A fast matrix-free algorithm for spectral approximations to the Schrödinger equation

Bernd Brumm
Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D-72076 Germany

Abstract. We consider the linear time-dependent Schrödinger equation with a time-dependent potential on an unbounded domain. Using a Galerkin spectral method with a tensor-product Hermite basis as a discretization in space and a Magnus integrator for the time approximation of the resulting ODE for the Hermite expansion coefficients, we propose a fast algorithm for the direct computation of the action of the stiffness matrix on a vector without actually assembling the matrix itself, as required in each time step. Together with the application of a hyperbolically reduced basis, this reduces the computational effort considerably and helps coping with the infamous curse of dimensionality. The analysis is based on a representation of the three-term recurrence relation for the one-dimensional Hermite functions as a full binary tree. The fast algorithm constitutes an efficient tool for schemes involving the action of a matrix due to spectral discretization on a vector, thus, it can be applied also in the context of splitting procedures as well as for spectral approximations for linear problems other than the Schrödinger equation.

Keywords: linear Schrödinger equation, spectral Galerkin methods, Magnus integrators, reduced index sets, fast algorithm, direct computation, curse of dimensionality, binary trees

Introduction

We consider the linear time-dependent Schrödinger equation

$$i\frac{\partial}{\partial t}\psi(x,t) = (H\psi)(x,t)$$

in $N$ spatial dimensions with $x = (x_1, \ldots, x_N) \in \mathbb{R}^N$, $t \in [0, T]$, where the Hamiltonian

$$(H\psi)(x,t) = (T\psi)(x,t) + (V\psi)(x,t) = \frac{1}{2}(\Delta\psi)(x,t) + V(x,t)\psi(x,t)$$

consists of the negative Laplacian plus a real-valued, possibly time-dependent multiplicative potential. For an underlying geometry as simple as in (1), spectral methods are a natural means of discretization in space. In a naive approach, the resulting system of ordinary differential equations grows exponentially with the spatial dimension, making an accurate approximation practically unfeasible even for moderate choices of $N$. For this difficulty, the catch phrase curse of dimensionality has been coined. Complicating this even more, time propagation typically requires computing the action of the stiffness matrix on a vector in each step, and, in case of a time-dependent potential, the matrix has to be re-assembled.

A promising strategy is a suitable reduction of the spectral approximation basis. E.g., Gradinaru (2007a) and Gradinaru (2007b) study a spectral approach with collocation on a sparse grid in case of a time-independent potential and periodic boundary conditions with a hyperbolically reduced tensor-product Fourier basis. Lubich (2008), Chapter III.1.4, points out that, unlike on a full grid,
the resulting coefficient ODE does not exhibit a Hermitian stiffness matrix, thus, possibly giving
rise to numerical troubles as well as limiting the range of applicable time-stepping methods. As a
remedy, amongst others, a Fourier Galerkin method with an approximated potential is proposed,
this being a model for our own approach – in the much simpler setting of a periodic problem.

In the present paper, allowing the potential to be time-dependent and considering an unbounded
domain instead of a periodic problem, we employ a spectral Galerkin approach. Hermite functions
are a natural and, thus, widely-used spectral basis for the Schrödinger equation on unbounded do-
main, see, e.g., Lubich (2008), Chapter III.1, Faou & Gradinaru (2009) for the linear and Gauckler
(2011) for a nonlinear case.

Besides working with a hyperbolically reduced basis, we develop a fast algorithm for the direct
computation of the aforementioned matrix-vector-product that speeds up propagation in time con-
siderably. First, we approximate the potential by a polynomial. Using a recurrence relation for the
univariate Hermite functions and orthogonality of the given basis, we define auxiliary matrices for
each coordinate direction to act directly on a vector. A suitable entrywise approximation of the
stiffness matrix by Gauss-Hermite quadrature is equivalent to the formal insertion of the auxiliary
matrices into the polynomially approximated potential – as long as the matrices are indexed over
a full grid. By a conversion of the Hermite recurrence relation into an underlying structure of full
binary trees, the resulting quadrature error as well as the error due to a hyperbolical index reduction
in the fast algorithm are analyzed. Both errors are well-behaved if the potential can be sufficiently
well approximated by a multivariate polynomial. If so, we get estimates $O(C(R, W)K^{-\beta})$ and
$O(C(N, R, W, \beta)K^{-\beta})$, respectively, see Theorems 1 and 2. Here, $W$ is the part of the potential $V$
that is approximated over an $N$-dimensional index set $R(R)$ with maximal univariate polynomial
degree $R$, $K \gg R$ is the maximal number of basis functions employed in each coordinate direction
in the Galerkin approximation, and the coefficients of the approximate solution exhibit a decay of
order $\beta$ with increasing index.

The main idea underlying the fast algorithm, in a very rudimentary form, can be found as early as in
Carrington & Roy (1996): to use constructive properties of the basis to compute directly the action
of a discretized Hamiltonian operator on a vector. The algorithm itself was proposed in Faou, Grad-
inaru & Lubich (2009) in the context of linear Schrödinger equations in the semi-classical regime.
We develop their idea further in linking the matrix representation of (parts of) the Hamiltonian
operator to a suitable quadrature formula, as is done – in case of a fully indexed basis – in Discrete
Variable Representations, see Light & Carrington (2000), thus bringing together grid reduction and
DVR techniques, and we provide a detailed analysis based on binary tree representations. Hence, the
algorithm constitutes a useful tool also in the context of Faou, Gradinaru & Lubich (2009) as well
as Gradinaru & Hagedorn (2013), where splitting procedures using time-dependent, semi-classical
Hagedorn wavepackets are proposed. Furthermore, besides splitting procedures, the fast algorithm
as developed in the present paper has a range of applications much wider than just spectral Galerkin
approximations to the linear Schrödinger equation.

In Section 1 after briefly reviewing the construction of Hermite functions, we deduce the ODE
system for the Hermite expansion coefficients from the Galerkin ansatz with a polynomially approx-
imated potential, give a short description of Gauss-Hermite quadrature in order to discretize the
entries of the stiffness matrix, and introduce hyperbolically reduced index sets. Section 2 outlines
the discretization in time using Magnus integrators, where the matrix exponential is approximated
using a Lanczos method. In Section 3 we develop the fast algorithm for the matrix-free computation
of the actions of the stiffness matrix on a vector in each Lanczos step: In detail, we deduce auxiliary
matrices, give an outline of the algorithm itself, comment on its usage for reduced index sets, and compare the computational costs to a naive approach. Section 4 provides an algorithmic description of the overall procedure. A detailed error analysis is given in Section 5, where, amongst others, we study the errors due to grid reduction and due to quadrature. Section 6 presents some numerical experiments confirming the theoretical results. In Section 7, we mention further applications of the fast algorithm.

1 Semi-discretization in space

1.1 Construction of Hermite basis

In one dimension: Let \((q\psi)(x) = x\psi(x)\) and \(p = -id/dx\) denote the one-dimensional position and momentum operators, respectively. We start from \(\varphi_{-1} \equiv 0\) and \(\varphi_0(x) = \pi^{-1/4} e^{-x^2/2}\). As shown in, e.g., [Thaller 2000], Section 7.7, or [Lubich 2008], Chapter III.1.1, the one-dimensional ladder operators given by

\[
A = \frac{1}{\sqrt{2}} (q + ip), \quad A^\dagger = \frac{1}{\sqrt{2}} (q - ip),
\]

yield raising and lowering relations \((k \geq 0)\)

\[
\varphi_{k+1} = \frac{1}{\sqrt{k+1}} A^\dagger \varphi_k, \quad \varphi_{k-1} = \frac{1}{\sqrt{k}} A \varphi_k
\]

for the Hermite functions \(\{\varphi_k\}_{k \in \mathbb{N}}\). The Hermite functions lie in the space \(S(\mathbb{R})\) of Schwartz functions and form a complete \(L^2(\mathbb{R})\)-orthonormal set, in particular, \((\varphi_j, \varphi_k) = \delta_{jk}\), where \((f, g) = \int f \overline{g}\) denotes the standard \(L^2\)-inner product. By virtue of the above construction, they are readily seen to be the eigenfunctions of the harmonic oscillator, i. e.,

\[
\frac{1}{2} (p^2 + q^2) \varphi_k = \left( k + \frac{1}{2} \right) \varphi_k,
\]

see, e. g., [Thaller 2000], Section 7.7. The useful three-term recurrence relation \((k \geq 0)\)

\[
x \varphi_k(x) = \sqrt{\frac{k+1}{2}} \varphi_{k+1}(x) + \sqrt{\frac{k}{2}} \varphi_{k-1}(x),
\]

easily follows from (3). An explicit expression is

\[
\varphi_k(x) = \pi^{-1/4} \left( 2^k k! \right)^{-1/2} H_k(x) e^{-x^2/2},
\]

where \(H_k\) denotes the classical Hermite polynomial, thus, \(\varphi_k\) is a polynomial of degree \(k\) times a Gaussian. We have \(|\varphi_k(x)| \leq 1\), for all \(k \in \mathbb{N}\) and \(x \in \mathbb{R}\), see, e. g., [Abramowitz & Stegun 1965], Section 22, for useful facts about classical orthogonal polynomials. Figure 1 shows some plots. The ladder operators (2) are adjoint to one another on \(S(\mathbb{R})\), i. e.,

\[
(\varphi, A^\dagger \psi) = (A \varphi, \psi), \quad \forall \varphi, \psi \in S(\mathbb{R}),
\]

which follows easily from integration by parts.

In higher dimensions: We consider tensor-products of Hermite functions, i. e.,

\[
\varphi_k(x) = \varphi_{k_1}(x_1) \ldots \varphi_{k_N}(x_N),
\]
where $k = (k_1, \ldots, k_N) \in \mathbb{N}^N$ is a multi-index and $\varphi_{k_l}$ are univariate Hermite functions as above, $1 \leq l \leq N$. Defining $A_l$ and $A_l^\dagger$ as the one-dimensional ladder operators \[^2\] with respect to the $l$-th coordinate, for $r = (r_1, \ldots, r_N) \in \mathbb{N}^N$, we write

$$A^r = A_1^{r_1} \ldots A_N^{r_N},$$

and $(A^\dagger)^r$ analogously. In particular, the corresponding one-dimensional relations \[^3\] and \[^6\] immediately yield (with $e_i \in \mathbb{N}^N$ being the $l$-th unit vector)

$$\varphi_{k+e_l} = \left( A^\dagger \right)^{e_l} \varphi_k = \frac{1}{\sqrt{k_l+1}} A_l^\dagger \varphi_k, \quad \varphi_{k-e_l} = \frac{1}{\sqrt{k_l}} A_l \varphi_k$$

as higher-dimensional counterparts for the ladder relations as well as adjointness

$$\langle \varphi, (A^\dagger)^r \psi \rangle = \langle A^r \varphi, \psi \rangle, \quad \forall \varphi, \psi \in \mathcal{S}(\mathbb{R}^N).$$

Again, \{$\varphi_k$\}_\{k\in \mathbb{N}^N\} forms a complete $L^2(\mathbb{R}^N)$-orthonormal set of functions. Due to the eigenfunction property \[^4\], we find

$$\frac{1}{2} \sum_{l=1}^{N} \left( p_l^2 + q_l^2 \right) \varphi_k = \frac{1}{2} \left( -\Delta + \sum_{l=1}^{N} q_l^2 \right) \varphi_k = \sum_{l=1}^{N} \left( k_l + \frac{1}{2} \right) \varphi_k,$$

where $q_l$ and $p_l$ denote the position and momentum operators with respect to the $l$-th coordinate, respectively, $1 \leq l \leq N$. For an expansion $\psi(x) = \sum_{k \in \mathbb{N}^N} \hat{\psi}_k \varphi_k(x)$ of $\psi \in L^2(\mathbb{R}^N)$ with $\|A^r \psi\|_{L^2} < \infty$, for every $r \in \mathbb{N}^N$ with $r_l < k_l$, $1 \leq l \leq N$, due to the relations \[^7\] and \[^8\], the coefficients decay as

$$|\hat{\psi}_k| = | \langle \psi, \varphi_k \rangle | = \prod_{l=1}^{N} (k_l(k_l-1) \ldots (k_l-r_l+1))^{-1/2} | \langle \psi, (A^\dagger)^r \varphi_{k-r} \rangle |$$

$$= \prod_{l=1}^{N} (k_l(k_l-1) \ldots (k_l-r_l+1))^{-1/2} | \langle A^r \psi, \varphi_{k-r} \rangle |$$

$$\leq \prod_{l=1}^{N} (k_l(k_l-1) \ldots (k_l-r_l+1))^{-1/2} \|A^r \psi\|_{L^2} = O \left( (k-r)^{-(1/2)r} \right).$$

---

Figure 1: Univariate Hermite functions for some choices of $k$. $\varphi_k$ is even if $k$ is even, otherwise odd. The largest zero is bounded by $\sqrt{2(k+1)}$. 

---

\[^2\] Property (4), we find

\[^3\] Again, \[^6\] respectively, 1

\[^4\] \[^7\] and \[^8\] immediately yield (with $e_i \in \mathbb{N}^N$ being the $l$-th unit vector)

\[^5\] as higher-dimensional counterparts for the ladder relations as well as adjointness

\[^6\] \[^7\] and \[^8\] analogously. In particular, the corresponding one-dimensional relations \[^3\] and \[^6\] immediately yield (with $e_i \in \mathbb{N}^N$ being the $l$-th unit vector)

\[^7\] \[^8\] analogously. In particular, the corresponding one-dimensional relations \[^3\] and \[^6\] immediately yield (with $e_i \in \mathbb{N}^N$ being the $l$-th unit vector)

\[^4\] \[^5\] as higher-dimensional counterparts for the ladder relations as well as adjointness

\[^6\] \[^7\] and \[^8\] analogously. In particular, the corresponding one-dimensional relations \[^3\] and \[^6\] immediately yield (with $e_i \in \mathbb{N}^N$ being the $l$-th unit vector)

\[^7\] \[^8\] analogously. In particular, the corresponding one-dimensional relations \[^3\] and \[^6\] immediately yield (with $e_i \in \mathbb{N}^N$ being the $l$-th unit vector)
1.2 Galerkin ansatz

The Galerkin method determines an approximation function
\[ \psi_\mathcal{K}(x,t) = \sum_{\mathbf{k} \in \mathcal{K}} c_\mathbf{k}(t) \varphi_\mathbf{k}(x) \in \text{span} \{ \varphi_\mathbf{k} \mid \mathbf{k} \in \mathcal{K} \} \subseteq L^2(\mathbb{R}^N) \]  
(10)
on a finite-dimensional subspace such that
\[ \left( i \frac{\partial}{\partial t} \psi_\mathcal{K} - H \psi_\mathcal{K}, \varphi_j \right) = 0, \quad \forall j \in \mathcal{K}, \]  
(11)
where
\[ \mathcal{K} = \mathcal{K}(K) = \{ \mathbf{k} = (k_1, \ldots, k_N) \in \mathbb{N}^N \mid 0 \leq k_i \leq K \} \]  
(12)
is a multi-dimensional index set with \( K+1 \) indices in each direction. Abbreviating \( c(t) = (c_\mathbf{k}(t))_{\mathbf{k} \in \mathcal{K}} \), inserting the ansatz (10) into (11) yields a linear system of ordinary differential equations
\[ i \mathcal{M}_\mathcal{K} \dot{c}(t) = \mathcal{H}_\mathcal{K}(t) c(t). \]

By orthonormality of \( \{ \varphi_\mathbf{k} \}_{\mathbf{k} \in \mathcal{K}} \), \( \mathcal{M}_\mathcal{K} \) reduces to the identity. Furthermore, the eigenfunction relation (9) yields a decomposition (j, \( \mathbf{k} \in \mathcal{K} \))
\[ (\mathcal{H}_\mathcal{K})_{jk} = (\varphi_j, H \varphi_k) = \left( \varphi_j, \frac{1}{2} \left( -\Delta + \sum_{l=1}^N q_l^2 \right) \varphi_k \right) = \varphi_j \left( V - \frac{1}{2} \sum_{l=1}^N q_l^2 \right) \varphi_k \]
\[ = \sum_{l=1}^N \left( k_l + \frac{1}{2} \right) \delta_{jk} + (\varphi_j, W \varphi_k) = (\mathcal{D}_\mathcal{K})_{jk} + (W_\mathcal{K})_{jk}, \]
where \( \mathcal{D}_\mathcal{K} = \text{diag}_{\mathbf{k} \in \mathcal{K}} \left( \sum_{l=1}^N \left( k_l + \frac{1}{2} \right) \right) \) is a diagonal matrix and \( (W_\mathcal{K})_{jk} = (\varphi_j, W \varphi_k) \) stems from a multiplicative potential \( W = W(x,t) = V(x,t) - \frac{1}{2} \sum_{l=1}^N x_l^2 \). Thus, we get the system
\[ i \dot{c}(t) = \mathcal{D}_\mathcal{K} c(t) + W_\mathcal{K}(t) c(t). \]  
(13)

1.3 Approximation of the potential

The fast algorithm as outlined in Section 3 requires the remaining potential \( W \) to be a (multivariate) polynomial. Using a domain \( \Omega = [-L, L]^N \), \( L = \sqrt{2(K+1)} + 1 \) (i.e., \( \psi_\mathcal{K} \) is negligibly small outside \( \Omega \)) , and an index set \( \mathcal{R}(\mathcal{R}) \subseteq \mathbb{N}^N \) as in (12) with \( |\mathcal{R}| \ll |\mathcal{K}| \), we consider Chebyshev interpolation on \( \Omega \) over \( \mathcal{R} \), i.e.,
\[ W(x,t) \approx W_{\text{pol}}(x,t) = \sum_{\mathbf{r} \in \mathcal{R}} \alpha_{\mathbf{r}}(t) T_\mathbf{r}(x/L) = \sum_{\mathbf{r} \in \mathcal{R}} \alpha_{\mathbf{r}}(t) \prod_{l=1}^N T_{r_l}(x_l^N/L) \]
with coefficients
\[ \alpha_{\mathbf{r}}(t) = \gamma_{r_1} \ldots \gamma_{r_N} \sum_{s \in \mathcal{R}} W(Lsz_\mathbf{s}, t) T_\mathbf{r}(z_\mathbf{s}), \quad \gamma_{r_l} = \begin{cases} 1/(R+1), & r_l = 0, \\ 2/(R+1), & 1 \leq r_l \leq R. \end{cases} \]
The univariate functions \( T_{r_l} \) are the Chebyshev polynomials of the first kind that obey the recurrence relation
\[ \begin{align*}
T_0(x) &= 1, \\
T_1(x) &= x, \\
T_{k+1}(x) &= 2xT_k(x) - T_{k-1}, \quad k \geq 1,
\end{align*} \]
x \( \in [-1, 1] \). 
(14)
and \( z_k = (z_{s1}, \ldots, z_{sN}) \) with \( z_{s_l} \) being the zeros of \( T_{R+1} \). The expansion coefficients \( \alpha_r \) decay as \( \alpha_s = O \left( (\max |s_l|)^{-p} \right) \) in case \( W(t, \cdot) \in H^p_\omega(\Omega) \) with \( p > N/2 \), where \( H^p_\omega(\Omega) \) is the weighted Sobolev space relative to the Chebyshev weight \( \omega = \prod_{l=1}^N (1 - x_l^2)^{-1/2} \). The interpolation error estimate with respect to the weighted \( L^2 \)-norm is
\[
\left\| W(\cdot, t) - W^{pol}(\cdot, t) \right\|_{L^2(\Omega)} \leq CR^{-p} |W(\cdot, t)|_{H^p_\omega(\Omega)},
\]
the latter norm being a weighted Sobolev seminorm, see, e.g., Canuto et al. (2006) for a detailed theory of approximation by orthogonal polynomials. In place of (13), this yields a coefficient ODE
\[
i c_{pol}(t) = D_{\mathcal{K}} c_{pol}(t) + W_{\mathcal{K}, pol}(t) c_{pol}(t).
\]

1.4 Gauss-Hermite quadrature

In order to approximate the entries of \( W_{\mathcal{K}, pol}(t) \), we choose Gaussian quadrature for the weight function \( e^{-x^2} \) over \( \mathbb{R} \) in each direction. Let \( \xi_0 < \ldots < \xi_M \) denote the zeros of \( H_{M+1} \), \( M > 0 \). Using weights
\[
w_k = \int_{\mathbb{R}} \prod_{j=0}^M \frac{x - \xi_j}{\xi_k - \xi_j} e^{-x^2} \, dx = \frac{2^M (M + 1)! \pi^{1/2}}{(M + 1)^2 H^2_M(\xi_m)}, \quad 0 \leq m \leq M,
\]
the resulting quadrature formula \((w_m, \xi_m)_{m=0}^M\) has the exactness property
\[
\int_{\mathbb{R}} f(x) e^{-x^2} \, dx = \sum_{m=0}^M w_m f(\xi_m)
\]
if \( f \) is a polynomial of degree \( \leq 2M + 1 \). In higher dimensions, we set
\[
\xi_m = (\xi_{m1}, \ldots, \xi_{mN}), \quad \omega_m = \prod_{l=1}^N \omega_{m_l} = \prod_{l=1}^N w_{m_l} e^{\xi_{m_l}^2}, \quad m \in \mathcal{M},
\]
where \( \mathcal{M} \) is a full \( N \)-dimensional index set of the form \([12]\) with \( M \) instead of \( K \). This yields a product quadrature formula
\[
(W_{\mathcal{K}, pol}(t))_{jk} \approx (W_{\mathcal{K}, pol}^{GH(M)}(t))_{jk} = \sum_{m_1=0}^{M} \ldots \sum_{m_N=0}^{M} W^{pol}(\xi_{m1}, \ldots, \xi_{mN}, t) \prod_{l=1}^N \omega_{m_l} \varphi_{j_l}(\xi_{m_l}) \varphi_{k_l}(\xi_{m_l})
\]
which is exact if \( W^{pol}(\cdot, t) H_j H_k \) is a polynomial of degree \( \leq 2M + 1 \) in each direction, where \( H_k(x) = \prod_{l=1}^N H_{k_l}(x_l) \) is a product of univariate Hermite polynomials. In Section 3 deriving the fast algorithm, we shall motivate the suitable choice of Gauss-Hermite quadrature with \( M = K \). We end up with the spatially discretized ODE system (omitting time-dependence in the vectors)
\[
d^{\mathcal{K}}_{\mathcal{K}} = D_{\mathcal{K}}^C + W_{\mathcal{K}, pol}^{GH(K)}(t) c_{pol}^{GH(K)}.
\]

1.5 Curse of dimensionality, reduced grids

In case of \( \mathcal{K} \) being a full index set as in [12], the system \([16]\) consists of \( |\mathcal{K}| = K^N \) equations. For growing \( N \) and \( K \) being only moderate, this is not feasible for time integration that requires, done naively, assembling the matrices \( D_{\mathcal{K}} \) (once) and \( W_{\mathcal{K}, pol}^{GH(K)}(t) \) (in each step) and multiplying them
with a vector. Thus, we replace $\mathcal{K}$ with a reduced index set $\mathcal{K}_s$ ("sparse"), where $|\mathcal{K}_s| \ll |\mathcal{K}|$. We study a hyperbolic cross

$$\mathcal{K}_s = \left\{ \mathbf{k} = (k_1, \ldots, k_N) \mid k_l \geq 0, \prod_{l=1}^{N} (1 + k_l) \leq K + 1 \right\},$$

see the illustration in Figure 2. The number of indices employed reduces to

$$|\mathcal{K}_s| = \mathcal{O}(K \ln(K)^{N-1}),$$

see Bungartz & Griebel (2004). As explained in Section 5, approximating a function $f \in L^2(\mathbb{R}^N)$ by a Hermite tensor-product expansion using only indices from $\mathcal{K}_s$ still gives a decent approximation, i.e., hyperbolic crosses preserve favorable convergence properties known as spectral convergence. The index set $\mathcal{R}$ for the Chebyshev nodes might also by reduced.

![Hyperbolic Cross Illustration](image)

**Figure 2:** Left: Hyperbolic cross for $N = 2$ and $K = 32$. Right: Hyperbolic cross for $N = 3$ and $K = 16$.

## 2 Discretization in time

We consider equations of the general form

$$i\dot{y}(t) = A(t)y(t) \quad (17)$$

with a time-dependent matrix $A(t)$. There are at least two general strategies to discretize (17), both amounting to the task of discretizing a matrix exponential: splitting procedures and Magnus integrators. In the present paper, we restrict our attention to the latter choice. See the review Blanes et al. (2009), in particular, Sections 5 and 6, for numerical integration methods based on Magnus expansions.

### 2.1 Magnus integrators

Using Magnus integrators, one approximates the solution of an equation (17) by an exponential stepping procedure of the form

$$y^{n+1} = e^{\Omega^n} y^n, \quad (18)$$

where $y^n \approx y(t_n)$, $t_n = hn$ with time-step size $h$, for a suitable choice of $\Omega^n$. Possible choices are the exponential mid-point rule

$$\Omega^n = -ihA(t^n + h/2) \quad (19)$$
or the method based on the 2-stage Gauss-Legendre quadrature with nodes $c_{1,2} = \frac{1}{2} \pm \frac{\sqrt{3}}{6}$,

$$
\Omega^n = -\frac{i}{2} h(A_1 + A_2) - \frac{\sqrt{3}}{12} h^2 [A_2, A_1],
$$

(20)

where $A_j = A(t^n + c_j h)$, $j = 1, 2$, and $[·, ·]$ denotes the commutator of matrices.

In our setting, we have

$$
A(t) = H_{K_{s, pol}}^{GH(K)}(t) = D_K + W_{K_{s, pol}}^{GH(K)}(t).
$$

Hochbruck & Lubich (2003) show that the methods (19) and (20) are of optimal temporal orders 2 and 4, respectively, for the Schrödinger equation with a bounded potential.

2.2 Lanczos method for the matrix exponential

We apply the Lanczos method in order to approximate the exponential in (18). See Lubich (2008), Chapter III.2.2, for a more detailed outline including further references. Consider a general initial value problem

$$
i y(t) = Ay(t)
$$

with an $n \times n$ Hermitian matrix $A$ and $y(0) = y_0$. Using Gram-Schmidt orthogonalization, an orthonormal basis $\{v_k\}_{k=1}^m$ of the $m$-th Krylov subspace

$$
K_m(A, y_0) = \text{span} \{y_0, Ay_0, \ldots, A^{m-1} y_0\} \subseteq \mathbb{C}^n
$$

with respect to $A$ and $y_0$ is constructed by successive orthogonalization and normalization, i.e.,

$$
\tau_{k+1,k} v_{k+1} = Av_k - \sum_{j=1}^k \tau_{jk} v_j,
$$

(21)

with $\tau_{jk} = v_j^T Av_k$ for $j \leq k$, and $\tau_{k+1,k} > 0$ being a normalization parameter. Thus, the Hermitian Lanczos process generates recursively the basis $V_m = (v_1 \mid \ldots \mid v_m) \in \mathbb{C}^{n \times m}$ and a tridiagonal coefficient matrix $T_m \in \mathbb{C}^{m \times m}$ such that

$$
T_m = V_m^* A V_m.
$$

(22)

This requires $m$ multiplications of $A$ on a vector, where $m \ll n$. For an algorithmic description (without reorthogonalization), see, e.g., Lubich (2008), Algorithm III.2.5. By a Galerkin ansatz on $K_m(A, y_0)$, the matrices $V_m$ and $T_m$ are used to approximate

$$
y(t) = e^{-itA} y_0 \approx V_m e^{-itT_m} e_1.
$$

(23)

In our setting, $-ihA = \Omega^n$, $y_0 = y^n$. In each time step $[18]$, for all specific choices of $\Omega^n$, this involves the action of $W_{K_{s, pol}}^{GH(K)}(t)$ on vectors $v_k$, evaluated at times $t$ depending on the chosen Magnus integrator.

3 Fast algorithm

We consider the product of $W_{K_{s, pol}}^{GH(M)}(t)v$, see [16], times a vector $v \in \mathbb{C}^{|K|}$ on a full grid $K$. 

8
3.1 Auxiliary matrices for full index sets

First, for each direction, we define auxiliary matrices

\[ X^{(l)}, \quad 1 \leq l \leq N, \quad (X^{(l)})_{jk} = (\varphi_j, Q_l \varphi_k), \quad j, k \in K. \]

The following considerations relate Gauss-Hermite quadrature with \( M = K \) for the entries of \( W^{\text{GH}(M)}_{\text{K,pol}} \) and formal insertion of \( X^{(l)} \) into \( W^{\text{pol}} \). The matrices

\[ \Xi^{(l)} = \text{diag}_{k \in K}(\xi_{m_l}) \in \mathbb{R}^{|K| \times |K|}, \quad U_{jk} = \sqrt{\omega_j} \varphi_k(\xi_j), \quad j, k \in K, \]

with \( U \) being independent of \( l \), yield a diagonalization

\[ X^{(l)} = U^T \Xi^{(l)} U \in \mathbb{R}^{|K| \times |K|}, \]

which is readily seen from

\[ (U^T \Xi^{(l)} U)_{jk} = \sum_{m \in K} \omega_m \xi_{m_l} \varphi_j(\xi_m) \varphi_k(\xi_m) = (\varphi_j, Q_l \varphi_k)^{\text{GH}(K)} = (\varphi_j, Q_l \varphi_k) = (X^{(l)})_{jk}, \quad (24) \]

by the fact that there are exactly \( K + 1 \) quadrature nodes in each direction and that this yields an exact integration. The matrix \( U \) is unitary, which follows from orthonormality of the basis and

\[ (U^T U)_{jk} = \sum_{m \in K} u_{mj} u_{mk} = \sum_{m \in K} \omega_m \varphi_j(\xi_m) \varphi_k(\xi_m) = (\varphi_j, \varphi_k)^{\text{GH}(K)} = (\varphi_j, \varphi_k) = \delta_{jk}. \]

This allows to compute

\[ X^r = (X^{(1)})^{r_1} \ldots (X^{(N)})^{r_N} = (U^T \text{diag}(\xi_{m_1}^{r_1})) U \ldots (U^T \text{diag}(\xi_{m_N}^{r_N})) U = U^T \text{diag}(\xi^r_m) U, \quad (25) \]

and we get the following

**Lemma 1** Choosing \( M = K \) for the quadrature and basis grids, respectively, we get

\[ W^{\text{pol}}(X, t)_{jk} = (W^{\text{GH}(K)}_{\text{K,pol}}(t))_{jk}, \quad j, k \in K, \]

where \( W^{\text{pol}}(X, t) \) denotes formal insertion of \( X \) into \( W^{\text{pol}} \) according to \( (25) \).

This result is commonly used in DVR techniques, see [Light & Carrington (2000)]. The ordering of the factors \( (X^{(l)})^{r_l} \) in \( W^{\text{pol}}(X, t) \) is arbitrary.

3.2 Fast algorithm

Due to the orthonormality of the basis and with the help of the one-dimensional recurrence relation \( (5) \), the action of \( X^{(l)} \) on a vector \( v \in \mathbb{C}^{|K|} \) is given by

\[
(X^{(l)} v)_j = \sum_{k \in K} (\varphi_j, Q_l \varphi_k) v_k
= \sum_{k \in K} (\varphi_j, \sqrt{\frac{k_l + 1}{2}} \varphi_{k + e_l} + \sqrt{\frac{k_l}{2}} \varphi_{k - e_l}) v_k
= \sum_{k \in K} \sqrt{\frac{k_l + 1}{2}} (\varphi_j, \varphi_{k + e_l}) v_k + \sum_{k \in K} \sqrt{\frac{k_l}{2}} (\varphi_j, \varphi_{k - e_l}) v_k
= \begin{cases} 
\sqrt{\frac{k_l}{2}} v_j + \sqrt{\frac{k_l + 1}{2}} v_{j+1}, & 1 \leq j \leq K - 1, \\
\sqrt{\frac{k_l}{2}} v_{j+1}, & j = 0, \\
\sqrt{\frac{k_l}{2}} v_j, & j = K.
\end{cases}
\]
The matrix-vector-product $X^{(l)}v$ can thus be computed directly using $O(|\mathcal{K}|)$ operations. By virtue of Lemma 1, the action of the quadrature matrix $W_{\mathcal{K},\text{pol}}^{GH(K)}(t)v$ is best computed using Horner’s method

$$W_{\mathcal{K},\text{pol}}^{GH(K)}(t)v = W^{\text{pol}}(X, t)v = \sum_{r \in \mathcal{R}} \alpha_r(t) \left( \prod_{l=1}^{N} T_{\tau_l} \left( \frac{1}{L} X^{(l)} \right) \right) v$$

$$= \sum_{r \in \mathcal{R}} \alpha_r(t) \left( T_{\tau_1} \left( \frac{1}{L} X^{(1)} \right) : \ldots : \left( T_{\tau_N} \left( \frac{1}{L} X^{(N)} \right) v \ldots \right),$$

where $T_{\tau_l} \left( \frac{1}{L} X^{(l)} \right) v$ is computed recursively with the help of [14].

### 3.3 On reduced index sets

Due to $\mathcal{K}$ growing exponentially in $N$, the fast algorithm on a full grid $\mathcal{K}$ is still prohibitively expensive. Let $\mathcal{K}_s$ denote an arbitrary reduced grid with $|\mathcal{K}_s| \ll |\mathcal{K}|$. The derivation of $W_{\mathcal{K},\text{pol}}^{GH(K)} = W^{\text{pol}}(X, t)$ requires a bijection $\mathcal{M} \leftrightarrow \mathcal{K}$ with $\mathcal{K}$ being chosen sufficiently large in order to guarantee exactness of quadrature. Simultaneously reducing $\mathcal{M}$ and $\mathcal{K}$ invalidates the exactness of the Gauss-Hermite quadrature, reducing only $\mathcal{K}$ makes the above diagonalization argument no longer correct at all. For a reduced grid $\mathcal{K}_s$, an assertion analogous to Lemma 1 can therefore not be expected. We define

$$\Omega_s : \mathbb{C}^{|\mathcal{K}| \times |\mathcal{K}|} \rightarrow \mathbb{C}^{|\mathcal{K}_s| \times |\mathcal{K}_s|}, \quad \Omega_s(A) = (A_{jk})_{j,k \in \mathcal{K}_s}$$

(26)

to be the operator that cuts a fully indexed matrix to a reduced index set and employ the above fast algorithm with the reduced auxiliary matrices

$$X^{(l)}_s = \Omega_s(X^{(l)})$$

applied to $v \in \mathbb{C}^{|\mathcal{K}_s|}$.

### 3.4 Computational complexity

We compare the naive approach, i.e., assembling $W_{\mathcal{K},\text{pol}}^{GH(M)}(t)$ and multiplying with a vector $v \in \mathbb{C}^{|\mathcal{K}_s|}$, to the direct approach due to the fast algorithm.

#### 3.4.1 Assembling the matrix

If the matrix $W_{\mathcal{K},\text{pol}}^{GH(M)}(t)$ is already given, the computation of $W_{\mathcal{K},\text{pol}}^{GH(M)}(t)v$ is done in $O(|\mathcal{K}_s|^2)$ operations. The computational bulk lies in assembling the matrix itself: In one dimension, we consider [5] in tail-recursive form, i.e., we compute successively

$$(\varphi_0, \varphi_1) \rightarrow \varphi_2, \quad (\varphi_1, \varphi_2) \rightarrow \varphi_3, \quad \ldots, (\varphi_{K-2}, \varphi_{K-1}) \rightarrow \varphi_K,$$

obtaining $\varphi_K(x)$ for fixed $x \in \mathbb{R}$ in $O(K)$ operations. Given weights and nodes $(\xi_m, \omega_m)_{m=0}^M$, the values $\varphi_k(\xi_m)$, $0 \leq k \leq K$, $0 \leq m \leq M$, are thus computed in $O(KM)$ operations. The values $T_{\tau}(\xi_m/L)$, $0 \leq r \leq R$, $0 \leq m \leq M$, are computed in $O(RM)$ operations. Given $\varphi_k(\xi_m)$ and $T_{\tau}(\xi_m/L)$, one obtains the multi-dimensional quadrature formulas

$$\sum_{m \in \mathcal{M}} \omega_m \varphi_j(\xi_m) W^{\text{pol}}(\xi_m, t) \varphi_k(\xi_m) = \sum_{r \in \mathcal{R}} \alpha_r(t) N \prod_{l=1}^{M} \sum_{m_l=0}^{M} (\omega_{m_l} \varphi_j(\xi_{m_l}) T_{\tau_l}(\xi_{m_l}/L) \varphi_k(\xi_{m_l}))$$

10
in $\mathcal{O}(|\mathcal{R}| \cdot |\mathcal{M}| \cdot N)$ operations, for every $j, k \in \mathcal{K}_s$, where we assume the term $\star$ to be computable in $\mathcal{O}(1)$ operations. Using a full quadrature index set $\mathcal{M}$, assembling the reduced matrix $W^{GH(K)}_{\mathcal{K}_s,\text{pol}}$ thus requires

$$\mathcal{O}\left(|\mathcal{K}_s|^2 \cdot |\mathcal{M}| \cdot |\mathcal{R}| \cdot N\right)$$

operations. As Lemma 1 requires $\mathcal{M} \leftrightarrow \mathcal{K}$, this approach is prohibitively expensive.

### 3.4.2 Direct computation using the fast algorithm

The fast algorithm on a reduced grid, in contrast, scales much more favorably, as explained in the following table:

| Direct computation | $\mathcal{O}(\cdot)$ |
|--------------------|-----------------------|
| $X^{(l)} v$ for fixed $l$ | $\mathcal{O}(|\mathcal{K}_s|)$ |
| $T_{l_1} \left(\frac{1}{l} X^{(l)}\right) v$ using (14) | $\mathcal{O}(r_1|\mathcal{K}_s|)$ |
| $\left(\prod_{i=1}^{N} T_{l_i} \left(\frac{1}{l} X^{(l)}\right)\right) v$ using Horner’s scheme | $\mathcal{O}(N R|\mathcal{K}_s|)$ |
| $W^{\text{pol}}(X^{(s)}, t)^{\text{fast}} v$ | $\mathcal{O}\left(|\mathcal{R}| N R|\mathcal{K}_s|\right)$ |

Because of $R \ll K$, in case of $W$ being time-dependent, the costs for re-computing the coefficients of the interpolation polynomial in each step are negligible.

### 3.4.3 Experimental comparison

In Figure 3, we compare assembling $W^{GH(K)}_{\mathcal{K}_s,\text{pol}}$ to a direct computation of $\left(W^{\text{pol}}(X) v\right)^{\text{fast}}$ with respect to CPU time for a (time-independent) stretched torsional potential

$$W(x) = \sum_{l=1}^{N} \left(1 - \cos(x_l/L)\right), \quad x \in \Omega, \quad (27)$$

as approximated by Chebyshev interpolation with $R = 8$ (yielding an interpolation error of size $\approx 1e-10$) and give computation times for some choices of $N$ and $K$. As the figures reveal, on a hyperbolically reduced grid, the fast algorithm lowers the computational effort by several orders of magnitude for reasonable choices of $K$. The larger $K$, the better the reduction (for fixed $N$). For the case of a full grid $\mathcal{K}$, the task is barely tractable: assembling the fully indexed matrix $W^{GH(K)}_{\mathcal{K},\text{pol}}$ and multiplying it with a random vector $v \in \mathbb{R}^{|\mathcal{K}|}$ takes $4.301e+03 \text{ secs} \approx 72 \text{ min}$ in case $N = 2$, $K = 60$, and $1.265e+05 \text{ secs} \approx 35 \text{ hrs}$ in case $N = 3$, $K = 20$. All figures have been obtained with a FORTRAN 95 implementation on an Intel Core 2 Duo E8400 3.00 GHz processor with 4 GB RAM in double precision arithmetics.

### 4 Algorithmic description

Start from given

- reduced index set $\mathcal{K}_s = \mathcal{K}_s(K)$ for the Hermite basis, determining the spatial accuracy,
- (full or reduced) index set $\mathcal{R} = \mathcal{R}(R)$ for the polynomial approximation of the potential $W$, $R \ll K$, 


coefficient vector $c_{\text{pol}}^{GH(K):n} = c(0)$, $\|c(0)\|_2 = 1$, with $c(0)$ obeying a decay condition as given below, see (30).

- time-step size $h$, and

- number $m$ of Lanczos steps in each time step.

for $n = 0, \ldots$, do the following:

(1) Compute the coefficients $\alpha_r(t)$ of the approximation

$$W(x,t) = V(x,t) - \frac{1}{2} \sum_{l=1}^{N} x_l^2 \approx \sum_{r \in \mathcal{R}} \alpha_r(t) T_r(x/L) = W_{\text{pol}}(x,t)$$

with $L = \sqrt{2(K+1)} + 1$, to be evaluated as prescribed by the chosen Magnus integrator.

(2) Do $m$ Lanczos steps to obtain matrices $V_m^{(n)}$ and $T_m^{(n)}$ starting from $c_{\text{pol}}^{GH(K):n}$.

In each step, for the action of $\Omega^n$ on the Lanczos basis vectors, use

$$\langle D_{\mathcal{K}_s} v \rangle_j = \sum_{l=1}^{N} (jl + \frac{1}{2}) v_j,$$

$$W_{\mathcal{K}_s,\text{pol}}^{GH(K)} v : \text{fast algorithm},$$

instead of assembling the matrices and doing matrix-vector-multiplication.

(3) Compute $c_{\text{pol}}^{GH(K):n+1} = V_m^{(n)} e^{-ihT_m^{(n)}} e_1$.

end

Step (3) is done using a (small) diagonalization of $T_m^{(n)}$, and the product $V_m^{(n)}$ times a vector is computed in $\mathcal{O}(|\mathcal{K}_s|m^2)$.

## 5 Error analysis

### 5.1 Preliminaries

**Definition of errors:** Consider an arbitrary vector $v \in \mathcal{C}^{[\mathcal{K}_s]}$. We are interested in computing the product $W_{\mathcal{K}_s,\text{pol}}(t)v$ with a matrix $W_{\mathcal{K}_s,\text{pol}}$ as given in Section 1.3. The fast algorithm as developed

| $N$ | $K$ | $(1) W_{\mathcal{K}_s,\text{pol}}^{GH(K)} v$ | $(2) (W_{\text{pol}}^{\text{fast}}(x)v)$ | $\approx \text{ratio } (1)/(2)$ |
|-----|-----|-----------------|-------------------------------|-----------------|
| 2   | 20  | $5.489e-01$     | $3.518e-03$                  | $1.56e+02$      |
|     | 40  | $5.678e+00$     | $7.064e-03$                  | $8.04e+02$      |
|     | 60  | $2.146e+01$     | $1.159e-02$                  | $1.85e+03$      |
|     | 80  | $5.878e+01$     | $1.127e-02$                  | $5.22e+03$      |
|     | 100 | $1.264e+02 \approx 2.1 \text{ min}$ | $2.108e-02$                  | $5.97e+03$      |
| 3   | 20  | $3.818e+01$     | $9.790e-02$                  | $3.90e+02$      |
|     | 40  | $4.918e+02 \approx 8 \text{ min}$ | $2.456e-01$                  | $2.00e+03$      |
|     | 60  | $2.332e+03 \approx 39 \text{ min}$ | $4.373e-01$                  | $5.33e+03$      |
|     | 80  | $6.994e+03 \approx 1.9 \text{ hrs}$ | $6.475e-01$                  | $1.08e+04$      |
|     | 100 | $1.571e+04 \approx 4.4 \text{ hrs}$ | $8.697e-01$                  | $1.81e+04$      |
in Section 3 gives rise to an error due to quadrature and to an error due to grid reduction, the former being given by

$$E_{\text{quad}} = \{(E_{j,k})_{j,k} \in \mathcal{K}_s \},$$

$$E_{j,k} = (W_{\mathcal{K}_s,\text{pol}}(t))_{j,k} - (W_{\mathcal{K}_s,\text{pol}}^{GH(K)}(t))_{j,k}. \quad (28)$$

Formally inserting the hyperbolically reduced auxiliary matrices into the polynomial yields an error

$$W^{\text{pol}}(X_s,t)v - W^{\text{GH(K)}}_{\mathcal{K}_s,\text{pol}}(t)v = \left[ W^{\text{pol}}(X_s,t)v - \Omega_s(W^{\text{pol}}(X,t))v \right]$$

$$+ \left[ \left( \Omega_s(W^{\text{pol}}(X,t)) - W^{\text{GH(K)}}_{\mathcal{K}_s,\text{pol}}(t) \right) \right] v.$$

The difference $\ast$ vanishes by virtue of Lemma 1. One easily verifies

$$W^{\text{pol}}(X_s,t)v - \Omega_s(W^{\text{pol}}(X,t))v = W^{\text{pol}}(X_s,t)v - \Omega_s\left( W^{\text{pol}}(X,t)\Omega_+(v) \right),$$

where

$$\Omega_+ : \mathbb{C}^{|\mathcal{K}|} \to \mathbb{C}^{|\mathcal{K}|}, \quad (\Omega_+(v))_j = \begin{cases} v_j, & j \in \mathcal{K}_s, \\ 0, & j \notin \mathcal{K}_s \end{cases}$$

is the function that blows up a hyperbolically indexed vector with zeros at missing indices and $\Omega_s$ is defined as in (26). Hence, the error due to grid reduction is given by

$$E^{\text{red}}(v) = (E_j)_{j \in \mathcal{K}_s},$$

$$E_j = \left( W^{\text{pol}}(X_s,t)v - \Omega_s\left( W^{\text{pol}}(X,t)\Omega_+(v) \right) \right)_j. \quad (29)$$

Assumption: For the following error analysis, we make the general decay assumption

$$v_k = \mathcal{O}\left(k^{-\beta}\right) = \mathcal{O}\left(\prod_{l=1, k_l \neq 0}^{N} k_l^{-\beta}\right), \quad k \in \mathcal{K}_s,$$

for the vector coefficients of $v$, with some $\beta \in \mathbb{N}$. The larger the index, the faster the decay. Assumption (30) is used in Sections 5.2 and 5.3 to compensate large error components in matrix-vector-products.

5.2 Error $E^{\text{quad}}$ due to quadrature

**Theorem 1** Let $W^{\text{pol}}(\cdot, t) \approx W(\cdot, t)$ be the Chebyshev interpolation polynomial of the potential $W$ on $\Omega = [-L, L]^N$ over $\mathcal{R}(R)$ with $L = \sqrt{2(K + 1)} + 1$ and $\mathcal{K}_s = \mathcal{K}_s(K)$ a hyperbolically reduced index set with $K \gg R$. Then, under assumption (30) on $v \in \mathbb{C}^{|\mathcal{K}_s|}$ (i.e., componentwise decay of order $\beta \in \mathbb{N}$), the error due to quadrature behaves as

$$\left| \left( \left( W_{\mathcal{K}_s,\text{pol}}(t) - W_{\mathcal{K}_s,\text{pol}}^{GH(K)}(t) \right) v \right) \right| \leq C(\mathcal{R}, W) K^{-\beta}, \quad j \in \mathcal{K}_s,$$

where the matrices $W_{\mathcal{K}_s,\text{pol}}(t)$ and $W_{\mathcal{K}_s,\text{pol}}^{GH(K)}(t)$ are defined according to Sections 1.2 and 1.3, respectively. The constant $C(\mathcal{R}, W)$ is given as in (39), see below, and depends on $\mathcal{K}$ and the regularity of $W$ only.
Proof: First, we decompose the interpolation polynomial

\[ W_{\text{pol}}^r(x,t) = \sum_{r \in \mathbb{R}} \alpha_r(t)W_{\text{pol}}^r(x), \quad W_{\text{pol}}^r(x) = (x/L)^r, \]

and consider each term separately, giving rise to errors

\[ E^r = \left( E_{j,k}^r \right)_{j,k \in K_+}, \quad E_{j,k}^r = (\varphi_{j,r}(x/L)^r\varphi_k)_{GH(K)} . \]

Summing up, we get

\[ E^{\text{quad}} = \sum_{r \in \mathbb{R}} \alpha_r(t)E^r(x). \quad (31) \]

For given \( r \in \mathbb{R} \), we define

\[ r_{\text{max}} = r_{\text{max}}(r) = \max_{1 \leq i \leq N} r_i. \]

Error matrix and corresponding binary tree (1D): In one dimension, the error \( E_{j,k}^r \) does not vanish iff \( r + j + k \geq 2K + 2 \), thus, the error matrix given by (28) has the structure

with \((K + r - 2) - j + 1\) non-vanishing entries in row \( j \), for \( K - r + 2 \leq j \leq K \).

We use the recurrence relation \([5]\) with the term \((x/L)^r\varphi_k\) to reduce powers of \( x \) completely according to the following binary tree pattern:

This converts \( E_{j,k}^r \) into a full binary tree \( \mathcal{T} \) of height \( r \), each node carrying a difference term

\[ \left( \varphi_{j,r}(x/L)^r\varphi_k - \sum_{m=0}^{K} \omega_m \varphi_j(\xi_m/L)^{r-\lambda}\varphi_{k+\lambda}(\xi_m) \right) . \]

where \( \lambda \) and \( \rho \) are the numbers of left or right descents along the path connecting the node to the root, respectively. Descending left does not alter the polynomial degree of the integrand, descending right reduces it by 2. Our strategy is to examine
• which leaves do not vanish in $\mathcal{T}$,
• how many these are, and
• what quantity they sum up to.

In case $r > k$, we may define $\varphi_k(x) = 0$ for $k \leq 1$, preserving the recurrence relation (5) for negative indices.

**Characterization of non-vanishing leaves:** Due to $\lambda + \rho = r$, the condition for a non-vanishing quadrature error at a particular leaf is

$$j + k + \lambda - \rho = j + k + r - 2\rho \geq 2K + 2.$$  

We define

$$\rho_{\text{max}} = \rho_{\text{max}}(j, k, r) = \left\lfloor \frac{j + k + r - 2K - 2}{2} \right\rfloor_+,$$

to be the maximal number of right descents that does not reduce the polynomial degree of the integrand sufficiently for exact quadrature, where $a_+ = a$ if $a \geq 0$, otherwise $a_+ = 0$. At non-vanishing leaves, the exact integral vanishes nevertheless, because the assumption $j = k - \rho + \lambda$ leads to the contradiction

$$2K + 2 \leq j + r + k - 2\rho = j + k + \lambda - \rho = 2j.$$  

**Number of non-vanishing leaves:** In an arbitrary full binary tree of height $r$, the number of leaves connected to the root by a path containing exactly $s$ right descents equals $\left( \begin{array}{c} r \\hfill \cr s \quad \hfill \end{array} \right)$. To see this, consider the following picture, showing a full binary tree of height $r$ with its left and right subtrees (of height $r - 1$ each) attached to its root:

Clearly, all relevant trees are those being connected by $s$ right descents in the left and $s - 1$ right descents in the right subtree, thus, our statement is just a reformulation of the binomial recursion formula

$$\left( \begin{array}{c} r \hfill \cr s \hfill \end{array} \right) = \left( \begin{array}{c} r - 1 \hfill \cr s \hfill \end{array} \right) + \left( \begin{array}{c} r - 1 \hfill \cr s - 1 \hfill \end{array} \right).$$

Hence, the number of non-vanishing leaves in $\mathcal{T}$ is given by

$$a(j, k, r) = \sum_{s=0}^{\rho_{\text{max}}} \left( \begin{array}{c} r \hfill \cr s \hfill \end{array} \right).$$

For a closer investigation of $a(j, k, r)$, consider the general sum

$$\sum_{s=a}^{b} \left( \begin{array}{c} c \hfill \cr s \hfill \end{array} \right).$$
with \(a, b, c \in \mathbb{N}\) and \(2b < c\). Using \(s \leq b\), we compute
\[
\binom{c}{s}/\binom{c}{b} = \frac{b!(c-b)!}{s!(c-s)!} = \frac{b \cdot \ldots \cdot (s+1)}{(c-s) \cdot \ldots \cdot (c-b+1)} < \left(\frac{b}{c-b+1}\right)^{b-s}.
\]

The assumption \(c > 2b\) yields
\[
\frac{2b}{c} = \frac{b}{b-c+1} \cdot \frac{2(c-b+1)}{c} > \frac{b}{b-c+1} \cdot \frac{c+2}{c} > \frac{b}{b-c+1}.
\]
Together with the geometric progression, this gives
\[
\sum_{s=a}^{b} \binom{c}{s} / \binom{c}{b} < \left(\frac{c}{b}\right)^{b-a} < \left(1 - \frac{2b}{c}\right)^{-1} \left(1 - \frac{2b}{c}\right)^{a}.
\]\nIn our particular case, by definition, we have \(2 \rho_{\max} < r\), thus,
\[
a(j,k,r) < \left(\frac{r}{\rho_{\max}}\right) \left(1 - \frac{2 \rho_{\max}}{r}\right)^{-1} = \left(\frac{r}{\rho_{\max}}\right) \frac{r}{r - 2 \rho_{\max}}.
\]"
Multiplying the matrix with a vector \( v \in \mathbb{C}^K \) that decays rapidly according to (30), we find

\[
(E^r v)_j = \sum_{K-r+2 \leq k \leq K, k+r+j \geq 2K+2} E^r_{jk} v_k = \mathcal{O} \left( \frac{(j-K+r-1)}{r/2} \right) r^{2-r} K v_{K-r+2} \]

\[
= \mathcal{O} \left( \left( \frac{r}{|r/2|} \right) r^{2-\beta} (K-r+2)^{-\beta} \right),
\]

the term \( \star \) being of size \( \mathcal{O}(r) \).

**Decomposition of error in multiple dimensions:** We set \( \mathcal{N} = \{1, \ldots, N\} \) and consider the error matrix for \( N \geq 2 \). On a full grid, one has \( \mathcal{O}(r^{2N}) \) non-vanishing entries. On a hyperbolically reduced grid, this number shrinks drastically. If \( E^r_{jk} \) vanishes, for all \( l \in \mathcal{N} \), then \( E^r_{vk} \) vanishes, for all \( j, k \in \mathcal{K} \) from an arbitrary index set. Fix \( j \). Thus, if \( k_1 \leq K - r_1 + 1 \), for all \( l \in \mathcal{N} \), then \( E^r_{jk} \) vanishes. Conversely, for a non-vanishing error \( E^r_{jk} \), there is a subset of components \( \hat{\mathcal{N}} = \hat{\mathcal{N}}(k) \subseteq \mathcal{N} \) such that, for every \( l \in \hat{\mathcal{N}} \), \( k_1 \geq K - r_1 + 2 \), and \( E^r_{jk} \) does not vanish. This allows for a decomposition

\[
E^r_{jk} = (\varphi_j, (x/L)^r \varphi_k) - (\varphi_j, (x/L)^r \varphi_k)^{GH(K)}
\]

\[
= \left[ \prod_{l=1}^N (\varphi_{j_l}, (x/L)^{r_l} \varphi_{k_l}) \right] - \left[ \prod_{l=1}^N (\varphi_{j_l}, (x/L)^{r_l} \varphi_{k_l})^{GH(K)} \right]
\]

\[
= \left[ \prod_{l \in \tilde{\mathcal{N}}} (\varphi_{j_l}, (x/L)^{r_l} \varphi_{k_l}) \right] - \left[ \prod_{l \in \tilde{\mathcal{N}}} (\varphi_{j_l}, (x/L)^{r_l} \varphi_{k_l})^{GH(K)} \right] \cdot \left[ \prod_{l \notin \tilde{\mathcal{N}}} (\varphi_{j_l}, (x/L)^{r_l} \varphi_{k_l}) \right]
\]

of a non-vanishing entry \( E^r_{jk} \). On a hyperbolic cross \( K_{s} \), non-vanishing errors \( E^r_{jk} \) have indices \( k \) satisfying

\[
\left( \prod_{l \notin \mathcal{N}} (k_l + 1) \right) \left( \prod_{l \in \mathcal{N}} (K - r_l + 3) \right) \leq K + 1.
\]

Clearly, for every \( k \in \mathcal{K}_s \), if \( K \gg r_{\text{max}} \), then \( |\hat{\mathcal{N}}(k)| \leq 1 \), hence, there is at most one such component \( l_0 \). In that case, the terms \( A \) and \( B \) consist of exactly one factor each, and \( A - B \) equals the one-dimensional quadrature error \( E_{j_0,k_0}^r \). Figure 5 shows the structure of \( (E^r_{jk})_{jk \in \mathcal{K}_s} \) for \( N = 2 \) with a lexicographical ordering of the multi-indices. First, consider the blocks corresponding to \( j_1 + k_1 + r_1 \leq 2K + 1 \), where quadrature with respect to the first coordinate is exact. The left upper \((0,0)\)-block represents \( j_1 = k_1 = 0 \). Due to (36), its entries are given by

\[
E^r_{(0,j_2),(0,k_2)} = E^r_{j_2,k_2} \cdot (\varphi_0, (x/L)^{r_l} \varphi_0).
\]

If \( r_1 \) is odd, these terms vanish. Due to the definition of \( \mathcal{K}_s \), the other blocks have a reduced range of \( j_2 \) and \( k_2 \), respectively. If \( K + \frac{K+1}{2} - 1 + r_2 \leq 2K + 1 \), thus, \( r_2 \leq \frac{K-1}{2} + 2 \), every \((1,0)\)- or \((0,1)\)-entry vanishes, as follows from the exactness properties of the chosen quadrature formula. If the \((0,1)\)-block contains only vanishing errors, any \((u,w)\)-block with \( u + w \geq 2 \) and exact quadrature with respect to the first coordinate also has only vanishing entries. Second, consider the blocks corresponding to \( j_1 + k_1 + r_1 \geq 2K + 2 \). If \( K \) is sufficiently large, due to (37), only \( j_2 = k_2 = 0 \) is
possible. Using \([36]\), we have

\[
E_{jk}^r = E_{j_1,k_1}^{r_1} \cdot (\varphi_{j_2}, (x_2/L)^r \varphi_{k_2}) = E_{j_1,k_1}^{r_1} \cdot (\varphi_0, (x_2/L)^r \varphi_0),
\]

which vanishes if \(r_2\) is odd. Therefore, if \(r_1\) and \(r_2\) are odd, even a moderate choice of \(K\) makes the error matrix vanish altogether.

In case \(N \geq 3\), the requirement \(K \gg r_{\text{max}}\) together with the decomposition \([36]\) and \(r_1\) chosen odd allow for only a single \(r_1, l > 1\), to be chosen odd in order to make the whole matrix vanish. If \(K\) is not sufficiently large or if \(r\) does not meet the required parity conditions for the matrix to vanish, the error can nevertheless be analyzed by the following reduction to the one-dimensional proceeding.

**Error estimation in multiple dimensions:** Consider a non-vanishing error term \(E_{jk}^r\) and suppose the whole error matrix does not vanish due to parity reasons. By the above considerations for the one-dimensional case, the term \(A\) vanishes. Due to \(|N(k)| = 1\), there is \(l_0 \in N\) such that \(B = E_{j_0 k_0}^{r_0}\). For a factor in \(C\), using \(x_l \in [-L, L]\) and symmetry of Gauss-Hermite nodes, we find

\[
(\varphi_{j_l}, (x_l/L)^{r_l} \varphi_{k_l}) = (\varphi_{j_l}, (x_l/L)^{r_l} \varphi_{k_l})^{GH(K)} = 2 \sum_{m = [K/2]+1}^{K} \omega_m \varphi_{j_l}(\xi_m)(\xi_m/L)^{r_l} \varphi_{k_l}(\xi_m)
\]

\[= \mathcal{O}\left((\varphi_{j_l}, \varphi_{k_l})^{GH(K)}\right) = \mathcal{O}(\mu(K, r_l)).\]

Therefore, from \([35]\), we have (using \(k_0 \approx K \gg r_{\text{max}}\))

\[E_{jk}^r = \mathcal{O}\left(E_{j_0 k_0}^{r_0} \mu(K, r_{\text{max}})^{N-1}\right) = \mathcal{O}\left(\left(\frac{r_0}{|r_0/2|}\right)^{r_0} 2^{-r_0}\right).\]

The \((0, 0)\)-block contains \(\mathcal{O}(r_{\text{max}}^2)\) non-vanishing entries (if \(r_1\) is even), and the total number of non-vanishing entries is of this order of magnitude. Multiplying the matrix with a rapidly decaying vector, we thus find

\[
(E^T v)_j = \sum_{k \in K, E_{jk}^r \neq 0} E_{jk}^r v_k = \mathcal{O}\left(\frac{r_{\text{max}}^2}{|r_{\text{max}}/2|} r_{\text{max}} 2^{-r_{\text{max}}} \max_{k \in K, E_{jk}^r \neq 0} \left\{k^{-\beta}\right\}\right)
\]

\[
= \mathcal{O}\left(\frac{3}{r_{\text{max}}^2} 2^{-r_{\text{max}}(K - r_{\text{max}} + 2)^{-\beta}}\right),
\]

\[\text{Figure 5: Error matrix } (E_{jk}^r)_{k \in K}\text{ for } N = 2\text{ with lexicographically ordered multi-indices from a hyperbolic cross. The outer numbers correspond to } j_1 \text{ and } k_1, \text{ respectively. Numbering with respect to the second coordinate is written inside each } (j_1, k_1)\text{-block. By definition of a hyperbolic cross, the blocks shrink with growing indices on the first coordinate-level. Inside the } (0,0)\text{-block and on the first coordinate-level, we indicate the exactness pattern for one-dimensional quadrature.}\]
Summing up as in (31) and setting
\[
C(\mathcal{R}, W) = \sum_{r \in \mathcal{R}} |\alpha_r(t)| \bar{C} r_{\text{max}}^3 \left( \frac{r_{\text{max}}}{r_{\text{max}}/2} \right)^{2-r_{\text{max}}}
\]
(39)
proves the claim, where the $\mathcal{R}$-, $W$-, and $K$-independent $\bar{C}$ accounts for constants having occurred throughout the analysis).

\[\square\]

5.3 Error $E^{\text{red}}$ due to grid reduction

**Theorem 2** Let $W^{\text{pol}}(\cdot, t) \approx W(\cdot, t)$ be the Chebyshev interpolation polynomial of the potential $W$ on $\Omega = [-L, L]^N$ over $\mathcal{R}(R)$ with $L = \sqrt{2}(K + 1) + 1$ and $K_s = K_s(K)$ a hyperbolically reduced index set with $K \gg R$. Then, under assumption (30) on $v \in C^{K_s}$ (i.e., componentwise decay of order $\beta \in \mathbb{N}$), the error due to grid reduction behaves as
\[
\left| \left( W^{\text{pol}}(\mathbf{s}, t) - W^{\text{GH}}_{K_s, \text{pol}}(t) v \right) \right|_j \leq C(N, R, W, \beta) K^{-\beta}, \quad j \in K_s.
\]

The matrix $W^{\text{pol}}(\mathbf{s}, t)$ results from formally inserting the hyperbolically reduced auxiliary matrices $X^{(l)}_s$ into the polynomial (see Section 3), $W^{\text{GH}}_{K_s, \text{pol}}(t)$ is defined as in Section 1.4. The constant $C(N, R, W, \beta)$ is given as in (43), see below, and depends on $N, R, \text{the regularity of } W,$ and $\beta$ only.

**Proof:** As in the beginning of the previous section, we decompose $W^{\text{pol}}$ and consider a partition of the error
\[
E^{\text{red}}(v)_j = \sum_{r \in \mathcal{R}} \alpha_r(t) E^r_j,
\]
where $E^r_j$ is defined as in (29) with $W^{\text{pol}}_r = (x/L)^r$ in place of $W^{\text{pol}}$.

**Construction of binary trees:** Applying the $l$-th auxiliary matrix twice starting from $v \in C^{K_s}$ yields (provided the occurring indices belong to $K$)
\[
\left( X^{(l)}_s v \right)_j = \sqrt{\frac{j_l}{2}} v_{j_l-e_l} + \sqrt{\frac{j_l + 1}{2}} v_{j_l+e_l},
\]
\[
\left( X^{(l)}_s \right)^2 v_j = \sqrt{\frac{j_l}{2}} \left( \sqrt{\frac{j_l - 1}{2}} v_{j_l-2e_l} + \sqrt{\frac{j_l}{2}} v_j \right) + \sqrt{\frac{j_l + 1}{2}} \left( \sqrt{\frac{j_l + 1}{2}} v_j + \sqrt{\frac{j_l + 2}{2}} v_{j_l+2e_l} \right).
\]

Inductively, the $r_l$-fold application of $\frac{1}{L} \cdot X^{(l)}_s$ can be interpreted as a full binary tree according to the following pattern:

```
height r_l

\sqrt{\frac{j_l+1}{2}}

v_j

\sqrt{\frac{j_l}{2}}

v_{j-e_l}

\sqrt{\frac{j_l+1}{2}}

v_{j+e_l}

\sqrt{\frac{j_l}{2}}

v_{j-2e_l}

\vdots

\sqrt{\frac{j_l}{2}}

v_{j+2e_l}
```

left descent: -1
right descent: +1
With each left descent, the \( l \)-th component of the index is reduced by 1, whereas a right descent increases it. Summing up the \( m \)-th row including the factors at the edges gives

\[
\left( \left( \frac{1}{L} X^{(l)} \right)^m \right)_j.
\]

Starting with \( v_j \) for fixed \( j \in K \), we expand the expression

\[
\left( \left( \frac{1}{L} X^{(r)} \right)^{r}\right)_j = \left( \left( \frac{1}{L} X^{(N)} \right)^{rN} \cdots \left( \frac{1}{L} X^{(1)} \right)^{r_1} \right)_j
\]

layer after layer according to the following pattern as a binary trees \( \mathfrak{T} \) based on \( K \):

The topmost and lowermost layers are numbered 1 and \( N \), respectively. A leaf in layer \( l \) (an \( l \)-leaf) is a root of a subtree in layer \( l + 1 \) (an \((l + 1)\)-root). Each node belonging to layer \( l \) has an index of the general form

\[
j + \sum_{i=1}^{l} \{ \rho_i e_i - \lambda_i e_i \},
\]

where \( \lambda_i \) is the number of left descents in layer \( i \) along the path to the node, \( \rho_i \) respectively. We have \( 0 \leq \lambda_i, \rho_i \leq r_i \), for \( 1 \leq i \leq l \), and \( \lambda_i + \rho_i = r_i \), for \( 1 \leq i \leq l - 1 \). \( \lambda_l = \rho_l = 0 \) is an \( l \)-root, \( \lambda_l + \rho_l = r_l \) is an \( l \)-leaf. In particular, \( \lambda_1 = \rho_1 = 0 \) describes the node \( v_j \) (the 1-root) and \( \lambda_N + \rho_N = r_N \) gives a proper leaf (an \( N \)-leaf). In layer \( l \), only the \( l \)-th component of \( j \) is changed.

The same considerations apply with \( X^{(l)}_s \) in place of \( X^{(l)} \), yielding an analogously defined binary tree \( \mathfrak{T}_s \). We consider the difference tree \( \mathfrak{D} = \mathfrak{T} - \mathfrak{T}_s \). If an index does not belong to \( K \) or \( K_s \), we say that the corresponding node vanishes in \( \mathfrak{T} \) or \( \mathfrak{T}_s \), respectively. By definition, a node in the difference tree \( \mathfrak{D} \) vanishes if both corresponding nodes in \( \mathfrak{T} \) and \( \mathfrak{T}_s \) vanish or do not vanish both at the same time, respectively.

We state the following obvious, yet important observations: Nodes with an index belonging not even to \( K \) vanish in \( \mathfrak{D} \) anyway. An \( N \)-leaf does not vanish in \( \mathfrak{D} \) iff, along the path connecting the 1-root \( v_j \) with the \( N \)-leaf, there is at least one node belonging to \( K \setminus K_s \). As in Section 5.2, we examine non-vanishing \( N \)-leaves in \( \mathfrak{D} \).

**Characterization of non-vanishing leaves:** The examination is done layer-wise. Starting with an \( l \)-root index \( m \in K_s \), we have the following requirements for an \( l \)-leaf depending on \( m \) not to vanish in \( \mathfrak{T} \) or \( \mathfrak{T}_s \), respectively:

(\( \mathfrak{T} \)) The \( l \)-th component index needs to lie between 0 and \( K \), thus

\[
m_l + \rho_l - \lambda_l = j_l + r_l - 2\lambda_l \geq 0, \quad m_l + \rho_l - \lambda_l = j_l + r_l - 2\lambda_l \leq K,
\]

which gives the bounds (independent of \( m \))

\[
\lambda^l_{\min}(r, j) = \left\lfloor \frac{j_l + r_l - K}{2} \right\rfloor \leq \lambda_l \leq \left\lceil \frac{j_l + r_l}{2} \right\rceil = \lambda^l_{\max}(r, j).
\]

20
The upper bound is the same as for $T$. By definition of the hyperbolic cross, one needs

$$1 + ji + rl - 2\lambda_l \leq (K + 1) \left( \prod_{i=1 \atop i \neq l}^{N} (1 + m_i) \right)^{-1} ,$$  \hspace{1cm} (40)$$

From $m \in K$, it follows

$$1 + ji = 1 + ml \leq (K + 1) \left( \prod_{i=1 \atop i \neq l}^{N} (1 + m_i) \right)^{-1} ,$$

thus, for $rl - 2\lambda_l \leq 0$, i.e.,

$$\lambda_l \geq \left[ \frac{rl}{2} \right] = \lambda_{l,\text{min}}(r),$$  \hspace{1cm} (41)$$

the $l$-leaf in $T_s$ does not vanish.

Non-vanishing $l$-leaves in $D$ satisfy ($T$), but not the more restrictive ($T_s$). The converse is not true, since an $l$-leaf violating $T_s$ might still fulfill condition (40), which depends on $m$, thus, be non-vanishing in $T_s$, hence, vanishing in $D$. However, we consider the simpler condition (41). Obviously,

$$\lambda_{l,\text{min}} \leq \lambda_{l,\text{min},s}.$$  

Number of non-vanishing leaves: Summing up as in Section 5.2, we have at most

$$\sum_{s=\lambda_{l,\text{min}}}^{\lambda_{l,\text{max}}} \left( \frac{rl}{s} \right) - \sum_{s=\lambda_{l,\text{min},s}}^{\lambda_{l,\text{max}}} \left( \frac{rl}{s} \right) = \lambda_{l,\text{min},s} - 1 = a_l(r, j)$$

non-vanishing $l$-leaves depending on $m$. We use (32) with $a = \lambda_{l,\text{min}}$, $b = \lambda_{l,\text{min},s} - 1$, $c = rl$. The fact $\lambda_{l,\text{min},s} = \left[ \frac{rl}{2} \right]$ implies $2(\lambda_{l,\text{min},s} - 1) < rl$, hence,

$$a_l(r, j) \approx \left( \frac{rl}{|rl/2|} \right)^{rl}.$$  

Error accumulation and decay condition: Along the path from $v_j$ to any $l$-leaf, the most unfavorable weight is the product

$$b_l(r, j) = 2^{-r_l/2} \prod_{s=1}^{rl} \frac{(ji + s)^{1/2}}{L} = \mathcal{O} \left( 2^{-r_l/2} \left( \frac{ji + rl}{K} \right)^{r_l/2} \right),$$

i.e., taking always a right descent. According to (30), the largest $N$-leaf is

$$c(r, j) = \prod_{l=1}^{N} \left\{ \max_{-r_l \leq s \leq rl} |v_j - s| \right\} = \mathcal{O} \left( \prod_{l=1}^{N} (ji - rl)^{-\beta} \right).$$

Thus, the error over all layers is bounded according to

$$|E_l^j| \leq \prod_{l=1}^{N} \{a_l(r, j) \cdot b_l(r, j)\} \cdot c(r, j) = \mathcal{O} \left( \prod_{l=1}^{N} a_l(r, j) \cdot b_l(r, j) \cdot (ji - rl)^{-\beta} \right).$$  \hspace{1cm} (42)$$
The error does not vanish only in case $j + r \notin \mathcal{K}_s$, thus,

$$K + 2 \leq \prod_{l=1}^{N}(1 + j_l + r_l) = \prod_{l=1}^{N}(j_l - r_l) \cdot \prod_{l=1}^{N} \left(1 + \frac{1 + 2r_l}{j_l - r_l}\right) \leq \prod_{l=1}^{N}(j_l - r_l) \cdot 2^N \prod_{l=1}^{N}(1 + r_l),$$

which yields

$$\prod_{l=1}^{N}(j_l - r_l)^{-\beta} \leq 2^{\beta N} \prod_{l=1}^{N}(1 + r_l)^{\beta}(K + 2)^{-\beta}.$$ 

Therefore, using $K \gg R$,

$$E_j^r = O \left( \left\{ \prod_{l=1}^{N} \left( \frac{r_l}{|r_l/2|} \right) r_l \right\} \cdot \left\{ \prod_{l=1}^{N} 2^{-r_l/2} \right\} \cdot \left\{ 2^{\beta N} \prod_{l=1}^{N}(1 + r_l)^{\beta}(K + 2)^{-\beta} \right\} \right)$$

$$= O \left( 2^{\beta N} \cdot \left\{ \prod_{l=1}^{N} \left( \frac{r_l}{|r_l/2|} \right) r_l 2^{-r_l/2} \right\} \cdot \left\{ \prod_{l=1}^{N}(1 + r_l)^{\beta} \right\} \cdot K^{-\beta} \right).$$

Summing up and setting

$$C(N, R, W, \beta) = \sum_{r \in \mathcal{R}} |\alpha_r(t)| \hat{C} 2^{\beta N} \left\{ \prod_{l=1}^{N} \left( \frac{r_l}{|r_l/2|} \right) r_l 2^{-r_l/2} \right\} \cdot \left\{ \prod_{l=1}^{N}(1 + r_l)^{\beta} \right\}$$
proves the claim, where the $\mathcal{R}$, $W$-, $K$-, and $\beta$-independent $\hat{C}$ accounts for constants having occurred throughout the analysis.

Remarks: According to the choice of $j \in \mathcal{K}_s$ or $\mathcal{R}(R)$, the above error estimate might improve:

1) More than one large component in $j$: If $n(\mathcal{R}, j) = \min_{r \in \mathcal{R}} n(j)$ and $n(j) \in \{1, \ldots, N\}$ is the number of components $j_l$ such that $K \approx j_l \gg r_l$, we get $c(r, j) = O(K^{-n(\mathcal{R}, j)\beta})$, thus,

$$E^{red}(u)_j = O \left( \sum_{r \in \mathcal{R}} |\alpha_r(t)| \left\{ \prod_{l=1}^{N} \left( \frac{r_l}{|r_l/2|} \right) r_l 2^{-r_l/2} \right\} K^{-n(\mathcal{R}, j)\beta} \right)$$

2) Only small index components: If $j + r \in \mathcal{K}_s$, all branches in $\mathcal{O}$ cancel out and the error $E_j^r$ vanishes altogether.

3) The factor $\prod_{l=1}^{N}(1 + r_l)^{\beta}$ as occurring in (43) improves if $\mathcal{R}(R)$ is reduced (hyperbolically, e.g.).

5.4 Analysis for the full integration

Finally, we briefly sketch an error analysis for the overall integration of $[1]$ citing well-known results from the literature and pointing out error contributions due to the fast algorithm. 

Notation: To facilitate the error analysis, consider the following notational conventions.

$$\psi = \psi(x, t)$$

$\psi_{\mathcal{K}_s}$

$\psi_{\mathcal{K}_s, pol}(x) = \sum_{k \in \mathcal{K}_s} c_{pol}^{GH(K):n}(x)\varphi_k(x)$$
\[
\begin{align*}
&c_s(t) &\text{exact solution of } (13) \text{ over } K_s \\
&c_{s,\text{pol}}(t) &\text{exact solution of } (15) \text{ over } K_s \\
&c_{s,\text{pol}}^{GH(K)}(t) &\text{exact solution of } (16) \text{ over } K_s \\
&c_{s,\text{pol}}^{GH(K):n} &\text{Magnus + Lanczos approx. to } c_{s,\text{pol}}^{GH(K)}(t_n) \text{ over } K_s, \text{ Lanczos perturbed by fast algorithm} \\
&c_{s,\text{pol}}^{GH(K):n} &\text{Magnus + Lanczos approx. to } c_{s,\text{pol}}^{GH(K)}(t_n) \text{ over } K_s, \text{ unperturbed Lanczos} \\
&c_{s,\text{pol}}^{GH(K):n} &\text{Magnus approx. to } c_{s,\text{pol}}^{GH(K)}(t_n) \text{ over } K_s, \text{ matrix exponential exact} \\
&W(x,t) &\text{original potential} \\
&W^{\text{pol}}(x,t) = \sum_{r\in\mathbb{R}} a_r(t) T_r(x/L) &\text{polynomial approx. to time-dependent potential } W \\
&W_{K_s}(t) = (\langle \varphi_j, W(t) \varphi_k \rangle)_{j,k\in K_s} &\text{exact stiffness matrix over } K_s \\
&W_{K_s,\text{pol}}(t) = (\langle \varphi_j, W^{\text{pol}}(t) \varphi_k \rangle)_{j,k\in K_s} &\text{exact stiffness matrix over } K_s \text{ with } W^{\text{pol}} \\
&W^{GH(M)}_{K_s,\text{pol}}(t) = (\langle \varphi_j, W^{\text{pol}}(t) \varphi_k \rangle^{GH(M)})_{j,k\in K_s} &\text{approximate stiffness matrix over } K_s \text{ with } W^{\text{pol}} \\
\end{align*}
\]

Analogous quantities defined over a full index set \( K \) are denoted in the same manner except for simply skipping subscripts \( s \).

**Decomposition of error:** At time \( t_n \), due to Parseval’s identity, we have

\[
\| \psi_{K_s,\text{pol}}^{GH(K):n} - \psi(\cdot, t_n) \|_{L^2(\mathbb{R}^N)} \leq \| \psi_{K_s,\text{pol}}^{GH(K):n} - \psi_{K_s}(\cdot, t_n) \|_{L^2(\mathbb{R}^N)} + \| \psi_{K_s}(\cdot, t_n) - \psi(\cdot, t_n) \|_{L^2(\mathbb{R}^N)} \\
= \left\| c_{s,\text{pol}}^{GH(K):n} - c_s(t_n) \right\|_2 + \| \psi_{K_s}(\cdot, t_n) - \psi(\cdot, t_n) \|_{L^2(\mathbb{R}^N)}.
\]

Error \( S_1 \) is the error due to the Galerkin approximation itself. \( \text{Lubich (2008), Thm. III.1.6, gives} \) the following result: For a time-independent potential \( V(x) = (1 + |x|^2) B(x) \) with bounded \( B \) and initial value \( \psi_{K_s}(x, 0) = \sum_{k\in K_s} \langle \psi(0), \varphi_k \rangle \varphi_k(x) \), for any fixed integer \( r \), the error is bounded by

\[
\| \psi_{K_s}(\cdot, t) - \psi(\cdot, t) \|_{L^2(\mathbb{R}^N)} \leq C(s,N) K^{-r/2} (1 + t) \max_{0 \leq \tau \leq t} \max_{0 \leq r \leq t + 2} \| A^r \psi(\tau) \|_{L^2(\mathbb{R}^N)}
\]

(provided \( \psi \) is sufficiently regular). By the same proof, adding time-dependence to \( B \) yields the same convergence result with \( C = C(s,N,t) \) depending on the bound of \( B \) with respect to \( t \). The error term \( * \) decomposes as follows:

\[
\begin{align*}
&c_{s,\text{pol}}^{GH(K):n} - c_s(t_n) = \underbrace{\left( c_{s,\text{pol}}^{GH(K):n} - c_{s,\text{pol}}^{GH(K):n} \right)}_{T_1} + \underbrace{\left( c_{s,\text{pol}}^{GH(K):n} - c_{s,\text{pol}}^{GH(K):n} \right)}_{T_2} + \underbrace{\left( c_{s,\text{pol}}^{GH(K):n} - c_{s,\text{pol}}^{GH(K):n} \right)}_{T_3} \\
&+ \left( c_{s,\text{pol}}^{GH(K)}(t_n) - c_{s,\text{pol}}(t_n) \right) + \left( c_{s,\text{pol}}(t_n) - c_s(t_n) \right)
\end{align*}
\]
Error terms denoted by $S$ stem from spatial discretization, terms denoted by $T$ are temporal errors. $S2$ is the error due to quadrature and $S3$ is the error due to polynomial approximation of the potential. Below, we comment on both of them using the analysis given in Section 5.2. $T1$ is the error due to perturbation in the Lanczos process, which is influenced by the error induced by the fast algorithm itself as analyzed in Section 5.3. We comment on $T1$ at the end of this paragraph. $T2$ is the error due to Lanczos itself. If all eigenvalues of $K(t)$ are in the intervall $[a, b]$ and if $c_{s, pol}^{GH(K)n}$ is of unit Euclidean norm, then, the error of the Lanczos method (23) is bounded by
\[
\|c_{s, pol}^{GH(K)n} - c_{s, pol}^{GH(K)n}\|_2 \leq O(t^{\beta/2}) \|D^{1/2} c_{s, pol}^{GH(K)}(t_n)\|_2,
\]
with $C$ depending only on bounds for first- and second-order spatial derivatives of $W_{K_{s, pol}}^{GH(K)}(t)$ and a certain commutator bound for $H_{K_{s, pol}}^{GH(K)}(t)$. The method (20) exhibits the error bound
\[
\|c_{s, pol}^{GH(K)n} - c_{s, pol}^{GH(K)}(t_n)\|_2 \leq Ch^2 t_n \max_{0 \leq t_n \leq t_{n+1}} \|D^{1/2} c_{s, pol}^{GH(K)}(t_n)\|_2
\]
with $C$ depending only on bounds for spatial derivatives of $W_{K_{s, pol}}^{GH(K)}(t)$ up to order 4, a commutator bound on $H_{K_{s, pol}}^{GH(K)}(t)$, and a mild time-step restriction $h \|D\|_2 \leq c$. For both estimates, see Hochbruck & Lubich (2003).

**Error due to quadrature:** As for the spatial errors $S2$ and $S3$, consider two systems of $n$ differential equations
\[
i\dot{y}(t) = H(t)y(t), \quad \dot{y}(t) = \hat{H}(t)\hat{y}(t),
\]
with $H$ and $\hat{H}$ both being Hermitian matrices and $y(0) = \hat{y}(0)$. We analyze the difference $y(t) - \hat{y}(t)$ for the corresponding solutions using standard techniques. Subtracting both equations, multiplying with $\hat{y} - \hat{y}$, taking the real part on both sides, and applying Cauchy-Schwarz gives
\[
\frac{d}{dt}\|\hat{y}(t) - y(t)\|_2 \leq \|(\hat{H}(t) - H(t))y(t)\|_2.
\]
Integrating from $0$ to $t$ yields
\[
\|\hat{y}(t) - y(t)\|_2 \leq \int_0^t \|(\hat{H}(\tau) - H(\tau))y(\tau)\|_2.
\]
(44)
For the error term $S2$, take $H(t) = D_{K_{s, pol}} + W_{K_{s, pol}}(t)$ and $\hat{H}(t) = D_{K_{s, pol}} + W_{K_{s, pol}}^{GH(K)}(t)$. From Theorem [1] we then have the following

**Theorem 3 (S2)** Under the same assumptions as in Theorem [1] the error due to quadrature is bounded by
\[
\|c_{s, pol}^{GH(K)}(t) - c_{s, pol}(t)\|_2 \leq \int_0^t \left\|\left(E_{j,k}(\tau)\right)_{j,k \in K_{s, pol}} \right\|_2 d\tau = O(tK^{N-\beta}).
\]
\[\square\]
The error $S3$ is dealt with in the same manner using [44] with $H(t) = D_{K_s} + W_{K_s}(t)$ and $\tilde{H}(t) = D_{K_s} + W_{K_s,\text{pol}}(t)$.

Perturbed Lanczos process: Using the Lanczos process [21] with 

$$(Av_k)^{fast} = Av_k - \left( Av_k - (Av_k)^{fast} \right) = Av_k - f_k$$

instead of $Av_k$ produces perturbed basis vectors and coefficients $\tilde{V}_m$ and $\tilde{T}_m$, respectively. This yields

$$A = \tilde{V}_m \tilde{T}_m \tilde{V}_m^* + F_m \tilde{V}_m^*,$$

where $F_m = \langle f_1 \ldots f_m \rangle$. Thus, by [22] and [45],

$$V_m \tilde{T}_m V_m^* = \tilde{V}_m \tilde{T}_m \tilde{V}_m^* + F_m \tilde{V}_m^*.$$

We approximate $e^{-ihA}v \approx \tilde{V}_m e^{-ih\tilde{T}_m}e_1$, and the local error is given by

$$V_m e^{-ihT}e_1 - \tilde{V}_m e^{-ih\tilde{T}_m}e_1 = V_m e^{-ihT}V_m^* V_m e_1 - \tilde{V}_m e^{-ih\tilde{T}_m}\tilde{V}_m^* \tilde{V}_m e_1 = (e^{V_m(-ih)T}V_m^* - e^{\tilde{V}_m(-ih)\tilde{T}_m}\tilde{V}_m^*) V_1 = \left(e^{\tilde{V}_m(-ih)\tilde{T}_m}e_1 - e^{\tilde{V}_m(-ih)\tilde{T}_m}\tilde{V}_m e_1 \right) - (\tilde{V}_m e^{-ih\tilde{T}_m}e_1) V_1.$$

Using the sensitivity analysis for the matrix exponential given in [Van Loan (1977)], we get

$$\left\| V_m e^{-ihT}e_1 - \tilde{V}_m e^{-ih\tilde{T}_m}e_1 \right\|_2 \leq h\|F_m\|2e^{|h\|A\|2+\|F_m\|2}.$$

Hence, the error $T1$ goes to zero if $h$ is sufficiently small and the fast algorithm is sufficiently accurate. Note, however, that the vectors $v_k$, $k \geq 2$, might fail to decay sufficiently fast if $m$ becomes too large or if $K$ is not large enough. Thus, we restrict ourselves to sufficiently small $m$ (say, $m \approx 5$).

### 6 Numerical experiments

All figures have been obtained with a FORTRAN 95 implementation on an Intel Core 2 Duo E8400 3.00 GHz processor with 4 GB RAM in double precision arithmetics.

Local errors due to quadrature and grid reduction: Let $K_s = K_s(K)$ be a hyperbolically reduced $N$-dimensional index set. First, we illustrate the error

$$E^{\text{quad}}_v = \left( W_{K_s,\text{pol}} - W_{K_s,\text{pol}}^{GH(K)} \right) v,$$

due to quadrature for different choices of $N$ and $K$, see Figure 6. Then, we consider the error

$$E^{\text{red}}(v) = W^\text{pol}(X_s) v - \Omega_s \left( W^\text{pol}(X) \Omega_s(v) \right)$$

due to grid reduction, see Figure 7. In both cases, the chosen potential is a stretched torsional potential as given in [27] approximated by its Chebyshev interpolation polynomial over $\mathcal{R}(R)$ with $R = 8$, yielding an interpolation error of size $\approx 1e-10$. For the vector $v \in \mathbb{C}^{[K_s]}$ to exhibit a decay behavior according to [30], we set

$$v_k = \prod_{l=1, k_l \neq 0}^{N} k_l^{-\beta},$$

(46)
and normalize so that $\|v\|_2 = 1$. We test different choices of $\beta$. As explained in Section 5.1, for $K$ being sufficiently large, the error $\left(E^{\text{red}}(v)\right)_j$ decreases the faster the more components $j$ of $\mathbf{j}$ are large with respect to $R$, see Theorem 2 and the remarks thereafter. Figure 8 illustrates this decay behavior in the individual components of $E^{\text{red}}(v)$ for $N = 2$.

**Lanczos process:** We approximate the matrix exponential $e^{-ih\mathcal{W}^K_{K_s,\text{pol}} v}$ using an $m$-step Lanczos process. Again, $\mathcal{W}$ is the above torsional potential (Chebyshev interpolation, $R = 8$) and $v$ decays as in (46). In each step, using the fast algorithm $\left(W^K_{K_s,\text{pol}} v_k\right)^{\text{fast}}$ instead of $W^K_{K_s,\text{pol}} v_k$ gives rise to a perturbation error

$$\left\|V_m e^{-ihT_m e_1} - \tilde{V}_m e^{-ihT_m e_1}\right\|_2 \leq h\|F_m\|_2 e^h \left(\left\|W^K_{K_s,\text{pol}}\right\|_2 + \|F_m\|_2\right),$$

(47) see Figure 9 for some numerical results in case $N = 2$. As pointed out in the end of the last Section, for $m \leq 2$, the Lanczos vectors $v_k, k \geq 2$, need not decay sufficiently fast. Their behavior is illustrated in Figure 10. If $m$ is too large, the perturbation error dominates the error due to Lanczos itself. Enlarging $K$ reduces the perturbation error, as illustrated by the examples below.

**Time integration:** We consider two instances of the general equation (1). First, as a time-independent problem, consider once again the torsional potential (27) (Chebyshev interpolation, $R = 8$), i.e.,

$$V(x) = \sum_{i=1}^{N} (1 - \cos(x_i/L)), \quad x \in [-L, L]^N, \quad L = \sqrt{2(K + 1) + 1},$$

(48)

with $N = 2$, $K = 25, 75$. The resulting ODE corresponding to (16) is integrated over $[0, 1]$ with initial value $v$ given as in (46) ($\beta = 5$) using the (time-independent) scheme (19) of order 2. In

| $K$ | $N = 2$ | $N = 3$ | $N = 4$ |
|-----|--------|--------|--------|
| $\beta$ | 25     | 50     | 100    |
| 2    | 6.821e-05 | 1.821e-05 | 4.777e-06 |
| 3    | 2.812e-06 | 3.755e-07 | 4.929e-08 |
| 4    | 1.130e-07 | 7.544e-09 | 4.959e-10 |
| 5    | 4.519e-09 | 1.512e-10 | 4.965e-12 |

Figure 6: Error $E^{\text{quad}}(v)$ due to quadrature for the torsional potential (27) approximated by its Chebyshev interpolation polynomial with $R = 8$. Upper half: $N = 2$, different choices of $\beta$ and $K$. The solid and dashed lines represent the observed errors and the bound $C(R, \mathcal{W})K^{-\beta}$, respectively, see Theorem 1. Lower half: $\beta = 5$, different choices of $N$ and $K$. The factor $C(R, \mathcal{W})$ does not depend on $N$, see (39). Selected errors in cases $N = 3, 4$ are indicated by plus signs and circles, respectively. In the last row (“time”) of the lower table, computation times for $W^K_{K_s,\text{pol}}$ (assembly and multiplication) in case $K = 60$ are shown. As for $N = 4$, assembling the matrix (plus operating on a vector) takes more than a day - even on a reduced grid.
Figure 7: Error $E^{red}v$ due to grid reduction for the torsional potential (as above, $R = 8$). Upper half: $N = 2$, different choices of $\beta$ and $K$, see Theorem 2. The small characteristic “steps” stem from layer-wise growing behavior of the hyperbolic cross for increasing $K$. Lower half: $\beta = 5$, different choices of $N$ and $K$. The solid line represents the observed error in case $N = 2$. Selected errors in cases $N = 3, 4$ are indicated by plus signs and circles, respectively. Increasing $N$ worsens the factor $C(N, R, W, \beta)$, see [13]. In the last row (“time”) of the lower table, computation times for the fast algorithm in case $K = 60$ are shown.

Errors $|E^{red}(v)|_j$, $N = 2$

Figure 8: Errors $(E^{red}(v))_j$ due to grid reduction for a torsional potential with $N = 2$ and different choices of $K$. Each entry represents an error vector component with an index taken from a hyperbolically reduced grid. The vector $v$ decays as above with $\beta = 3$. Errors which are small with respect to the largest observed error component $e_{max} \approx 3.998e^{-4}$ are simply indicated by a dot, indices carrying larger errors being indicated by a grey box. The darker the box, the closer the error to $e_{max}$. The picture corresponding to $K = 30$ shows an enlarged view. The symbol $\#$ points to the number of large error components. Clearly, their number decreases with growing $K$, and the errors decrease alike as indicated by increasingly lighter boxes.

Figure 11 convergence results for different choices of Lanczos steps $m$ and of time-step $h$ together with corresponding Lanczos perturbation errors are shown.
Second, as a time-dependent example, consider a stretched Hénon-Heiles potential with a linear time-dependent perturbation, i.e.,

\[ V(x, t) = \sum_{l=1}^{N-1} \left[ \left( \frac{x_l}{L} \right)^2 (x_{l+1}/L) - \frac{1}{3} \left( x_{l+1}/L \right)^3 \right] - \sin^2(t)x_1, \quad x \in [-L, L]^N, \]

(49)

where \( L = \sqrt{2(K+1)} + 1 \) as above. This models the interaction of an atom / a molecule with a high-intensity CW laser in \( x_1 \)-direction, see Peskin et al. (1994) (with a quantum Harmonic oscillator in place of a HH-potential). We choose \( N = 2, 3, m = 5 \), and test with varying \( K \). To approximate the corresponding potential \( W \), Chebyshev interpolation with \( R = 3 \) is used, the interpolation error being of size \( \approx 1e-12 \). Convergence results for an integration over \([0, 1]\) with initial value \( v \) given as in (46) \( (\beta = 3) \) using the scheme (20) of order 4 are shown in Figure 12.

### 7 Further applications of the fast algorithm

The fast algorithm is designed for accelerating (or making feasible, in the first place) time discretization of a resulting coefficient ODE after a spectral approximation of the linear Schrödinger equation in space. Using an ONB \( \{ \varphi_k \}_{k \in \mathbb{N}} \) other than (tensor-products of) Hermite functions, one might also consider more general linear problems leading to coefficient ODEs

\[ \dot{c}(t) = S(t)c(t), \quad S_{jk}(t) = (\varphi_j, L(t)\varphi_k), \quad j, k \in \mathcal{K}, \]

where, e.g., \( L(t) \) is an elliptic operator with time-dependent coefficients. In case \( \{ \varphi_k \} \) consists of algebraic orthogonal polynomials (say, Legendre or Chebyshev polynomials), with the help of

---

**Figure 9:** Comparison of the error due to a perturbation of the Lanczos process by the fast algorithm (see (47), black figures) to the error due to classical Lanczos without using the fast algorithm (gray figures) for \( N = 2, K = 30 \) and \( \beta = 3 \). Clearly, increasing \( m \) or decreasing \( h \) improves the error due to Lanczos. However, if \( m \) is not sufficiently small, the error due to the perturbation becomes dominant.

**Figure 10:** Decay behavior of the Lanczos vectors \( v_k \) in case \( m = 4 \). As in Figure 8, each entry represents a vector component \( |(v_k)_j| \), \( 1 \leq k \leq 4, j \in \mathcal{K} \). Components being larger than \( c \cdot |v_j| \) (see (46)) are marked by boxes corresponding to different values of \( c \). The darker the box, the larger is \( c \) and the less sharp is the decay by contrast with \( v_j \).
max. error torsional potential at $t = 1$

Figure 11: Error $\max_j \left| e^{GH(K)_n}_{s,\text{pol}} - e^{GH(K)}(t^n) \right|_{j \in K_s}$ at time $t^n = 1$ when integrating (1) with $V(x)$ as in (48) (Chebyshev interpolation, $R = 8$) using the scheme (19) with different choices of time step $h$. The initial value is given as in (46) with $\beta = 5$. The number $m$ of Lanczos steps varies as indicated in the figures. Corresponding errors due to a perturbation of Lanczos are indicated by plus signs, circles, and asterisks, respectively. Left: $N = 2$, $K = 25$. If $m$ is too small ($m = 2$), the error due to Lanczos itself is dominant. The choice $m = 3$ makes visible the desired order of convergence until the perturbation error becomes dominant, whereas for $m = 4$, the latter error dominates even for moderate choices of $h$. Right: $N = 2$, $K = 75$. Increasing $K$ improves the perturbation error, whereas the error due to Lanczos itself requires smaller choices of $h$.

max. error perturbed Hénon-Heiles at $t = 1$

Figure 12: Error $\max_j \left| e^{GH(K)_n}_{s,\text{pol}} - e^{GH(K)}(t^n) \right|_{j \in K_s}$ at time $t^n = 1$ when integrating (1) with $V(x,t)$ as in (49) (Chebyshev interpolation, $R = 3$) using the 2-stage Gauss-Legendre Magnus integrator (20). The choice is $N = 2$ (left) and $N = 3$ (right), $m = 5$, and the initial value is given as in (46) with $\beta = 3$. The number $K$ of basis functions varies as indicated in the figure. Clearly, the perturbation error decreases as $K$ increases, and it dominates unless $K$ is chosen sufficiently large. Due to the constant in the error estimate from Theorem 2 depending on $N$, making visible the order of the employed scheme and obtaining equally small error results requires larger choices of $K$ in case $N = 3$ than in case $N = 2$. To obtain a reference, the scheme (20) has been employed with with $h = 1e-5$ and 15 Lanczos steps in each time step.

existing recurrence relations, every entry in $S$ can be taken into the form

$$f(k)(\varphi_j, W(t)\varphi_{k+\mu e_m + \nu e_n}),$$

29
where $1 \leq m, n \leq N$ and $\mu, \nu \in \{0, \pm 1\}$. If $\mu \neq 0$ or $\nu \neq 0$, the diagonalization given in Section 3.1 is no longer valid. A modified version of the fast algorithm first multiplies each component of $v \in \mathbb{C}^{|K_s|}$ with $f(k)$. Then, one uses a shift in the vector, namely, $\tilde{v}_k = v_k - \mu e_m - \nu e_n$, and operates with the original auxiliary matrices on $\tilde{v}$. Again, $W$ has to be approximated by a polynomial. As long as the problem is linear, this strategy works for a broad class of equations. After discretization in space, any means of time integration involving an approximation of the matrix exponential can be treated with the fast algorithm.

**Conclusion**

We have presented a fast algorithm for the efficient treatment of the coefficient ODE resulting from spatial discretization of the linear Schrödinger equation in higher dimensions with a time-dependent potential by a spectral Galerkin method. As time discretization of this ODE typically involves products of the time-dependent Galerkin matrix with a vector, assembling this matrix and doing the multiplication explicitly is prohibitive due to the complexity of the problem – even more so in each time step. Together with a hyperbolical reduction of the spectral basis, the fast algorithm provides a direct approach for this problem to circumvent complexity issues and reduce computational efforts considerably. It consists of a Horner-like, fast application of auxiliary matrices formally inserted into the polynomially approximated potential. On a full grid, this procedure is equivalent to Gauss-Hermite quadrature with exactly as many nodes as there are basis functions in each direction. On a reduced grid, we have analyzed the resulting quadrature and grid reduction errors by casting the problem as an examination on binary trees. As it turns out, if the underlying potential is sufficiently smoother than the exact solution, both errors decay rapidly. Approximating the potential on a reduced grid further improves the error. We have pointed out that the fast algorithm constitutes a tool that can be applied for spectral discretizations of linear problems based on orthogonal polynomials other than the Schrödinger equation with Hermite functions.

**Acknowledgement:** The author is grateful to Christian Lubich for his valuable suggestions and a helpful supervision. Bernd Brumm is funded by the DFG Priority Program 1324.

**References**

M. Abramowitz & I.A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1965.

S. Blanes & P.C. Moan, *Splitting methods for the time-dependent Schrödinger equation*, Phys. Lett. A 265 (2000), 35-42.

H.-J. Bungartz & M. Griebel, *Sparse Grids*, Acta Numerica 13 (2004), 147-269.

C. Canuto, A. Quarteroni, M.Y. Hussaini, T.A. Zang, *Spectral Methods: Fundamentals in Single Domains*, Springer, Berlin, 2006.

S. Blanes, F. Casas, J.A. Oteo & J. Ros, *The Magnus expansion and some of its applications*, Physics Reports 470 (2009), 151-238.

T. Carrington & P.-N. Roy, *A direct-operation Lanczos approach for calculating energy levels*, Chemical Physics Letters 257 (1996), 98-104.

E. Faou & V. Gradinaru, *Gauss-Hermite wavepacket dynamics: convergence of the spectral and pseudo-spectral approximation*, IMA Journal of Numerical Analysis 29 (2009), 1023-1045.
E. Faou, V. Gradinaru & Ch. Lubich, *Computing semiclassical quantum dynamics with Hagedorn wavepackets*, SIAM J. Sci. Comp. 31 (2009), 3027-3041.

L. Gauckler, *Convergence of a split-step Hermite method for the Gross-Pitaevskii equation*, IMA J. Numer. Anal. 31 (2011), 396-415.

V. Gradinaru, *Fourier Transform on Sparse Grids: Code Design and Application to the Time Dependent Schrödinger Equation*, Computing 80 (2007), 1-22.

V. Gradinaru, *Strang Splitting for the Time Dependent Schrödinger Equation on Sparse Grids*, SIAM J. Num. Analysis 46 (2007), 103-123.

V. Gradinaru & G. Hagedorn, *Convergence of a semiclassical wavepacket based time-splitting for the Schrödinger equation*, to appear in Numerische Mathematik, see [www.math.vt.edu/people/hagedorn/gradhag2.pdf](http://www.math.vt.edu/people/hagedorn/gradhag2.pdf).

M. Hochbruck & Ch. Lubich, *On Magnus integrators for time-dependent Schrödinger equations*, SIAM J. Numer. Anal. 41 (2003), 945-963.

J.C. Light & T. Carrington, *Discrete variable representations and their utilization*, Adv. Chem. Phys. 114 (2000), 263-310.

Ch. Lubich, *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis*, Europ. Math. Soc., Zurich, 2008.

U. Peskin, R. Kosloff & N. Moiseyev, *The Solution of the Time-Dependent Schrödinger Equation by the (t, t’) Method: The Use of Global Polynomial Propagators for Time-Dependent Hamiltonians*, J. Chem. Phys. 100 (1994), 8849-8855.

B. Thaller, *Visual Quantum Mechanics*, Springer, New York, 2000.

Ch. Van Loan, *The Sensitivity of the Matrix Exponential*, SIAM J. Numer. Anal. 14 (1977), 971-981.