



HAL
open science

Computational performance of simple and efficient sequential and parallel Dirac equation solvers

Xavier Antoine, Emmanuel Lorin

► **To cite this version:**

Xavier Antoine, Emmanuel Lorin. Computational performance of simple and efficient sequential and parallel Dirac equation solvers. *Computer Physics Communications*, 2017, 220, pp.150-172. 10.1016/j.cpc.2017.07.001 . hal-01496817

HAL Id: hal-01496817

<https://hal.science/hal-01496817>

Submitted on 27 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computational performance of simple and efficient sequential and parallel Dirac equation solvers

X. Antoine^a, E. Lorin^{b,c}

^a*Institut Elie Cartan de Lorraine, Université de Lorraine, Sphinx team, Inria Nancy-Grand Est, F-54506 Vandoeuvre-lès-Nancy Cedex, France*

^b*School of Mathematics and Statistics, Carleton University, Ottawa, Canada, K1S 5B6*

^c*Centre de Recherches Mathématiques, Université de Montréal, Montréal, Canada, H3T 1J4*

Abstract

This paper is dedicated to the study of the computational performance of basic and efficient pseudo-spectral methods [9, 20, 27, 28] and of a more recent Quantum Lattice Boltzmann-like approach [13, 15, 26, 32, 35] for solving the Time Dependent Dirac Equation (TDDE) modeling the interaction of classical electromagnetic fields with quantum relativistic particles.

Keywords: Dirac equation, pseudo-spectral method, high performance computing, wave equation.

1. Introduction

In this work, we are interested in the simulation of the relativistic dynamics of an electron of mass m coupled to an external classical electromagnetic field. The time-dependent Dirac equation under consideration reads [24]

$$i\partial_t\psi(t, \mathbf{x}) = H\psi(t, \mathbf{x}) \quad (1)$$

where $\psi(t, \mathbf{x})$ is the time and coordinate dependent four-spinor, and H is the Hamiltonian operator. The latter is given by

$$H = \boldsymbol{\alpha} \cdot [c\mathbf{p} - e\mathbf{A}(t, \mathbf{x})] + \beta mc^2 + \mathbb{I}_4 V(t, \mathbf{x}), \quad (2)$$

where the momentum operator is $\mathbf{p} = -i\nabla$. More specifically, the Dirac equation under consideration reads [24]

$$i\partial_t\psi(t, \mathbf{x}) = \left\{ \alpha_x \left[-ic\partial_x - eA_x(t, \mathbf{x}) \right] + \alpha_y \left[-ic\partial_y - eA_y(t, \mathbf{x}) \right] + \alpha_z \left[-ic\partial_z - eA_z(t, \mathbf{x}) \right] + \beta mc^2 + \mathbb{I}_4 V(t, \mathbf{x}) \right\} \psi(t, \mathbf{x}), \quad (3)$$

Email addresses: xavier.antoine@univ-lorraine.fr (X. Antoine), elorin@math.carleton.ca (E. Lorin)

where $\psi(t, \mathbf{x}) \in L^2(\mathbb{R}^3) \otimes \mathbb{C}^4$ is the time and coordinate ($\mathbf{x} = (x, y, z)$) dependent four-spinor. In (3), $\mathbf{A}(t, \mathbf{x})$ represents the three space components of the electromagnetic vector potential, $V(t, \mathbf{x}) = eA_0(t, \mathbf{x}) + V_{\text{nuc.}}(\mathbf{x})$ is the sum of the scalar and interaction potentials, e is the electric charge (with $e = -|e|$ for an electron), \mathbb{I}_4 is the 4×4 unit matrix and $\boldsymbol{\alpha} = (\alpha_\gamma)_{\gamma=1, \dots, 4}, \beta$ are the Dirac matrices. In this work, the Dirac representation is used, where

$$\alpha_\gamma = \begin{bmatrix} 0 & \sigma_\gamma \\ \sigma_\gamma & 0 \end{bmatrix}, \quad \beta = \begin{bmatrix} \mathbb{I}_2 & 0 \\ 0 & -\mathbb{I}_2 \end{bmatrix}. \quad (4)$$

The σ_γ are the usual 2×2 Pauli matrices defined as

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{and} \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad (5)$$

while \mathbb{I}_2 is the 2×2 unit matrix. Note that the light velocity c and fermion mass m are kept explicit in Eq. (2), allowing to adapt the method easily to natural or atomic units (a.u.).

Throughout this work, we consider the single particle Dirac equation which is relevant for calculations describing Quantum Electrodynamics (QED) processes coupled to strong classical fields (e.g. particle-antiparticle pair creation from very high intensity electromagnetic classical fields [14]). *The purpose of this paper is not to derive original numerical solvers for the Dirac equation, or to provide simulations of specific quantum relativistic problems, but to compare the efficiency of two existing simple but accurate methods for solving the Dirac equation.* Although *pseudo-spectral methods* are used for decades for solving the Dirac equation, for instance for Graphene or pair production simulations [1, 9, 10, 11, 18, 31, 33, 36], recently a real space method [13, 15, 26, 32] was established to address these physical questions. This real space method, which has closed connection with the Quantum Lattice Boltzmann method [12, 35] is simply a finite-difference method at CFL = 1 [34], for solving a first order linear hyperbolic system with eigenvalues of equal magnitude. This method which is based on the Method of Characteristics (MC) will be referred in this paper to as a *MC-based method*. Thanks to its simplicity, the MC-based method is shown to be highly scalable and much more efficient sequentially and in parallel than pseudo-spectral methods for a fixed number of degrees of freedom (dof). Its main weakness is however the strict condition which is imposed on the space step. In comparison, the pseudo-spectral methods are much more flexible regarding the choice of the spatial discretization step, that is the number of dof. In summary, for physical problems involving very small space scales, the MC-based will be shown to be much more efficient than pseudo-spectral methods. This will be the case for instance for problems involving heavy ions, or for intense and short laser-molecule interactions. In the opposite, when a fine spatial resolution is not required pseudo-spectral methods will be shown to be more efficient thanks to the possibility to select much coarser meshes, while keeping a very good accuracy (spectral convergence). Naturally, other types of methods exist for solving the Dirac equation, such as variational methods [16, 17, 19, 21, 25, 37] or Krylov-type methods [8], but these are not discussed in this paper, which focuses on simple and efficient techniques.

The paper is organized as follows. In Section 2 (resp. Section 3), we recall the basics of the MC-based (resp. basic pseudo-spectral) method for the Dirac equation modeling the interaction of an electron with an external electromagnetic field. We next present in Section 4, an overview of the parallel computing aspects. In Section 5 (resp. Section 6), a series of sequential and parallel experiments is presented to illustrate the strengths and weaknesses of each method in the one-dimensional (resp. multi-dimensional) case. We conclude in Section 7.

2. Method of characteristic-based TDDE Solver in cartesian coordinates

2.1. Operator Splitting (first order)

We here recall the principle of operator splitting for the computation of an approximate solution to Eqs. (1) and (2) at time t_{n+1} , and denoted by $\psi^{n+1}(\mathbf{x})$. The initial condition at time t_n is assumed given by

$$\psi(t_n, \mathbf{x}) = \psi^n(\mathbf{x}). \quad (6)$$

As previously discussed in [26], this can be done [13, 15] with an operator splitting scheme. We first define the operators

$$A = -i\alpha_x \partial_x \quad (7)$$

$$B = -i\alpha_y \partial_y \quad (8)$$

$$C = -i\alpha_z \partial_z \quad (9)$$

$$D = \beta mc^2 + \mathbb{1}_4 V(t, \mathbf{x}) - e\boldsymbol{\alpha} \cdot \mathbf{A}(t, \mathbf{x}). \quad (10)$$

The following splitting in Cartesian coordinates is considered [26] (the \mathbf{x} -dependence in the wavefunction argument for notational convenience):

$$i\partial_t \psi^{(1)}(t) = A\psi^{(1)}(t), \quad \psi^{(1)}(t_n) = \psi^n, \quad t \in [t_n, t_{n+1}) \quad (11)$$

$$i\partial_t \psi^{(2)}(t) = B\psi^{(2)}(t), \quad \psi^{(2)}(t_n) = \psi^{(1)}(t_{n+1}), \quad t \in [t_n, t_{n+1}) \quad (12)$$

$$i\partial_t \psi^{(3)}(t) = C\psi^{(3)}(t), \quad \psi^{(3)}(t_n) = \psi^{(2)}(t_{n+1}), \quad t \in [t_n, t_{n+1}) \quad (13)$$

$$i\partial_t \psi^{(4)}(t) = D\psi^{(4)}(t), \quad \psi^{(4)}(t_n) = \psi^{(3)}(t_{n+1}), \quad t \in [t_n, t_{n+1}) \quad (14)$$

$$\text{and } \psi^{n+1} = \psi^{(4)}(t_{n+1}) \quad (15)$$

where the upper subscript in parenthesis on the wavefunction denotes the splitting step number. Note that this simple splitting scheme leads to an error that scales in $O(\Delta t^2)$, corresponding to a first-order numerical scheme (for more details on the analysis of the method, see [26]). The method consists of solving each equation independently with an initial condition given by the solution of the previous step. Note also that for every step, the time increment is identical, i.e. $\Delta t \equiv t_{n+1} - t_n$. This splitting approach will be used in this paper with both the MC-based and pseudo-spectral-methods.

2.2. Method of characteristics (MC) based method

In this section, we summarize the MC-based method which was proposed in [26, 13]. Eqs. (11) to (13) can actually be solved independently using the method of characteristics (MC). We first diagonalize the Dirac matrix, thus decoupling the spinor components. The resulting equation has a form similar to set of advection equations (linear first-order in time and space derivative). The method of characteristics is then used to find an explicit analytical solution. Finally, the solution is transformed back to the original Dirac matrix representation. The explicit solution to these equations and computational details can be found in [13], as well as [15] in the cylindrical coordinates. The final result for the solutions of Eqs. (11) to (13) is

$$\begin{aligned} \psi^{(1)}(t_{n+1}, \mathbf{x}) = & \frac{1}{2} \left\{ [\mathbb{I}_4 + \alpha_x] \psi^n(x - c\Delta t, y, z) \right. \\ & \left. + [\mathbb{I}_4 - \alpha_x] \psi^n(x + c\Delta t, y, z) \right\} \end{aligned} \quad (16)$$

$$\begin{aligned} \psi^{(2)}(t_{n+1}, \mathbf{x}) = & \frac{1}{2} \left\{ [\mathbb{I}_4 + \alpha_y] \psi^{(1)}(t_{n+1}, x, y - c\Delta t, z) \right. \\ & \left. + [\mathbb{I}_4 - \alpha_y] \psi^{(1)}(t_{n+1}, x, y + c\Delta t, z) \right\} \end{aligned} \quad (17)$$

$$\begin{aligned} \psi^{(3)}(t_{n+1}, \mathbf{x}) = & \frac{1}{2} \left\{ [\mathbb{I}_4 + \alpha_z] \psi^{(2)}(t_{n+1}, x, y, z - c\Delta t) \right. \\ & \left. + [\mathbb{I}_4 - \alpha_z] \psi^{(2)}(t_{n+1}, x, y, z + c\Delta t) \right\} \end{aligned} \quad (18)$$

The space domain is then discretized in cubic elements with edges of length $a = \Delta x = \Delta y = \Delta z$, inside which the wavefunction is constant (Q_0 -type elements). The discretized wavefunction and electromagnetic field are then written as

$$\psi_h(t, \mathbf{i}) = \sum_{m=1}^N \mathbf{1}_m(\mathbf{i}) \psi(t, \bar{\mathbf{x}}_m) \quad (19)$$

$$\mathbf{A}_h(t, \mathbf{i}) = \sum_{m=1}^N \mathbf{1}_m(\mathbf{i}) \mathbf{A}(t, \bar{\mathbf{x}}_m) \quad (20)$$

where $N = N_x N_y N_z$ is the total number of elements, $\psi_h(t, \mathbf{i})$ and $\mathbf{A}_h(t, \mathbf{i})$ are the discretized wavefunction and electromagnetic field ($\mathbf{i} \equiv (i, j, k) \in \mathbb{Z}^3$ are indexing the volumes), the function $\mathbf{1}_m(\mathbf{i})$ is equal to 1 in volume m indexed by \mathbf{i} and is zero outside, while $\bar{\mathbf{x}}_m$ is the vector pointing to the center of volume m with the following components:

$$\bar{\mathbf{x}}_m = \left(x_{\min} + (i + \frac{1}{2})a, y_{\min} + (j + \frac{1}{2})a, z_{\min} + (k + \frac{1}{2})a \right) \quad (21)$$

where $x_{\min}, y_{\min}, z_{\min}$ are the lower domain boundary coordinates. The constraint $c\Delta t = a$ (with space step Δt and space step a) has to be satisfied for stability and accuracy reasons

[26]. It then allows to write $\psi(x \pm c\Delta t) \rightarrow \psi_h(i \pm K)$, with $K \in \mathbb{N}^*$. This discretized scheme is then exact in each dimension (up to errors coming from the projection of the grid) for any $K \in \mathbb{N}^*$ such that $c\Delta t = Ka$. Finally Eqs. (16) to (18) become for $K = 1$

$$\psi_h^{n1}(\mathbf{i}) = \frac{1}{2} \left\{ [\mathbb{I}_4 + \alpha_x] \psi_h^n(i-1, j, k) + [\mathbb{I}_4 - \alpha_x] \psi_h^n(i+1, j, k) \right\} \quad (22)$$

$$\psi_h^{n2}(\mathbf{i}) = \frac{1}{2} \left\{ [\mathbb{I}_4 + \alpha_y] \psi_h^n(i, j-1, k) + [\mathbb{I}_4 - \alpha_y] \psi_h^n(i, j+1, k) \right\} \quad (23)$$

$$\psi_h^{n3}(\mathbf{i}) = \frac{1}{2} \left\{ [\mathbb{I}_4 + \alpha_z] \psi_h^n(i, k, k-1) + [\mathbb{I}_4 - \alpha_z] \psi_h^n(i, k, k+1) \right\} \quad (24)$$

$$\psi_h^{n+1}(\mathbf{i}) = \exp \left[-i\beta mc^2 \Delta t - i\tilde{V}_h^n(\mathbf{i}) + i\boldsymbol{\alpha} \cdot \tilde{\mathbf{A}}_h(\mathbf{i}) \right] \psi_h^{n3}(\mathbf{i}) \quad (25)$$

where

$$\begin{aligned} \tilde{\mathbf{A}}^n(\mathbf{x}) &= e \int_{t_n}^{t_{n+1}} d\tau \mathbf{A}(\tau, \mathbf{x}) \approx \mathbf{A}^n(\mathbf{x}) \Delta t \\ \tilde{V}^n(\mathbf{x}) &= e \int_{t_n}^{t_{n+1}} d\tau V(\tau, \mathbf{x}) \approx V^n(\mathbf{x}) \Delta t. \end{aligned} \quad (26)$$

The constraint $a = c\Delta t/K$ is very restrictive and makes sense when \mathbf{A} , V (including $V_{\text{nuc.}}$) or $\psi(\cdot, 0)$ possess wavenumbers of the order of $1/mc$. In Eq. (14), the solution is simply given by

$$\begin{aligned} \psi^{(4)}(t_{n+1}, \mathbf{x}) &= T \exp \left[-i \int_{t_n}^{t_{n+1}} d\tau [\beta mc^2 - e\boldsymbol{\alpha} \cdot \mathbf{A}(\tau, \mathbf{x})] \right] \\ &\times \exp \left[-ie \int_{t_n}^{t_{n+1}} d\tau V(\tau, \mathbf{x}) \right] \psi^{(3)}(t_{n+1}, \mathbf{x}), \end{aligned} \quad (27)$$

where T is the time-ordering operator. In practice it is shown in [13], that the solution in Eq. (25) can be rewritten as:

$$\psi_h^{n+1}(\mathbf{i}) = U(\mathbf{i}) \exp \left[-i\tilde{V}_h^n(\mathbf{i}) \right] \psi_h^{n3}(\mathbf{i}) \quad (28)$$

where $U(\mathbf{i})$ is a matrix given explicitly by

$$U(\mathbf{i}) \equiv \begin{bmatrix} c(A) - i \frac{mc^2 \Delta t}{A} s(A) & 0 & i \frac{\tilde{A}_{h,z}(\mathbf{i})}{A} s(A) & \frac{[i\tilde{A}_{h,x}(\mathbf{i}) + \tilde{A}_{h,y}(\mathbf{i})] s(A)}{A} \\ 0 & c(A) - i \frac{mc^2 \Delta t}{A} s(A) & \frac{[i\tilde{A}_{h,x}(\mathbf{i}) - \tilde{A}_{h,y}(\mathbf{i})] s(A)}{A} & -i \frac{\tilde{A}_{h,z}(\mathbf{i})}{A} s(A) \\ i \frac{\tilde{A}_{h,z}(\mathbf{i})}{A} s(A) & \frac{[i\tilde{A}_{h,x}(\mathbf{i}) + \tilde{A}_{h,y}(\mathbf{i})] s(A)}{A} & c(A) + i \frac{mc^2 \Delta t}{A} s(A) & 0 \\ \frac{[i\tilde{A}_{h,x}(\mathbf{i}) - \tilde{A}_{h,y}(\mathbf{i})] s(A)}{A} & -i \frac{\tilde{A}_{h,z}(\mathbf{i})}{A} s(A) & 0 & c(A) + i \frac{mc^2 \Delta t}{A} s(A) \end{bmatrix} \quad (29)$$

and where

$$c(A) \equiv \cos(A) \quad , \quad s(A) \equiv \sin(A) \quad \text{and} \quad A \equiv \sqrt{(mc^2 \Delta t)^2 + \tilde{\mathbf{A}}_h(\mathbf{i}) \cdot \tilde{\mathbf{A}}_h(\mathbf{i})}. \quad (30)$$

This method was implemented and tested in [13, 15]. Although very robust, linear and very easy to efficiently implement in parallel (see Sections 5 and 6), the restriction on the spatial step makes it irrelevant in several physical configurations.

Notice finally, that the MC-method can also be implemented in cylindrical coordinates with for instance azimuthal symmetry, as proposed in [15]. In that case, the equation is again split in r and z , and the MC-based method can still be implemented in the z -direction, while in the r -direction a Poisson equation solver is used.

Regarding the mathematical analysis, we refer to [26], where the authors study the convergence analysis of the MC-based method coupled with a Maxwell equation Q_1 -finite element solver. For the MC-based method only, it is easy to show that it is diffusion-less in any dimension, and dispersion-less only in one dimension. The dispersion issue in higher dimension was considered in [23], where the authors used staggered grids to reduce the numerical dispersion. Let us remark that the boundary condition problem is not discussed in this paper, but we refer to [3, 4, 30], for the interested reader.

Notice that in the MC-based method, the main numerical error comes from the operator splitting in four operators. We recall that splitting $\psi_t = (A + B + C + D)\psi$ (with $\psi(\cdot, t_n) = \psi^n(\cdot)$), where A, B, C, D are four spatial differential or algebraic operators, involves the combination of four equations $\psi_t = A\psi, \psi_t = B\psi, \psi_t = C\psi, \psi_t = D\psi$. For a second order splitting, we get the following approximation:

$$\|(e^{\Delta t(A+B+C+D)} - e^{\Delta t A/4} e^{\Delta t B/2} e^{\Delta t A/4} e^{\Delta t C/2} e^{\Delta t D} e^{\Delta t C/2} e^{\Delta t A/4} e^{\Delta t B/2} e^{\Delta t A/4})\psi^n\|_{(L^2)^4} = \mathcal{O}(\Delta t^3) \quad (31)$$

In other words at each time iteration, an error in Δt^3 is produced by the above time-splitting. The splitting errors will also be produced in the pseudo-spectral methods (except for the so-called Unsplit-Pseudo-Spectral method) presented below.

3. Pseudo-spectral (PS) methods

The main issue with the finite volume/difference approach proposed in [13] for discretizing the Dirac equation on the cartesian grid, or in the z -direction in cylindrical coordinates [15], is due to a combination of a stability/ ℓ^2 -conservation constraint (CFL = 1), and a physical constraint ($\Delta t \lesssim 1/mc^2$). The consequence is that, in order to avoid numerical diffusion, we need to choose a very small space step ($\Delta z \lesssim 1/mc$). As recalled above, this restriction is only acceptable when the particle is subject to a classical electromagnetic field with very high wavenumbers, or when the interaction potential or the initial data possess small spatial scales. Although, the corresponding approach is very attractive in particular from a parallel computing viewpoint, it then suffers from a need of a very high number of gridpoints or finite volumes, or more generally of degrees of freedom (dof). In this section, we present simple alternatives based on pseudo-spectral methods, which simultaneously allow for i) preserving the high accuracy of the generalized transport equation solver, ii) the high scalability of the overall solver, and finally iii) to release the constraint on the size of the space step that is to significantly reduce of the overall number of degrees of freedom. The methodology which is now detailed is also valid in cylindrical coordinates in the z -direction

and in cartesian coordinates in the (r, θ) directions. It basically consists of solving the corresponding hyperbolic systems by using one-dimensional FFTs, generalizing the method applied now to the generic one-directional scalar transport equation in 3d, $u_t(t, x, y, z) + cu_x(t, x, y, z) = 0$. The solution to this equation can indeed be simply computed by solving the following ordinary differential equation:

$$u_t(t, \xi, y, z) + c\mathcal{F}_x^{-1}\left(\mathbf{i}\xi\mathcal{F}_x(u(t, \xi, y, z))\right) = 0$$

where \mathcal{F}_x is the Fourier transform with respect to x , and ξ denotes the co-variable of x . Naturally, this will be performed numerically using a discrete Fourier transform.

More generally, we consider for convenience the 3-dimensional system in cartesian coordinates, for $\gamma = x, y, z$ in the domain $\Omega = [-a_x, a_x] \times [-a_y, a_y] \times [-a_z, a_z]$:

$$\mathbf{i}\partial_t\psi(t) = -\mathbf{i}c\alpha_\gamma\partial_\gamma\psi(t), \quad \psi(t_n) = \psi^n, \quad t \in [t_n, t_{n+1})$$

and we use the same notation as [2]. We first diagonalize $\alpha_\gamma = \Pi_\gamma D_\gamma \Pi_\gamma^T$ where

$$D_\gamma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

The matrices Π_γ are defined as follows.

$$\Pi_x = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix}, \quad \Pi_y = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & -\mathbf{i} & -\mathbf{i} & 0 \\ 1 & 0 & 0 & 1 \\ -\mathbf{i} & 0 & 0 & \mathbf{i} \\ 0 & 1 & -1 & 0 \end{pmatrix}, \quad \Pi_z = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix}.$$

We set $\phi := \Pi_\gamma^T \psi$, which then satisfies

$$\mathbf{i}\partial_t\phi(t) = -\mathbf{i}cD_\gamma\partial_\gamma\phi(t), \quad \phi(t_n) = \Pi_\gamma^T\psi^n, \quad t \in [t_n, t_{n+1})$$

We denote the grid-point set by

$$\mathcal{D}_{N_x, N_y, N_z} = \{\mathbf{x}_{k_1, k_2, k_3} = (x_{k_1}, y_{k_2}, z_{k_3})\}_{(k_1, k_2, k_3) \in \mathcal{O}_{N_x, N_y, N_z}}$$

with

$$\mathcal{O}_{N_x, N_y, N_z} = \{(k_1, k_2, k_3) \in \mathbb{N}^3, : h = 0, \dots, N_x - 1; k = 0, \dots, N_y - 1; \ell = 0, \dots, N_z - 1\}.$$

Then, we define

$$x_{k_1+1} - x_{k_1} = h_x = 2a_x/N_x, \quad y_{k_2+1} - y_{k_2} = h_y = 2a_y/N_y, \quad z_{k_3+1} - z_{k_3} = h_z = 2a_z/N_z.$$

The corresponding discrete wavenumbers are defined by $\boldsymbol{\xi} := (\xi_p, \xi_q, \xi_r)$, where $\xi_p = p\pi/a_x$ with $p \in \{-N_x/2, \dots, N_x/2 - 1\}$, $\xi_q = q\pi/a_y$ with $q \in \{-N_y/2, \dots, N_y/2 - 1\}$ and $\xi_r =$

$r\pi/a_z$ with $r \in \{-N_z/2, \dots, N_z/2 - 1\}$. Then, we can define the partial Fourier pseudo-spectral approximations $\tilde{\phi}^{(\ell)}$ with $\ell \in \{1, 2, 3, 4\}$, in the x , y - and z -directions

$$\left\{ \begin{array}{l} \tilde{\phi}^{(\ell)}(t, x, y, z) = \frac{1}{N_x} \sum_{p=-N_x/2}^{N_x/2-1} \widehat{\tilde{\phi}_p^{(\ell)}}(t, y, z) e^{i\xi_p(x+a_x)} \\ \tilde{\phi}^{(\ell)}(t, x, y, z) = \frac{1}{N_y} \sum_{q=-N_y/2}^{N_y/2-1} \widehat{\tilde{\phi}_q^{(\ell)}}(t, x, z) e^{i\xi_q(y+a_y)} \\ \tilde{\phi}^{(\ell)}(t, x, y, z) = \frac{1}{N_z} \sum_{r=-N_z/2}^{N_z/2-1} \widehat{\tilde{\phi}_r^{(\ell)}}(t, x, y) e^{i\xi_r(z+a_z)} \end{array} \right.$$

The Fourier coefficients are defined as follows, where we denote $\tilde{\phi}_{k_1}^{(\ell)}(t, y, z) = \tilde{\phi}^{(\ell)}(t, x_{k_1}, y, z)$, $\tilde{\phi}_{k_2}^{(\ell)}(t, x, z) = \tilde{\phi}^{(\ell)}(t, x, y_{k_2}, z)$ and $\tilde{\phi}_{k_3}^{(\ell)}(t, x, y) = \tilde{\phi}^{(\ell)}(t, x, y, z_{k_3})$.

$$\left\{ \begin{array}{l} \widehat{\tilde{\phi}_p^{(\ell)}}(t, y, z) e^{i\xi_p(x+a_x)} = \sum_{k_1=0}^{N_x-1} \tilde{\phi}_{k_1}^{(\ell)}(t, y, z) e^{-i\xi_p(x_{k_1}+a_x)} \\ \widehat{\tilde{\phi}_q^{(\ell)}}(t, x, z) e^{i\xi_q(y+a_y)} = \sum_{k_2=0}^{N_y-1} \tilde{\phi}_{k_2}^{(\ell)}(t, x, z) e^{-i\xi_q(y_{k_2}+a_y)} \\ \widehat{\tilde{\phi}_r^{(\ell)}}(t, x, y) e^{i\xi_r(z+a_z)} = \sum_{k_3=0}^{N_z-1} \tilde{\phi}_{k_3}^{(\ell)}(t, x, y) e^{-i\xi_r(z_{k_3}+a_z)} \end{array} \right.$$

We finally define the approximate first-order partial derivatives

$$\left\{ \begin{array}{l} \partial_x \phi^{(\ell)}(t_n, \mathbf{x}_{k_1, k_2, k_3}) \approx \{[[\partial_x]]\tilde{\phi}^{(\ell)}\}_{k_1, k_2, k_3} := \frac{1}{N_x} \sum_{p=-N_x/2}^{N_x/2-1} i\xi_p \widehat{(\tilde{\phi}_{k_2, k_3}^{(\ell)})}_p e^{i\xi_p(x_{k_1}+a_x)} \\ \partial_y \phi^{(\ell)}(t_n, \mathbf{x}_{k_1, k_2, k_3}) \approx \{[[\partial_y]]\tilde{\phi}^{(\ell)}\}_{k_1, k_2, k_3} := \frac{1}{N_y} \sum_{q=-N_y/2}^{N_y/2-1} i\xi_q \widehat{(\tilde{\phi}_{k_1, k_3}^{(\ell)})}_q e^{i\xi_q(y_{k_2}+a_y)} \\ \partial_z \phi^{(\ell)}(t_n, \mathbf{x}_{k_1, k_2, k_3}) \approx \{[[\partial_z]]\tilde{\phi}^{(\ell)}\}_{k_1, k_2, k_3} := \frac{1}{N_z} \sum_{r=-N_z/2}^{N_z/2-1} i\xi_r \widehat{(\tilde{\phi}_{k_1, k_2}^{(\ell)})}_r e^{i\xi_r(z_{k_3}+a_z)} \end{array} \right.$$

In the following, the index h will be used (e.g. in $\phi_h^n = \left\{ \phi_{k_1, k_2, k_3}^n \right\}_{k_1, k_2, k_3}$) to denote a spectral approximation to a given wavefunction (e.g. ϕ^n). This discretization not only allows to select the spatial step as large as wanted, but it also allows to preserve the very high spatial accuracy, the parallel computing structure and the scalability of the split method developed in [13].

In the following subsections, we propose different pseudo-spectral methods by using the formalism that presented above.

3.1. Classical FFT-method

It has to be noticed that the pseudo-spectral method which will be presented below, differs from the classical Fourier-type method, which typically requires a splitting of the equation described in this subsection. The equation is first split in $[-a_x, a_x] \times [-a_y, a_y] \times [-a_z, a_z]$ using (7), (8), (9), (10). The first order splitting from t_n, t_{n+1} , reads as follows.

1. First step: integration of the source from t_n to t_{n+1} .

$$\begin{aligned} \psi_h^{n_1} &= T \exp \left[-i \int_{t_n}^{t_{n+1}} d\tau [\beta mc^2 - e\boldsymbol{\alpha} \cdot \mathbf{A}_h(\tau)] \right] \\ &\times \exp \left[-ie \int_{t_n}^{t_{n+1}} d\tau V_h(\tau) \right] \psi_h^n. \end{aligned}$$

This is performed using the time-ordering operator.

2. Second step: integration of the generalized transport equation in Fourier space. One sets $\phi_h^{n_1} := \Pi_x^T \psi_h^{n_1}$, and solve

$$\partial_t \phi + \Lambda_x \partial_x \phi = 0$$

basically by discretizing in space: $\phi(t_{n+1}, \cdot) = \mathcal{F}_x^{-1} \left(e^{-i\xi \Delta t \Lambda_x} \mathcal{F}_x(\phi) \right)$, where \mathcal{F}_x is the Fourier transform in the x -direction that is for $\ell = 1, \dots, 4$

$$\phi_h^{(\ell), n_2} = \frac{1}{N_x} \sum_{p=-N_x/2}^{N_x/2-1} \left(e^{-i\xi_p \Delta t \lambda_\ell^{(x)}} \sum_{k_1=0}^{N_x-1} \phi_{h, k_1}^{(j), n_1} e^{-i\xi_p(x_{k_1} + a_x)} \right) e^{i\xi_p(x + a_{x_j})}$$

where $\lambda_\ell^{(x)}$ is the ℓ th eigenvalues of Λ_x .

3. Similarly one computes

$$\phi_h^{(\ell), n_3} = \frac{1}{N_y} \sum_{q=-N_y/2}^{N_y/2-1} \left(e^{-i\xi_q \Delta t \lambda_\ell^{(y)}} \sum_{k_2=0}^{N_y-1} \phi_{h, k_2}^{(\ell), n_2^*} e^{-i\xi_q(y_{k_2} + a_y)} \right) e^{i\xi_q(y + a_{y_j})}$$

where $\phi_h^{n_2^*} = \Pi_y^T \Pi_x \phi_h^{n_2}$ and finally

$$\phi_h^{(\ell), n_4} = \frac{1}{N_z} \sum_{r=-N_z/2}^{N_z/2-1} \left(e^{-i\xi_r \Delta t \lambda_\ell^{(z)}} \sum_{k_3=0}^{N_z-1} \phi_{h, k_3}^{(\ell), n_3^*} e^{-i\xi_r(z_{k_3} + a_z)} \right) e^{i\xi_r(z + a_{z_j})}$$

where $\phi_h^{n_3^*} = \Pi_z^T \Pi_y \phi_h^{n_3}$.

Although this approach is classical and allows for a very accurate and fast approximation of each split equation, it also systematically necessitates a splitting of the TDDE, which to a certain extent limits its interest. In the following this method will be referred as the pseudo-spectral *FFT-method*. Alternatively, it is possible to directly apply 3d FFTs.

3.2. SPS-scheme

We here describe the numerical pseudo-spectral method for solving the *split* multi-dimensional TDDE in cartesian coordinates (3). It will be referred as the (split-pseudo-spectral) SPS-method. As proposed before, this will be performed by splitting the TDDE into four time dependent systems (7), (8), (9) and (10).

1. First step: integration of the generalized transport equation in Fourier space in the x -direction. One sets $\phi_h^n := \Pi_x^T \psi_h^n$, and the system

$$\partial_t \phi + c \Lambda_x \partial_x \phi = 0, \quad \phi(t_n, \cdot) = \Pi_x^T \psi(t_n, \cdot)$$

is approximately solved by

$$\phi_h^{n1} = \phi_h^n - c \Delta t \Lambda_x [[\partial_x]] \tilde{\phi}_h^n.$$

We then deduce $\psi_h^{n1} = \Pi_x \phi_h^{n1}$. Notice that it is possible to use a higher order discretization in time, such as

$$\begin{cases} \phi_h^{n*} &= \phi_h^n - c \Delta t \Lambda_x [[\partial_x]] \tilde{\phi}_h^n \\ \phi_h^{n1} &= \phi_h^n - c \frac{\Delta t}{2} \Lambda_x \left([[\partial_x]] \tilde{\phi}_h^n + [[\partial_x]] \tilde{\phi}_h^{n*} \right). \end{cases}$$

The corresponding operator is denoted by $\mathcal{P}_x(\Delta t)$.

2. Second step: integration of the generalized transport equation in Fourier space in the y -direction

$$\partial_t \phi + c \Lambda_y \partial_y \phi = 0.$$

We set $\phi_h^{n1*} := \Pi_y^T \psi_h^{n1}$, and we solve

$$\phi_h^{n2} = \phi_h^{n1*} - \Delta t \Lambda_y [[\partial_y]] \tilde{\phi}_h^{n1*}.$$

We then deduce $\psi_h^{n2} = \Pi_y \phi_h^{n2}$.

3. Third step: setting $\phi_h^{n2*} := \Pi_z^T \psi_h^{n2}$ we apply the same procedure in the z -direction, and deduce ψ_h^{n3} . The corresponding operator is denoted by $\mathcal{P}_z(\Delta t)$.
4. Last step: integration of the source from t_n to t_{n+1} .

$$\begin{aligned} \psi_h^{n+1} &= T \exp \left[-i \int_{t_n}^{t_{n+1}} d\tau [\beta m c^2 - e \boldsymbol{\alpha} \cdot \mathbf{A}_h(\tau)] \right] \\ &\quad \times \exp \left[-i e \int_{t_n}^{t_{n+1}} d\tau V_h(\tau) \right] \psi_h^{n3}. \end{aligned} \quad (32)$$

The corresponding operator is denoted by $\mathcal{Q}(\Delta t)$.

This scheme is fully explicit, and can be compactly expressed as

$$\psi_h^{n+1} = \mathcal{P}_x(\Delta t) \mathcal{P}_y(\Delta t) \mathcal{P}_z(\Delta t) \mathcal{Q}(\Delta t) \psi_h^n.$$

In practice, a splitting of order at least 2, should be implemented, which reads

$$\psi_h^{n+1} = \mathcal{P}_x\left(\frac{\Delta t}{4}\right) \mathcal{P}_y\left(\frac{\Delta t}{2}\right) \mathcal{P}_x\left(\frac{\Delta t}{4}\right) \mathcal{P}_y\left(\frac{\Delta t}{2}\right) \mathcal{P}_z(\Delta t) \mathcal{P}_y\left(\frac{\Delta t}{2}\right) \mathcal{P}_x\left(\frac{\Delta t}{4}\right) \mathcal{P}_y\left(\frac{\Delta t}{2}\right) \mathcal{P}_x\left(\frac{\Delta t}{4}\right) \psi_h^n.$$

Basically, compared to [13], the method of characteristics is replaced by a pseudo-spectral method. In space, the convergence is spectral (FFT). In cartesian coordinates the above methodology is applied in each split direction. In cylindrical coordinates, the Poisson solution is required in the r -direction. Another important interest of this method is that the order of accuracy of the splitting can easily be increased, unlike the MC-method which does not allow for splitting of order higher than 2.

3.3. UPS-scheme

The approach developed in this subsection is close to the SPS-method, but without splitting the system. The corresponding method will be referred as the (Unsplit Pseudo-spectral) UPS-method. By using an unsplit method, we avoid errors coming from the non-commutation of the Dirac matrices. Notice however that the mass operator contribution βmc^2 , cannot be solved analytically anymore, and then still requires a very small time step or higher order discretization in time to accurately approximate its contribution. We first present the methodology in one-dimension.

One-dimensional UPS-method.

$$\mathbf{i}\partial_t\psi(t,x) = \left(-\mathbf{i}c\sigma_x\partial_x + mc^2\sigma_z + V(t,x)I_2 - eA(t,x)\sigma_x \right)\psi(t,x) \quad (33)$$

We introduce the notations

$$\Lambda_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \Pi_x = \Pi_x^T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

where $\sigma_x = \Pi_x\Lambda_x\Pi_x^T$, $\phi = \Pi_x^T\psi$ and

$$\Gamma_x(t,x) = \Pi_x^T (mc^2\sigma_z + V(t,x)I_2) \Pi_x^T = \begin{pmatrix} V(t,x) & mc^2 \\ mc^2 & V(t,x) \end{pmatrix}.$$

The TDDE is then rewritten

$$\partial_t\phi + c\Lambda_x\partial_x\phi - \mathbf{i}eA(t,x)\Lambda_x\phi + \mathbf{i}\Gamma_x(t,x)\phi = 0. \quad (34)$$

In the UPS-formalism, we consider the semi-discrete in space system:

$$\partial_t\phi + c\Lambda_x[[\partial_x]]\phi - \mathbf{i}eA(t,x)\Lambda_x\phi + \mathbf{i}\Gamma_x(t,x)\phi = 0. \quad (35)$$

Using the same notation as above, and denoting by ψ_h the discrete solution to (33) and ϕ_h the one to (35), the numerical approximation reads

$$\phi_h^{n+1} = \phi_h^n - c\Delta t\Lambda_x[[\partial_x]]\tilde{\phi}_h^n + \mathbf{i}\Delta t(eA_h^n\Lambda_x - \Gamma_{x,h}^n)\phi_h^n.$$

Any higher order approximation of the equation is actually possible, such as the following order-2 scheme.

$$\begin{aligned} \phi_h^{n*} &= \phi_h^n - \Delta t\Lambda_x[[\partial_x]]\tilde{\phi}_h^n + \mathbf{i}\Delta t(eA_h^n\Lambda_x - \Gamma_{x,h}^n)\phi_h^n, \\ \phi_h^{n+1} &= \phi_h^n - \frac{\Delta t}{2}\left(\Lambda_x[[\partial_x]]\tilde{\phi}_h^{n*} + \Lambda_x[[\partial_x]]\tilde{\phi}_h^n\right) - \mathbf{i}\frac{\Delta t}{2}\left((eA_h^n\Lambda_x - \Gamma_{x,h}^n)\phi_h^n + (eA_h^{n*}\Lambda_x - \Gamma_{x,h}^{n*})\phi_h^{n*}\right). \end{aligned}$$

We now extend this approach in the multi-dimensional case.

Multi-dimensional UPS-method. We now solve from t_n to t_{n+1}

$$\mathbf{i}\partial_t\psi(t,\mathbf{x}) = \left\{ \alpha_x \left[-\mathbf{i}c\partial_x - eA_x(t,\mathbf{x}) \right] + \alpha_y \left[\mathbf{i}c\partial_y - eA_y(t,\mathbf{x}) \right] + \alpha_z \left[-\mathbf{i}c\partial_z - eA_z(t,\mathbf{x}) \right] + \beta mc^2 + \mathbb{I}_4 V(t,\mathbf{x}) \right\} \psi(t,\mathbf{x}).$$

Assuming that ψ_h is known at time t_n , the semi-discrete in space scheme reads in $[-a_x, a_x] \times [-a_y, a_y] \times [-a_z, a_z]$

$$\begin{aligned} \mathbf{i}\partial_t\psi_h = & \left\{ \alpha_x \left[-\mathbf{i}c[[\partial_x]]\tilde{\psi}_h - eA_{x,h}(t)\psi_h \right] + \alpha_y \left[-\mathbf{i}c[[\partial_y]]\tilde{\psi}_h - eA_{y,h}(t)\psi_h \right] \right. \\ & \left. + \alpha_z \left[-\mathbf{i}c[[\partial_z]]\tilde{\psi}_h - eA_{z,h}(t)\psi_h \right] + \beta mc^2 + \mathbb{I}_4 V_h(t)\psi_h \right\} \end{aligned} \quad (36)$$

which is nothing but a system of differential equations, which can be solved using an explicit or implicit high order scheme. Assuming that ψ_h^n is known, a first order explicit scheme in time reads

$$\begin{aligned} \mathbf{i}\psi_h^{n+1} = & \mathbf{i}\psi_h^n + \Delta t \left\{ \alpha_x \left[-\mathbf{i}c[[\partial_x]]\tilde{\psi}_h^n - eA_{x,h}^n\psi_h^n \right] + \alpha_y \left[-\mathbf{i}c[[\partial_y]]\tilde{\psi}_h^n - eA_{y,h}^n\psi_h^n \right] \right. \\ & \left. + \alpha_z \left[-\mathbf{i}c[[\partial_z]]\tilde{\psi}_h^n - eA_{z,h}^n\psi_h^n \right] + \beta mc^2 + \mathbb{I}_4 V_h^n\psi_h^n \right\} \end{aligned}$$

where $A_{x,h}^n$ stands for $A_{x,h}(t_n)$, etc. However such a scheme would not ensure stability and accuracy in principle. In practice higher order scheme will be used. An alternative from (36) is then to formally integrate analytically this expression by expliciting the $[[\partial_\gamma]]\tilde{\psi}$ terms ($\gamma = x, y, z$). Say from time t_{n+1} to t_n :

$$\begin{aligned} \psi_h(t_{n+1}) = & \exp \left\{ \mathbf{i}e \left[\alpha_x \int_{t_n}^{t_{n+1}} A_{x,h}(s)ds + \alpha_y \int_{t_n}^{t_{n+1}} A_{y,h}(s)ds + \alpha_z \int_{t_n}^{t_{n+1}} A_{z,h}(s)ds \right. \right. \\ & \left. \left. - \mathbf{i}mc^2(t_{n+1} - t_n)\beta - \mathbf{i}\mathbb{I}_4 \int_{t_n}^{t_{n+1}} V_h(s)ds \right] \right\} \\ & \times \left\{ c(t_{n+1} - t_n) \left[\alpha_x [[\partial_x]]\tilde{\psi}_h(t_n) + \alpha_y [[\partial_y]]\tilde{\psi}_h(t_n) + \alpha_z [[\partial_z]]\tilde{\psi}_h(t_n) \right] \times \right. \\ & \left. \int_{t_n}^{t_{n+1}} \exp \left\{ -\mathbf{i}e \left[\alpha_x \int_{t_n}^s A_{x,h}(\tau)d\tau + \alpha_y \int_{t_n}^s A_{y,h}(\tau)d\tau + \alpha_z \int_{t_n}^s A_{z,h}(\tau)d\tau \right. \right. \right. \\ & \left. \left. \left. - \mathbf{i}mc^2(s - t_n)\beta - \mathbf{i}\mathbb{I}_4 \int_{t_n}^s V_h(\tau)d\tau \right] \right\} \right\} \end{aligned}$$

which then requires high order approximations of time integrals.

4. Parallel computing

The parallel computation of the generalized transport systems which is detailed in [13, 15], is highly scalable thanks to a spatial domain decomposition (in any direction x, y, z). At each subdomain interface, the Dirac equation is diagonalized in the direction orthogonal to the subdomain interface. We then impose Dirichlet boundary conditions at these interfaces and on the components corresponding to the outgoing characteristics, and these conditions are sent by message passing. The spatial domain can be decomposed in layers in one specific directions, or alternatively in small cubic subdomains.

It is well known that the parallel implementation of the FFT suffers from a relative lack of scalability. However the pseudo-spectral methods presented in this paper allow for using the high accuracy and relative efficiency of the parallel one-dimensional FFTs in the z -direction, and also keeping the nice scalability properties of [13, 15] in the x - and y -directions where sequential FFTs are performed. More specifically in 3 dimensions, we proceed by alternating the directions. We first decompose the domain by layers in the z -directions (see Fig. 1). Then

- We successively perform the evolution (FFT) sequentially in the x - and y -directions, by layer in the z -direction. Each processor manages one layer in z . We expect a perfect scaling for this step, as it does not require any transmission from one node to another.
- For all x and y , we perform the evolution in the z -direction using the parallel FFT (`fftw`). The performance of this step is fully dependent on the parallelization of the one-dimensional FFT.

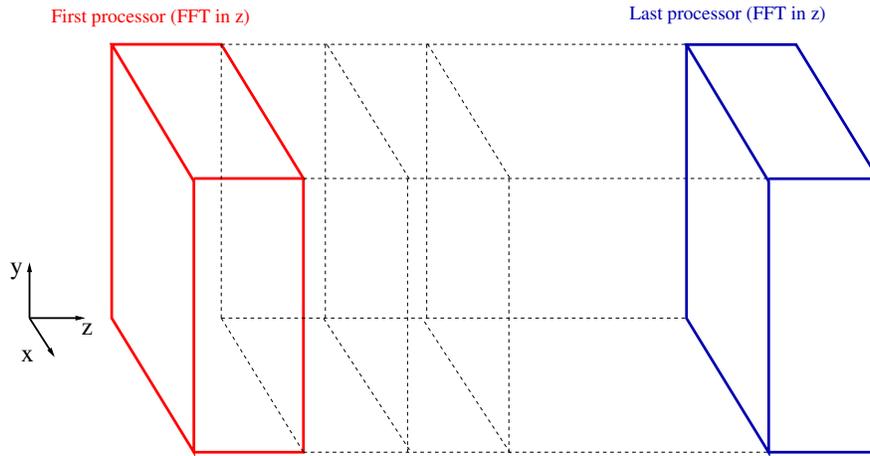


Figure 1: Parallelization principle of the FFT in the z -direction.

The main difference with the MC-based methods proposed in [13, 15] is that we do not have the space-step restriction due to the CFL-number, but keeping simultaneously a high accuracy and high scalability. Stability and convergence properties are also preserved. As a consequence, the number of degrees of freedom in space can largely be reduced. Notice that the parallelization approach is valid for the classical FFT-method, as well as for the SP-methods presented in this paper.

The computational complexity (CC) is relatively simple to established. Assume that the real-space grid is composed by $N_x \times N_y \times N_z$ volumes. We denote by N_{T_f} the number of time steps to reach a final physical time T_f . The computation of the MC-solver for the split generalized transport equation (22) (resp. (23), resp. (24)) requires $c_x N_{T_f} N_x N_y N_z$ operations (resp. $c_y N_{T_f} N_x N_y N_z$, resp. $c_z N_{T_f} N_x N_y N_z$), where the integers $c_x \approx c_y \approx c_z$ are typically integers smaller than 10. The solution to the differential system (27) also requires

$c_{T_f} N_{T_f} N_x N_y N_z$ operations, where c_{T_f} is dependent on the order of the numerical integrator and of the time-ordering operator. In fine, the total complexity C_{MC} , is then given by

$$C_{MC} = \mathcal{O}(N_{T_f} N_x N_y N_z)$$

with a small prefactor. In addition this method is pleasingly parallel as discussed in [13], and efficiency greater than 1 can even be observed due to cache effects. By comparison, the CC of the UPS-method, C_{UPS} , and for the SPS-method, C_{SPS} , are

$$C_{UPS,SPS} = \mathcal{O}(N_{T_f} N_x N_y N_z \log(N_x N_y N_z)).$$

Compared to the MC-method, the prefactor is however much higher. The SPS-method also allows for a very efficient parallel computing of the “many” one-dimensional FFTs. Indeed, the UPS/SPS require the computation of one-dimensional FFTs of three-dimensional wavefunctions. Two of three real variables are then fixed, while the FFT is applied in the third one. Notice however that multi-dimensional FFTs can be very efficiently computed in parallel, while one-dimensional parallel FFT versions may suffer from a loss of efficiency. We refer to [5, 6, 22] for the presentation and mathematical analysis of pseudo-spectral methods in different regimes. In the relativistic regime, we also refer to [7, 29, 9] for efficient pseudo-spectral methods.

In the following, the first three one-dimensional tests are performed by using `matlab`. The fourth one-dimensional test, as well as all the multi-dimensional tests are performed by using a `C++`-code with `MPI`-library. The parallel tests are realized on the computer `mammouth-parallel II` from Compute Canada, with 30984 cores, 24 cores/node, 32 G/node & FAT nodes, opteron 2.1 GHz.

5. One-dimensional experiments

We now compare the performance of the pseudo-spectral methods with the MC-based method. It is important to recall that pseudo-spectral methods are expected to be more attractive than the MC-based method, only if the space variations of either the electromagnetic field (EM), the interaction potential, or the initial data are much larger than $1/mc$. Typically, denoting by k_∞ the largest wavenumber of the EM and of V_{mic} , the space steps for respectively the PS-methods and MC-method, are selected as follows:

$$\Delta x_{\text{PS}} \lesssim k_\infty/M, \quad \Delta x_{\text{MC}} \lesssim \min(k_\infty/M, 1/mc)$$

where M is a non-zero integer. Notice that the interaction of the intense EM with quantum particle generates very high frequency photons, and may potentially necessitate even finer spatial resolution.

More specifically, we compare the efficiency of the FFT-, SPS-, UPS- and MC-schemes which are presented in Section 3 and Appendix A. Recall that the strength of the UPS-scheme compared to the SPS- and FFT-scheme is that it does not require the split of the Dirac

Hamiltonian. On the other side, the FFT-, SPS- and UPS-schemes do not require a strict condition on the space step, unlike the MC-based scheme ($\Delta x = c\Delta t$). In the following, the external field the particle is subject to, is assumed space-independent, and is given by

$$A(t) = A_0 \sin \exp\left(-\alpha \frac{(t - T_f)^2}{4}\right) \sin(2\mathcal{C}\pi t/T_f). \quad (37)$$

In (37), A_0 denotes the maximal amplitude to the field, T_f the pulse duration, \mathcal{C} the number of cycles and α a real parameter characterizing the pulse envelope.

5.1. Test 1.

In this first test, we compare the MC-based scheme on a very fine mesh ($N^{(\text{mc})} = 54799$), with the pseudo-spectral methods on coarser meshes. In (37), we select $A_0 = -100$, the number of cycles $\mathcal{C} = 6$, $T_f = 5 \times 10^2/137^2$, and $\alpha = 10^7$. The field is represented in Fig. 2, as function of time in $[0, T_f]$. The initial wavefunction is given by

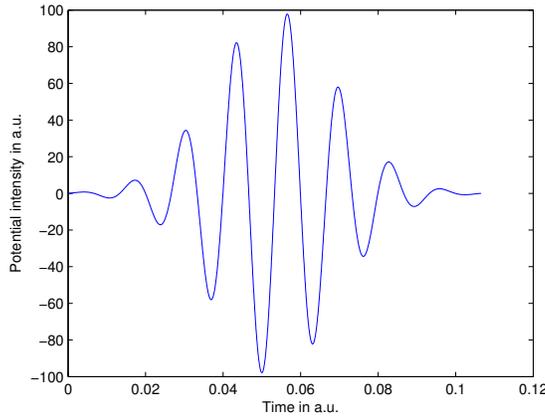


Figure 2: Electric potentials $A(t)$.

$$(\psi^{(1)}(0, x), \psi^{(2)}(0, x))^T = (e^{ik_0x - x^2/10}, 0)^T$$

where the wavenumber $k_0 = 4$. The scalar potential is hence null and, the nuclear potential is of Coulomb type with charge $Z = 1000$, and softcore parameter $\varepsilon = 10^{-1}$:

$$V_{\text{nuc.}}(x) = -\frac{Z}{\sqrt{x^2 + \varepsilon}}.$$

That is we have $V(t, x) = V_{\text{nuc.}}(x)$. The other physical data are as follows (in atomic units): $c = 137$, $m = 1$ and the computational domain is given by $(-a, a)$, with $a = 20$ and $\Delta t = 10^{-1}/mc^2 = 10^{-1}/137^2$. We report in Table 1, the CPU-time for solving the above problem, using the MC-based, FFT-, SPS-, UPS-schemes, and with a variable number of dof.

# dof	CPU-time MC	CPU-time SPS	CPU-time UPS	CPU-time FFT
54799	19.46	-	-	-
5481	-	2.86	3.87	2.15
1097	-	0.63	0.81	0.48
275	-	0.13	0.19	0.10

Table 1: Computational time in second: MC-, SPS-, UPS-, FFT-schemes.

The MC-based scheme for a fixed Δt , requires $\Delta x = c\Delta t = 1/mc$, that is $N^{(\text{mc})} \approx 2amc$ dof. Unlike the MC-scheme, the pseudo-spectral methods allowed for choosing space steps (much) larger than $c\Delta t$, then at a cheaper computational complexity, while keeping a good accuracy. For instance, it is reported in Figs. 3, 4, the bispinor components at final time, using the MC-based scheme with $N^{(\text{mc})} = 54799$ and the UPS-scheme with $N^{(\text{ups})} = 5481, 1097, 275$. We observe that with much less dof (1097 vs 54799), we still keep a good accuracy.

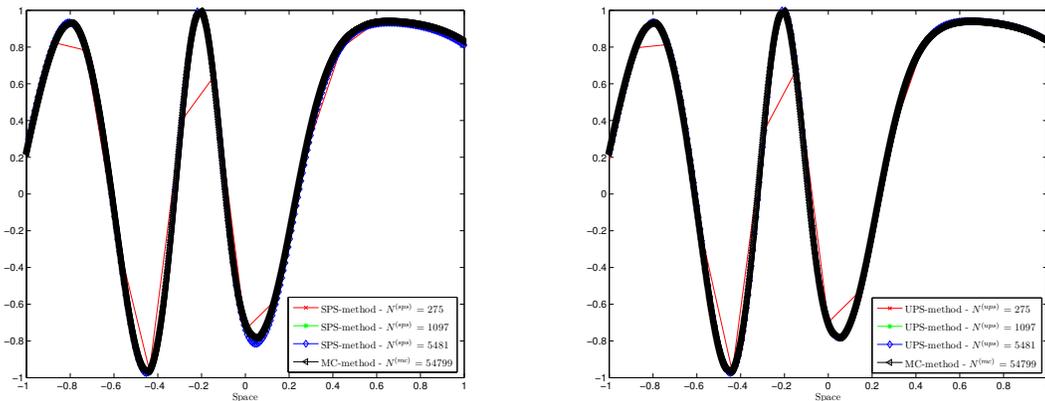


Figure 3: Comparison: bispinor first component at final time T_f , computed with MC-based scheme with $N^{(\text{mc})} = 54799$ gridpoints, and SPS-method (Left) and UPS-method (Right) with $N^{(\text{ps})} = 275, 1097, 5481$ gridpoints.

Notice that the simple FFT-method is the most efficient method, but requires an operator splitting unlike the UPS-method.

5.2. Test 2.

In the following, we still consider a simple test with small scales in space (and time). The use of the MC-based is shown to be more competitive than pseudo-spectral methods (more specifically the so-called FFT-method), as it is required to select Δx at least as small as $c\Delta t$ for an accurate spatial resolution. The EM is given by (37) where $A_0 = -500$, the number of cycles $\mathcal{C} = 6$, $T_f = 5 \times 10^2/137^2$, and $\alpha = 10^7$. The initial wavefunction is given by

$$(\psi^{(1)}(0, x), \psi^{(2)}(0, x))^T = (e^{ik_0x - x^2/2}, 0)^T$$

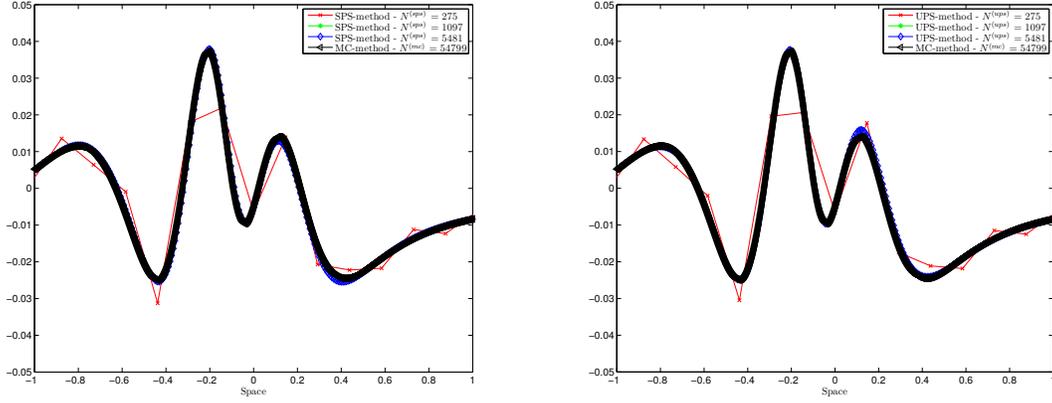


Figure 4: Comparison: bispinor second component at final time T_f , computed with MC-based scheme with $N^{(\text{mc})} = 54799$ gridpoints, and SPS-method (Left) and UPS-method (Right) with $N^{(\text{ps})} = 275, 1097, 5481$ gridpoints.

where the wavenumber $k_0 = 4$. The scalar potential is hence null and, the nuclear potential is a combination of three nuclear potentials with $Z = 500$, and a softcore parameter is $\varepsilon = 10^{-6}$

$$V_{\text{nuc.}}(x) = -\frac{2Z}{\sqrt{(x - 2.01)^2 + \varepsilon}} - \frac{Z}{\sqrt{(x - 2)^2 + \varepsilon}} - \frac{3Z}{\sqrt{(x - 1.99)^2 + \varepsilon}}.$$

This potential is represented in Fig. 5.

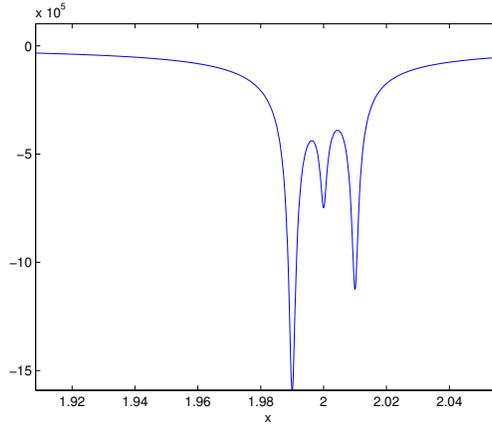


Figure 5: Interaction potential $V_{\text{nuc.}}$.

The other physical data are as follows (in atomic units): $c = 137$, $m = 1$, and the computational domain is given by $(-5, 5)$, $\Delta t = 10^{-2}/mc^2 = 1/137^2$. We compare the MC-based solution with $N^{(\text{mc})} = 1.37 \times 10^5$ grid points and the FFT-method with respectively

$N^{(\text{mc})} = 1.37 \times 10^5$ and $N^{(\text{fft})} = 1.37 \times 10^4$. More specifically, in Fig. 6, we represent the first and second components of the MC- ($N^{(\text{mc})} = 1.37 \times 10^5$) and FFT-solutions with $N^{(\text{fft})} = 1.37 \times 10^4$, close to the nuclei (Top-left), (Top-right), as well as the spatial spectrum of $\psi_1(T_f, \cdot)$ (i.e. $\{(k, |\mathcal{F}_x \psi_1(T_f, k)|^2)\}$ in logscale). Although, the computational time is much smaller using the FFT-method (5.8 seconds versus 48.7 seconds for the MC-based scheme) the accuracy in space is insufficient to capture the very small spatial scales in $V_{\text{nuc.}}$. In comparison, if we also take $N^{(\text{fft})} = 1.37 \times 10^5$ for the FFT-method, the two solutions are comparable as observed in Figs. 7, as well as the spatial spectrum of $\psi_1(T_f, \cdot)$. However the MC-method is now shown to be more efficient: 48.7 seconds versus 68.5 seconds for the FFT-method.

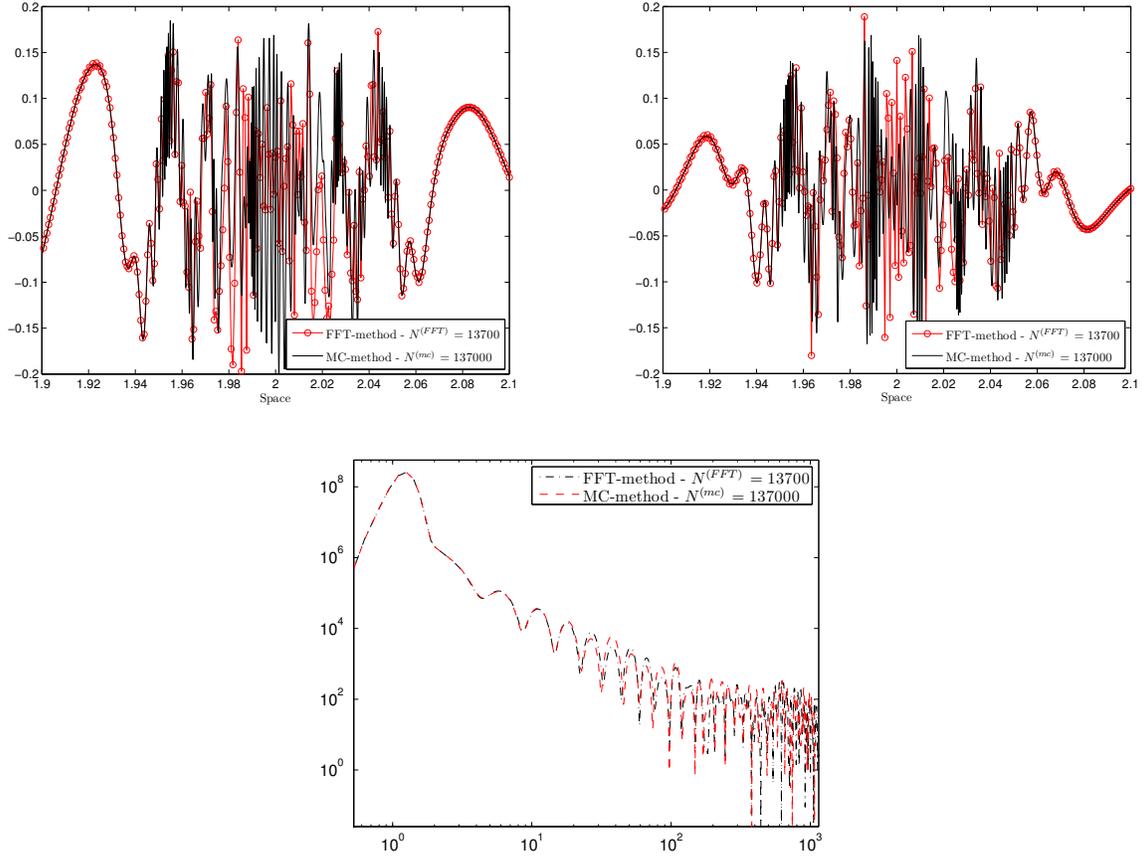


Figure 6: Comparison: bispinor first (Top-left) and second (Top-right) components at final time T_f , computed with MC-based scheme with $N^{(\text{mc})} = 1.37 \times 10^5$ gridpoints, and FFT-method with $N^{(\text{fft})} = 1.37 \times 10^4$ gridpoints. (Bottom) Spatial spectrum of $\psi_1(T_f, \cdot)$ in logscale: $\{(k, |\mathcal{F}_x \psi_1(T_f, k)|^2)\}$.

Although simple, this test illustrates that the MC-based method is sequentially more competitive than the FFT-method for multiscale problems in space.

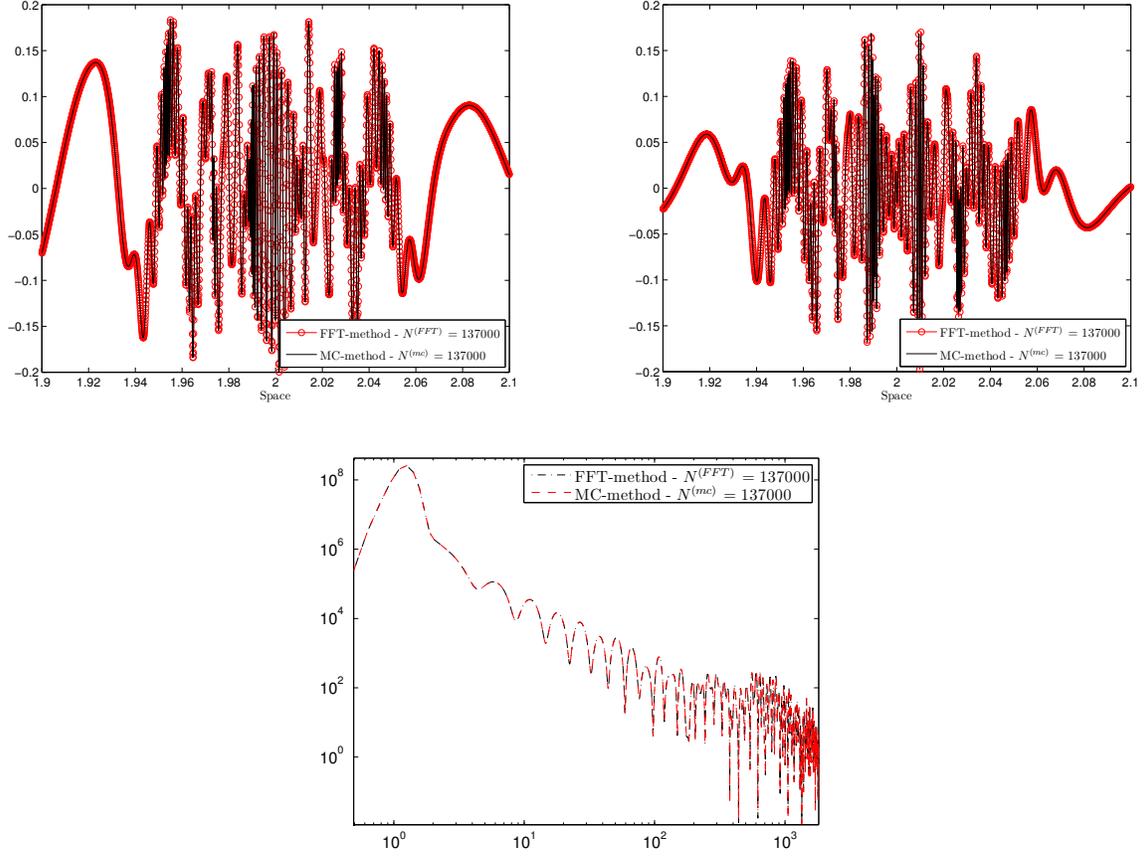


Figure 7: Comparison: bispinor first (Top-left) and second (Top-right) components at final time T_f , computed with MC-based scheme with $N^{(mc)} = 1.37 \times 10^5$ gridpoints, and FFT-method with $N^{(fft)} = 1.37 \times 10^5$ gridpoints. (Right) Spatial spectrum of $\psi_1(T_f, \cdot)$ in logscale: $\{(k, |\mathcal{F}_x \psi_1(T_f, k)|^2)\}$.

5.3. Test 3.

We again compare the performance of the MC-based scheme with the pseudo-spectral scheme. The EM is assumed space-independent and given by (37), with $A_0 = -100$, the number of cycles $\mathcal{C} = 8$, $T_f = 2 \times 10^3 / 137^2$, and $\alpha = 2 \times 10^3$. The initial wavefunction is given by

$$(\psi^{(1)}(0, x), \psi^{(2)}(0, x))^T = (e^{ik_0 x - x^2/2}, 0)^T$$

where the wavenumber $k_0 = 5$. The scalar potential is hence null and, the nuclear potential is of Coulomb type with charge $Z = 100$, and softcore parameter $\varepsilon = 10^{-1}$

$$V_{\text{nuc.}}(x) = -\frac{Z}{\sqrt{x^2 + \varepsilon}}.$$

Therefore, one gets $V(t, x) = V_{\text{nuc.}}(x)$. The other physical data are as follows (in atomic units): $c = 137$, $m = 1$, and the computational domain is given by $(-20, 20)$ with $\Delta t = 10^{-1}/mc^2 = 1/137^2$. For both schemes, a second-order splitting (Strang) scheme is used to approximate the TDDE. When using the SPS-scheme, we take $\Delta x = c\Delta t \approx 7.3 \times 10^{-4}$ (resp. $c\sqrt{c}\Delta t/2 \approx 1.46 \times 10^{-3}$, $c\sqrt{c}\Delta t/2 \approx 4.27 \times 10^{-3}$, $c\sqrt{c}\Delta t \approx 8.54 \times 10^{-3}$, $2c\sqrt{c}\Delta t \approx 1.71 \times 10^{-2}$, $5c\sqrt{c}\Delta t \approx 4.27 \times 10^{-2}$, $c^2\Delta t \approx 10^{-1}$) the corresponding number of spatial grid-points is given by $N^{(\text{sps})} = 54800$ (resp. 27400, 9364, 4682, 2341, 937, 401). For the *MC-based scheme*, we take $\Delta x = c\Delta t \approx 7.3 \times 10^{-3}$ and the number of spatial grid-points is fixed and given by $N^{(\text{mc})} = 54800$. Although the MC-based scheme is much simpler to implement, its computational cost is naturally much higher whenever $N^{(\text{mc})} \gg N^{(\text{sps})}$.

We first report in Fig. 8 the bispinor components at final time T_f by using the MC-based method with $N^{(\text{mc})}$ grid-points and the SPS-method with $N^{(\text{sps})} = 937$ and 401 gridpoints.

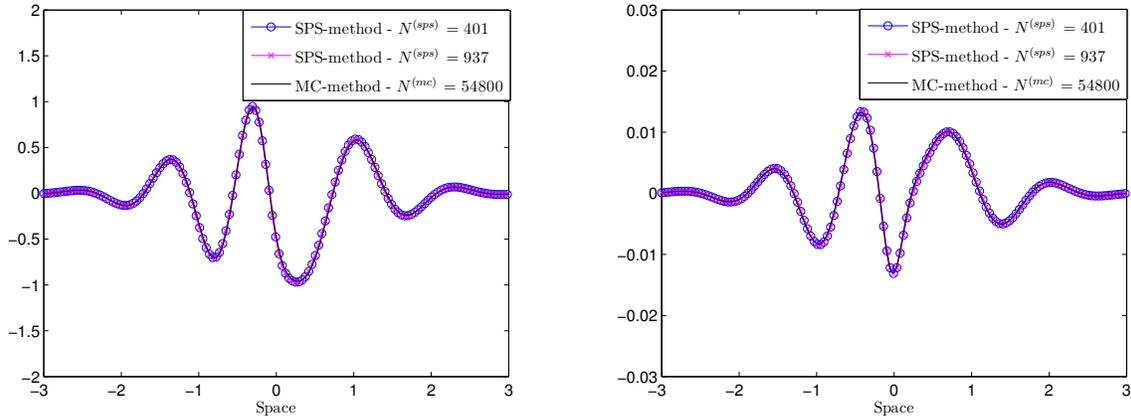


Figure 8: Comparison: bispinor components at final time T_f , computed with MC-based scheme with $N^{(\text{mc})} = 54800$ gridpoints, and SPS-method with $N^{(\text{sps})} = 401$ and 937 gridpoints.

In Table 2, we report the CPU-time to solve the TDDE using the MC-based method on $N^{(\text{mc})} = 54800$ grid-points ($\Delta x = c\Delta t = 1/mc$), and the SPS-method using $N^{(\text{sps})} = 54800$ ($\Delta x = c\Delta t$), $N^{(\text{sps})} = 9364$ ($\Delta x = \sqrt{c}\Delta t/2 = 1/2\sqrt{cm}$), $N^{(\text{sps})} = 4682$ ($\Delta x = c\sqrt{c}\Delta t = 1/m\sqrt{c}$), $N^{(\text{sps})} = 2341$ ($\Delta x = 2c\sqrt{c}\Delta t = 1/m\sqrt{c}$), $N^{(\text{sps})} = 937$ ($\Delta x = 5\sqrt{c}\Delta t = 5/\sqrt{cm}$), and $N^{(\text{sps})} = 401$ ($\Delta x = c^2\Delta t = 1/m$). Recall again that the CFL= 1 condition constraining the MC-method, no more occurs with the SPS-method which allows for choosing space steps much larger than $c\Delta t$.

5.4. Test 4.

The following test is devoted to the analysis of the scalability of the MC-based and FFT-methods, still in one-dimension. The domain we consider is $(-64, 64)$. The initial density ($\rho_0(\cdot) = \sum_{i=1}^4 |\psi_4(0, \cdot)|^2$) and the interaction potential (with 3 nuclei) are represented in Figs. 9 (Top-left) and (Top-right). The nuclei are respectively located in $z = -10, 0$, and 10 , and their respective charge is $Z_1 = 30$, $Z_2 = 60$ and $Z_3 = 70$. Softcore constants are used to

# dof	CPU-time MC	CPU-time SPS
54800	39.52	87.63
27400	-	42.1
9364	-	13.62
4682	-	6.71
2341	-	3.35
937	-	1.34
401	-	0.58

Table 2: Computational time in second: MC- and SPS-schemes (MC-scheme with 54800 points) and SPS-solution.

regularize the potential singularities, and are respectively equal to 5.0, 0.791 and 0.071. The time step is fixed to 1.78158×10^{-6} and the number of degrees of freedom is $N = 524288$. The final physical time is $T_f = 1.78158 \times 10^{-4}$. We also report the density at final time in the global domain in Figs. 9 (Bottom-left) and (Bottom-right). We compare the scalability of the MC-based and FFT-method for 2^{2p} processors with $p = 0, \dots, 3$. The MC-based scheme is parallelized by domain decomposition thanks to the algorithm described in [13], and shortly recalled in Section 4. The efficiency of the parallelization for this method comes from i) the hyperbolicity, ii) the linearity of the Dirac equation and iii) the use of alternate direction splitting. The FFT-method which is used here, is based on the standard, and celebrated parallel version of `fftw`. In Fig. 10, we report the CPU-times for both the FFT- and MC-based methods. More specifically, we provide i) the efficiency T_1/pT_p (where T_p denotes the CPU time for solving the equation on p processors with $p = 1, 4, 16, 64$), where p denotes the number of processors in Fig. 10 (Top-left), and ii) the CPU-time in logscale as a function of the number of processors (Top-right).

We previously noticed on low dimensional problems, that for a fixed number of dof, the MC-based method was more efficient than the FFT-method. It is still observed on this higher dimensional problem that the MC-based method has a much better speed-up than the FFT-method, in particular when increasing the number of processors with a fixed number of dof.

Unlike the MC-based method, the FFT-method permits for fixed Δt to select much larger spatial steps while keeping a very good accuracy in space. To illustrate this fact, we report on Fig. 10 (Bottom), and for 1 processor the CPU-time with the FFT-method on the same problem as above, but with a total number of dof: $524288/2^{2p}$, with $p = 0, \dots, 3$. The MC-based method maintains an excellent efficiency up to 64 processors, unlike the FFT-method for which a strong discrepancy is observed beyond 16 processors.

6. Multi-dimensional experiments

In this section, we present large scale experiments in 2-d and 3-d in order to compare the efficiency of the MC-based and the pseudo-spectral FFT-methods. The latter is the most efficient and simple pseudo-spectral method. Due to the dimension of the problem, the

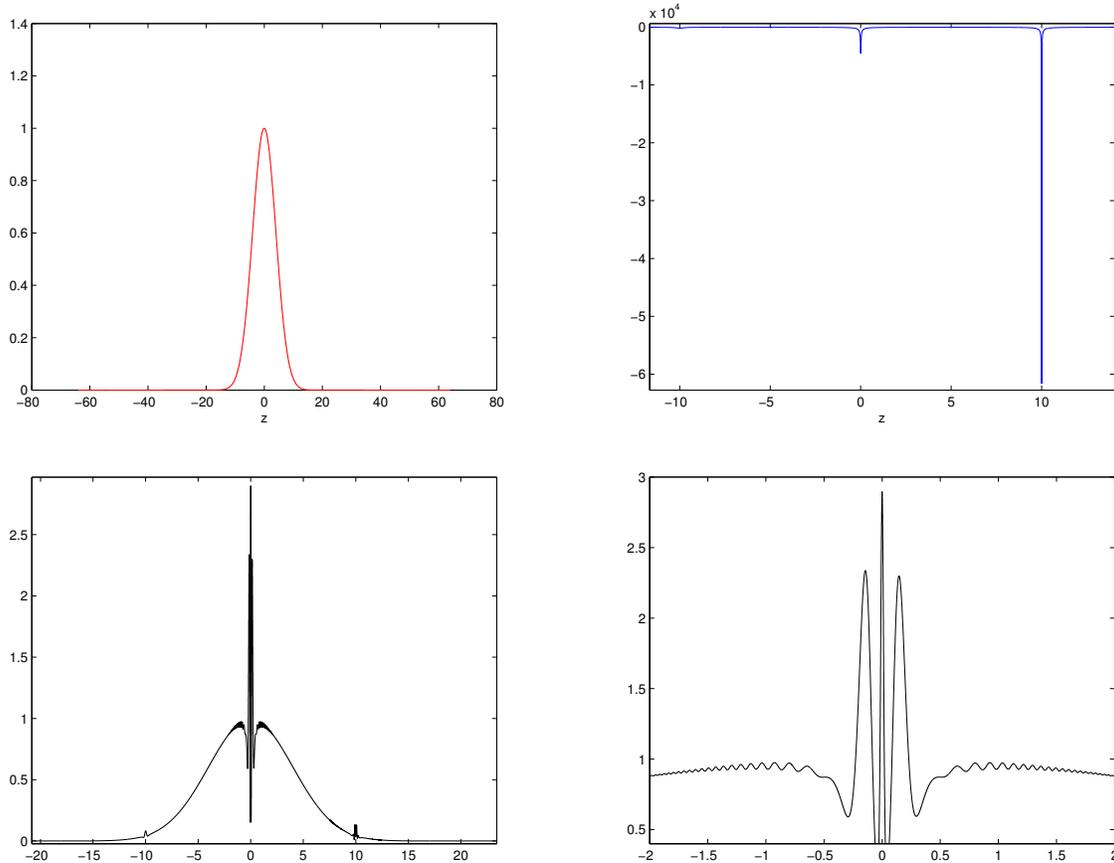


Figure 9: (Top-left) Initial spinor density. (Top-right) Interaction potential (Bottom-left) Spinor density at final time. (Bottom-right) Spinor density at final time: zoom in the central nucleus.

methods are implemented in parallel following the approach described in Section 4. The MC-based algorithm is parallelized by using a simple domain decomposition strategy: the whole domain is divided into subdomains and the mesh data in each of these subdomains is managed to a different process. The communication between the subdomains which is required in the computation of the solution in the elements close to the subdomain boundaries is performed by using the Message Passing Interface (MPI) library.

6.1. Test 1.

This test is dedicated to the two-dimensional evolution in the (x, z) -plane of a wavepacket with wavenumbers $k_x = k_z = 5$, angular momentum $j_z = 0.5$, and its interaction with 3 atoms of equal charge $Z_1 = Z_2 = Z_3 = 20$, with softcore constant equal to 5. The nuclei are respectively located in $(10, -10)$, $(0, 0)$ and $(-10, 10)$. We report the initial density $\rho_0(\cdot, \cdot) = \sum_{i=1}^4 |\psi_i(0, \cdot, \cdot)|^2$, and potential in Fig. 11 (Left) and (Right).

The overall computational domain is $(-64, 64)^2$, the time step $\Delta t = 1/mc^2 = 2.28042 \times$

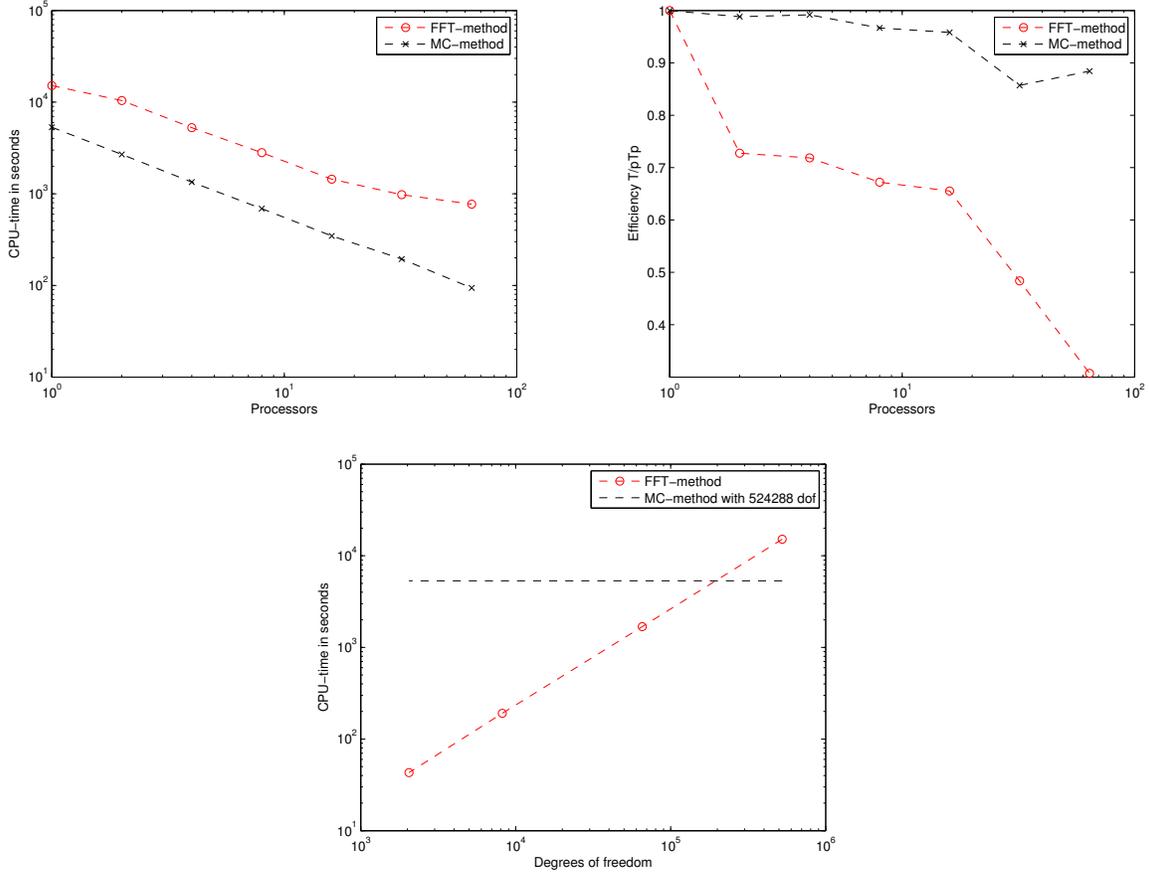


Figure 10: (Top-left) Efficiency T/pT_p , (Top-right) CPU-time as a function of the number of processors ($p = 1, 4, 16, 64$), (Bottom) CPU time for FFT-method with $N_z = 524288/2^{2p}$, with $p = 0, \dots, 3$, and MC-based method with 524288 dof.

10^{-4} , and the final physical time $T_f = 0.912169$ (corresponding to 4000 iterations). The number of degrees of freedom is $N_x \times N_z = 4096 \times 4096$. We report in Fig. 12, the overall density at final time. The main purpose of this test is to illustrate the efficiency of the MC-based method. We denote by p_x (resp. p_z) the number of processors in the x -direction (resp. z -direction), and by p the total number of processors with $p = p_x \times p_z$. For the MC-based method, we respectively present some efficiency results in the case where i) $p = p_x \times p_z = p_z$ (that is parallelism only in the z -direction), and in the case ii), where $p_x = p_z = \sqrt{p}$. In both cases, we will use up to $p = 256$ processors. The FFT-method is parallelized in the (x, z) -plane, by decomposing the domain only in layers in the z -direction ($p = p_z$). The main issue regarding the FFT-method is the use of parallel one-dimensional FFTs in the z -direction. Its poor efficiency when the number of dof is relatively “small” deteriorates the overall efficiency of the method despite a perfect scaling in the x -direction (as the one-dimensional FFTs in the x -direction are computed independently on different processors). As $N_z = 4096$ is too

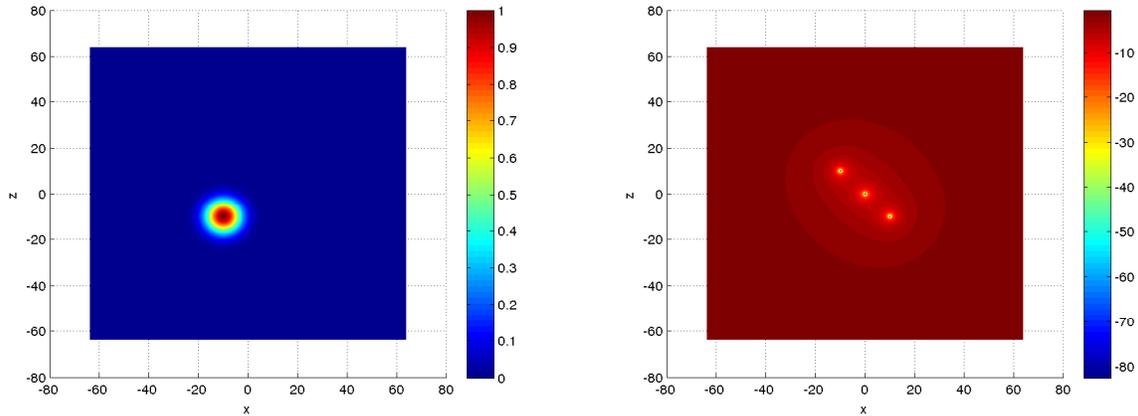


Figure 11: (Left) Density of the initial spinor. (Right) Interaction potential.

small for efficient parallelization of the one-dimensional FFT, we only report the efficiency of the FFT-method for 1 to 4 processors. Even with this small number of processors, we indeed observe already a strong discrepancy of the efficiency. In comparison, the MC-based method shows an excellent efficiency (even higher than 1 due to cache-effects) up to 64 processors for both decompositions i) z -direction ($p = p_z$) and in x, z ($p_x = p_z = \sqrt{p}$).

We observe that with both types of decomposition and parallelism, the efficiency is relatively identical. However, it is interesting to notice that at 256 processors, the second approach ($p_x = p_z = \sqrt{p}$) has a better efficiency than the first one ($p = p_z$). In the following, we will only use a decomposition by layers in the z -direction.

6.2. Test 2.

In the following test, we still propose a two-dimensional simulation in the (x, z) -plane, in order to illustrate that due to the parallel one-dimensional FFT in the z -direction, the relative loss of efficiency of the method, can actually be reduced if N_z is taken larger. A wavepacket with wavenumbers $k_x = k_z = 5$, angular momentum $j_z = 0.5$, is interacting with 2 atoms of equal charge $Z_1 = Z_2 = 50$, and with softcore constants equal to 1. The nuclei are respectively located in $(-1, -10)$, and $(-1, 10)$. We report the initial density and potential in Fig. 13 (Left) and (Right).

The overall computational domain is $(-4, 4) \times (-64, 64)$, the time step is $\Delta t = 5.70106 \times 10^{-5}$, and the physical times $T = 0.05521$ and $T_f = 0.1142$ (corresponding to 1000 and 2000 iterations). The number of degrees of freedom is $N_x \times N_z = 1024 \times 16384$ (resp. $N_x \times N_z = 256 \times 4096$). We report in Fig. 14, the overall density at final time. The MC-based method is parallelized in the z -direction. We observe in Fig. 15 that the efficiency of the FFT-method is indeed slightly improved when a larger N_z is selected (4096 vs 16384). This is due to an improvement of the efficiency of the parallel one-dimensional FFT, for larger numbers of dof. As mentioned several times along this paper, although at fixed number of dof (that is fixed space step), the FFT-method is much less efficient (sequentially and in

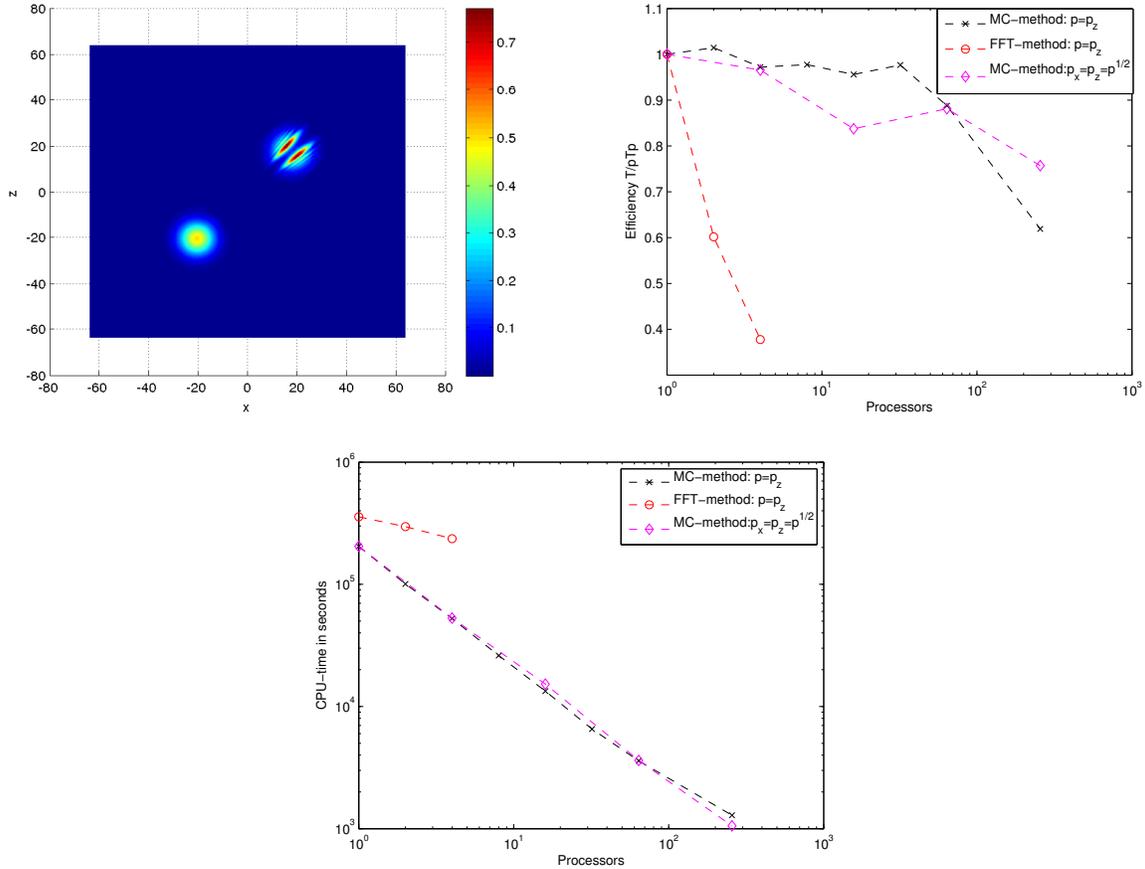


Figure 12: (Top-left) Density at final time: $T_f = 0.912169$. (Top-right) Efficiency T_1/pT_p for the MC- and FFT-methods. (Bottom) CPU-time as function of number of processors ($p = 1, 2, 4, 8, 16, 64, 256$ processors).

parallel) than the MC-based method, it allows to select much larger space steps (less dof for given spatial domain, for fixed imposed Δt) but still keeping a good accuracy, *except for multiscale problems*. In the latter case, a very fine resolution in space is indeed required, making the MC-method more attractive. This remark is also illustrated in Fig. 15, where it is observed that on a coarse mesh ($N_x = 256, N_z = 4096$) and still taking $\Delta t = 1/mc^2$, the FFT-method is more efficient than the MC-method when $N_x = 1024, N_z = 16384$, at least for a small number of processors.

6.3. Test 3.

In this last test, we consider the evolution of a wavepacket subject to an external dynamic and static fields in 3d. We again compare the performance of the MC-based and FFT-methods. The computational domain is $(-8, 8) \times (-8, 8) \times (-512, 512)$, and the time

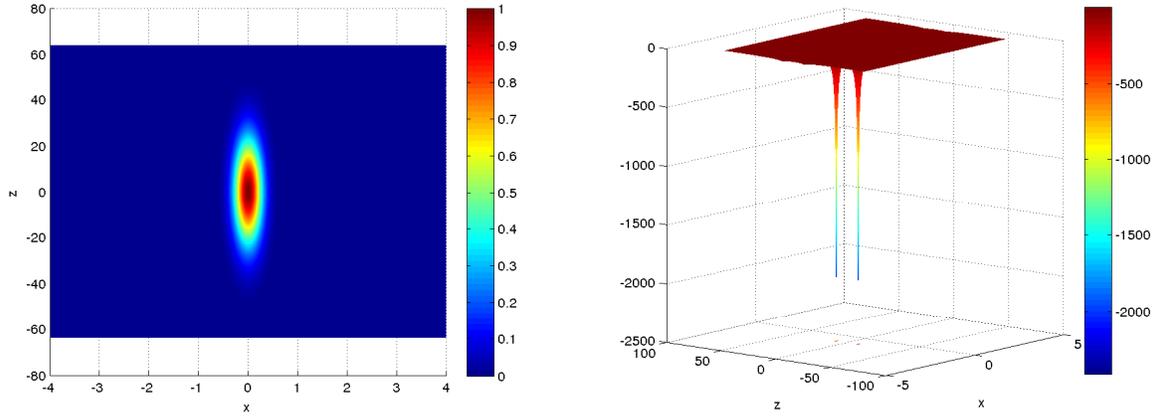


Figure 13: (Left) Density of the initial spinor. (Right) Interaction potential.

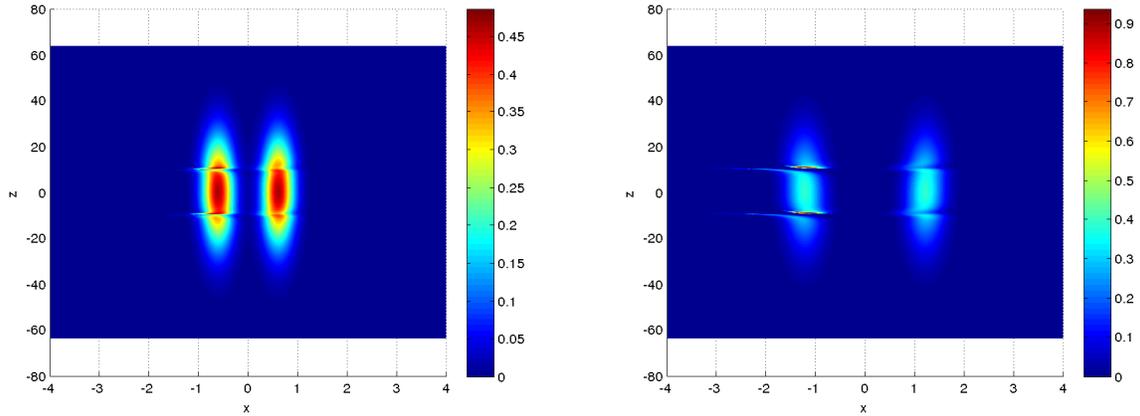


Figure 14: Density at time: (Left) $T = 0.05521$ and (Right) $T_f = 0.1142$.

integration domain is $[0, T_f]$ with $T_f = 4.561 \times 10^{-2}$. The static field is defined by

$$V(x, y, z) = \begin{cases} 0, & (x, y, z) \in (-8, 8)^2 \times \{z < 0\}, \\ 10, & (x, y, z) \in (-8, 8)^2 \times \{z > 0\}. \end{cases}$$

The magnetic field is defined by $\mathbf{A} = (A_x, A_y, A_z)$, with $A_x = A_y = 0$ and A_z is represented at final time T_f in the (x, y) -plane at $z = 0$, and in Fig. 17 (Left), and in the (x, z) -plane and at $y = 0$ in Fig. 17 (Right). The initial data is a wavepacket centered at $(-2.5, 0, 0)$ with wavenumber $k_x = 10$, $k_y = -10$, $k_z = -10$, and the corresponding density $\rho_0(\cdot) = \sum_{i=1}^4 |\psi_i(0, \cdot)|^2$ is represented in the (x, y) -plane at $z = 0$ in Fig. 16 (Left), and in the (x, z) -plane at $y = 0$ in Fig. 16 (Right). We select the time step equal to $\Delta t = \Delta x/c$, that is $\Delta t \approx 9.1 \times 10^{-4}$. We then report in Fig. 18 the density $\rho(T_f, x, y, 0)$ in the (x, y) -plan at $z = 0$ at final time T_f (Left), as well as $\rho(T_f, x, 0, z)$ in the (x, z) -plan at $y = 0$ at T_f (Right).

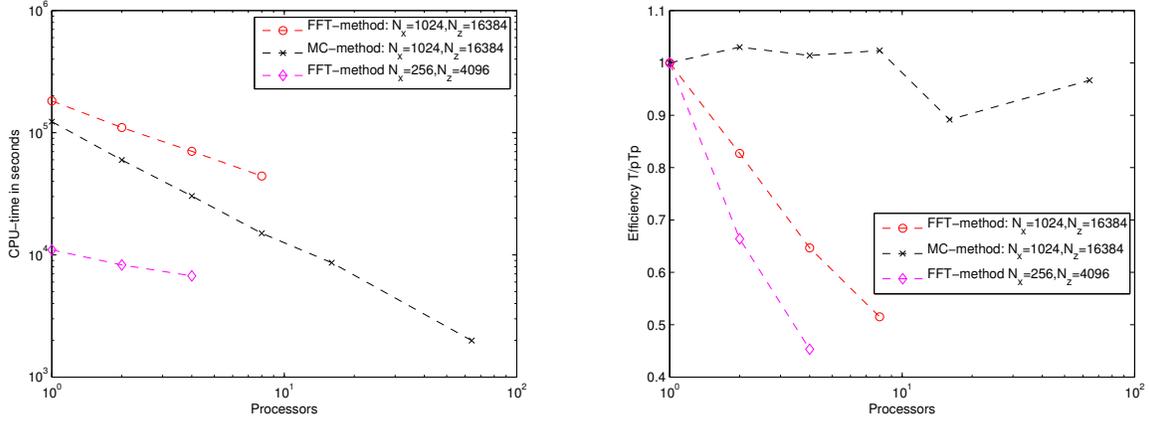


Figure 15: (Left) CPU-time as function of the number of processors. (Right) Efficiency T_1/pT_p for the MC-based and FFT methods: $p = 1, 2, 4, 16, 64$ processors.

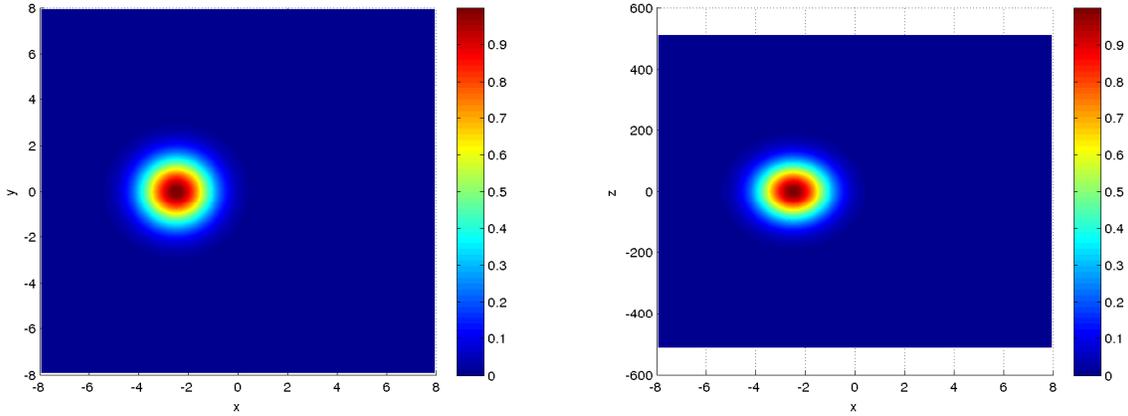


Figure 16: Density at initial time. (Left) $\{(x, y, \rho_0(x, y)), \text{ for } (x, y) \in (-8, 8)^2\}$. (Right) $\{(x, z, \rho_0(x, 0, z)), \text{ for } (x, z) \in (-8, 8) \times (-512, 512)\}$.

We denote by p_x (resp. p_y, p_z) the number of processors in the x -direction (resp. y -, z -direction), and by p the total number of processors with $p = p_x \times p_y \times p_z$. For the MC-based method, we respectively present efficiency results in the case i) $p = p_x \times p_y \times p_z = p_z$ (that is parallelism only in the z -direction) with $p = p_z = 1, 4, 16, 64, 256$, and ii) p_x, p_y and p_z are greater than 1 (parallelism in x, y, z -directions): $p_x = p_y = p_z = 1$ ($p = 1$), $p_x = p_y = p_z = 2$ ($p = 8$), $p_x = p_y = p_z = 4$ ($p = 64$) and $p_x = p_y = 4, p_z = 16$ ($p = 256$). We report in Fig. 19, the efficiency and CPU-time as a function for the MC- and FFT-methods. We notice that the MC-based method is again the most efficient, with in particular, a better efficiency with parallelism in the z -direction. However there is a noticeable deterioration of the efficiency of the MC-based method beyond 16 processors, and unlike the 2d-case. This

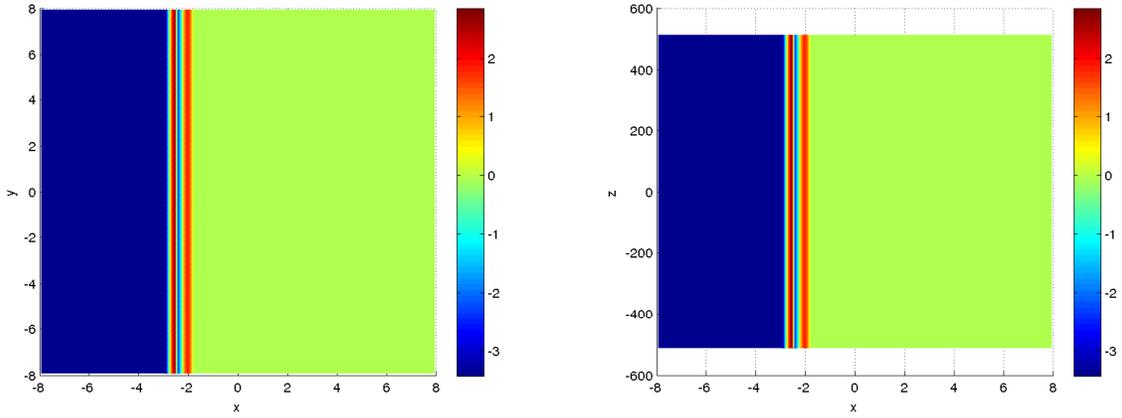


Figure 17: Electric potential. (Left) $\{(x, y, A_z(T_f, x, y, 0)), \text{ for } (x, y) \in (-8, 8)^2\}$. (Right) $\{(x, z, A_z(T_f, x, 0, z)), \text{ for } (x, z) \in (-8, 8) \times (-512, 512)\}$.

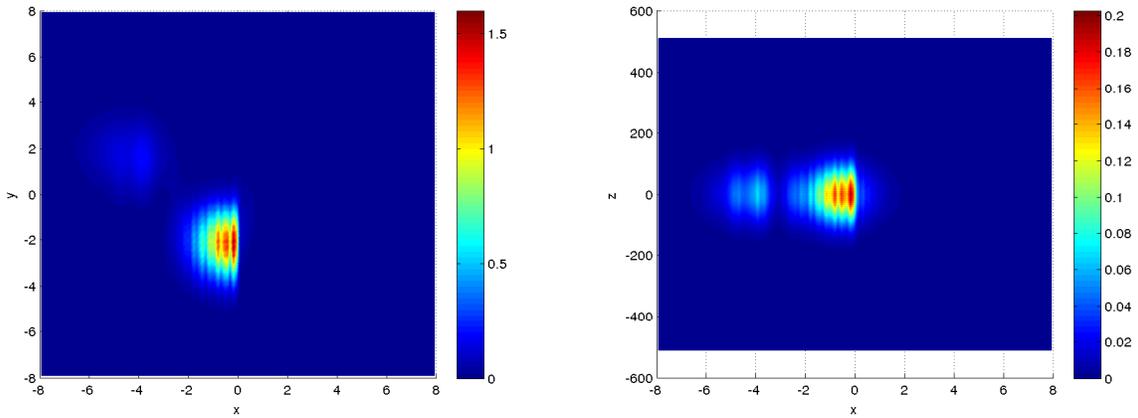


Figure 18: Density at final time. (Left) $\{(x, y, \rho(x, y, 0, T_f)), \text{ for } (x, y) \in (-8, 8)^2\}$. (Right) $\{(x, z, \rho(x, 0, z, T_f)), \text{ for } (x, z) \in (-8, 8) \times (-512, 512)\}$

is due to the heavier load of data to exchange between processors in 3-d.

7. Conclusion

In this paper we have proposed a performance comparison of simple and efficient Dirac equation solvers: a real space method based on the characteristic equation (MC-based method), also referred as a Quantum Lattice Boltzmann method, with simple pseudo-spectral methods (PS-methods). We have established that for a given number of degrees of freedom, the MC-based method is much more efficient sequentially and in parallel than PS-methods. The price to pay is a strict condition imposed on the spatial discretization step. However and

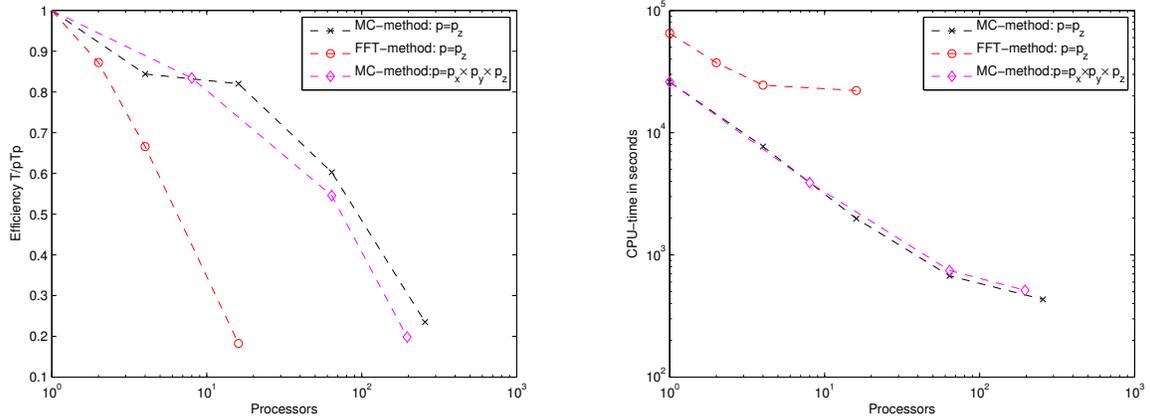


Figure 19: (Left) Efficiency T_1/pT_p for the MC- and FFT-methods. (Right) CPU-time as function of number of processors ($p = 1, 2, 4, 8, 16, 64, 256$ processors).

except for multiscale problems in space, the PS-methods do not impose such constraints, and allow then for taking much larger spatial discretization steps (then reducing the overall computational complexity), while keeping a good accuracy. In conclusion, the MC-based method should be preferred for problems involving very small spatial scales, such as those involving heavy ions, ultrashort laser-atom interaction, while the PS-methods are more adapted for other the physical configurations.

Acknowledgments. The authors would like Dr F. Fillion-Gourdeau (INRS) for helpful discussions.

Appendix A. One-dimensional numerical schemes

Classical FFT-scheme. This scheme necessarily requires the splitting of the Dirac Hamiltonian. From time t_n to t_{n+1} the schemes reads as follows.

1. Source term integration, and $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned}\psi_{h,j}^{(1),n_1} &= \exp(-i\Delta t(mc^2 + V_{\text{nuc},j})/2)\psi_{h,j}^{(1),n}, \\ \psi_{h,j}^{(2),n_1} &= \exp(-i\Delta t(-mc^2 + V_{\text{nuc},j})/2)\psi_{h,j}^{(2),n}.\end{aligned}$$

2. We then solve the potential-free and mass-free Dirac Hamiltonian.

- We set $\phi_h^{n_1} = (\phi_h^{(1),n_1}, \phi_h^{(2),n_1})^T$, where

$$\phi_h^{(1),n_1} = \frac{\psi_h^{(1),n_1} + \psi_h^{(2),n_1}}{\sqrt{2}}, \quad \phi_h^{(2),n_1} = \frac{\psi_h^{(1),n_1} - \psi_h^{(2),n_1}}{\sqrt{2}}.$$

Then

$$\begin{aligned}\phi_h^{(1),n_2} &= \frac{1}{N^{(\text{fft})}} \sum_{p=-N^{(\text{fft})}/2}^{N^{(\text{fft})}/2-1} \left(e^{-i c \xi_p \Delta t} \sum_{k_1=0}^{N^{(\text{fft})}-1} \phi_{h,k_1}^{(j),n_1} e^{-i \xi_p (x_{k_1} + a_x)} \right) e^{i \xi_p (x + a_{x_j})}, \\ \phi_h^{(2),n_2} &= \frac{1}{N^{(\text{fft})}} \sum_{p=-N^{(\text{fft})}/2}^{N^{(\text{fft})}/2-1} \left(e^{i c \xi_p \Delta t} \sum_{k_1=0}^{N^{(\text{fft})}-1} \phi_{h,k_1}^{(j),n_1} e^{-i \xi_p (x_{k_1} + a_x)} \right) e^{i \xi_p (x + a_{x_j})}.\end{aligned}$$

- Then, we get

$$\psi_h^{(1),n_2} = \frac{\phi_h^{(1),n_2} + \phi_h^{(2),n_2}}{\sqrt{2}}, \quad \psi_h^{(2),n_2} = \frac{\phi_h^{(1),n_2} - \phi_h^{(2),n_2}}{\sqrt{2}}.$$

3. Finally, we integrate one more time the source term (Strang-splitting), for $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned}\psi_{h,j}^{(1),n+1} &= \exp(-i \Delta t (m c^2 + V_{\text{nuc.},j})/2) \psi_{h,j}^{(1),n_2}, \\ \psi_{h,j}^{(2),n+1} &= \exp(-i \Delta t (-m c^2 + V_{\text{nuc.},j})/2) \psi_{h,j}^{(2),n_2}.\end{aligned}$$

Split Pseudo-spectral (SPS) scheme. From time t_n to t_{n+1} the scheme reads

1. Source term integration, and $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned}\psi_{h,j}^{(1),n_1} &= \exp(-i \Delta t (m c^2 + V_{\text{nuc.},j})/2) \psi_{h,j}^{(1),n}, \\ \psi_{h,j}^{(2),n_1} &= \exp(-i \Delta t (-m c^2 + V_{\text{nuc.},j})/2) \psi_{h,j}^{(2),n}.\end{aligned}$$

2. We set $\phi_h^{n_1} = (\phi_h^{(1),n_1}, \phi_h^{(2),n_1})^T$, where

$$\phi_h^{(1),n_1} = \frac{\psi_h^{(1),n_1} + \psi_h^{(2),n_1}}{\sqrt{2}}, \quad \phi_h^{(2),n_1} = \frac{\psi_h^{(1),n_1} - \psi_h^{(2),n_1}}{\sqrt{2}}$$

then

- For any $1 \leq j \leq N^{(\text{ps})} - 2$

$$[[\partial_x]] \tilde{\phi}_h^{n_1} = \frac{1}{N^{(\text{ps})}} \sum_{p=-N^{(\text{ps})}/2}^{N^{(\text{ps})}/2-1} i \xi_p \left(\sum_{k_1=0}^{N^{(\text{ps})}-1} \tilde{\phi}_{k_1}^{n_1} e^{-i \xi_p (x_{k_1} + a)} \right) e^{i \xi_p (x_j - a)}.$$

- Then, for all $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned}\phi_{h,j}^{(1),n_1^*} &= \phi_{h,j}^{(1),n_1} - c \Delta t [[\partial_x]] \tilde{\phi}_{h,j}^{(1),n_1} + i \Delta t e A^{n_1} \phi_{h,j}^{(1),n_1}, \\ \phi_{h,j}^{(2),n_1^*} &= \phi_{h,j}^{(2),n_1} + c \Delta t [[\partial_x]] \tilde{\phi}_{h,j}^{(2),n_1} - i \Delta t e A^{n_1} \phi_{h,j}^{(2),n_1}.\end{aligned}$$

- Followed, for all $0 \leq j \leq N^{(\text{ps})} - 1$, by

$$\begin{aligned}\phi_{h,j}^{(1),n_2} &= \phi_{h,j}^{(1),n} - \frac{c \Delta t}{2} \left([[\partial_x]] \tilde{\phi}_{h,j}^{(1),n_1} + [[\partial_x]] \tilde{\phi}_{h,j}^{(1),n_1^*} \right) + i \frac{\Delta t}{2} \left(e A^{n_1} \phi_{h,j}^{(1),n_1} + e A^{n_1^*} \phi_{h,j}^{(1),n_1^*} \right) \\ \phi_{h,j}^{(2),n_2} &= \phi_{h,j}^{(2),n} + \frac{c \Delta t}{2} \left([[\partial_x]] \tilde{\phi}_{h,j}^{(2),n_1} + [[\partial_x]] \tilde{\phi}_{h,j}^{(2),n_1} \right) - i \frac{\Delta t}{2} \left(e A^{n_1} \phi_{h,j}^{(2),n_1} + e A^{n_1^*} \phi_{h,j}^{(2),n_1^*} \right).\end{aligned}$$

- Then, we get

$$\psi_h^{(1),n_2} = \frac{\phi_h^{(1),n_2} + \phi_h^{(2),n_2}}{\sqrt{2}}, \quad \psi_h^{(2),n_2} = \frac{\phi_h^{(1),n_2} - \phi_h^{(2),n_2}}{\sqrt{2}}$$

3. Finally, we integrate one more time the source term (Strang-splitting), for $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned} \psi_{h,j}^{(1),n+1} &= \exp(-i\Delta t(mc^2 + V_{\text{nuc.},j})/2) \psi_{h,j}^{(1),n_2}, \\ \psi_{h,j}^{(2),n+1} &= \exp(-i\Delta t(-mc^2 + V_{\text{nuc.},j})/2) \psi_{h,j}^{(2),n_2}. \end{aligned}$$

Unsplit Pseudo-spectral (UPS) scheme. This scheme corresponds to the unsplit version of the SPS-scheme. From time t_n to t_{n+1} the schemes reads, as follows. We set $\phi_h^{n_1} = (\phi_h^{(1),n_1}, \phi_h^{(2),n_1})^T$, where

$$\phi_h^{(1),n} = \frac{\psi_h^{(1),n} + \psi_h^{(2),n}}{\sqrt{2}}, \quad \phi_h^{(2),n} = \frac{\psi_h^{(1),n} - \psi_h^{(2),n}}{\sqrt{2}}.$$

then

- For any $1 \leq j \leq N^{(\text{ps})} - 2$

$$[[\partial_x]]\tilde{\phi}_h^n = \frac{1}{N^{(\text{ps})}} \sum_{p=-N^{(\text{ps})}/2}^{N^{(\text{ps})}/2-1} i\xi_p \left(\sum_{k_1=0}^{N^{(\text{ps})}-1} \tilde{\phi}_{k_1}^n e^{-i\xi_p(x_{k_1}+a)} \right) e^{i\xi_p(x_j-a)}.$$

- Then for all $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned} \phi_{h,j}^{(1),n_1^*} &= \phi_{h,j}^{(1),n} - c\Delta t [[\partial_x]]\tilde{\phi}_{h,j}^{(1),n} + i\Delta t \left(eA^n \phi_{h,j}^{(1),n} - V_{\text{nuc.},j} \phi_{h,j}^{(1),n} - mc^2 \phi_{h,j}^{(2),n} \right), \\ \phi_{h,j}^{(2),n_1^*} &= \phi_{h,j}^{(2),n} + c\Delta t [[\partial_x]]\tilde{\phi}_{h,j}^{(2),n} - i\Delta t \left(eA^n \phi_{h,j}^{(2),n} + mc^2 \phi_{h,j}^{(1),n} + V_{\text{nuc.},j} \phi_{h,j}^{(2),n} \right). \end{aligned}$$

- Followed, for all $0 \leq j \leq N^{(\text{ps})} - 1$, by

$$\begin{aligned} \phi_{h,j}^{(1),n_1} &= \phi_{h,j}^{(1),n} - \frac{c\Delta t}{2} \left([[\partial_x]]\tilde{\phi}_{h,j}^{(1),n} + [[\partial_x]]\tilde{\phi}_{h,j}^{(1),n_1^*} \right), \\ &\quad + i\frac{\Delta t}{2} \left(eA^n \phi_h^{(1),n} + eA^{n_1^*} \phi_h^{(1),n_1^*} - V_{\text{nuc.},j} (\phi_h^{(1),n_1} + \phi_h^{(1),n_1^*}) - mc^2 (\phi_h^{(2),n_1} + \phi_h^{(2),n_1^*}) \right) \\ \phi_{h,j}^{(2),n_1} &= \phi_{h,j}^{(2),n} + \frac{c\Delta t}{2} \left([[\partial_x]]\tilde{\phi}_{h,j}^{(2),n} + [[\partial_x]]\tilde{\phi}_{h,j}^{(2),n} \right) \\ &\quad - i\frac{\Delta t}{2} \left(eA^n \phi_{h,j} h^{(2),n} + eA^{n_1^*} \phi_{h,j} j^{(2),n_1^*} + V_{\text{nuc.},j} (\phi_{h,j}^{(1),n_1} + \phi_{h,j}^{(1),n_1^*}) \right. \\ &\quad \left. + mc^2 (\phi_{h,j}^{(2),n_1} + \phi_{h,j}^{(2),n_1^*}) \right). \end{aligned}$$

Finally, we have

$$\psi_h^{(1),n+1} = \frac{\phi_h^{(1),n_1} + \phi_h^{(2),n_1}}{\sqrt{2}}, \quad \psi_h^{(2),n+1} = \frac{\phi_h^{(1),n_1} - \phi_h^{(2),n_1}}{\sqrt{2}}.$$

MC-based scheme. From time t_n to t_{n+1} the schemes reads

1. Source term integration and $0 \leq j \leq N^{(\text{ps})} - 1$

$$\begin{aligned} \psi_{h,j}^{(1),n_1} &= \exp(-i\Delta t(mc^2 + V_{\text{nuc.},j})/2)\psi_{h,j}^{(1),n}, \\ \psi_{h,j}^{(2),n_1} &= \exp(-i\Delta t(-mc^2 + V_{\text{nuc.},j})/2)\psi_{h,j}^{(2),n}. \end{aligned}$$

2. We set $\phi_h^{n_1} = (\phi_h^{(1),n_1}, \phi_h^{(2),n_1})^T$, where

$$\phi_h^{(1),n_1} = \frac{\psi_h^{(1),n_1} + \psi_h^{(2),n_1}}{\sqrt{2}}, \quad \phi_h^{(2),n_1} = \frac{\psi_h^{(1),n_1} - \psi_h^{(2),n_1}}{\sqrt{2}}$$

- For any $1 \leq j \leq N^{(\text{mc})} - 2$

$$\phi_{h,j}^{(1),n_2} = \phi_{h,j-1}^{(1),n_1} + i\Delta t e A^{n_1} \phi_{h,j}^{(1),n_1}, \quad \phi_{h,j}^{(2),n_2} = \phi_{h,j+1}^{(2),n_1} - i\Delta t e A^{n_1} \phi_{h,j}^{(2),n_1}.$$

- Then, we get

$$\psi_1^{(1),n_2} = \frac{\phi_h^{(1),n_2} + \phi_h^{(2),n_2}}{\sqrt{2}}, \quad \psi_h^{(2),n_2} = \frac{\phi_h^{(1),n_2} - \phi_h^{(2),n_2}}{\sqrt{2}}.$$

3. Finally, we integrate one more time the source term (Strang splitting) for $0 \leq j \leq N^{(\text{mc})} - 1$

$$\begin{aligned} \psi_{h,j}^{(1),n+1} &= \exp(-i\Delta t(mc^2 + V_{\text{nuc.},j})/2)\psi_{h,j}^{(1),n_2}, \\ \psi_{h,j}^{(2),n+1} &= \exp(-i\Delta t(-mc^2 + V_{\text{nuc.},j})/2)\psi_{h,j}^{(2),n_2}. \end{aligned}$$

References

- [1] E. Ackad and M. Horbatsch. Numerical solution of the Dirac equation by a mapped Fourier grid method. *J. of Phys. A: Mathematical and General*, 38(14):3157, 2005.
- [2] X. Antoine, C. Besse, and V. Rispoli. High-order IMEX-spectral schemes for computing the dynamics of systems of nonlinear Schrödinger/GrossPitaevskii equations. *J. Comput. Phys.*, 327:252–269, 2016.
- [3] X. Antoine, E. Lorin, J. Sater, F. Fillion-Gourdeau, and A.D. Bandrauk. Absorbing boundary conditions for relativistic quantum mechanics equations. *J. Comput. Phys.*, 277:268–304, 2014.

- [4] X. Antoine, E. Lorin, and Q. Tang. A friendly review of absorbing boundary conditions and perfectly matched layers for classical and relativistic quantum wave equations. *to appear in Molecular Physics*, 2017.
- [5] W. Bao, Y. Cai, X. Jia, and Q. Tang. Numerical methods and comparison for the Dirac equation in the nonrelativistic limit regime. *J. of Sc. Comput.*, pages 1–41, 2017.
- [6] W. Bao and X.-G. Li. An efficient and stable numerical method for the Maxwell-Dirac system. *J. Comput. Phys.*, 199(2):663–687, 2004.
- [7] H. Bauke and C.H. Keitel. Accelerating the Fourier split operator method via graphics processing units. *Comput. Phys. Commun.*, 182(12):2454–2463, 2011.
- [8] R. Beerwerth and H. Bauke. Krylov subspace methods for the Dirac equation. *Comput. Phys. Commun.*, 188:189 – 197, 2015.
- [9] J. W. Braun, Q. Su, and R. Grobe. Numerical approach to solve the time-dependent Dirac equation. *Phys. Rev. A*, 59(1):604–612, Jan 1999.
- [10] A. Di Piazza, C. Müller, K. Z. Hatsagortsyan, and C. H. Keitel. Extremely high-intensity laser interactions with fundamental quantum systems. *Rev. Mod. Phys.*, 84:1177–1228, Aug 2012.
- [11] F. H. M. Faisal. A four-component Dirac theory of ionization of a Hydrogen molecular ion in a super-intense laser field. *J. Phys. B: At. Mol. Opt. Phys.*, 42(171003), 2009.
- [12] F. Fillion-Gourdeau, H. J. Herrmann, M. Mendoza, S. Palpacelli, and S. Succi. Formal analogy between the Dirac equation in its Majorana form and the discrete-velocity version of the Boltzmann kinetic equation. *Phys. Rev. Lett.*, 111:160602, Oct 2013.
- [13] F. Fillion-Gourdeau, E. Lorin, and A. D. Bandrauk. Numerical solution of the time-dependent Dirac equation in coordinate space without fermion-doubling. *Comput. Phys. Comm.*, 183(7):1403 – 1415, 2012.
- [14] F. Fillion-Gourdeau, E. Lorin, and A.D. Bandrauk. Resonantly enhanced pair production in a simple diatomic model. *Phys. Rev. Lett.*, 110(1), 2013.
- [15] F. Fillion-Gourdeau, E. Lorin, and A.D. Bandrauk. A split-step numerical method for the time-dependent Dirac equation in 3-d axisymmetric geometry. *J. Comput. Phys.*, 272:559–587, 2014.
- [16] F. Fillion-Gourdeau, E. Lorin, and A.D. Bandrauk. Galerkin method for unsplit 3-d Dirac equation using atomically/kinetically balanced B-spline basis. *J. Comput. Phys.*, 307:122–145, 2016.
- [17] C. Fischer and O. Zatsarinny. A B-spline Galerkin method for the Dirac equation. *Comput. Phys. Comm.*, 180(6):879 – 886, 2009.

- [18] A. Gonoskov, I. Gonoskov, C. Harvey, A. Ilderton, A. Kim, M. Marklund, G. Mourou, and A. Sergeev. Probing nonperturbative QED with optimally focused laser pulses. *Phys. Rev. Lett.*, 111:060404, Aug 2013.
- [19] I. P. Grant. Variational methods for Dirac wave equations. *J. of Phys. B: Atomic and Molecular Physics*, 19(20):3187, 1986.
- [20] I. P. Grant. *Relativistic Quantum Theory of Atoms and Molecules: Theory and Computation*. Springer Series on Atomic, Optical, and Plasma Physics. Springer, 2006.
- [21] I. P. Grant. B-spline methods for radial Dirac equations. *J. of Phys. B: Atomic, Molecular and Optical Physics*, 42(5):055002, 2009.
- [22] B.-Y. Guo, J. Shen, and C.-L. Xu. Spectral and pseudospectral approximations using Hermite functions: Application to the Dirac equation. *Advances in Computational Mathematics*, 19(1-3):35–55, 2003.
- [23] R. Hammer, W. Ptz, and A. Arnold. A dispersion and norm preserving finite difference scheme with transparent boundary conditions for the Dirac equation in (1+1)d. *J. Comput. Phys.*, 256:728–747, 2014.
- [24] C. Itzykson and J. B. Zuber. *Quantum Field Theory*. Mcgraw-hill, 1980.
- [25] O. Kullie, C. Dusterhoft, and D. Kolb. Dirac-Fock finite element method (FEM) calculations for some diatomic molecules. *Chemical Physics Letters*, 314(3-4):307 – 310, 1999.
- [26] E. Lorin and A. Bandrauk. A simple and accurate mixed P_0 - Q_1 solver for the Maxwell-Dirac equations. *Nonlin. Anal. Real World Appl.*, 12(1):190–202, 2011.
- [27] G. R. Mocken and C. H. Keitel. Quantum dynamics of relativistic electrons. *J. Comput. Phys.*, 199(2):558 – 588, 2004.
- [28] G. R. Mocken and C. H. Keitel. FFT-split-operator code for solving the Dirac equation in 2+1 dimensions. *Comput. Phys. Commun.*, 178(11):868 – 882, 2008.
- [29] G.R. Mocken and C.H. Keitel. Quantum dynamics of relativistic electrons. *J. Comput. Phys.*, 199(2):558–588, 2004.
- [30] O. Pinaud. Absorbing layers for the Dirac equation. *J. Comput. Phys.*, 289:169–180, 2015.
- [31] Y. Salamin, S. X. Hu, K. Z. Hatsagortsyan, and C. H. Keitel. Relativistic high-power laser-matter interactions. *Physics Reports*, 427(2-3):41 – 155, 2006.
- [32] S. Salomonson and P. Öster. Relativistic all-order pair functions from a discretized single-particle Dirac Hamiltonian. *Phys. Rev. A*, 40(10):5548–5558, Nov 1989.

- [33] S. Selstø, E. Lindroth, and J. Bengtsson. Solution of the Dirac equation for hydrogenlike systems exposed to intense electromagnetic pulses. *Phys. Rev. A*, 79(4):043418, Apr 2009.
- [34] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2004.
- [35] S. Succi and R. Benzi. Lattice Boltzmann equation for quantum mechanics. *Physica D: Nonlinear Phenomena*, 69(34):327 – 332, 1993.
- [36] J. C. Wells, B. Segev, and J. Eichler. Asymptotic channels and gauge transformations of the time-dependent Dirac equation for extremely relativistic heavy-ion collisions. *Phys. Rev. A*, 59(1):346–357, Jan 1999.
- [37] L. Yang, D. Heinemann, and D. Kolb. Fully numerical relativistic calculations for diatomic molecules using the finite-element method. *Phys. Rev. A*, 48:2700–2707, Oct 1993.