

# Nonlinear grid mapping applied to an FDTD-based, multi-center 3D Schrödinger equation solver

Nicolas Bigaouette\*, Edward Ackad, Lora Ramunno

Department of Physics, University of Ottawa, 150 Louis Pasteur, Ottawa ON, K1N 6N5, Canada

## ARTICLE INFO

### Article history:

Received 16 February 2011

Received in revised form 15 August 2011

Accepted 18 August 2011

Available online 26 August 2011

### Keywords:

Quantum mechanics

Time-dependent Schrödinger equation

Nonlinear grid mapping

Finite-difference time domain

Coulomb potential

## ABSTRACT

We developed a straightforward yet effective method of increasing the accuracy of grid-based partial differential equation (PDE) solvers by condensing computational grid points near centers of interest. We applied this “nonlinear mapping” of grid points to a finite-differenced explicit implementation of a time-dependent Schrödinger equation solver in three dimensions. A particular multi-center mapping was developed for systems with multiple Coulomb potentials, allowing the solver to be used in complex configurations where symmetry cannot be used for simplification. We verified our method by finding the eigenstates and eigenenergies of the hydrogen atom and the hydrogen molecular ion ( $H_2^+$ ) and comparing them to known solutions. We demonstrated that our nonlinear mapping scheme – which can be readily added to existing PDE solvers – results in a marked increase in accuracy versus a linear mapping with the same number of (or even much fewer) grid points, thus reducing memory and computational requirements by orders of magnitude.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

There are many works devoted to solving the time-dependent Schrödinger equation [1,2]. Recently there have been several based on the finite-difference time domain (FDTD) technique [3–7], which is a standard algorithm in computational electrodynamics. In usual FDTD, Maxwell's equations are discretized on a grid and solved explicitly using leap-frog integration in time [8,9]. Since the FDTD algorithm is explicit and local, it is relatively straightforward to implement and parallelize.

Applying FDTD to solve the Schrödinger equation yields a method that is generally straightforward to implement and computationally efficient. This quantum mechanical version of FDTD has been developed by Sullivan et al., and later by Soriano et al., to study the eigenstates and dynamics of various quantum nanostructures [3], including two electrons in a quantum dot [4], quantum well wires [5] and spin evolution in a quantum dot [6]. In 2007, Sudiarta and Geldart proposed a modification for time-independent Hamiltonians which involved performing a Wick rotation of the time variable by  $\pi/2$  in the complex plane [10]. The wavefunction evolves in imaginary time, and the eigenstates and eigenvalues are then extracted, even in the presence of a magnetic field [11].

In implementing either the “real time” or “imaginary time” method to solve the Schrödinger equation in three dimensions for a Coulomb potential, one quickly realizes the burden that the

grid imposes. The Coulomb potential has an infinite range but is singular at the charge's location. The simulation domain must be large enough to fit all the eigenstates of interest (especially for the higher excited states), but the computational grid itself must be fine enough to resolve the potential and the states with reasonable accuracy. For the usual discretization scheme which uses a regularly spaced grid, this can lead to large memory requirements and correspondingly long computation times.

Nonuniform grids have been developed for the electrodynamics FDTD method [8], but they typically require a redefinition of the equations underlying the simulation. Thus they tend to be difficult to implement and prone to subtle errors. Curvilinear coordinates and unstructured grids have also been used in FDTD [12–16], but require revisiting the equations, for example taking Maxwell's equations in integral form instead of differential form. Gordon et al. [17] proposed a nonlinear discretization that can be applied to any explicit method of solving the Schrödinger equation with a Coulomb potential, taking into account prior knowledge of the asymptotic behavior of the wavefunction near the singularity. While this gives more accurate results (e.g., a decrease of the relative error of eigenenergies), their nonlinear grid becomes linear away from the singularity, and thus would be susceptible to the same memory and computational restrictions of a purely linear grid.

A different approach to concentrate grid points near regions of interest uses a canonical coordinate transformation to map an infinite or semi-infinite domain onto a finite one [18–20]. Other groups used a variation of this method to concentrate points where the solution exhibits high gradients [21], in spectral methods to

\* Corresponding author. Tel.: +1 613 562 5757; fax: +1 613 562 5190.

E-mail address: [nbigaouette@gmail.com](mailto:nbigaouette@gmail.com) (N. Bigaouette).

reduce the stiffness of a problem [22] or for a moving mesh in two-phase flows problem [23], finite differences [24] or matrix representations in 1D [25]. This technique (which we will refer to as “mapping”) is also useful in the Fourier method since it provides an evenly spaced grid, which is necessary for the FFT algorithm [26,27]. Kokoouline et al. [28] developed a technique based on a Fourier grid method for solving the radial Schrödinger equation with a  $R^{-n}$  potential for  $n = 3$  and  $n = 6$ . They construct a nonlinear mapping function from the shape of the potential. Similarly, others have used a nonlinear mapping to solve the (one-dimensional) radial Schrödinger equation using direct methods [29] or a plane-wave expansion in the transformed coordinate [30].

While mapping has been used extensively in spectral methods, there are only few works that combine it with FDTD. Xiaojun et al. [31] implemented a mapping in their 2D electrodynamics FDTD. Yang et al. [32] adapted a Crank–Nicolson (CN) electrodynamics FDTD on a mapped grid, but due to the implicit nature of the CN-FDTD, the grid cell sizes still appear in the equations, complicating the implementation. FDTD, to solve either Maxwell's or Schrödinger's equations, could be greatly enhanced with a robust mapping procedure.

In this paper, we propose a simple method of constructing an orthogonal, nonuniform grid via a nonlinear mapping of the spatial coordinates and apply it to an FDTD-based Schrödinger solver. Rather than using the mapping between space coordinates where the problem is more easily solved (for example to reduce the stiffness of the problem or to switch from an infinite to a finite domain), we perform a mapping to a discrete, integer space suitable for representing the index of an array in computer memory. Additionally, treating each dimension individually allows a straightforward implementation procedure. Furthermore, our approach is used to concentrate grid points in different regions of space and is not bound to a single center of interest as is the case generally in the literature. Grabowski et al. [33] did use three different subdomains, each with a different mapping function, to study a single center system (helium) using the pseudo-spectral method, but the three overlapping subdomains require special treatment, increasing the complexity of the implementation. While we apply our nonlinear mapping to the Schrödinger equation with single and multiple Coulomb potentials, it is generally applicable to the discretization of any system of partial differential equations.

We organize our paper as follows. In Section 2, we introduce the Schrödinger solver, for both real and imaginary time evolution. We describe the general considerations of our nonlinear mapping scheme in Section 3.1, and provide details on obtaining a mapping function in Section 3.2. In Section 3.3 we describe the particular nonlinear mapping we have developed for one or multiple charge centers of interest within the Schrödinger solver. We validate our method for real and imaginary time evolution in Section 4, where we calculate the eigenenergies and eigenstates for two known systems: hydrogen and  $H_2^+$ . We compare our nonlinear mapping results to similar calculations that employ a linear grid, and find that the nonlinear mapping provides a much more efficient calculation for the same level of accuracy, or much higher accuracy for the same number of (or even much fewer) grid points.

## 2. Finite-differenced Schrödinger equation

The spinless one electron Schrödinger equation in a time-dependent potential  $V(\mathbf{r}, t)$  is, in atomic units,

$$i\frac{\partial}{\partial t}|\psi(\mathbf{r}, t)\rangle = \left(-\frac{1}{2}\nabla^2 + V(\mathbf{r}, t)\right)|\psi(\mathbf{r}, t)\rangle. \quad (1)$$

We solve Eq. (1) using an approach similar to the FDTD algorithm that is widely used in computational electrodynamics [8], where

the electric and magnetic vector fields are solved on a computational grid. Since FDTD uses an explicit method for solving partial differential equations (*i.e.*, by using the current value of the fields to calculate the field values at a later time), they are generally easier to implement than implicit methods. In addition, FDTD algorithms can be efficiently parallelized due the local nature of the operators. This general approach can be adapted for a scalar field, such as the electron wavefunction [9]. We do this for both real time and imaginary time propagation, and describe the basic elements of each propagation scheme in the following subsections.

### 2.1. Real time

Here we describe the basic algorithm of real time propagation. First, the complex valued wavefunction is split into real and imaginary components [34]

$$|\psi(\mathbf{r}, t)\rangle = |\psi_R(\mathbf{r}, t)\rangle + i|\psi_I(\mathbf{r}, t)\rangle. \quad (2)$$

The Schrödinger equation (1) then becomes

$$\frac{\partial}{\partial t}|\psi_R\rangle = -\frac{1}{2}\nabla^2|\psi_I\rangle + V(\mathbf{r}, t)|\psi_I\rangle, \quad (3a)$$

$$\frac{\partial}{\partial t}|\psi_I\rangle = \frac{1}{2}\nabla^2|\psi_R\rangle - V(\mathbf{r}, t)|\psi_R\rangle. \quad (3b)$$

Time derivatives can be approximated using central finite differences, as per the FDTD method, giving

$$|\psi_I\rangle_{ijk}^{n+1/2} = |\psi_I\rangle_{ijk}^{n-1/2} - \Delta t \left( V_{ijk}^n - \frac{1}{2}\nabla^2 \right) |\psi_R\rangle_{ijk}^n, \quad (4a)$$

$$|\psi_R\rangle_{ijk}^{n+1} = |\psi_R\rangle_{ijk}^n + \Delta t \left( V_{ijk}^{n+1/2} - \frac{1}{2}\nabla^2 \right) |\psi_I\rangle_{ijk}^{n+1/2}. \quad (4b)$$

The spatial derivatives are also discretized via central differences [3], which can be according to a linear grid (the usual case) or a nonlinear grid (the method we develop in this paper).

Due to the explicit discretization of the time derivative, there is an upper bound for the time step for stability to be ensured. The stability criterion for  $\Delta t$  was derived in Ref. [35], and depends on both the grid cell size and the maximum absolute value of the potential within the computational domain.

This explicit scheme lends itself to a leap-frog integration in time where the real and imaginary parts of the wavefunction are alternatively updated. Unlike in electrodynamics FDTD where there is a shifting in space of the vector components of the electric and magnetic fields according to the Yee cell, the real and imaginary parts of the electron wavefunction can both be defined at the same spatial position,  $(i, j, k)$ .

The real time evolution method as described can be used to solve general time-dependent problems, including a time-dependent potential. In this paper, however, we apply the real time evolution to obtain the eigenenergies of a given (static) potential. This is analogous to techniques in electrodynamics where bound modes of an optical structure are obtained via FDTD. Here, we take the initial wavefunction to be white noise, *i.e.*, we choose a random value for the wavefunction at each spatial location in the grid. This samples all the available eigenmodes. We then allow this “noise” wavefunction to evolve in time. The eigenenergy spectrum is obtained by taking a Fourier transform of the resulting time domain signal. The accuracy of the spectrum will depend in part on the size of the computational domain and the spatial resolution of the grid, but also on the total time duration of the simulation,  $t_{\max}$ . To resolve two distinct neighboring eigenenergies  $E_m$  and  $E_{m+1}$ , where  $E_m$  is the eigenenergy of the  $m$ th state, we require

$$t_{\max} \gg \left| \frac{2\pi}{E_m - E_{m+1}} \right|, \quad (5)$$

since the frequency resolution of a discrete Fourier transform is  $2\pi/t_{\max}$ .

Previous work used a single spatial grid point to construct the time domain signal [34]. While a reasonable approach, there is some risk that some eigenenergies will be missed. This could happen, for example, if the chosen point in space happens to be close to a nodal point of the associated eigenstate. To remedy this problem, we construct a time domain signal that is based on a sum of the values of the wavefunction at each grid point, effectively using the whole simulation box as the source of the Fourier transform. But special care needs to be taken here too, as antisymmetric states can vanish if a simple sum is used. To avoid this, each value of the wavefunction to be summed is first multiplied by an arbitrary and unique weighting constant. The linearity property of the Fourier transform ensures the validity of this approach. In our code, the weight is taken as a random number between 0.5 and 1.5. The randomness assures a fair distribution of the weighting constants among the grid while the limited range of the random numbers prevent some cells to have significantly more weight than others. We are thus assured that all eigenstates present in the simulation box will be found.

Once the eigenenergy spectrum has been calculated, the associated eigenstates can be obtained as a function of space by re-running the same real time simulation and taking a running Fourier transform at each spatial location for the eigenenergy or eigenenergies of interest [34].

## 2.2. Imaginary time

Another approach for obtaining the eigenenergies and eigenstates of a time-independent potential  $V(\mathbf{r})$  is to perform a Wick rotation of  $\pi/2$  in the complex plane of the time variable [10]. Time becomes a purely imaginary quantity, and the Schrödinger equation (1) is transformed into the heat equation

$$-\frac{\partial}{\partial \tau} |\psi(\mathbf{r}, \tau)\rangle = \left( -\frac{1}{2} \nabla^2 + V(\mathbf{r}) \right) |\psi(\mathbf{r}, \tau)\rangle. \quad (6)$$

To solve this equation numerically for a given initial wavefunction, we apply finite differencing to advance the system forward in imaginary time  $\tau$  in an explicit scheme, similar to Section 2.1. The major difference is that Eq. (6) is real, thus the wavefunction can also be taken as real. Consequently, when using central finite differencing for the Laplace operator in Eq. (6), special care must be taken as the wavefunction update requires neighboring cells' values. This could cause an update operation to use an already updated value of its neighbor. A simple workaround is to store two grids instead of just one: one for the wavefunction at the current time step and one at the previous time step, using one to update the other.

Similar to the real time method, this imaginary time scheme is explicit, and thus still subject to an upper bound on the time step for stability [10].

The way in which the eigenstates emerge from imaginary time evolution can be understood by expanding the initial wavefunction,  $|\psi(\mathbf{r}, 0)\rangle$ , in the (yet unknown) eigenstate basis  $|\phi_n(\mathbf{r})\rangle$

$$|\psi(\mathbf{r}, \tau = 0)\rangle = \sum_{n=0}^{\infty} c_n |\phi_n(\mathbf{r})\rangle. \quad (7)$$

Considering the time evolution of the wavefunction, we obtain

$$|\psi(\mathbf{r}, \tau)\rangle = \sum_{n=0}^{\infty} e^{-E_n \tau} c_n |\phi_n(\mathbf{r})\rangle, \quad (8)$$

where  $E_n$  are the eigenvalues of the respective eigenstates  $|\phi_n\rangle$ . The amplitude of these eigenstates evolve in imaginary time according to

$$c_n(\tau) = e^{-E_n \tau} c_n. \quad (9)$$

Note that for bound states, the eigenvalues  $E_n$  are negative resulting in increasing exponentials. As the system evolves in imaginary time, the ground state quickly dominates because its amplitude grows exponentially faster than the amplitudes of the other states, given that its corresponding eigenenergy  $E_0$  has a larger (negative) value than the other states.

Refs. [10,36] describe a method to obtain the eigenstates. However, due to the exponential increase of the eigenstates as seen in Eq. (9), we found that the ground state was growing to the point where the other states were buried in the floating point precision of the computer, limiting the duration of the simulation and the number of states found.

We propose a more robust and faster way of finding the excited states that is not limited by this problem. The (imaginary) time evolution of Eq. (6) is calculated until the ground state becomes dominant in Eq. (8), specifically when the amplitude increases from 1 initially to above 10,000. Then, the wavefunction is normalized, stored and labeled as the ground state  $|\phi_0\rangle$ . The simulation continues for a single time step and the projection of the ground state onto the wavefunction is removed to obtain a new wavefunction  $|\psi(\tau_n)'\rangle$  that will be used to continue the simulation:

$$|\psi(\tau_n)'\rangle = |\psi(\tau_n)\rangle - |\phi_0\rangle \langle \phi_0 | \psi(\tau_n) \rangle.$$

The new wavefunction  $|\psi(\tau_n)'\rangle$  should now be free of the ground state and the simulation continues updating it. The dominant states should now be the first excited state  $|\phi_1\rangle$ . When the amplitude of  $|\psi(\tau_n)'\rangle$  again reaches the maximum chosen, the wavefunction is once more normalized, stored and labeled as the first excited state  $|\phi_1\rangle$ . Another time step is performed and the projection of both the ground state and first excited state is removed from the wavefunction:

$$|\psi(\tau_n)''\rangle = |\psi(\tau_n)'\rangle - |\phi_0\rangle \langle \phi_0 | \psi(\tau_n)' \rangle - |\phi_1\rangle \langle \phi_1 | \psi(\tau_n)' \rangle.$$

The procedure continues until all wanted states are found. Since the wavefunction used in the calculation is free of all found lower states, they will never interfere with the current calculation: it could continue indefinitely.

Removing the states projection's completely is impossible as there will always be some rounding errors in the computer calculation. Moreover, the time evolution of Eq. (6), once discretized, will by itself introduce some rounding errors that will cause a certain overlap of the removed states into the wavefunction. Because of this, even though states projections are being removed, they will always reappear from the floating point noise. Anticipating this “rise from the ashes”, the projections can be removed from the wavefunction at each time step, an unnecessary heavy burden. To prevent this overhead, one can easily calculate when the exponential growth of the ground state, the state with the fastest evolution, reaches from zero a certain value that is comparable to the main wavefunction. In our calculation, we remove the projections at periods of 3 atomic units of (imaginary) time, a conservative value when considering hydrogen and  $\text{H}_2^+$ . Note that the projections must be removed before a wavefunction is labeled as an eigenstate, insuring it does not contain any contribution from lower energy states.

## 3. Details of the nonlinear mapping

Spatial discretization is normally achieved using a regularly spaced grid. For some problems, however, it is advantageous to

use a nonuniform spatial grid; the Schrödinger equation with a Coulomb potential is a prime example. To get adequate resolution near the ion's location, the grid spacing must be small. The computational box, however, must remain relatively large to capture the infinite nature of the electronic wavefunction, a burdensome requirement even considering that this wavefunction is exponentially decreasing at large distances. For higher energy states, which extend over a larger volume with increasing energy, the computational box must be even larger. A regular, equally spaced grid would impose small grid spacing throughout the entire computational domain. We thus develop a scheme to increase the grid spacing at points far from the charge center to reduce the computational load, while keeping the grid spacing small near the charge center. In Section 3.1, we describe a general approach to a nonuniform grid that we call “nonlinear mapping” which can be readily incorporated into an existing PDE grid-based solver. We outline one general method for finding the mapping function in Section 3.2. We apply it in Section 3.3 to the Schrödinger equation for the special case of one or more Coulomb potentials, without requiring any symmetry properties thus ensuring generality of the solver.

### 3.1. General approach

Consider a one-dimensional problem where a function  $f$  depends on the position  $x$ . The generalization to three dimensions is straightforward in our case since all dimensions are independent of each other. We assume the position  $x$  to be a function of another variable  $i$

$$x \rightarrow x(i). \quad (10)$$

Generally, the mapping transforms one spatial coordinate, in this case  $x$ , to another spatial coordinates  $x'$ . The transformed equation will then need to be discretized for a computer calculation. Instead of mapping onto a temporary transformed space which must then be discretized, we propose transforming directly to the discrete, integer space suitable for computer memory storage and calculation.

Typically, a grid is chosen such that the position of a grid cell would be linear in  $i$ , i.e.,

$$x(i) = i\Delta x, \quad (11)$$

where  $i$  is an integer; usually  $i$  is the index of the memory location where  $x(i)$  and  $f(x(i))$  are stored. This is a linear mapping between the index and the grid cell position, and gives a regular grid of spacing  $\Delta x$ .

We do not need to choose the mapping between  $x$  and  $i$  to be linear. For an arbitrary mapping  $x(i)$ , the spatial derivatives of  $f(x(i))$  become

$$\frac{\partial f}{\partial x} = \frac{\partial i}{\partial x} \frac{\partial f}{\partial i} = \frac{1}{J_1} \frac{\partial f}{\partial i}, \quad (12a)$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{J_1^2} \frac{\partial^2 f}{\partial i^2} - \frac{J_2}{J_1^3} \frac{\partial f}{\partial i}, \quad (12b)$$

where we have defined

$$J_1 \equiv \frac{\partial x}{\partial i}, \quad J_2 \equiv \frac{\partial^2 x}{\partial i^2}. \quad (13)$$

Because we consider each dimension independently, the Jacobian matrix describing the transformation is diagonal and  $J_1$  represents the element of the corresponding dimension. Even though a transformation could be taken such that all dimensions are coupled as in Ref. [26], our decoupling greatly simplifies the implementation by dropping the Jacobian's matrix representation requirement.  $J_1$  is thus associated with the grid points interdistance (a function

of  $i$  for nonuniform grids) and  $J_2$  is its rate of change. In the case of linear mapping,  $J_2 = 0$ , corresponding to a uniform grid cell interdistance  $J_1 = \Delta x$ .

The function  $f$  is discretized with respect to  $i$ , and these discrete values are used in the finite differencing. For example, forward difference discretization would give

$$\frac{\partial f}{\partial i} \approx \frac{f(i + \Delta i) - f(i)}{\Delta i} \equiv f_{i+1} - f_i. \quad (14)$$

Note that we have made the association between the variable  $i$  and the index in memory (the subscripts) where the corresponding value of  $f$  is stored. Since the indices  $i$  are integers, we have  $\Delta i = 1$ .

More complicated, nonlinear functions can thus be readily applied to PDEs in place of the linear mapping, Eq. (11). Required properties of a mapping functions are given by Andrae et al. [37]. First, the mapping and its inverse should exist (uniqueness or bijectivity). In other words,  $x(i(x)) = x$ . Second, the mapping function should be monotonic and third, it should be differentiable. Additionally, it should distribute the points in space in a sensible way, such that the spacing is smaller where more spatial resolution is required, and larger where it is not.

### 3.2. Details of the implementation

In this section we discuss a straightforward way of obtaining the mapping function  $x(i)$  from an arbitrary source function  $S(x)$ . Consider first a single center of interest in one dimension located at  $x = x_0$ , around which greater resolution is required, down to a finite (minimum) cell size; this simplifies the derivation and can be easily generalized to many centers in three dimensions. The values of  $x$  will be between  $x_{\min}$  and  $x_{\max}$  and  $i$  between  $i_{\min}$  and  $i_{\max}$ , usually 0 and  $N - 1$ ,  $N$  being the total number of cells.

Instead of finding a single strictly increasing monotonic function for  $x(i)$ , a difficult task, we will use three different functions that will describe three sections of the ideal monotonic function. Region 1 begins at the lower part of the subdomain and finishes at a distance  $d$  before the center's position  $x_0$ . Region 2 then covers from  $x_0 - d$  to  $x_0 + d$  and finally region 3 begins at  $x_0 + d$  and finishes at the end of the subdomain. In regions 1 and 3, a nonlinear mapping is used while in region 2 (of width  $2d$ ), a linear mapping is used. This linear part located on top of the center's location allows a free parameter in our mapping procedure: all grid points in that region will be equidistant of  $\Delta x_{\min}$ , the minimum that is required. The three regions are thus

Region 1:  $x \in [x_{\min}, x_0 - d]$ ,

Region 2:  $x \in [x_0 - d, x_0 + d]$ ,

Region 3:  $x \in [x_0 + d, x_{\max}]$ .

The parameter  $d$  depends on the subdomain's properties (size, number of cells, etc.) and also on the shape of the source function  $S(x)$  which is described in the following part. Since the position of the center of interest is ill-defined in  $i$ -space even though in  $x$ -space it is well known ( $x_0$ ), we will thus find an expression for  $i(x)$ , the inverse mapping, instead of the mapping  $x(i)$ .

The fundamental and novel step of our approach is to define the inverse mapping  $i(x)$  as a cumulative sum of the absolute value of a “source” function  $S(x)$ :

$$i(x) = A \int_{x_{\min}}^x |S(x')| dx' + i_{\min}, \quad (15)$$

where  $A$  is a normalization constant such that  $i(x_{\max}) = i_{\max}$  is the highest cell index. This cumulative sum guarantees the monotonicity and uniqueness of the mapping which are essential properties.



Any function can be used for  $S(x)$ , reducing the difficulty of finding a suitable mapping function.

For a position  $x < x_0 - d$ , the cumulative sum runs over only region 1. For a position  $x_0 - d \leq x < x_0 + d$  the cumulative sum runs over regions 1 and 2 and for  $x_0 + d \leq x$  the three regions are covered. When the integration is performed we have:

$$i(x) - i_{\min} = \begin{cases} i_1(x) & \text{in region 1,} \\ i_1(x_0 - d) + i_2(x) & \text{in region 2,} \\ i_1(x_0 - d) + i_2(x_0 + d) + i_3(x) & \text{in region 3,} \end{cases} \quad (16)$$

where  $i_1(x)$ ,  $i_2(x)$  and  $i_3(x)$  are the individual functions of each region.

To find  $d$ , we use  $i(x_{\max}) \equiv i_{\max}$  and isolate  $d$  from the expression:

$$i_{\max} = i_1(x_0 - d) + i_2(x_0 + d) + i_3(x_{\max}) + i_{\min}. \quad (17)$$

Depending upon the source function that generates  $i_1$  and  $i_3$ , this may yield a transcendental equation for  $d$ , which may be solved using a root finding algorithm.

Having solved for  $d$ , we can now find the boundaries in  $i$ -space between the different regions. We call these  $j_1$  and  $j_2$ :

$$i(x_0 - d) = i_1(x_0 - d) \equiv j_1, \quad (18a)$$

$$i(x_0 + d) = i_1(x_0 - d) + i_2(x_0 + d) \equiv j_2. \quad (18b)$$

This allows us to define the three regions in  $i$ -space as

Region 1:  $i \in [i_{\min}, j_1]$ ,

Region 2:  $i \in [j_1, j_2]$ ,

Region 3:  $i \in [j_2, i_{\max}]$ .

Once expressions for  $i_1(x)$ ,  $i_2(x)$  and  $i_3(x)$  have been found, it is easy to invert each of them individually to get  $x_1(i)$ ,  $x_2(i)$  and  $x_3(i)$ . The full mapping is

$$x(i) = \begin{cases} x_1(i) & \text{in region 1,} \\ x_1(j_1) + x_2(i) & \text{in region 2,} \\ x_1(j_1) + x_2(j_2) + x_3(i) & \text{in region 3.} \end{cases} \quad (19)$$

Finally, we obtain the mapping's first and second derivatives  $J_1$  and  $J_2$  analytically from Eq. (19).

Generalization to multiple centers is straightforward. Since the previous derivation only requires  $i_{\min}$ ,  $i_{\max}$ ,  $x_{\min}$  and  $x_{\max}$ , one can redefine these values for each center for each dimension, then apply the previous analysis. First, the main domain is separated into smaller subdomains at mid-points between centers, defining the  $x$  boundaries. Second, the cells also need to be distributed amongst the various centers. One can choose to distribute equally the cells to each subdomains, but in the case where the different subdomains vary greatly in size, the concentration of cells will be too high for smaller subdomains compared to bigger ones. A more careful approach is to give the same ratio of total cells to the ratio of space covered by the subdomains.

So far, we have outlined the general expressions of the mapping  $x(i)$  and the procedure to analytically calculate the curve in the three regions without defining the source function  $S(x')$  that is used in (15). This function can be user-defined depending on the problem to be solved. We explore in the next section a shape for  $S(x')$  that can be used in our specific case of a Schrödinger solver.

### 3.3. Nonlinear mapping for the Coulomb potential

In this section we discuss a form of  $x(i)$  which may be used to solve the Schrödinger equation with one or several Coulomb

potentials. Near the singularity, the potential is deep and the electrostatic field is strong, requiring a finer grid to resolve the wavefunction. We have found that we can base our nonlinear mapping function on the electrostatic potential. Using

$$S(x') = \frac{-1}{|x' - x_0|} \quad (20)$$

in Eq. (15), the procedure in the previous section leads to the following inverse-mapping function:

$$i_1(x) = A \ln \left( \frac{x_0 - x_{\min}}{x_0 - x} \right) + i_{\min}, \quad (21a)$$

$$i_2(x) = (x - x_0 + d) / (\Delta x_{\min}) + i_{\min}, \quad (21b)$$

$$i_3(x) = A \ln \left( \frac{x - x_0}{d} \right) + i_{\min}. \quad (21c)$$

To obtain  $d$ , we substitute Eqs. (21) in Eq. (17) and isolate  $d$ . In this specific case, we obtain

$$\Delta x_{\min} (i_{\max} - i_{\min}) = d \left[ \ln \frac{(x_0 - x_{\min})(x_{\max} - x_0)}{d^2} + 2 \right] \quad (22)$$

which we solve using bisection method with an initial guess of  $d = 0$  for a given minimum cell size  $\Delta x_{\min}$ . The following mapping is then obtained

$$x_1(i) = x_0 + (x_{\min} - x_0) \exp \left\{ \frac{i_{\min} - i}{A} \right\}, \quad (23a)$$

$$x_2(i) = \Delta x (i - j_1) + x_{\min}, \quad (23b)$$

$$x_3(i) = d \left( \exp \left\{ \frac{i - j_2}{A} \right\} - 1 \right) + x_{\min}. \quad (23c)$$

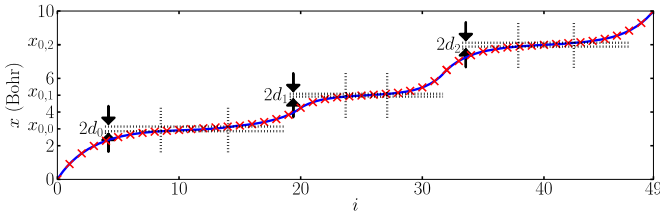
This mapping can be applied to any number of centers by adjusting the  $i_{\min}$ ,  $i_{\max}$ ,  $x_{\min}$  and  $x_{\max}$  parameters accordingly for each dimension.

Kokoouline et al. [28] also proposed a mapping procedure based on the potential of the system, in their case for  $R^{-3}$  and  $R^{-6}$  potentials. Our main idea for the mapping, while similar, is more generic and can be applied to more problems. Indeed, the systems solved here are three-dimensional and are not restricted to one dimension. In addition, due to the Hamiltonian diagonalization required by the Fourier method, a second transformation is required in Ref. [28] to obtain an easier to diagonalize symmetrical matrix. Such a limitation is absent in the method proposed here since the Schrödinger equation is solved explicitly using the FDTD algorithm to obtain the actual wavefunction without the need to re-transform it.

As a simple example, Fig. 1 shows the mapping  $x(i)$  for a domain  $0 \leq x \leq 10$  Bohr where three centers are located at  $x = 3, 5$  and  $8$  Bohr. The continuous blue curve shows the nonlinear mapping function obtained from Eqs. (23). The  $N = 50$  integer values of  $i$  are indicated by red crosses. With a minimum cell size of  $\Delta x_{\min} = 0.05$  Bohr, the linear regions have widths of  $d_0 = 0.138$ ,  $d_1 = 0.0853$  and  $d_2 = 0.114$  Bohr. These widths are not identical since Eq. (22) depends on the different center positions and subdomain sizes. We see that the density of points is higher near the location of the charge centers, where the mapping is linear. As illustrated by our example, the nonlinear mapping procedure is insensitive to whether or not the centers are equally spaced.

The script used to generate Fig. 1 is available in Ref. [38] where the reader can see the details of the implementation.

Applying this one-dimensional method to three dimensions is straightforward. All ion locations are projected on the three spatial axes. The nonlinear mapping is then carried out along each axis separately. If two or more charge centers are located at nearly the same point along one of the axes, then the mapping is calculated such that these charge centers have "merged" and are located at the center of charge of the merged group.



**Fig. 1.** (Color online.) Mapping  $x(i)$  for three ions at  $x_{0,0} = 3$ ,  $x_{0,1} = 5$  and  $x_{0,2} = 8$  Bohr for  $N = 50$ . The regions near the ion that employ a linear mapping ( $\Delta x_{\min} = 0.05$  Bohr) are enclosed within the black dashed lines. Red crosses are located at integer values of  $i$ . See Ref. [38] for the implementation details.

#### 4. Validation and results

To validate our nonlinear mapping model, we consider a static potential  $V(\mathbf{r})$  resulting from  $N_{\text{ion}}$  positive charges  $Z_n$  at positions  $\mathbf{r}_n$ ,

$$V(\mathbf{r}) = - \sum_{n=0}^{N_{\text{ion}}} \frac{Z_n}{|\mathbf{r} - \mathbf{r}_n|}. \quad (24)$$

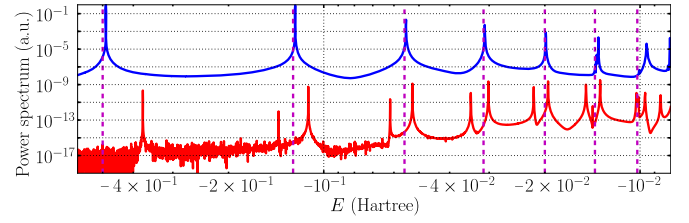
This is used with both the real and imaginary time evolution methods, even though the former can, in general, be applied to systems with a time-dependent  $V$ . Due to the varying cell sizes in the simulation domain, a different upper bound for the time step is required by each grid point. By taking the smallest value set by the smallest inter-grid-point distance, stability is ensured for the whole domain. We consider two cases below: the hydrogen atom and the hydrogen cation molecule  $\text{H}_2^+$ .

##### 4.1. Hydrogen atom

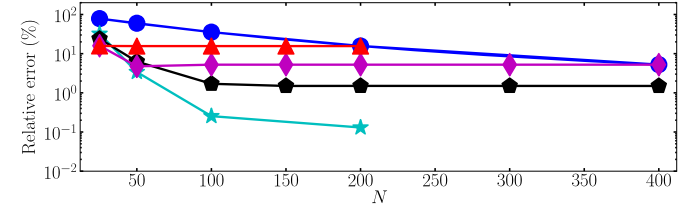
As an initial test our nonlinear mapping, we apply it in this section to a single center system, the hydrogen atom. We use the real time method described in Section 2.1 to calculate the energy eigenvalue spectrum. Fig. 2 shows the calculated spectrum of hydrogen using both linear mapping (red, lower curve) and nonlinear mapping (blue, upper curve) for the same domain size and same number of grid points. The hydrogen ion has a charge state of  $Z = +1$  and is located at the center of a box of size  $150 \times 150 \times 150 \text{ Bohr}^3$ ; for the computation,  $100 \times 100 \times 100 = 10^6$  grid cells were used. The red spectrum was obtained with linear mapping, where  $\Delta x = 1.5$  Bohr. The minimum cell size of the nonlinear mapping was  $\Delta x_{\min} = 0.25$  Bohr. The dashed magenta vertical lines show the analytical eigenvalues  $E_n = -0.5n^{-2}$  Hartree where  $n$  is the principal quantum number.

The results of Fig. 2 indicate that the nonlinear mapping gives much better results than the linear mapping, for the same box size and same number of cells. For the first six eigenvalues, the spectrum obtained via nonlinear mapping has a relative error 0.2–2% with respect to the expected analytical value. The relative error for the ground state energy obtained via the linear mapping is 26%. The other spectral peaks obtained via linear mapping have split as the grid is just too coarse to adequately sample the eigenstates. The splitting is probably due to a lifting of the degeneracy for the  $n \geq 2$  states.

From Fig. 2, we can see that the nonlinear mapping did not adequately resolve the seventh state. This is due to the box size we have chosen. By plotting the analytical solution of the eigenstates associated with the seventh eigenvalue, we verified that there is significant amplitude outside of our chosen simulation box. We were restricted in the size of the chosen box by the linear mapping, since increasing the size while keeping the number of points constant would have reduced significantly the precision. With nonlinear mapping we could have easily extended the box size without sacrificing performance or accuracy.



**Fig. 2.** (Color online.) Hydrogen spectrum calculated via the real time method of Section 2.1 using both linear mapping (red, lower curve,  $\Delta x = 1.5$  Bohr) and nonlinear mapping (blue, upper curve,  $\Delta x_{\min} = 0.25$  Bohr). Each method used a  $150 \times 150 \times 150 \text{ Bohr}^3$  domain and  $100 \times 100 \times 100 = 10^6$  grid cells.

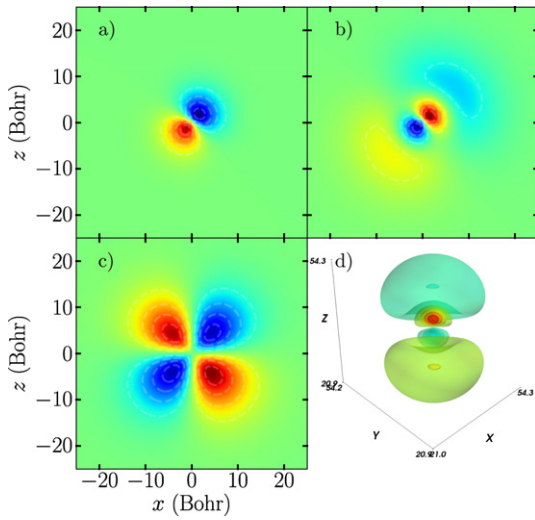


**Fig. 3.** (Color online.) Relative error in the ground state eigenenergy for a hydrogen atom as a function of the number of grid cells per dimension for the linear mapping method (blue circles), and the nonlinear mapping method with  $\Delta x_{\min} = 0.01$  (cyan stars), 0.25 (black pentagons), 0.5 (magenta diamonds) and 1.0 (red triangles). The size of the simulation box was  $200 \times 200 \times 200 \text{ Bohr}^3$ .

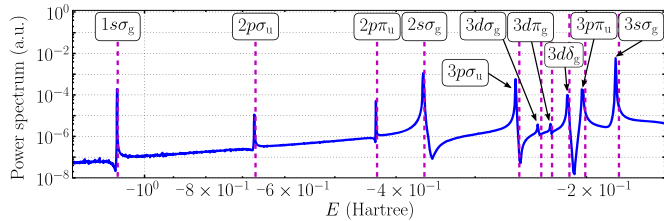
We could improve upon the results of the linear mapping by choosing the grid cell size to be the same as the minimum cell size chosen for the nonlinear mapping. This would give us the same resolution near the ion as that given by the nonlinear mapping. However, for our example, we would then need to increase the number of cells in each dimension by a factor of 6, or the total number of cells by a factor of 216. This would increase the memory requirement from 64 MB to 13.5 GB. In addition to the increase in the actual number of computations, one must also consider that fetching data from main memory is a very slow operation compared to processor speed, so that increasing the memory requirement can significantly augment the code execution time. Thus to achieve the same level of resolution as the nonlinear mapping simulation, which took us 8 days of wall clock time, the linear mapping simulation would require approximately 6 years of wall clock time, clearly an unacceptable duration. These execution times were observed or estimated on a dual-socket Intel Xeon E5520 Nehalem (2.26 GHz) SMP machine parallelized with 16 OpenMP threads.

Consequently, not only can we increase the precision of our calculations using the nonlinear mapping for the same number of cells, but nonlinear mapping also allows a tremendous speedup in the running time over the linear mapping for the same precision, bringing the calculation from almost impossible to easily performed.

In a second set of tests, we calculated the relative error in the calculated eigenenergy of the ground state as a function of the number of grid cells, for nonlinear mapping with different minimum cell sizes and for the linear mapping with the cell size simply determined by the number of points. The box size was kept fixed at  $(200 \text{ Bohr})^3$ . In this case, the results were obtained using the imaginary time method described in Section 2.2, and are plotted in Fig. 3. The linear mapping (blue circles) gives 5% error even with 400 cells per dimension, corresponding to cell size of 0.5 Bohr. Using our nonlinear mapping, the relative error is lowered by a factor of 140 using only 100 cells per dimension for  $\Delta x_{\min} = 0.01$  (cyan stars). The apparent increase in error for smaller values of  $\Delta x_{\min}$  and very small number of grid points is due to a too dense cluster-



**Fig. 4.** (Color online.) Real part of hydrogen's (a) 2p, (b) 3p and (c) 3d eigenstates, found by our imaginary time solver with nonlinear mapping ( $\Delta x_{\min} = 0.25$  Bohr). The 3p state is shown in three dimensions in (d). Green is null, red is positive and blue is negative.



**Fig. 5.** (Color online.) The  $H_2^+$  spectrum for  $R = 2$  Bohr as calculated via our real time nonlinear mapping method, with  $\Delta x_{\min} = 0.05$  Bohr,  $100^3$  grid cells, and a computational volume of  $100^3$  Bohr<sup>3</sup>. The vertical magenta dotted lines are the expected values [39].

ing of these points near the ion with not enough points far from it.

Finally, we demonstrate that eigenstates can be calculated using imaginary time evolution with nonlinear mapping ( $\Delta x_{\min} = 0.25$  Bohr). As an example, we plot in Fig. 4 some of the eigenstates found by our imaginary time solver with nonlinear mapping: a 2p in Fig. 4(a), a 3p in Fig. 4(b) and a 3d in Fig. 4(c). Fig. 4(d) shows the same 3p state as Fig. 4(b) but represented in three dimensions. The resulting energies have been found to match those of hydrogen with an error of 0.4–4%.

#### 4.2. Hydrogen cation molecule

In this section we further validate our nonlinear mapping scheme by applying it to a multi-center system, the hydrogen cation molecule  $H_2^+$ , where two hydrogen nuclei share a single electron.

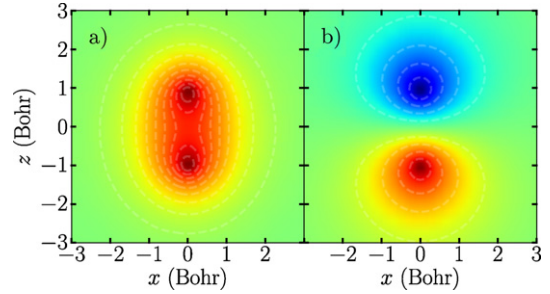
First, we use the real time method described in Section 2.1 to obtain the spectrum of  $H_2^+$  for  $R = 2$  Bohr, where  $R$  is the distance between the two nuclei. The results are plotted in Fig. 5. The vertical magenta dotted lines indicate the known values of the eigenenergies from reference [39] where the internuclear energy has been removed. Unlike the case of the hydrogen atom, the  $n > 1$  states of  $H_2^+$  are not degenerate. Fig. 5 shows that our method captures these states. The eigenenergies of higher excited states could be obtained by choosing a bigger computational volume and a longer running time.

Table 1 compares the values of the first five eigenenergies obtained with our code to the known values [39]. We find that the

**Table 1**

The first five eigenenergies of  $H_2^+$  from known values (first row) [39], and obtained via our real time nonlinear mapping simulation of Fig. 5 (second row). The relative error of our calculated values with respect to the known values is listed in the third row.

| $H_2^+$ states | $\langle E \rangle_{H_2^+}$ | $\langle E \rangle_{NL}$ | Error |
|----------------|-----------------------------|--------------------------|-------|
| $1s\sigma_g$   | -1.1026                     | -1.1054                  | 0.25% |
| $2p\sigma_u$   | -0.66753                    | -0.67034                 | 0.42% |
| $2p\pi_u$      | -0.42877                    | -0.43055                 | 0.41% |
| $2s\sigma_g$   | -0.36086                    | -0.36242                 | 0.43% |
| $3p\sigma_u$   | -0.25541                    | -0.25887                 | 1.3%  |



**Fig. 6.** (Color online.) Real part of  $H_2^+$  ground state and first excited state: (a) shows the ground state ( $1s\sigma_g$  bonding orbital) where blue is the null value and red is positive; (b) shows the first excited state ( $2p\sigma_u$  antibonding orbital), where green is null, red is positive and blue is negative. Parameters are the same as those of Fig. 5.

relative error is between 0.25 and 1.4%. Previous work used an FDTD-based Schrödinger equation solver with a linear mapping to calculate the first two eigenenergies of  $H_2^+$  [3]; their eigenenergies had a relative error compared to known values of 26% for the ground state and 6% for the first excited state. Our nonlinear mapping both increases the precision of the eigenenergies and allows us to access higher excited states than was previously possible.

Finally, we apply the imaginary time method to find the eigenstates of  $H_2^+$ . Figs. 6(a) and 6(b) show the real parts of the ground state ( $1s\sigma_g$ , bonding) and of the first excited state ( $2p\sigma_u$ , antibonding) of  $H_2^+$ , respectively, for an internuclear distance of  $R = 2$  Bohr. These eigenstates were obtained after only 12 and 45 atomic units of imaginary time evolution. These calculated eigenstates agree well with the known solutions [40].

#### 5. Conclusion

We have introduced a novel nonlinear mapping of spatial coordinates applicable to grid-based numerical implementations of partial differential equations. Contrary to previous work, a spatial and continuous variable is directly mapped to a discrete space suitable for computer memory storage, simplifying greatly the implementation details. Moreover, due to the cumulative sum approach, the mapping function can be sourced by any user defined function for a better grid points location control. A single free parameter allows for accurate adjustment of the mapping and thus of the calculation's precision. Furthermore, our mapping can be applied in three-dimensional and nonsymmetric systems and is not tied to any specific problem or numerical method.

We show that, in comparison to standard linear mapping, our method significantly increases accuracy while decreasing memory usage and computation time. We applied our nonlinear mapping to the time-dependent Schrödinger equation for multiple Coulomb potentials. Our solver is based on the FDTD method widely used in electrodynamics, and we have implemented wavefunction propagation in both real and imaginary time. Although we used the Coulomb potential itself as the basis for the nonlinear mapping, the technique itself is generic and can be applied to other partial differential equation solvers.

With nonlinear mapping, we were able to calculate the eigenenergies of hydrogen to within 0.2–2%. Linear mapping gave quantitatively inaccurate results for the same number of grid cells; to obtain the same accuracy as nonlinear mapping would require an increase in the computational time by almost three orders of magnitude. Eigenstates have also been found for hydrogen using nonlinear mapping. This increased efficiency and the fact that our nonlinear mapping scheme is independent of any symmetry, allows for its application to multi-center systems. In particular, we have considered the hydrogen cation molecule  $H_2^+$ , and were able to obtain eigenstates for the bonding and antibonding states. We were also able to calculate eigenenergies for the ground state and several excited states, giving good agreement with expected results, in contrast to an earlier study that employed a linear grid. Our solver thus allows quantum mechanical computations of multiple charge center systems, and could be applied, for example, to studies of atom clusters and molecules exposed to intense radiation.

## References

- [1] C. Lubich, From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis, Zurich Lectures in Advanced Mathematics, 2000.
- [2] A.M.J. Grotendorst, D. Marx (Eds.), Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms, NIC Series, vol. 10, 2002.
- [3] D.M. Sullivan, D.S. Citrin, Determining quantum eigenfunctions in three-dimensional nanoscale structures, *Journal of Applied Physics* 97 (2005) 104305–104311.
- [4] D.M. Sullivan, D.S. Citrin, Time-domain simulation of two electrons in a quantum dot, *Journal of Applied Physics* 89 (2001) 3841–3847.
- [5] A. Soriano, Analysis of the finite difference time domain technique to solve the Schrödinger equation for quantum devices, *Journal of Applied Physics* 95 (2004) 8011–8019.
- [6] D.M. Sullivan, D.S. Citrin, Time-domain simulation of a universal quantum gate, *Journal of Applied Physics* 96 (2004) 1540–15407.
- [7] J. Nagel, A review and application of the finite-difference time-domain algorithm applied to the Schrödinger equation, *Applied Computational Electromagnetics Society Journal* 24 (2009) 1–8.
- [8] A. Taflov, S. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, third edition, Artech House, 2005.
- [9] D.M. Sullivan, Electromagnetic Simulation Using the FDTD Method, Wiley-IEEE Press, 2000.
- [10] I.W. Sudiarta, D.J.W. Geldart, Solving the Schrödinger equation using the finite difference time domain method, *Journal of Physics A: Mathematical and Theoretical* 40 (2007) 1885–1896.
- [11] I.W. Sudiarta, D.J.W. Geldart, Solving the Schrödinger equation for a charged particle in a magnetic field using the finite difference time domain method, *Physics Letters A* 372 (2008) 3145–3148.
- [12] R. Holland, Finite-difference solution of Maxwell's equations in generalized nonorthogonal coordinates, *IEEE Transactions on Nuclear Science* 30 (1983) 4589–4591.
- [13] K.K. Mei, A. Cangellaris, D.J. Angelakos, Conformal time domain finite difference method, *Radio Science* 19 (1984) 1145–1147.
- [14] M. Fusco, FDTD algorithm in curvilinear coordinates (EM scattering), *IEEE Transactions on Antennas and Propagation* 38 (1990) 76–89.
- [15] M. Fusco, M. Smith, L. Gordon, A three-dimensional FDTD algorithm in curvilinear coordinates (EM scattering), *IEEE Transactions on Antennas and Propagation* 39 (1991) 1463–1471.
- [16] P. Mezzanotte, L. Roselli, R. Sorrentino, A simple way to model curved metal boundaries in FDTD algorithm avoiding staircase approximation, *IEEE Microwave and Guided Wave Letters* 5 (1995) 267–269.
- [17] A. Gordon, C. Jirauschek, F.X. Kärtner, Numerical solver of the time-dependent Schrödinger equation with Coulomb singularities, *Physical Review A* 73 (2006) 042505.
- [18] E. Kálnay de Rivas, On the use of nonuniform grids in finite-difference equations, *Journal of Computational Physics* 10 (1972) 202–210.
- [19] C.E. Grosch, S.A. Orszag, Numerical solution of problems in unbounded regions: Coordinate transforms, *Journal of Computational Physics* 25 (1977) 273–295.
- [20] B. Alabi, A mapping finite difference model for infinite elastic media, *Applied Mathematical Modelling* 9 (1985) 6.
- [21] C. Basdevanta, M. Deville, P. Haldenwang, J. Lacroix, J. Ouazzani, R. Peyret, O.P.A. Pateraf, Spectral and finite difference solutions of the Burgers equation, *Computers & Fluids* 14 (1986) 18.
- [22] H. Guillard, R. Peyret, On the use of spectral methods for the numerical solution of stiff problems, *Computer Methods in Applied Mechanics and Engineering* 66 (1988) 17–43.
- [23] J. Shen, X. Yang, An efficient moving mesh spectral method for the phase-field model of two-phase flows, *Journal of Computational Physics* 228 (2009) 2978–2992.
- [24] H. Dwya, B. Sandersa, Numerical modeling of unsteady flame propagation, *Acta Astronautica* 5 (1978) 1171–1184.
- [25] M. Znojil, One-dimensional Schrödinger equation and its “exact” representation on a discrete lattice, *Physics Letters A* 223 (1996) 5.
- [26] E. Fattal, R. Baer, R. Kosloff, Phase space approach for optimizing grid representations: The mapped Fourier method, *Physical Review E* 53 (1996) 1217–1227.
- [27] U. Kleinekathofer, D.J. Tannor, Extension of the mapped Fourier method to time-dependent problems, *Physical Review E* 60 (1999) 4926.
- [28] V. Kokoouline, O. Dulieu, R. Kosloff, F. Masnou-Seeuws, Mapped Fourier methods for long-range molecules: Application to perturbations in the  $Rb_2(O_u^+)$  photoassociation spectrum, *Journal of Chemical Physics* 110 (1999) 9865.
- [29] V. Meshkov, A. Stolyarov, R. Le Roy, Adaptive analytical mapping procedure for efficiently solving the radial Schrödinger equation, *Physical Review A* 78 (2008) 052510.
- [30] J.M. Pérez-Jordá, Variational solution of the Schrödinger equation using plane waves in adaptive coordinates: The radial case, *Journal of Chemical Physics* 132 (2010) 024110.
- [31] Z. Xiaojun, Y. Zhiyuan, L. Weigan, Characteristics of eccentric coaxial line using conformal mapping and finite-difference time-domain method, *Microwave and Optical Technology Letters* 16 (1997) 249–252.
- [32] Y. Yang, R.S. Chen, Z. Ye, An improved formalism for the Crank–Nicolson finite-difference time-domain by conformal mapping method, *Microwave and Optical Technology Letters* 51 (2009) 2378–2382.
- [33] P.E. Grabowski, Pseudospectral calculation of the wave function of helium and the negative hydrogen ion, *Physical Review A* 81 (2010) 032508.
- [34] D.M. Sullivan, D.S. Citrin, Determination of the eigenfunctions of arbitrary nanostructures using time domain simulation, *Journal of Applied Physics* 91 (2002) 3219–3227.
- [35] W. Dai, G. Li, R. Nassar, S. Su, On the stability of the FDTD method for solving a time-dependent Schrödinger equation, *Numerical Methods for Partial Differential Equations* 21 (2005) 1140–1154.
- [36] M. Strickland, D. Yager-Elorriaga, A parallel algorithm for solving the 3d Schrödinger equation, *Journal of Computational Physics* 229 (2010) 6015–6026.
- [37] D. Andrae, J. Hinze, Numerical electronic structure calculations for atoms. I. Generalized variable transformation and nonrelativistic calculations, *International Journal of Quantum Chemistry* 63 (1997) 65–91.
- [38] N. Bigaouette, E. Ackad, L. Ramunno, <https://github.com/nbigaouette/nonlinearmapping>, 2010.
- [39] M.M. Madsen, J.M. Peek, Eigenparameters for the lowest twenty electronic states of the hydrogen molecule ion, *Atomic Data and Nuclear Data Tables* 2 (1970) 171–204.
- [40] R.F.P. Atkins, *Molecular Quantum Mechanics*, 4th edition, Oxford University Press, 2004.