Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 199 (2004) 558–588
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Quantum dynamics of relativistic electrons

Guido R. Mocken *, Christoph H. Keitel

Theoretische Quantendynamik, Physikalisches Institut, Universität Freiburg, Hermann-Herder-Straße 3, D-79104 Freiburg, Germany

Received 1 January 2004; received in revised form 20 February 2004; accepted 23 February 2004
Available online 12 April 2004

Abstract

A new implementation of the split-operator method for calculating the time-evolution of Dirac wave functions is presented. With the help of self-adaptive numerical grids we are able to study the dynamics of an initially free electronic Dirac wave packet under the influence of an ultra-intense laser pulse and its scattering at a highly charged ion. Furthermore, we provide the necessary tools for constructing the Dirac ground state wave function of any single-electron ion and demonstrate them for hydrogenic oxygen. Bound Dirac dynamics in a super-strong laser field is finally investigated.

Keywords: Dirac dynamics; Ultra-intense laser–matter interaction; Self-adaptive grids

1. Introduction

The Dirac equation provides a Lorentz-invariant, i.e. relativistically correct quantum-mechanical description of single particles in the presence of arbitrary time- and position-dependent classical (non-quantized) electromagnetic fields. In the past, to solve it, especially for time-dependent systems, has represented both a challenge and a problem [1–4]. Analytical solutions are known for just a few simples cases. Numerically speaking, the grid sizes and dimensionality have always been a problem. Because of the high demands on computing power only a few attempts to solve the Dirac equation numerically have been made so far. In atomic physics the split operator method was successfully applied in many Schrödinger-type calculations (e.g. [5–7]), but, with a few exceptions [8–10], most of this work was non-relativistic. The split operator method can also be employed to solve the Dirac equation. In fact, some attempts to do so have already been undertaken, mostly high-resolution one-dimensional work [11–13], as well as two- and three-dimensional with either relatively low time resolution [14,15], or for short periods of time [16]. Some analytical work [17,18] has been carried out on the problem of free Dirac particles under the influence of a laser field, i.e. wave packets built from relativistic Volkov states. Several authors have also employed

* Corresponding author. Tel.: +49-761-2037694; fax: +49-761-2035883.
E-mail addresses: mocken@physik.uni-freiburg.de (G.R. Mocken), keitel@physik.uni-freiburg.de (C.H. Keitel).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.02.020

S-matrix-type approaches to work out cross sections for laser assisted scattering [19] or relativistic strong-field photoionization [20]. In general little emphasis was placed on the time-resolved relativistic wave packet dynamics of these processes at short distances when studied for long periods of time and at high time resolution, especially to all orders in the interaction with the nucleus.

The development of fast multi-processor computer systems on the hardware side, and the invention of the fast Fourier transform and efficient implementation [21] thereof on the software side are encouraging enough to study the Dirac problem in further detail. Three-dimensional calculations turn out to be still way out of reach, apart from employing tiny grid sizes or simulations of extremely short time intervals. As most of the interesting physics happens on a somewhat larger scale, we have created a two dimensional implementation of standard algorithms – and incorporated various improvements aimed to significantly increase their efficiency.

We present an analysis of the scattering of a single laser driven electron, represented by a Volkov wave packet, at a single highly charged ion. We carry out two-dimensional calculations over periods of time longer than 10 a.u. using a high time resolution of 2×10^{-5} a.u. and spatial resolutions that are high enough to accommodate any occurring momentum. To be able to investigate bound systems as well, we have modified our code in such a way as to not only propagate wave functions in time, but also to construct stationary states. Those in turn can be used as the starting point in time-dependent problems such as high harmonic generation, above-threshold ionization or high energy [22] processes. In this case, we restrict ourselves to the simulation of an ionization process that is induced by a very strong laser pulse.

The remaining of this paper is divided into sections as follows: Section 2 is dedicated to our implementation of the split-operator method for propagating Dirac wave packets in time. Our approach is different from earlier attempts in that we avoid the standard four-component spinor notation and the inevitable matrix diagonalizations that follow from this. Our implementation not only incorporates the different spinor representation as a C++ class library, which makes manipulation of the discretized Dirac wave function extremely easy, it is also using an adaptive grid approach and multiple threads for maximum performance on shared memory multiple processor architectures. As an example application, we study the dynamics of an initially free electronic Dirac wave packet under the influence of an ultra-intense laser pulse and a single highly charged ion. As an extension to our earlier related work [23], we analyze the influence of a variation of laser parameters on the observed scattering patterns. Section 3 is devoted to the numerical preparation of bound states. We also demonstrate the propagation of an initially bound system. We finish with concluding remarks in Section 4. Appendix A is provided to give some insight into the not so well-known mathematical representation of spinors and the Dirac equation that we use throughout the article. Here, we also describe interesting aspects of FFT implementations in standard numerical libraries and how to properly use them in a physical context.

2. Scattering scenarios of laser-driven Dirac wave functions

We study the dynamics of an initially free electronic Dirac wave packet under the influence of an ultra-intense laser pulse and a single highly charged ion. Our approach is distinguished by two aspects: First, from the mathematical point of view, and second in terms of algorithms and implementation. In the traditional Dirac theory [24] one has to deal with complex four component spinors and complex four by four matrices. But there exists another, less well-known representation of Dirac theory [25] involving real vectors of different degrees of a so-called Clifford or spacetime algebra, i.e. multivector spinors of eight real components, as well as inner and outer products of the algebra elements. Using this representation has not only increased our insight into the physics involved, but has also helped to remove the necessity to diagonalize matrices at any point of the calculation. This numerically expensive calculation was replaced by exact analytical expressions, which, of course, can in principle be found in the form of analytical matrix diagonalizations, too, when one translates back to the standard representation. The spacetime algebra

requires the definition of numerous different operations, such as addition and subtraction of the elements of a certain degree, as well as all sorts of special multiplications of elements of different degrees. This was all implemented using C++ operator overloading techniques, and the elements of the algebra are represented by C++ classes. That way, throughout our code, we have achieved an extremely compact and handy notation for the manipulation of spinors and in fact whole spinor-valued discretized wave functions. This makes it less error prone and much easier to maintain and expand. In high performance computing, one traditionally uses programming languages such as Fortran or pure C, because the corresponding compilers often generate more efficient code. While more complex object oriented languages such as C++ and others will hardly ever be equally efficient, progress in the development of compilers has narrowed the gap, and combined with the huge advances in processor speed, the potential overhead due to less efficient code generation becomes negligible, or in fact, if one considers convenience and reduced time requirements for the actual code development, it becomes well worth its price. Thus we opted for C++. We do not know if there is a substantial price in terms of performance that we pay for such convenience (because to do so, we would have to rewrite the whole code and compare), but we believe our code to be very competitive, if not state-of-the-art. In any case various techniques to be described in the following have shown to give rise to substantial enhancements in the computational efficiency.

2.1. Adaptive numerical grids and multi-threaded programming

To keep down the demand on computing power, we employed the following two techniques: First, in position space, the calculation is not actually carried out for the whole region of interest, but only in a much smaller area that is always kept centred around the wave packet – a so called ''moving-grid'' approach. Second, the grid size, too, is dynamically adapted to the problem. Even a freely evolving wave packet spreads as time goes by, and a scattered one does so even more. As our simulation has to cover the whole laser pulse, including times where the packet is still quite small, it is possible to save a lot of CPU (Central Processing Unit) time here, because one of the most demanding operations, the two-dimensional fast Fourier transform, scales as $O(N^2 \log N)$, where $N \times N$ equals the grid size. This so-called ''growing-grid'' approach is also our pragmatic solution to the well-known boundary problem [4,26] in Dirac calculations. We apply a damping function to the wave function to simulate an absorbing boundary around the edge of the numerical grid, knowing that this is an insufficient solution. However, this damping effectively only comes into play after the grid has reached its maximum size (which is given by practical considerations such as memory and time requirements). Prior to this situation, there is no boundary problem simply because the wave packet is far away from the boundary. By dynamically adjusting the boundary position, we are able to maintain this favourable condition for at least a while.

The whole code is written to take advantage of multiple CPUs, whenever available. For the operations described in Section 2.4, this can be easily realized by sharing the numerical grid among several CPUs and letting each of them act only on a specific part of it, and all in parallel. This was accomplished by using so-called POSIX-threads, which are available on any UNIX-like operating system. If N is the number of grid rows and n the number of CPUs available, then the following formula ensures optimum distribution of grid rows among the units:

$$p = N \operatorname{div} n, \qquad r = N \bmod n, \tag{1}$$

$$\min_i = \begin{cases} i(p+1), & i \leqslant r \\ r(p+1) + (i-r)p, & i > r \end{cases} \qquad \max_i = \min_{(i+1)} - 1. \tag{2}$$

By ''div'' and ''mod'' in Eq. (1) we mean the usual integer operations for division and modulus. The index $i \in \{0, 1, \ldots, n-1\}$ in Eq. (2) enumerates the CPUs, and the set $\{\min_i, \min_i + 1, \ldots, \max_i\}$ represents

the row numbers of the part of the grid that is assigned to the ith CPU. As an example, suppose $N = 11$ and $n = 3$. Then $p = 3$, $r = 2$ and consequently, the three CPUs are acting on rows 0–3, 4–7 and 8–10, respectively. Thus, two of them are dedicated to four rows and one to three rows. Eq. (2) ensures that for any N and n, the difference in size of the individual grid parts is always less or equal to a single row. Hence, the execution times of the separate threads are guaranteed to be almost equal, and waiting times for thread synchronization are minimized. For the multi-threaded implementation of Fourier transforms, we simply rely on the appropriate version of the FFTW library [21].

2.2. The Dirac–Hestenes equation

Hestenes' version [25] of the Dirac equation reads as follows:

$$\hbar\partial\psi i_3 - \frac{q}{c}A\psi = mc\psi\gamma_0, \quad \text{where } i_3 = i\vec{\sigma}_3, \;\; \psi = \langle\psi\rangle_0 + \langle\psi\rangle_2 + \langle\psi\rangle_4. \tag{3}$$

In the above, $\hbar$ is Planck's constant, q and m represent the particle's charge and rest mass, and c is the speed of light. We use atomic units, so for an electron $\hbar = m = -q = 1$ a.u., and $c = 137.036$ a.u. The mathematical objects that occur in (3) all represent elements of the spacetime algebra as explained in detail in Appendix A.1. Specifically, ψ is an even multivector that consists of a scalar $\langle\psi\rangle_0$, a pseudo-scalar $\langle\psi\rangle_4$ and a six-component bivector $\langle\psi\rangle_2$. Both the derivative ∂ and the electromagnetic potential A are vectors of first degree, as well as γ_0. The i_3 is a special constant bivector and plays a role similar to the i in standard notation (its square gives minus one), however it is also closely related to the spin quantization axis $\vec{\sigma}_3$. It is important to see, that it is the i_3 (and not i) that is required for carrying out linear combinations of solutions of (3). Let ψ_1, ψ_2 be solutions to the Dirac equation, then for any $\alpha, \beta \in \mathbb{R}$, the expression $\alpha\psi_1 + \beta\psi_2 i_3$ is also a solution.

The Dirac equation (3) for a time t dependent spinor ψ_t can be formally solved by a unitary operator acting on an arbitrary initial spinor ψ_0

$$\psi_t = \mathrm{e}^{-\frac{\mathrm{i}t}{\hbar}\mathscr{P} i_3}\psi_0, \tag{4}$$

$$\mathscr{P}\psi \overset{\text{Def.}}{=} \underbrace{-\hbar\vec{\partial}\psi i_3 + mc\gamma_0\psi\gamma_0}_{\overset{\text{Def.}}{=}\mathscr{P}_{\vec{\partial}}\psi} + \underbrace{\frac{q}{c}\gamma_0 A_{(\vec{x})}\psi}_{\overset{\text{Def.}}{=}\mathscr{P}_{\vec{x}}\psi} = \mathscr{P}_{\vec{\partial}}\psi + \mathscr{P}_{\vec{x}}\psi, \tag{5}$$

(where "$\overset{\text{Def.}}{=}$" means equal by definition) provided that the potential

$$A = \left(A_0 + \vec{A}\right)\gamma_0, \quad \partial_t A = 0 \tag{6}$$

is time-independent, as indicated above. We use the expressions $\mathscr{P}_{\vec{\partial}}$ and $\mathscr{P}_{\vec{x}}$ as shortcuts for the derivative- and position-dependent parts of the operator $\mathscr{P}$, as defined in (5). However, our main interest are in fact time-dependent potentials

$$A = A(t, \vec{x}). \tag{7}$$

These can be treated approximately, as will be shown in what follows.

2.3. Split operator approximation

If variation of the potential with time is slow, and the potential itself can be regarded as constant during a short time interval Δt, then we can use Eq. (4) in an approximate manner for the short-time propagation

of ψ_{t_0}, the wave function at time $t = t_0$. By this we mean that for $\Delta t \lesssim \frac{\hbar}{2mc^2}$ one can still write, omitting an integral in the exponent,

$$\psi_{(t_0+\Delta t)} \approx \mathrm{e}^{-\frac{c\Delta t}{\hbar}\mathscr{P} i_3}\psi_{t_0} \quad \text{where} \quad \mathscr{P} = \mathscr{P}_{\vec{\partial}} + \mathscr{P}_{t,\vec{x}} = \frac{1}{2}\mathscr{P}_{\vec{\partial}} + \mathscr{P}_{t,\vec{x}} + \frac{1}{2}\mathscr{P}_{\vec{\partial}}. \tag{8}$$

We now proceed to the so-called "split-operator" approximation [12,27,28] which is

$$\psi_{(t_0+\Delta t)}(\vec{x}) \approx \exp\left(-\frac{c\Delta t}{2\hbar}\mathscr{P}_{\vec{\partial}} i_3\right) \exp\left(-\frac{c\Delta t}{\hbar}\mathscr{P}_{\left(t_0+\frac{\Delta t}{2},\vec{x}\right)} i_3\right) \exp\left(-\frac{c\Delta t}{2\hbar}\mathscr{P}_{\vec{\partial}} i_3\right)\psi_{t_0}(\vec{x}) + \mathrm{O}(\Delta t^3). \tag{9}$$

Eq. (9) can be iterated, hence providing a means to calculate the wave function at arbitrary times in the future, following the scheme $\psi_t = \psi_{(t_0+N\Delta t)}$, where N is the number of iterations required. As one can see, the error per step of iteration is of the order $\mathrm{O}(\Delta t^3)$ [29,30]. If Δt is chosen small enough, then this procedure will converge towards the exact solution of the time-dependent problem. However, there still is the problem to evaluate the application of the two different exponential operators in (9) in a convenient way. In a Schrödinger split operator theory, this is very easy, because one is then dealing with exponentials of just single-component multiplication and derivative operators. The latter can be transformed into the former by means of a Fourier transform, and the remaining problem is trivial exponentiation of complex numbers. But in the case of a Dirac theory, one has to deal with more complicated multivector operators, or, in the standard representation, matrix operators, and exponential functions thereof. The usual way to deal with the latter is according to the scheme

$$e^A = \exp\left(S(S^\dagger A S)S^\dagger\right) = S\exp\left(\mathrm{diag}(a_1,\ldots,a_4)\right)S^\dagger = S\,\mathrm{diag}(e^{a_1},\ldots,e^{a_4})\,S^\dagger \tag{10}$$

if one assumes, that a unitary matrix S ($SS^\dagger = 1$) can be found, that diagonalizes the hermitian matrix A: $S^\dagger A S = \mathrm{diag}(a_1,\ldots,a_4)$. Then, the only problem is to find S. This can be accomplished numerically, but then at the price of substantial CPU time, because this calculation is required at any time t for all possible values of $\vec{x}$. As we will see now, for the special cases involved here, there also is an analytic solution. It is based on series expansions, but from the result one could also obtain the matrix S and use it for diagonalization purposes. We now proceed with this specific transformation of the exponential operators in Eq. (9).

2.4. Transformation of the exponential evolution operators

Formally, the result for the first and third exponential in Eq. (9) can be written as

$$\begin{aligned}
&\exp\left(-\frac{c}{\hbar}\frac{\Delta t}{2}\mathscr{P}_{\vec{\partial}} i_3\right)\psi_{t_0}(\vec{x}) = \mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\psi_{t_0}(\vec{x}), \quad \text{where}\\
&\mathscr{G}_{\vec{k}}\hat{\psi}_{(\vec{k})} \overset{\text{Def.}}{=} \cos\left(\phi(\vec{k})\right)\hat{\psi} - \sin\left(\phi(\vec{k})\right)\frac{\vec{k}\hat{\psi} + \frac{mc}{\hbar}\hat{\psi}^*}{\sqrt{\vec{k}^2 + \left(\frac{mc}{\hbar}\right)^2}} i_3,\\
&\hat{\psi}_{(\vec{k})} \overset{\text{Def.}}{=} (\mathscr{F}\psi)_{(\vec{k})} \overset{\text{Def.}}{=} \frac{1}{(2\pi)^{\frac{3}{2}}}\int \mathrm{d}^3k\,\psi_{(\vec{x})}\,\mathrm{e}^{-i_3\vec{k}\cdot\vec{x}},\\
&\phi(\vec{k}) \overset{\text{Def.}}{=} c\frac{\Delta t}{2}\sqrt{\vec{k}^2 + \left(\frac{mc}{\hbar}\right)^2}.
\end{aligned} \tag{11}$$

In order to derive the above, after inserting the definitions, we have used Fourier transforms $\mathscr{F}$ and inverse transforms $\mathscr{F}^{-1}$ (see also (A.27)) and then expanded the exponential into a power series. After sorting the

latter with respect to even and odd powers and by collecting them into a sine and a cosine series, both of which only depend on a scalar phase, we arrive at an expression which can be calculated immediately by our code.

Because of the non-commutativity of the algebra, there is a difference between operators $\mathscr{A}$ that are applied from the left and other operators $\mathscr{B}$ applied from the right onto a spinor. We use the notation $(\mathscr{A}\Box\mathscr{B})(\psi) = \mathscr{A}\psi\mathscr{B}$, i.e. the "$\Box$" marks the position where, upon application of the operator $(\mathscr{A}\Box\mathscr{B})$, the spinor has to be inserted. In particular, $(\mathscr{A}\Box\mathscr{B})^2(\psi) = \mathscr{A}^2\psi\mathscr{B}^2$. Keeping this in mind, the second exponential in (9) splits exactly into another two:

$$\begin{aligned}\exp\left(-\frac{c}{\hbar}\Delta t\,\mathscr{P}_{(t,\vec{x})}i_3\right) &= \exp\left(-\frac{\Delta t\,q}{\hbar}\gamma_0 A_{(t,\vec{x})}\Box i_3\right)\\ &= \exp\left(-\frac{\Delta t\,q}{\hbar}\left(A_{0(t,\vec{x})} - \vec{A}_{(t,\vec{x})}\right)\Box i_3\right)\\ &\overset{A_0\vec{A}=\vec{A}A_0}{=} \exp\left(-\frac{\Delta t\,q}{\hbar}A_{0(t,\vec{x})}\Box i_3\right)\exp\left(\frac{\Delta t\,q}{\hbar}\vec{A}_{(t,\vec{x})}\Box i_3\right).\end{aligned} \tag{12}$$

We evaluate both of them similarly to the one already shown. The final results are again in a numerically attractive form:

$$\exp\left(-\frac{\Delta t\,q}{\hbar}A_{0(t,\vec{x})}\Box i_3\right) = \cos\left(\frac{\Delta t\,q}{\hbar}A_{0(t,\vec{x})}\right)\Box - \sin\left(\frac{\Delta t\,q}{\hbar}A_{0(t,\vec{x})}\right)\Box i_3 \overset{\text{Def.}}{=} \mathscr{D}^{A_0(t)}\Box, \tag{13}$$

$$\exp\left(\frac{\Delta t\,q}{\hbar}\vec{A}_{(t,\vec{x})}\Box i_3\right) = \cos\left(\frac{\Delta t\,q}{\hbar}\left|\vec{A}_{(t,\vec{x})}\right|\right)\Box + \frac{\vec{A}_{(t,\vec{x})}}{\left|\vec{A}_{(t,\vec{x})}\right|}\sin\left(\frac{\Delta t\,q}{\hbar}\left|\vec{A}_{(t,\vec{x})}\right|\right)\Box i_3 \overset{\text{Def.}}{=} \mathscr{D}^{\vec{A}(t)}\Box. \tag{14}$$

2.5. Iteration for finite times

If one assembles the last two results (13) and (14) according to (15) and combines them with (11) according to (9), then, after N iterations, one arrives at

$$\mathscr{D}^A_j \overset{\text{Def.}}{=} \mathscr{D}^{A_0(t_0+(j-\frac{1}{2})\Delta t))}\mathscr{D}^{\vec{A}(t_0+(j-\frac{1}{2})\Delta t)}, \tag{15}$$

$$\begin{aligned}\psi_{(t_0+\Delta t)}(\vec{x}) &\approx \mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\mathscr{D}^A_1\mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\psi_{(t_0)}(\vec{x}),\\ \psi_{(t_0+2\Delta t)}(\vec{x}) &\approx \left(\mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\mathscr{D}^A_2\mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\right)\left(\mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\mathscr{D}^A_1\mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\right)\psi_{(t_0)}(\vec{x}),\\ &= \mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\mathscr{D}^A_2\mathscr{F}^{-1}\left((\mathscr{G}_{\vec{k}})^2\mathscr{F}\mathscr{D}^A_1\mathscr{F}^{-1}\right)\mathscr{G}_{\vec{k}}\mathscr{F}\psi_{(t_0)}(\vec{x}),\\ &\cdots\\ \psi_{(t_0+N\Delta t)}(\vec{x}) &\approx \mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\mathscr{D}^A_N\mathscr{F}^{-1}\left(\prod_{n=1}^{N-1}(\mathscr{G}_{\vec{k}})^2\mathscr{F}\mathscr{D}^A_n\mathscr{F}^{-1}\right)\mathscr{G}_{\vec{k}}\mathscr{F}\psi_{(t_0)}(\vec{x}).\end{aligned} \tag{16}$$

2.6. Boundary problems

As a consequence of the finite grid size, parts of the wave packet eventually reach the grid boundaries. The adaptive-grid approach can only delay but not completely avoid this behaviour, because there are

practical limits for the grid size. We usually used a limit of 768×768 points. Grids that are much larger than this do not only need copious amounts of memory, but also the execution time per iteration step is becoming unbearably long. Therefore, when we have reached this limiting size, the grid's centre is still mapped to the particle's centre of mass, but it is no longer enlarged, and therefore parts of the wave function can reach the boundaries. To avoid reflections, we use an absorbing boundary [31]. This is a well-established technique for Schrödinger-type calculations [7,10,28,32–35]. When transferring to Dirac calculations, however, some peculiarities have to be considered.

A popular choice for a boundary mask function is the one shown in Fig. 1(a). Here the parameter x_0 designates the boundary position and α influences the steepness. However, it is rather unsuitable in the context of Dirac calculations, because after many iterations (corresponding to multiplication of the wave function by large powers of the mask function), the effective boundary moves inwards, which means that the actual amount of space in which the wave packet is allowed to live, is decreasing the longer the simulation goes. A much more suitable mask function is the one shown in Fig. 1(b). With this choice, the effective boundary position is fixed in space, and clearly defined by the parameters x_0 and x_1. Even when iterated many times, the effective boundary does not move, it only becomes steeper. Since our calculations typically require about a million steps, we used $f^{0.001}$ instead of f, so effectively, at the at of the simulation, the values on the whole grid were multiplied by f^{1000}. Because $f \in C^\infty$, it is very smooth, and in our simulations, this particular choice turned out to be a very good compromise between the introduction of new errors (which is inevitable for any mask function) and the reduction of errors caused by reflections at the grid border.

Another question is, whether a damping function is correct in the context of Dirac theory in the first place: As pointed out in [4,26], the only solution of the Dirac equation with Dirichlet boundary condition $\psi|_{\partial V} = 0$ is the trivial one $\psi(\vec{x}) = 0\ \forall \vec{x} \in V$, so one might ask how we can obtain a non-trivial solution although the wave function is damped down to zero when it approaches the boundaries. The explanation is the following: We are not actually solving the Dirac equation under those strict boundary conditions, instead, we take an arbitrary initial function (which in our case, since it is a Gaussian wave packet, just happens to be almost zero at the boundaries, but does not have to be) and apply a unitary time evolution operator to it, thereby making it a solution to the Dirac equation. It is only after this step that we apply the mask function, thereby destroying the exact solution. This now only approximate solution is exactly propagated in time again, masked, propagated further, and so on. Thus, there is no reason why it should collapse to the trivial solution. Of course, the successive masking gradually degrades the quality of the

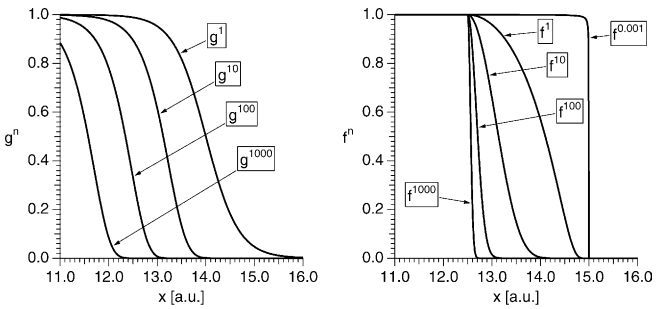

Fig. 1. Behaviour of different boundary mask functions g^n, f^n when iterated $n \in \{1, 10, 100, 1000\}$ times. In addition, $f^{0.001}$ is plotted.

approximation. Note that our masking approach is very cautious: We choose x_1 right on the grid's edge and x_0 at a distance of 10% times the current grid size away from the edge. Furthermore, and as pointed out before, we use $f^{0.001}$ instead of f. Finally, because of the adaptive grids, the simulated wave packet hardly ever reaches the boundary region. Consequently, for calculations such as those in Section 2.10, after about 8 a.u. evolution time, the typical loss of normalization due to boundary damping is still less than 0.1%.

When dealing with stationary states, one way to treat the boundary problem is by setting only the probability current across the border to zero, but not the wave function itself, similar as in the often quoted [26] "MIT-bag-model" [36]. The latter, however, is a "particle-in-a-box" system, unlike our case: We envisage the particle to flow across the border undisturbed, therefore the transmitted probability current should not be limited at all, but the reflected current should be completely suppressed. The mask function approach employed is a simulation of this absorbing behaviour.

2.7. From 3D to 2D

Using the methods reported before, despite all simplification and the availability of fast computers, it is still almost impossible to perform calculations in three spatial dimensions, at least not for grids of interesting physical extent and for times larger than small fractions of a laser cycle. Therefore, it is interesting to ask, to what extent lower dimensional model systems are a good approximation of fully 3D reality.

2.7.1. The 2D Dirac equation

Suppose we have the 3D Dirac equation, but a potential A that is invariant under translations in one particular direction designated by $\vec{n}$

$$\hbar\partial\psi(\vec{x})i_3 - \frac{q}{c}A(\vec{x}_\perp)\psi(\vec{x}) = mc\psi(\vec{x})\gamma_0, \quad \text{where } \vec{x}_\perp = \vec{n}\times(\vec{x}\times\vec{n}), \tag{17}$$

then we can use the separation ansatz

$$\psi(\vec{x}) = \psi'(\vec{x}_\perp)\frac{1}{\sqrt{2\pi}}\exp\left(\frac{i_3}{\hbar}p_\parallel^0 x_\parallel\right), \quad \text{where } x_\parallel = \vec{n}\cdot\vec{x}, \tag{18}$$

and insert it here:

$$\hbar\left(\partial_\perp + \gamma_0\vec{n}\partial_\parallel\right)\psi i_3 - \frac{q}{c}A(\vec{x}_\perp)\psi = mc\psi\gamma_0. \tag{19}$$

The result is:

$$\hbar\partial_\perp\psi' i_3 - \frac{q}{c}\left(A(\vec{x}_\perp) + \frac{c}{q}p_\parallel^0\gamma_0\vec{n}\right)\psi' = mc\psi'\gamma_0. \tag{20}$$

In general, there will not be a single momentum $p_\parallel^0$, but rather a momentum distribution giving specific weights for arbitrary $p_\parallel \in \mathbb{R}$ and the solution is a wave packet instead of (18). We restrict ourselves to the simple choice $p_\parallel^0 = 0$ to get rid of this constant. Other values would not provide more physical insight, and true wave packet calculations would cause the computational times to grow into the regime of true three-dimensional calculations. The remaining is the 2D Dirac equation:

$$\hbar\partial_\perp\psi' i_3 - \frac{q}{c}A(\vec{x}_\perp)\psi' = mc\psi'\gamma_0, \tag{21}$$

Two-dimensional ("2D") does not mean that there is no magnetic field $\vec{B}$ (which points in the $\parallel$-direction). Note that so far, our treatment is *exact*.

A vector potential that describes a laser beam of electric field amplitude E_0 and frequency ω, which is linearly polarized in direction $\hat{a}$ and propagating in direction of the wave vector $\vec{k}$, such as (note the space-time-split of the vector A according to (A.15)).

$$A \overset{(A.15)}{=} \left(A_0 + \vec{A}\right)\gamma_0, \quad \text{with } \vec{A} = -\hat{a}\frac{c}{\omega}E_0 \sin(\omega t - \vec{k}\cdot\vec{x}), \quad A_0 = 0 \tag{22}$$

is invariant in direction $\vec{n} \overset{\text{Def.}}{=} \vec{k} \times \hat{a}$. However, the 3D Coulomb potential

$$A_0 = \frac{Ze}{r} = \frac{Ze}{\sqrt{x_\parallel^2 + \vec{x}_\perp^2}}, \quad \vec{A} = \vec{0} \tag{23}$$

depends on all three directions. Thus, one has to make an approximation here. By replacing the influence of the third variable by a constant parameter α, we arrive at the so-called softcore-potential [37]:

$$A_0 = \frac{Ze}{\sqrt{\alpha + \vec{x}_\perp^2}}, \quad \vec{A} = \vec{0}. \tag{24}$$

For maximum consistency with three-dimensional reality, this parameter should be chosen in such a way as to yield the same ground state energy as for the three-dimensional Coulomb potential.

2.7.2. Observables and expectation values

Let $\mathscr{A} = \mathscr{A}_\perp$ be the operator of an observable, that is restricted to the $\perp$-directions (e.g. $\mathscr{A} = \vec{x}_\perp$ position operator). Then:

$$(\psi, \mathscr{A}\psi) \underset{(A.25)}{\overset{(18)}{=}} \hbar\,\delta(0) \int \mathrm{d}^2 x_\perp \left\langle \psi'^{\dagger}_{p_\parallel^0}(\vec{x}_\perp)\mathscr{A}_\perp \psi'_{p_\parallel^0}(\vec{x}_\perp) \right\rangle_0 . \tag{25}$$

As one can see, for every bilinear form of ψ one has to omit the factor $\hbar\,\delta(0)$ in order to obtain reasonable results. Normalization suffers from the same problem. This can be seen by setting $\mathscr{A} = 1$.

2.8. Program

Instead of documenting the whole source code of well over ten thousand lines, which is clearly beyond the scope of this paper, we prefer to explain the essential part of the code using a flow chart. Fig. 2 shows the core of the quantum mechanical calculations. Essentially, it is the realization of Eq. (16):

$$\psi_{(t_0+N\Delta t)}(\vec{x}) \approx \mathscr{F}^{-1}\mathscr{G}_{\vec{k}}\mathscr{F}\mathscr{D}_N^A\mathscr{F}^{-1}\left(\prod_{n=1}^{N-1}(\mathscr{G}_{\vec{k}})^2\mathscr{F}\mathscr{D}_n^A\mathscr{F}^{-1}\right)\mathscr{G}_{\vec{k}}\mathscr{F}\psi_{(t_0)}(\vec{x}).$$

First, some necessary initializations are performed, such as opening files that the various outputs will be written to, and determining reasonable storage intervals. For the latter, one has to find a compromise, because the outputs require additional operations which slow down the code considerably. The initial wave function is set, either as a simple Gaussian wave packet, or by using a previously calculated ground state wave function (see Section 3). After that, the above equation is gone through from right to left, i.e. the initial wave function $\psi_{(t_0)}$ is Fourier transformed into momentum space and operated on by $\mathscr{G}_{\vec{k}}$. Then, inside the main loop, there are $(N-1)$ successive applications of inverse Fourier transform, operation $\mathscr{D}^A$, forward Fourier transform and double $\mathscr{G}_{\vec{k}}$ operation. The final state is produced by applying yet another inverse Fourier transform, the Nth $\mathscr{D}^A$ operation, again a Fourier transform followed by a single $\mathscr{G}_{\vec{k}}$ operation and a final transformation back to position space. At the previously chosen storage intervals, this

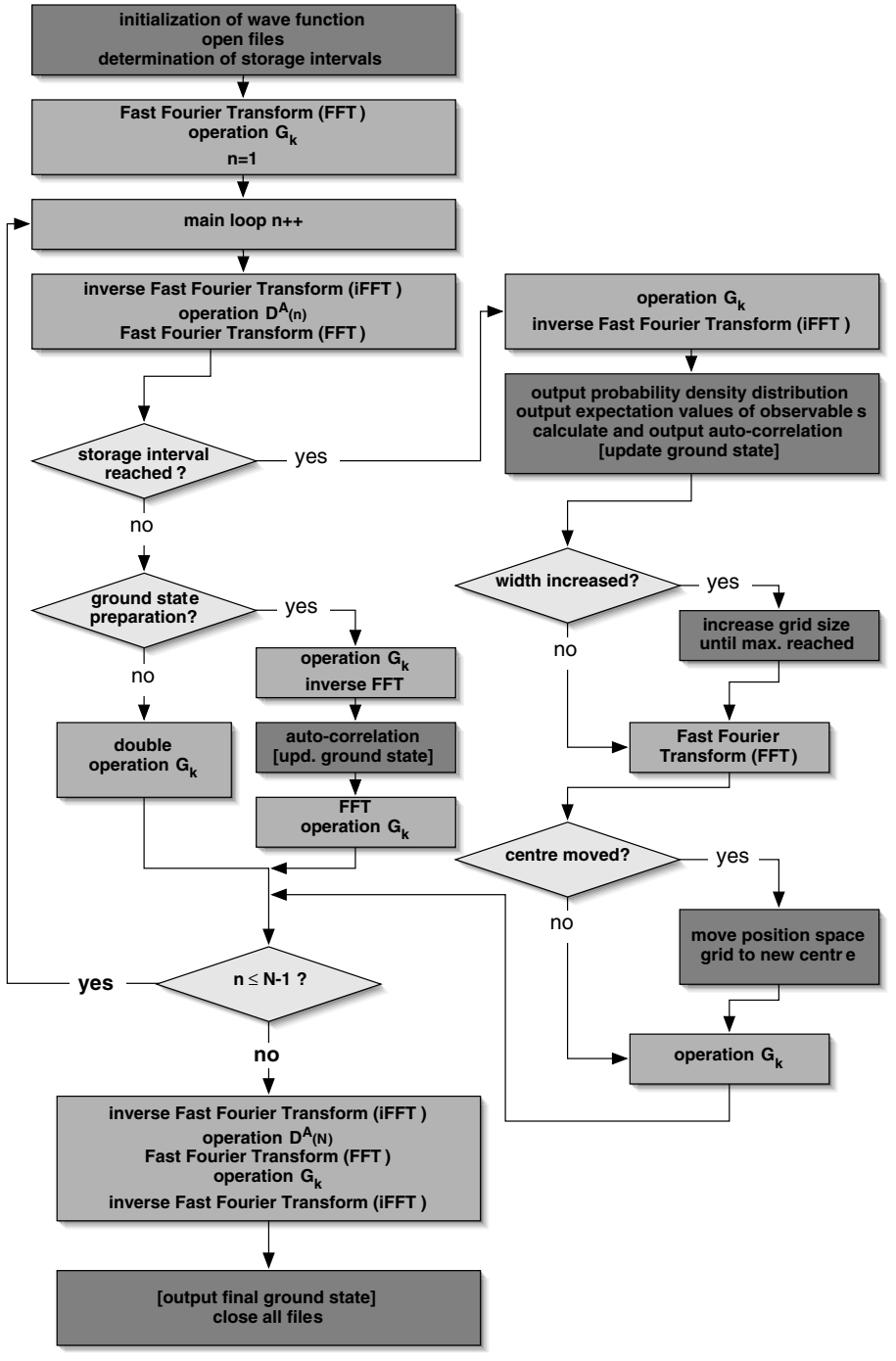

Fig. 2. Flow chart illustrating our code. The dark rectangular boxes represent complex operations that are described in the text, whereas the light ones simply contain the elementary operations of Eq. (16). For the storage intervals, the actions inside the main loop on the left side are replaced by those on the right side of the chart. The operations in square brackets are only executed if ground state preparation is the goal, and then only during the mandatory second pass of the whole program.

scheme is interrupted to calculate some interesting observables and save them to disk. Then, instead of the faster double $\mathscr{G}_{\vec{k}}$ operation, we have to apply two single ones as well as backward and forward Fourier transforms in between. After the first of these two Fourier transforms, the correct intermediate state is realized in position space and can be further analyzed. Interesting properties include the probability density distribution, which one can use to create very illustrative animations of the process, the centre of mass, i.e. the expectation value of the position operator, and the wave function's norm, i.e. the integral of the density over the whole grid. When it is not particle propagation in space and time but the preparation of a suitable ground state wave function that one is interested in, then these intermediate steps are mandatory all the time, because the auto-correlation function (see Section 3 for definition and usage) is needed at full time resolution: The energy of bound states is just slightly lower than $+mc^2$, so the corresponding frequencies are very high. The summation of the ground state (see Section 3 for an explanation of this procedure) for an energy value that was previously determined from the auto-correlation spectrum has to be carried out during a second pass of the whole program, and is therefore indicated in square brackets only.

One special feature of our code is the adaptive numerical grid. It is continuously adapted in both size and position to keep the computation time as short and the calculation as accurate as possible at the same time. Therefore, whenever we take one of the above-mentioned intermediate steps, in addition to the centre of mass coordinate, the width of the wave packet is also estimated. Whenever the width has grown by more than two grid spacings, the whole grid is copied to another one, that is enlarged by one grid point on both sides with respect to the original one. Note that this does not occur very often (a few hundred times at maximum), so it is not worth optimizing this procedure. Whenever the centre has moved by more than one grid spacing, the grid's position it shifted as well. To save the time spent for a true copy procedure, this is effectively achieved by an additional phase during the inverse Fourier transform (see Appendix A: this corresponds to an adjustment of t_0 in (A.53)), or to say it differently: The grid in position space is effectively moved while the wave function resides in momentum space. Note that the movement of the grid in position space is *not* equivalent to a Kramers–Henneberger-transform [38,39] to the comoving system. All our calculations are carried out in the laboratory frame, we are only dynamically deciding which points can be omitted. There is no true Lorentz transform to the comoving frame taking place. Also note that we turned off the adaptive grids for the ground state calculations in Section 3, because this by definition is a stationary problem.

The spatial resolution of the grid is kept fixed during the evaluations. It has to be chosen high enough to accommodate the maximum occurring *canonical* momentum $p_{\mathrm{can,max}}$, i.e. $\frac{\pi}{\Delta x} \gtrsim \frac{p_{\mathrm{can,max}}}{\hbar}$, where Δx denotes the grid's step size. For both polarization and propagation direction, this condition is easily met for the calculations presented in this work, which produce only high *kinetic* momenta. We have chosen $\Delta x = 0.118$ a.u. (corresponding to $p_{\mathrm{can,max}} = 26.6$ a.u.) for the calculations in Section 2.10, and $\Delta x = 0.01174$ a.u. ($p_{\mathrm{can,max}} = 267.6$ a.u.) for the ones of Sections 3.3 and 3.4.

2.9. Scaling of run time with respect to the number of processors

To fully exploit the potential of shared memory multiple processor architectures, we designed our code to be multi-threaded. In Fig. 3 we show the dependence of the execution time per iteration step on the number of CPUs employed. In order to measure these values, we used a fixed grid size of 64×64 cells and took the time for a thousand steps when utilizing one to four threads. Then we repeated the experiment for a considerably larger grid size of 1024×1024 cells, but only ten steps. Not surprisingly, large grids turn out to scale more favourably than small ones. These measurements were performed on a SGI Origin machine with four R10000 processors running at 225 MHz. Note that we used this somewhat obsolete machine only for this measurement of *relative* performance, and for the single reason that we needed exclusive access to all four CPUs during this experiment. Our production machines, although much faster on average, are not well-suited for this particular task, because they have either just two CPUs (dual Intel Itanium 1.4 GHz), or

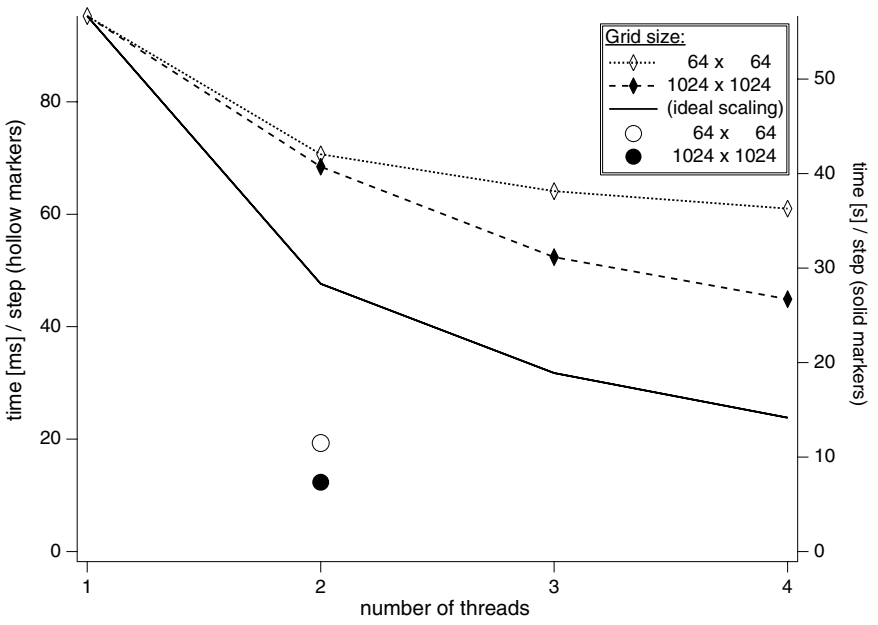

Fig. 3. Scaling of execution time per iteration step as a function of the number of threads employed on a SGI Origin machine with four R10000 (225 MHz) processors. The left axis indicates the time per iteration step for the measurements that are based on a small 64×64 points grid and designated by hollow markers. Similarly, the right axis stands for a larger 1024×1024 grid and refers to solid markers. The dotted line corresponds to a small grid, whereas the dashed one represents the same measurements on the same machine for a large grid. The solid line, as a reference, indicates ideal hyperbolic scaling. The two single points indicate the performance of our code on a dual Itanium 1.4 GHz machine.

they have four logical but only two physical CPUs (dual Intel Xeon 2.4 GHz), or they have plenty of CPUs but are shared among many users in a not-so-well-controlled way as to permit exact measurement of scaling. Our actual calculations were mostly executed on such a shared IBM p-series 690 machine featuring 24 POWER-4 processors at 1.1 GHz, and used eight threads. To give an idea of the *absolute* performance, we included two points in Fig. 3 that represent a recent dual Itanium 1.4 GHz machine. Note however, that in most of our calculations, the grid size is not constant, and scaling is therefore more complex and, in fact, unpredictable, since the grid size depends on the evolution of the simulated particle. Also, unlike in these simplified examples, arbitrary grid sizes $N \times M$ ($N \neq M$) can occur, and the FFTW performs suboptimally if N or M is no power of two.

2.10. Physical results for single ion laser-assisted scattering

In the following we apply our computational approach to the scenario of laser assisted scattering at a single ion in the relativistic regime. The setup is very similar to the situation that we examined in [23]. This time, however, we use a laser pulse of frequency $\omega = 2$ a.u., travelling in the $+\vec{\sigma}_1$ direction, linearly polarized in $+\vec{\sigma}_2$ direction. The simulation starts at $t = -0.055$ a.u. and goes on for about 17 a.u. At time $t = 0$ a.u. the pulse front reaches the origin. The pulse's envelope function consists of a two cycle $\sin^2$-shaped turn-on, a single cycle plateau and, symmetrically, a two cycle turn-off. In polarization direction, there is no amplitude dependence, which is a good approximation for the spatial distances of less than a hundred atomic units considered here. We vary the laser electric field's amplitude E_0 from 75 to 150 a.u. in steps of 25 a.u. and calculate the time evolution of a Gaussian wave packet (full width 2 a.u. in momentum space at

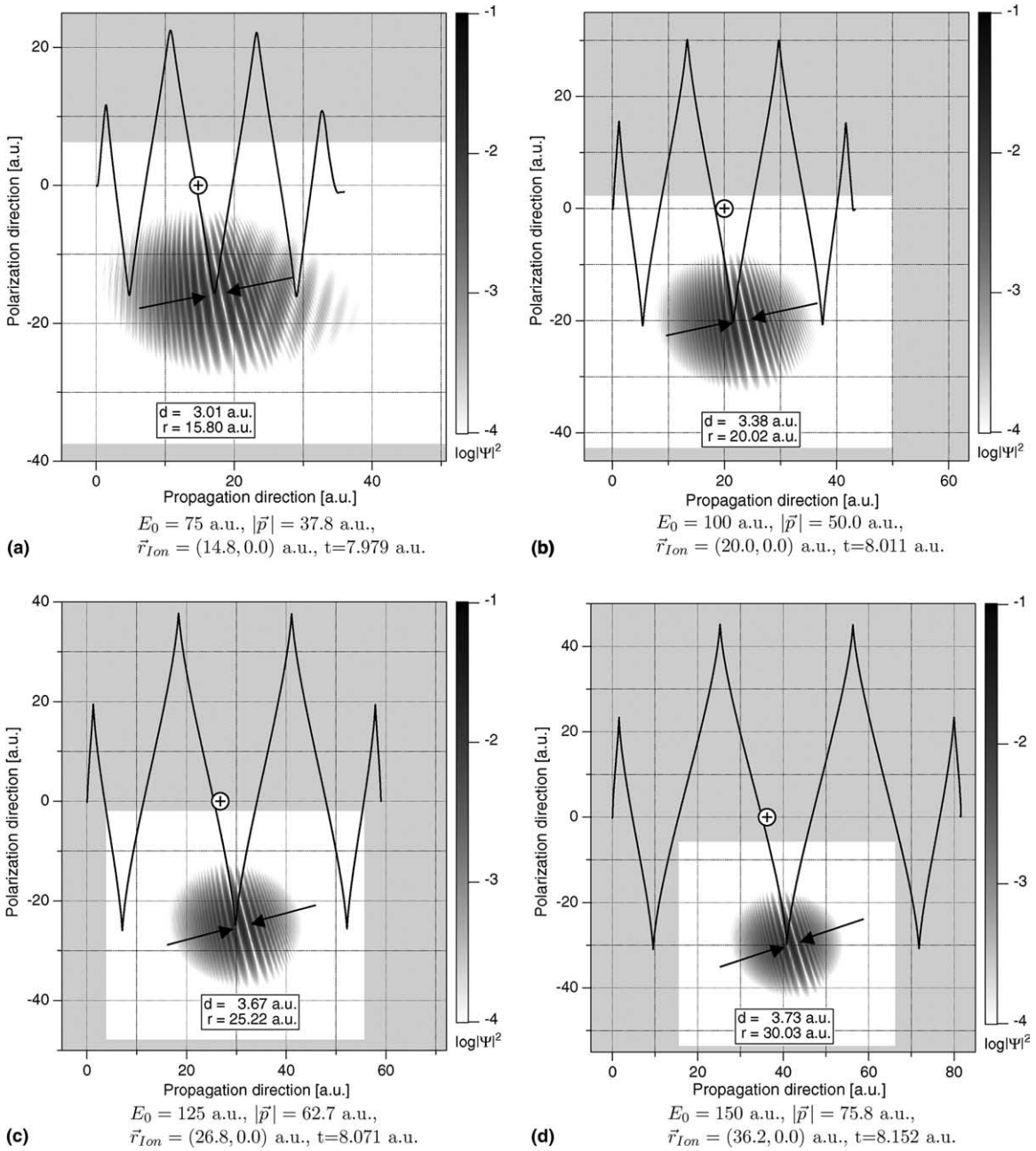

Fig. 4. Initially Gaussian wave packets were propagated in a laser field ($\omega = 2$ a.u.) and photographed right at the first lower turning point after the scattering. The logarithm of the probability density of the scattered wave packet is given by the greyscale image plot in the background. The white rectangle corresponds to the numerical grid. The black thick line marks the trajectory of the position operator's expectation value, the single "+" marker indicates the ion's ($Z = 50$) position. The arrowheads designate the left and right first order maxima and the value d is the spatial distance between these two points. r is the distance measured from the middle between these points to the ion's position. Note that the scales are different in all these plots. The individual captions provide values for the laser electric field amplitude E_0, the kinetic momentum magnitude $|\vec{p}|$ at the time of the scattering event, the ion's position $\vec{r}_{\mathrm{Ion}}$, and the time t at which the snapshot was taken.

$1/e$th of the maximal height, and solely with positive energy and spin up with respect to $\vec{\sigma}_3$) which is initially at rest at the origin. For each run, we have beforehand estimated the optimum position for a single naked ion of charge $Z = 50$ times the elementary charge by looking at a classical trajectory calculation. Each time the ion was placed on the horizontal zero axis in such a way as to be encountered by the electronic wave packet after just roughly $2\frac{1}{4}$ cycles, which would correspond to a zero crossing of the laser's electromagnetic field. We evaluate the momentum's expectation value $\vec{p}$ at this moment. Then, after another $\frac{1}{4}$ cycle, when the scattered wave packet has come to a rest again, we measured the spatial distance d between the two first order maxima, and their distance r to the ion. Snapshots of the simulations at this very moment are plotted in Fig. 4(a)–(d); and the numerical results for d and r are analyzed in Fig. 5. The simple classical relation [40]

$$|\vec{p}_{\max}| = \frac{qE_0}{\omega}\sqrt{1 + \frac{q^2E_0^2}{4m^2c^2\omega^2}} \approx \frac{qE_0}{\omega}, \tag{26}$$

(which can be linearly approximated as shown above with less than 4% error for the values used here) between momentum and field amplitude is verified in Fig. 6. A linear fit recovers the frequency $\omega = \frac{1}{0.507} \approx 2$ a.u.

From Fig. 5 we learn, that the higher the momentum during the scattering, the larger the fringes spacing $\frac{1}{4}$ cycle later and the distance r reached at this point in time.

The latter is obvious and can be quantified as follows: r is roughly half the oscillation extent in polarization direction Δx_{pol}, which can be classically calculated [40] for a free particle with mass m and charge q as

$$\Delta x_{\text{pol}} = \frac{2qE_0}{m\omega^2}, \tag{27}$$

however it is slightly reduced by the Coulomb attraction of the core. It is even more shifted to lower values because, due to the non-symmetric turn-on phase, the ion's position right on the horizontal zero line is not exactly halfway between two turning points.

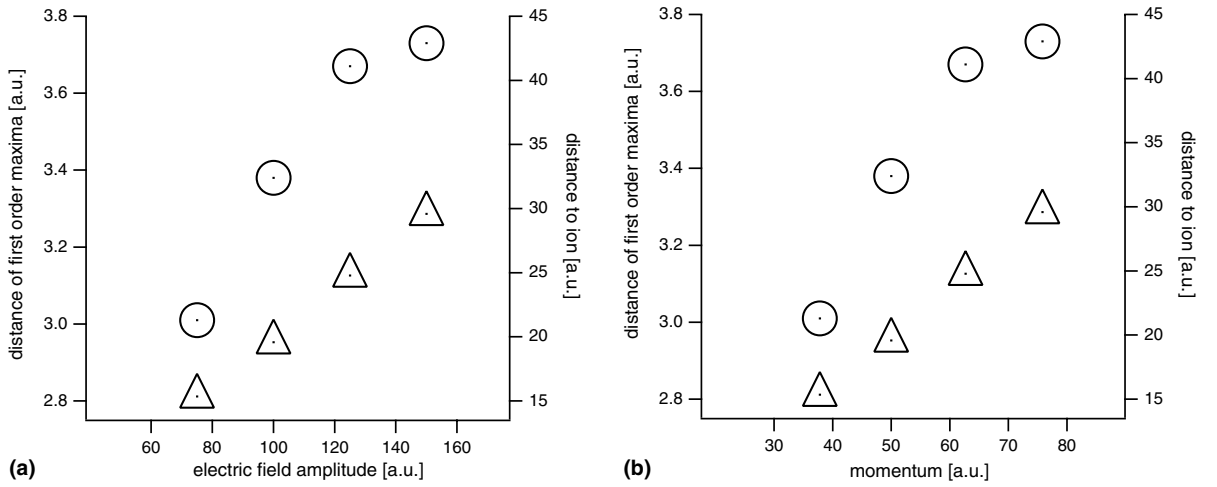

Fig. 5. Distance between the first order maxima (circular markers, left axis) and their distance to the ion (triangular marker, right axis) plotted versus the electric field amplitude (a) or the momentum at the moment of scattering (b).

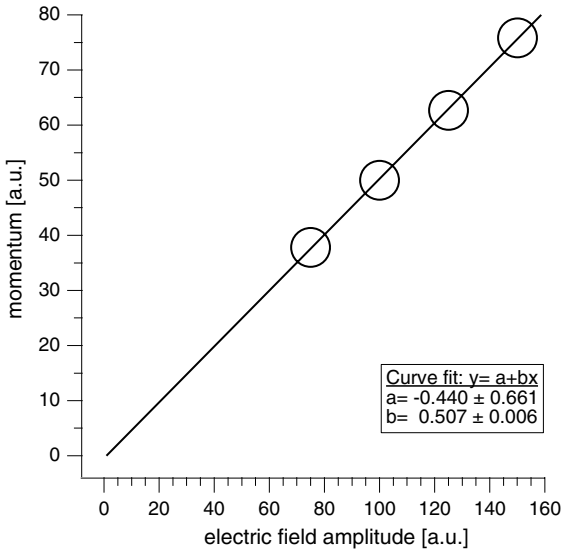

Fig. 6. Mean momentum of the electronic wave packet measured at the moment of scattering versus electric field amplitude.

The former relation regarding d is more complicated, and certainly non-linear. For the scattering angles $\vartheta = \pm \arctan\left(\frac{d}{2r}\right)$, the simple model that we suggested in [23] predicts (ignoring the laser field's influence after the scattering) values similar to the ones that we see here. In any case, increasing scattering momenta which are the outcome of an increase in the electric field amplitude during the particle's initial accelerating phase, cause the scattering angles to decrease. This is because the distance to the ion grows much faster than the distance between the first order maxima.

It must be noted that all the quantities given in these figures, although stated without error bounds, are subject to a certain arbitrariness of the order of several atomic units. This does not arise from erroneous numerics, it is simply due to the inevitable inability to clearly define a precise "moment in time" for the scattering of a spatially extended particle. The same is true for the exact definition of the lower turning point at which we measured d and r. We also did not precisely match the phase at the time and location of the scattering to exactly $2\frac{1}{4} \times 2\pi$ for the above, merely qualitative analysis.

Note that in Fig. 4(a), apart from the scattering already described, some additional fringes are visible on the right hand side. These are left behind from a partial scattering of the right edge of the wave packet half a cycle earlier than the main scattering event. Their orientation reflects the direction of motion at this moment. In Fig. 4(b)–(d) the forward drift is large enough to avoid this effect.

3. Bound Dirac wave functions

Before we turn to the numerical procedure for creating stationary states in arbitrary potentials, we briefly indicate the analytical solution for the case of a Coulomb potential in two dimensions. As one will see later, the latter represents not only a crude approximation for the true solution in the softcore potentials that we use in our time-evolution simulations, but is the perfect candidate for the trial wave function required by the numerical procedure that we employ to calculate the exact ground state.

3.1. Analytical ground state for the Coulomb potential

For a plain Coulomb potential ($\alpha = 0$) in two dimensions (24)

$$A = (A_0 + \vec{0})\gamma_0 = \frac{Ze}{\sqrt{\vec{x}_\perp^2}}\gamma_0, \tag{28}$$

we choose $\vec{n} = \vec{\sigma}_3$, so

$$\vec{n} = \vec{\sigma}_3 \quad \Rightarrow \quad \vec{x}_\perp = \vec{\sigma}_3 \times (\vec{x} \times \vec{\sigma}_3), \quad x_\parallel = \vec{\sigma}_3 \cdot \vec{x}. \tag{29}$$

We switch to cylindrical coordinates:

$$\vec{x}(r, \varphi, x_\parallel) = r\,\mathrm{e}^{-i_3\frac{\varphi}{2}}\vec{\sigma}_1\,\mathrm{e}^{i_3\frac{\varphi}{2}} + x_\parallel\vec{\sigma}_3 = \vec{x}_\perp + x_\parallel\vec{\sigma}_3, \tag{30}$$

and employ the separation ansatz (18) to arrive at the effective 2D equation (20). We separate the time by inserting the ansatz

$$\psi'(\vec{x}_\perp) = \psi''(\vec{x}_\perp)\exp\left(-\frac{i_3 E x_0}{\hbar c}\right) \tag{31}$$

in

$$\hbar\gamma_0\left(\partial_0 + \vec{\partial}_\perp\right)\psi' i_3 - \frac{q}{c}\left(A(\vec{x}_\perp) + \frac{c}{q}p_\parallel^0\gamma_0\vec{\sigma}_3\right)\psi' = mc\psi'\gamma_0, \tag{32}$$

to obtain a time-independent 2D Dirac equation:

$$\frac{E}{c}\gamma_0\psi'' + \hbar\gamma_0\vec{\partial}_\perp\psi'' i_3 - \frac{q}{c}\left(A(\vec{x}_\perp) + \frac{c}{q}p_\parallel^0\gamma_0\vec{\sigma}_3\right)\psi'' = mc\psi''\gamma_0. \tag{33}$$

For the special case $p_\parallel^0 = 0$, it is possible to analytically find the eigenstates ψ'' and energy eigenvalues E of the above equation, both dependent on quantum numbers l and n_r. For these, there are some restrictions:

$$\begin{aligned} &-\left(l + \frac{1}{2}\right) < 0 \Rightarrow n_r \in \mathbb{N}_0, \\ &-\left(l + \frac{1}{2}\right) > 0 \Rightarrow n_r \in \mathbb{N} \setminus \{0\}. \end{aligned} \tag{34}$$

The energy eigenvalues can be found to be as follows:

$$\frac{E}{mc^2} = \left\{\frac{(Z\alpha)^2}{\left(n_r + \sqrt{\left(l + \frac{1}{2}\right)^2 - (Z\alpha)^2}\right)^2} + 1\right\}^{-\frac{1}{2}}, \tag{35}$$

where $\alpha = \frac{e^2}{\hbar c}$, and the analytical result for the ground ($n_r = 0, l = 0$) state energy is

$$E_0 = mc^2\sqrt{1 - 4(Z\alpha)^2}, \tag{36}$$

$n_r = 0, l = -1$ would give the same value, but would violate the conditions (34). The correctly normalized ground state wave functions with spin up and down with respect to $\vec{\sigma}_3$ are finally given by

$$\psi''_{0\uparrow}(x_0,r,\varphi)=\frac{(2\lambda_0)^{\frac{\epsilon_0+1}{2}}}{\sqrt{4\pi\Gamma(\epsilon_0+1)}}\left(\sqrt{1+\epsilon_0}+\vec{\sigma}_2\,\mathrm{e}^{i_3\varphi}\sqrt{1-\epsilon_0}\right)\mathrm{e}^{-\lambda_0 r}r^{\frac{\epsilon_0-1}{2}}\mathrm{e}^{-\frac{i_3E_0x_0}{\hbar c}}$$

$$\psi''_{0\downarrow}(x_0,r,\varphi)=\frac{(2\lambda_0)^{\frac{\epsilon_0+1}{2}}}{\sqrt{4\pi\Gamma(\epsilon_0+1)}}\left(\sqrt{1+\epsilon_0}-\vec{\sigma}_2\,\mathrm{e}^{i_3\varphi}\sqrt{1-\epsilon_0}\right)\mathrm{e}^{-\lambda_0 r}r^{\frac{\epsilon_0-1}{2}}i_2\mathrm{e}^{-\frac{i_3E_0x_0}{\hbar c}}\quad\text{where}$$

$$\epsilon_0\overset{\text{Def.}}{=}\sqrt{1-4(Z\alpha)^2}=\frac{E_0}{mc^2}$$

$$\lambda_0\overset{\text{Def.}}{=}mc^2\frac{2Z\alpha}{\hbar c}. \tag{37}$$

These are the reduced 2D wave functions, now normalized to unity in two spatial dimensions. To derive the above results, we have imitated Landau–Lifshitz' [41] calculation for the 3D Coulomb problem (which itself relies on the original work by Gordon [42]). Similar work can be found in [43], which relies on [44]. Some intermediate steps were inspired by [45]. Our calculation differs from the one in [46,47] in the sense, that our spinors ψ,ψ',ψ'' are not defined in a two-dimensional world, but in three dimensions, and therefore have all eight real components (as defined in (A.14) of the Appendix A), not just two complex ones. It is just the potential that does not depend on one particular direction, and therefore the corresponding dependence of the wave functions is trivial (compare (18)).

3.2. Numerical ground state for the softcore potential

Many ways to numerically calculate the Dirac ground state in a given scalar potential are known. Just to mention a few, there is inverse iteration [14], the methods by Lehmann-Maehly and Kato [48], simple variational techniques [49] and algorithms that resort to Monte-Carlo-methods [50]. There also is a method [51] that is built upon the Foldy–Wouthuysen transformation [52].

In this section, our aim is clearly *not* to compete with any of the above mentioned techniques in terms of accuracy or efficiency. Moreover, we want to show how, with a minimum of additional programming, one can calculate the initial wave functions for the bound state problems that we are to analyze, within acceptable time and with satisfying accuracy. There are two so far unmentioned methods that seem to meet these requirements: The first one is the so-called propagation in imaginary time [53], and the second one is the spectral method [54]. The former consists of putting $\tau=it$ and then propagating in τ instead of t. It is easily shown, that the renormalized propagated state is converging towards the lowest energy eigenstate. The method is fast converging and easy to implement – but unfortunately of no use for the Dirac case, because there simply is no lowest energy eigenvalue in the Dirac theory. The discrete part of the spectrum is ranging from $-mc^2$ to $+mc^2$, with most states lying close to these boundaries. Instead of converging to the positive energy ground state, the procedure will run into the negative energy Rydberg states. A similar problem is inherent to many of the other above-mentioned methods and is known in the literature under the name "variational collapse" [48,55].

Therefore, the method that we used after all, is the slower converging spectral method [54]. Its advantage is that it can also deliver excited states, just as needed. Degenerated states however, cannot be differentiated. In the section to come, we will discuss all the mathematics involved in much detail.

3.3. Spectral method

The method splits in two parts: First, the energy spectrum of a trial wave function is determined. From this spectrum one chooses the lowest energy peak for which, in a second step, the corresponding wave function is generated.

3.3.1. Energy spectrum

We define the projection (sometimes called auto-correlation) of a wave function ψ at time t onto the one at time t_M under the name $P(t)$ as follows

$$P(t) \overset{\text{Def.}}{=} (\psi_{t_M}, \psi_t). \tag{38}$$

We expand the wave function

$$\psi_t = \sum_{n,j} \phi_{n,j} c_{n,j} \, \mathrm{e}^{-\frac{i_3 t}{\hbar} E_n}, \tag{39}$$

where $\phi_{n,j}$ are the bound eigenstates, E_n the corresponding energies, degenerated in index j, and $c_{n,j}$ constant i_3-complex coefficients. Thus

$$\begin{aligned} P(t) = (\psi_{t_M}, \psi_t) &\overset{(39)}{=} \left(\sum_{n',j'} \phi_{n',j'} c_{n',j'} \, \mathrm{e}^{-\frac{i_3 t_M}{\hbar} E_{n'}}, \sum_{n,j} \phi_{n,j} c_{n,j} \, \mathrm{e}^{-\frac{i_3 t}{\hbar} E_n} \right) \\ &= \sum_{n,j} \left(\sum_{n',j'} \phi_{n',j'} c_{n',j'} \, \mathrm{e}^{-\frac{i_3 t_M}{\hbar} E_{n'}}, \phi_{n,j} \right) c_{n,j} \, \mathrm{e}^{-\frac{i_3 t}{\hbar} E_n} \\ &= \sum_{n,j} \sum_{n',j'} \underbrace{(\phi_{n',j'}, \phi_{n,j})}_{=\delta_{nn'}\delta_{jj'}} (c_{n',j'} \, \mathrm{e}^{-\frac{i_3 t_M}{\hbar} E_{n'}})^{\dagger} c_{n,j} \, \mathrm{e}^{-\frac{i_3 t}{\hbar} E_n} = \sum_{n,j} c_{n,j}^{\dagger} c_{n,j} \, \mathrm{e}^{-\frac{i_3 (t - t_M)}{\hbar} E_n}. \end{aligned} \tag{40}$$

It is obvious from Eq. (40) that

$$P(t_M - t) = P(t_M + t)^{\dagger}, \tag{41}$$

which explains the name t_M as the middle of the interval $[t_M - t, t_M + t]$. For any arbitrarily chosen window function $W(t)$ we can evaluate the one-dimensional Fourier transform (A.29)

$$(\mathbb{F} f)_{(\omega)} \overset{\text{Def.}}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}t \, f_{(t)} \, \mathrm{e}^{-i_3 \omega t}, \tag{42}$$

of the product $W(t)P^{\dagger}(t)$ with the help of the convolution (denoted by "$*$") theorem (A.31):

$$\begin{aligned} \left(\mathbb{F}(WP^{\dagger})\right)_{(\omega)} &\overset{(A.31)}{=} \frac{1}{\sqrt{2\pi}} (\mathbb{F}W)_{(\omega)} \; * \, (\mathbb{F}P^{\dagger})_{(\omega)} \\ &\overset{(A.31)}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}\omega' (\mathbb{F}W)_{(\omega')} (\mathbb{F}P^{\dagger})_{(\omega - \omega')} \\ &\overset{\substack{(40) \\ (A.29)}}{=} \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{d}\omega' (\mathbb{F}W)_{(\omega')} \int_{-\infty}^{+\infty} \mathrm{d}t \sum_{n,j} |c_{n,j}|^2 \, \mathrm{e}^{\frac{i_3 (t - t_M)}{\hbar} E_n} \, \mathrm{e}^{-i_3 (\omega - \omega') t} \\ &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{d}\omega' (\mathbb{F}W)_{(\omega')} \int_{-\infty}^{+\infty} \mathrm{d}t \sum_{n,j} |c_{n,j}|^2 \, \mathrm{e}^{i_3 t \left(\frac{E_n}{\hbar} - (\omega - \omega') \right)} \, \mathrm{e}^{-i_3 \frac{t_M E_n}{\hbar}} \\ &\overset{(A.30)}{=} \int_{-\infty}^{+\infty} \mathrm{d}\omega' (\mathbb{F}W)_{(\omega')} \sum_{n,j} |c_{n,j}|^2 \, \delta\left(\frac{E_n}{\hbar} - (\omega - \omega') \right) \mathrm{e}^{-i_3 \frac{t_M E_n}{\hbar}} \\ &= \sum_{n,j} (\mathbb{F}W)_{\left(\omega - \frac{E_n}{\hbar}\right)} |c_{n,j}|^2 \, \mathrm{e}^{-i_3 \frac{t_M E_n}{\hbar}}. \end{aligned} \tag{43}$$

Note that we have offset the frequency argument to avoid its misinterpretation as a multiplicative factor. Now, if we choose $W(t+t_M)=g(t)$ as a real, even (with respect to zero) function $g(-t)=g(+t)$, then we arrive at

$$W(t)=g(t-t_M), \tag{44}$$

$$\begin{aligned}(\mathbb{F}W)_{\left(\omega-\frac{E_n}{\hbar}\right)} &\overset{(42)}{=} \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \mathrm{d}t\, W(t)\mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)t} \overset{(44)}{=} \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \mathrm{d}t\, g(t-t_M)\mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)t} \\ &\overset{t'=t-t_M}{=} \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \mathrm{d}t'\, g(t')\mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)(t'+t_M)} = \mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)t_M}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \mathrm{d}t'\, g(t')\mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)t'} \\ &\overset{(42)}{=} \mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)t_M}(\mathbb{F}g)_{\left(\omega-\frac{E_n}{\hbar}\right)},\end{aligned} \tag{45}$$

and since $g(-t)=g^{\dagger}(+t)=g(t)$, $\mathbb{F}g$ is real. If, for example, one chooses a Gaussian for g, then $\mathbb{F}g$ is a real Gaussian, again. Combining all, we obtain

$$\begin{aligned}\left(\mathbb{F}(WP^{\dagger})\right)_{(\omega)} &\overset{(45)}{=}_{(43)} \sum_{n,j}\mathrm{e}^{-i_3\left(\omega-\frac{E_n}{\hbar}\right)t_M}(\mathbb{F}g)_{\left(\omega-\frac{E_n}{\hbar}\right)}|c_{n,j}|^2\,\mathrm{e}^{-i_3\frac{t_M E_n}{\hbar}} = \sum_{n,j}\mathrm{e}^{-i_3\omega t_M}(\mathbb{F}g)_{\left(\omega-\frac{E_n}{\hbar}\right)}|c_{n,j}|^2 \\ &= \sum_{n}\mathrm{e}^{-i_3\omega t_M}(\mathbb{F}g)_{\left(\omega-\frac{E_n}{\hbar}\right)}\sum_{j}|c_{n,j}|^2,\end{aligned} \tag{46}$$

which, in our example's case, is a sum over Gaussian peaks located at the frequencies $\frac{E_n}{\hbar}$, which are the energy eigenvalues that we are looking for, multiplied by their statistical weight in the initial wave function and another phase factor $\mathrm{e}^{-i_3\omega t_M}$. Now, the problem is the following: At first, there is a known (but arbitrary) initial wave function ψ_{t_0}, where in general $t_0\neq 0$. The time propagation is running from t_0 until t_1. The choice $t_M=\frac{t_0+t_1}{2}$ appears obvious at first glance, but is incorrect. At the beginning of the calculation, $\psi_{\frac{t_0+t_1}{2}}$ is still unknown and cannot be used to evaluate $P(t)$. The only possible choice in order to be able to calculate $P(t)$ for all $t\in[t_0,t_1]$, is $t_M=t_0$. But then all values for $t\in[t_0-(t_1-t_0),t_0]$ are missing. However, $P(t)$ is known from (41) even for theses values of t, without knowing ψ_t. Thus, in order to obtain the proper energy spectrum, the following steps have to be taken:

- Evaluate $P(t)$ for $t\in[t_0,t_1]$ during the propagation of ψ_t.
- Determine the missing values of $P(t)$ for the range $[t_0-(t_1-t_0),t_0]$ by using a reflection (41) of the known ones, in order to have $P(t)$ known for the whole range $t\in[t_0-(t_1-t_0),t_1]$ and let $t_M=t_0$ be the middle of this interval.
- Choose a suitable function $W(t)$, whose support is identical with the above range and which is symmetric with respect to the middle of it.

If, finally, one wants to obtain a real spectrum, then it is mandatory to shift t_0 to 0 (and t_1 to t_1-t_0) first, which also means $t_M=0$. This also simplifies the above discussion. In praxis, a good choice for W is the so-called Hann window [56]:

$$W(t)=\begin{cases}1-\cos\left(\frac{2\pi(t-t_l)}{t_r-t_l}\right), & t\in[t_l,t_r],\\ 0, & \text{otherwise},\end{cases} \tag{47}$$

which is symmetric around $\frac{t_l+t_r}{2}$. In our special case with symmetry around zero, one simply has to choose $t_l=-(t_1-t_0)$ and $t_r=+(t_1-t_0)$. This function's advantage is, that its Fourier transform exhibits just a single peak and no further oscillations as it would be the case for $W=1$ or a Gaussian. As the Fourier

transform is executed numerically on a finite grid, the unity function becomes a rectangular one and the Gaussian becomes a cropped Gaussian – and that is why in these cases one would expect the result to contain δ- or Gaussian peaks accompanied by unwanted oscillations.

3.3.2. Ground state wave function

Having successfully extracted the desired ground state energy E_0 from the energy spectrum, one calculates

$$\begin{aligned}\int_{-\infty}^{+\infty} \mathrm{d}t\, \psi_t W_{(t)} \mathrm{e}^{\frac{i_3 t}{\hbar} E_0} &\overset{(39)}{=} \int_{-\infty}^{+\infty} \mathrm{d}t \left(\sum_{n,j} \phi_{n,j} c_{n,j} W_{(t)} \mathrm{e}^{-\frac{i_3 t}{\hbar} E_n} \right) \mathrm{e}^{\frac{i_3 t}{\hbar} E_0} \\ &= \sum_{n,j} \phi_{n,j} c_{n,j} \int_{-\infty}^{+\infty} \mathrm{d}t\, W_{(t)} \mathrm{e}^{-\frac{i_3 t}{\hbar}(E_n - E_0)} \overset{(\mathrm{A}.29)}{=} \sum_{n,j} \phi_{n,j} c_{n,j} \sqrt{2\pi} (\mathbb{F}W)_{\left(\frac{E_n - E_0}{\hbar}\right)} \\ &\overset{\text{see below}}{=} \sum_j \phi_{0,j} c_{0,j} \sqrt{2\pi} (\mathbb{F}W)_{(0)} \propto \sum_j \phi_{0,j} c_{0,j}, \end{aligned} \tag{48}$$

which is, if one assumes $(\mathbb{F}W)_{(\omega)}$ to be strongly localized around zero, the wanted ground state wave function or at least a linear combination of all degenerated states of energy E_0. The window function W (47)

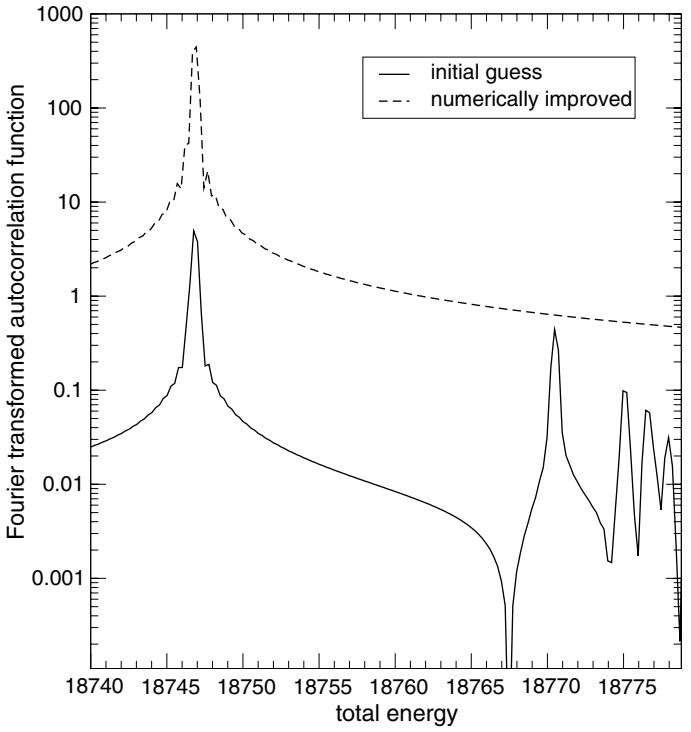

Fig. 7. Energy spectrum of both the analytic ground state wave function of the Coulomb potential, which we take as our initial guess for the ground state wave function of the softcore potential (solid line) and the same one after it was numerically improved (dashed line). Note that the left axis is in arbitrary units and that the dashed curve was multiplied by 50 to separate it from the solid one. The bottom axis shows total energy and its right limit corresponds to the rest mass energy $mc^2 = 18778.865$. The ground state peak is located at $E_0 = 18746.8$ a.u.

can be chosen independently from the previous section. In fact, it does not matter at all, where the value E_0 came from. For this step, the function has to be chosen in such a way that the integral is only running over times where ψ_t is known, i.e. $t_l = t_0$ and $t_r = t_1$. A calculation similar to the one in Eq. (45) shows that the expression $(\mathbb{F}W)_{(0)}$ is complex then. Fortunately, it does not matter, as the state has to be normalized and pinned down to a particular phase anyway.

As one can easily see, the whole procedure is also applicable for excited states and not restricted to the ground state. In that case one would have to replace E_0 with the energy of the state that one is interested in. These energy eigenvalues are also available from the auto-correlation spectrum (46).

For the initial wave function, theoretically any function would do. Practically, one is restricted to wave functions that do never touch the grid boundaries too much, because here they get altered in unphysical ways. Therefore, the best starting point one can think of in the case of a softcore potential, is the analytical solution for the Coulomb potential from Section 3.1. This is shown in Fig. 7 for the case $Z = 8, \alpha = 0.01$. In this plot one can see the energy spectrum of the analytical Coulomb ground state when subjected to this softcore potential. As one can see, apart from the new ground state, it still contains several excited states. The second spectrum illustrates the energies that are contained in the new ground state, that has been extracted using the procedure described above: There is just one peak at the right place.

3.4. Sample bound dynamics

Using the ground state wave function from the previous section, we proceed by presenting a sample application for bound dynamics. In Fig. 8, we have exposed an O^{7+} ion whose single electron was prepared

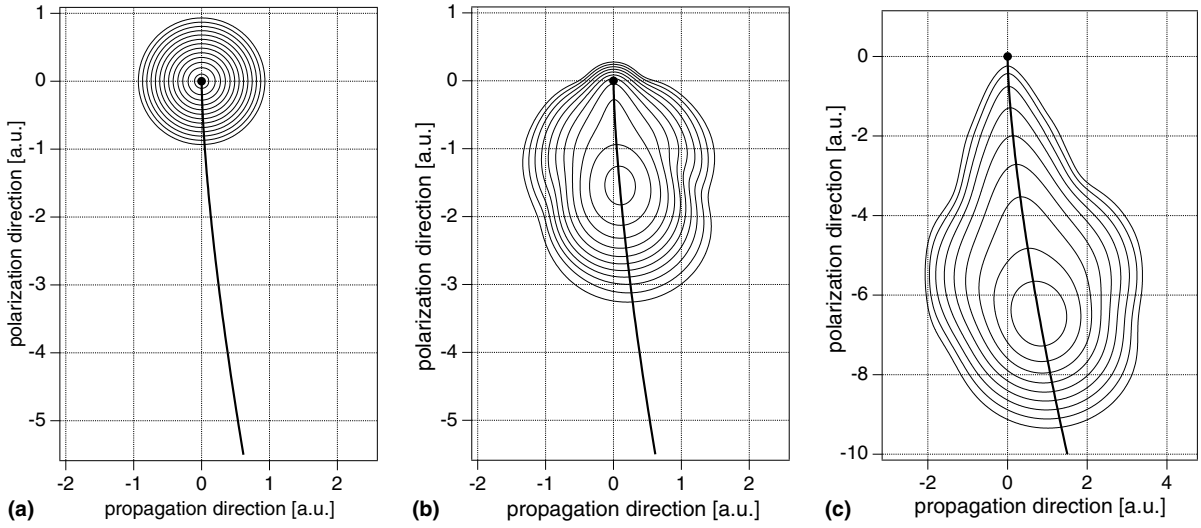

Fig. 8. The electronic wave function corresponding to the softcore ($Z = 8, \alpha = 0.01$) ground state was propagated in a laser field ($E_0 = 218, \omega = 1$ a.u., sudden turn-on) and photographed at the times given in the individual captions. At $t = 0$ a.u. the pulse front reaches the origin, therefore (a) shows the undisturbed ground state, whereas (b) and (c) depict the ionization process. The logarithm of the probability density is given by the contour lines, which start on the outside at -4 and have a spacing of 0.4. The black thick line marks the trajectory of the position operator's expectation value, the black dot indicates the ion's position at the origin. Note the change of scale in (c).

in the ground state as described before, to an electromagnetic field. The origin of the latter is a high intensity laser pulse modelled by a vector potential (22). The ionic core's potential itself is given by (24), with the parameters as indicated in the figure caption. At $t = 0$ a.u. the pulse front reaches the origin, hence Fig. 8(a) shows the undisturbed ground state, whereas Fig. 8(a) and (b) depict the ionization process. As can be seen in these plots, the electric field is so strong as compared to the attractive core potential, that the latter cannot hold back the originally spherically symmetric ground state, which is entirely released to the continuum and only slightly distorted by the influence of the core. This is known as over-the-barrier-ionization (OTBI) [1,2]. Note also that the Lorentz force induces ionization also partly in laser propagation direction.

4. Conclusions

Numerical solutions of the Dirac equation are still a difficult task. The biggest problem remains the high time resolution that is required. Using state-of-the-art computer hardware and the adaptive grid methods described in this work, we are now able to simulate at least a certain class of problems from atomic physics fully quantum-relativistically, i.e. those problems where the wave function stays rather limited in its spatial extent. This is true for free wave packet dynamics and scattering processes of these, undisturbed bound systems, and over the barrier ionization dynamics. We gave example simulations for the latter three. Other interesting problems, such as tunnelling-recollision dynamics, where a fraction of the wave function tunnels out and, before coming back to the core, moves along an elongated trajectory, are still a challenge and require further optimizations.

Acknowledgements

The German Science Foundation (SFB 276) is acknowledged for financial support. We thank Andreas Staudt and Jens Prager for helpful discussions.

Appendix A

A.1. Spacetime algebra

The following sections briefly summarize the most important definitions and expressions that are related to spacetime algebra and that are used throughout this article. For more details, examples and other applications see [25,57–61].

A.1.1. General definitions and formulas

Let $\vec{\sigma}_k$ be the basis vectors of Euclidean $\mathbb{R}^3$. Define a non commutative, associative Clifford product by

$$\vec{\sigma}_k\vec{\sigma}_l + \vec{\sigma}_l\vec{\sigma}_k = 0 \quad \Longleftrightarrow \quad k \neq l, \quad \vec{\sigma}_k^2 = 1 \quad \text{where } k, l \in \{1, 2, 3\}. \tag{A.1}$$

The Clifford algebra over $\mathbb{R}^3$ has elements of degree zero to three and is also called the Pauli algebra. The following table illustrates the basis elements and all possible multiplicative combinations thereof:

Name	Degree	Basis elements	Count
Scalar	0	1	$1 = \binom{3}{0}$
Vector	1	$\vec{\sigma}_1, \vec{\sigma}_2, \vec{\sigma}_3$	$3 = \binom{3}{1}$
Bivector	2	$\vec{\sigma}_2\vec{\sigma}_3 = i\vec{\sigma}_1, \vec{\sigma}_3\vec{\sigma}_1 = i\vec{\sigma}_2,$ $\vec{\sigma}_1\vec{\sigma}_2 = i\vec{\sigma}_3$	$3 = \binom{3}{2}$
Pseudoscalar	3	$\vec{\sigma}_1\vec{\sigma}_2\vec{\sigma}_3 = i$	$1 = \binom{3}{3}$

For the product of any two vectors

$$\vec{a} = a^k\vec{\sigma}_k, \quad \vec{b} = b^k\vec{\sigma}_k, \tag{A.2}$$

one obtains

$$\begin{aligned}\vec{a}\,\vec{b} &= a^1b^1 + a^2b^2 + a^3b^3 + i\Big(\vec{\sigma}_1(a^2b^3 - a^3b^2) + \vec{\sigma}_2(a^3b^1 - a^1b^3) + \vec{\sigma}_3(a^1b^2 - a^2b^1)\Big)\\ &= \vec{a}\cdot\vec{b} + i(\vec{a}\times\vec{b}),\end{aligned} \tag{A.3}$$

if one defines the following products:

$$\vec{a}\cdot\vec{b} \overset{\text{Def.}}{=} \frac{1}{2}(\vec{a}\vec{b} + \vec{b}\vec{a}), \tag{A.4}$$

$$\vec{a}\times\vec{b} \overset{\text{Def.}}{=} \frac{1}{2i}(\vec{a}\vec{b} - \vec{b}\vec{a}). \tag{A.5}$$

We now simply repeat the above procedure, but for another basis. Let γ_ν be the basis vectors of Minkowski spacetime $\mathbb{R}^{1,3}$. Define a non-commutative, associative Clifford product by

$$\begin{aligned}&\gamma_\nu\gamma_\mu + \gamma_\mu\gamma_\nu = 0 \quad \Longleftrightarrow \quad \mu \neq \nu \quad \text{where } \nu, \mu \in \{0,1,2,3\}\\ &\gamma_0^2 = 1, \quad \gamma_k^2 = -1 \quad \text{where } k \in \{1,2,3\}.\end{aligned} \tag{A.6}$$

Then, the following table illustrates the Clifford algebra over $\mathbb{R}^{1,3}$, which is also called Dirac or spacetime algebra:

Name	Degree	Basis elements	Count
Scalar	0	1	$1 = \binom{4}{0}$
Vector	1	$\gamma_0, \gamma_1, \gamma_2, \gamma_3$	$4 = \binom{4}{1}$
Bivector	2	$\gamma_1\gamma_0 = \vec{\sigma}_1, \gamma_2\gamma_0 = \vec{\sigma}_2, \gamma_3\gamma_0 = \vec{\sigma}_3$	$6 = \binom{4}{2}$
Trivector	3	$\gamma_3\gamma_2 = i\vec{\sigma}_1, \gamma_1\gamma_3 = i\vec{\sigma}_2, \gamma_2\gamma_1 = i\vec{\sigma}_3$ $\gamma_1\gamma_2\gamma_3 = \gamma_0 i, \gamma_0\gamma_2\gamma_3 = \gamma_1 i$	$4 = \binom{4}{3}$
Pseudoscalar	4	$\gamma_0\gamma_3\gamma_1 = \gamma_2 i, \gamma_0\gamma_1\gamma_2 = \gamma_3 i$ $\gamma_0\gamma_1\gamma_2\gamma_3 = i$	$1 = \binom{4}{4}$

Here we have used the standard abbreviations

$$\vec{\sigma}_k = \gamma_k\gamma_0, \qquad i = \gamma_0\gamma_1\gamma_2\gamma_3, \qquad i_k = i\vec{\sigma}_k, \tag{A.7}$$

which, of course, are chosen in such a way as to be compatible with the Pauli algebra. An arbitrary multivector A can now be written as a linear combination of all sixteen basis elements using real coefficients $\alpha, \beta, a^\nu, b^\nu, c^k, d^k$:

$$A = \alpha + a^\nu\gamma_\nu + c^k\vec{\sigma}_k + d^k i\vec{\sigma}_k + b^\nu i\gamma_\nu + i\beta. \tag{A.8}$$

Brackets are used to write down projectors

$$\langle A\rangle_r \quad \text{where } r \in \{0,1,2,3,4\}, \tag{A.9}$$

where r designates the vectorial degree (with respect to the Dirac algebra) that one wants to project out. Apart from the most general Clifford product, two more special products of any two arbitrary multivectors A and B are defined in the following way:

$$\langle A\rangle_r \cdot \langle B\rangle_s \overset{\text{Def.}}{=} \langle\langle A\rangle_r\langle B\rangle_s\rangle_{|s-r|}, \tag{A.10}$$

$$\langle A\rangle_r \wedge \langle B\rangle_s \overset{\text{Def.}}{=} \langle\langle A\rangle_r\langle B\rangle_s\rangle_{s+r}. \tag{A.11}$$

The zeroth degree is analogous to the trace in a matrix formalism and has the property

$$\langle AB\rangle_0 = \langle BA\rangle_0. \tag{A.12}$$

The reciprocal vectors γ^ν are defined by the equation

$$\gamma_\mu \cdot \gamma^\nu = \delta_\mu^\nu. \tag{A.13}$$

The following represents an even multivector, as it occurs in the Dirac equation:

$$\psi = \langle\psi\rangle_0 + \langle\psi\rangle_2 + \langle\psi\rangle_4 = \alpha + c^k\vec{\sigma}_k + d^k i\vec{\sigma}_k + i\beta. \tag{A.14}$$

It is a linear combination using eight real coefficients α, β, c^k, d^k. The spacetime position "four"-vector is given in the following way:

$$x = x^\mu\gamma_\mu = (x^0 + x^k\gamma_k\gamma_0)\gamma_0 = (x_0 + \vec{x})\gamma_0. \tag{A.15}$$

Vectors of three-dimensional Euclidean space are bivectors of the Dirac algebra:

$$\vec{r} \equiv \vec{x} = x^k\vec{\sigma}_k = x^k\gamma_k\gamma_0. \tag{A.16}$$

Unit vectors are called $\hat{l}$, $\hat{r}$ and so on. The quantity r designates the real magnitude of the position vector $\vec{r}$ and is not to be mistaken as x, the spacetime vector of degree one.

$$r = \sqrt{\vec{r}^2}, \quad \hat{r} = \frac{\vec{r}}{r}. \tag{A.17}$$

However, both the letters r and x are used for the components, since there is no danger of mistake: $x_\mu \equiv r_\mu$. The derivative of a position dependent multivector field $A(x)$ in direction of a first-degree-vector b is defined by

$$A_b\,. \stackrel{\text{Def.}}{=} \frac{\partial}{\partial\epsilon}A(x+\epsilon b)\bigg|_{\epsilon=0}. \tag{A.18}$$

Building upon this, the vectorial derivative ∂ is defined by

$$\partial A \stackrel{\text{Def.}}{=} \gamma^\mu A_{\gamma_\mu}, \tag{A.19}$$

where on the right hand side, we sum over the index μ as usual. Of course, ∂ can be decomposed into space and time derivatives:

$$\partial = \partial_\mu\gamma^\mu = (\partial_0 + \partial_k\gamma^k\gamma^0)\gamma^0 = (\partial_0 - \partial_k\gamma_k\gamma_0)\gamma_0 = (\partial_0 - \vec{\partial})\gamma_0. \tag{A.20}$$

In the above we have used the abbreviation

$$\partial_k = \frac{\partial}{\partial x^k}. \tag{A.21}$$

Furthermore, there are the two useful operations "$\sim$" (reversion) and "$*$" (spatial inversion) for any arbitrary multivector A. The combined action of both is designated by †:

$$\langle A\rangle_r^\sim = (-1)^{\frac{r(r-1)}{2}}\langle A\rangle_r, \qquad A^* \stackrel{\text{Def.}}{=} \gamma_0 A\gamma_0, \qquad A^\dagger \stackrel{\text{Def.}}{=} \tilde{A}^*. \tag{A.22}$$

In particular:

$$(AB)^\sim = \tilde{B}\tilde{A}, \quad (A+B)^\sim = \tilde{A} + \tilde{B}. \tag{A.23}$$

A.1.2. Hermitian scalar product

The following is a definition of a hermitian scalar product of two even multivector valued functions:

$$(\psi, \phi) \stackrel{\text{Def.}}{=} \int \mathrm{d}^3x\big(\langle\psi^\dagger\phi\rangle_0 - \langle i_3\psi^\dagger\phi\rangle_0 i_3\big). \tag{A.24}$$

Using the above to define expectation values, one easily finds that hermitian operators have real ones, as usual:

$$(\psi, \mathscr{A}\psi) \overset{\mathscr{A}=\mathscr{A}^\dagger}{=} \int \mathrm{d}^3x \big(\langle \psi^\dagger \mathscr{A}\psi \rangle_0 - \langle i_3 \psi^\dagger \mathscr{A}\psi \rangle_0 i_3 \big) = \int \mathrm{d}^3x \langle \psi^\dagger \mathscr{A}\psi \rangle_0. \tag{A.25}$$

A.1.3. Translation table

The following decomposition leads the way to conversion into the standard formulation:

$$\psi = \alpha + E_k\vec{\sigma}_k + B_k i_k + \beta i = \underbrace{(\alpha + B_3 i_3)}_{\Psi_1} + \vec{\sigma}_3 \underbrace{(E_3 + \beta i_3)}_{\Psi_3} + \vec{\sigma}_1 \underbrace{(E_1 + E_2 i_3)}_{\Psi_4} - i_2 \underbrace{(-B_2 + B_1 i_3)}_{\Psi_2}. \tag{A.26}$$

However, when assigning the Ψ_k one has to exchange the i_3 with the ordinary i, in order to arrive at $\Psi \in \mathbb{C}^4$. The above, of course is no proof of equivalence, it is only meant to give an idea of it – and a translation table, too. The complete proof can be found in [25].

A.2. Fourier transforms

Fourier transforms are defined in the usual way. However, since there is more than one element in the algebra whose square gives minus unity, one has to be careful about which of them to use. This will be explained in the following section, as well as some peculiarities that occur when dealing with existing FFT libraries.

A.2.1. Fourier transforms in space

When dealing with Fourier transforms in a quantum mechanical context, i_3 is the best choice, because this leads to momentum space and to plane wave solutions of the Dirac equation.

$$\hat{\psi}_{(\vec{k})} = \frac{1}{(2\pi)^{\frac{3}{2}}} \int \mathrm{d}^3x \psi_{(\vec{x})} \mathrm{e}^{-i_3\vec{k}\cdot\vec{x}}, \qquad \psi_{(\vec{x})} = \frac{1}{(2\pi)^{\frac{3}{2}}} \int \mathrm{d}^3k \hat{\psi}_{(\vec{k})} \mathrm{e}^{i_3\vec{k}\cdot\vec{x}}. \tag{A.27}$$

This can be mapped onto four standard Fourier transforms in the following way:

$$\psi \overset{(A.26)}{=} (\alpha + B_3 i_3) + \vec{\sigma}_3(E_3 + \beta i_3) + \vec{\sigma}_1(E_1 + E_2 i_3) + i_1(B_1 + B_2 i_3). \tag{A.28}$$

The above decomposition is used in the following way: The pairs (α, B_3), (E_3, β), (E_1, E_2), and (B_1, B_2) are considered as ordinary complex numbers and fed into standard FFT libraries.

A.2.2. Fourier transforms in time

We use the following definitions for the Fourier transform $\mathbb{F}$ and its inverse in the time domain:

$$F_{(\omega)} \overset{\text{Def.}}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}t f_{(t)} \mathrm{e}^{-i_3\omega t}, \qquad f_{(t)} \overset{\text{Def.}}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}\omega F_{(\omega)} \mathrm{e}^{+i_3\omega t}. \tag{A.29}$$

In this context, a useful representation of the δ-function is

$$\delta(\omega - \omega') \overset{(A.29)}{=} \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{d}t \mathrm{e}^{-i_3(\omega-\omega')t}. \tag{A.30}$$

The convolution theorem then is

$$\mathbb{F}(fg) \overset{(A.29)}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}\omega' F(\omega') G(\omega - \omega') = \frac{1}{\sqrt{2\pi}} (F * G). \tag{A.31}$$

A.2.3. Fast Fourier transforms from numerical libraries

In this section, we do *not* want to explain the FFT algorithm itself, but only shine some light onto a pretty complicated detail that one has to deal with when using existing FFT libraries [21,56] in physical contexts. The problem is concerned with the region on which the input function $h(t)$ and the output transformed function $H(\omega)$ are defined. Note that in the equations to follow, when employed in the context of all previous sections, the "i" has to be replaced by "i_3". In this section, we stick to standard complex numbers because they are sufficient in this context and the results are easy to generalize.

Suppose that $\mathrm{supp}(h) \subset [t_0, t_1]$ where t_0, t_1 are suitably chosen. Then:

$$\begin{aligned} H(\omega) &\overset{(A.29)}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}t\, h(t) \mathrm{e}^{-i\omega t} = \frac{1}{\sqrt{2\pi}} \int_{t_0}^{t_1} \mathrm{d}t\, h(t) \mathrm{e}^{-i\omega t} \\ &\overset{t'=t-t_0}{=} \frac{1}{\sqrt{2\pi}} \int_0^{t_1-t_0} \mathrm{d}t'\, h(t'+t_0) \mathrm{e}^{-i\omega(t'+t_0)} = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-i\omega t_0} \int_0^{t_1-t_0} \mathrm{d}t'\, h(t'+t_0) \mathrm{e}^{-i\omega t'}. \end{aligned} \tag{A.32}$$

This is usually [56] discretized in the following way:

$$\Delta t = \frac{t_1 - t_0}{N-1} \quad \text{where } N = \text{number of samples}, \tag{A.33}$$

$$t' = n\Delta t, \quad \forall n = 0, \ldots, N-1, \tag{A.34}$$

$$\omega = \frac{2\pi m}{N\Delta t}, \quad \forall m = -\frac{N}{2}, \ldots, +\frac{N}{2}, \tag{A.35}$$

$$\omega_{\max} = \frac{\pi}{\Delta t}. \tag{A.36}$$

Inserting (A.33) and (A.34) into (A.32), one obtains

$$H(\omega) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-i\omega t_0} \sum_{n=0}^{N-1} \Delta t \underbrace{h(n\Delta t + t_0)}_{\overset{\text{Def.}}{=} h_n} \mathrm{e}^{-i\omega n \Delta t} \overset{(A.35)}{=} \frac{\Delta t}{\sqrt{2\pi}} \mathrm{e}^{-i\frac{2\pi m}{N\Delta t} t_0} \underbrace{\sum_{n=0}^{N-1} h_n \mathrm{e}^{-i\frac{2\pi m n}{N}}}_{\overset{\text{Def.}}{=} H'_m}, \tag{A.37}$$

with definitions

$$h_n \overset{\text{Def.}}{=} h(n\Delta t + t_0), \qquad H'_m \overset{\text{Def.}}{=} \sum_{n=0}^{N-1} h_n \mathrm{e}^{-i\frac{2\pi m n}{N}}. \tag{A.38}$$

We want to call

$$H_m \overset{\text{Def.}}{=} H\left(\omega = \frac{2\pi m}{N\Delta t}\right) = \frac{\Delta t}{\sqrt{2\pi}} \mathrm{e}^{-i\frac{2\pi m}{N\Delta t} t_0} H'_m. \tag{A.39}$$

Obviously, we have the relations

$$H'_{m+N} = H'_m, \tag{A.40}$$

$$H'_{-\frac{N}{2}} = H'_{\frac{N}{2}}. \tag{A.41}$$

Usual implementations of the Fourier transform simply compute the formula for H'_m (A.38) without caring too much for the mappings $n \to t$ and $m \to \omega$. Even worse, just to avoid negative indices and to improve code efficiency, they make use of (A.40). Negative frequencies, however, correspond to oscillations in the opposite sense and are physically important. Often, a symmetrical frequency domain $-\omega_{\max}, \ldots, +\omega_{\max}$ is desired. Instead of (A.35), standard libraries usually take $m = 0, \ldots, N-1$ and demand to shift the negative indices upwards by $+N$. After all, they calculate

$$H_k^{\text{lib}} \overset{\text{Def.}}{=} \sum_{n=0}^{N-1} h_n \mathrm{e}^{-i\frac{2\pi kn}{N}}, \quad \forall k = 0, \ldots, N-1. \tag{A.42}$$

A physically logical mapping of the frequencies is such, that $j = 0, \ldots, N-1$ corresponds directly to $\omega \in [-\omega_{\max}, \omega_{\max}]$, or in other words

$$m = j - \frac{N}{2} \quad \Rightarrow \quad \omega \overset{(A.35)}{=} \frac{2\pi\left(j - \frac{N}{2}\right)}{N\Delta t}, \quad \forall j = 0, \ldots, N-1. \tag{A.43}$$

Therefore define

$$H''_j \overset{\text{Def.}}{=} H_{j-\frac{N}{2}} \overset{(A.39)}{=} \frac{\Delta t}{\sqrt{2\pi}} \mathrm{e}^{-i\frac{2\pi\left(j-\frac{N}{2}\right)}{N\Delta t} t_0} H'_{j-\frac{N}{2}}, \quad \forall j = 0, \ldots, N-1. \tag{A.44}$$

Furthermore define the index mapping function

$$f(j) \overset{\text{Def.}}{=} \begin{cases} j + \frac{N+1}{2}, & 0 \leqslant j < \frac{N}{2}, \\ j - \frac{N}{2}, & \frac{N}{2} \leqslant j < N, \end{cases} \tag{A.45}$$

where we understand all divisions as integer divisions without modulus. The above is valid for both even and odd N. For even N, ± 1 can be ignored in the nominator of the above and the following. As defined, the map $f : [0, N-1] \to [0, N-1]$ effectively acts on both domains as follows:

$$\begin{aligned} f\left(\left[0, \frac{N}{2} - 1\right]\right) &= \left[\frac{N+1}{2}, N-1\right], \\ f\left(\left[\frac{N}{2}, N-1\right]\right) &= \left[0, \frac{N-1}{2}\right], \end{aligned} \tag{A.46}$$

i.e. small j (corresponding to negative frequencies) are shifted upwards, and large j (corresponding to positive frequencies) vice versa. If we set $k = f(j)$ and use (A.42), we are now able to directly address the desired frequencies inside the array that is returned as a result from the library. Hence, set

$$H'_{j-\frac{N}{2}} = H^{\text{lib}}_{f(j)} \quad \forall j = 0, \ldots, N-1. \tag{A.47}$$

Altogether we have

$$H''_j \overset{(A.47)}{=} \frac{\Delta t}{\sqrt{2\pi}} \mathrm{e}^{-i\frac{2\pi\left(j-\frac{N}{2}t_0\right)}{N\Delta t}} H^{\text{lib}}_{f(j)}. \tag{A.48}$$

Summing up all the above: If one supplies the samples h_n $(n = 0, \ldots, N-1)$ which correspond to the times $t = t_0 + n\frac{t_1 - t_0}{N-1}$ as input to a standard FFT library, and if this library returns the result in the form H_k^{lib} $(k = 0, \ldots, N-1)$, then the following does the proper mapping to physically relevant quantities:

$$h_n \overset{(A.38)}{=} h(t = n\Delta t + t_0), \quad \forall n = 0, \ldots, N-1, \tag{A.49}$$

$$H_k^{\text{lib}} \overset{(A.42)}{=} \sum_{n=0}^{N-1} h_n \, \mathrm{e}^{-i\frac{2\pi kn}{N}}, \quad \forall k = 0, \ldots, N-1, \tag{A.50}$$

$$\begin{aligned} &H\left(\omega = \frac{2\pi\left(j - \frac{N}{2}\right)}{N\Delta t}\right) = \frac{\Delta t}{\sqrt{2\pi}} \mathrm{e}^{-i\frac{2\pi\left(j-\frac{N}{2}\right)t_0}{N\Delta t}} H_{f(j)}^{\text{lib}}, \quad \forall j = 0, \ldots, N-1 \\ &\text{where } f(j) \overset{(A.45)}{=} \begin{cases} j + \frac{N+1}{2}, & 0 \leqslant j < \frac{N}{2}, \\ j - \frac{N}{2}, & \frac{N}{2} \leqslant j < N, \end{cases} \end{aligned} \tag{A.51}$$

For the inverse transform, it is reasonable to go the same way backwards inside the code instead of tediously working out the inverse of the above relation:

$$\begin{aligned} &H_{f(j)} = \frac{\sqrt{2\pi}}{N\Delta t} \mathrm{e}^{i\frac{2\pi\left(j-\frac{N}{2}\right)t_0}{N\Delta t}} H\left(\omega = \frac{2\pi\left(j - \frac{N}{2}\right)}{N\Delta t}\right), \quad \forall j = 0, \ldots, N-1 \\ &\text{where again } f(j) = \begin{cases} j + \frac{N+1}{2}, & 0 \leqslant j < \frac{N}{2}, \\ j - \frac{N}{2}, & \frac{N}{2} \leqslant j < N, \end{cases} \end{aligned} \tag{A.52}$$

$$h_k \overset{(A.38)}{=} h(t = k\Delta t + t_0) = \sum_{n=0}^{N-1} H_n \, \mathrm{e}^{i\frac{2\pi kn}{N}}, \quad \forall k = 0, \ldots, N-1. \tag{A.53}$$

A generalization to two or more dimensions corresponding to transformations $\vec{x} \leftrightarrow \vec{k}$ in the space/momentum domain instead of $t \leftrightarrow \omega$ and the time/frequency domain is easily possible. The above formulation also allows for a simple movement of the numerical grid by adjusting the value t_0: If one wants to move a function, that is defined on a grid in the range $[t_0, t_0 + T]$ to another grid range $[t_0', t_0' + T]$, then one can carry out the Fourier transform according to (A.51) followed by an inverse transform (A.53) that incorporates the replacement $t_0 \to t_0'$, and obtain the desired result. At first sight, this certainly seems to be the most awkward way to accomplish this task. But in the context of the split-operator scheme, where all the Fourier transforms have to be carried out anyway, the *additional* cost of it is almost zero – or at least far less than for any copy-transfer algorithm used instead.

References

[1] M. Protopapas, C.H. Keitel, P.L. Knight, Atomic physics with super-high intensity lasers, Rep. Prog. Phys. 60 (1996) 389–486.
[2] C.J. Joachain, M. Dörr, N. Kylstra, High-intensity laser-atom physics, Adv. At. Mol. Opt. Phys. 42 (2000) 225–286.
[3] C.H. Keitel, Relativistic quantum optics, Contemp. Phys. 42 (2001) 353–363.
[4] A. Maquet, R. Grobe, Atoms in strong laser fields: challenges in relativistic quantum mechanics, J. Mod. Opt. 49 (2002) 2001–2018.
[5] O. Latinne, C.J. Joachain, M. Dörr, Atomic hydrogen in a superintense high-frequency field: testing the dipole approximation, Europhys. Lett. 26 (5) (1994) 333–338.
[6] J.R.V. de Aldana, L. Roso, Nonrelativistic numerical study of atomic ionization by strong laser fields without the dipole approximation in a flat-atom model, Phys. Rev. A 61 (2000) 0434031.

[7] J. Prager, C.H. Keitel, Laser-induced nonsequential double ionization approaching the relativistic regime, J. Phys. B: At. Mol. Opt. Phys. 35 (2002) L167–L174.
[8] M. Protopapas, C.H. Keitel, P.L. Knight, Relativistic mass shift effect in adiabatic intense laser field stabilization of atoms, J. Phys. B: At. Mol. Opt. Phys. 29 (1996) L591–L598.
[9] S.X. Hu, C.H. Keitel, Spin signatures in intense laser-ion interaction, Phys. Rev. Lett. 83 (23) (1999) 4709–4712.
[10] S.X. Hu, C.H. Keitel, Dynamics of multiply charged ions in intense laser fields, Phys. Rev. A 63 (2001) 053402.
[11] N.J. Kylstra, A.M. Ermolaev, C.J. Joachain, Relativistic effects in the time evolution of a one-dimensional model atom in an intense laser field, J. Phys. B: At. Mol. Opt. Phys. 30 (1997) L449–L460.
[12] J.W. Braun, Q. Su, R. Grobe, Numerical approach to solve the time-dependent Dirac equation, Phys. Rev. A 59 (1999) 604–612.
[13] M. Casu, C.H. Keitel, Resonant multiphoton X-ray optics with highly charged ions, Europhys. Lett. 58 (2002) 496–502.
[14] U.W. Rathe, C.H. Keitel, M. Protopapas, P.L. Knight, Intense laser-atom dynamics with the two-dimensional Dirac equation, J. Phys. B: At. Mol. Opt. Phys. 30 (1997) L531–L539.
[15] U.W. Rathe, P. Sanders, P.L. Knight, A case study in scalability: an ADI method for the two-dimensional time-dependent Dirac equation, Parallel Comput. 25 (1999) 525–533.
[16] Q. Su, B.A. Smetanko, R. Grobe, Relativistic suppression of wave packet spreading, Opt. Express 2 (1997) 277–281.
[17] J.S. Román, L. Plaja, L. Roso, Relativistic quantum dynamics of a localized Dirac electron driven by an intense-laser-field pulse, Phys. Rev. A 64 (2001) 063402.
[18] J.S. Román, L. Roso, H.R. Reiss, Evolution of a relativistic wavepacket describing a free electron in a very intense laser field, J. Phys. B: At. Mol. Opt. Phys. 33 (2000) 1869–1880.
[19] P. Panek, J.Z. Kamiński, F. Ehlotzky, Relativistic electron-atom scattering in an extremely powerful laser field: relevance of spin effecs, Phys. Rev. A 65 (2002) 033408.
[20] H.R. Reiss, Relativistic strong-field photoionization, J. Opt. Soc. Am. B 7 (1990) 574–586.
[21] M. Frigo, S.G. Johnson, FFTW: an adaptive software architecture for the FFT, in: Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing, vol. 3, IEEE, 1998, pp. 1381–1384.
[22] G.R. Mocken, C.H. Keitel, Bound atomic dynamics in the MeV regime, submitted for publication.
[23] G.R. Mocken, C.H. Keitel, Quantum signatures in laser-driven relativistic multiple-scattering, Phys. Rev. Lett. 91 (2003) 173202.
[24] J. Bjorken, S.D. Drell, Relativistic Quantum Mechanics, Mc Graw-Hill, New York, 1964.
[25] D. Hestenes, Observables, operators and complex numbers in the Dirac theory, J. Math. Phys. 16 (1975) 556–572.
[26] V. Alonso, S.D. Vincenzo, L. Mondino, On the boundary conditions for the Dirac equation, Eur. J. Phys. 18 (1997) 315–320.
[27] J.A. Fleck, J.R. Morris, M.D. Feit, Time-dependent propagation of high energy laser beams through the atmosphere, Appl. Phys. 10 (1976) 129–160.
[28] R. Heather, An asymptotic wavefunction splitting procedure for propagating spatially extended wavefunctions: application to intense field photodissociation of H_2^+, Comput. Phys. Commun. 63 (1991) 446–459.
[29] A.D. Bandrauk, H. Shen, Improved exponential split operator methods for solving the time-dependent Schrödinger equation, Chem. Phys. Lett. 176 (5) (1991) 428–432.
[30] A.D. Bandrauk, H. Shen, Exponential split operator methods for solving coupled time-dependent Schrödinger equations, J. Chem. Phys. 99 (2) (1993) 1185–1193.
[31] D. Neuhauser, M. Baer, The time-dependent Schrödinger equation: application of absorbing boundary conditions, J. Chem. Phys. 90 (8) (1988) 4351–4355.
[32] J.L. Krause, K.J. Schafer, K.C. Kulander, Calculation of photoemission from atoms subject to intense laser fields, Phys. Rev. A 45 (1997) 4998–5010.
[33] S.X. Hu, W.X. Qu, Z.Z. Xu, Enhanced ionization of H_3^{2+} in ultra-short intense laser fields, J. Phys. B: At. Mol. Opt. Phys. 31 (1998) 1523–1531.
[34] E.S. Smyth, J.S. Parker, K.T. Taylor, On the boundary conditions for the Dirac equation, Comput. Phys. Commun. 114 (1998) 1–14.
[35] R. Grobe, S.L. Haan, J.H. Eberly, A split-domain algorithm for time-dependent multi-electron wave functions, Comput. Phys. Commun. 117 (1999) 200–210.
[36] A.W. Thomas, Chiral symmetry and the bag model: a new starting point for nuclear physics, Adv. Nucl. Phys. 13 (1984) 1–137.
[37] J. Javanainen, J.H. Eberly, Q. Su, Numerical simulations of multiphoton ionization and above-threshold electron spectra, Phys. Rev. A 38 (7) (1988) 3430–3446.
[38] H.A. Kramers, Collected Scientific Papers, North-Holland Publishing Company, Amsterdam, 1956.
[39] W.C. Henneberger, Pertubation method for atoms in intense light beams, Phys. Rev. Lett. 21 (12) (1968) 838–841.
[40] Y.I. Salamin, F.H.M. Faisal, Harmonic generation by superintense light scattering from relativistic electrons, Phys. Rev. A 54 (1996) 4383–4395.
[41] V.B. Berestetskii, E.M. Lifshitz, L.P. Pitaevskii, Relativistic quantum theory (part 1), in: Course of Theoretical Physics (Landau and Lifshitz), Vol. 4, Pergamon Press, Oxford, 1971.

[42] W. Gordon, Die Energieniveaus des Wasserstoffatoms nach der Diracschen Quantentheorie des Elektrons, Z. Phys. 48 (1928) 11–14.
[43] H.A. Bethe, E.E. Salpeter, Quantum Mechanics of One- and Two-electron Atoms, Plenum Publishing Corporation, New York, 1977.
[44] C.G. Darwin, The wave equations of the electron, Proc. Roy. Soc. Lond., Ser. A 118 (1928) 654–680.
[45] H. Krüger, Quantenmechanik II, Vorlesungsskript, Universität Kaiserslautern (Sommersemester 1990).
[46] C.-L. Ho, V.R. Khalilov, Planar Dirac electron in Coulomb and magnetic fields, Phys. Rev. A 61 (2000) 032104.
[47] V.R. Khalilov, C.-L. Ho, Dirac electron in a Coulomb field in (2+1) dimensions, Mod. Phys. Lett. A 13 (1998) 615–622.
[48] P. Falsaperla, G. Fonte, Two methods for solving the Dirac equation without variational collapse, Phys. Rev. A 56 (2) (1997) 1240–1248.
[49] W.E. Baylis, S.J. Peel, Stable variational calculations with the Dirac Hamiltonian, Phys. Rev. A 28 (4) (1983) 2552–2554.
[50] G. Aissing, Monte Carlo variance minimization for the Dirac equation, Phys. Rev. A 44 (5) (1991) R2765–R2768.
[51] H. Wallmeier, W. Kutzelnigg, Basis-set expansion of the Dirac equation without variational collapse: Numerical test of the forth-back free-particle Foldy–Wouthuysen transformation, Phys. Rev. A 28 (5) (1983) 3092–3094.
[52] L.L. Foldy, S.A. Wouthuysen, On the Dirac theory of spin $\frac{1}{2}$ particles and its non-relativistic limit, Phys. Rev. 78 (1) (1950) 29–36.
[53] J.R. Hiller, I.D. Johnston, D.F. Styer, Quantum Mechanics Simulations: The Consortium for Upper Level Physics Software, 1st edition, John Wiley & Sons, Inc, New york, 1994.
[54] M.D. Feit, J.A. Fleck, A. Steiger, Solution of the Schrödinger equation by a spectral method, J. Comput. Phys. 47 (1982) 412–433.
[55] S. Goldman, Comment on "Finite-basis-set variational solution of the Dirac equation", Phys. Rev. A 41 (11) (1990) 6526–6528.
[56] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerial recipes in C, The Art of Scientific Computing, Cambridge University Press, Cambridge, 1992.
[57] D. Hestenes, Proper particle mechanics, J. Math. Phys. 15 (10) (1974) 1768–1777.
[58] D. Hestenes, Proper dynamics of a rigid point particle, J. Math. Phys. 15 (10) (1974) 1778–1786.
[59] W.E. Baylis, Classical eigenspinor and the Dirac equation, Phys. Rev. A 45 (7) (1992) 4293–4302.
[60] W.E. Baylis, Y. Yao, Relativistic dynamics of charges in electromagnetic fields: an eigenspinor approach, Phys. Rev. A 60 (2) (1999) 785–795.
[61] H. Krüger, Classical limit of real Dirac theory: quantization of relativistic central field orbits, Foundations Phys. 23 (1993) 1265–1288.