Commun. Comput. Phys.
doi: 10.4208/cicp.101214.021015a

Vol. **20**, No. 1, pp. 60-85
July 2016

A Time-Space Adaptive Method for the Schrödinger Equation

Katharina Kormann[1,*]

[1] *Technische Universität München, Zentrum Mathematik, Boltzmannstr. 3, 85747 Garching, Germany.*

Received 10 December 2014; Accepted (in revised version) 2 October 2015

Abstract. In this paper, we present a discretization of the time-dependent Schrödinger equation based on a Magnus-Lanczos time integrator and high-order Gauss-Lobatto finite elements in space. A truncated Galerkin orthogonality is used to obtain duality-based a posteriori error estimates that address the temporal and the spatial error separately. Based on this theory, a space-time adaptive solver for the Schrödinger equation is devised. An efficient matrix-free implementation of the differential operator, suited for spectral elements, is used to enable computations for realistic configurations. We demonstrate the performance of the algorithm for the example of matter-field interaction.

AMS subject classifications: 65M50, 65M60, 65Z05, 81-08

Key words: a posteriori error estimate, adaptive mesh refinement, finite/spectral elements, time-dependent Schrödinger equation

1 Introduction

With the emergence of femtosecond laser pulses and the seminal experiments by Zewail [58], it became possible to experimentally study molecular processes and chemical reactions. The time-dependent Schrödinger equation (TDSE) gives a theoretical description of the dynamics within molecules,

$$\begin{aligned} &\mathrm{i}\hbar\frac{\partial}{\partial t}\psi(x,t)=\hat{H}(t)\psi(x,t),\quad x\in\mathbb{R}^d,\quad 0<t<t_f<\infty,\\ &\psi(x,0)=\psi_0. \end{aligned} \tag{1.1}$$

The differential operator $\hat{H}(t)=\hat{T}(\frac{\partial^2}{\partial x_1^2},\cdots,\frac{\partial^2}{\partial x_d^2})+\hat{V}(x,t)$ comprises the kinetic ($\hat{T}$) and the potential ($\hat{V}$) energy of the system. Since close analytical expressions for the wave function are only known for very simple configurations, numerical simulations are essential

*Corresponding author. *Email address:* katharina.kormann@tum.de (K. Kormann)

 ©2016 Global-Science Press

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

for the understanding of the outcome of experimental studies. The major challenge in simulations is the curse of dimensionality: The spatial variables in the model describe the degrees of freedom in the molecule, i.e., the coordinates of each particle, and the dimension of the underlying partial differential equation grows therefore linearly with the number of particles.

Various semiclassical methods have been developed that combat the course of dimensionality by employing some type of classical modeling (see e.g. [10, 20, 30, 45, 46, 57]). These techniques are very powerful but they have difficulties in covering the full quantum dynamical nature of the underlying system like energy transfer via laser fields or tunneling effects. The goal of this work is to solve the TDSE for as large systems as possible without any (semi-)classical approximation. Our focus is on the simulation of the interaction of molecules with time-dependent fields.

For small molecules, it is very common to use pseudospectral methods [12, 19, 22, 40, 56] to solve the full TDSE. Because the solutions are smooth, high accuracy can be obtained with a relatively small number of grid points. However, the method is generalized to higher dimensions by tensor products and a flexible adaption of the mesh to the solution is not possible. An approach for improving the efficiency of the Fourier method is the so-called mapped Fourier method where the coordinates are transformed according to the properties of the Hamiltonian [21, 35]. Also, a sparse grid algorithm for the Fourier method was devised by Hallatschek [28] and later employed for the Schrödinger equation [24, 25]. Still these improvements do not enable solution- or residual-based mesh adaption. Moreover, the one-dimensional stencil couples all the grid points and there is no locality in the method which makes parallelization rather difficult.

On the other hand, much research has been done on error estimation and mesh adaptation for finite element discretizations. A posteriori error theory, based on a global energy norm, was initiated by Babuška and Rheinboldt [3, 4]. The work by Babuška and Miller [2] introduced duality arguments. Later Becker and Rannacher [8, 9] designed the dual-weighted residual (DWR) method for error control. This method allows to design adaptive meshes for more general functionals of the error and in this way the error in some physically meaningful function of the solution can be controlled. For the TDSE, it is common that one is interested in the value of some physical observable that can be expressed as a functional of the solution rather than the full distribution function. Duality-based adaptivity is a means to concentrate the computational effort to regions that are most relevant for simulating this observable instead of on minimizing the global L_2 error. Such an observable could be the crosscorrelation, i.e., the overlap of the solution with a predefined state. While DWR is well-established for elliptic differential equations, the temporal dimension poses additional difficulties. Therefore, most work on a posteriori error estimation for parabolic problems do not directly rely on the solution of the dual problem (see e.g. [18, 48]). Schmich and Vexler [52] have extended the DWR theory to parabolic problems in a setting with a discontinuous Galerkin formulation in time. They separate temporal and spatial error estimation and treat the temporal dimension in a time-stepping approach. Still the fact that the primal and the dual problem are solved

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

in different directions requires storing a large amount of data. Bermejo and Carpio [11] therefore propose an adaptive algorithm that uses the duality argument only locally per time step. However, they cannot show any global error bounds. Duality-based a posteriori error estimates have also been discussed for time stepping in ordinary differential equations (ODE) [14] and applied to the semidiscretized Schrödinger equation in [37] to achieve an adaptive time-stepping method. In this paper, we use the temporal adaptivity proposed in [37] and enhance it by introducing spatial adaptivity that is optimized for the computation of some given functional of the error.

Norm-based error estimation has been studied for the Schrödinger equation in various settings. Karakashian and Makridakis [33] have considered a space-time finite element discretization of the cubic Schrödinger equation and Kyza [42] the spatially continuous Crank-Nicolson approximation to the linear equation. Dörfler [16] considers a spatial finite element discretization combined with an implicit propagation scheme and also designs an adaptive algorithm around the estimate.

The focus of this paper is to design an efficient and robust method for the discretization of the Schrödinger equation. Aside from error estimation that we obtain unlike the aforementioned papers based on a duality argument, we also focus on an efficient and parallel implementation. As in [52], we split temporal and spatial error estimation but rely on a Magnus-Lanczos propagator to discretize in time. This propagator has excellent stability properties [43] while being explicit and is therefore very efficient for the Schrödinger equation, especially when the Hamiltonian is time dependent. This has been found in the comparative study of the propagator in [36]. When working with explicit time propagators, mass matrices that appear in a standard finite element formulation are bothersome. To minimize the additional effort related to the mass matrix, we will work with Gauss-Lobatto elements, which allow for high-quality representations with diagonal mass matrices [15]. Another advantage of these elements, also referred to as spectral elements, is that they cluster the nodes towards the element boundary and are therefore better conditioned than standard finite elements for increasing element order (see [34]).

In a standard implementation based on sparse matrices, however, finite element stencils are rather wide compared to other localized methods like finite differences. Also, the memory consumption of the matrices is prohibitive. We therefore use a matrix-free implementation of the matrix-vector product that reuses unit-cell shape function information and makes efficient use of the tensor-product formulation of the stencil (cf. [38, 41]). In a matrix-free implementation, the operations per node for Gauss-Lobatto elements have a similar arithmetic complexity as finite differences according to the summation-by-parts principle [26] for high orders. Hence, high-order Gauss-Lobatto elements represent a good compromise of accuracy and flexibility.

The outline of the paper is the following. The next section discusses the used discretization techniques. In Section 3, we derive the a posteriori estimate for our numerical method. Based on the estimate, we devise an adaptive algorithm in Section 4. This is followed by a discussion of some implementation aspects in Section 5. Results are presented in Section 6 and Section 7 concludes this work.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

2 Approximation of the Schrödinger equation

A weak solution $\psi \in L_2(H^1(\mathbb{R}^d);(0,t_f])$ with temporal derivative

$$\frac{\partial}{\partial t}\psi \in L_2(H^{-1}(\mathbb{R}^d),(0,t_f])$$

satisfies $\psi(x,0)=\psi_0$ and

$$\int_{\mathbb{R}^d}\left(\varphi^*\frac{\partial}{\partial t}\psi+\frac{\mathrm{i}}{\hbar}(\nabla\varphi)^* a(x)\nabla\psi+\frac{\mathrm{i}}{\hbar}\varphi^*\hat{V}\psi\right)\mathrm{d}x=0,$$

for all $\varphi\in H^1(\mathbb{R}^d)$ and $t\in(0,t_f]$. Here, we have used the form $\hat{T}=-\nabla a(x)\nabla$ of the kinetic energy operator. The coefficient $a(x)$ contains the mass fraction $\frac{\hbar^2}{2m_i}$ for each dimension and possibly a (real-valued) variable coefficient originating from coordinate transformation. We consider atomic units, i.e. $\hbar$ and m_i are set to one.

In its most general formulation, the Schrödinger equation is formulated on the whole $\mathbb{R}^d$. The coordinates are usually transformed by exploiting symmetries and there are often angular dimensions with periodic boundary conditions and distances that can take values between zero and infinity. This is not a practical domain for computations whence artificial boundaries need to be included. For bound states, the eigenfunctions of the Hamiltonian decay very fast towards infinity [23] and we can thus assume that the wave packet stays within certain limits and restrict the simulation to a certain interval assuming homogenous Dirichlet or periodic boundary conditions. This is common practise in quantum dynamics simulations. If dissociative states are involved, the wave packet may propagate towards infinity when a bound within the molecule breaks. In this case it is common to truncate the domain and to include some damping layer [1]. Within the layer, there is either a complex potential or a complex coefficient a. In the following, we assume that we have chosen a proper computational domain Ω and boundary conditions where the error due to domain truncation can be neglected. Hence, we consider $\psi\in V$ to be the solution to the problem posed on some finite domain Ω, where $V\subset H^1(\Omega)$ is the space of functions satisfying periodic or homogeneous boundary conditions. The derived estimates do not include the error done when reformulating the problem on the computational domain which is considered to be part of the model.

In our theoretical derivation of the a posteriori error estimate in Section 3, damping layers are allowed. However, the case of bound states is simpler since the Hamiltonian is real and symmetric and the Schrödinger equation is self-adjoint. Hence, the norm of the wave function is conserved and the equation is time-reversible. In the design of our adaptive algorithm, we will exploit these properties to simplify the computations. In the case of a dissociative state modeled by an absorbing layer, these properties are lost. However, we still know that the norm is decreasing for both the primal and the dual problem (cf. [37, Sec. 2.2]). A combination of a temporal error control with perfectly matched layers can be found in [39]. To include absorbing boundaries in our adaptive finite element

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

solver, damping layers for our adaptive Gauss-Lobatto finite elements would also be necessary. Artificial boundary conditions for the solution of the TDSE in three dimension for a finite-element discretization has for instance been considered in [29].

In the formulation of a posteriori error estimates, the adjoint or dual partial differential equation (PDE),

$$\begin{aligned} &\mathrm{i}\hbar\frac{\partial}{\partial t}\chi(x,t)=\widehat{H}^*\chi(x,t), \quad x\in\mathbb{R}^d,\ 0\le t<t_f<\infty,\\ &\chi(x,t_f)=\phi(x), \end{aligned} \tag{2.1}$$

plays an important role. Here, $\phi\in V$ with $\|\phi\|_{L_2}=1$ is some given state at final time t_f. The (normalized) final state can be chosen arbitrarily and defines the physical quantity of interest for which the error estimate should be formulated and the adaptivity should be optimized.

2.1 Finite element discretization

The finite element approximation $\psi^{\mathcal{T},p}$ of ψ in a finite dimensional subspace $V^{\mathcal{T},p}\subset V$ solves

$$\int_\Omega\left(\varphi^*\frac{\partial}{\partial t}\psi^{\mathcal{T},p}+\frac{\mathrm{i}}{\hbar}\varphi^*\widehat{H}\psi^{\mathcal{T},p}\right)\mathrm{d}x=0, \quad \text{for all } \varphi \text{ in } V^{\mathcal{T},p},\ t\in(0,t_f]. \tag{2.2}$$

The subspace $V^{\mathcal{T},p}\subset V$ is determined by the tessellation $\mathcal{T}$, the element order p, and the type of elements (see Section 2.2).

Let N be the dimension of the subspace $V^{\mathcal{T},p}$ and let $\{\varphi_1,\cdots,\varphi_N\}$ be the finite element basis. The approximate solution can be written as a weighted sum of basis functions, $\psi^{\mathcal{T},p}=\sum_{i=1}^N v_i^{\mathcal{T},p}\varphi_i$. Eq. (2.2) can then be solved for $\psi^{\mathcal{T},p}$ by considering the following matrix equation,

$$M\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}^{\mathcal{T},p}=S\mathbf{v}^{\mathcal{T},p},$$

with stiffness matrix element $S_{i,j}=-\frac{\mathrm{i}}{\hbar}\int_\Omega\left(\nabla\varphi_i^*a\nabla\varphi_j+\varphi_i^*V\varphi_j\right)\mathrm{d}x$ and mass matrix elements $M_{i,j}=\int_\Omega\varphi_i^*\varphi_j\mathrm{d}x$.

Note that the L_2 norm of two functions $\psi_k^{\mathcal{T},p}=\sum_{i=1}^N v_{ki}^{\mathcal{T},p}\varphi_i\in\hat{V}^{\mathcal{T},p}$, $k=1,2$, corresponds to the M norm of the coefficient vectors, i.e.,

$$\int_\Omega\left(\psi_1^{\mathcal{T},p}\right)^*\psi_2^{\mathcal{T},p}\mathrm{d}x=\left(\mathbf{v}_1^{\mathcal{T},p}\right)^*M\mathbf{v}_2^{\mathcal{T},p}, \quad \forall\psi_1^{\mathcal{T},p},\psi_2^{\mathcal{T},p}\in V^{\mathcal{T},p}.$$

2.2 Mass-lumped finite elements

We use an explicit solver for the time-propagation in order to avoid solving linear equation systems based on the discrete Hamiltonian in each time step (cf. Section 3.2). However, in the standard finite element formulation the presence of the mass matrix M that

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

has off-diagonal entries makes it still necessary to solve a linear system. Therefore, we construct our finite elements as Lagrange polynomials on the nodes of Gauss-Lobatto quadrature and approximate the integrals in mass and stiffness matrix by Gauss-Lobatto quadrature in these nodes. This way, the mass matrix becomes diagonal and can easily be inverted (cf. [15]). In our experiments, we use $\mathcal{Q}_4$, $\mathcal{Q}_5$, and $\mathcal{Q}_6$ elements and show that the use of high-order pays off in our application. This was expected since it is well-known that high-order methods are efficient for the TDSE. These kind of finite elements have also been applied to quantum scattering problems by Manolopoulos and Wyatt [44] as well as Rescigno and McCurdy [49].

For this discretization, however, mass and stiffness matrix are not computed exactly and the approximation order of the Gauss-Lobatto formula on the support points is $2p$. Nevertheless, the convergence order is not affected by the use of approximate quadrature. This can be shown using the reasoning from Durufle, Grob, and Joly [17] for Cartesian meshes, which we use in our method. Let us denote the truncated (diagonal) mass matrix by $\tilde{M}$ and the truncated stiffness matrix by $\tilde{S}$ and let $\mathbf{a},\mathbf{b}$ be two vectors containing the weights of the FE approximant of a smooth function. Then, it holds that $\mathbf{a}^*\widetilde{M}\mathbf{b}=\mathbf{a}^*M\mathbf{b}+\mathcal{O}(h^{2p})$ and $\mathbf{a}^*\widetilde{S}\mathbf{b}=\mathbf{a}^*S\mathbf{b}+\mathcal{O}(h^{2p-2})$, where $h=\max_{K\in\mathcal{T}}h_K$ and h_K is the diameter of cell K. Note that we assume that there is a bound on the anisotropy of the cells. These estimates can be shown by a modification of Proposition 3.2 in Durufle et al. Considering that the error for a pth order FE approximation is of order h_K^{p+1}, the error in the FE matrices will only perturb the leading order term of our estimate for element order three and less. Therefore, we will assume approximations with $p>3$ in the following and dismiss the error term due to numerical integration for our error estimate which is a first order estimate.

After semi-discretization in space, we have to propagate the following ODE system in time,

$$\widetilde{M}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{v}}^{\mathcal{T},p}=\widetilde{S}\tilde{\mathbf{v}}^{\mathcal{T},p}, \tag{2.3}$$

where we denote by $\tilde{\psi}^{\mathcal{T},p}=\sum_{i=1}^{N}\tilde{v}_i^{\mathcal{T},p}\varphi_i$ the solution that we get when approximating the integrals. When computing an a posteriori error estimate, we have to keep in mind that the Galerkin orthogonality does not hold for the lumped FE matrices (see Section 3.1 below).

2.3 Magnus-Lanczos/Arnoldi propagator

For a time-independent Hamiltonian operator the evolution operator $U(t_f,t_0)$ of system (2.3) can be expressed as $\exp((t_f-t_0)A)$ with $A=\widetilde{M}^{-1}\widetilde{S}$. Since the matrix-matrix product does not commute, the evolution operator is not as simple when $A=A(t)$. However, the Magnus expansion [13] provides a series expansion on sufficiently small time intervals

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

$[t_n,t_{n+1}]$ of the form

$$\begin{aligned} &U(t_{n+1},t_n)=\exp(\Omega(t_n,t_{n+1})), \quad \text{with} \\ &\Omega(t_n,t_{n+1})=\int_{t_n}^{t_{n+1}} A(t)\,\mathrm{d}t-\frac{1}{2}\int_{t_n}^{t_{n+1}}\left[\int_t^{t_{n+1}} A(s)\,\mathrm{d}s,A(t)\right]\mathrm{d}t+\cdots, \end{aligned} \tag{2.4}$$

where $[B,C]=BC-CB$ is the commutator of two matrices B and C. In our simulations, we use a second-order Magnus propagator which only includes the first term in (2.4). The second term will be used for error estimation (cf. Section 3.2). The reason for choosing a low-accuracy approximation is the fact that the terms in the Magnus expansion contain an increasing number of commutators that are expensive to evaluate. All appearing integrals are numerically evaluated using two-point Gauss quadrature. Note that a midpoint rule would suffice to obtain a second order propagator. However, in our adaptive algorithm we want the quadrature error to be negligible compared to the truncation error in the Magnus series.

For a discretization of the TDSE based on a method-of-lines approach, evaluating (2.4) means computing a matrix exponential. Since a direct computation would be too expensive, we compute an approximation of $U(t_{n+1},t_n)\psi(t_n)$ using a Krylov subspace method. For a symmetric Hamiltonian and a symmetric discretization, the (more efficient) Lanczos algorithm [47] can be used, otherwise the Arnoldi method. Note that our operator matrix $\widetilde{M}^{-1}\widetilde{S}$ is not symmetric. However, we formulate the Lanczos algorithm for the generalized eigenvalue problem $\lambda\widetilde{M}v=\widetilde{S}v$ based on the $\widetilde{M}$-norm. This is also the natural norm for our problem, i.e., the norm in which we can formulate the conservation properties. Since $\widetilde{M}$ is diagonal, the overhead for the norm computations is not severe.

Due to the truncation of the Magnus expansion and to the fact that the dimension of the Krylov subspace is much smaller than the dimension of $\tilde{\mathbf{v}}^{\mathcal{T},p}$, we actually solve a perturbed version of Eq. (2.3),

$$\widetilde{M}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}^{\mathcal{T},p}(t)=\widetilde{S}\mathbf{u}^{\mathcal{T},p}(t)+\widetilde{M}\mathbf{R}_T(t), \tag{2.5}$$

where $\mathbf{R}_T(t)$ denotes the perturbation due to numerical time-propagation (see Section 3.2 for a more detailed discussion). Let us assume that the discrete values are computed by the Magnus-Arnoldi/Lanczos propagator on the temporal grid $0=t_0,\cdots,t_{N_t}=t_f$, and denote by $\mathbf{u}^{\mathcal{T},p}(t_n)$ the fully discrete solution. Finding an explicit expression for $\mathbf{R}_T(t)$ is complicated in general. However, we will simplify the estimate in Section 3.2 using the linearity of the TDSE.

3 A posteriori error estimate

In the first part of this section, we will derive an a posteriori estimate for a functional $\langle\phi,\cdot\rangle$ of the error in the propagation of the Schrödinger equation (1.1) with a spatial finite

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

element approximation and a Magnus-Lanczos/Arnoldi time discretization. Thereafter, we will review a technique proposed in [37] to estimate the temporal error only. This will be needed for the adaptive algorithm discussed in the subsequent section.

3.1 Estimate for the complete discretization

An approximation with standard finite elements satisfies the weak form of the TDSE, that is, we have Galerkin orthogonality. The solution $\psi^{\mathcal{T},p}$ therefore satisfies Eq. (2.2). In our case, we do not compute the exact Galerkin projection since we approximate the integrals according to Section 2.2. Nevertheless, we can show that the weak form of the TDSE is satisfied up to a certain order.

Lemma 3.1. *Let $\Psi=\sum_{i=1}^N u_i^{\mathcal{T},p}(t)\varphi_i(x)$, where the weights $\mathbf{u}^{\mathcal{T},p}$ satisfy the perturbed TDSE (2.5). Then, the Galerkin orthogonality holds up to order $2p-2$, i.e., for all $\varphi\in V^{\mathcal{T},p}$ we have*

$$\int_\Omega \varphi^*\frac{\partial}{\partial t}\Psi\,\mathrm{d}x=-\int_\Omega \frac{\mathrm{i}}{\hbar}\nabla\varphi^* a\nabla\Psi\,\mathrm{d}x-\int_\Omega\frac{\mathrm{i}}{\hbar}\varphi^*\hat{V}\Psi\,\mathrm{d}x+\int_\Omega\varphi^*\rho_T\,\mathrm{d}x+\mathcal{O}(h^{2p-2}),$$

where $\rho_T(x,t)=\sum_{i=1}^N R_{T,i}(t)\varphi_i(x)$ with $R_{T,i}$ being the ith component of $\mathbf{R}_T$, and h is the maximum diameter of all cells.

Proof. Since $\varphi\in V^{\mathcal{T},p}$, we can represent it as a linear combination of the basis vectors, i.e., $\varphi=\sum_{i=1}^N w_i\varphi_i$. Exploiting this, we find

$$\int_\Omega\varphi^*\frac{\partial}{\partial t}\Psi\,\mathrm{d}x=\sum_{i,j}\int_\Omega w_i\varphi_i^*\frac{\mathrm{d}}{\mathrm{d}t}u_j^{\mathcal{T},p}\varphi_j\,\mathrm{d}x=\mathbf{w}^*M\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}^{\mathcal{T},p}.$$

As a next step, we want to use that $\mathbf{u}^{\mathcal{T},p}$ solves the truncated ODE (2.5). We further use that $(\mathbf{w})^*M\mathbf{u}^{\mathcal{T},p}=(\mathbf{w})^*\widetilde{M}\mathbf{u}^{\mathcal{T},p}+\mathcal{O}(h^{2p})$ and $(\mathbf{w})^*S\mathbf{u}^{\mathcal{T},p}=(\mathbf{w})^*\widetilde{S}\mathbf{u}^{\mathcal{T},p}+\mathcal{O}(h^{2p-2})$,

$$\begin{aligned}\int_\Omega\varphi^*\frac{\partial}{\partial t}\Psi\,\mathrm{d}x&=\mathbf{w}^*\widetilde{M}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}^{\mathcal{T},p}+\mathcal{O}(h^{2p})=\mathbf{w}^*\widetilde{S}\mathbf{u}^{\mathcal{T},p}+\mathbf{w}^*\widetilde{M}\mathbf{R_T}+\mathcal{O}(h^{2p})\\&=\mathbf{w}^*S\mathbf{u}^{\mathcal{T},p}+\mathbf{w}^*M\mathbf{R_T}+\mathcal{O}(h^{2p-2}).\end{aligned}$$

Finally, we reshape the formulation into functions again yielding

$$\int_\Omega\varphi^*\frac{\partial}{\partial t}\Psi\,\mathrm{d}x=-\int_\Omega\frac{\mathrm{i}}{\hbar}\nabla\varphi^*a\nabla\Psi\,\mathrm{d}x-\int_\Omega\frac{\mathrm{i}}{\hbar}\varphi^*\hat{V}\Psi\,\mathrm{d}x+\int_\Omega\varphi^*\rho_T\,\mathrm{d}x+\mathcal{O}(h^{2p-2}).$$

The proof is completed. □

With Lemma 3.1 we have the necessary relation at hand to prove the a posteriori error estimate for the discretized TDSE. By the fundamental theorem of calculus it holds that

$$\langle\phi,e\rangle=\int_0^{t_f}\left(\langle\frac{\partial}{\partial t}\chi,e\rangle+\langle\chi,\frac{\partial}{\partial t}e\rangle\right)\mathrm{d}t,\tag{3.1}$$

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

where χ is the solution of the adjoint Schrödinger equation (2.1). Let us split the dual solution into its projection $\chi_\parallel$ onto the finite element space and the part $\chi_\perp$ within the orthogonal complement. For the orthogonal part $\chi_\perp$ it holds

Theorem 3.1. *Define the error $e(x)=\psi(x,t_f)-\Psi(x,t_f)$ at final time t_f. Then, it holds for $\phi\in L_2(\Omega)$,*

$$\begin{aligned}\langle\phi,e\rangle=-\Bigg[&\sum_{K\in\mathcal{T}}\int_0^{t_f}\int_K\chi_\perp^*\left(\frac{\mathrm{i}}{\hbar}\hat{H}\Psi\right)\mathrm{d}x\mathrm{d}t\\&+\sum_{K\in\mathcal{T}}\int_0^{t_f}\frac{\mathrm{i}}{\hbar}\langle a\chi_\perp,[\![\nabla\Psi\cdot\nu]\!]\rangle_{\partial K\setminus\partial\Omega}\mathrm{d}t\mathrm{d}x+\int_0^{t_f}\int_\Omega\chi^*(t)\rho_T(t)\mathrm{d}x\mathrm{d}t\Bigg]\\&+\mathcal{O}(h^{2p-2}).\end{aligned}\tag{3.2}$$

By $\langle\cdot,\cdot\rangle$ we have denoted the $L_2(\Omega)$ inner product and by $[\![\cdot]\!]$ the jump over the face ∂K.

Proof. Firstly, we rewrite the first term in (3.1) by substituting the adjoint Schrödinger equation for $\frac{\partial\chi}{\partial t}$. As a second step, we apply integration by parts to find the weak formulation,

$$\begin{aligned}\langle\phi,e\rangle&=\int_0^{t_f}\left(\langle-\frac{\mathrm{i}}{\hbar}\hat{H}\chi,e\rangle+\langle\chi,\frac{\partial}{\partial t}e\rangle\right)\mathrm{d}t\\&=\int_0^{t_f}\int_\Omega\left(\frac{\mathrm{i}}{\hbar}\left(\nabla\chi^*a\nabla e+\chi^*\hat{V}e\right)+\chi^*\frac{\partial}{\partial t}e\right)\mathrm{d}x\,\mathrm{d}t.\end{aligned}$$

We use the definition of $e=\psi-\Psi$. Since the exact solution satisfies the weak Schrödinger equation, it vanishes in the estimate. In order to be able to apply the approximate Galerkin orthogonality for the numerical solution, we split the dual solution into $\chi_\parallel$ and $\chi_\perp$. Using Lemma 3.1, the contribution of $\chi_\parallel$ vanishes up to order h^{2p-2} except for the perturbation due to the temporal discretization, ρ_T. Hence, we have

$$\begin{aligned}\langle\phi,e\rangle=&-\int_0^{t_f}\int_\Omega\left(\frac{\mathrm{i}}{\hbar}\left(\nabla\chi_\perp^*a\nabla\Psi+\chi_\perp^*\hat{V}\Psi\right)+\chi_\perp^*\frac{\partial}{\partial t}\Psi\right)\mathrm{d}x\,\mathrm{d}t\\&-\int_0^{t_f}\int_\Omega\chi_\parallel^*\rho_T\mathrm{d}x\mathrm{d}t+\mathcal{O}(h^{2p-2}).\end{aligned}$$

Since Ψ and $\frac{\partial}{\partial t}\Psi$ are both in $V^{\mathcal{T},p}$, the term $\int_0^{t_f}\int_\Omega\chi_\perp^*\frac{\partial}{\partial t}\Psi\mathrm{d}x\mathrm{d}t$ vanishes. We eventually apply the full Hamiltonian to Ψ. When using integration by parts once more, we have to take care at the cell boundaries where the gradient of the FE solution is discontinuous. Therefore, we have to consider each cell separately, yielding

$$\begin{aligned}\int_0^{t_f}\int_\Omega\frac{\mathrm{i}}{\hbar}\left(\nabla\chi_\perp^*a\nabla\Psi+\chi_\perp^*\hat{V}\Psi\right)\mathrm{d}x\mathrm{d}t=&\sum_{K\in\mathcal{T}}\int_K\int_0^{t_f}\chi_\perp^*\left(\frac{\mathrm{i}}{\hbar}\hat{H}\Psi\right)\mathrm{d}x\mathrm{d}t\\&+\sum_{K\in\mathcal{T}}\int_0^{t_f}\frac{\mathrm{i}}{\hbar}\langle a\chi_\perp,[\![\nabla u\cdot\nu]\!]\rangle_{\partial K\setminus\partial\Omega}\mathrm{d}t.\end{aligned}$$

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

By finally taking into account that $\rho_T \in V^{\mathcal{T},p}$, we can replace $\int_\Omega \chi_\parallel^* \rho_T \,\mathrm{d}x$ by $\int_\Omega \chi^* \rho_T \,\mathrm{d}x$. □

3.2 Estimate for the temporal error

While the local error of numerical integrators can generally be approximated, a closed expression for the global error at a certain time is more difficult to obtain. In order to compute the temporal part of the error, i.e. the term $\int_0^{t_f} \int_\Omega \chi^*(t)\rho_T(t)\,\mathrm{d}t$, we therefore use the idea of [37] to apply a posteriori error estimation theory separately on the individual time steps: The estimated error from the previous time step is treated as a perturbation of the starting value and the perturbation due to numerical propagation is given by the *local* remainder in each step. For the Magnus-Lanczos propagator, the first dismissed term of the Magnus expansion is used to estimate the Magnus error and for the Lanczos error we use the remainder term as proposed in [32, 51]. Since the temporal error estimation is formulated for individual time steps, we can also combine the temporal error estimation with dynamic mesh refinement where the system of ordinary differential equations resulting from semidiscretization changes after a number of time steps.

If we are only interested in estimating the time error, explicit computations of the dual problem can be avoided by using

$$\left| \int_\Omega \int_0^{t_f} \chi^*(T) r(t) \,\mathrm{d}t \,\mathrm{d}x \right| \leq \int_0^{t_f} \|\chi(t)\| \, \|r(t)\| \,\mathrm{d}t.$$

At this point, we can use that the norm is conserved (for symmetric Hamiltonians) or decreasing (for non-reflection boundary conditions) to remove the dual solution from the estimate. This means the local perturbations simply add up to the global error. This idea has been tested in [37] and will be employed in the adaptive algorithm proposed in Section 4.

4 Adaptive solution of the TDSE

Theorem 3.1 suggests a means to control the global error with the help of the residuals on cells and time intervals. Since we split time and space discretization, we are interested in measuring the errors due to time propagation and spatial meshing separately. The estimate (3.2) suggests that the term

$$\Xi_T := -\int_0^{t_f} \int_\Omega \chi^*(t)\rho_T(t) \,\mathrm{d}x \,\mathrm{d}t$$

describes an error due to temporal approximation in our method-of-lines derivation. If we replace the fully discrete solution Ψ by the (temporally) semidiscretized solution $\psi^{\mathcal{T},p}$ in the remaining terms,

$$\Xi_S := \sum_{K\in\mathcal{T}} \int_0^{t_f} \int_K \chi_\perp^* \left(\frac{\mathrm{i}}{\hbar} \hat{H} \Psi \right) \mathrm{d}x \,\mathrm{d}t + \sum_{K\in\mathcal{T}} \int_0^{t_f} \frac{\mathrm{i}}{\hbar} \langle a\chi_\perp, [\![\nabla \Psi \cdot \nu]\!] \rangle_{\partial K \setminus \partial\Omega} \,\mathrm{d}t \,\mathrm{d}x,$$

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

we have an estimate of the spatial error. Following [52], we assume that $\Psi-\psi^{\mathcal{T},p}$ is small and use Ξ_S to estimate the spatial error.

In the following, we analyze this assumption to understand potential shortcomings and the interplay of spatial and temporal tolerance. Let us split Ξ_S into two parts

$$\begin{aligned}\Xi_S := &-\left[\sum_{K\in\mathcal{T}}\int_0^{t_f}\langle\chi_\perp,\frac{\mathrm{i}}{\hbar}\hat{H}\widetilde{\psi}^{\mathcal{T},p}\rangle_K\mathrm{d}t+\sum_{K\in\mathcal{T}}\int_0^{t_f}\frac{\mathrm{i}}{\hbar}\left\langle a\chi_\perp,[\![\nabla\widetilde{\psi}^{\mathcal{T},p}\cdot\nu]\!]\right\rangle_{\partial K\setminus\partial\Omega}\mathrm{d}t\right]\\ &+\left[\sum_{K\in\mathcal{T}}\int_0^{t_f}\langle\chi_\perp,\frac{\mathrm{i}}{\hbar}\hat{H}(\widetilde{\psi}^{\mathcal{T},p}-\Psi)\rangle_K\mathrm{d}t\right.\\ &\left.+\sum_{K\in\mathcal{T}}\int_0^{t_f}\frac{\mathrm{i}}{\hbar}\left\langle a\chi_\perp,[\![\nabla(\widetilde{\psi}^{\mathcal{T},p}-\Psi)\cdot\nu]\!]\right\rangle_{\partial K\setminus\partial\Omega}\mathrm{d}t\right].\end{aligned}\tag{4.1}$$

The first part gives a purely spatial error whereas the second one is a mixed spatial-temporal error term. For the mesh adaption, we will compute cellwise contributions to the total error. Hence, we want the first part of Ξ_S as split in (4.1) to dominate the error on each cell. This means taking the whole term Ξ_S as an estimate for the spatial error is most critical on cells where the solution is rather small.

Let us assume that we have computed the solution using the temporal error estimation presented in Section 3.2 to a given error tolerance δ_t. Then, we know that any functional of $\widetilde{\psi}^{\mathcal{T},p}-\Psi$ is bounded by δ_t. In particular, we have a bound on the pointwise error. If we apply the Hamiltonian to a function from $V^{\mathcal{T},p}$, the function values on cell K are amplified on the order $\frac{1}{h_K^2}$ at most. This can be seen from the Gershgorin circle theorem for the largest eigenvalue of $\hat{H}$ discretized on the finite element space. So we get the estimation

$$\left\|\hat{H}\left(\widetilde{\psi}^{\mathcal{T},p}(\cdot,t)-\Psi(\cdot,t)\right)\right\|_K\leq C\sqrt{\mathrm{vol}(K)}\frac{\delta_t t}{t_f}\frac{1}{h_K^2}.$$

Furthermore, we have that $\chi_\perp\in\mathcal{O}(h_K^{p+1})$ due to the interpolation property of the finite element (cf. e.g. [54, Ch. 1]). This gives the following estimate

$$\begin{aligned}\int_K\int_0^{t_f}\chi_\perp^*\left(\frac{\mathrm{i}}{\hbar}\hat{H}(\widetilde{\psi}^{\mathcal{T},p}-\Psi)\right)\mathrm{d}x\mathrm{d}t&\leq\int_0^{t_f}\|\chi_\perp(\cdot,t)\|_K\left\|\frac{\mathrm{i}}{\hbar}\hat{H}(\widetilde{\psi}^{\mathcal{T},p(\cdot,t)}-\Psi(\cdot,t))\right\|_K\mathrm{d}t\\ &\leq C\frac{h_K^{p+1}}{h_K^2}\mathrm{vol}(K)\delta_t.\end{aligned}\tag{4.2}$$

The constant depends on the shape of ψ and χ on K as well as the finite element order (but not on h_K). For the jump term, a corresponding estimate can be derived.

If we want to refine the mesh according to some error tolerance δ_s, we would want the perturbation (4.2) of the spatial error estimate on cell K to be below $\delta_s\cdot\mathrm{vol}(K)$. Due to the factor h_k^{p-1} the term is usually small enough when $\delta_s=\delta_t$. In practice, the relation between the functional (that is χ) and the solution plays a crucial role. In case χ and ψ

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

are related, $\chi_\perp$ — and hence also (4.2) — will be small on cells where the spatial error is small. Recall that the term is only dominating when it is larger than the truly spatial part of Ξ_s. When deciding on the relation of δ_t and δ_s, we should therefore have in mind how much the primal and dual solutions overlap.

The integrals in the estimate have to be discretized on the temporal mesh since we know the solution in these points only. Also, the estimate (3.2) contains the exact solution of the dual problem which is, of course, unknown. Becker and Rannacher [9, Sec. 5.1] discuss different possibilities of estimating $\chi_\perp = \chi - \chi^{\mathcal{T},p}$ for a posteriori estimates. We choose to solve the dual with a global high-order approximation on the same grid and increased order of the finite elements (usually order $p+1$). In order to compute the part of the dual that is perpendicular to $V^{\mathcal{T},p}$, we interpolate the solution in the space $V^{\mathcal{T},p+1}$ to $V^{\mathcal{T},p}$ by injection. Note that this is not exact but for spatial dimensions larger than one (where the number of nodes on the boundary increases with the element order) this gives a good estimate.

4.1 Solving forward and backward

A major task when evaluating (3.2) is to find a memory and computing-time efficient method to handle the fact that the dual problem is solved starting from $t=t_f$ whereas the primal starts at $t=0$. A general strategy would be to use reversal schedules [55], i.e., to save the primal solution at a number of checkpoints and to recompute it between those. The strategy can also be applied recursively. This is both computational and memory intense. An alternative is to use Hölder's inequality to split the inner products of the residual and the dual solution in estimate (3.2). However, that procedure would strongly overestimate the error. On the other hand, this rough estimate can be achieved memory and CPU-time efficiently compared to reversal scheduling because we can precompute the dual solution and only save the accumulated norms for each cell. Both these strategies work in the general case. For the special case of a symmetric Hamiltonian, there is a simple and efficient alternative. One can exploit that the TDSE is time-reversible: The primal problem is solved forward in time without saving any intermediate information. Then both problems are solved backward synchronously which regenerates the needed information from the primal problem.

When solving numerically, we have to be careful not to accumulate errors when solving forward and backward. Concerning the spatial discretization, $u^{\mathcal{T},p}$ is an exact solution to the problem restricted to $V^{\mathcal{T},p}$ which is why the initial value (projected onto $V^{\mathcal{T},p}$) would be exactly recovered after the backward sweep (without discretization in time). The Magnus expansion is time-reversible, too. This leaves us with the error of the Lanczos algorithm and truncation errors (which are mostly negligible). Therefore, we use a sharper tolerance for the Lanczos algorithm by a safety factor for the error estimation.

4.2 Adaptive mesh refinement

In this section, we present our mesh refinement strategy based on the a posteriori estimate (3.2). Usually, we are not interested in a mesh that is adjusted after every single time

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

step but only after a certain number of time steps since mesh refinement and coarsening are associated with considerable computational costs. For bound states it might even be reasonable to only use one mesh with high resolution in interesting regions but no dynamic adaption. Since the estimate involves a forward and a backward problem, we use a (coarse) fixed mesh to evaluate the estimate.

After having computed an estimate of the error on each of the cells on the present grid, we have to decide where to refine or coarsen. Becker and Rannacher [9, Sec. 5.2] discuss different strategies. We choose an error-balancing strategy where the error on each cell should be below the tolerance divided by the total number of cells. In order to readjust the error estimate after a first sweep of grid refinement, we can assume that the error is proportional to h^{p+1} (cf. [9, Rem. 3.1 and Prop. 3.1]).

Our solution strategy can be summarized as follows:

- Solve primal problem forward (with temporal error control).
- Solve backwards: Solve primal and dual problem together for each time step and estimate the error Ξ_S (separately for each interval with constant grid in the second run).
- Adapt the grid:
  - Refine where the size of the error in the cell exceeds the threshold divided by the total number of cells.
  - Coarsen where the size of the error in the cell is below the threshold divided by the total number of cells and 2^{p+1} times some safety factor.
  - Update error accounting for the new element volumes using that the error is proportional to h^{p+1} and refine (and coarsen) once more in case the total error is still larger than the threshold.
- Solve the primal problem on the refined grid with error control in time.

Remark 4.1. In [31], it was pointed out that the error constant in the truncated Magnus expansion depends on the spatial mesh size. This problem does, however, only appear if both the temporal and the spatial step size are reduced simultaneously. In our algorithm, however, we only use temporal error control on a fixed spatial grid. Hence, the dependence of the temporal error constant on the spatial mesh is insignificant in our case.

5 Implementation

In order to be able to solve problems of practical interest, an efficient and parallel implementation of the algorithm discussed in the previous section is essential. As mentioned earlier, high-order methods are favorable to use for the solution of the TDSE. In a standard implementation of the finite element method based on sparse matrices, each degree

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

of freedom (DoF) is coupled to all the other DoFs within the cell(s) it belongs to. Therefore, the complexity per DoF for evaluating the Hamiltonian is $\mathcal{O}((p+1)^d)$, where $p+1$ is the order of the derivative approximation and d is the dimension of the problem. For higher order methods, this is considerably more expensive than finite difference stencils with complexity $\mathcal{O}(d(p+1))$. Moreover, the number of entries per row of the sparse matrix is quite large and the memory consumption of the sparse matrix is the limiting factor. In [41], an alternative matrix-free implementation that is based on a unit-cell stencil was proposed. This implementation is much less memory consuming and its complexity is of the same leading order as finite difference stencils. Still the rate constant can be larger than in the finite difference case. On the other hand, the spectral elements are much more flexible concerning variable coefficients and adaptivity. We base our implementation of the adaptive TDSE solver on the implementation described in [38,41] using the finite element library `deal.II` [5,6]. This implementation also provides routines for parallelizing the matrix-vector product both for shared and distributed memory. Furthermore, explicit vectorization is employed in order to best make use of arithmetic processing power. We will shortly review the construction principle in the first paragraph of this section.

For an implementation of our solver, we also need to be able to handle constraints on nodes that appear on faces where cells of different refinement level meet (see [7,50]). In combination with Gauss-Lobatto elements, care has to be taken not to destroy the diagonality of the mass matrix. We will shortly explain the construction principle of constraints and then how to retain a diagonal mass matrix in combination with constraints.

5.1 Matrix-free finite element operator

The idea for a memory-lean implementation of high-order finite elements is based on the observation that one can split the finite element operator application into a sum of local operations on the cells,

$$v = Au = \left(\sum_{k=1}^{n_{\text{cells}}} C^T P_k^T A_k P_k C \right) u. \tag{5.1}$$

The matrix P_k is a very sparse matrix (exactly one nonzero entry per row) that describes the local to global mapping of degrees of freedom on cell k and A_k is the matrix representing the finite element operation on cell k. Finally, C is the constraint matrix taking care of hanging nodes that need to be constrained to maintain C^0 continuity if the mesh is adaptively refined. The basic idea is not to form the matrix A explicitly, but to compute Au by the three successive steps on each cell, $k=1,\cdots,n_{\text{cells}}$:

- Extract element-local components of solution vector, $u_k = P_k C u$.
- Evaluate $v_k = A_k u_k$ by an efficient quadrature approach without forming A_k.
- Add local contributions into global result vector, $v \leftarrow v + C^T P_k^T v_k$.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

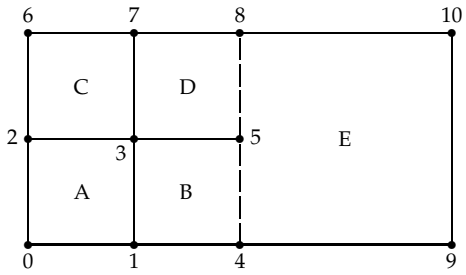


Figure 1: Example of a hanging node. DoF 5 has to be constrained along the dashed face.

Since the local evaluation has the same structure on all cells, we only have to store a unit-cell shape function information once and the information on the cell size for each of the elements. Moreover, one can exploit the tensor product structure of the Gauss-Lobatto elements and apply the local operations dimensionwise to reduce the complexity. For details on the efficient quadrature, we refer to [41].

5.2 Implementation of constraints

Fig. 1 shows a simple 2D example with hanging nodes for a linear element. The DoF with index five only exists on cells D and E on the left hand side of the edge but not on cell E. Since we want our solution to be continuous along the dashed face, we have to put a constraint on DoF 5 that forces the FE interpolation along the whole face to be linear.

Let us denote by ϕ_{Xi} the nodal linear basis function in node $i\in\{0,\cdots,10\}$ and cell $X\in\{\text{A,B,C,D,E}\}$. Then, the function u locally on cell B is given by

$$u(x)=u_1\phi_{\text{B1}}(x)+u_3\phi_{\text{B3}}(x)+u_4\phi_{\text{B4}}(x)+u_5\phi_{\text{B5}}(x). \tag{5.2}$$

Since DoF 5 is not present on cell E, we fix its value to $u_5=\frac{1}{2}(u_4+u_8)$ in order to ensure a linear function along the whole face in B and D. Taking this constraint into account, we can express (5.2) as

$$u(x)=u_1\phi_{\text{B1}}(x)+u_3\phi_{\text{B3}}(x)+u_4\left(\phi_{\text{B4}}(x)+\frac{1}{2}\phi_{\text{B5}}(x)\right)+u_8\frac{1}{2}\phi_{\text{B5}}(x). \tag{5.3}$$

Writing out the formulae, one can see that, restricted to the face connecting DoF 4 and 5, $\phi_{\text{B4}}(x)+\frac{1}{2}\phi_{\text{B5}}(x)$ is the same as $\phi_{\text{E4}}(x)$, and $\frac{1}{2}\phi_{\text{B5}}(x)$ equals ϕ_{E8}. Hence, the finite element approximation is continuous across the dashed edge. For higher order elements and higher dimensions the constraints become more complicated and involve a larger number of DoFs but the construction principle is the same (see [7]).

5.3 On the diagonality of the mass matrix in the presence of constraints

When constructing the mass matrix, we use the structure (5.1). So in case there are constraints, we also have to apply them to the entries of the mass matrix. Hence, the mass

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

matrix becomes nondiagonal on indices for which the matrix C is different from the identity despite the use of inexact quadrature in conjunction with Gauss-Lobatto elements.

To illustrate this, let us again consider the problem from Fig. 1 and denote the length of the large cell E by h. We use Gauss-Lobatto integration for linear elements (trapezoidal rule) on the displayed nodes and follow the assembly formula (5.1) for the global mass matrix. Consider the contribution of cell B on the rows and columns of global degrees of freedom 4, 5, and 8. This results in the submatrix

$$\begin{aligned} C^T\left(P_{\mathrm{B}}^T M_{\mathrm{B}} P_{\mathrm{B}}\right) C\big|_{4,5,8} &= \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix} \frac{h^2}{16} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \\ &= \frac{h^2}{16} \begin{pmatrix} \frac{5}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{pmatrix}. \end{aligned} \tag{5.4}$$

Hence, the global mass matrix based on assembly formula (5.1) is not diagonal, despite diagonal contributions from the cells.

However, the nondiagonality of the mass matrix is on the order of the error due to numerical quadrature: Alternatively, we could cluster cells B and C and consider the basis functions that we obtain after applying the constraints on the face with different refinement levels on the two sides (cf. (5.3)). This means we have Lagrange polynomials on the global degrees of freedom 1, 3, 4, 7, and 8. If we apply the Gauss-Lobatto quadrature rule on these degrees of freedom, we get a diagonal mass matrix.

In practice, we still want to evaluate the finite element operator for one cell at a time and avoid handling two cells together. Therefore, we use a modified assembly for the diagonal of the mass matrix,

$$\mathrm{diag}(M) = \sum_k C^T P_k^T \mathrm{diag}(M_k).$$

In the example for the contribution from cell B into degrees of freedom 4, 5, and 8 considered above, this gives

$$C^T\left(P_{\mathrm{B}}^T \mathrm{diag}(M_{\mathrm{B}})\right)\big|_{4,5,8} = \frac{h^2}{16} \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \frac{h^2}{16} \begin{pmatrix} \frac{3}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix},$$

i.e., the entries in the diagonal of the matrix correspond to the row-sum of entries in the matrix in (5.4). Since constant functions are integrated exactly, this lumping technique must give the same result as Gauss-Lobatto quadrature on the combined faces of cells B and C with transformed basis functions. For higher order elements, the argumentation is analogous.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

6 Numerical experiments

Firstly, we numerically analyze the spatial estimate on the sample problem of a harmonic oscillator. Since we know the analytical solution in this case, we can verify and analyze our numerical results. We also study dynamic mesh refinement for a 2D channel. Finally, we show results for the example of the OClO molecule which is a part of the ozone cycle [53].

6.1 Study of the spatial estimate

The potential of the harmonic oscillator is given by $V(x)=\frac{1}{2}\|x\|_2^2$ and the masses m_i are set to one. We choose the initial value

$$\psi_0(x)=\frac{1}{(\pi)^{d/4}}\mathrm{e}^{-0.5\|x-x_0\|_2^2} \tag{6.1}$$

with $x_0=(-1,\cdots,-1)$. The inner product of the solution with a predefined state is an important measurand in the chemical literature, the so-called cross-correlation. As an example, we consider the error in the cross-correlation with the (artificial) state

$$\phi(x)=\left(\frac{6}{\pi}\right)^{d/4}\mathrm{e}^{-3\|x\|_2^2+\mathrm{i}0.3\sum_{j=1}^{d}x_j}. \tag{6.2}$$

Since we are interested in analyzing the quality of the spatial discretization, we compute with a very strict tolerance in time so that we can neglect the temporal error in the computations. Looking at the reliability of the estimated error in the cross-correlation, we use the (reciprocal) effectivity index $\eta:=\left|\frac{\Xi_S}{\langle\phi,e\rangle}\right|$ being the ratio between estimated error and real error. Thus, $\eta=1$ means that the error estimate is perfect. The larger the factor η gets, the more the error is overestimated, and the smaller η the more underestimated.

Error verification. We start with the one-state harmonic oscillator with known analytical solution. The simulation time is set to $t_f=1.7\pi$ and the cross-correlation is 0.292 in 2D.

The convergence rates collected in Table 1 show that we get a good overall convergence of the method. For $\mathcal{Q}_p$ elements, the theoretical rate of convergence is $p+1$. However, the results show some wiggling around the theoretical rates. Especially, the error in the cross-correlation is not exactly predictable. This makes the error estimation and the adaptation of the mesh more difficult. Nevertheless, the error is overestimated by at most a factor 1.6 and underestimated by at most a factor 3.2.

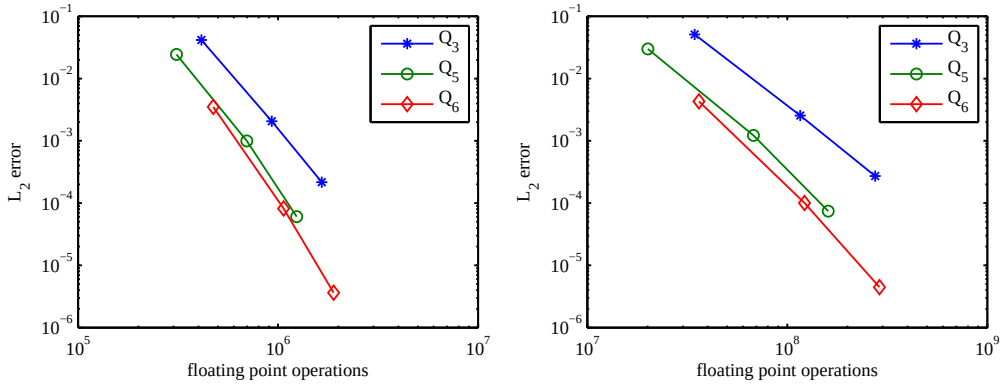
Higher order pays off. To show that high order pays off for the solution of the time-dependent Schrödinger equation, we compare the error for different elements after time propagation over 1.7π time units for the harmonic oscillators with analytical solution.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

Table 1: Two-dimensional harmonic oscillator (example with analytical solution). Actual errors (both L_2 and in cross-correlation), convergence rates as well as estimated errors are shown.

order	# cells	ℓ_2 error	rate	error cross.	rate	est. error	η
$\mathcal{Q}_4$	64	$1.79\cdot10^{-2}$	—	$1.94\cdot10^{-2}$	—	$6.77\cdot10^{-3}$	0.35
	256	$1.78\cdot10^{-3}$	6.7	$2.17\cdot10^{-4}$	6.5	$5.89\cdot10^{-5}$	0.27
	1024	$6.34\cdot10^{-6}$	8.1	$1.99\cdot10^{-7}$	10	$3.23\cdot10^{-7}$	1.63
$\mathcal{Q}_5$	64	$2.45\cdot10^{-2}$	—	$1.00\cdot10^{-3}$	—	$1.09\cdot10^{-3}$	1.1
	256	$6.05\cdot10^{-5}$	8.7	$8.19\cdot10^{-6}$	6.9	$2.52\cdot10^{-6}$	0.31
	1024	$2.98\cdot10^{-7}$	7.7	$1.80\cdot10^{-9}$	12	$2.25\cdot10^{-9}$	1.2
$\mathcal{Q}_6$	64	$3.51\cdot10^{-3}$	—	$1.96\cdot10^{-4}$	—	$1.55\cdot10^{-4}$	0.70
	256	$3.64\cdot10^{-6}$	9.9	$8.91\cdot10^{-8}$	11	$6.43\cdot10^{-8}$	0.72
	1024	$1.39\cdot10^{-8}$	8.0	$3.62\cdot10^{-11}$	11	$2.12\cdot10^{-11}$	0.59

The value of the cross-correlation is 0.292 in 2D and 0.157 in 3D. We have done four pairs of computations for $\mathcal{Q}_3$ and $\mathcal{Q}_6$ elements with the same numbers of nodes on $8^d/4^d$, $24^d/12^d$, and $32^d/16^d$ cells. The errors are shown in Figs. 2(a) and 2(b) as a function of the number of floating point operations per matrix-vector product in 2D and 3D, respectively. The error for the sixth order element is up to two orders of magnitude lower than the third order one for the same number of degrees of freedom. On the other hand, the computational complexity is only increased by about 15% in 2D and 5% in 3D. Moreover, we have done computations on the coarser meshes with $\mathcal{Q}_5$ elements. The error for $\mathcal{Q}_5$ elements is smaller than for $\mathcal{Q}_3$ elements on a mesh that is one level finer in all cases despite the lower computing costs. Moreover, higher arithmetic intensity can be achieved with increasing order which gives additional speedup in practice.



(a) 2D (b) 3D

Figure 2: Comparison of error per node for third, fifth and sixth order elements on a harmonic oscillator in two and three dimensions. The L_2 error is plotted over the number of floating point operations per matrix-vector product.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

Explicit time-dependence. We consider the example of two coupled harmonic oscillators with

$$V(x,t)=\begin{pmatrix} \frac{1}{2}\|x\|_2^2 & \mathrm{e}^{-2(t-0.5)^2}\cos(t-0.5) \\ \mathrm{e}^{-2(t-0.5)^2}\cos(t-0.5) & \frac{1}{2}\|x\|_2^2+1 \end{pmatrix},$$

where a time-dependence is introduced to the potential. Since we do not know the analytical solution in this case, we compare against a solution computed with $\mathcal{Q}_{10}$ elements. The simulation time is set to 0.7π and the value of the cross-correlation is -0.223 in 2D and -0.200 in 3D. Table 2 shows the results for 2D and Table 3 for 3D. Again the observed convergence rates are close to the theoretically predicted rates and the error estimate is over- and underestimating at most by a factor six.

Table 2: Two coupled harmonic oscillators. 2D.

order	# cells	L_2 error	rate	error cross.	rate	est. error	η
$\mathcal{Q}_4$	64	$9.34\cdot10^{-2}$	—	$5.12\cdot10^{-3}$	—	$3.83\cdot10^{-3}$	0.75
	256	$1.63\cdot10^{-3}$	5.8	$5.04\cdot10^{-5}$	6.7	$1.76\cdot10^{-5}$	0.35
	1024	$4.49\cdot10^{-5}$	5.1	$2.40\cdot10^{-8}$	11	$1.44\cdot10^{-7}$	6.0
$\mathcal{Q}_5$	64	$1.56\cdot10^{-2}$	—	$6.71\cdot10^{-4}$	—	$3.25\cdot10^{-4}$	0.48
	256	$1.99\cdot10^{-4}$	6.3	$1.65\cdot10^{-6}$	8.7	$9.51\cdot10^{-7}$	0.58
	1024	$3.16\cdot10^{-6}$	6.0	$1.02\cdot10^{-9}$	11	$7.08\cdot10^{-10}$	0.69
$\mathcal{Q}_6$	64	$3.19\cdot10^{-3}$	—	$8.54\cdot10^{-5}$	—	$3.45\cdot10^{-5}$	0.40
	256	$2.45\cdot10^{-5}$	7.0	$1.80\cdot10^{-8}$	12	$1.58\cdot10^{-8}$	0.88
	1024	$2.05\cdot10^{-7}$	6.9	$1.55\cdot10^{-11}$	10	$1.07\cdot10^{-11}$	0.69

Table 3: Two coupled harmonic oscillators. 3D.

order	# cells	L_2 error	rate	error cross.	rate	est. error	η
$\mathcal{Q}_4$	512	$1.14\cdot10^{-1}$	—	$1.77\cdot10^{-2}$	—	$9.86\cdot10^{-3}$	0.56
	1728	$9.43\cdot10^{-3}$	6.2	$2.41\cdot10^{-4}$	11	$2.56\cdot10^{-4}$	1.1
	4096	$2.00\cdot10^{-3}$	5.4	$1.21\cdot10^{-4}$	2.4	$3.45\cdot10^{-5}$	0.28
$\mathcal{Q}_5$	512	$1.91\cdot10^{-2}$	—	$1.75\cdot10^{-3}$	—	$3.22\cdot10^{-4}$	0.18
	1728	$1.45\cdot10^{-3}$	6.4	$3.06\cdot10^{-5}$	10	$2.20\cdot10^{-5}$	0.72
	4096	$2.43\cdot10^{-4}$	6.2	$4.27\cdot10^{-6}$	6.8	$1.73\cdot10^{-6}$	0.41
$\mathcal{Q}_6$	512	$3.91\cdot10^{-3}$	—	$2.18\cdot10^{-4}$	—	$9.25\cdot10^{-5}$	0.42
	1728	$2.20\cdot10^{-4}$	7.1	$1.39\cdot10^{-6}$	12	$1.07\cdot10^{-6}$	0.77
	4096	$3.01\cdot10^{-5}$	6.9	$9.60\cdot10^{-8}$	9.3	$2.31\cdot10^{-8}$	0.24

Efficiency of the adaptivity. In order to investigate how efficient the adaptivity based on our error estimate works, we redo the experiment with the coupled harmonic oscillator. We solve the problem with $\mathcal{Q}_6$ elements on a grid with 6^d cells and use $\mathcal{Q}_7$ elements

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

to estimate the error in the cross-correlation. Then, we refine the grid with a tolerance of 10^{-5}. In two dimensions this gives an adaptive mesh with 54 cells and an error of $4.96\cdot10^{-6}$ in the cross-correlation. For this adapted grid we hence get an error that is about a factor 17 smaller compared to the equidistant grid with 64 cells (cf. Table 2). A three dimensional simulation gives an adaptive mesh with 321 cells and an error of $1.98\cdot10^{-6}$ in the cross-correlation, a factor 110 better than on a uniform mesh with 512 cells.

Discussion. Since the error estimate (4.1) gives an upper bound of the error, a (slight) overestimation of the error should not be surprising. In our experiments, we often observe an underestimation instead. The reason for this is that we do not use the exact value of the dual problem as it appears in the theoretical estimate but use an approximation on the same grid with an FE approximation based on one order higher elements. Hence, we get the projection of the error onto the subspace $V^{p+1,h}\backslash V^{p,h}$ which is only a part of the error. An improvement can be achieved by choosing an even higher order space for the approximation of the dual. However, this will increase the computational costs which is usually not worthwhile.

6.2 Dynamic mesh refinement

We study the motion of a 2D wave packet in a channel defined by a harmonic-oscillator potential in one of the dimensions, $V(x_1,x_2)=\frac{1}{2}x_2^2$. The solution at time t is given by

$$\psi(x,t)=\pi^{-1/2}(1+\mathrm{i}t)^{-1/2}\exp\left(-\frac{(x_1-1-2t)^2}{2+2\mathrm{i}t}-\frac{x_2^2}{2}+2\mathrm{i}(x_1-1-2t)+\mathrm{i}\frac{3}{2}t\right).$$

The simulation time is 4 and the mesh is dynamically adapted four times. We choose $\mathcal{Q}_6$ elements and a tolerance of 10^{-3}. The error functional is defined by a state similar to (6.2) but shifted to $(7,0)$. Fig. 3 shows the four grids together with the real part of the solution just after changing the grid. It can be seen that the adapted grid nicely follows the solution that is traveling from the left to the right. One of the grids is not symmetric about $x_2=0$ which is an effect of the (non-symmetric) dual solution. The error in the cross-correlation on the refined mesh is $6.0\cdot10^{-4}$ so the error estimation is both efficient and effective.

6.3 Adaptivity for OClO

In order to challenge our algorithm, we solve a practical problem with the method described in Section 4.2. We consider a coupled system of the form

$$\mathrm{i}\hbar\frac{\partial}{\partial t}\begin{pmatrix}\psi_a\\ \psi_b\end{pmatrix}=\begin{pmatrix}H_a & V_{a,b}\\ V_{a,b}^* & H_b\end{pmatrix}\begin{pmatrix}\psi_a\\ \psi_b\end{pmatrix},$$

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

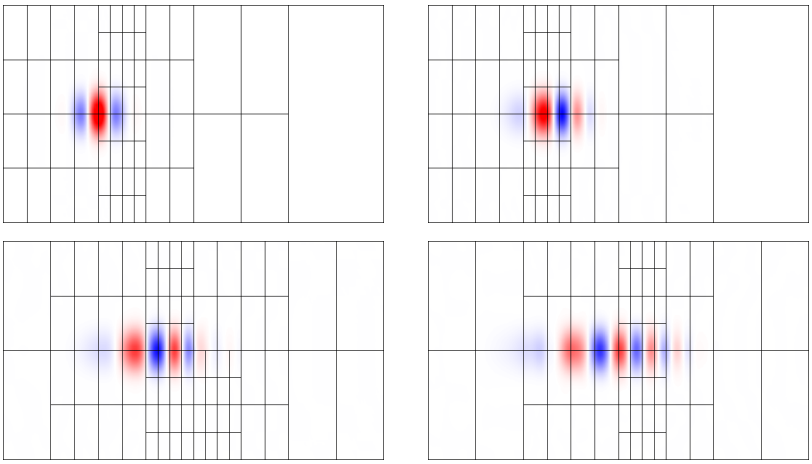


Figure 3: Dynamic mesh.

describing matter-field interaction in the OClO molecule. For the parameters that describe the Hamiltonian operator, we refer to [53]. For the initial value, $\psi_{a,0}$ is a state as defined in (6.1) with $x_0=(2.3,2.3,2.1)^T$ and $\psi_{b,0}=0$. The coupling term modeling the laser pulse is

$$V_{a,b}(t)=10^{-3}\cdot\exp\left(-0.5\frac{(t-t_0)^2}{\sigma^2}\right)\cdot\cos(\omega(t-t_0)),$$

where $\sigma=4.56$ fs, $t_0=7.5$ fs, and $\omega=0.1179$ fs^{-1}. The simulation time is 10 fs.

For this example, we use $\mathcal{Q}_7$ elements for the primal solution and $\mathcal{Q}_8$ elements for the dual. For the initial error estimation, we use a mesh that consists of $24\times24\times12$, that is 6912, cells. Then, we estimate the error in the auto-correlation functional, i.e., in the overlap of the solution at final time with the initial value, and adjust the mesh.

We work with a tolerance of 10^{-3}. After adjusting the mesh to the solution according to the estimated error, we get a mesh with 31,433 cells and 11,231,828 nodes or about 45 million DoFs (accounting for real and imaginary part on two energy levels). In order to check the quality of the estimate, we consider the error in the functional compared to a simulation on the same mesh with $\mathcal{Q}_8$ elements which is computed to be $4.7\cdot10^{-5}$. Considering the fact that the error on that mesh was estimated to be $3.5\cdot10^{-5}$ by extrapolation from the coarse mesh, one can conclude that the estimation has worked quite well. Fig. 4 shows the mesh together with the absolute value of the error. On the first sight, this result might be surprising since the major part of the error is located in a region with quite coarse mesh resolution. However, we have to bear in mind that we did not aim at reducing the error itself but a functional of it. This functional is defined by the initial value which is localized in the part of the mesh that is refined. Hence, the illustration shows that the mesh refinement is indeed adapted to a specific functional. The simulation is not only adaptive in the spatial coordinates but also in time. It performs 24,958 time steps with 184,334 matrix-vector products and the wall time for the computation on eight cores

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

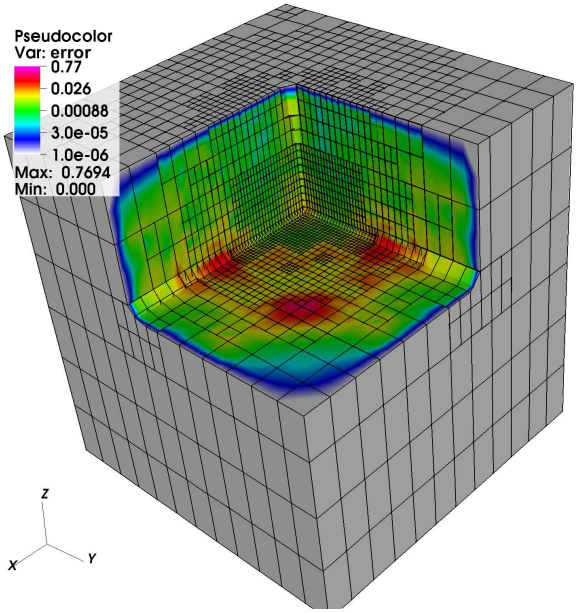


Figure 4: OCIO simulation: Refined mesh with absolute value of the error.

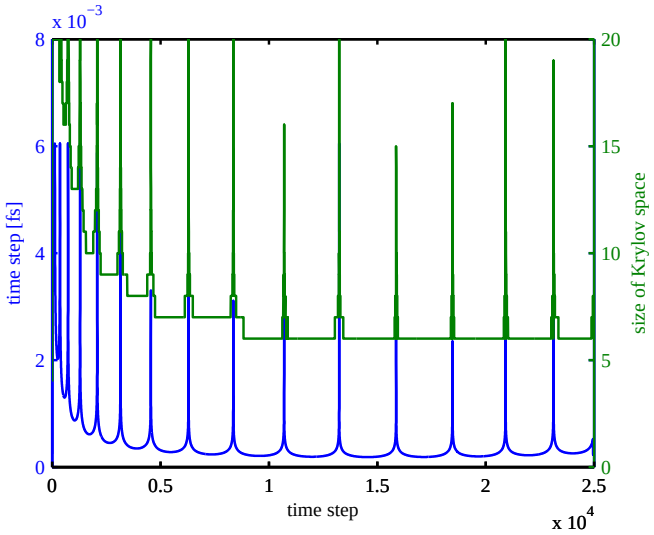


Figure 5: OCIO simulation: Adaptive choice of time step and size of Krylov subspace.

on a Intel Xeon E5520 (Nehalem-EP) system is about 38 hours. Fig. 5 shows how the time step size and the dimension of the Krylov space are chosen. This temporal adaptivity is governed by the oscillations in the time-dependent part of the Hamiltonian.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

7 Conclusions

We have derived an a posteriori error estimate for discretization of the time-dependent Schrödinger equation with spectral elements in space and an exponential integrator in time. Numerical simulations have shown that the error estimate is efficient. Based on this estimate, we have explained how to design adaptive meshes that reduce the number of degrees of freedom needed to solve the TDSE to a given accuracy. We have demonstrated how this technique can be applied to the interaction of the three dimensional OClO molecule with a laser pulse. This large simulation was enabled by using an efficient and parallel implementation of the matrix-vector products and by exploiting the time-reversibility of the equation.

The prototype implementation is based on the finite element library deal.II which can only handle problems in one to three dimensions. In order to be able to solve large-scale application problems, it would be necessary to build the implementation on another framework like the HAParaNDA code [27] that is tailored to high-dimensional problems. Since global time steps are applied, relatively small time steps are enforced by the finest level of the grid. Employing some local time stepping strategy is therefore expected to improve the performance of our framework.

Acknowledgments

Discussions with Wolfgang Bangerth, Sverker Holmgren, Hans O. Karlsson, Emil Kieri, Martin Kronbichler, and Axel Målqvist are gratefully acknowledged. Most of the work was performed under the previous affiliation of the author at the Department of Information Technology, Uppsala University, and the computations were performed on resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under projects p2003013 and p2005005.

References

[1] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt, and A. Schädle. A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations. *Commun. Comput. Phys.*, 4:726–796, 2008.

[2] I. Babuška and A. D. Miller. The post-processing approach in the finite element method, III: A posteriori error estimation and adaptive mesh selection. *Int. J. Numer. Meth. Eng.*, 20:2311–2324, 1984.

[3] I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *Int. J. Numer. Meth. Eng.*, 12:1597–1615, 1978.

[4] I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15:736–754, 1978.

[5] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II — a general-purpose object-oriented finite element library. *ACM Trans. Math. Softw.*, 33(4), 2007.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

[6] W. Bangerth, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and T. D. Young. The deal.II library, version 8.1. *arXiv preprint*, 2013.
[7] W. Bangerth and O. Kayser-Herold. Data structures and requirements for hp finite element software. *ACM Trans. Math. Softw.*, 36(1):4/1–4/31, 2009.
[8] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.
[9] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.
[10] M. Ben-Nun, J. Quenneville, and T. J. Martínez. Ab initio multiple spawning: Photochemistry from first principles quantum molecular dynamics. *J. Phys. Chem. A*, 104:5161–5175, 2000.
[11] R. Bermejo and J. Carpio. A space-time adaptive finite element algorithm based on dual weighted residual methodology for parabolic equations. *SIAM J. Sci. Comput.*, 31(5):3324–3355, 2009.
[12] G. D. Billing. Time-dependent quantum dynamics in a Gauss–Hermite basis. *J. Chem. Phys.*, 110:5526–5537, 1998.
[13] S. Blanes, F. Casas, J. Oteo, and J. Ros. The Magnus expansion and some of its applications. *Phys. Rep.*, 470:151 – 238, 2009.
[14] Y. Cao and L. Petzold. A posteriori error estimate and global error control for ordinary differential equations by the adjoint method. *SIAM J. Sci. Comput.*, 26(3):359 – 374, 2004.
[15] G. C. Cohen. *Higher-Order Numerical Methods for Transient Wave Equations*. Springer Verlag, Berlin, 2002.
[16] W. Dörfler. A time- and spaceadaptive algorithm for the linear time-dependent Schrödinger equation. *Numer. Math.*, 73:419 – 448, 1996.
[17] M. Durufle, P. Grob, and P. Joly. Influence of Gauss and Gauss–Lobatto quadrature rules on the accuracy of a quadrilateral finite element method in the time domain. *Numer. Meth. Part. D. E.*, 25:526–551, 2009.
[18] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problem. I: A linear model problem. *SIAM J. Numer. Anal.*, 28:43–77, 1991.
[19] E. Faou and V. Gradinaru. Gauss-Hermite wave packet dynamics: convergence of the spectral and pseudo-spectral approximation. *IMA J. Numer. Anal.*, 29:1023–1045, 2009.
[20] E. Faou, V. Gradinaru, and C. Lubich. Computing semiclassical quantum dynamics with Hagedorn wavepackets. *SIAM J. Sci. Comput.*, 31:3027–3041, 2009.
[21] E. Fattal, R. Baer, and R. Kosloff. Phase space approach for optimizing grid representations: The mapped Fourier method. *Phys. Rev. E*, 53:1217–1227, 1996.
[22] M. D. Feit, J. Fleck, J. A., and A. Steiger. Solution of the Schrödinger equation by a spectral method. *J. Comput. Phys.*, 47:412–433, 1982.
[23] J. Gagelman and H. Yserentant. A spectral method for Schrödinger equations with smooth confinement potentials. *Numer. Math.*, 122:383–398, 2012.
[24] V. Gradinaru. Fourier transform on sparse grids: Code design and the time dependent Schrödinger equation. *Computing*, 80:1–22, 2007.
[25] M. Griebel and J. Hamaekers. Sparse grids for the Schrödinger equation. *ESAIM-Math. Model. Num.*, 41:215–247, 2007.
[26] B. Gustafsson. *High Order Difference Methods for Time Dependent PDE*. Springer, Berlin, 2008.
[27] M. Gustafsson and S. Holmgren. An implementation framework for solving high-dimensional PDEs on massively parallel computers. In G. Kreiss, P. Lötstedt, A. Målqvist, and M. Neytcheva, editors, *Numerical Mathematics and Advanced Applications 2009*, pages

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

417–424. Springer, Berlin, 2010.
[28] K. Hallatschek. Fouriertransformation auf dünnen Gittern mit hierarchischen Basen. *Numer. Math.*, 63:83–97, 1992.
[29] H. Han, D. Yin, and Z. Huang. Numerical solutions of Schrödinger equations in r^3. *Numer. Meth. Partial Diff. Eqs.*, 23:511–533, 2007.
[30] E. J. Heller. Frozen Gaussians: A very simple semiclassical approximation. *J. Chem. Phys.*, 75:2923–2931, 1981.
[31] M. Hochbruck and C. Lubich. On Magnus integrators for time-dependent Schrödinger equations. *SIAM J. Numer. Anal.*, 41:945–963, 2003.
[32] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.*, 19(5), 1999.
[33] O. Karakashian and C. Makridakis. A space-time finite element method for the nonlinear Schrödinger equation: the continuous Galerkin method. *SIAM J. Numer. Anal.*, 36:1779–1807, 1999.
[34] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2nd edition, 2005.
[35] U. Kleinekathöfer and D. J. Tannor. Extension of the mapped Fourier method to time-dependent problems. *Phys. Rev. E*, 60:4926, 1999.
[36] K. Kormann, S. Holmgren, and H. O. Karlsson. Accurate time propagation for the Schrödinger equation with an explicitly time-dependent Hamiltonian. *J. Chem. Phys.*, 128:184101, 2008.
[37] K. Kormann, S. Holmgren, and H. O. Karlsson. Global error control of the progagation for the Schrödinger equation with a time-dependent Hamiltonian. *J. Comput. Sci.*, 2:178–187, 2011.
[38] K. Kormann and M. Kronbichler. Parallel finite element operator application: Graph partitioning and coloring. In *2011 Seventh IEEE International Conference on eScience*, pages 332 –339, 2011.
[39] K. Kormann and A. Nissen. Error control for simulations of a dissociative quantum system. In G. Kreiss, P. Lötstedt, A. Målqvist, and M. Neytcheva, editors, *Numerical Mathematics and Advanced Applications 2009*, pages 523–531. Springer, Berlin, 2010.
[40] D. Kosloff and R. Kosloff. A Fourier method solution for the time dependent Schrödinger equation as a tool in molecular dynamics. *J. Comput. Phys.*, 52:35–53, 1983.
[41] M. Kronbichler and K. Kormann. A generic interface for parallel cell-based finite element operator application. *Comp. Fluids*, 63:135–147, 2012.
[42] I. Kyza. A posteriori error analysis for the Crank-Nicolson method for linear Schrödinger equations. *ESAIM-Math. Model. Num.*, 45:761–778, 2011.
[43] C. Lubich. Integrators for Quantum Dynamics: A Numerical Analyst's Brief Review. In J. Grotendorst, D. Marx, and A. Muramatsu, editors, *Quantum Simulation of Complex Many-Body Systems: From Theory to Algorithms*, pages 459 – 466. John von Neumann Institute for Computing, 2002.
[44] D. E. Manolopoulos and R. E. Wyatt. Quantum scattering via the log derivative version of the Kohn variational principle. *Chem. Phys. Lett.*, 152:23–32, 1988.
[45] H.-D. Meyer, F. Gatti, and G. W. (Eds.). *Multidimensional Quantum Dynamics: MCTDH Theory and Applications*. Wiley-VCH, Weinheim, 2009.
[46] W. H. Miller. Semiclassical methods in chemical physics. *Science*, 233:171–177, 1986.
[47] T. J. Park and J. C. Light. Unitary quantum time evolution by iterative Lanczos reduction. *J. Chem. Phys.*, 85(10), 1986.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press

[48] M. Picasso. Adaptive finite elements for a linear parabolic problem. *Comput. Methods Appl. Mech. Eng.*, 167:223–237, 1998.
[49] T. N. Rescigno and C. W. McCurdy. Numerical grid methods for quantum-mechanical scattering problems. *Phys. Rev. A*, 62:032706, 2000.
[50] W. C. Rheinboldt and C. K. Mesztenyi. On a Data Structure for Adaptive Finite Element Mesh Refinements. *ACM Trans. Math. Softw.*, 6:166–187, 1980.
[51] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29:209–228, 1992.
[52] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.*, 30(1):369–393, 2008.
[53] Z. Sun, N. Lou, and G. Nyman. Time-Dependent Wave Packet Split Operator Calculations on a Three-Dimensional Fourier Grid in Radau Coordinates Applied to the OClO Photoelectron Spectrum. *J. Phys. Chem. A*, 108:9226–9232, 2004.
[54] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems (2nd edn.)*, volume 25 of *Springer Series in Computational Mathematics*. Springer, Berlin, 2006.
[55] A. Walther. *Program Reversal Schedules for Single and Multi-processor Machines*. PhD thesis, Institute of Scientific Computing, TU Dresden, 1999.
[56] J. A. C. Weideman. Spectral differentiation matrices for the numerical solution of Schrödinger's equation. *J. Phys. A: Math. Gen.*, 39:10229–10237, 2006.
[57] Y. Wu and V. S. Batista. Matching-pursuit for simulations of quantum processes. *J. Chem. Phys.*, 118:6720–6724, 2003.
[58] A. H. Zewail. Laser femtochemistry. *Science*, 242:1645–1653, 1988.

https://doi.org/10.4208/cicp.101214.021015a Published online by Cambridge University Press