A PSEUDOSPECTRAL METHOD FOR SOLVING THE BLOCH EQUATIONS OF THE POLARIZATION DENSITY IN $e^-$ STORAGE RINGS

Klaus A. Heinemann*, Oleksii Beznosov, James A. Ellison, UNM, Albuquerque, New Mexico
Desmond P. Barber, DESY, Hamburg and UNM
Daniel Appelö, University of Colorado, Boulder, Colorado

INTRODUCTION

We consider the numerical evolution of the Bloch equations of Derbenev and Kondratenko (DK) for the polarization density in high-energy electron storage rings, such as the proposed FCC-eel and CEPC. Equilibrium spin polarization is well characterized by the DK formulas for current rings (see [1] and [2]), but deviations may be important at the high energies we have in mind. We believe these Bloch equations, in three degrees of freedom (DOF), derived in [3] give a more complete description at all energies. The equations are a system of three coupled linear partial differential equations for the three components of the polarization density and include the spin flip polarization effect. Following [4] we formulate the equations in action-angle variables and approximate the Fokker-Planck terms. Our goal is to integrate these equations numerically in order to approximate the equilibrium and compare with the DK formulas.

For 3 DOF the polarization density has $6 + 1$ independent variables. For simplicity, suppose that each of the space-like variables has been discretized on a grid with $N$ grid-points, then the computational cost of each time step will scale no better than $O(N^6)$. The presence of parabolic terms in the governing equations necessitates implicit time stepping and thus solutions of linear systems of equations. For a fully coupled problem this will bring the per time step cost to $O(N^{6q})$, with $1 \leq q \leq 3$, depending on the algorithms used for the linear solve. However, only algorithms with $q \approx 1$ are feasible (for Gaussian elimination $q = 3$). Fortunately, as we outline below, the structure of the equations allows us to group the space-like variables into two groups resulting in a cost that, to leading order, scales as $O(MN^{3q})$.

In the next section we present the full and the reduced Bloch equations and the underlying physics. Then we discuss the details of the numerical algorithm outlined above and the issue of complexity in higher dimensions. Finally we present two numerical results showing a depolarization calculation in 1DOF and spectral convergence in a 2DOF example.

BLOCH EQUATION

In a semiclassical probabilistic description of an electron bunch the spin-orbit dynamics is described by the density matrix function (or spin-1/2 Wigner function) $\rho$ written as

$$\rho(t; z) = f(t; z)\frac{1}{2} \left( I_{2\times 2} + \vec{\sigma} \cdot \vec{P}_{loc}(t; z) \right),$$

(1)

where $f$ is the classical phase-space density normalized by $\int f(t; z)dz = 1$ and $\vec{P}_{loc}$ is the local polarization vector.

* Corresponding author: heinemann@math.unm.edu
For a numerical discretization of (2)-(4) it is more efficient to pose the equations in polar coordinates, \((r_k, \phi_k)\) (transformed by first going to action angle variables, [4]) and to parametrize time by the angular azimuth \(\theta\). The reduced Bloch equations (4) then become

\[
\partial_{\theta} \vec{\eta} = L_{FP} \vec{\eta} + \mathcal{A} (r_1, \ldots, r_d, \phi_1, \ldots, \phi_d, \theta) \vec{\eta}.
\]  

(5)

Here the coupling matrix \(\mathcal{A}\) is skew-symmetric and defined by \(\mathcal{A}_{12} = -\Omega_3, \mathcal{A}_{13} = \Omega_2, \mathcal{A}_{23} = -\Omega_1\), and

\[
L_{FP} = \frac{L}{2\pi} \sum_{l=1}^{d} \left\{ -b_l \frac{\partial}{\partial \phi_l} + \frac{a_l}{r_l} \frac{\partial}{\partial r_l} r_l^2 + \frac{a_l}{2} \Delta_{r,l} \phi_l \right\},
\]  

(6)

\[
\hat{\Omega} = \hat{\Omega}^r (\theta) + \sum_{l=1}^{d} r_l [\hat{\Omega}^r (\theta) \cos \phi_l + \hat{\Omega}^s (\theta) \sin \phi_l].
\]  

(7)

Here \(d \in \{1, 2, 3\}\) is the number of degrees of freedom to be considered. \(\Delta_{r,l}\) is the Laplacian in polar coordinates and \(L\) is the length of the design orbit. For a complete derivation of (6)-(7) with underlying assumptions see [4].

The polarization vector \(\hat{P}(\theta)\) of the bunch at time \(\theta\) is

\[
\hat{P}(\theta) = \int f(\theta, z) \hat{P}_{loc}(\theta, z) dz = \int \vec{\eta}(\theta, z) dz,
\]  

(8)

where \(z = (r_1, \ldots, r_d, \phi_1, \ldots, \phi_d)\) and the polarization is \(\hat{P}(\theta)\). This is to be compared with the DK formulas [2] elsewhere.

**NUMERICAL ALGORITHM**

Consider (5) with \(d = 1\),

\[
\partial_{\theta} \vec{\eta} = \frac{L}{2\pi} \left\{ -b \frac{\partial}{\partial \varphi} + \frac{a}{\varphi} \frac{\partial}{\partial r} r^2 + \frac{a}{2} \Delta_{r,\varphi} \right\} \vec{\eta} + \mathcal{A}(r, \varphi, \theta) \vec{\eta},
\]  

(9)

posed on a disk \(r \leq r_{\text{max}}, \varphi \in [0, 2\pi]\). The boundary conditions are periodic in \(\varphi\) and we take \(r_{\text{max}}\) large enough to impose homogenous Dirichlet boundary conditions at \(r = r_{\text{max}}\), since the initial condition has support. We seek approximations to \(\vec{\eta}\) on a Chebyshev grid in \(r\) and a uniform grid in \(\varphi\),

\[
r_i = -\cos \left( \frac{\pi i}{n_r} \right), \quad i = 0, \ldots, n_r,
\]

\[
\varphi_j = j \frac{2\pi}{n_{\varphi}}, \quad j = 1, \ldots, n_{\varphi},
\]

and expand it in a Fourier series in the \(\varphi\) direction:

\[
\vec{\eta}(r_i, \varphi, \theta) \approx \sum_{k=1}^{n_{\varphi}} \hat{\eta}_k(r_i, \theta) e^{ik\varphi_j}.
\]

For the \(k\)th Fourier mode we determine \(\hat{\eta}_k(r, \theta)\) from

\[
\frac{\partial \hat{\eta}_k}{\partial \theta} = \frac{L}{2\pi} \left( 2\hat{\eta}_k + \frac{1}{2} (2r + \frac{1}{r}) \frac{\partial \hat{\eta}_k}{\partial r} + \frac{1}{2} \frac{\partial^2 \hat{\eta}_k}{\partial r^2} - ikb \hat{\eta}_k - \frac{ak^2}{2r^2} \hat{\eta}_k \right) + \left[ \text{FFT}[\mathcal{A}(r, \varphi, \theta) \vec{\eta}] \right]_k.
\]  

(10)

Here [FFT\([f \cdot g]\)]_k represents the discrete Fourier transform of the \(k\)th Fourier mode of the convolution \(f \ast g\).

Now denote by \(\hat{u}_k(\theta)\) the grid function on the \(r\) grid for a fixed mode, i.e. \(\hat{u}_k(\theta) = \hat{u}_k(r_0, \theta), \ldots, \hat{u}_k(r_{n_r}, \theta)\) T, describing one of the three components of \(\vec{\eta}_k\). Then for each component in (10) we have

\[
\frac{d\hat{u}_k}{d\theta} = \{C_1 + C_2(k)\} \hat{u}_k + F^u_k(\vec{\eta}).
\]  

(11)

Here \(C_1\) and \(C_2(k)\) are matrices representing the mode independent and mode dependent components of the Fokker-Planck operator

\[
C_1 = \frac{L}{2\pi} \left( 2I + (R + \frac{1}{2} R^{-1}) D_1 + \frac{1}{2} D_2 \right),
\]

\[
C_2(k) = \frac{L}{2\pi} \left( -ikbI - ak^2(R^{-1})^2 \right),
\]

and \(F^u_k\) represents the \(k\)th mode of the \(\Omega\) coupling term as well as contributions from boundary conditions in \(r\). Further, \(I\) is the identity matrix, \(R = \text{diag}(r_0, \ldots, r_{n_r})\), and \(D_1\) and \(D_2\) are spectral differentiation matrices. The entries of the differentiation matrices are found by the techniques for constructing finite difference approximations of any order of accuracy, for any order of the derivative and on general grids described by Fornberg in [10]. To be precise, the coefficients are computed using a numerically stable recursion relation derived from the Lagrange interpolant associated with the grid points (see also the subroutine \texttt{weights.f} provided in [10]).

To evolve in time we use a second order splitting scheme, see e.g. [11]. Let \(\hat{u}_k^\nu = \hat{u}_k(\nu \Delta \theta)\) then, for each mode, we compute

\[
\hat{u}_k^\nu = 2\hat{u}_k^\nu - \frac{1}{2} \hat{u}_k^{\nu-1} + \Delta \theta \left[ 2(F^u_k(\hat{\eta}^\nu) - F^u_k(\hat{\eta}^{\nu-1}) \right],
\]  

(12)

and solve

\[
\left[ \frac{3}{2} I - \Delta \theta (C_1 + C_2(k)) \right] \hat{u}_k^{\nu+1} = \hat{u}_k^\nu.
\]  

(13)

Thus at each step we compute \(\hat{\eta}^{\nu+1}\) given \(\hat{\eta}^\nu\) and \(\hat{\eta}^{\nu-1}\). For \(d = 2\) and \(d = 3\) the algorithm stays the same.

The dominating cost in the step (12) is the computation FFT\([\mathcal{A}(r, \varphi, \theta) \vec{\eta}]\) for each gridline in \(r\) using the fast Fourier transform. The cost of this is \(O(n_r n_{\varphi} \log n_{\varphi})\). As indicated in the introduction, the cost of the solve in the step (13) depends on the choice of algorithm but can always be split into an initial cost (e.g. LU-factorization) and a solve cost (e.g. back and forward substitution). Ignoring the startup cost, which can be amortized over many time steps, the cost per time step is that of \(n_{\varphi}\) solves of size \(n_r\), i.e. \(O(n_{\varphi} n_r^3)\).

For 3 DOF the complexity estimate becomes

\[
C = O((n_r n_{\varphi})^3 \log n_{\varphi}) + O(n_{\varphi}^3 n_r^3).
\]

Assuming that \(n_r = n_{\varphi} = N\) we find that for \(N = 50\) and \(q = 1, 4/3, 2\) the cost \(C = 2 \cdot 10^{11}, 9.6 \cdot 10^{11}\) and \(2 \cdot 10^{15}\), respectively.
As a single modern processing unit may be able to carry out $O(10^8 - 10^9)$ arithmetic operations per second it appears plausible that an efficient parallel implementation can result in time-per-time step on the order of one to several seconds for $q = 4/3$.

Note that there are several modern solution techniques, like the Hierarchical Poincaré-Steklov operator technique by Martinsson [12] that can reach $q \approx 1$ for spectrally accurate discretizations.

**NUMERICAL RESULTS**

**Example 1: A Single Resonance Model (SRM) in 1DOF**

We now discuss the following SRM for which $d = 1$ and where $\tilde{\Omega} = (\sigma_0 + \sigma_1 r \cos \varphi + \sigma_2 r \sin \varphi)(0,0,1)^T$. Here the reduced Bloch equations (5) can be solved exactly [13]. For example if the initial condition on $\tilde{\eta}$ is chosen as $\tilde{\eta}(r, \varphi, 0) = (1/\pi) \exp(-r^2)(1,0,0)^T$ then the polarization at time $\theta$ reads as

$$|	ilde{P}(\theta)| = \exp \left\{ \frac{\pi^2}{(a^2 + b^2)^2} \left[ 2 \left( a^2 - b^2 - a\theta (a^2 + b^2) \right) + 2e^{-a\theta} \left( (b^2 - a^2) \cos(b\theta) + 2ab \sin(b\theta) \right) \right] \right\}. \quad (14)$$

This example was used to verify the spectral convergence of our 1DOF method outlined above (see also Example 2). In Fig. 1 we show the polarization $|\tilde{P}(\theta)|$ obtained by integrating our numerical solution (see (8)). The result is very close to the exact solution of (14), within the error we expect. In addition, the figure shows the effect of the transitional terms in (14) as well as the subsequent exponential decay of the polarization with the depolarization time $\tau_{dep}$ defined by $\tau_{dep}^{-1} = \frac{2a \pi^2 (\sigma_1^2 + \sigma_2^2)}{a^2 + b^2}$.

**Example 2: Spectral convergence for 2DOF**

To confirm the spectral convergence in $r_1$ and $r_2$ we evolve (5) in 2DOF with the initial data taken to be the exact solution (obtained for constant $\tilde{\Omega}$)

$$\tilde{\eta}(\theta, \vec{r}, \vec{\phi}) = \exp(\mathcal{A} \theta) \tilde{\eta}(\theta, \vec{r}, \vec{\phi}). \quad (15)$$

at $\theta = 0$. Here each component of $\tilde{\eta}$ satisfies $\partial_{\theta} f = L_{FP} f$ which can be solved in terms of the transition probability density for a Gaussian process with easily determined mean and covariance matrix. Moreover, it has an equilibrium solution $\pi^{-2} \exp(-(r_1^2 + r_2^2))$ making it a good test case for the numerical method. To be precise, the errors displayed in Fig. 2 are the maximum deviation from the exact solution $\tilde{\eta}(\theta, \vec{r}, \vec{\phi}) = \pi^{-2} \exp(-(r_1^2 + r_2^2)) \exp(\mathcal{A} \theta)(1,1,1)^T$, taken over all grid points and all the variables at $\theta = \pi/4$ and with $\tilde{\Omega} = (0,0,1)^T$ and $a_1 = b_1 = a_2 = b_2 = 1$. The results clearly show the spectral accuracy of the spatial discretization as well as the second order accuracy of the temporal discretization. A similar plot was obtained for 1DOF in Example 1.

**DISCUSSION AND NEXT STEPS**

We are preparing an extended version of this brief note for an archival journal which will complete the work on the reduced Bloch equation of (5) in 2DOF. An important aspect will be a more detailed discussion of the algorithm. The codes will be made available in a repository. A goal is to make our work easily reproducible. Next we will incorporate the spin flip by considering the full Bloch equations and do a careful study of the depolarization and polarization effects for the simple lattice incorporated by the $(a_i, b_i)$ in (6). This will include depolarization and polarization times and equilibrium. We will then study a more realistic lattice in the 2DOF case and begin the 3DOF work, where a parallel algorithm will surely be important/necessary.

**ACKNOWLEDGEMENT**

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under Award Number DE-SC00180808.
REFERENCES


