test case.

As the relaxation–nutation operator is linear and time invariant, the solution of Equation (4) is

$$\rho(t) = \exp(L(t-t_0))\rho(t_0). \tag{6}$$

Since the potential V generally depends on time, the solution of the interaction equation (5) is

$$\rho(t) = \exp\left(-i\int_{t_0}^{t} V(\tau)\mathrm{d}\tau\right)\rho(t_0)\exp\left(i\int_{t_0}^{t} V(\tau)\mathrm{d}\tau\right). \tag{7}$$

4.1 Exact discretization of the relaxation–nutation equation

An exact finite difference scheme for Equation (4) is easily deduced from its analytical solution (6) and one time-step of the relaxation–nutation equation reads

$$\rho^{n+1} = e^{L\Delta t}\rho^n. \tag{8}$$

4.2 Exact discretization of the Liouville equation

Let $V^{n+1/2}$ be the mean of the function V on the time interval $[t_n, t_{n+1}]$:

$$V^{n+1/2} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} V(\tau) \mathrm{d}\tau.$$

Then one time-step of the Liouville equation is easily deduced from (7), namely

$$\rho^{n+1} = \exp(-i\Delta t V^{n+1/2}) \rho^n \exp(i\Delta t V^{n+1/2}). \tag{9}$$

4.3 Strang splitting

To construct the splitting scheme, we choose the Strang formula [22] in order to achieve second order precision, which would prove useful when coupling with an order-two scheme for the electromagnetic field in a Maxwell–Bloch context (see [4]).

Furthermore, this method is consistent (see [11], for details) according to the discretization rule for complex equations, and preserves the symmetry and positiveness properties of the density matrix [4], provided relaxation terms satisfy the conditions given in [5]. Thus, the Strang splitting method for the Bloch model reads

$$\rho^{n+1} = \exp(L\Delta t/2) \exp(-i\Delta t V^{n+1/2}) \exp(L\Delta t/2) \rho^n \exp(i\Delta t V^{n+1/2}). \tag{10}$$

Here, we begin and end up with the relaxation–nutation term in the splitting. This is the good choice, since this term is the steepest one when the electromagnetic field is small, which necessarily happens in the test cases (oscillating field, wave paquet). This appropriate treatment of stiffness is the reason why splitting behaves better than no splitting. In the sequel, we give equivalent formulations to $\exp(i\Delta t V^{n+1/2})$, to avoid the possible complexity of the calculation of matrix exponentials at each time-step.

5 Exponential of $N \times N$ matrices

Dozens of methods for calculating the exponential of a matrix can be obtained from more or less classical results in analysis, approximation theory, and matrix theory. In [17], the authors describe nineteen methods that seem to be practical. The relative effectiveness of each method is evaluated according to

the following attributes, listed in decreasing order of importance: generality, reliability, stability, accuracy, efficiency, storage requirements, ease of use and simplicity. In particular, generality means that the process is applicable to large classes of matrices. For example, a method that only works on matrices with distinct eigenvalues will not be much appreciated. By these standards, none of the algorithms we know is satisfactory, although some are much better than others.

5.1 Interpretation as an interpolation problem

The Cayley–Hamilton theorem applied to a matrix $A \in \mathcal{M}_N(\mathbb{C})$ ensures that $p(A) = 0$ where p is the characteristic polynomial of A defined by $p(\lambda) = \det(\lambda I - A)$. This allows to express A^N and higher powers as $N-1$-degree polynomials of A. More generally it allows to express an analytical function of A as such a polynomial. This is the case for the exponential of A.

Let $\gamma \in \mathbb{C}$, the function $\exp(i\gamma A)$ can also be expressed by a polynomial that we denote $\mathcal{P}_\gamma(A)$:

$$\mathcal{P}_\gamma(A) = \sum_{j=0}^{N-1} \alpha_j(\gamma) A^j. \tag{11}$$

If the eigenvalues $\lambda_1, \ldots, \lambda_N$ are distinct, then there is a basis in which A is the diagonal matrix D. Let P be the change of basis matrix, which does not depend on γ, we have

$$\exp(i\gamma D) = P^{-1} \exp(i\gamma A) P = P^{-1} \mathcal{P}_\gamma(A) P = \mathcal{P}_\gamma(D).$$

This relation involving diagonal matrices is actually a system of N independent equations

$$\exp(i\gamma\lambda_k) = \mathcal{P}_\gamma(\lambda_k), \;\; k = 1, \ldots, N,$$

which admits a unique solution, because it amounts to inverting a Vandermonde matrix. We however do not want to invert it, but interpret this system as an interpolation problem, i.e. interpolate function $\mathcal{P}_\gamma$ at the locations λ_k, $k = 1, \ldots, N$, with values $\exp(i\gamma\lambda_k)$. Since the function $\mathcal{P}_\gamma$ is an $(N-1)$-degree polynomial, the interpolation polynomial at these N locations will be exactly the function itself.

5.2 Newton interpolation

In [20], in the case of 2-level Bloch equations, the interpolation polynomial is expressed in the canonical basis (I, A). This can of course be extended in the case of $N \times N$ matrices in the canonical basis $(I, A, \ldots, A^{N-1})$ but the formulae are quite intricate and have to be derived individually for each value of N. The 3×3 case is treated in an Appendix. To avoid this we use here the Newton basis

$$I, (\gamma A - \lambda_1 I), (\gamma A - \lambda_1 I)(\gamma A - \lambda_2 I), \ldots, (\gamma A - \lambda_1 I) \ldots (\gamma A - \lambda_{N-1} I).$$

In this decomposition the coefficients are divided differences

$$\mathcal{P}_\gamma(A) = f[\lambda_1] + \sum_{\ell=2}^{N} f[\lambda_1, \ldots, \lambda_\ell] \prod_{k=1}^{\ell-1} (\gamma A - \lambda_k I),$$

where $f[\lambda_k] = \exp(i\gamma\lambda_k)$, $k = 1, \ldots, N$ and we have the recursion formula

$$f[\lambda_k, \ldots, \lambda_\ell] = \frac{f[\lambda_k, \ldots, \lambda_{\ell-1}] - f[\lambda_{k+1}, \ldots, \lambda_\ell]}{\lambda_k - \lambda_\ell},\ 1 \leq k < \ell \leq N.$$

The calculation of the polynomial is then done iteratively using the Hörner algorithm. Indeed, we have

$$\begin{aligned} \mathcal{P}_\gamma(A) &= c_0 + \sum_{\ell=1}^{N-1} c_\ell \prod_{k=1}^{\ell} (\gamma A - \lambda_k I) \\ &= c_0 + (\gamma A - \lambda_1)(c_1 + (\gamma A - \lambda_2)(c_2 + \cdots + (\gamma A - \lambda_{N-1}))). \end{aligned} \tag{12}$$

The advantage of this approach, compared to the one based on the canonical basis, is that the numerical code produced is generic for all N.

5.3 NSFD interpretation for the Liouville equation

In Equations (9) and (10) we need to evaluate $\exp(i\Delta t V^{n+1/2})$. The matrix $V^{n+1/2}$ is the product of the scalar electric field $E^{n+1/2}$ and a constant polarisability matrix p (see [4]). We can therefore write

$$\exp(i\Delta t V^{n+1/2}) = \mathcal{P}_{\Delta t E^{n+1/2}}(p).$$

Therefore, the exact scheme for the Liouville equation can be written as

$$\rho^{n+1} = \mathcal{P}^{-1}_{\Delta t E^{n+1/2}}(p)\rho^n \mathcal{P}_{\Delta t E^{n+1/2}}(p), \tag{13}$$

or equivalenty

$$\mathcal{P}_{\Delta t E^{n+1/2}}(p)\rho^{n+1} = \rho^n \mathcal{P}_{\Delta t E^{n+1/2}}(p). \tag{14}$$

The polynomial $\mathcal{P}_{\Delta t E^{n+1/2}}(p)$ is of course also equal to the series expansion of the exponential $\exp(i\Delta t V^{n+1/2})$. Therefore, in the limit $\Delta t \to 0$,

$$\alpha_j(\Delta t E^{n+1/2}) = \frac{(i\Delta t E^{n+1/2})^j}{j!} + O(\Delta t^{j+1}).$$

In particular

$$\alpha_0(\Delta t E^{n+1/2}) = 1 + O(\Delta t) \text{ and } \alpha_1(\Delta t E^{n+1/2}) = i\Delta t E^{n+1/2} + O(\Delta t^2).$$

Let us set $\alpha_1(\Delta t E^{n+1/2}) = iE^{n+1/2}\tilde{\alpha}_1(\Delta t E^{n+1/2})$. For a small enough Δt, we can ensure that $\alpha_0(\Delta t E^{n+1/2})$ and $\tilde{\alpha}_1(\Delta t E^{n+1/2})$ are non zero and rewrite Equation (11) as

$$\mathcal{P}_{\Delta t E^{n+1/2}}(p) = \alpha_0(\Delta t E^{n+1/2}) + \tilde{\alpha}_1(\Delta t E^{n+1/2}) i E^{n+1/2} \mathcal{Q}_{\Delta t E^{n+1/2}}(p),$$

where $\mathcal{Q}_{\Delta t E^{n+1/2}}(p)$ is a $(N-1)$-order polynomial with smaller degree term equal to p. The exact scheme for the N-level Liouville equation (14) reads

$$\begin{cases} (\Phi^{n+1/2}(\Delta t))^{-1}(\rho^{n+1} - \rho^n) = -i\left\{\tilde{V}^{n+1/2}\rho^{n+1} - \rho^n \tilde{V}^{n+1/2}\right\}, \\ \Phi^{n+1/2}(\Delta t) = \dfrac{\tilde{\alpha}_1(\Delta t E^{n+1/2})}{\alpha_0(\Delta t E^{n+1/2})} I, \end{cases} \tag{15}$$

where

$$\tilde{V}^{n+1/2} = E^{n+1/2}\mathcal{Q}_{\Delta t E^{n+1/2}}(p).$$

Remark 1. *Observe a nonlocal discretization and a renormalisation of the step size in the exact scheme for the N-level Liouville equation, according to the Mickens rules. In particular $\Phi^{n+1/2}(\Delta t) = \Delta t I + \mathcal{O}(\Delta t^2)$.*

Remark 2. *$\tilde{V}^{n+1/2}$ is not just equal to $V^{n+1/2}$ but there is an additional higher order term which is called a recovery factor because it comes from estimating the matrix exponential. The addition of this term contradicts one of the basic principles of the NSFD schemes theory, which forbids any form of adjustment by adding ad hoc terms.*

6 Numerical simulations

The decomposition of the Bloch equation into the relaxation–nutation evolution and the evolution of interaction with the electromagnetic field has two main advantages. First, part of the computations can be performed off-line, before the time iterations, calculating once and for all the eigenvalues of the electric dipole matrix, as well as the change of basis matrix and its inverse. Here we deal only with the Bloch equation, but this is even more efficient when there is space dependence, such as when coupling with Maxwell equations. The second advantage has been pointed out in [20] and demonstrated on a Self Induced Transparency test case, and is due to the fact that it decouples stiff and non-stiff parts of the equation.

In addition, the analysis of the error with respect to the time step size shows that decoupling the stiff and non-stiff parts decreases the error by a factor of 10 in favor of the splitting method. In Figure 1, we compare the errors for the exact scheme for the raw Bloch equation (of type (9) where V also contains the nutation term) with the Strang splitting scheme (10) for three-level case, for the schenario described in Section 6.1. We can then observe the gain linked to splitting even if the two schemes are of order two.

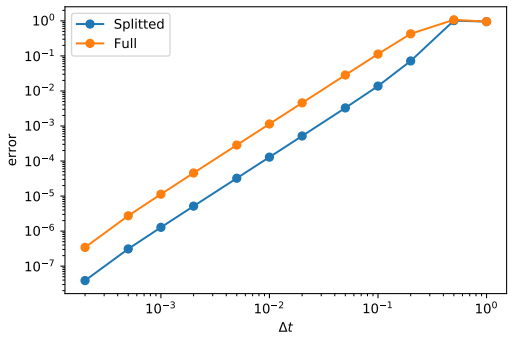

Figure 1: Error of the splitting and no-splitting methods with respect to the time-step in log-log scale.

In the following, we compare various schemes for the Bloch model, always performing the same decomposition but varying the way the exponential is calculated or approximated. We consider the historical method for the Bloch equation [26, 27, 28], namely the Crank–Nicolson method, although it has

been shown in Reignier's thesis [19] that it does not preserve the property of positivity for more than three levels. Other methods such as the fourth-order Runge–Kutta method are not adapted to preserve the physical properties of the Bloch equation [20]. We compare the Crank–Nicolson method with the methods presented in this paper, and the computation of the matrix exponential. We describe below the four methods.

Exponential method.

$$\rho^{n+1} = \exp(L\Delta t/2)\exp(-i\Delta t V^{n+1/2})\exp(L\Delta t/2)\rho^n \exp(i\Delta t V^{n+1/2}).$$

Crank–Nicolson method. The matrix exponential is approximated by the Crank–Nicolson scheme:

$$\begin{cases} \rho^{n+1} = \exp(L\Delta t/2)(A^{n+1/2})^{-1}\exp(L\Delta t/2)\rho^n A^{n+1/2}, \\ A^{n+1/2} = \left(I + \dfrac{i}{2}\Delta t V^{n+1/2}\right)\left(I - \dfrac{i}{2}\Delta t V^{n+1/2}\right)^{-1}. \end{cases} \tag{16}$$

Newton method. This is the method described in this paper using a Newton basis and Hörner algorithm for the construction of the polynomial:

$$\begin{cases} \rho^{n+1} = \exp(L\Delta t/2)(B^{n+1/2})^{-1}\exp(L\Delta t/2)\rho^n B^{n+1/2}, \\ B^{n+1/2} = \mathcal{P}_{\Delta t E^{n+1/2}}(p). \end{cases} \tag{17}$$

Canonical method. As in [20] the polynomial equivalent to the exponential is expressed in the canonical form. Details for three levels can be found in Appendix A:

$$\begin{cases} \rho^{n+1} = \exp(L\Delta t/2)(C^{n+1/2})^{-1}\exp(L\Delta t/2)\rho^n C^{n+1/2}, \\ C^{n+1/2} = \displaystyle\sum_{j=0}^{N-1} \alpha_j^{n+1/2}(V^{n+1/2})^j. \end{cases} \tag{18}$$

6.1 Three-level test case

We first compare the methods on a three-level test case. We suppose there is no relaxation ($Q(\rho) = 0$) and apply a sinusoidal input electrical field $E(t) = \sin(2\pi t)$ (recall we deal with dimensionless equations). The level frequencies are chosen to be 0, π and 2π since resonance with the input wave

is required for the system to evolve nontrivially. Besides the polarizability matrix p is chosen to be

$$p = \begin{pmatrix} 0 & 1 & 1.1 \\ 1 & 0 & 1 \\ 1.1 & 1 & 0 \end{pmatrix}.$$

Let n_{p} be the number of discretization times within one period of the input signal. The time-step is therefore equal to $1/n_{\mathrm{p}}$. The time evolution of populations over 20 periods of the input signal is displayed on Figure 2. This result has been obtained with the Python implementation of the matrix exponential $\exp(i\Delta t V^{n+1/2})$, and $n_{\mathrm{p}} = 20$, but similar results can be obtained with the other methods described in this paper.

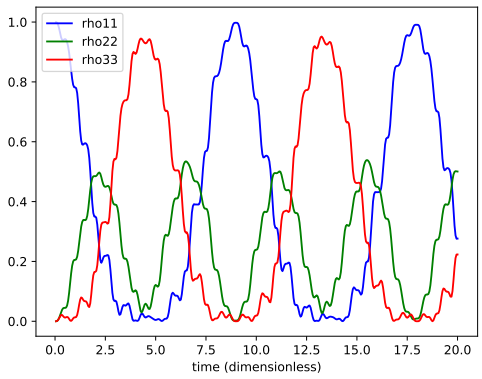

Figure 2: Time evolution of populations over 20 periods for a 3-level test-case.

Notice that the pseudo-period of the population is not that of the wave. The period of the wave can be however seen on the plots, since every flat part corresponds to a vanishing input electric field.

6.1.1 Computational time

We have two varying parameters, namely the numerical method and the time-step. We compare the computational time (for a not especially optimized python implementation on a small laptop). The results are gathered in Table 1. To homogenize the results, 2000 periods have been simulated (which,

n_p	Exponential	Crank–Nicolson	Newton	Canonical
5	10	4 (out)	4	3
10	14	6 (bad)	7	5
20	27	11 (bad)	15	11
100	131	38	78	53

Table 1: Computational time (in seconds) for a 3-level test-case.

coming back to a dimensional world, would correspond for light waves to about 10 ps).

The Exponential method is clearly the most expensive. The Crank–Nicolson is the less expensive, but the quality of the results disqualifies this method since we need a lot of points to ensure the same quality as with the other methods. With $n_\mathrm{p} = 5$, positiveness is violated from the very first periods on. Therefore, the Newton and Canonical methods seem to be the best, from the computational time point of view, with a little advance for the Canonical methods. We see next why we however prefer the Newton method.

6.1.2 Scheme order

The global error of the Strang method (for operators which do not commute) is of order 2 [21]. On the other hand, the Crank-Nicolson method, which corresponds to approximate exponentials at order 1, is of order 1.

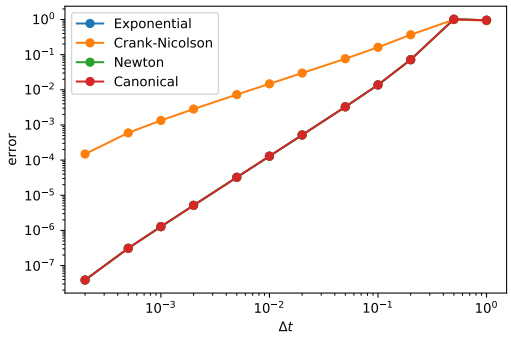

Figure 3: Error of the splitting methods with respect to the time-step in log-log scale.

Figure 3 shows the error of the splitting methods with respect to the time-step in log-log scale. The plots for the exponential, Newton and canonical methods are superimposed. The corresponding order is 2. As anticipated, Crank-Nicolson method is order 1.

6.1.3 Behavior in a degenerate case

We have previously chosen a strange matrix p to prevent it to have equal eigenvalues. Let us now take

$$p = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix},$$

which eigenvalues are -1 (double) and 2, and come back to 20 periods of the input signal and $n_{\mathrm{p}} = 20$. The results are displayed on Figure 4 for the Newton and the Canonical methods. The Exponential method serves here as a reference solution.

While both the Newton and the Canonical methods use the eigenvalues of the problem to compute the matrix exponential, the Canonical method is very sensitive to degenerate situations. In this case, the Jordan form of the the matrix is not diagonal and different formulae should be used (see [10] for full details). This does not solve the problem, since in the case of a nearly degenerate situation (two very close eigenvalues) the formulae presented in this paper (see Appendix) can used but would be very unstable.

6.2 N-level test case

Now, we compare only the Exponential and Newton methods, and have N vary. The results, namely the computational times, are gathered in Table 2 for various values of N and anew 2000 periods of the input signal.

We are not very fair with the Newton method, since we compare it with a clearly well optimized Python matrix exponential which computational time barely depends on the dimension of the matrix. In many studies however we are dealing with small density matrices, describing 2, 3 or 4 levels. In these cases the computational gain using the Newton method is not impressive, but can prove very useful for simulation over long physical times or involving many space locations.

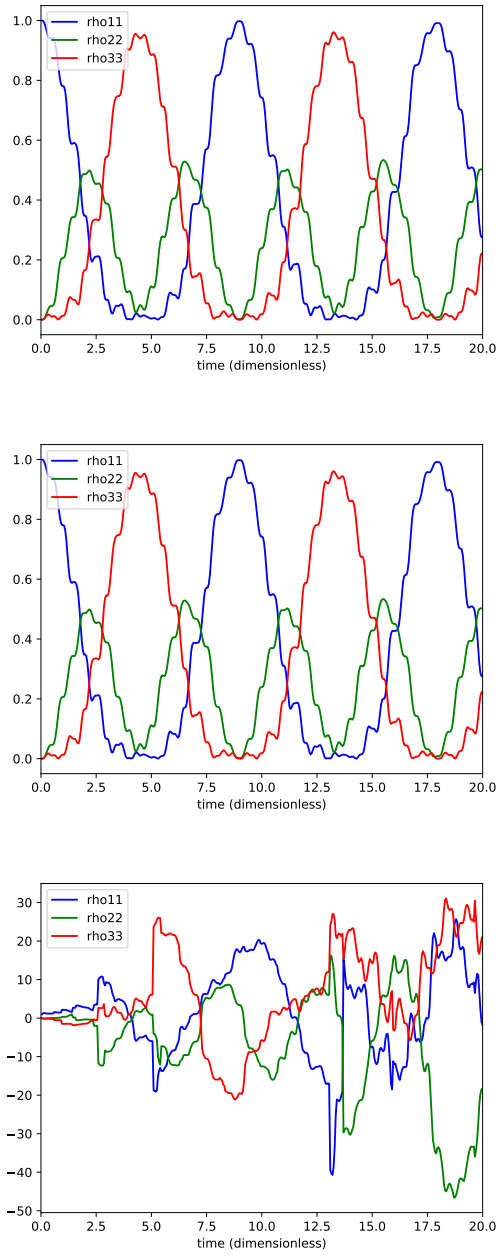

Figure 4: Time evolution of populations over 20 periods in a degenerate situation. From top to bottom: Exponential, Newton, and Canonical method.

N	Exponential	Newton
2	27	9
3	27	16
4	27	23
5	27	34
10	34	125

Table 2: Computational time (in seconds) for a N-level test-case.

7 Conclusion

We have derived splitting schemes for the N-level Bloch model. To this aim, the Bloch equation has been decomposed into a relaxation–nutation evolution and the interaction with the electromagnetic field (which is a Liouville equation). We are able to obtain exact solutions for the resulting sub-equations, and construct a Strang splitting scheme. The solution of the Liouville equation involves matrix exponentials and we discuss whether it is reasonable or not to compute it. Indeed, thanks to the Cayley–Hamilton theorem, it can be replaced by the computation of a polynomial. We used in particular Newton interpolation to define this polynomial. The resulting scheme has a variable time-step size and satisfies the nonstandard discretization rules of Mickens. Moreover, the splitting scheme preserves the qualitative and quantitative properties (Hermicity, trace conservation, positiveness) of Bloch equations.

The numerical comparison of the methods shows that computing a polynomial instead of the exponential is advantageous for small density matrices, i.e. a small number of levels. If the gain is relatively low, the number of such computations for a full Maxwell–Bloch simulation can really make it a good track to improve the computational load.

The choice of the splitting scheme aims at solving exactly each sub-equation, but also at dealing correctly with terms with different stiffness. An interesting goal in this direction is to derive methods that preserve the asymptotic behavior to the rate equations. If splitting methods are not direct candidates for this since they dissociate different parts of the Bloch equations which are intimately connected in the Boltzmann equation, it would be interesting to connect numerical solutions of these two sub-equations in an implicit method by the NSFD technique, hoping thus to obtain asymptotic preserving schemes. This is the object of our future research.

References

[1] J. Andreasen and H. Cao, *Finite–difference time–domain formulation of stochastic noise in macroscopic atomic systems,* J. Light. Technol. 27 (2009), pp. 4530–4535.

[2] R. Anguelov and J.M.-S. Lubuma. *Nonstandard finite difference method by nonlocal approximation,* Math. Comput. Simul. 61 (2003), pp. 465–475.

[3] B. Bidégaray, *Time discretizations for Maxwell–Bloch equations,* Numer. Methods Partial Differ. Equ. 19 (2003), pp. 284–300.

[4] B. Bidégaray-Fesquet, *Hiérarchie de modèles en optique quantique. De Maxwell–Bloch à Schrödinger non–linéaire,* Mathématiques et Applications Vol. 49, Springer, Berlin, 2006.

[5] B. Bidégaray, A. Bourgeade, and D. Reignier, *Introducing relaxation terms in Bloch equations,* J. Comput. Phys. 170 (2001), pp. 603–613.

[6] K. Blum, *Density Matrix. Theory and Applications,* 3rd ed., Springer, New York, 2012.

[7] R.W. Boyd, *Nonlinear Optics,* 3rd ed., Academic Press, 2008.

[8] C. Cohen-Tannoudji, J. Dupont–Roc, and G. Grynberg, *Processus d'interaction entre photons et atomes,* EDP sciences, CNRS Editions, 1996.

[9] T. Colin and B. Nkonga, *A Numerical model for light interaction with two-level atoms medium,* Phys. D 188 (2004), pp. 92–118.

[10] Q.A. Dang and M.T. Hoang, *Exact finite difference schemes for three-dimensional linear systems with constant coefficients,* Vietnam J. Math. 46 (2018), pp. 471–492.

[11] I. Farago and A. Havasi, *Consistency Analysis of Operator Splitting Methods for C_0-Semigroups,* Semigroup Forum 74 (2007) pp. 125–139.

[12] J.A. Gruetzmacher and N.F. Scherer, *Finite-difference time-domain simulation of ultrashort pulse propagation incorporating quantum-mechanical response functions,* Opt. Lett. 28 (2003), pp. 573–575.

[13] R. Loudon, *The Quantum Theory of Light,* 3rd ed., Oxford University Press, New York, 2000.

[14] R.E. Mickens, *Nonstandard Finite Difference Models of Differential Equations,* World Scientific, 1994.

[15] R.E. Mickens, *Nonstandard finite difference schemes,* In *Applications of Nonstandard Finite Difference Schemes,* R.E. Mickens, ed., World Scientific, 2000, pp. 1–54.

[16] R.E. Mickens, *Nonstandard finite difference methods,* In *Advanced in the applications of Nonstandard Finite Difference Schemes,* R.E. Mickens, ed., World scientific, 2005, pp. 1–9.

[17] C. Moler and C. Van Loan, *Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty–Five Years Later,* SIAM Rev. Soc. Ind. Appl. Math. 45 (2003), pp. 3–49.

[18] A.S. Nagra and R.A. York, *FDTD analysis of wave propagation in nonlinear absorbing and gain media,* IEEE Trans. Antennas Propag. 46 (1998), pp. 334–340.

[19] D. Reignier, *Couplage des équations de Maxwell avec les équations de Bloch pour la propagation d'une onde électromagnétique,* PhD thesis, Université Paul Sabatier, March 2000.

[20] M.E. Songolo and B. Bidégaray-Fesquet., *Nonstandard finite-difference schemes for the two-level Bloch model,* Int. J. Model. Simul. Sci. Comput. 9 (2018), pp. 1850033:1–23.

[21] B.Sportisse, *An analysis of operator splitting techniques in the stiff case,* J. Comput. Phys. 161 (2000), pp. 140–168.

[22] G. Strang, *On the construction and comparison of difference schemes,* SIAM J. Numer. Anal. 5 (1968), pp. 506–517.

[23] C. Uwizeye, *Asymptotic preserving scheme for Bloch equations,* Master Thesis, Université Joseph Fourier, June 2015.

[24] J. Weiner and P.-T. Ho, *Light–Matter Interaction. Fundamentals and Applications,* Wiley–Interscience, 2003.

[25] S. Weintraub, *Jordan Canonical Form: Application to Differential Equations,* Morgan and Claypool Publishers, 2008.

[26] R.W. Ziolkowski, J.M. Arnold, and D.M. Gogny, *Ultrafast pulse interaction with two–level atoms,* Phys. Rev. A 53 (1995), pp. 3082–3094.

[27] R.W. Ziolkowski, *Realization of an all–optical triode and diode with a two–level atom loaded diffraction grating,* Appl. Opt. 36 (1997), pp. 8547–8556.

[28] R.W. Ziolkowski, *The incorporation of microscopic material models into the FDTD approach for ultrafast optical pulse simulations,* IEEE Trans. Antennas Propag. 45 (1997), pp. 375–391.

A Alternative method for 3×3 matrix exponentials

A.1 Problem setting

The problem of the expression of exponential of matrices as exact finite difference schemes has been studied in [10] for general 3-order systems

$$\mathbf{x}'(t) = A\mathbf{x}(t);\ \mathbf{x}(t) = (x_1(t), x_2(t), x_3(t))^T,\ A \in \mathcal{M}_3(\mathbb{C}).$$

Here we restrict to the case where matrix A is similar to the canonical Jordan form

$$J_3 = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix},$$

where λ_1, λ_2, and λ_3 are distinct.
We have already seen that solving numerically $\mathbf{x}'(t) = J_3\mathbf{x}(t)$ amounts to solving three decoupled systems

$$x_1^{n+1} = x_1^n e^{\lambda_1 \Delta t},\ x_2^{n+1} = x_2^n e^{\lambda_2 \Delta t},\ x_3^{n+1} = x_3^n e^{\lambda_3 \Delta t}. \tag{A.1}$$

Back in the original basis, this can be written as

$$\mathbf{x}^{n+1} = (\alpha(\Delta t)I + \beta(\Delta t)A + \gamma(\Delta t)A^2)\mathbf{x}^n,$$

which we prefer to express as a polynomial of $\beta(\Delta t)A$:

$$\mathbf{x}^{n+1} = (\alpha(\Delta t)I + \beta(\Delta t)A + \xi(\Delta t)(\beta(\Delta t))^2 A^2)\mathbf{x}^n.$$

The explicit exact finite difference form is

$$\frac{\mathbf{x}^{n+1} - \alpha(\Delta t)\mathbf{x}^n}{\beta(\Delta t)} = A\mathbf{x}^n + \xi(\Delta t)\beta(\Delta t)A^2\mathbf{x}^n,$$

where $\alpha(\Delta t)$, $\beta(\Delta t)$, and $\xi(\Delta t)$ are parameters to be determined. The same coefficients appear in the Jordan basis, and identifying in Equation (A.1) we obtain

$$\alpha(\Delta t) + \beta(\Delta t)\lambda_j + \xi(\Delta t)(\beta(\Delta t))^2\lambda_j^2 = e^{\lambda_j \Delta t},\ j = 1, 2, 3.$$

A.2 Coefficients

Solving the Vandermonde system (A.1) is a classical problem. It can be expressed using determinants. Defining the determinant of the system:

$$\delta = \begin{vmatrix} 1 & \lambda_1 & \lambda_1^2 \\ 1 & \lambda_2 & \lambda_2^2 \\ 1 & \lambda_3 & \lambda_3^2 \end{vmatrix},$$

the coefficients are equal to

$$\delta\alpha(\Delta t) = \begin{vmatrix} e^{\lambda_1\Delta t} & \lambda_1 & \lambda_1^2 \\ e^{\lambda_2\Delta t} & \lambda_2 & \lambda_2^2 \\ e^{\lambda_3\Delta t} & \lambda_3 & \lambda_3^2 \end{vmatrix}, \delta\beta(\Delta t) = \begin{vmatrix} 1 & e^{\lambda_1\Delta t} & \lambda_1^2 \\ 1 & e^{\lambda_2\Delta t} & \lambda_2^2 \\ 1 & e^{\lambda_3\Delta t} & \lambda_3^2 \end{vmatrix}, \delta\gamma(\Delta t) = \begin{vmatrix} 1 & \lambda_1 & e^{\lambda_1\Delta t} \\ 1 & \lambda_3 & e^{\lambda_2\Delta t} \\ 1 & \lambda_3 & e^{\lambda_3\Delta t} \end{vmatrix}.$$

More explicitely we have

$$\begin{aligned}
\delta &= (\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2), \\
\alpha(\Delta t) &= \frac{e^{\lambda_1\Delta t}\lambda_2\lambda_3(\lambda_3 - \lambda_2) + e^{\lambda_2\Delta t}\lambda_3\lambda_1(\lambda_1 - \lambda_3) + e^{\lambda_3\Delta t}\lambda_1\lambda_2(\lambda_2 - \lambda_1)}{\delta}, \\
\beta(\Delta t) &= \frac{e^{\lambda_1\Delta t}(\lambda_2^2 - \lambda_3^3) + e^{\lambda_2\Delta t}(\lambda_3^2 - \lambda_1^3) + e^{\lambda_3\Delta t}(\lambda_1^2 - \lambda_2^3)}{\delta}, \\
\gamma(\Delta t) &= \frac{e^{\lambda_1\Delta t}(\lambda_3 - \lambda_2) + e^{\lambda_2\Delta t}(\lambda_1 - \lambda_3) + e^{\lambda_3\Delta t}(\lambda_2 - \lambda_1)}{\delta}, \\
\xi(\Delta t) &= \frac{\gamma}{\beta^2}.
\end{aligned} \tag{A.2}$$

Theorem 1. *For any matrix $A \in \mathcal{M}_3(\mathbb{C})$, whose eigenvalues are distinct,*

$$\exp(\Delta t A) = \alpha(\Delta t) I + \beta(\Delta t) A + \xi(\Delta t)(\beta(\Delta t))^2 A^2 \tag{A.3}$$

where $\alpha(\Delta t)$, $\beta(\Delta t)$ and $\xi(\Delta t)$ are determined by the relations (A.2)*.*

A.3 Application to the three-level Bloch equation

We now want to make explicit the coefficients α_0, α_1 and α_2 in Equation (18) to construct matrix $C^{n+1/2}$:

$$\exp(i\Delta t V^{n+1/2}) = \alpha_0^{n+1/2} I + \alpha_1^{n+1/2} V^{n+1/2} + \alpha_2^{n+1/2} (V^{n+1/2})^2. \tag{A.4}$$

Compared Equation (A.3) of Theorem 1 and Equation (A.4), we have $A = iV^{n+1/2}$, $\alpha_0^{n+1/2} = \alpha(\Delta t)$, $\alpha_1^{n+1/2} = i\beta(\Delta t)$, $\alpha_2^{n+1/2} = -\xi(\Delta t)\beta^2(\Delta t)$. Denoting $\theta_j^{n+1/2} = \lambda_j^{n+1/2}\Delta t$, $j = 1, 2, 3$, where the $\lambda_j^{n+1/2}$ are the distinct eigenvalues of matrix $V^{n+1/2}$, we obtain

$$\begin{aligned}
\delta^{n+1/2} &= (\lambda_2^{n+1/2} - \lambda_1^{n+1/2})(\lambda_3^{n+1/2} - \lambda_1^{n+1/2})(\lambda_3^{n+1/2} - \lambda_2^{n+1/2}),\\
\delta^{n+1/2}\alpha_0^{n+1/2} &= e^{i\theta_1^{n+1/2}} \lambda_2^{n+1/2}\lambda_3^{n+1/2}(\lambda_3^{n+1/2} - \lambda_2^{n+1/2})\\
&\quad + e^{i\theta_2^{n+1/2}} \lambda_3^{n+1/2}\lambda_1^{n+1/2}(\lambda_1^{n+1/2} - \lambda_3^{n+1/2})\\
&\quad + e^{i\theta_3^{n+1/2}} \lambda_1^{n+1/2}\lambda_2^{n+1/2}(\lambda_2^{n+1/2} - \lambda_1^{n+1/2}),\\
\delta^{n+1/2}\alpha_1^{n+1/2} &= i\Big(e^{i\theta_1^{n+1/2}}((\lambda_2^{n+1/2})^2 - (\lambda_3^{n+1/2})^2)\\
&\quad + e^{i\theta_2^{n+1/2}}((\lambda_3^{n+1/2})^2 - (\lambda_1^{n+1/2})^2)\\
&\quad + e^{i\theta_3^{n+1/2}}((\lambda_1^{n+1/2})^2 - (\lambda_2^{n+1/2})^2)\Big),\\
\delta^{n+1/2}\alpha_2^{n+1/2} &= -\Big(e^{i\theta_1^{n+1/2}}(\lambda_3^{n+1/2} - \lambda_2^{n+1/2})\\
&\quad + e^{i\theta_2^{n+1/2}}(\lambda_1^{n+1/2} - \lambda_3^{n+1/2})\\
&\quad + e^{i\theta_3^{n+1/2}}(\lambda_2^{n+1/2} - \lambda_1^{n+1/2})\Big).
\end{aligned}$$

Setting $\alpha^{n+1/2} = \alpha_0^{n+1/2}$, $\beta^{n+1/2} = -i\alpha_1^{n+1/2}$ and $\xi^{n+1/2} = \alpha_2^{n+1/2}/(\alpha_1^{n+1/2})^2$, the exact scheme for the Liouville equation becomes

$$\begin{aligned}
\rho^{n+1} = (&\alpha^{n+1/2} I + \beta^{n+1/2} V^{n+1/2} + \xi^{n+1/2}(\beta^{n+1/2})^2 (V^{n+1/2})^2)^{-1} \rho^n\\
&(\alpha^{n+1/2} I + \beta^{n+1/2} V^{n+1/2} + \xi^{n+1/2}(\beta^{n+1/2})^2 (V^{n+1/2})^2)
\end{aligned} \tag{A.5}$$

As we have seen for the two-level model in [20], it is also possible to write the exact scheme for the three-level Liouville equation in the form of a NSFD model. Indeed the scheme (A.5) can be written as

$$\begin{aligned}\alpha^{n+1/2}(\rho^{n+1}-\rho^n) = \beta^{n+1/2}[&\rho^n(V^{n+1/2}+\xi^{n+1/2}\beta^{n+1/2}(V^{n+1/2})^2)\\ &-(V^{n+1/2}+\xi^{n+1/2}\beta^{n+1/2}(V^{n+1/2})^2)\rho^{n+1}].\end{aligned}$$

If Δt is small enough, we can ensure that $\alpha^{n+1/2}$ and $\beta^{n+1/2}$ are nonzero and

$$\begin{aligned}i\frac{\alpha^{n+1/2}}{\beta^{n+1/2}}(\rho^{n+1}-\rho^n) = &-i[(V^{n+1/2}+\xi^{n+1/2}\beta^{n+1/2}(V^{n+1/2})^2)\rho^{n+1}\\ &-\rho^n(V^{n+1/2}+\xi^{n+1/2}\beta^{n+1/2}(V^{n+1/2})^2)]\end{aligned}$$

or equivalently

$$\begin{cases}(\Phi^{n+1/2}(\Delta t))^{-1}(\rho^{n+1}-\rho^n) = -i[(V^{n+1/2}+\xi^{n+1/2}\beta^{n+1/2}(V^{n+1/2})^2)\rho^{n+1}\\ \qquad\qquad -\rho^n(V^{n+1/2}+\xi^{n+1/2}\beta^{n+1/2}(V^{n+1/2})^2)],\\ \Phi^{n+1/2}(\Delta t) = -i\dfrac{\beta^{n+1/2}}{\alpha^{n+1/2}}I.\end{cases}$$

In the left-hand side, we can recognize a nonstandard discretization in which the discretization time-step size undergoes a renormalization. We also notice that the renormalization matrix $\Phi^{n+1/2}$ has the following property:

$$\Phi^{n+1/2}(\Delta t) = \Delta t I + \mathcal{O}(\Delta t^2) \text{ when } \Delta t \to 0,$$

because

$$\lim_{\Delta t\to 0}\alpha^{n+1/2} = 1;\ \lim_{\Delta t\to 0}\beta^{n+1/2} = i\Delta t;\ \lim_{\Delta t\to 0}\xi^{n+1/2} = \frac{1}{2}.$$

The Strang splitting scheme derived from Equations (8) and (A.5) is exactly (18). This scheme has a variable time-step size and preserves positiveness, because both steps (8) and (A.5) are positive. The trace is also conserved.