Krylov subspace methods for the Dirac equation

Randolf Beuerwerth and Heiko Bauke
Max-Planck-Institut für Kernphysik, Saupfercheckweg 1, 69117 Heidelberg, Germany

The Lanczos algorithm is evaluated for solving the time-independent as well as the time-dependent Dirac equation with arbitrary electromagnetic fields. We demonstrate that the Lanczos algorithm can yield very precise eigenenergies and allows very precise time propagation of relativistic wave packets. The Dirac Hamiltonian’s property of not being bounded does not hinder the applicability of the Lanczos algorithm. As the Lanczos algorithm requires only matrix-vector and inner products, which both can be efficiently parallelized, it is an ideal method for large-scale calculations. The excellent parallelization capabilities are demonstrated by a parallel implementation of the Dirac Lanczos propagator utilizing the Message Passing Interface standard.

PACS numbers: 02.70.-c, 03.65.Pm, 02.70.Bf

1. Introduction

The Dirac equation is the fundamental equation of motion for describing the quantum evolution of a charged spin one-half particle in a Lorentz invariant manner. It predicts the existence of antimatter and finds its application not only in the growing field of light-matter interactions at relativistic intensities [1, 2] but also in condensed matter theory of graphene [3] and in relativistic quantum information [4–6]. Deducing analytical solutions of this equation, however, poses a major problem. Analytical methods for determining solutions of the Dirac equation usually require physical setups with a high degree of symmetry [7–11]. Thus, approximations or numerical methods have to be applied. In the past, various numerical schemes have been developed to solve the time-dependent Dirac equation numerically, including Fourier split operator approaches [12–15], real space split operator methods based on the method of characteristics [16, 17], finite differences [18], finite elements [19], or employing either spherical harmonics or plane waves as basis functions and integrating the resulting ordinary differential equations [20, 21]. Complementary to the mentioned numerical methods quantum simulations of the Dirac equation [22] are also an active field of current research.

State of the art large scale calculations of the time-dependent Schrödinger equation [23, 24], which is the nonrelativistic limit of the Dirac equation, often utilize Krylov subspace methods and in particular the Lanczos algorithm [25, 27]. Although the Lanczos algorithm was known since the 1950s, this approach gained popularity not until the 1980s when it was applied to time-independent [23] and time-dependent [29, 30] problems of nonrelativistic quantum mechanics. Krylov subspace methods have the virtue that their applicability does not depend on the manner how time and space are discretized. Furthermore, they are rather easy to parallelize and therefore suitable for today’s prevalent parallel hardware architectures like compute clusters [31] or high-performance graphics cards [32]. As the Dirac equation and the Schrödinger equation share the same Hermitian structure, it appears natural to apply Krylov subspace methods also to relativistic quantum mechanical problems. Thus, it is the purpose of this article to evaluate the performance of the Lanczos algorithm when it is applied to the relativistic Dirac equation.

This paper is organized as follows. In order to make the paper self-contained and to introduce some notations we characterize the Lanczos algorithm in section 2 and show how it can help to solve Hermitian eigenvalue problems. In section 3 we will briefly cover the Dirac equation. Furthermore, we will explain in section 4 how the time evolution operator can be approximated using the Lanczos algorithm. Numerical results are presented in section 5 for the relativistic eigenproblem and in section 6 for the time-dependent Dirac equation. Finally, we will present our parallel implementation of the Lanczos Dirac propagator and show some benchmark results in section 7.

2. The Lanczos algorithm

Before we will briefly summarize the Lanczos algorithm and its properties we have to introduce the notion of a Krylov subspace. A Krylov subspace of dimension $N$ of a matrix $A$ is spanned by the successive powers of $A$ applied to some given vector $b$

$$K_N(A, b) = \text{span}\left\{b, Ab, A^2b, \ldots, A^{N-1}b\right\}, \quad 1 \leq k \leq N.$$  \hspace{1cm} (1)

For every Hermitian matrix $A$ there is a unitary transformation that turns the matrix into tridiagonal form, that is,

$$Q^*AQ = T = \begin{pmatrix}
\alpha_1 & \beta_1 & & \\
\beta_1 & \alpha_2 & \beta_2 & \\
& \ddots & \ddots & \ddots \\
& & \beta_{N-2} & \alpha_{N-1} & \beta_{N-1} \\
& & & \beta_{N-1} & \alpha_N
\end{pmatrix}. \hspace{1cm} (2)
$$

The unitary transformation matrix $Q$ and the tridiagonal matrix $T$ can be determined via the Lanczos algorithm. The columns of the matrix $Q$ are conveniently labeled $q_i$, and are called the Lanczos vectors. The vectors $q_1$ to $q_k$ form an orthonormal
which is frequently called the Lanczos relation, holds, where

\[ z = A q_i, \]

for \( i = 1 \) to \( k - 1 \) do

\[ \alpha_i = q_i \cdot z, \]
\[ z = z - \alpha_i q_i, \]
\[ \beta_i = ||z||, \]
\[ q_{i+1} = z/\beta_i, \]
\[ z = A q_{i+1} - \beta_i q_i, \]
end for

\[ \alpha_k = q_k \cdot z. \]

These equations correspond to a classical Gram-Schmidt orthogonalization. One can show that due to the Hermiticity of \( A \), the newly constructed vector \( q_{i+1} \) is automatically orthogonal to all previous Lanczos vectors, except the last two. Therefore, the classical Gram-Schmidt orthogonalization reduces to the subtraction of the contributions from two previous vectors, shown in equation (3b).

Due to the well-known stability issues of the classical Gram-Schmidt algorithm, Paige [33] suggested to replace (3) by a modified Gram-Schmidt orthogonalization against the last two previous vectors. With the modified Gram-Schmidt orthogonalization equation (3a) becomes

\[ \alpha_i = q_i \cdot (A q_i - \beta_{i-1} q_{i-1}), \]

which is in exact arithmetic equivalent to (3a) because \( q_i \) and \( q_{i-1} \) are orthogonal. This leads to the Lanczos algorithm as shown in Fig. 1. Using an arbitrary nonzero starting vector \( b \) it calculates the columns of the matrix \( Q \) iteratively such that (2) is fulfilled. After each iteration with \( 1 < k < N \) the relation

\[ AQ^{(k)} = Q^{(k)}T^{(k)} + [0, \ldots, 0, \beta_k q_{k+1}] \]

which is frequently called the Lanczos relation, holds, where \( Q^{(k)} \) is formed by the first \( k \) column vectors \( q_i \) and \( T^{(k)} \) is the symmetric tridiagonal matrix formed by \( \alpha_1 \) to \( \alpha_k \) and \( \beta_1 \) to \( \beta_{k-1} \). Because the Lanczos algorithm in Fig. 1 performs only matrix-vector products (and inner products) it is sufficient if the action of the matrix \( A \) on some vector can be computed. It is not necessary to store the matrix elements of \( A \) explicitly, which is a major advantage of the Lanczos algorithm.

Since the transformation \( Q \) is unitary, the matrices \( A \) and \( T \) are similar and therefore they have the same set of eigenvalues and eigenvectors \( a_i \) of \( A \) are related to the eigenvectors \( t_i \) of \( T \) via

\[ a_i = Qt_i. \]

In many applications it is sufficient to know some eigenvalues and eigenvectors of \( A \). Because the matrix \( T^{(k)} \) is the representation of \( A \) in the Krylov subspace \( K_k(A, b) \) and as a consequence of the Lanczos relation (2) the approximation

\[ A \approx Q^{(k)}T^{(k)}Q^{(k)\dagger} \]

holds. Thus, some of the eigenvalues \( \lambda_i \) of \( A \) may be approximated by the eigenvalues \( \lambda_i^{(k)} \) of \( T^{(k)} \) with \( k \ll N \), that is when the Lanczos iteration is stopped after \( k \) iterations. Approximate eigenvectors of \( A \) can be obtained by

\[ a_i \approx a_i^{(k)} = Q^{(k)}t_i^{(k)}. \]

The error of the approximate eigenvalues is bounded by [34]

\[ \Delta \lambda_i^{(k)} = \min_j |t_j^{(k)} - \lambda_i| \leq |\beta_i t_i^{(k)}|, \]

where \( t_i^{(k)} \) denotes the \( k \)th (the last) component of the \( i \)th eigenvector of the matrix \( T^{(k)} \). Furthermore, \( |\beta_i t_i^{(k)}| \) equals the residual of the \( i \)th eigenvector, that is,

\[ \|A a_i^{(k)} - \lambda_i a_i^{(k)}\|_2 = |\beta_i t_i^{(k)}|, \]

where \( \|\cdot\|_2 \) denotes the Euclidian vector norm. The error estimate (9) allows one to monitor the error after each iteration and to stop as soon as a sufficient accuracy has been reached.

A notable deficiency of the simple Lanczos algorithm as presented here is its numerical instability. In exact arithmetic the matrix \( Q \) is unitary. This property, however, is lost in floating point arithmetic when rounding errors occur. Paige’s theorem [34] shows that the Lanczos vector \( q \) looses its orthogonality with respect to the other Lanczos vectors as the corresponding eigenvalue converges. Thus, convergence comes at the price of loss of orthogonality. The numerical stability can, however, be increased by applying reorthogonalization. This means, the orthogonalization against the two previous Lanczos vectors is replaced by an orthogonalization against all previous Lanczos vectors using either the classical Gram-Schmidt algorithm or the modified Gram-Schmidt algorithm. This is still not unconditionally stable. Unconditional stability is ensured by applying orthogonalization twice [35]. This extension of the Lanczos algorithm is usually called the Lanczos algorithm with full reorthogonalization. In the following, it will be used to demonstrate the convergence behavior and to demonstrate that degenerate states are found correctly. An alternative to the very expensive full reorthogonalization is partial reorthogonalization [36]. All reorthogonalization approaches, however, have the common disadvantage that they require all Lanczos vectors to be stored. If no reorthogonalization is applied, storage of three vectors is sufficient to determine the approximate eigenvalues. If, however, also the approximate eigenvectors \( a_i^{(k)} \) are required then the whole matrix \( Q^{(k)} \) has to be stored.

Alternatively one can run the Lanczos algorithm a second time to calculate the approximate eigenvectors \( a_i^{(k)} \) after the vectors \( t_i^{(k)} \) have been determined. The application of the matrix \( Q^{(k)} \) to the vector \( t_i^{(k)} \) is calculated while performing the second Lanczos iteration.
3. The Dirac equation

The Dirac equation

\[ i\hbar \frac{\partial \Psi(x,t)}{\partial t} = \hat{H}(t)\Psi(x,t) \]  

(11)
describes a relativistic spin one-half particle with rest mass \( m \) and charge \( q \) moving in the electromagnetic potentials \( A(x,t) \) and \( \phi(x,t) \). The in general time-dependent Hamiltonian \( \hat{H}(t) \) is in \( d \) spatial dimensions given by

\[ \hat{H}(t) = e \sum_{i=1}^{d} \alpha_i (\hat{p}_i - qA_i(x,t)) + \beta mc^2 + q\phi(x,t), \]  

(12)

where \( \hat{p}_i = -i\hbar \partial_{x_i} \) and \( A_i(x,t) \) denote the components of the canonical momentum operator and the vector potential, respectively, and \( \alpha \) and \( \beta \) are the Dirac matrices. These matrices obey the Dirac algebra

\[ \begin{align*}
\alpha^2 = &\beta^2 = 1, \\
&\alpha\beta + \beta\alpha = 0, \\
&\alpha_i\alpha_j + \alpha_j\alpha_i = 2\delta_{ij},
\end{align*} \]

(13a)

(13b)

(13c)

for \( i, j = 1, \ldots, d \). In one dimension the Dirac matrices are given by \( 2 \times 2 \) matrices. The choice \( \alpha_1 = \sigma_1 \) and \( \beta = \sigma_3 \) with \( \sigma_1 \), \( \sigma_2 \), and \( \sigma_3 \) denoting the three Pauli matrices is the standard representation for one-dimensional systems. In three space dimensions, however, \( 4 \times 4 \) matrices are required with the standard representation

\[ \alpha_i = \begin{pmatrix} 0 & \sigma_i \\ \sigma_i & 0 \end{pmatrix}, \quad \beta = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \]

(14)
The resulting \( 2 \times 2 \) Pauli matrices also fulfill the Dirac algebra in two dimensions with \( \alpha_1 = \sigma_1 \), \( \alpha_2 = \sigma_2 \), and \( \beta = \sigma_3 \). The resulting two-dimensional Dirac equation does, however, not include the spin degree of freedom. To incorporate the electron spin into a two-dimensional Dirac equation the \( 4 \times 4 \) matrices \([14]\) have to be employed. Because the Dirac Hamiltonian is a matrix operator Dirac wave functions have two or four complex components. As in the nonrelativistic case the eigenvalue equation resulting from the time-dependent problem reads

\[ \hat{H}\Psi(x) = E\Psi(x). \]

(15)
The spectrum of a free particle consists of a positive and a negative continuum \( (-\infty,-mc^2) \cup [mc^2, \infty) \), which are usually associated to particles and anti-particles.

A formal solution of the time-dependent problem of the Dirac equation (11) can be given using the time-ordering operator. For numerical calculations we neglect time ordering and approximate the time evolution operator \( U(t,t+\Delta t) \) in first order [37], which results in

\[ \Psi(x, t + \Delta t) = U(t, t + \Delta t)\Psi(x, t) \approx \exp \left( -\frac{i}{\hbar} \int_t^{t+\Delta t} \hat{H}(t') dt' \right) \Psi(x, t) + O(\Delta t^2). \]

(16)

By discretization of the wave function \( \Psi(x, t) \) the Hamiltonian \( \hat{H}(t) \) becomes a matrix. Thus, the numerical propagation of the wave function \( \Psi(x, t) \) by one time step \( \Delta t \) requires the calculation of a matrix exponential, followed by the application to the state vector. Similarly, the numerical solution of the time-independent problem [15] involves the computation of the eigenvalues and the eigenvectors of a matrix.

4. Time propagation and matrix exponentials

The standard method for computing the exponential of a Hermitian matrix is to calculate its eigendecomposition. Then, the matrix exponential in the space of the matrix’ eigenvectors reduces to exponentials of the eigenvalues. Typical matrices that result from the discretization of partial differential equations, however, are so large that full diagonalization is not feasible. Therefore, the matrix exponential is calculated approximately in a Krylov subspace of dimension much smaller than the dimension of the original matrix.

Let \( A \) now denote the Hermitian matrix that results from the discretization of the operator \( \int_t^{t+\Delta t} \hat{H}(t') dt'/\hbar \) and \( \phi(t) \) the corresponding discrete representation of the wave function \( \Psi(x, t) \). Utilizing the approximation (7) the exponential of the antihermitean matrix \( -iA \) applied to the vector \( \psi \) can be calculated approximately in the Krylov subspace \( K_k(A, \psi(t)) \) as

\[ \psi(t + \Delta t) = \exp(-iA)\psi(t) \approx ||\psi(t)||_2 Q^{(k)} \exp(-iT^{(k)}) e_1, \]

(17)

where \( e_1 \) denotes the \( k \)-dimensional unit vector \((1,0,0,\ldots)^T\). The remaining matrix exponential of the matrix \( T^{(k)} \) can be computed by performing a numerical eigendecomposition using any standard method optimized for tridiagonal real symmetric matrices. This diagonalization is not very expensive because the dimension of \( T^{(k)} \) is chosen to be very small compared to the dimension of \( A \). Note that it is required not to include the factor \(-i\) into the matrix \( A \) in order to ensure the Hermiticity of \( A \) and thus the applicability of the Lanczos algorithm. Note that [17] involves all Lanczos vectors \( q_1 \) to \( q_k \). This means that these vectors have to be stored and can not be discarded when no longer needed in the Lanczos algorithm. In order to save memory one can run the Lanczos algorithm a second time when the matrix \( Q^{(k)} \) is multiplied to the vector \( \exp(-iT^{(k)}) e_1 \).

The error that is introduced by the approximation (17) may be estimated as [38]

\[ ||\Delta\psi||_2 \approx \beta_k ||\psi(t)||_2 ||e_k^T \exp(-iT^{(k)}) e_1||, \]

(18)

where \( \Delta\psi \) is the residual vector \( \psi(t + \Delta t) - \exp(-iA)\psi(t) \). This error estimate can be used to monitor the accuracy of a calculation and to adjust the size of the time step \( \Delta t \) or the dimension of the Krylov subspace \( k \) adaptively. However, the number of Lanczos iterations should be kept on a moderate level since the orthogonality of the Lanczos vectors may be lost when...
As an illustrative example for the calculation of eigenstates we consider the two-dimensional soft-core potential with

\[ q\phi(r) = -\frac{3}{2} \frac{Ze^2}{4\pi\varepsilon_0 \sqrt{r^2 + Z^2 / 2}}. \]  

(19)

For the remainder of this article atomic units will be employed. In this system of units the electron mass \( m \), the elementary charge \( e \), the Bohr radius \( a_0 \), and the reduced Planck constant \( \hbar \) and as a consequence \( 1/(4\pi\varepsilon_0) \) are unity. The speed of light equals the inverse fine structure constant and energy is measured in units of hartree, 1 au = 27.211 385 eV = 1 \( E_\text{F} \). The magnetic field is measured in units of 1 au = 2.350 52 \( \times 10^3 \) T. The soft-core parameters in (19) are chosen such that the ground-state energy of this potential yields in case of the nonrelativistic Hamiltonian, which is not included in the estimate (18).

Furthermore, the total error in solving the time-dependent Dirac equation depends also on the discretization of the Hamiltonian, which is not included in the estimate (18).

**5. Time-independent problems**

Paige \[39, 40\] demonstrated that despite the loss of orthogonality the Lanczos algorithm is a capable method to find few of the extreme eigenvalues of large Hermitian matrices. This makes it the ideal tool for computing the bound states of nonrelativistic Hamiltonians, which lie at the lower end of the energy spectrum. In case of the Dirac equation, however, the bound states appear inside the band gap in the middle of the spectrum. Cullum and Willoughby \[41\] pointed out that the convergence rate crucially depends on the gap structure of the eigenvalue spectrum. The \( \lambda^k \) converge particularly fast to eigenvalues of the matrix A that are well-separated from other eigenvalues. Thus, well-separated interior eigenvalues may converge faster than clustered extreme eigenvalues. In this section we will demonstrate that the Lanczos algorithm is able to find also the eigenvalues of bound states of Dirac Hamiltonians, which are neither bounded from above nor bounded from below. In fact, bound states in the band gap converge much faster than states in the continuum part of the spectrum.

**5.1. Soft-core potential in two dimensions**

As an illustrative example for the calculation of eigenstates we consider the two-dimensional soft-core potential with

\[ \Psi(r) = \frac{2e^{\sqrt{3}Z}(1 + \sqrt{r^2Z^2 + 3})}{\sqrt{3\pi}(10\sqrt{3} + 17)} \exp\left(-\sqrt{r^2Z^2 + 3}\right). \]  

(20)

Orthogonality is lost, further Lanczos iterations will not further improve the approximation (17). The estimate (18) is especially useful because it can be computed with very little extra cost. It is, however, very loose and tends to overestimate the actual error. There are more precise but much more expensive to compute estimates, which can be found, for example, in \[38\].

### Table 1: Eigenenergies (minus the rest-mass energy \( mc^2 \)) of the two-dimensional soft-core potential (19)

| state number | Lanczos | Fourier |
|--------------|---------|---------|
| 1            | -1250.55965 | -1250.559 |
| 2            | -695.15042 | -695.151 |
| 3            | -688.06828 | -688.067 |
| 4            | -487.52777 | -487.529 |
| 5            | -380.4172 | -380.425 |
| 6            | -376.4623 | -376.471 |
| 7            | -320.86 | -320.633 |
| 8            | -318.84 | -318.601 |
| 9            | -247.66 | -251.149 |

The four-component Dirac wave functions are discretized using a pseudospectral method. For this purpose each component of the wave function is expanded into a finite set of \( N \times N \) two-dimensional basis functions, which are defined via a tensor product of the first \( N \) Hermite functions. During the calculation wave functions are represented via function values at the collocation points (rather than by expansion coefficients). The collocation points correspond to the roots of the \( N \)th Hermite function. Derivatives are then expressed by some dense matrices. The choice of Hermite functions as the underlying basis functions ensures that the wave function obeys the boundary condition that \( \Psi(x, r) \) goes to zero as \( |x| \to \infty \).

The convergence behavior of the Lanczos algorithm for the two-dimensional Dirac equation with a soft-core potential with \( Z = 50 \) is shown in Fig. 2. Here a basis set of \( N = 64 \) basis functions per dimension was used, which corresponds to a Hamiltonian matrix of size \( 4 \times 4 \) equal to 16384. A Gaussian wave packet of width \( \sim 1/Z \) was used as an initial starting vector for the Lanczos algorithm. As the soft-core potential goes to zero for \( r \to \infty \) bound states must lie in the range \((0, mc^2)\) and can therefore be identified easily. Figure 2(a) shows the eigenvalues of the six bound states with the lowest energy in \((0, mc^2)\) as a function of the number of iterations \( k \). One can see that the ground state converges very fast, excited states, however, follow significantly later. Note that the bound states do not appear in their energetic order. The second spin state of the degenerated ground state, for example, needs about 300 iterations to appear for the first time. While converging, extremal eigenvalues always obey a monotonic behavior \[34\]. This is not expected to happen for eigenvalues of the inner part of the spectrum. Figure 2(b) shows for each eigenvalue of Fig. 2(a) the corresponding error estimate (9). After about 600 iterations both ground state energies are converged to machine precision. Figure 2 also illustrates that seemingly converged eigenvalues may cross over to another value, as for example at \( k \approx 300 \), where the second ground state of the soft-core potential appears.

In order to validate the quality of the ground state energies obtained by the Lanczos algorithm we also compared these
FIG. 2: Iterative calculation of the eigenenergies of the two-dimensional Dirac Hamiltonian with the soft-core potential (19) with \( Z = 50 \) via the Lanczos algorithm with full reorthogonalization. A basis set of \( N = 64 \) basis functions per dimension was employed. Part (a) shows the eigenvalues (rest-mass energy \( mc^2 \) subtracted) of the first six bound states as a function of the iteration \( k \), while (b) shows the error bound (9). For clarity data of only every tenth Lanczos iteration is presented in both plots.

TABLE 2: Ground state energies (minus the rest-mass energy \( mc^2 \)) of the two-dimensional soft-core potential (19) for different values of the atomic number \( Z \). The results obtained with the Lanczos algorithm using 1000 iterations are compared to perturbation-theory results. In addition the exact ground state energy for the corresponding nonrelativistic Schrödinger equation is given.

| \( Z \) | Energy in au | Lanczos | Perturbation theory | Nonrelativistic |
|---|---|---|---|---|
| 1 | \(-0.500000089\) | \(-0.500000090\) | \(-0.5\) |
| 2 | \(-2.00000144\) | \(-2.000001441\) | \(-2.0\) |
| 3 | \(-4.50000728\) | \(-4.500007297\) | \(-4.5\) |
| 5 | \(-12.5000562\) | \(-12.50005638\) | \(-12.5\) |
| 10 | \(-50.00008998\) | \(-50.00090875\) | \(-50.0\) |
| 50 | \(-1250.55965\) | \(-1250.697356\) | \(-1250.0\) |

5.2. Zeeman effect

We consider the Coulomb potential with an external magnetic field of magnitude \( B \) as an application of the Lanczos algorithm to a fully three-dimensional problem. The magnetic field leads to the well-known Zeeman splitting of the degenerate hydrogenic eigenstates. In this section, we analyze the splitting of the ground state in detail. A perturbative solution of the relativistic splitting of the hydrogenic ground state is known to

...
Lanczos method agrees very well with perturbation theory. Both results disagree, however, for fields stronger than $B \approx 1000$ au by a few percents. Here we leave the parameter domain where perturbation theory is applicable.

6. Time-dependent problems

After demonstrating the application of the Lanczos algorithm to time-independent relativistic eigenproblems we address time-dependent problems now. The motion of a free wave packet is one of the rare systems where an analytical solution to the time-dependent Dirac equation is known. Thus, it provides an ideal benchmark system for time-dependent problems. We chose a relativistic one-dimensional wave packet containing both positive-energy and negative-energy free-particle states with Gaussian momentum distribution. When using the standard position operator, the center of mass $\langle \chi \rangle = \langle \Psi | x | \Psi \rangle$ shows so-called Zitterbewegung, which is caused by an interference between positive and negative energy contributions.

In our numerical test we employed the pseudospectral method with a basis set of 512 basis functions for the spatial discretization of the wave function. Time steps of size $\Delta t = 10^{-5}$ au and $k = 8$ Lanczos iterations per time step were used. The numerical result for the wave function $\Psi_{\text{num}}(x,t)$, which is represented by the vector $\psi(t)$, can be compared to the analytical solution

$$
\Psi(x,t) = \frac{1}{\sqrt{2\pi \hbar}} \int e^{-i\sigma p t/4\hbar^2} \frac{1}{\sqrt{2\pi \sigma}} \left( u^+(p) + u^-(p) \right) dp,
$$

with the free-particle momentum states of positive and negative energy

$$
u^+(p) = \frac{d^+(p)}{\text{sgn}(p)d^-(p)} e^{i(p+\delta E)(t)/\hbar},
$$

$$
u^-(p) = \frac{-\text{sgn}(p)d^+(p)}{d^-(p)} e^{i(p+\delta E)(t)/\hbar},
$$

which are defined via

$$
d^\pm(p) = \left( \frac{1}{2} \pm \frac{1}{2\sqrt{1 + p^2/(mc^2)}} \right)^{1/2}
$$

and

$$
\delta E = \frac{B}{2m} \left( 1 + 2 \sqrt{1 - \left( \frac{Z}{c} \right)^2} \right).
$$

The width of the momentum distribution was chosen as $\sigma = 50$ au. The numerical result of the wave packet’s center-of-mass motion is presented in Fig. 5(a). It features the expected oscillation around the origin with decaying amplitude for larger times. Figure 5(b) shows the difference $\langle x \rangle_{\text{num}}(t) - \langle x \rangle(t) = |\langle \Psi_{\text{num}}(t) | x | \Psi_{\text{num}}(t) \rangle - \langle \Psi(t) | x | \Psi(t) \rangle|$. On average the center-of-mass error increases linearly with time $t$ but remains at the order of $10^{-12}$ au which is eight orders of magnitude smaller than the amplitude of the Zitterbewegung.
 For the simulation shown in Fig. 5.

FIG. 5: (a) Center-of-mass evolution of a highly relativistic wave packet with vanishing mean momentum. It was set up to contain both positive and negative energy components such that Zitterbewegung can be observed. (b) Difference between the numerical solution based on the Lanczos algorithm and the exact center-of-mass evolution. The numerical solution was obtained with $N = 512$ basis functions and $k = 8$ Lanczos iterations per time step.

FIG. 6: The cumulated error bound (error bound at each time step given by (18) and the actual numerical error as functions of the time for the simulation shown in Fig. 5.

An upper bound of the error that is introduced due to the Lanczos approximation of the matrix exponential can be estimated via equation (18). Thus, an upper bound of the total error in a series of time steps can be estimated by adding up the estimates (18) at every time step. We call this estimate the cumulated error bound, which is indicated by the dashed line in Fig. 6. For the specific set of parameters the error bound (18) turns out to be almost constant and therefore the cumulated error bound grows linearly. In addition, we computed at every time step the real error of the wave function defined as $||\psi(t) - \psi_{exact}(t)||_2$, where the vector $\psi_{exact}(t)$ equals the analytical solution $\Psi(x, t)$ given by (22) at the collocation points. This error is indicated by the solid line in Fig. 6. Comparing the two lines in Fig. 6 illustrates that the error bound (18) indeed overestimates the actual numerical error. At the end of the propagation at $t = 0.006 \text{ au}$ the error of the wave function is almost an order of magnitude smaller than the cumulated estimate, despite the fact that the error introduced by discretization is not included in the estimate (18). Note that, in Fig. 6 it appears as if for times $t < 0.0003 \text{ au}$ the numerical error $||\psi(t) - \psi_{exact}(t)||_2$ would be larger than the cumulated error bound. This, however, is an artifact caused by the numerical evaluation of the exact wave function (22) which involves a numerical integration via a discrete Fourier transform on a regular grid and interpolation to the collocation points. In conclusion, our results show that very precise approximations of the time evolution operator of the time-dependent Dirac equation can be calculated by the Lanczos algorithm.

7. Parallel implementations

Our evaluation of the Lanczos propagator for the Dirac equation was mainly motivated by the fact that the Lanczos algorithm has the potential to scale well in parallel implementations on various parallel computing architectures including distributed memory systems. Thus, it might be suitable for large-scale computations. A parallel Lanczos algorithm only requires parallel computation of inner products and a parallel implementation of the Hamiltonian’s action on a state vector.

We implemented a parallel version of the Lanczos propagator for the two-dimensional Dirac equation by utilizing the Message Passing Interface (MPI) standard [45]. The differential operators of the Hamiltonian were approximated via first order finite differences. In this case (and in contrast to the pseudospectral method used in the previous sections) the differentiation matrices are sparse and the Hamiltonian’s action on a state vector can be parallelized efficiently via decomposition of the total rectangular computational grid into smaller subgrids; one sub-grid per process. This domain decomposition approach can also be applied to other finite-difference based algorithms, for example, real space split operator schemes for the Klein-Gordon equation [46] and the Dirac equation [16]. The exchange of boundary data between neighboring domains is implemented via the nonblocking MPI functions MPI_Isend and MPI_Irecv [31]. Parallel scalar products are calculated via MPI_Allreduce.

For the following benchmarks we consider a free two-dimensional wave packet. It is constructed by a Gaussian superposition of positive-energy free-particle states with mean momentum $p = (100 \text{ au}, 0 \text{ au})$ and a momentum space width of $\sigma_x = \sigma_y = 400 \text{ au}$. Due to the broad momentum distribution and the nonlinear relativistic relation between velocity and momentum a shock front with a ring structure emerges during the temporal evolution of the wave packet as show in Fig. 7. The figure’s left part shows the initial wave packet and...
with a 10 Gigabit Ethernet interconnect. With increasing number of processes the nodes were filled up successively. In order to avoid performance degeneration due to intra-node process migration between different CPUs each process was pinned to a specific core. The measured speedup is shown in Fig. 8. The speedup is approximately linear in the number of processes but differs from the ideal scaling $s_p \approx p$ due to the communication overhead, which also grows with the number of processes.

The discretization by finite differences leads to a consecutive memory-accesses pattern when calculating the inner products and applying the Hamiltonian with very few accesses per fetched memory element. Consequently, the CPUs’ cache hierarchy cannot be used efficiently, so that the memory bandwidth limits the performance. The inset of Fig. 8 illustrates the memory-bandwidth limitation of the Lanczos propagator. For $p \leq 10$, that is when a node with 16 cores is still under-utilized, we have an almost ideal scaling with $s_p \approx p$. But for $p > 10$ limitations due to the memory bandwidth set in. This efficiency degradation occurs before the first node is completely filled with 16 processes. Thus, network traffic is not the main performance limiting factor. Further numerical experiments showed that the speedup and the parallel efficiency can be increased when the nodes are systematically under-utilized. This means that, for example, only four cores per CPU are used. This also supports our assertion that the Lanczos propagator is memory-bandwidth bounded.

8. Conclusions

We evaluated the Lanczos algorithm for the application to the time-independent and the time-dependent Dirac equation. Our results indicate that the Lanczos algorithm is able to compute precise solutions of the Dirac equation. This was demonstrated for the time-independent eigenvalue problem solving the two-dimensional soft-core potential and the three-dimensional Coulomb potential with an additional strong magnetic field. While in the case of the soft-core potential the numerical instabilities of the Lanczos algorithm were circumvented by applying full reorthogonalization, the plain Lanczos algorithm was applied to the Coulomb potential with magnetic field. As in the case of the soft-core potential the Lanczos algorithm allowed us to calculate the pair of bound states with lowest energy, such that we were able to calculate the Zeeman splitting in a strong magnetic field. Furthermore, we demonstrated for one- and two-dimensional wave packets that the Dirac-equation’s time evolution operator can be approximated very precisely using the Lanczos algorithm.

The Lanczos approach is not specific to particular means of discretization of the Dirac Hamiltonian as long as the discretization preserves hermiticity. Here we employed pseudospectral methods and finite differences. The latter yield a sparse matrix representation of the Dirac Hamiltonian and in this case the Lanczos propagator can be parallelized very efficiently via domain decomposition as demonstrated in this paper. The benefit of pseudospectral discretizations is to approximate the differential operators of the Dirac Hamiltonian very precisely and in this way allowing for very accurate bound-state calculations and time-propagation. The accuracy of the Lanczos Dirac propagator combined with its property to perform well on modern parallel hardware architectures makes it a good choice for large-scale calculations of relativistic quantum dynamics. It might be interesting to implement the Dirac Lanczos propagator also on other parallel hardware architectures, as for example graphics processing units.
[39] C. C. Paige, Computational variants of the Lanczos method for the eigenproblem, IMA Journal of Applied Mathematics 10 (3) (1972) 373–381. doi:10.1093/imamat/10.3.373

[40] G. Meurant, The Lanczos and conjugate gradient algorithms: from theory to finite precision computations, Vol. 19, SIAM, 2006.

[41] J. Cullum, R. A. Willoughby, Computing eigenvalues of very large symmetric matrices-an implementation of a Lanczos algorithm with no reorthogonalization, Journal of Computational Physics 44 (2) (1981) 329–358. doi:10.1016/0021-9991(81)90056-5

[42] C. W. Clark, Closed-form solutions of the Schrödinger equation for a class of smoothed Coulomb potentials, Journal of Physics B: Atomic, Molecular and Optical Physics 30 (11) (1997) 2517–2527. doi:10.1088/0953-4075/30/11/006

[43] M. D. Feit, J. A. Fleck, A. Steiger, Solution of the Schrödinger equation by a spectral method, Journal of Computational Physics 47 (3) (1982) 412–433. doi:10.1016/0021-9991(82)90091-2

[44] N. F. Mott, H. S. W. Massey, The theory of atomic collisions, 3rd Edition, The international series of monographs on physics, Clarendon Press, Oxford, 1971.

[45] Message Passing Interface Forum, MPI: A message-passing interface standard version 3.0, http://www.mpi-forum.org (2012).

[46] M. Ruf, H. Bauke, C. H. Keitel, A real space split operator method for the Klein-Gordon equation, Journal of Computational Physics 228 (24) (2009) 9092–9106. doi:10.1016/j.jcp.2009.09.012