A fast matrix-free spectral method for the nonlinear Schrödinger equation

Bernd Brumm

January 31, 2016

We consider the Schrödinger equation with a cubic nonlinearity and a multiplicative potential. Using a Hermite–Galerkin ansatz for a discretization in space, time-propagation typically necessitates to compute matrix-vector products with large and dense Galerkin matrices. We propose an efficient procedure to approximate these matrix-vector products without assembling the matrices themselves—and with costs that essentially scale only linearly with the size of the Galerkin basis if the squared modulus is considerably more regular than the wave function itself. The underlying ideas are based on a previous approach for the linear Schrödinger equation due to the same author. Additionally, we provide an analysis for the spatial error. Time-propagation is done using Strang splitting. Numerical experiments corroborate the accuracy of our method.

AMS subject classifications: 65M12, 65M15, 65M70, 65Z05, 81-08, 81Q05

1 Introduction

We consider the Schrödinger equation with a cubic nonlinearity in dimensionless units,

\[ i\dot{\psi}(x,t) = -\frac{1}{2}(\Delta \psi)(x,t) + \frac{1}{2} \sum_{\alpha=1}^{d} x_{\alpha}^{2} \psi(x,t) + W(x,t) + |\psi(x,t)|^{2} \psi(x,t), \]

where \( W \) is a real-valued multiplicative, possibly time-dependent potential. Eq. (1) is usually referred to as the Gross–Pitaevskii equation and describes the wave function of a Bose–Einstein condensate at ultra-low temperature; see [14, 21] and, more recently, [9]. Propagation in time is commonly done using a splitting ansatz in combination with a
pseudospectral method in space, and time-splitting for the nonlinear Schrödinger equation has gained much attention in recent years; see, e.g., [2, 3, 11, 16, 17, 24, 25] and the abundance of references therein. While elaborate time-stepping schemes have been devised, in higher dimensions, spatial discretization is usually based on a simple tensorized approach with a number of basis functions that depends exponentially on the dimension. Since the resulting matrices are thus very large and in general dense, this constitutes a computational bottleneck for which the catch phrase “curse of dimensionality” has been coined; cf. [4]. Fast transforms can be a loophole (see, e.g., [22], Ch. 12), but for many spectral bases, there are no such transforms. This computational challenge from spatial discretization is addressed in the present work.

We discretize (1) in space by a Hermite–Galerkin method. Using Strang splitting in time, the harmonic oscillator part yields a trivially computable analytic solution. For the remainder part, however, time-stepping involves matrix-vector products with matrix representations of the external potential \( W \) and of the squared modulus, and with computational costs that scale quadratically with the size of the spectral basis (leaving aside the costs for an explicit assembly of the dense Galerkin matrices). This can easily make practical computations infeasible. As a remedy, we propose an efficient method to compute these matrix-vector products approximately, and where the result vector is obtained directly without assembling the matrices themselves. Under suitable regularity assumptions on the squared modulus of the wave function, the overall costs essentially scale only linearly with the size of the spectral basis. This speeds up computations drastically.

Our method is based on a previous approach for the linear Schrödinger equation; see [6, 7]. In the linear setting, the underlying ideas for an efficient computation of matrix-vector products involving a matrix representation of the potential \( W \) go along the following lines: First, we approximate \( W \) by a polynomial—say, we employ Chebyshev interpolation to obtain a polynomial approximate \( W^{\text{int}} \). Second, we use a Galerkin basis of tensor-products of mutually orthogonal functions defined by a short recurrence, such as classical orthogonal polynomials and their associated functions; see, e.g., [12]. Matrix representations of the position operators can then be formally inserted into \( W^{\text{int}} \). Our choice of basis allows to compute the action of these coordinate matrices onto a vector directly (i.e., without assembling the matrices) and in linear time only, and the action of the overall matrix due to formal insertion eventually boils down to sequential applications of the direct operation procedure for these coordinate matrices done in a suitable way. If the Chebyshev approximate of \( W \) contains only few terms as compared to the size of the Galerkin basis, the essentially linear scaling is retained. Computation times for the matrix-vector product as compared to an explicit assembly are reduced by several orders of magnitude. This has originally been proposed as a tentative idea in [10] and turned into a rigorous algorithm including an error analysis in [6]. In [7], the methodology has been further developed to comprise initial-boundary value problems and more complicated differential operators with non-constant coefficients as well.

The present work carries forward the techniques from these sources to an efficient treatment of the cubic nonlinearity—as regards both the derivation of the algorithm and the subsequent error analysis. As a first step, we project the squared modulus of
the wave function onto a suitable subspace of the Galerkin approximation space. It
is then seen that the unknown approximation coefficients for the squared modulus can
be obtained from the Galerkin coefficients for the wave function itself. Computing the
former set of coefficients from the latter involves exactly the same efficient matrix-vector
procedure as is needed for the matrix representation of the squared modulus anyway,
with matrices of triple products of Hermite basis functions. And this procedure, in turn,
emulates the original linear fast algorithm due to [6, 7], but with tensor-products of
Hermite functions in place of Chebyshev polynomials. Roughly, we can thus treat the
cubic nonlinearity in a similar way as an external potential.

The paper is structured as follows: Section 2 contains a step-by-step derivation of
the fast algorithm. Starting from brief outlines of the Strang splitting ansatz in time
(Section 2.1) as well as the Hermite–Galerkin ansatz in space (Section 2.2), Sections 2.3
– 2.6 contain a detailed discussing of the proceeding in four steps. In Section 2.7, we
eventually give an overview of the overall propagation in time as it involves the newly
devised method. Section 3 presents an analysis for the error due to spatial discretization
with a stress on those error contributions that stem from the efficient matrix-vector
procedures. Numerical experiments given in Section 4 corroborate the theoretical results.

2 The fast algorithm

We search for \( \psi = \psi(x,t) \), \( x \in \mathbb{R}^d \) and \( t \geq 0 \), such that (omitting unnecessary arguments)
\[
   i \psi_t = D \psi + (W \psi + |\psi|^2) \psi,
\]
with initial data \( \psi_0(x) \in L^2(\mathbb{R}^d) \), where \( D \) is the harmonic oscillator, viz.,
\[
   (D \psi)(x,t) = -\frac{1}{2}(\Delta \psi)(x,t) + \frac{1}{2} \sum_{a=1}^{d} x_a^2 \psi(x,t),
\]
and \( W = W(x,t) \) is a real-valued multiplicative, possibly time-dependent potential. This
section is dedicated to a step-by-step deduction of our method. Using a Strang splitting
in time (Section 2.1) and a Hermite–Galerkin ansatz in space (Section 2.2), we first show
what exactly necessitates the efficient procedure that is afterwards devised in Sections 2.3
– 2.6. Section 2.7 contains a description of the overall time-stepping procedure.

2.1 Strang splitting in time

For an abstract evolution equation of the form
\[
   u_t(t) = A u(t) + B(u(t),t), \quad t \geq 0, \quad u(0) \text{ given},
\]
where \( A \) is a linear and \( B \) is a nonlinear operator, we consider a splitting ansatz built
from the evolution equations
\[
   \begin{cases}
   u^A_t(t) = A u^A(t), \\
   u^A(0) = u^A_0 \text{ given}, \\
   u^B(t) = B(u^B(t),t), \\
   u^B(0) = u^B_0 \text{ given},
\end{cases}
\]
(2)
with exact solutions and evolution operators
\[ u^A(t) = \Phi^A(t)[u^A_0], \quad u^B(t) = \Phi^B(t)[u^B_0], \quad t \geq 0, \]
respectively; see the review article [19] for a general overview of splitting methods. E.g., the Strang or symmetric Lie–Trotter splitting scheme reads
\[ u(t_{n+1}) \approx u_{n+1} = \Phi^A(h/2)[\Phi^B(h)[\Phi^A(h/2)[u_n]]], \quad n \geq 0, \quad t_n = nh, \]
and \( u_0 = u(0) \). In our case, \( u \) represents the unknown wave function \( \psi \), and
\[ A = -iD, \quad B(\psi^B) = -i(W\psi^B + |\psi^B|^2\psi^B), \]
where \( \psi^A \) and \( \psi^B \) denote the respective exact solutions.

### 2.2 Spectral discretization in space

Tensor-products of univariate Hermite functions,
\[ \varphi_k(x) = \prod_{\alpha=1}^d \varphi_{k_\alpha}(x_\alpha), \quad k = (k_1, \ldots, k_d) \in \mathbb{N}^d, \]
form an orthonormal basis of \( L^2(\mathbb{R}^d) \) with respect to the standard \( L^2 \)-inner product denoted by \((\cdot, \cdot)\),
\[ (\varphi_j, \varphi_k) = \int_{\Omega} \varphi_j \varphi_k dx = \delta_{jk}, \quad j, k \in \mathbb{N}^d, \]
where, in 1D,
\[ \varphi_k(x) = f_k H_k(x) \exp(-x^2/2), \quad f_k = \left(2^k k! \pi^{1/2}\right)^{-1/2}, \quad k \geq 0, \]
and \( H_k \) is the \( k \)-th order Hermite polynomial. Analogously, we use the notation \( H_1(x) = \prod_{\alpha=1}^d H_{k_\alpha}(x_\alpha) \) throughout the paper. An important property of the univariate Hermite functions is the three-term recurrence relation
\[ x\varphi_k(x) = \sqrt{\frac{k+1}{2}} \varphi_{k+1}(x) + \sqrt{\frac{k}{2}} \varphi_{k-1}(x), \quad k \geq 0, \quad (3) \]
where we set \( \varphi_{-1}(x) = 0 \). The efficient evaluation procedures as given in later sections crucially make use of (3). See, e.g., the standard reference [1], Section 22, or the quantum dynamics-related books [18], Chapter III.1.1, and [26], Section 7.7, for these facts.

We search for approximations
\[ \psi^A(x, t) \approx \psi^A_h^k(x, t) = \sum_{k \in K} c^k(t) \varphi_k(x) \in \text{span} \{ \varphi_k | k \in K(d, K) \}, \]
and analogously for $\psi^B$, to the solutions of the split equations (2), with a set of multi-
indices

$$K(d, K) = \left\{ k \in \mathbb{N}^d \mid 0 \leq k_\alpha \leq K \right\} \quad (4)$$

defined by some threshold $K \in \mathbb{N}$. For the sake of simplicity, we often omit the arguments $d$ and $K$ in $K(d, K)$. Using a Galerkin ansatz, we determine the unknown coefficients $c^A_k(t)$ and $c^B_k(t)$ by the orthogonality requirements (omitting all arguments)

$$\left( \phi_j, i\dot{\psi}^A_K \right) = \left( \phi_j, i\dot{\psi}^B_K - (W + |\psi^B_K|^2)\psi^B_K \right) = 0 \quad \forall \ j \in K. \quad (5)$$

Slightly overloading the notation, we let $x_\alpha$ denote also the position operator w.r.t. the $\alpha$th coordinate. Using the eigenfunction relation for the Hermite functions,

$$D \varphi_k = \sum_{\alpha=1}^d \left( k_\alpha + \frac{1}{2} \right) \varphi_k$$

and orthonormality of the basis, eq. (5) can equivalently be written as

$$i\dot{c}^A_j = \sum_{\alpha=1}^d \left( j_\alpha + \frac{1}{2} \right) c^A_j, \quad i\dot{c}^B_j = \sum_{k \in K} \left( \varphi_j, (W + |\psi^B_K|^2)\varphi_k \right) c^B_k. \quad (6)$$

Setting $c^A = (c^A_k)_{k \in K}$, and analogously for $c^B$, this yields the spatially discrete systems of ODEs

$$i\dot{c}^A = Dc^A, \quad i\dot{c}^B = (W + P)c^B \quad (7)$$

with a diagonal matrix $D = \text{diag}_{k \in K}(\sum_{\alpha=1}^d (k_\alpha + \frac{1}{2}))$ and $(|K| \times |K|)$-matrices

$$W_{jk}(t) = (\varphi_j, W(\cdot, t)\varphi_k), \quad P_{jk}(t) = (\varphi_j, |\psi^B_K(\cdot, t)|^2 \varphi_k), \quad j, k \in K. \quad (8)$$

The left equation from (7) is trivially solved exactly by

$$c^A(t) = \Phi^A_{\text{discr}}(t)c^A(0) = \text{diag}_{k \in K} \left( \exp \left( -it \sum_{\alpha=1}^d \left( k_\alpha + \frac{1}{2} \right) \right) \right) c^A(0). \quad (9)$$

Since only the discretization $\Phi^B_{\text{discr}}$ of $\Phi^B$ necessitates some efforts, we shall skip the superscripts $A$ and $B$ in the subsequent sections.

As for the right equation of (7), during time-propagation, one typically needs to compute the action of the matrices $W$ and $P$ on vectors. Since the matrices are of size $K^d$, this would necessitate a number of operations that scales as $\sim K^{2d}$ in every time step—left aside the costs for an assembly of $W(t)$ and $P(t)$ (in every time step). In many practical computations, this is prohibitively expensive. As for the product $Wv$ with a vector $v \in \mathbb{C}^{|K|}$, an efficient procedure that scales only linearly with $|K|$ and goes without an explicit assembly of $W$ has been presented in [6] and further developed
in [7]. The basic idea is to first approximate the potential $W$ by a polynomial, and then to formally insert matrix representations of the position operations with respect to every single coordinate into this polynomial approximate. Using orthogonality of the underlying Galerkin basis together with the defining three-term recurrence (3), the action of these coordinate matrices on vectors can be done in linear time only, which the overall computation essentially boils down to. An adaptation of this fast algorithm to the somewhat more intricate product $Pv$ is the topic of the present work. The subsequent sections shall exclusively be dedicated to this task, which we perform in four steps. We briefly turn back to the computation of $Wv$ in Section 2.7.

From now on, let $v$ denote a generic vector in $\mathbb{C}^{\left|K\right|}$.

### 2.3 Step 1: Approximation of the squared modulus

The first step in devising the fast algorithm is a suitable approximation of $|\psi_K|^2$. The following procedure shall turn out as a reasonable move: Let $L(d,L)$ be a set of multi-indices (cf. (4)) and let $P_L$ denote the $L^2$-orthogonal projection onto the span of $\varphi_l$, $l \in L$. We approximate $|\psi_K(\cdot,t)|^2 \approx P_L|\psi_K(\cdot,t)|^2 = \sum_{l \in L} d_l(t) \varphi_l$, $d_l(t) = (\varphi_l, |\psi_K(\cdot,t)|^2)$. By the fact that the Hermite functions are real-valued, we can compute $d_l(t) = (\varphi_l, |\psi_K(\cdot,t)|^2) = (\sum_k \tilde{c}_k(t) \varphi_k)(\sum_j c_j(t) \varphi_j) = \sum_k \tilde{c}_k(t) \sum_j (\varphi_k, \varphi_l \varphi_j)c_j(t)$, or, equivalently,

$$ d_l(t) = c^\ast(t) P^{(l)} c(t) \quad \forall l \in L, \quad t \geq 0, \quad (10) $$

with time-independent triple-product matrices

$$ P^{(l)}_{jk} = (\varphi_j, \varphi_l \varphi_k), \quad j, k \in K, \quad l \in L. \quad (11) $$

The coefficients $d_l(t)$ can thus be computed from $c(t)$ and $P^{(l)}c(t)$, and the product of $P(t)$ times a vector $v$ can approximately be done according to

$$ P(t)v \approx \sum_{l \in L} d_l(t) P^{(l)}v = \sum_{l \in L} c^\ast(t) P^{(l)}c(t) P^{(l)}v. \quad (12) $$

Thus, we aim for an efficient way to compute all products $c^\ast(t)P^{(l)}c(t)$ and $P^{(l)}v$ (in every time step). If this can be done in $O(|L||K|)$, the overall costs for a computation of $Pv$ will be of this very size. And if the index set thresholds satisfy $L \ll K$, this scaling is essentially linear with $|K|$. In Step 4, we shall eventually come up with efficient procedures for this task. The following Steps 2 and 3 contain the necessary building blocks.
2.4 Step 2: Factorization of triple-product matrices

An efficient procedure for the product (12) necessitates a suitable factorization of \( P^{(l)} \), which we deduce in this section.

We start with some general considerations about Gauß–Hermite quadrature, which, in 1D, is the instance of Gaussian quadrature that uses \( e^{-x^2} \) as a weight function and approximates integrals of weighted functions over the whole real line, see, e.g., [13], Chapter 3.2. The nodes \( \xi_m \) are the zeros of, say, \( H_{M+1} \) with corresponding weights \( w_m \), and the resulting quadrature formula \( (w_m, \xi_m)_{m=0}^M \) is exact for polynomials of degree \( \leq 2M+1 \).

In higher dimensions, we define

\[
\xi_m = (\xi_{m,1}, \ldots, \xi_{m,d}), \quad \omega_m = \prod_{\alpha=1}^d \omega_{m,\alpha} = \prod_{\alpha=1}^d w_{m,\alpha} e^{\xi_{m,\alpha}^2}, \quad m \in \mathcal{M}(d,M),
\]

which constitutes a product of 1D Gauß–Hermite quadrature formulas with \( M+1 \) nodes in every direction. We speak of \((M+1)\)-nodes full-product quadrature. This \( d \)-dimensional quadrature rule is exact for polynomials of degree up to \( 2M+1 \) in each variable, denoted henceforth by a subscript or a superscript “quad”.

Let us now define the matrices

\[
U_{jk} = \sqrt{\omega_j} \varphi_k(\xi_j) = \prod_{\alpha=1}^d \sqrt{\omega_{j,\alpha}} \varphi_{k,\alpha}(\xi_{j,\alpha}), \quad j, k \in \mathcal{K}, \alpha = 1, \ldots, d.
\]

Using the exactness of Gaussian quadrature with \( M = K \), it is readily checked that \( U \) is orthogonal. Consider an arbitrary multivariate polynomial \( q : \mathbb{R}^d \to \mathbb{R} \) and its matrix representation

\[
Q^{\text{quad}}_{jk} = (\varphi_j, q \varphi_k)_{\text{quad}}, \quad j, k \in \mathcal{K},
\]

with entrywise \((K+1)\)-nodes full-product quadrature. Orthogonality of \( U \) gives

\[
Q^{\text{quad}} = U^T \text{diag}_{m \in \mathcal{K}}(q(\xi_m)) U = q(X),
\]

where the right-hand side denotes formal insertion of the coordinate matrices \( X^{(\alpha)} \) in place of \( x_{\alpha} \) into \( q \); see [6] for a proof. As a particular instance of (15), we find

\[
U^T \exp \left( \text{diag}_{m \in \mathcal{K}} \left( -\frac{1}{2} \sum_{\alpha=1}^d \xi_{m,\alpha}^2 \right) \right) U = \exp \left( -\frac{1}{2} \sum_{\alpha=1}^d (X^{(\alpha)})^2 \right).
\]

Setting \( f_1 = f_{l,1} \cdots f_{l,d} \), we can then approximate \( P^{(l)} \) by quadrature and compute

\[
P^{(l)} \approx P^{(l)}_{\text{quad}} = \left( (\varphi_j, \varphi_l \varphi_k)_{\text{quad}} \right)_{j, k \in \mathcal{K}} = f_1 \left( (\varphi_j, H_l \varphi_0 \varphi_k)_{\text{quad}} \right)_{j, k \in \mathcal{K}}
\]

\[
= f_1 U^T \text{diag}_{m \in \mathcal{K}} \left( H_l(\xi_m) \varphi_0(\xi_m) \right) U
\]

\[
= f_1 U^T \text{diag}_{m \in \mathcal{K}} \left( H_l(\xi_m) \right) UU^T \text{diag}_{m \in \mathcal{K}} \left( \varphi_0(\xi_m) \right) U
\]

\[
= f_1 H_l(X) \exp \left( -\frac{1}{2} \sum_{\alpha=1}^d (X^{(\alpha)})^2 \right).
\]
which is the desired factorization. The products \( e^*(t)P(t)c(t) \) and \( P(t)v \) can thus be done sequentially: an efficient procedure for the action of (the exponential of) sums of squared coordinate matrices is devised in Step 3; given the result vector, the action of \( H_l(X) \) is then computed as explained in Step 4. Both procedures boil down to an efficient computation of the action of \( X^{(a)} \) on a vector.

2.5 Step 3: Direct operation procedure and matrix exponentials

In this section, we devise an efficient procedure to compute

\[
w = \exp \left( -\frac{1}{2} \sum_{\alpha=1}^{d} (X^{(a)})^2 \right) v. \tag{16}
\]

The product \( X^{(a)}v \) can be computed directly, i.e., without assembling the matrix, and in linear time only. Using the recurrence relation (3) together with the orthogonality of the univariate Hermite functions, we find

\[
(X^{(a)}v)_j = \sum_{k \in K} \sqrt{\frac{j_\alpha + 1}{2}} (\varphi_j, \varphi_{k+e_\alpha}) + \sum_{k \in K} \sqrt{\frac{j_\alpha}{2}} (\varphi_j, \varphi_{k-e_\alpha}) \tag{17}
\]

for all \( j \in K \), where \( e_\alpha \) is the \( \alpha \)th unit vector, and \( v_{j-e_\alpha} = 0 \) if \( j_\alpha = 0 \) and \( v_{j+e_\alpha} = 0 \) if \( j_\alpha = K \). Thus, \( X^{(a)}v \) is obtained via (17) in \( O(|K|) \) operations, and \( -\frac{1}{2} \sum_{\alpha=1}^{d} (X^{(a)})^2 v \) is obtained by looping over \( \alpha \) and using (17) in \( O(d|K|) \) operations.

For a general \( N \times N \) Hermitian matrix \( A \) and a vector \( v \in \mathbb{C}^N \) of unit norm, the action \( \exp(A)v \) of the corresponding matrix exponential on \( v \) can be approximated using a Galerkin approach in combination with the Lanczos method; see [20] and the references therein: The \( m \)-step Hermitian Lanczos process generates recursively the basis \( V_m = (v_1 | \ldots | v_m) \in \mathbb{C}^{N \times m} \) and a tridiagonal coefficient matrix \( T_m \in \mathbb{C}^{m \times m} \) such that \( T_m = V_m^*AV_m \), starting from \( v_1 = v \). This requires \( m \) multiplications of \( A \) with a Lanczos vector, where \( m \ll N \). The matrices \( V_m \) and \( T_m \) are then used to approximate \( \exp(A)v \approx V_m \exp(T_m)e_1 \), \( e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^m \).

This reduces the problem to a diagonalization of the small matrix \( T_m \).

In our setting, we have \( A = -\frac{1}{2} \sum_{\alpha=1}^{d} (X^{(a)})^2 \) and \( N = |K| \). In each step of the Lanczos process, the action of \( A \) on a vector is realized efficiently via (17) in linear time only, and the overall Lanczos process takes \( O(md|K|) \) operations.

2.6 Step 4: Efficient matrix-vector products

By the above considerations, the task of computing \( P(t)v \) translates to

\[
P(t)v \approx \sum_{l \in \mathcal{L}} d_l(t)P^{(l)}v = \sum_{l \in \mathcal{L}} d_l(t)f_lH_l(X)w, \quad w = \exp \left( -\frac{1}{2} \sum_{\alpha=1}^{d} (X^{(a)})^2 \right) v.
\]

8
Once \( \mathbf{w} \) has been obtained via the procedure given in Step 3, a slight modification of the fast algorithm as presented in [7] can be used to compute \( \sum_{l \in \mathcal{L}} d_l(t) f_l H_l(\mathbf{X}) \mathbf{w} \). Consider the pseudocode formulation given in Algorithm 1: The first term in Hermite polynomial expansion is the one corresponding to \( l = 0 \), which is also the initial term in this termwise procedure. Starting from \( l = 0 \) and \( \alpha = 1 \), to compute \( f_l H_l(\mathbf{X}^\alpha) \mathbf{w} \), we carry out a single step of the 1D Hermite recurrence

\[
H_0(\mathbf{X}^\alpha) \mathbf{w} = \mathbf{w}, \quad H_1(\mathbf{X}^\alpha) \mathbf{w} = 2\mathbf{X}^\alpha H_0(\mathbf{X}^\alpha) \mathbf{w},
\]

in combination with (17) (lines 3 to 11) and then carry on recursively with

\[
l \leftarrow l = (l_1, \ldots, l_\alpha - 1, l, l_\alpha + 1, \ldots, l_d)
\]

instead of \( l \) and with \( \alpha + 1 \) instead of \( \alpha \) (lines 13, 15). Having reached \( \alpha = d \), we multiply with \( d_l(t) \), where \( l \) indicates the currently considered term in the Hermite expansion, and sum up the result (line 17). Since, for every choice of \( l \in \mathcal{L} \), the 1D Hermite recurrence (18) is invoked exactly once, this yields overall computational costs of only

\[
\mathcal{O}(|\mathcal{L}||\mathcal{K}|).
\]

The computation of \( \mathbf{c}^* (t) \mathbf{P}^{(l)} \mathbf{c}(t), l \in \mathcal{L} \), is done in almost exactly the same manner: In line 17 of Algorithm 1, instead of adding up successive terms, we have to compute \( \mathbf{c}^* (t) \tilde{\mathbf{y}} \) and store in the \( l \)th entry of an output array of size \( \mathcal{O}(|\mathcal{L}|) \).

### 2.7 Time-stepping: overview

Having chosen spatial discretization parameters \( K, L \) and \( R \) for the multi-index sets \( \mathcal{K}(d, K), \mathcal{L}(d, L) \), and \( \mathcal{R}(d, R) \) that underly the Hermite and Chebyshev approximations

\[
\psi^A \approx \psi^A_K, \quad \psi^B \approx \psi^B_K, \quad |\psi^B_K|^2 \approx \mathcal{P}_L |\psi^B_K|^2, \quad W \approx W^{int},
\]

respectively, we propagate the systems (7) employing the Strang splitting scheme as given in Section 2.1, viz.,

\[
\mathbf{c}(t_{n+1}) \approx \mathbf{c}_{n+1} = \Phi^A_{\text{discr}}(h/2) (\Phi^B_{\text{discr}}(h) (\Phi^A_{\text{discr}}(h/2) (\mathbf{c}_n)), \quad n \geq 0,
\]

where \( \Phi^A_{\text{discr}} \) is the exact evolution of the harmonic oscillator part (A), which is trivially computable by (9) with linearly scaling cost; and \( \Phi^B_{\text{discr}} \) is a suitable approximation to the propagator \( \Phi^B \) of the remainder part (B)—polynomial interpolation of \( W \), projection of the squared modulus, and quadrature taken into account. To illustrate time-propagation of the latter part, we choose, e.g., the classical 4th-order Runge–Kutta method. In each time-step, matrix-vector products with the matrix \( \mathbf{P} \) as required for the computation of the Runge–Kutta increments are done using Algorithm 1 from Section 2.6 at costs
Algorithm 1: Fast algorithm for

\[ y = \sum_{l \in \mathcal{L}} d_l f_l(X) w \]

for a given vector \( w \in \mathbb{C}^{|K|} \) and given coefficients \( (d_l)_{l \in \mathcal{L}} \) starting from \( y = 0 \), \( \alpha = 1 \), and \( l = (0, \ldots, 0) \in \mathbb{N}^d \).

1 function \( y = \text{fastalgorithm}(w, y, \alpha, 1) \)
2 for \( l = 0 \) to \( L \) do
3     if \( l = 0 \) then
4         \( w_+ := f_0 X^{(\alpha)} w \)
5     else if \( l = 1 \) then
6         \( w_+ := f_1 X^{(\alpha)} w \)
7     else
8         \( \text{temp} := w_+ \)
9         \( \text{temp}_2 := X^{(\alpha)} w_+ \)
10     \( w_+ := f_l (2 \text{temp}_2 - 2lw_-) \)
11     \( w_- := \text{temp} \)
12     \( \tilde{y} := \begin{cases} w_-, & l = 0, \\ w_+, & \text{else}, \end{cases} \)
13     \( l := 1 \leftarrow l \)
14     if \( \alpha < d \) then
15         \( y := \text{fastalgorithm}(\tilde{y}, y, \alpha + 1, 1) \)
16     else
17     \( y := y + d_l \tilde{y} \)

of \( O(|\mathcal{L}| |K|) \) each. The unknown expansion coefficients of \( \mathcal{P}_d L|\mathcal{L}\mathcal{K}|(\cdot, t_n)^2 \) are computed from \( \Phi_{\text{discr}}(h/2)(c^n) \) using essentially the same procedure (see the remarks at the end of Section 2.6).

The action of the matrix representation \( W \) of \( W \) is done in \( O(|\mathcal{R}| |K|) \) using the fast algorithm due to [7]. For the fast algorithm to be applicable to \( W \), we first have to approximate \( W \) by some interpolation polynomial \( W^{\text{int}} \). If \( W \) depends on \( t \), this needs to be repeated in every time step. In the below pseudocode, \( W = W^{\text{int}} \) is tacitly assumed to be a polynomial when writing \( W^{\text{int}} \) in place of \( W \). The procedure from [7] is then the same as Algorithm 1, but for an appropriate polynomial basis (say, tensor-products of Chebyshev polynomials) in place of Hermite polynomials, and with \( \mathcal{L}(d, L) \) replaced by an appropriate index set \( \mathcal{R}(d, R) \) and corresponding polynomial approximation coefficients \( w_r \) in place of \( d_l \).

Given a choice of time step size \( h \) and a choice of number of Lanczos steps \( m \) for the matrix exponentials as occurring in (16), propagation is explicitly done as given in
Algorithm 2: The overall computational costs for a single time step scale as
\[ O((|L| + |R| + md)|K|). \]
If \(|L|, |R| \ll |K|\), this is essentially linear in \(|K|\).
Alternatively, for the remainder part, we can employ another Lanczos-based Galerkin approximation to the matrix exponential as required, e.g., when using a Magnus integrator; see the review article [5] on Magnus integrators.

3 Error due to spatial discretization

The present section is dedicated to an analysis of the overall error due to spatial discretization. This comprises not only the approximation of the wave function \(\psi\) by a Galerkin approach, but also the errors due to projecting the squared modulus of \(\psi\) and...
interpolating $W$ (as required for the fast algorithms to be applicable) as well as the errors due to the above efficient evaluation procedures based on the coordinate matrices $X^{(\alpha)}$. Since the latter procedures are equivalent to a suitable choice of quadrature, there is an error contribution from Gauß–Hermite quadrature. But for the Galerkin error itself, all these error contributions are specific features of our very method and therefore deserve our attention. Apart from that, an overview of error studies for existing, standard full discretizations of nonlinear Schrödinger equations can be found in [11, 25]. In particular, we are not concerned with temporal errors. To begin with, we need to clarify some notations and state all necessary assumptions.

**Index sets and their relations.** We make use of the multi-index sets $K(d, K)$, $L(d, L)$, and $R(d, R)$, where the former contains the indices for the Galerkin basis as explained in Section 2.2, and the latter index sets are used for a Hermite and Chebyshev approximation of the squared modulus of the wave function and the potential, respectively. In the following, we shall comment on this in more detail.

We assume that the potential $W$ can be approximated by a polynomial with significantly fewer terms of significantly lower degree than the wave function wherever the latter does not essentially vanish. Formally, a Chebyshev interpolate of the potential $W$ is given by

$$ W(x, t) \approx W^{\text{int}}(x, t) = \sum_{r \in R} w_r(t) T_r(x/\bar{R}) = \sum_{r \in R} w_r(t) \prod_{\alpha=1}^{d} T_{r_{\alpha}}(x_{\alpha}/\bar{R}), \quad x \in \Omega_R = [-\bar{R}, \bar{R}]^d, \quad \text{where } \psi(\cdot, t) \text{ is assumed to have essential support within the cube } \Omega_R, \text{ and } R \ll K. $$

We also choose $L$ such that $L \ll K$. These restrictions on $K, L, \text{ and } R$ shall become clear from the subsequent error analysis.

**Projection estimate and norms.** Again, for any function $\chi \in L^2(\mathbb{R}^d)$, let $P_K \chi$ be the $L^2$-orthogonal projection of $\chi$ onto the span of $\varphi_k$, $k \in K$, and $P^\perp_K \chi$ be its orthogonal complement. Introducing the seminorm

$$ |\chi|_k^2 = \sum_{|k|_\infty = k} \left\| 2^{-d/2} \prod_{\alpha=1}^{d} \left( x_{\alpha} + \frac{\partial}{\partial x_{\alpha}} \right) \chi \right\|, \quad k \in \mathbb{N}, $$

we can deduce the following projection estimate: For every fixed integer $k$ and for all $\chi \in L^2(\mathbb{R}^d)$ such that $|\chi|_k$ is bounded,

$$ \| \chi - P_K \chi \| \leq C(d, k) K^{-k/2} |\chi|_k, $$

where the constant $C(d, k)$ depends on $d$ and $k$ only; cf. [23]. We let $\| \cdot \|$ denote the standard $L^2(\mathbb{R}^d)$-norm, or the standard Euclidean norm in case of a vector-valued argument. For matrix-valued arguments, the norm is understood as the spectral matrix norm.
Decomposition of wave function. To facilitate the error analysis, we introduce the Hermite expansions

\[ \psi(x, t) = \sum_{k \in \mathbb{N}^d} u_k(t) \varphi_k(x), \quad \psi^{\text{int}}(x, t) = \sum_{k \in \mathbb{N}^d} u_k^{\text{int}}(t) \varphi_k(x) \]

together with the truncated Hermite expansion

\[ \psi_{\mathcal{K}}(x, t) = \sum_{k \in \mathcal{K}} c_k(t) \varphi_k(x). \]

For the sake of simplicity, we henceforth omit explicit time-dependency whenever possible. The coefficients \( u_k \) are chosen such that \( \psi \) solves the original weakly formulated problem,

\[ i(\varphi, \psi_t) = (\varphi, D\psi) + (\varphi, W\psi(\cdot, t)) + (\varphi, |\psi|^2 \psi) \tag{22} \]

for all \( \varphi \in H^1(\Omega) \) and for all \( t \geq 0 \), with initial data \( \psi(x, 0) = \psi^0(x) \). Analogously, \( \psi^{\text{int}} \) is the solution to the same problem, but with \( W \) replaced by \( W^{\text{int}} \) and with \( |\psi|^2 \) replaced by the Hermite projection \( \mathcal{P}_{\mathcal{L}}|\psi^{\text{int}}|^2 \), i.e.

\[ i(\varphi, \psi^{\text{int}}_t) = (\varphi, D\psi^{\text{int}}) + (\varphi, W^{\text{int}}\psi^{\text{int}}) + (\varphi, \mathcal{P}_{\mathcal{L}}|\psi^{\text{int}}|^2 \psi^{\text{int}}) \tag{23} \]

for all \( \varphi \in H^1(\Omega) \) and for all \( t \geq 0 \), with the same initial data \( \psi^{\text{int}}(x, 0) = \psi^0(x) \). We recall the definition of the matrix representations of \( W^{\text{int}} \) and \( \mathcal{P}_{\mathcal{L}}|\psi^{\text{int}}|^2 \),

\[ W^{\text{int}}_{jk}(t) = (\varphi_j, W^{\text{int}}(\cdot, t) \varphi_k), \quad \mathcal{P}_{\mathcal{L}}^{\text{int}}(t) = (\varphi_j, \mathcal{P}_{\mathcal{L}}|\psi^{\text{int}}|^2(\cdot, t) \varphi_k), \quad j, k \in \mathcal{K}, \ t \geq 0. \tag{24} \]

The function \( \psi_{\mathcal{K}} \) is the Galerkin approximation to \( \psi^{\text{int}} \), where entrywise \((K+1)\)-nodes full-product Gauß–Hermite quadrature on the right-hand side has been taken into account,

\[ i(\varphi_j, (\psi_{\mathcal{K}})_t) = (\varphi_j, D\psi_{\mathcal{K}}) + (\varphi_j, W^{\text{int}}(\psi_{\mathcal{K}})_{\text{quad}} + (\varphi_j, \mathcal{P}_{\mathcal{L}}|\psi^{\text{int}}|^2(\psi_{\mathcal{K}})_{\text{quad}} \tag{25} \]

for all \( j \in \mathcal{K} \) and \( t \geq 0 \) or, equivalently,

\[ ic = Dc + W_{\mathcal{L}}c + \mathcal{P}_{\mathcal{K}}c, \tag{26} \]

where we set \( c(t) = (c_k(t))_{k \in \mathcal{K}} \), \( D = \text{diag}_{k \in \mathcal{K}} \left( \sum_{\alpha=1}^d \left( k_\alpha + \frac{1}{2} \right) \right) \), and

\[ (W_{\mathcal{K}})_{jk}(t) = (\varphi_j, W^{\text{int}}(\cdot, \cdot) \varphi_k)_{\text{quad}}, \quad (\mathcal{P}_{\mathcal{L}})_{jk}(t) = (\varphi_j, \mathcal{P}_{\mathcal{L}}|\psi^{\text{int}}|^2(\cdot, \cdot) \varphi_k)_{\text{quad}}. \tag{27} \]

As an initialization, we choose \( \psi_{\mathcal{K}}(x, 0) = \mathcal{P}_{\mathcal{K}}\psi^0(x) \).

Interpolating \( |\psi^{\text{int}}|^2 \). Finally, let \( \mathcal{T}_{\mathcal{L}}|\psi^{\text{int}}|^2 \) denote the Chebyshev interpolate of \( |\psi^{\text{int}}|^2 \) over the cube

\[ \Omega_{\mathcal{L}} = [-\sqrt{2(L+1)}-1, \sqrt{2(L+1)}+1]^d, \]

which is defined in analogy to \((20)\), using \( \mathcal{L}(d, L) \) instead of \( \mathcal{R}(d, R) \) as the underlying index set.

The main result can then be formulated as follows:
Theorem 1. Let $\psi, \psi_{\text{int}} \in H^2(\Omega)$, and $\psi_{\mathcal{K}}$ denote the solutions to (22), (23), and (25), respectively. We assume that there is some $k \in \mathbb{N}$ such that $|\psi_{\text{int}}(\cdot, t)|_k$ is bounded. In addition, for some given accuracy parameters $\varepsilon_R$, $\varepsilon_L$, and $\varepsilon_{\text{cheb}}$, we assume

$$
\|(W - W_{\text{int}})(\cdot, t)\| \leq \varepsilon_R
$$

and

$$
\|(\psi_{\text{int}}^2 - \mathcal{P}_L|\psi_{\text{int}}^2|)(\cdot, t)\| \leq \varepsilon_L, \quad \max_{x \in \Omega_L} \left| (|\psi_{\text{int}}^2 - \mathcal{I}_L|\psi_{\text{int}}^2|(x, t)) \right| \leq \varepsilon_{\text{cheb}}
$$

for all $t \geq 0$. The error due to spatial discretization is then given as

$$
\|(\psi - \psi_{\mathcal{K}})(\cdot, t)\| \leq (\varepsilon_R + \varepsilon_L) t \|\psi(\cdot, 0)\|
$$

$$
+ C_1 (K - R)^{-k/2} \int_0^t |\psi_{\text{int}}(\cdot, \tau)|_k d\tau, \quad t \geq 0,
$$

$$
+ C_2 (K - L)^{-k/2} \int_0^t |\psi_{\text{int}}(\cdot, \tau)|_k d\tau + (\varepsilon_L + \varepsilon_{\text{cheb}}) t \|\psi_{\text{int}}(\cdot, 0)\|,
$$

where

$$
C_1 = C_1(W_{\text{int}}, d, k), \quad C_2 = C_2(\mathcal{P}_L|\psi_{\text{int}}^2|, d, k).
$$

Comments. We briefly comment on the estimate given in Theorem 1 and its implications. First, it makes clear the significance of the choices

$$
L \ll K, \quad R \ll K
$$
in order to guarantee convergence of the respective terms on its right-hand side w.r.t. $K$. Second, in order to provide for a decent approximation by the projection $\mathcal{P}_L|\psi_{\text{int}}^2|$ (and by the interpolate $\mathcal{I}_L|\psi_{\text{int}}^2|$), the necessary choice $L \ll K$ then yields the requirement that the squared modulus $|\psi_{\text{int}}^2|$ be significantly smoother than $\psi_{\text{int}}$ itself. Formally, if $\|\psi_{\text{int}}^2\|_l$ is bounded for some $l \in \mathbb{N}$ with $l$ being sufficiently larger than $k$, we have

$$
\varepsilon_L \sim C(d, l) L^{-l} \|\psi_{\text{int}}^2\|_l,
$$

by the projection estimate (21), and the contribution from projecting the squared modulus does not dominate the overall error. Analogously, the factors $\varepsilon_{\text{cheb}}$ and $\varepsilon_R$ are of negligible sizes $\sim L^{-p}$ and $\sim R^{-r}$, respectively, for some $p, r$ if $|\psi_{\text{int}}^2|$ and $W$ are only sufficiently regular; see, e.g., [8] for a detailed theory of polynomial approximation.

Proof. The proof consists of four steps. As for the easily manageable error contributions, an appropriate decomposition allows to use well-known techniques to be found in [18] (Step 1) and in [27] (Step 2) together with standard projection estimates. For the more intricate parts, we invoke a recently devised projection technique due to [7] (Step 3) and a suitable adaptation thereof (Step 4). The overall structure of the proof is due to the error decomposition

$$
\psi - \psi_{\mathcal{K}} = (\psi - \psi_{\text{int}}) + (\psi_{\text{int}} - \mathcal{P}_K\psi_{\text{int}}) + (\mathcal{P}_K\psi_{\text{int}} - \psi_{\mathcal{K}}).
$$
The first and second terms, i.e., the error due to polynomial interpolation and approximation of the squared modulus as well as the standard projection error, are discussed in Step 1 of the subsequent proof. Steps 2 to 4 deal with the more intricate error contribution coming from the third term, which is the error due to quadrature. Throughout, let $C$ denote a generic constant.

**Step 1.** First, by subtracting (23) from (22), the error $e_{\text{int}} = \psi - \psi_{\text{int}}$ is seen to obey

$$i(\varphi, e_{\text{int}}) = (\varphi, (D + W_{\text{int}})e_{\text{int}}) + (\varphi, (W - W_{\text{int}})\psi)$$

$$+ (\varphi, \mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2 e_{\text{int}}) + (\varphi, (|\psi_{\text{int}}|^2 - \mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2)\psi).$$

(29)

Following [18], Thm. II.1.5, we use (29) with $\varphi = e_{\text{int}}$ and take the real part. Since $D$ is self-adjoint on $H^2(\mathbb{R}^d)$ and since $W_{\text{int}}$ and $\mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2$ are both real-valued, we find

$$\frac{d}{dt} \|e_{\text{int}}\| \leq \|(W - W_{\text{int}})\psi\| + \|(\psi_{\text{int}}|^2 - \mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2)\psi\|.$$ 

The second term in (28) is directly subsumed into the second term of the error estimate using the projection estimate (21).

**Step 2.** The third term is estimated following the ansatz due to [27]. Using (23) with $\varphi = \varphi_j, j \in \mathcal{K}$, and since

$$(\varphi_j, (\mathcal{P}_\mathcal{L}^{\perp} K\psi_{\text{int}})_t) = (\varphi_j, D\mathcal{P}_\mathcal{L}^{\perp} K\psi_{\text{int}}) = 0, \quad j \in \mathcal{K},$$

the vector $u_{\text{int}} = (u_k)_{k \in \mathcal{K}}$ is seen to satisfy

$$i\dot{u}_{\text{int}} = Du_{\text{int}} + W_{\text{int}}u_{\text{int}} + P_{\text{int}}u_{\text{int}} + w + p$$

(30)

with defects

$$w_j = (\varphi_j, W_{\text{int}}\mathcal{P}_\mathcal{L}^{\perp} K\psi_{\text{int}}), \quad p_j = (\varphi_j, \mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2\mathcal{P}_\mathcal{L}^{\perp} K\psi_{\text{int}}), \quad j \in \mathcal{K}.$$

As for the error $e_{\mathcal{K}} = u_{\text{int}} - c$, subtracting (26) from (30) yields

$$i\dot{e}_{\mathcal{K}} = (D + W_{\mathcal{K}})e_{\mathcal{K}} + P_{\mathcal{K}}e_{\mathcal{K}} + (W_{\text{int}} - W_{\mathcal{K}})u_{\text{int}} + (P_{\text{int}} - P_{\mathcal{K}})u_{\text{int}} + w + p.$$

Multiplying with $e_{\mathcal{K}}^*$ and taking the real part gets us

$$\frac{d}{dt} \|e_{\mathcal{K}}\| \leq \|(W_{\text{int}} - W_{\mathcal{K}})u_{\text{int}}\| + \|(P_{\text{int}} - P_{\mathcal{K}})u_{\text{int}}\| + \|w\| + \|p\|.$$ 

(31)

The defects are readily controlled via

$$\|w\|^2 \leq \|W_{\text{int}}\|^2\|\mathcal{P}_\mathcal{L}^{\perp} K\psi_{\text{int}}\|^2 \leq C(W_{\text{int}}, d, k)K^{-k}|\psi_{\text{int}}|^2,$$

$$\|p\|^2 \leq \|\mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2\|\|\mathcal{P}_\mathcal{L}^{\perp} K\psi_{\text{int}}\|^2 \leq C(\mathcal{P}_\mathcal{L}|\psi_{\text{int}}|^2, d, k)K^{-k}|\psi_{\text{int}}|^2;$$

see again the projection estimate (21). Estimating the first and second terms in (31) is more involved. This is done in Steps 3 and 4, respectively.
Step 3. The entrywise error \((W^{\text{int}} - W_\mathcal{K})_{jk}\) due to quadrature vanishes if \(|j + k|_\infty = \max_\alpha (j_\alpha + k_\alpha) \leq 2K + 1 - R\). Following [7], we define a diagonal projection matrix \(Q\),

\[
Q_{ij} = \begin{cases} 
1, & |j|_\infty > K + 1 - R, \\
0, & \text{else,}
\end{cases}
\]

The projection \(Q\) thus constitutes a link between the exactness of Gaussian quadrature and the polynomial degree of the integrand. Evidently, this requires \(W^{\text{int}}\) to be a polynomial in the first place. We can now write

\[
W^{\text{int}} - W_\mathcal{K} = Q(W^{\text{int}} - W_\mathcal{K})Q
\]

The formal extension of \(Q\) to an infinite matrix, with ones on the new diagonal elements, can be interpreted as the matrix representation of an operator \(Q : L^2 \to L^2\). This gives

\[
\|Qu^{\text{int}}\| = \|QP^{\text{int}}\| \leq \|Q\|\|P^{\text{int}}\| \leq C(d, k)(K - R)^{-k/2}|\psi^{\text{int}}|_k.
\]

Using Rayleigh quotients, the spectral matrix norm of \(Q\) is controlled via

\[
\|W^{\text{int}}\| = \sup_{\|w\| = 1} \sum_j w_j W_{jk} w_k = \sup_{\|w\| = 1} \left( \left[ \sum_{j \in \mathcal{K}} w_j \varphi_j \right], W^{\text{int}} \left[ \sum_{k \in \mathcal{K}} w_k \varphi_k \right] \right) \leq \max_{x \in \Omega} |W^{\text{int}}(x)|^2,
\]

and, using the factorization \(W_\mathcal{K} = U^T \text{diag}_{m \in \mathcal{K}}(W^{\text{int}}(\zeta_m))U\) with the unitary matrix \(U\), we find \(\|W_\mathcal{K}\| \leq \max_{m \in \mathcal{K}} |W^{\text{int}}(\zeta_m)|\). Altogether, this allows for

\[
\|(W^{\text{int}} - W_\mathcal{K})u^{\text{int}}\| \leq \|Q\|\|W^{\text{int}}\|\|Qu^{\text{int}}\| \leq C(W^{\text{int}}, d, k)(K - R)^{-k/2}|\psi^{\text{int}}|_k.
\]

Step 4. We cannot simply mimic the proceeding from Step 3 with \(P_{\mathcal{L}}|\psi^{\text{int}}|^2\) in place of \(W^{\text{int}}\) since the former is not a polynomial. This obstacle is overcome by decomposing

\[
P^{\text{int}} - P_\mathcal{K} = (P^{\text{int}} - P^{\text{cheb}}) + (P^{\text{cheb}} - P_\mathcal{K}),
\]

where

\[
P^{\text{cheb}}_{jk} = (\varphi_j, T_{\mathcal{L}}|\psi^{\text{int}}|^2 \varphi_k), \quad (P^{\text{cheb}}_\mathcal{K})_{jk} = (\varphi_j, T_{\mathcal{L}}|\psi^{\text{int}}|^2 \varphi_k)_{\text{quad}}, \quad j, k \in \mathcal{K}.
\]

The first and third terms occurring in (32) are again readily bounded as above,

\[
\|P^{\text{int}} - P^{\text{cheb}}\| \leq \max_{x \in \Omega} ||P_{\mathcal{L}}|\psi^{\text{int}}|^2 - T_{\mathcal{L}}|\psi^{\text{int}}|^2)(x)\|^2,
\]

\[
\|P^{\text{cheb}}_\mathcal{K} - P_\mathcal{K}\| \leq \max_{m \in \mathcal{K}} ||P_{\mathcal{L}}|\psi^{\text{int}}|^2 - T_{\mathcal{L}}|\psi^{\text{int}}|^2)(\zeta_m)||,
\]

using Rayleigh quotients and the factorizations

\[
P^{\text{cheb}}_\mathcal{K} = U^T \text{diag}_{m \in \mathcal{K}}(T_{\mathcal{L}}|\psi^{\text{int}}|^2(\zeta_m))U, \quad P_\mathcal{K} = U^T \text{diag}_{m \in \mathcal{K}}(P_{\mathcal{L}}|\psi^{\text{int}}|^2(\zeta_m))U,
\]

respectively. The second term from (32) allows for the very same treatment as in Step 3 with the polynomial \(T_{\mathcal{L}}|\psi^{\text{int}}|^2\) in place of \(W^{\text{int}}\). We find

\[
\|(P^{\text{cheb}} - P^{\text{cheb}}_\mathcal{K})u^{\text{int}}\| \leq C(d, k)(K - L)^{-k/2}|\psi^{\text{int}}|_k.
\]

Putting everything together, integrating, and noting that \(\|\psi\|, |\psi^{\text{int}}|\) are conserved then proves the claim. \(\square\)
Remark. In view of Step 4 of the above proof, one might be tempted to use $|\psi^{\text{int}}|^2 \approx \mathcal{T}_L|\psi^{\text{int}}|^2$ in deducing the algorithm instead of making use of $\mathcal{P}_L|\psi^{\text{int}}|^2$; cf. Section 2.3. However, this is not a viable option since the crucial relation (10) between the coefficients $c_k$ and $d_l$ involving the matrix $P^{(1)}$ would not be preserved.

4 Numerical experiments

To conclude with, we demonstrate our method by some numerical experiments. Throughout, plots are semilogarithmic. The vectors $u^{\text{int}}$ and $c_n$ are defined as in Sections 3 and 2.7, respectively. The function $\psi^{\text{int}}$ is defined as in Section 3, and $\psi_K^n = \sum_{k \in K} (c_n)_k \varphi_k$. Let $\| \cdot \|$ denote again the Euclidean vector norm or $L^2(\mathbb{R}^d)$-norm, respectively.

Since the error contribution from replacing the potential by a polynomial approximate has already been studied in [6] and is therefore of no interest to us, as a particular instance of (1), we use the example of a stretched Hénon–Heiles potential, viz.,

$$W(x) = \sum_{\alpha=1}^{d-1} \left[ (x_{\alpha}/\tilde{R})^2 (x_{\alpha+1}/\tilde{R}) - \frac{1}{3} (x_{\alpha+1}/\tilde{R})^3 \right], \quad x \in [-\tilde{R}, \tilde{R}]^d, \quad \tilde{R} = 16, \quad (33)$$

Throughout the subsequent experiments, the potential (33) is then represented exactly by its Chebyshev interpolate of the form (20) on a grid with only $R = 3$ nodes per direction, and we have $R \ll K$. Other choices of $W$ can be found in [2] and the references therein. The chosen initial data is a simple Gaussian,

$$\psi_0(x) = \varphi_0(x).$$

The following figures reveal that we can then rightly expect $|\psi_K|^2$ to be sufficiently smoother than the wave function itself; see Figure 3, in particular.

We propagate the split, spatially discrete system (7), as it has been brought into a form suitable for an application of the fast algorithm, as explained in Section 2.7. Our aim is to show numerically the expected order of convergence with respect to $h$ and, in view of Theorem 1, to illustrate the error behavior if $K$ and $L$ vary individually.

First, we fix $K$ and $L$, and let $h$ vary. We compare the errors

$$\|c_n - u^{\text{int}}(t_n)\|, \quad t_n = nh, \quad (34)$$

at $t_n = 1$ for different choices of time-step size $h$. To obtain a reference, we propagate (26) using Algorithm 2 with time-step size 1e-03. For $K = 25$, $L = 4$, and $d \in \{2, 3\}$, the expected order of convergence w.r.t. $h$ is visualized in Figure 1.

Next, we fix $h$ and $L$, and let $K$ vary. For $h = 1e-03$ (such that the error contribution from temporal discretization does not interfere), $L \in \{2, 4, 6\}$, and $d \in \{2, 3\}$, the errors

$$\|\psi_{K}^{1000} - \psi^{\text{int}}(\cdot, 1)\| \quad (35)$$

for different choices of $K$ are seen to decay as given in Figure 2. To obtain a reference, we propagate (26) with the same choice of $h$, and with $K = 50$. The $L^2$-error (35) is
<table>
<thead>
<tr>
<th>$h \downarrow$</th>
<th>$d = 2$</th>
<th>$d = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>ratio</td>
<td>error</td>
</tr>
<tr>
<td>1/10</td>
<td>4.335e-04</td>
<td>1.202e-03</td>
</tr>
<tr>
<td>1/20</td>
<td>1.079e-04</td>
<td>4.02</td>
</tr>
<tr>
<td>1/40</td>
<td>2.691e-05</td>
<td>4.01</td>
</tr>
<tr>
<td>1/80</td>
<td>6.693e-06</td>
<td>4.02</td>
</tr>
<tr>
<td>1/160</td>
<td>1.641e-06</td>
<td>4.08</td>
</tr>
</tbody>
</table>

Figure 1: Error (34) for $K = 25$, $L = 4$, $d \in\{2, 3\}$, and various choices of $h$ at time $t_n = 1$. The columns labeled “ratio” show the error ratios of two consecutive choices of time step. As expected for a second-order method, halving $h$ reduces the error by a factor of $\approx 4$.

<table>
<thead>
<tr>
<th>$K \downarrow$</th>
<th>$d = 2$</th>
<th>$d = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 2$</td>
<td>$L = 4$</td>
<td>$L = 6$</td>
</tr>
<tr>
<td>10</td>
<td>3.119e-04</td>
<td>6.144e-04</td>
</tr>
<tr>
<td>15</td>
<td>2.995e-05</td>
<td>6.958e-05</td>
</tr>
<tr>
<td>20</td>
<td>2.808e-06</td>
<td>7.739e-06</td>
</tr>
<tr>
<td>25</td>
<td>3.851e-07</td>
<td>1.193e-06</td>
</tr>
<tr>
<td>30</td>
<td>6.894e-08</td>
<td>2.329e-07</td>
</tr>
</tbody>
</table>

Figure 2: Error (35) for $h = 1e-03$, $L \in\{2, 4\}$, $d \in\{2, 3\}$, and various choices of $K$ at time $t_n = 1$. In view of Theorem 1, larger choices of $L$ yield larger errors w.r.t. $K$. In 2D, since $|\psi_K|^2$ is well-approximated even by small choices of $L$ (see the below Figure 3), the errors for $L = 4$ and $L = 6$ almost coincide, and likewise in 3D (not shown).

obtained by transformation of the propagated coefficients to function values on a spatial grid over $[-\tilde{R}, \tilde{R}]^d$ with 250 grid points per dimension.

Finally, we fix $h$ and $K$, and let $L$ vary. For $h = 1e-03$, $K = 25$, and $d \in\{2, 3\}$, the $L^2$-errors (obtained as above) for different choices of $L$ are shown in Figure 3. To obtain
a reference, we propagate (26) with the same choices of $h$ and $K$, and with $L=16$.

<table>
<thead>
<tr>
<th>$L$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d=2$</td>
<td>2.263e-02</td>
<td>2.588e-04</td>
<td>4.226e-08</td>
<td>4.076e-14</td>
</tr>
</tbody>
</table>

**Figure 3:** $L^2$-errors when comparing $\psi_{100}^K$ for different choices of $L$. We fix $h=1e-03$, $K=25$, and $d \in \{2, 3\}$. It is seen that, due to the high regularity of $|\psi_K|^2$, the errors decay rapidly with increasing $L$. In view of Figure 2, the choice $L=4$ makes the error due to projecting $|\psi_K|^2$ dominant, while the error (35) dominates in case $L=6$. The figures reveal that the above example allows for a choice of $L$ such that $L \ll K$ and still yields a decent overall approximation to the wave function.

Throughout, we have chosen $m=25$ as the number of Lanczos steps such that the error contribution from approximating the matrix exponential does not interfere; cf. Section 2.5. For practical computations, however, smaller choices of $m$ are viable; see the discussion in [15].

### 5 Conclusion

We have presented a Hermite–Galerkin spectral method for the Schrödinger equation with a cubic nonlinearity in arbitrary dimensions that essentially scales only linearly with the size of the tensorized Galerkin basis in every time step. The approach is based on two main ideas: First, we project the squared modulus of the wave function onto a subspace of the chosen approximation space and compute the corresponding coefficients from the coefficients of the wave function Galerkin approximate. The latter involves matrices of triple-products of Hermite functions. Second, the action of the representation matrix of the squared modulus on a vector can effectively be reduced to such triple-product matrices, and it can be computed efficiently by a modification of the fast algorithm due to [6, 7]—without assembling any matrices. Our method is restricted to the case when the squared modulus is considerably more regular than the wave function itself. The analysis of the spatial error makes use of a projection technique developed in [7].

**Acknowledgments:** The author is grateful to Christian Lubich and Balázs Kovács for valuable suggestions and for helpful discussions. Bernd Brumm is funded by the DFG Priority Program 1324 and is associated with DFG Research Training Group 1838.

### References


11.001.

1137/140981022.

umi-tuebingen.de/~brumm/.

978-3-540-30726-6.


