PHYSICAL REVIEW E **73**, 036708 (2006)

Parallel solver for the time-dependent linear and nonlinear Schrödinger equation

Barry I. Schneider*
Physics Division, National Science Foundation, Arlington, Virginia 22230, USA
and Electron and Optical Physics Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA

Lee A. Collins† and S. X. Hu‡
Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

(Received 8 December 2005; revised manuscript received 3 February 2006; published 28 March 2006)

A solution of the time-dependent Schrödinger equation is required in a variety of problems in physics and chemistry. These include atoms and molecules in time-dependent electromagnetic fields, time-dependent approaches to atomic collision problems, and describing the behavior of materials subjected to internal and external forces. We describe an approach in which the finite-element discrete variable representation (FEDVR) is combined with the real-space product (RSP) algorithm to generate an efficient and highly accurate method for the solution of the time-dependent linear and nonlinear Schrödinger equation. The FEDVR provides a highly accurate spatial representation using a minimum number of grid points (N) while the RSP algorithm propagates the wave function in $O(N)$ operations per time step. Parallelization of the method is transparent and is implemented here by distributing one or two spatial dimensions across the available processors, within the message-passing-interface scheme. The complete formalism and a number of three-dimensional examples are given; its high accuracy and efficacy are illustrated by a comparison with the usual finite-difference method.

DOI: 10.1103/PhysRevE.73.036708 PACS number(s): 02.70.Dh, 32.80.Pj, 32.80.Rm

I. INTRODUCTION

Many problems in atomic-scale physics rely on the development of efficient numerical methods for the solution of the time-dependent Schrödinger equation (TDSE). A representative, although by no means exhaustive, sample includes (1) the scattering of electrons or atoms from other atomic or molecular targets [1–6], (2) the interaction of short-pulse laser fields with atoms [7–11], and (3) the evolution of ultracold atomic systems such as Bose-Einstein condensates [12–14]. The latter, which involves the solution of a nonlinear, time-dependent equation, bears marked similarity to the equations governing the propagation of optical pulses in nanostructured optical fibers. Light pulses need to be coupled and controlled at will through appropriately engineered materials with wavelength-scale morphology. Theoretical modeling of these phenomena is important to gain a fundamental understanding of atomic and optical processes at the microscopic level and to provide a scientific basis for the design and development of nanostructured materials for use in optical communication, optical sensors, and advanced computation, which demands theoretical and large-scale computational modeling of all of these phenomena.

The numerical solution of the time-dependent Schrödinger equation relies heavily on the discretization of the variables $(\boldsymbol{r},t)$ in coordinate or function space [15]. Boundary conditions, which depend on the values (derivatives) of the wave function at specific spatial points or surfaces, may be incorporated by imposing these particular constraints on the solution at those points or surfaces. For example, if finite-difference techniques are used for the spatial variables, the values of the wave function at these special locations are required to take specific values which make the discretization process deterministic. Discretization procedures in function space, typically, incorporate the boundary conditions by requiring that the basis set satisfy the requisite values. Initial conditions, such as those involved in the time coordinate, are imposed by requiring the wave function to behave in a fixed fashion at the initial time point. As the system evolves in time, it may be critical from a computational standpoint to adapt the grid or basis set to reflect the changes in the wave function which result from the propagation. In addition, the time propagation itself may require time steps which vary significantly over the physical domain of interest. While it is straightforward to change the size of the time step in the calculation, it is more difficult and time consuming to change the basis set and/or grid. The latter requires a recomputation of matrix elements of the Hamiltonian which for many methods is not a trivial part of the overall calculation. The “regridding” question will be addressed in a later paper.

In this exposition, we develop the basic formulation of the real-space-product, finite-element discrete-variable-representation (RSP-FEDVR) approach [16] for solving the three-dimensional, multiparticle time-dependent Schrödinger equation. The approach employs a RSP technique for evolving the solution in time, which gives the optimal factorization of the exponential propagation operator for a particular order in the temporal variable. A partition of the spatial variables into finite elements (FE’s) provides an efficient means of spanning space with a flexible grid that yields a sparse representation. A DVR basis then determines the wave function within each element. The DVR gives an economical means of accurately calculating the matrix elements of the

*Electronic address: bschneid@nsf.gov
†Electronic address: lac@lanl.gov
‡Electronic address: suxing@lanl.gov

various operators on this basis. This occurs through the close association of the DVR basis functions with the classical orthogonal polynomials and with a particular Gauss quadrature. The integrated method supplies a powerful means of addressing very complicated applications. In addition, the form taken by the propagation operator readily accommodates current parallel computer architectures and routinely demonstrates linear scaling.

We organize the paper into six sections. Section II gives a general, formal overview of the RSP-FEDVR method while Sec. III provides the intricate computational details. In Sec. IV, we describe several representative applications that test both the efficiency and scaling of the approach, and in Sec. V, we present the results of the application of the RSP-FEDVR on these specific cases. In this section, we also detail several parallel scaling strategies. Finally, Sec. V contains a brief summary.

II. GENERAL FORMALISM

A. Algebraic reduction

The N-particle, time-dependent Schrödinger equation may be generally written as

$$i\hbar\frac{\partial}{\partial t}\Psi(\boldsymbol{r},t)=\hat{H}(\boldsymbol{r},t)\Psi(\boldsymbol{r},t)\equiv[\hat{T}(\boldsymbol{r})+\hat{V}(\boldsymbol{r},t)]\Psi(\boldsymbol{r},t), \quad (1)$$

where the caret designates an operator. The Hamiltonian contains kinetic $\hat{T}(\boldsymbol{r})$ and potential $\hat{V}(\boldsymbol{r},t)$ energy components in which the latter may contain nonlinear and time-dependent contributions. In an N-particle system, the kinetic energy has the form

$$\hat{T}(\boldsymbol{r})=\sum_{i=1}^{N}\hat{T}(\boldsymbol{r}_i), \quad (2)$$

with the individual operators usually given by

$$\hat{T}(\boldsymbol{r}_i)=-\frac{\hbar^2}{2M_i}\nabla_i^{\,2}, \quad (3)$$

where M_i is the mass of the ith particle. The potential energy typically consists of a complex function of the interparticle coordinates with $\boldsymbol{r}=(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N)$. The usual strategy expands the unknown wave function in a finite set of N_b known spatial functions (similar to the pseudospectral method [17]),

$$\Psi(\boldsymbol{r},t)=\sum_{q=1}^{N_b}C_q(t)\Phi_q(\boldsymbol{r}), \quad (4)$$

and reduces the time-dependent Schrödinger equation to a set of coupled equations for the coefficients C_q,

$$i\hbar\frac{\partial}{\partial t}C_q(t)=\sum_{p=1}^{N_b}H_{q,p}(t)C_p(t), \quad (5)$$

where

$$H_{q,p}(t)=\int\Phi_q(\boldsymbol{r})^{\dagger}\hat{H}(\boldsymbol{r},t)\Phi_q(\boldsymbol{r})d\boldsymbol{r}, \quad (6)$$

with the dagger (†) the Hermitian conjugate.

We could just as easily have included some of the time dependence directly into the expansion basis. In general, this leads to a set of coupled, nonlinear, time-dependent equations of greater complexity than Eq. (5), but needing many fewer expansion functions. One example of this is the multiconfiguration, time-dependent Hartree method developed by Beck *et al.* [2]. For illustrative purposes, we shall employ a time-independent basis since many important features become more transparent in this representation. In addition, we shall confine our treatment to a product spatial basis although the approach yields to more complex formulations. For a single particle in a three-dimensional (3D) Cartesian space, the expansion basis becomes

$$\Phi_q(\boldsymbol{r}_1)=u_\alpha(x_1)u_\beta(y_1)u_\gamma(z_1), \quad (7)$$

where q signifies the composite label $(\alpha\beta\gamma)$ and the expansion of a single-particle wave function has the form

$$\Psi(\boldsymbol{r}_1,t)=\sum_{\alpha,\beta,\gamma}C_{\alpha\beta\gamma}(t)u_\alpha(x_1)u_\beta(y_1)u_\gamma(z_1). \quad (8)$$

B. Finite element

Global basis functions can encounter difficulties in describing a complicated function over all space. In this case, FE approaches have demonstrated considerable flexibility [18,19]. These generally involve dividing space into a set of elements with a local basis, connected at the boundaries, within each element. Since the size and number of basis functions may vary among the elements, they can accurately map very complicated structures. As an example, a product function in one of the spatial dimensions that has been divided into N_{elm} elements with n_e basis functions $(k=1,\ldots,n_e)$ in element e, spanning the region $[x_l^{\,e},x_r^{\,e}]$, has the form

$$u_\alpha(x)\equiv u_k^{\,e}(x)\quad[x_l^{\,e}\leqslant x\leqslant x_r^{\,e}], \quad (9)$$

where $\alpha=\sum_{m=1}^{e}n_m+k$ and gives the global label of the kth basis function within the mth element. We select the basis within each element according to a DVR. Drawing, for example, on the close relationship between classical orthogonal polynomials and Gaussian quadratures, the DVR provides an accurate, economical means of representing a function within an element. We present the details in Secs. III A and III B.

C. Temporal propagation

Now that we have established a spatial basis, we return to Eq. (5) rewritten in matrix notation as

$$i\frac{\partial}{\partial t}\boldsymbol{C}(t)=\boldsymbol{H}(t)\boldsymbol{C}(t), \quad (10)$$

where $[\boldsymbol{C}(t)]_q=C_q(t)$ and $[\boldsymbol{H}(t)]_{qp}=H_{qp}(t)$. This represents a first-order differential equation in time for which the propagation from t_i to t_f is given by

$$\boldsymbol{C}(t_f)=T\exp\left(-\frac{i}{\hbar}\left[\int_{t_i}^{t_f}dt'\boldsymbol{H}(t')\right]\right)\boldsymbol{C}(t_i) \quad (11a)$$

$$\equiv U(\boldsymbol{H},t_f,t_i)\boldsymbol{C}(t_i), \tag{11b}$$

where T is the time-ordering operator and U the temporal propagator. For a time-independent Hamiltonian or one that changes little over the interval $\delta t = t_f - t_i$, we have

$$U(\boldsymbol{H},\delta t) = \exp\left[-\frac{i}{\hbar}\boldsymbol{H}\delta t\right]. \tag{12}$$

We must develop techniques to determine efficiently the propagator in order to advance the coefficient vector in time. The simplest scheme involves expanding the exponential operator [20] in Eq. (12) in a Taylor series of δt as

$$U(\boldsymbol{H},\delta t) = \sum_n \left[\frac{-i\delta t}{\hbar}\right]^n \boldsymbol{H}^n. \tag{13}$$

Approximations to the propagator result from truncations of this expansion at different orders of δt. It is important to examine these truncations carefully as they are not always stable. For example, the Euler method, which keeps terms to $O(\delta t)$, is unstable. If we disregard the effort needed to compute the time-dependent part of the interaction potential, the numerical effort required to advance the wave function from one time to the next depends, at a minimum ($O[\delta t]$), on $N_b{}^2$, the cost of a single, matrix-vector multiplication (N_b is the total number of basis functions). Typically, the cost is even larger since most approaches require more than a single matrix-vector multiplication or even a matrix inversion. In most instances, the time dependence of the interaction potential is simple and does not dominate the cost of the computation. We have found that the basic real-space-product form [21] and its derivatives [10,14] provide a systematic means of representing the temporal propagator to various orders as well as maintaining stability and conserving the norm. We give further details in Sec. III C.

III. COMPUTATIONAL DETAILS

A. Discrete variable representation

The discrete variable representation was introduced into chemical physics by Light, Hamilton, and Lill in 1985 [22]. Since that time it has played a major role in reducing the numerical effort required to compute Hamiltonian matrix elements in a broad array of quantum-mechanical problems. These include the calculation of the vibrational levels of complex molecules, heavy particle nonreactive and reactive scattering, and solutions of the nonlinear Schrödinger equation. The major advantage of the DVR derives from the relationship between the DVR basis and a numerical grid. In the DVR basis, matrix elements of local operators of the coordinates may be replaced, to a very high degree of accuracy, by their values on the numerical grid. A similarity transformation, which diagonalizes the coordinate operator in the DVR basis, mathematically connects the numerical grid and the basis. For our application, we employ the DVR to construct a basis within each of the finite elements. We shall use as an example one element e with n_e basis functions spanning $[x_l{}^e, x_r{}^e]$ for a single spatial coordinate x since repeated application of the prescription in each spatial dimension will permit the construction of the full multidimensional product basis.

The DVR's used in the present study all derive from the classical orthogonal functions, satisfying the three-term recursion relationship

$$\beta_n\phi_n = (x-\alpha_{n-1})\phi_{n-1} - \beta_{n-1}\phi_{n-2}. \tag{14}$$

An associated Gauss quadrature rule exists having the same number of points and weights $[x_i{}^e, w_i{}^e]$ that will exactly integrate any integrand whose degree is $(2n_e - 1)$ or less. The points and weights may be obtained from diagonalization of the tridiagonal matrix built from the recursion coefficients α and β. The recursion coefficients are known analytically for all classical orthogonal functions but may be computed for any positive-definte weight function via the Lanczos procedure [16]. As a corollary the orthonormality integral and the matrix representation of the coordinate operator involving the truncated basis are given *exactly* within the quadrature rule. Under these conditions, we can transform from the original spectral basis of n_e functions, ϕ^e, to a new basis of DVR or coordinate functions called, u^e, as follows:

$$u_i{}^e(x) = \sum_{n=0}^{n_e-1} c_n{}^e \phi_n{}^e(x), \tag{15a}$$

$$c_n{}^e = \langle \phi_n{}^e | u_i{}^e \rangle = \int_{x_l{}^e}^{x_r{}^e} w(x)\phi_n{}^e(x)u_i{}^e(x)dx \tag{15b}$$

$$= \sqrt{w_i{}^e}\phi_n{}^e(x_i^e). \tag{15c}$$

The coordinate functions also have a representation in terms of the Lagrange interpolation prescription, which may prove more convenient for calculational purposes,

$$u_i^e(x) = \sum_{n=0}^{n_e-1} \sqrt{w_i{}^e}\phi_n{}^e(x_i^e)\phi_n{}^e(x),$$

$$= \frac{1}{\sqrt{w_i{}^e}} \prod_{q=1}^{n_e}{}' \frac{x - x_q{}^e}{x_i - x_q{}^e}. \tag{16}$$

These function also have several simple, useful properties

$$u_i{}^e(x_j{}^e) = \frac{\delta_{i,j}}{\sqrt{w_i{}^e}}, \tag{17a}$$

$$\langle u_i{}^e | x | u_j{}^e \rangle = \delta_{i,j} x_i{}^e. \tag{17b}$$

While we have performed all of the integrals using the Gauss quadrature rule, we have made no approximations as a consequence of the fact that the quadrature rule is exact for all integrands $(2n_e - 1)$ or less. Also, the DVR functions are defined as continuous functions of the spectral basis. This exact duality is guaranteed when ϕ^e is one of the classical orthogonal functions. Moving from the spectral to the DVR representation offers no particular advantage unless we do something further. Consider a matrix element of some general $V(x)$ in the DVR basis,

$$\langle u_i^{\,e}|V|u_j^{\,e}\rangle = \sum_{q=0}^{\infty} \frac{V_q}{q!}\langle u_i^{\,e}|x^q|u_j^{\,e}\rangle, \tag{18}$$

where we have expanded the potential in an infinite power series in x. If the basis set u^e were complete, we could insert the complete set between each power of the coordinate, use the fact that x is diagonal in the basis, and the result would be

$$\langle u_i^{\,e}|V|u_j^{\,e}\rangle = V_i^e \delta_{i,j}. \tag{19}$$

Exactly the same result may be obtained by using the Gauss quadrature rule to evaluate the integral. This is not an accident since the two representations are equivalent. The DVR representation assumes that the quadrature-matrix representation is sufficiently accurate and complete that matrix elements of a general $V(x)$ may be evaluated by the N-point quadrature rule. In the DVR representation these matrix elements are then diagonal and equal to their value at the Gauss quadrature points. This approximation has been shown to be accurate in a broad class of problems. Consequently, transforming to the DVR basis has significant advantages over the original spectral basis in that complex matrix elements of the potential are diagonal and equal to their values as the quadrature points. The resulting sparse matrix representation of the Hamiltonian may be exploited when performing matrix vector multiplications such as those required in iterative eigenvalue or linear systems methods.

B. Finite-element DVR

The FEDVR takes these ideas one step further by combining the basic finite-element method with the DVR [19]. In its most elementary incarnation, each spatial coordinate is divided into subregions (elements) and a Legendre DVR of arbitrary order used in each element. Requiring continuity of the DVR functions at the element boundaries connects the subregions. To do this requires that the first and last points in each interval coincide. This in turn necessitates the introduction of a simple generalization of the Gauss quadrature rules, known as Gauss-Lobatto quadrature, in which the end points of the interval are specified in advance. A "bridge" function of construct

$$\hat{u}^e_{n_e}(x) = \frac{u^e_{n_e}(x) + u_1^{e+1}(x)}{\sqrt{w^e_{n_e} + w_1^{e+1}}} \tag{20}$$

guarantees the continuity across the element boundaries. The union of all of the internal DVR functions in each interval plus the bridge functions forms the global basis. For example, in the special case that all N_{elm} elements had the same number of functions, n_e, the total number of basis functions would equal $N_{elm}n_e-(N_{elm}-1)$. One remaining point concerning boundary conditions, discussed in more detail in Ref. [23], deserves mentioning. If the solution to the Schrödinger equation is required to vanish at the boundary points, it is necessary to discard any basis function from the expansion set which is nonvanishing at that point. Since these points are part of the Gauss-Lobatto rule, this is simple to do. Failure to do so will result in unpredictable and erroneous results.

Very sparse Hamiltonian matrices can result from the FEDVR. The sparsity depends on the number of DVR basis functions in an element and purely derives from the kinetic energy matrix in that dimension. For example, if we had four elements with a $(4/3/3/2)$ basis set distribution, the Hamiltonian would have the following appearance:

$$\begin{pmatrix} H_{1,1} & H_{1,2} & H_{1,3} & H_{1,4} & 0 & 0 & 0 & 0 & 0 \\ H_{2,1} & H_{2,2} & H_{2,3} & H_{2,4} & 0 & 0 & 0 & 0 & 0 \\ H_{3,1} & H_{3,2} & H_{3,3} & H_{3,4} & 0 & 0 & 0 & 0 & 0 \\ H_{4,1} & H_{4,2} & H_{4,3} & H_{4,4} & H_{4,5} & H_{4,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & H_{5,4} & H_{5,5} & H_{5,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & H_{6,4} & H_{6,5} & H_{6,6} & H_{6,7} & H_{6,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & H_{7,6} & H_{7,7} & H_{7,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & H_{8,6} & H_{8,7} & H_{8,8} & H_{8,9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & H_{9,8} & H_{9,9} \end{pmatrix}.$$

In the next section, we shall demonstrate how to exploit this sparsity in the temporal propagation.

C. Real-space-product formalism

A variety of approaches exist for propagating an initial wave packet from $t=0$ to times at which specific physical parameters may be extracted from the time-evolved wave function. These include all of the standard techniques for integrating any first-order differential equation, as well as more specialized approaches such as the Arnoldi-Lanczos method [24,25], the Crank-Nicholson method, and split-operator technique [21,25–30]. These methods are either termed explicit or implicit depending on whether the calculation of the wave function at $t=t_0+\delta t$ does (implicit) or does not (explicit) require the solution of a set of linear equations. Explicit methods typically require one or more operations of the Hamiltonian matrix on a known vector. Of the commonly used approaches, the Crank-Nicholson method is an implicit method, while the split-operator technique is explicit. Implicit methods necessitate the efficient solution of a large set of algebraic equations. If the equations have some specific structure, such as being tridiagonal, this is easy. For the explicit approaches, the rate determining step is the multiplication of the Hamiltonian on a known vector. Again, exploitation of the structure of the Hamiltonian becomes of paramount concern. For example, if the matrix is sparse, this can be exploited to reduce the cost of propagation significantly. Another desirable feature of any method that propagates the quantum-mechanical wave function is stability and, for real-time propagation, unitarity.

We return now to Eq. (12) for the general temporal propagator and note that this expression remains valid for any value of $\delta t=(t_f-t_i)$. However, if δt is small, then we may approximate this exponential using a variety of methods. These include power-series expansions, Runge-Kutta, Crank-Nicholson, Lanczos-Arnoldi, Chebyshev, and split operator techniques. For a time-dependent Hamiltonian, approximations exist to the time-ordered exponential for short enough times, but the time required in the Hamiltonian may not be t_i. This can lead to extra work if the Hamiltonian contains nonlinear terms that depend on the wave function. The most frequently used approximation replaces the time in the Hamiltonian by t_i and thus Eq. (11) becomes

$$\boldsymbol{C}(t_i + \delta t) = U(\boldsymbol{H}(t_i),\ \delta t))\boldsymbol{C}(t_i) \tag{21}$$

$$= \exp\left(- \frac{i}{\hbar}\boldsymbol{H}(t_i)\delta t \right)\boldsymbol{C}(t_i). \tag{22}$$

Other methods, such as the Crank-Nicholson or exponential midpoint method, need the Hamiltonian at $\left(t_i + \frac{\delta t}{2}\right)$ while a time-reversed symmetry propagator requires the Hamiltonian at both t_i and t_f. Suzuki and Yamauchi [30] provided a time-dependent generalization of the Lie-Trotter-Suzuki (LTS) technique they developed for a time-independent Hamiltonian. The time-dependent version of the Suzuki-Yamauchi approach requires that the Hamiltonian be evaluated at a number of intermediate times, whose values depend on the order of the method. This is needed in order to retain the formal accuracy of the method. In practice, the higher-order (>2) Suzuki-Yamauchi formulas are cumbersome for some nonlinear problems, but this may not be of importance in practice. For our purposes, the main advantage of the LTS formalism rests with the development of approximations to the time evolution operator in which the Hamiltonian consists of a sum of non-commuting operators. A suitable decomposition of a Hamiltonian matrix of the form $H_1 + H_2(t)$ to second-order accuracy is

$$U_2(\boldsymbol{H}, t + \delta t, t) = U_1\left(\boldsymbol{H}_2(t + \delta t), \frac{\delta t}{2}\right)U_1(\boldsymbol{H}_1, \delta t)U_1\left(\boldsymbol{H}_2(t), \frac{\delta t}{2}\right) \tag{23}$$

in terms of the simple first-order propagators U_1,

$$U_1(\boldsymbol{H}_1, \tau) = \exp\left(- \frac{i}{\hbar}\boldsymbol{H}_1 \tau \right),$$

$$U_1(\boldsymbol{H}_2(t), \tau) = \exp\left(- \frac{i}{\hbar}\boldsymbol{H}_2(t) \tau \right). \tag{24}$$

Similar splitting algorithms have been examined in recent investigations [31–34]. Higher-order schemes arise from successive applications of U_2, evaluated at intermediate times. Ultimately, the practical question reduces to how efficiently can U_1 be evaluated for the operators in the Hamiltonian. The issue for using higher-order as opposed to lower-order approaches remains clearly a question of accuracy versus the number of arithmetic operations for a fixed time step [16]. We leave that discussion for future publications and only remark here that such comparisons can depend strongly on the problem.

The explicit manner of the implementation of the second-order LTS (RSP2) scheme depends on the structure of the Hamiltonian. In a Cartesian coordinate system in three dimensions, we typically have a Hamiltonian of the form (where the kinetic and potential energy terms should be regarded as operators or matrices)

$$H = T_x + T_y + T_z + V(x, y, z, t). \tag{25}$$

The first step is to split off the time-dependent potential to get

$$\begin{aligned} U_2(H, t + \delta t, t) &= \exp\left(- \frac{i}{\hbar}V(x, y, z, t + \delta t)\frac{\delta t}{2} \right)\exp\left(- \frac{i}{\hbar}[T_x + T_y + T_z]\delta t \right) \\ &\quad \times \exp\left(- \frac{i}{\hbar}V(x, y, z, t)\frac{\delta t}{2} \right). \end{aligned} \tag{26}$$

Since the kinetic energy operators commute, the middle term of the above equation may also be written as the product of exponential operators of T_x, T_y, and T_z.

We may now introduce the FEDVR basis as a representation for the operators. In this basis, $V(x, y, x, t)$ is just a diagonal operator, evaluated at the DVR grid point (x_i, y_j, z_k). Each of the kinetic operators has the block structure of the array pictured in Sec. III B. All that remains is to find an efficient way to compute the exponential of the matrix in Sec. III B. This matrix has an overlapping block structure, a simple generalization of the three-point finite difference formula to larger blocks. The strategy is now obvious; each of the kinetic energy matrices is written as a sum of nonoverlapping, "odd" and "even" blocks, as exemplified by

$$\begin{pmatrix} T_{1,1} & T_{1,2} & T_{1,3} & T_{1,4} & 0 & 0 & 0 & 0 & 0 \\ T_{2,1} & T_{2,2} & T_{2,3} & T_{2,4} & 0 & 0 & 0 & 0 & 0 \\ T_{3,1} & T_{3,2} & T_{3,3} & T_{3,4} & 0 & 0 & 0 & 0 & 0 \\ T_{4,1} & T_{4,2} & T_{4,3} & T_{4,4} & T_{4,5} & T_{4,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & T_{5,4} & T_{5,5} & T_{5,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & T_{6,4} & T_{6,5} & T_{6,6} & T_{6,7} & T_{6,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & T_{7,6} & T_{7,7} & T_{7,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & T_{8,6} & T_{8,7} & T_{8,8} & T_{8,9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_{9,8} & T_{9,9} \end{pmatrix}$$

$$= \begin{pmatrix} T_{1,1} & T_{1,2} & T_{1,3} & T_{1,4} & 0 & 0 & 0 & 0 & 0 \\ T_{2,1} & T_{2,2} & T_{2,3} & T_{2,4} & 0 & 0 & 0 & 0 & 0 \\ T_{3,1} & T_{3,2} & T_{3,3} & T_{3,4} & 0 & 0 & 0 & 0 & 0 \\ T_{4,1} & T_{4,2} & T_{4,3} & T_{4,4}/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & T_{6,6}/2 & T_{6,7} & T_{6,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & T_{7,6} & T_{7,7} & T_{7,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & T_{8,6} & T_{8,7} & T_{8,8}/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & T_{4,4}/2 & T_{4,5} & T_{4,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & T_{5,4} & T_{5,5} & T_{5,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & T_{6,4} & T_{6,5} & T_{6,6}/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_{8,8}/2 & T_{8,9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_{9,8} & T_{9,9} \end{pmatrix}, \tag{27}$$

where we have used T to symbolize any one of the kinetic energy operators. Each of the matrices on the right-hand side of Eq. (27) is composed of block-diagonal submatrices. Each of these small blocks may be easily diagonalized before beginning the propagation [10,16].

By splitting each of the exponentials involving the kinetic energy operator as

$$U_2^T(T,t+\delta t,t) = \exp\left(-\frac{i}{\hbar}T_{odd}\frac{\delta t}{2}\right)\exp\left(-\frac{i}{\hbar}T_{even}\delta t\right)\exp\left(-\frac{i}{\hbar}T_{odd}\frac{\delta t}{2}\right), \tag{28}$$

we can apply the full exponential on a given vector as a series of small, independent, prediagonalized matrices. Communication between the subblocks occurs via the overlapping, interface matrix elements. Since the communication between the blocks is limited, parallelization becomes immediate and transparent (see Sec. V A). In addition, the number of nonzero multiplies needed to propagate a vector from one time step to the next is proportional to N_b, where N_b is the number of three-dimensional grid points. The proportionality factor depends on the number elements employed and the size of the DVR representation in each element. Clearly, it is advantageous to use a large number of elements with as small a DVR per element as possible since that produces the largest number of zeros in the matrix. Therefore, to advance the solution from one time (t) to another $(t+\delta t)$, the RSP-FEDVR approach requires only mN_b operations $(m \ll N_b)$, instead of ${N_b}^2$ for a general dense Hamiltonian matrix.

IV. APPLICATIONS

We consider several representative examples that demonstrate the efficacy of the RSP-FEDVR approach. These include both effective one-particle problems as well as a two-electron case. For the effective one-particle cases $(N=1)$, we solve a 3D TD Schrödinger equation of the form

$$i\hbar\frac{\partial}{\partial t}\Psi(\boldsymbol{r},t) = \left[-\frac{\hbar^2}{2M}\nabla^2 + V(\boldsymbol{r},t)\right]\Psi(\boldsymbol{r},t) \tag{29}$$

for three cases.

(i) *Harmonic oscillator* with a potential (in "trap units," $\omega_x=\omega_y=\omega_z$)

$$V(x,y,z) = \frac{1}{2}(x^2+y^2+z^2). \tag{30}$$

We generally employ "trap units" with length given in units of $d_x=\sqrt{\hbar/M\omega_x}$, time in units of ${\omega_x}^{-1}$, and energy in units of $\hbar\omega_x$.

(ii) *Bose-Einstein condensate* in an aspherical harmonic trap in which the wave function represents all N_a correlated atoms in the system, and the nonlinear equation (Gross-Pitaevski) has the following potential form:

$$V(x,y,z,t) = \frac{M{\omega_x}^2}{2}(x^2+\alpha^2y^2+\beta^2z^2) + g|\Psi(x,y,z,t)|^2, \tag{31}$$

where M is the particle mass, ω_i, the frequency in the ith dimension, $\alpha=\omega_y/\omega_x$, $\beta=\omega_z/\omega_x$, and $g=\frac{4\pi\hbar^2 aN_a}{M}$ with a the scattering length. We consider two trap configurations.

(a) *Case 1 (cylindrical).* Rb(M=87 amu), $\alpha=1$, $\beta=0.6386$, $\omega_x=2\pi\times 8.3$ rad/s, a=5.77 nm, $N_a=1.\times 10^6$, giving d_x=3.74 μm and $g=5.816\times 10^4$;

(b) *Case 2 (aspherical).* Na(M=23 amu), $\alpha=\sqrt{2}$, $\beta=2$, $\omega_x=2\pi\times 354$ rad/s, a=2.75 nm, $N_a=2^{17}$, giving d_x=1.565 μm and $g=2.873\times 10^3$.

(iii) *Hydrogen atom in a laser field* of an ultrashort, circularly polarized (CP) few-cycle pulse:

$$V(\boldsymbol{r},t) = -\frac{1}{\sqrt{x^2+y^2+z^2}} + xE_x(t) + yE_y(t). \tag{32}$$

The CP field has been decomposed into two components E_x and E_y, which are derived from the vector potentials $A_x(t) = -\frac{E_0}{\omega}\sin^2\left[\frac{\pi t}{T}\right]\sin(\omega t+\phi)$ and $A_y(t) = -\frac{E_0}{\omega}\sin^2\left[\frac{\pi t}{T}\right]\cos(\omega t+\phi)$, where T is the pulse duration; E_0 and ω are the laser peak field strength and frequency, respectively. In our example, the CP-FCP has a wavelength of 800 nm and a duration (T) of 5 fs. Its maximum intensity equals 2×10^{14} W/cm^2, with a carrier-envelope phase (ϕ) of 60°.

We also make applications to a representative *two-electron system* (N=2): helium. In order to avoid large grids, we take account of spherical symmetry and expand the total wave function in terms of a complete set of two-electron angular functions $\mathcal{Y}_{l_1m_1l_2m_2}^{LM}$,

$$\Psi(\boldsymbol{r}_1,\boldsymbol{r}_2,t) = \sum_{LM}\sum_{l_1m_1}\sum_{l_2m_2}\Psi_{l_1m_1l_2m_2}^{LM}(r_1,r_2,t)\mathcal{Y}_{l_1m_1l_2m_2}^{LM}(\Omega_1,\Omega_2), \tag{33}$$

where $M=m_1+m_2$ and $|(l_1-l_2)| \leq L \leq (l_1+l_2)$. The spin variables have been omitted since they are conserved in our applications and limited to either singlets or triplets. Substituting the expansion (33) into the Schrödinger equation leads to a set of time-dependent, two-dimensional, coupled partial differential equations in the radial variables (r_1,r_2),

$$i\hbar\frac{\partial}{\partial t}\Psi_i(r_1,r_2,t) = (\hat{T}_1 + \hat{T}_2)\Psi_i(r_1,r_2,t) + \sum_{i'} V_{i,i'}(r_1,r_2,t)\Psi_{i'}(r_1,r_2,t), \quad (34)$$

where i stands for the collection of angular variables $(LMl_1m_1l_2m_2)$. The coupling between the partial wave components is due to the Coulomb repulsion of the electrons; the diagonal terms represent the electron-nuclear attraction. The full form of the matrix elements appears in Ref. [8]. The Coulomb interactions conserve the total angular momentum quantum numbers (LM), and in the absence of an external electromagnetic field, the equations block-diagonalize into (LM) components. An external electromagnetic field, depending on the dipole operator, will induce a coupling between the L and $L+1$ and $L-1$ components of the wave function. It is desirable and possible to exploit this limited coupling in the time propagation step. The final step in the process is the expansion of the radial wave function as

$$\Psi_i(r_1,r_2,t) = \sum_{p,q} d_{pq,i}(t)\mathcal{O}(u_p(r_1)u_q(r_2)), \quad (35)$$

where $\mathcal{O}$ symmetrizes or antisymmetrizes the product basis. Thus the problem is reduced to the time propagation of a set of linear or nonlinear algebraic equations.

V. RESULTS AND DISCUSSION

A. Parallel scaling

Calculations to propagate the multidimensional TDSE can often consume considerable computational resources. Consequently, an approach such as the RSP-FEDVR, which, as we shall demonstrate, yields rather easily to parallel constructs, appears very attractive. For this implementation, we have employed the well-developed message passing interface (MPI) [35] on state-of-the-art supercomputers in both a "one-dimensional" and "two-dimensional" decomposition scheme with the details described in the next two sections.

1. 1D decomposition

The 1D-decomposition method provides a simple, straightforward MPI scheme for ease of implementation. For a 3D TDSE in Cartesian coordinates, the kinetic energy operators commute so that we may choose to decompose in one of the dimensions. For example, we consider a 3D space, spanned by $M_x \times M_y \times M_z$ elements, where M_x, M_y, and M_z give the number of elements in the x, y, and z dimensions, respectively. We choose one dimension—say, the x—in which to divide M_x elements onto N CPU's. Thus, on each CPU we have $(M_x/N) \times M_y \times M_z$ elements for processing. Each CPU does its own operations on the loaded elements. In order to balance exactly the loading within the total number of CPU's, we require the ratio of M_x/N be an integer. Furthermore, since the RSP2 propagator splits the elements into even and odd subblocks, we set the ratio M_x/N to an *even* integer. Therefore, by construction, we have an even number of elements on each CPU. This maximizes the number of arithmetic operations within each CPU and minimizes the communication between neighboring CPU's. The basic operation on each CPU consists of the performance of the required matrix-vector multiplications and communication of the values of the new vector at the "interface points" to neighboring CPU's. The communication between CPU's needs to be performed three times, corresponding to the odd-even-odd splitting of the exponential [see Eq. (28)]. The data communication is limited to the boundary ("interface") points of wave functions in neighboring nodes. Defining the coefficient vector on the jth CPU as $C_{n_{x:jq},n_y,n_z}$, where q is the collection of all x points on that CPU, we must send $C_{n_{x:jn_1},n_y,n_z}$, the value of the vector at the first point for all (n_y,n_z) points to the $(j-1)$th CPU. Similarly, we need to pass $C_{n_{x:jn_f},n_y,n_z}$, the last point for all (n_y,n_z) points to the $(j+1)$th CPU. The CPU ID number j runs from 0 to $N-1$ in the MPI scheme. In essence, we are cutting the 3D cubic elements (points) into N *slices*, and only the boundary points need passing between neighboring nodes.

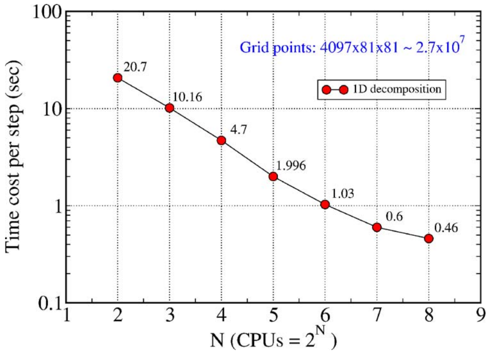

FIG. 1. (Color online) The scaling test of RSP-FEDVR on a supercomputer "QSC" of LANL for the ground state of Bose-Einstein condensate in a 3D cylindrical trap (case 1), which is spanned by a grid size of $4097 \times 81 \times 81 \sim 2.7 \times 10^7$ points. Since such an elongated data set favors our 1D-decomposition scheme, we are observing an "superscaling" from 4 to 128 CPU's.

We have tested the performance of the FEDVR 1D decomposition on the LANL's Alpha cluster "QSC," a supercomputer with 256 Compaq AlphaServer ES45 (a 4-CPU SMP with EV68 1.25-GHz processors). The test problem involves the determination of the ground-state energy of a Bose-Einstein condensate (BEC) in a 3D cylindrical harmonic trap as described in case 1 in Sec. IV. The used grid consisted of $4097 \times 81 \times 81 \sim 2.7 \times 10^7$ points. We used 2048 elements in the x dimension and 40 elements in both the y and z dimensions with a three-point (basis function) DVR per element. Since such an elongated data set favors the current 1D-decomposition scheme, we observe a "superscaling" from 4 to 128 CPU's in Fig. 1. Thus, when we double the number of CPU's, we get more than a factor of 2 increase in speed. This "superscaling" likely results from a reduction in memory accessing as the number of CPU's increases.

For the second scaling test, we examine the same BEC but with less disparity between the grids in each dimension. We employ 1024 elements to span x-dimensional space and 256 elements for y and z axes. Again, a three-point DVR was

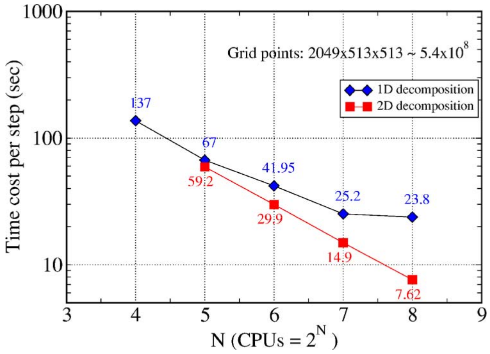

FIG. 2. (Color online) The scaling test of RSP-FEDVR for a larger and less elongated data. We treat the same Bose-Einstein condensate as in Fig. 1 but on a larger 3D grid, which is spanned by a grid size of $2049 \times 513 \times 513 \sim 5.4 \times 10^8$ points. We show both the "1D-decomposition" and "2D-decomposition" results, respectively, in (blue) diamonds and (red) squares. The "1D decomposition" begins to break down as the number of CPU's increases from $N=128$ to $N=256$. However, it shows excellent linear scaling up to $n=256$ CPU's in the case of "2D decomposition."

used per element. Thus, we get a total grid size of $2049 \times 513 \times 513 \sim 5.4 \times 10^8$ points. Another supercomputer cluster "Flash," which is a dual Opteron cluster (2 AMD 2.0-GHz) having 747 nodes (1494-CPU) with Myrinet connection, performed this scaling test. The results are shown in Fig. 2 by the blue "diamond" symbols. We obtained an almost linear scaling as the number of processors increased to $N=128$ CPU's. However, the linear scaling breaks down from $N=128$ to $N=256$ CPU's as a consequence of the lack of balance between arithmetic operations and communications as the number of CPU's increases. Under the 1D-decomposition scheme, the amount of data passed has a fixed size of $(n_y \times n_z)$ points, independent of the number of processors. Thus, the more CPU's used, the more data passed to adjacent CPU's. At some point, message passing begins to dominate the calculation, and as shown in Fig. 2, the scaling begins to break down.

2. 2D decomposition

As we have seen in Fig. 2 the "1D decomposition" begins to deteriorate as the number of CPU's increases from $N=128$ to $N=256$. In this subsection, we shall discuss an improvement based on a two-dimensional decomposition scheme. Instead of slicing a 3D data set along a single dimension (e.g., x dimension as shown above), we can perform a two-dimensional slicing—say, in the xy plane—that reduces the number of "interface points" passed between CPU's as the number of CPU's increases. Intuitively, this approach should lead to better scaling for a very large number of nodes. To accomplish this requires assigning to each CPU a data set, finding its nearest neighbor CPU's, and appropriately handling the "boundary" CPU's. Such an implementation is more complicated than the "1D decomposition." Fortunately, the MPI contains a mechanism for addressing such problems, called the Cartesian communicator [35], which creates a virtual topological, two-dimensional network, in which all neighbors and their boundaries are automatically treated. In addition, we can direct the computer to allocate the virtual neighbor CPU's into real physical neighbor nodes. Utilizing such a virtual topological 2D decomposition provides a more efficient parallelization scheme for the RSP-FEDVR code. Figures 2 and 3 give results for the 2D scheme. In Fig. 2, the same BEC problem was tested by using our 2D-decomposed RSP-FEDVR code, for which the results are plotted in (red) squares. We clearly find that the scaling is now linear up to $n=256$ CPU's. For the largest number of CPU's, the 2D decomposition gives a threefold speeding-up over the 1D-decomposition case, for the same data set.

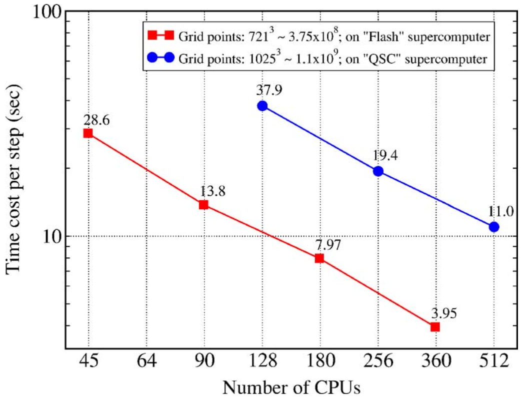

FIG. 3. (Color online) Other examples of the "2D-decomposition" parallelization with the RSP-FEDVR code. Same BEC test model (case 1) as in previous two figures. Tests are performed for different data sets (marked in the figure) on the two LANL's supercomputers "Flash" and "QSC," respectively.

Figure 3 shows other examples of the "2D-decomposition" scheme, for the same BEC problem with equal-dimensional data sets having $721 \times 721 \times 721 \sim 3.75 \times 10^8$ and $1025 \times 1025 \times 1025 \sim 1.1 \times 10^9$ points. Here, we obtain a linear scaling up to 360 and 512 CPU's for the two data sets, respectively. Timing results have been obtained by averaging over 500 time steps.

3. Scaling on the grid size

The above tests considered varying the number of CPU's for the same data set, using a fixed grid size. Now, we turn to another important issue, the scaling of the method with grid size for a fixed number of CPU's. For this purpose, we have performed with the 2D-decomposed RSP-FEDVR code a sequence of tests for a fixed number (=64) of CPU's on the LANL's supercomputer "Flash," varying the total number of elements from 32^3 to 512^3 for the 3D BEC (case 1). We still used the smallest three-point basis for all cases so that the grid size varied from $\sim 2.75 \times 10^5$ to $\sim 1.1 \times 10^9$ points in total. Figure 4 displays the time cost per step as a function of the total number of finite elements. Except for the case of 64^3 elements, which shows superscaling, Fig. 4 clearly demonstrates linear scaling over grid sizes varying by four orders of magnitude. The superscaling at 64^3 may result from the coincidence of the topological decomposition with the physical allocation of 64 CPU's requested.

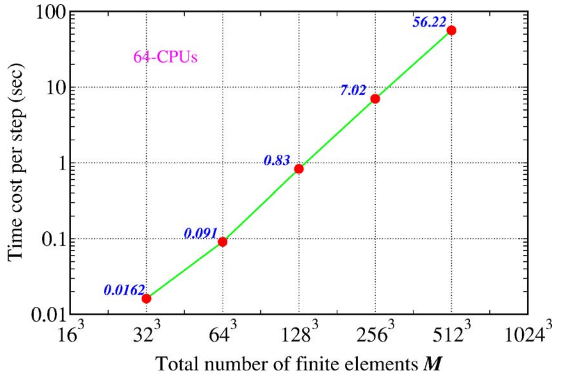

FIG. 4. (Color online) Tests of the RSP-FEDVR scaling on the grid size of the 3D BEC problem. It shows a perfect linear scaling over grid sizes varying by four orders of magnitude.

B. Demonstrations

1. 3D harmonic oscillator

As a first example, we consider the calculation of the ground-state (g.s.) energy and wave function of a three-dimensional spherical harmonic oscillator $[\omega_x=\omega_y=\omega_z]$. The potential in trap units has the form $V(x,y,z)=\frac{1}{2}(x^2+y^2+z^2)$ with an exact analytical g.s. energy of $1.5\hbar\omega_x$. We accomplish this by evolving a trial wave packet in imaginary time (t replaced by $-it$), which converts the Schrödinger into a diffusion equation. Propagating the trial wave packet for sufficiently large times relaxes the solution to the ground state of the Hamiltonian. The size of grid was taken as $[-10.0d_x,+10.0d_x]$ in each dimension. Twenty finite elements were employed, and the number of basis functions was varied from 3 to 8 per element. The results of the ground-state energy are shown in Table I. We find that even with four basis functions per element, we compute an energy of $E_0=1.499\,996\,307$. In this case, the total number of points is only 226 981 (61^3). When the number of basis functions within each element is doubled (meaning a much higher-order representation), we compute an energy of $E_0=1.500\,000\,001$, which is essentially exact. However, the number of points is still reasonably small, 141^3 points to discretize the 3D cube of size 20^3. We compare these results to those of the conventional three-point finite-difference (FD) method [21], for different mesh sizes, in Table I. From Table I, we see that it is possible to achieve higher-accuracy results by increasing the number of spatial points with the RSP-FD method. But the convergence is quite slow. Even with the largest grid size of $500\times500\times500$, we only obtain an energy of $E_0=1.499\,907\,103$, accurate to only the fifth digit. Using the RSP-FD approach, it was impossible, with current computational resources, to achieve the same accuracy as that of the RSP-FEDVR utilizing much smaller basis set sizes.

TABLE I. The comparison of calculated ground-state of a 3D harmonic oscillator, with the RSP-FD method and the RSP-FEDVR solver. Tests are done on the LANL's Beowulf cluster "Grendels," a 256-CPU dual xeon machine with Myrinet connection.

Method	No. elements (× basis)	Total grid points	E_0
RSP-FD	1	104^3	1.496524844
RSP-FD	1	144^3	1.498189365
RSP-FD	1	200^3	1.499061950
RSP-FD	1	300^3	1.499632781
RSP-FD	1	500^3	1.499907103
RSP-FEDVR	20(×3)	41^3	1.497422285
RSP-FEDVR	20(×4)	61^3	1.499996307
RSP-FEDVR	20(×6)	101^3	1.500000028
RSP-FEDVR	20(×8)	141^3	1.500000001
EXACT			1.500000000

TABLE II. The ground-state energy of a BEC in a 3D aspherical trap (case 2). For our RSP-FEDVR solver, on each dimension the 20 elements span a space of $[-9.5,10.5]$. Tests are done on the LANL's Beowulf cluster "Grendels," a 256-CPU dual xeon machine with Myrinet connection.

Method	No. elements (× basis)	Total grid points	E_0
RSP-FEDVR	20	41^3	19.85562355
RSP-FEDVR	20(×4)	61^3	19.84855573
RSP-FEDVR	20(×6)	101^3	19.84925147
RSP-FEDVR	20(×8)	141^3	19.84925687
3D Diag			19.847

2. BEC in an aspherical trap

We now turn to the case of the nonlinear Schrödinger equation and treat the GP equation describing a Bose-Einstein condensate in an aspherical harmonic trap (case 2). To find the ground state of a BEC in a trap, we apply the time-dependent method (RSP-FEDVR) to relax a trial wave packet in imaginary time. Although direct diagonalization of the 3D problem is possible, the presence of the nonlinear term $g|\Psi|^2$ in the Hamiltonian makes the method somewhat unwieldy. The time-dependent method is straightforward—the only complication being that the time steps required are often small for large nonlinearities. The spatial size used in each dimension was taken to be $[-9.5,10.5]$. Twenty equal-size elements were employed to discrete the space for each dimension; within each element, we varied the number of basis functions from 3 to 8. The results are shown in Table II. Reliable results were obtained with a small number of points, which is comparable to the result from direct diagonalization [36] marked "3D diag" in Table II.

3. Ground state of the helium atom

In Sec. IV, we have outlined the formalism for treating two electrons by employing the time-dependent close-coupling method. Expanding the solution in terms of coupled spherical harmonics, the six-dimensional problem is reduced to a set of coupled partial differential equations in the radial

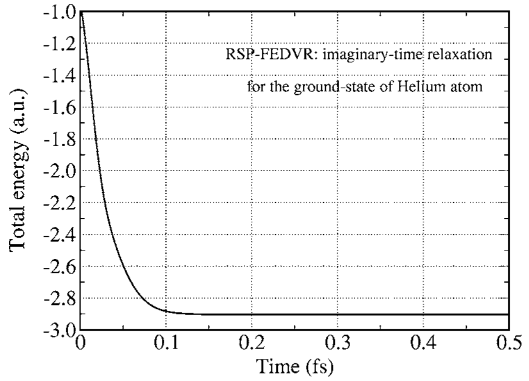

FIG. 5. The total energy versus time during the imaginary-time relaxation of a trial wave packet to the ground state of the helium atom.

variables r_1 and r_2. The FEDVR is then used to discretize r_1 and r_2 and to efficiently represent the two-electron radial wave function.

As an example, we used the RSP-FEDVR to relax a trial wave packet in imaginary time to find the ground state ($L=0$) of the He atom. We included six partial waves for each electron—i.e., $l_1=l_2=0-5$. The calculation utilized 160 finite elements to discretize each radial variable. The elements sizes were varied in a geometric progression of 2.5 with the smallest element size equal to 0.16 a.u. This implies a radial box size of 115.403 (bohrs=0.0529 nm). Four basis functions were used in each element, so that the total number of points is equal to 480 [note that the first point at $r_1(r_2)=0$ must be thrown away in order to impose the boundary condition]. With this grid, we obtain a ground-state energy of $E_0=-2.9031$ a.u., which is in excellent agreement with the most accurate value of -2.9037 a.u. [37]. The evolution of the total energy as a function of the damping time is plotted in Fig. 5. We require only 0.5 fs to converge the ground-state energy. Again, as in all of the other examples, such accuracy (only 0.021% deviation) cannot be achieved with a low-order FD method with such a small number of points. Finally, we plot the radial probability distribution $(\Sigma_i|\Psi_i(r_1,r_2)|^2)$ of the ground-state wave function of He at the end of the propagation process in Fig. 6. The figure shows that the symmetric $(1s^2)$ ground state of He has the peak electron probability around $r_1=r_2\simeq 0.93$ bohr, which is what quantum mechanics predicts.

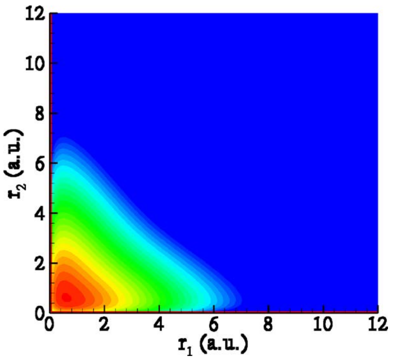

FIG. 6. (Color online) The ground-state probability of He on the (r_1,r_2) plane.

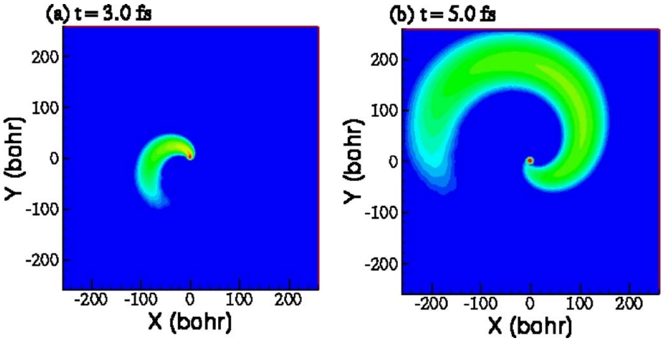

FIG. 7. (Color online) The snapshots of electron probability of a hydrogen atom, exposed to a CP-FCP, are shown for instants of (a) $t=3.0$ fs and (b) $t=5.0$ fs. The laser pulse has a duration of 5 fs, a wavelength of 800 nm, and an intensity of 2×10^{14} W/cm^2. The carrier-envelope phase is equal to 60°, and the laser field is left-hand circularly polarized.

4. Strong-field problem: Hydrogen atom exposed to a circularly polarized and intense few-cycle pulse

All of the examples above have employed the RSP-FEDVR for imaginary-time propagation. We conclude with an example that illustrates the efficacy of the RSP-FEDVR for real-time propagation. The computational demands required to solve a 3D problem, such as an atom driven by CP intense laser pulses, are substantial. The problem requires very large grids since the laser field can drive the initial electron wave packet to very large distances. This represents a case for which our RSP three-point finite-difference approach [10] simply proved intractable, given the requisite computational resources required.

Using the RSP-FEDVR code, we have investigated the dynamics of a hydrogen atom exposed to an intense circularly polarized few-cycle pulse (FCP). Starting from the ground state, we run the RSP-FEDVR code for an interaction time of 5 fs; we observed that the "left-hand" circularly polarized field is continuously ionizing the electron wave packet and rotating it clockwise. The snapshots of electron probability are shown in Fig. 7, for instants of (a) $t=3.0$ and (b) $t=5.0$ fs.

We used Cartesian coordinates, employing a maximum 600 elements in both the x and y dimensions where the FCP is circularly polarized. The z coordinate was divided into 80 elements. The wave function in each element was expanded using only four bases. The final number of grid points was $1801\times1801\times241\sim8\times10^8$ points. The calculation took almost 50 hours per run ($\sim$10 000 time steps) utilizing 100 nodes of the LANL supercomputer cluster "Flash." This demonstrates a new category of TDSE problem now tractable with this method.

VI. SUMMARY

We have developed a real-space-product, finite-element discrete-variable-representation approach [16] for solving the three-dimensional, multi-particle time-dependent Schrödinger equation. The approach employs a RSP technique for evolving the solution in time, which gives the optimal factorization of the exponential propagation operator for a particular order in the temporal variable. A partition of the spatial variables into finite elements provides an efficient means of spanning space with a flexible grid that yields a sparse representation. A DVR basis then determines the wave function within each element. The DVR gives an economical means of accurately calculating the matrix elements of the various operators on this basis. This occurs by drawing on the close association of the DVR basis functions with the classical orthogonal polynomials and with a particular Gauss quadrature. The integrated method supplies a powerful means of addressing very complicated applications. In addition, the form taken by the propagation operator readily accommodates current parallel-computer architectures and routinely demonstrates linear scaling.

We have applied the techniques to four representative problems that include (1) the 3D harmonic oscillator, (2) a Bose-Einstein condensate in a harmonic trap, (3) an atom ionized by a circularly polarized laser field, and (4) the two-electron helium atom. The method handles these cases efficiently and accurately with parallel scaling demonstrated on hundreds of processors on current distributed-memory architectures. The approach brings many outstanding 3D, multielectron problems within computational range.

ACKNOWLEDGMENTS

We gratefully acknowledge invaluable help from Hal Marshall from the Computing, Communications, and Networking (CCN) Division at the Los Alamos National Laboratory in aspects of the implementation of the parallel version of the method. The work was performed under the auspices of the U.S. Department of Energy through the Los Alamos National Laboratory.

[1] V. S. Melezhik, J. Comput. Phys. **92**, 67 (1991); V. S. Melezhik, and C.-Y. Hu, Phys. Rev. Lett. **90**, 083202 (2003).

[2] M. H. Beck, A. Jackle, G. A. Worth, and H.-D. Meyer, Phys. Rep. **324**, 1 (2000).

[3] R. Baer, Phys. Rev. A **62**, 063810 (2000).

[4] G. D. Billing, J. Chem. Phys. **114**, 6641 (2001).

[5] D. R. Schultz and P. S. Krstic, Phys. Rev. A **67**, 022712 (2003).

[6] J. Colgan, M. S. Pindzola, F. J. Robicheaux, D. C. Griffin, and M. Baertschy, Phys. Rev. A **65**, 042721 (2002).

[7] S. Chelkowski, C. Foisy, and A. D. Bandrauk, Phys. Rev. A **57**, 1176 (1998); A. D. Bandrauk and H. Z. Lu, *ibid.* **72**, 023408 (2005).

[8] M. S. Pindzola and F. Robicheaux, Phys. Rev. A **57**, 318 (1998); J. Colgan, M. S. Pindzola, and F. Robicheaux, *ibid.* **72**, 022727 (2005).

[9] K. C. Kulander, K. J. Schafer, and J. L. Krause, in *Atoms in Intense Radiation Fields*, edited by M. Gravila (Academic Press, New York, 1992); T. O. Clatterbuck, C. Lynga, P. M. Paul, L. F. Di Mauro, M. B. Gaarde, K. J. Schafer, P. Agostini, K. C. Kulander, and I. Walmsley, Phys. Rev. A **69**, 033807 (2004).

[10] L. A. Collins, J. D. Kress, and R. B. Walker, Comput. Phys. Commun. **114**, 15 (1998).

[11] S. X. Hu and L. A. Collins, Phys. Rev. A **69**, 033405 (2004); Phys. Rev. Lett. **94**, 073004 (2005).

[12] M. L. Chiofalo, S. Succi, and M. P. Tosi, Phys. Rev. E **62**, 7438 (2000); S. Succi, F. Toschi, M. P. Tosi, and P. Vignolo, Comput. Sci. Eng. **7**, 48 (2005).

[13] S. K. Adhikari, Phys. Rev. E **65**, 016703 (2000); Phys. Rev. A **66**, 043601 (2002).

[14] J. Denschlag *et al.*, Science **287**, 97 (2000).

[15] C. LeForestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karlein, H. D. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, J. Comput. Phys. **94**, 59 (1991).

[16] B. I. Schneider and L. A. Collins, J. Non-Cryst. Solids **351**, 1551 (2005).

[17] P. Muruganandam and S. K. Adhikari, J. Phys. B **36**, 2501 (2003).

[18] J. Ackermann and J. Shertzer, Phys. Rev. A **54**, 365 (1996).

[19] T. N. Rescigno and C. W. McCurdy, Phys. Rev. A **62**, 032706 (2000).

[20] C. Moler and C. Van Loan, SIAM Rev. **45**, 3 (2003).

[21] H. DeRaedt, Comput. Phys. Rep. **7**, 1 (1987).

[22] J. C. Light, I. P. Hamilton, and J. V. Lill, J. Chem. Phys. **82**, 1400 (1985).

[23] B. I. Schneider and N. Nygaard, J. Phys. Chem. A **106**, 10773 (2002).

[24] T. J. Park and J. C. Light, J. Chem. Phys. **85**, 5870 (1986).

[25] A. Castro, M. A. L. Marques, and A. Rubio, J. Chem. Phys. **121**, 3425 (2004).

[26] H. F. Trotter, Proc. Am. Math. Soc. **10**, 545 (1959).

[27] M. D. Feit and J. A. Fleck, J. Chem. Phys. **78**, 301 (1982).

[28] M. Suzuki, J. Math. Phys. **26**, 601 (1985).

[29] M. Suzuki, J. Math. Phys. **32**, 400 (1991).

[30] M. Suzuki and T. Yamauchi, J. Math. Phys. **34**, 4892 (1992).

[31] S. K. Adhikari and P. Muruganandam, J. Phys. B **35**, 2831 (2002).

[32] S. A. Chin and E. Krotscheck, Phys. Rev. E **72**, 036705 (2005).

[33] H. Q. Wang, Appl. Math. Comput. **170**, 17 (2005).

[34] W. Z. Bao and J. Shen, SIAM J. Sci. Comput. (USA) **26**, 2010 (2005).

[35] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI—Portable Parallel Programming with the Message-Passing Interface*, 2nd ed. (MIT Press, Cambridge, MA, 1999).

[36] B. I. Schneider and D. L. Feder, Phys. Rev. A **59**, 2232 (1999).

[37] J. S. Sims and S. A. Hagstrom, Int. J. Quantum Chem. **90**, 1600 (2002).