COMPUTING SEMI-CLASSICAL QUANTUM DYNAMICS WITH HAGEDORN WAVEPACKETS

ERWAN FAOU*, VASILE GRADINARU†, AND CHRISTIAN LUBICH‡

Abstract. We consider the approximation of multi-particle quantum dynamics in the semiclassical regime by Hagedorn wavepackets, which are products of complex Gaussians with polynomials that form an orthonormal L^2 basis and preserve their type under propagation in Schrödinger equations with quadratic potentials. We build a fully explicit, time-reversible time-stepping algorithm to approximate the solution of the Hagedorn wavepacket dynamics. The algorithm is based on a splitting between the kinetic and potential part of the Hamiltonian operator, as well as on a splitting of the potential into its local quadratic approximation and the remainder. The algorithm is robust in the semi-classical limit. It reduces to the Strang splitting of the Schrödinger equation in the limit of the full basis set, and it advances positions and momenta by the Störmer–Verlet method for the classical equations of motion. The algorithm allows for the treatment of multi-particle problems by thinning out the basis according to a hyperbolic cross approximation, and of high-dimensional problems by Hartree-type approximations in a moving coordinate frame.

Key words. semi-classical, quantum dynamics, splitting, hyperbolic cross, Hagedorn functions

AMS subject classifications. 65M70, 65Z05, 81-08

1. Introduction.

We consider the time-dependent Schrödinger equation in semiclassical scaling,

$$i\varepsilon \frac{\partial \psi}{\partial t} = H\psi, \tag{1.1}$$

where $\psi = \psi(x,t)$ is the wave function depending on the spatial variables $x = (x_1, \dots, x_N) \in \mathbb{R}^N$ and the time $t \in \mathbb{R}$. Here, ε is a small positive number representing the scaled Planck constant and i is the imaginary unit. The Hamiltonian operator H, which depends on ε, is written

$$H = T + V$$

with the kinetic and potential energy operators

$$T = -\sum_{j=1}^{N} \frac{\varepsilon^2}{2m_j} \frac{\partial^2}{\partial x_j^2} \quad \text{and} \quad V = V(x),$$

where $m_j > 0$ is a particle mass and where the real-valued potential V acts as a multiplication operator on ψ.

For example, in molecular quantum dynamics, (1.1) is a Schrödinger equation for the nuclei on an electronic energy surface in the time-dependent Born–Oppenheimer approximation (see, e.g., [20, 21, 14]). In this situation, ε^2 is the mass ratio between electrons and nuclei, of magnitude 10^{-4}.

Numerical approaches to solving (1.1) face two principal difficulties:

*INRIA & Ecole Normale Supérieure de Cachan Bretagne, Avenue Robert Schumann, F-35170 Bruz, France (Erwan.Faou@inria.fr)

†Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, CH-8092 Zürich, Switzerland, (vasile.gradinaru@sam.math.ethz.ch)

‡ Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D-72076 Tübingen, Germany (lubich@na.uni-tuebingen.de)

- *Highly oscillatory solutions:* Typical solutions are wavepackets of width $\sim \sqrt{\varepsilon}$, oscillatory with wavelength $\sim \varepsilon$, with the envelope moving at velocity ~ 1.
- *High dimension:* For n particles, the spatial dimension in (1.1) is $N = 3n$.

Because of the highly oscillatory solution behaviour, grid-based methods need very fine resolution for small ε and hence become computationally expensive or infeasible; cf. [15]. This also precludes the approximation of the wave function on sparse grids in higher dimensions, because the necessary smoothness requirements for this technique are not met for small ε [4].

On the other hand, it is known that moving complex Gaussians approximate solutions to (1.1) with an error of $O(\sqrt{\varepsilon}t)$ [6]. The model reduction from full quantum dynamics to Gaussian wavepacket dynamics [11, 12] allows for computationally efficient algorithms [3], but is often not accurate enough. Higher asymptotic accuracy in ε can be analytically proved for approximations of the wave function by complex Gaussians times polynomials [7, 8]. For the proof, Hagedorn [8] constructs particular, parameter-dependent L^2-orthonormal basis functions. In one space dimension, they are just scaled and shifted Hermite functions, but in higher space dimensions they are both more general and suitable than tensor products of Hermite functions. *The main contribution of the present paper is to turn the Hagedorn functions into a computational tool for the numerical solution of* (1.1).

In Section 2 we briefly review Hagedorn's parametrization of Gaussian wave packets and his parameter-dependent orthonormal basis functions (see also [14, Chap. V] for a self-contained concise review of Hagedorn's [6, 8] approach).

Section 3 describes our time-stepping algorithm working with the Hagedorn functions. It is based on a splitting into kinetic and potential energy in (1.1) and on a further splitting into the local quadratic approximation at the current classical position and the non-quadratic remainder. The latter is treated by a Galerkin approach with the Hagedorn basis for the current parameters, while the kinetic and quadratic part yield simple equations for the time evolution of the parameters. This approach yields an explicit time-stepping algorithm that is robust for $\varepsilon \to 0$ and enjoys a number of remarkable properties.

Section 4 deals with the computation of the Galerkin matrix for the non-quadratic remainder or of its action on coefficient vectors, which is all that is needed in a short Lanczos iteration for computing the time-dependent Galerkin approximation. With the use of a hyperbolic-cross reduction of the multi-dimensional basis set, the computational work is reduced from K^N to $O(K(\log K)^{N-1})$, where K is the maximum number of basis functions and quadrature points along a single coordinate direction. We present a fast algorithm that permits us to approximately compute the product of the Galerkin matrix with a coefficient vector in a number of operations and memory that are linear in the number of basis functions, without ever building up the matrix.

Section 5 presents numerical experiments. We give detailed comparisons with the full and sparse Fourier method in dimension 2 and also present some results of computations up to dimension 12.

In the present paper we concentrate on the conceptual and algorithmic aspects of the approach. Error analyses of (some of) the various approximations involved will be given elsewhere.

2. Building blocks.

2.1. Hagedorn wavepackets. We are looking for approximations to the Schrödinger equation that are products of complex Gaussians with polynomials. Representing the polynomials in a basis of scaled Hermite polynomials is appropriate in 1 space dimension [2], but in the multi-dimensional case, simply taking tensor products of Hermite polynomials (be it with a moving frame of coordinates) turns out to lead to a number of both theoretical and computational difficulties. These are overcome in an alternative extension to higher dimensions due to Hagedorn [8]. While the beautiful theoretical properties of this approach are evident from [8], it appears that so far they have not been put to use in computational algorithms.

In Hagedorn's approach [6], a Gaussian wavepacket is parametrized as[1]

$$\begin{aligned}&\varphi_0^\varepsilon[q,p,Q,P](x)\\ &= (\pi\varepsilon)^{-N/4}(\det Q)^{-1/2}\exp\Big(\frac{i}{2\varepsilon}(x-q)^T PQ^{-1}(x-q)+\frac{i}{\varepsilon}p^T(x-q)\Big)\,, \qquad (2.1)\end{aligned}$$

where $q \in \mathbb{R}^N$ and $p \in \mathbb{R}^N$ represent the position and momentum, respectively, and Q and P are complex $N \times N$ matrices satisfying the relations

$$Q^T P - P^T Q = 0 \qquad (2.2)$$

$$Q^* P - P^* Q = 2iI\,. \qquad (2.3)$$

Here Q^T denotes the transpose of Q, and Q^* is the transpose and conjugate complex matrix. As is explained in [8], these two equations imply that both Q and P are invertible, and PQ^{-1} is complex symmetric with positive definite imaginary part:

$$\mathrm{Im}\; PQ^{-1} = (QQ^*)^{-1}\,. \qquad (2.4)$$

Conversely, every complex symmetric matrix with positive definite imaginary part can be written as PQ^{-1} with Q and P satisfying (2.2),(2.3). We further note that (2.2),(2.3) are equivalent to stating that the matrix

$$Y = \begin{pmatrix}\mathrm{Re}\,Q & \mathrm{Im}\,Q\\ \mathrm{Re}\,P & \mathrm{Im}\,P\end{pmatrix} \quad \text{is symplectic:} \quad Y^T J Y = J \quad \text{with} \quad J = \begin{pmatrix}0 & -I\\ I & 0\end{pmatrix}.$$

Hagedorn constructs a complete L^2-orthonormal set of functions

$$\varphi_k(x) = \varphi_k^\varepsilon[q,p,Q,P](x)\,,$$

for multi-indices $k = (k_1, \dots, k_N)$ with non-negative integers k_j. This is done recursively as follows. Let x denote the position operator (acting on functions of x by multiplication with x), and $y = -i\varepsilon\nabla_x$ the momentum operator, and introduce the *raising operator* $\mathcal{R}$ and *lowering operator* $\mathcal{L}$ as

$$\begin{aligned}\mathcal{R} = (\mathcal{R}_j) &= -\frac{i}{\sqrt{2\varepsilon}}\Big(P^*(x-q) + Q^*(y-p)\Big)\,.\\ \mathcal{L} = (\mathcal{L}_j) &= \frac{i}{\sqrt{2\varepsilon}}\Big(P^T(x-q) + Q^T(y-p)\Big)\,.\end{aligned}$$

With $\langle j\rangle = e_j = (0\dots1\dots0)$ denoting the jth unit vector, set

$$\varphi_{k+\langle j\rangle} = \frac{1}{\sqrt{k_j+1}}\mathcal{R}_j\varphi_k\,. \qquad (2.5)$$

[1] In the notation used here, Q and P correspond to A and iB of [6, 7, 8], respectively. This notation is motivated by the equations of motion of Q and P, which then become the linearized classical equations for position and momentum, respectively.

It then turns out that these functions are orthonormal, as the eigenfunctions of the hermitian operators $\mathcal{L}_j\mathcal{R}_j = \mathcal{R}_j\mathcal{L}_j + I$. Moreover, we have

$$\varphi_{k-\langle j\rangle} = \frac{1}{\sqrt{k_j}}\,\mathcal{L}_j\varphi_k\,,$$

(the right-hand side is zero if $k_j = 0$), and the functions φ_k are polynomials of degree $|k| = k_1 + \cdots + k_N$ multiplied with the Gaussian φ_0. Since the above relations imply [8, (3.28)]

$$x - q = \sqrt{\frac{\varepsilon}{2}}(Q\mathcal{R} + \overline{Q}\mathcal{L})\,,$$

we obtain the recurrence relation

$$Q\big(\sqrt{k_j+1}\,\varphi_{k+\langle j\rangle}(x)\big)_{j=1}^N = \sqrt{\frac{2}{\varepsilon}}\,(x-q)\varphi_k(x) - \overline{Q}\big(\sqrt{k_j}\,\varphi_{k-\langle j\rangle}(x)\big)_{j=1}^N\,, \tag{2.6}$$

which permits us to compute the functions φ_k at any given value x.

2.2. Approximation by semi-classical wave packets. We approximate solutions to the Schrödinger equation (1.1) in the form of a finite linear combination of Hagedorn functions, with a common highly oscillatory phase factor,

$$\psi(x,t) \approx u(x,t) = e^{iS(t)/\varepsilon} \sum_{k\in\mathcal{K}} c_k(t)\,\varphi_k^\varepsilon[q(t),p(t),Q(t),P(t)](x)\,. \tag{2.7}$$

This ansatz is motivated by the remarkable fact from [8] that in the case of a *quadratic* (possibly time-dependent) potential V, the Hagedorn functions $\varphi_k^\varepsilon[q(t),p(t),Q(t),P(t)]$ multiplied with a phase factor $e^{iS(t)/\varepsilon}$ are exact solutions to the Schrödinger equation, if the position and momentum parameters follow the classical equations of motion (with the mass matrix $M = \mathrm{diag}(m_j)$)

$$\dot{q} = M^{-1}p, \qquad \dot{p} = -\nabla V(q)$$

and the linearized equations of motion

$$\dot{Q} = M^{-1}P, \qquad \dot{P} = -\nabla^2 V(q)Q,$$

and if $S(t) = \int_0^t \big(\frac{1}{2}p(s)^T M^{-1}p(s) - V(q(s))\big)\,ds$ is the classical action; see [8] (and [14, Chap. V] in the present notation). Hence, in the case of a *quadratic* potential, functions (2.7) with *constant* coefficients c_k are solutions to the Schrödinger equation (1.1). On the other hand, for a non-quadratic potential, the wave function can be expanded in the orthogonal basis of Hagedorn functions with time-dependent coefficients, with parameters determined by the equations of motion corresponding to a local quadratic approximation of the potential. For the numerical approximation, we make the ansatz (2.7) of a truncated expansion, which for this problem-adapted set of basis functions can be expected to yield good approximations even with a relatively small number of basis functions and with slowly varying coefficients.

The finite multi-index set $\mathcal{K}$ in (2.7) must be such that for every $k \in \mathcal{K}$, also $k - \langle j\rangle \in \mathcal{K}$ if $k_j > 0$. Some possibilities are:

- the *full cube* $k_j \le K$ $(j = 1,\dots,N)$;
- the *hyperbolic cross* $(1+k_1)\cdot\ldots\cdot(1+k_N) \le K$;
- the *axes* where $k_j > 0$ only for a single component j in each k.

In higher dimensions, the full cube is not computationally tractable. In our numerical experiments we use the hyperbolic cross as the multi-index set $\mathcal{K}$, since it yields a substantial reduction in higher dimensions, from K^N to at most $K(\log K)^{N-1}$ basis functions, and can nevertheless be expected to allow for a satisfactory approximation theory; cf. [14, Sect. III.1.2] for the related situation of the hyperbolically reduced tensor-product basis of Hermite functions. The choice of the axes can be viewed as a Hartree-type approximation in a moving frame. It gives an even more drastic reduction of basis functions but it cannot be expected to yield convergent approximations as the number of points on the axes is increased.

In the following sections we will give a fully discrete, explicit, and time-reversible time-stepping algorithm to propagate the Gaussian parameters $q(t), p(t), Q(t), P(t)$, the phase $S(t)$, and the coefficients $c_k(t)$ in the ansatz (2.7).

2.3. Splitting into bits and pieces. Our algorithm is based on the splitting between the kinetic and potential operators T and V. We consider the free linear Schrödinger equation

$$i\varepsilon\frac{\partial\psi}{\partial t} = -\sum_{j=1}^{N}\frac{\varepsilon^2}{2m_j}\frac{\partial^2\psi}{\partial x_j^2} \tag{2.8}$$

and the potential equation

$$i\varepsilon\frac{\partial\psi}{\partial t} = V(x)\psi. \tag{2.9}$$

The potential will be further decomposed into its quadratic part at the current position q and the non-quadratic remainder.

We now describe the three main ingredients in the time-stepping algorithm. Starting with a Hagedorn wavepacket (2.7) as initial data for the Schrödinger equation, we make use of the following:

- We can solve exactly the free linear Schrödinger equation (2.8), with the wavefunction remaining in the Hagedorn wavepacket form (2.7) with unaltered coefficients c_k.
- For a quadratic potential, we can solve exactly the potential equation (2.9) with the wavefunction remaining in the Hagedorn wavepacket form (2.7) with the same coefficients c_k.
- For an arbitrary potential, we can compute the Galerkin approximation of the potential equation (2.9) on the linear space spanned by the functions φ_k with fixed parameters q, p, Q, P, letting the coefficients c_k in the formulation (2.7) vary.

2.4. Kinetic part and quadratic potential. The following two propositions are direct consequences of [8, Theorem 3.4].

Proposition 2.1. *A time-dependent Hagedorn wavepacket (2.7) solves the free Schrödinger equation (2.8) if*

$$\begin{aligned} q(t) &= q(0) + tM^{-1}p(0) \\ Q(t) &= Q(0) + tM^{-1}P(0) \\ S(t) &= S(0) + \tfrac{1}{2}\,t\,p(0)^T M^{-1} p(0) \end{aligned}$$

and $p(t) = p(0)$, $P(t) = P(0)$, $c_k(t) = c_k(0)$. (Here, M =diag(m_j) is the mass matrix.)

PROPOSITION 2.2. *Let $U(x)$ be a quadratic potential. A time-dependent Hagedorn wavepacket (2.7) solves the potential equation (2.9) with $V = U$ if*

$$\begin{aligned} p(t) &= p(0) - t\,\nabla U(q(0)) \\ P(t) &= P(0) - t\,\nabla^2 U(q(0))Q(0) \\ S(t) &= S(0) - t\,U(q(0)) \end{aligned}$$

and $q(t) = q(0)$, $Q(t) = Q(0)$, $c_k(t) = c_k(0)$. (Here, ∇U denotes the gradient and $\nabla^2 U$ the Hessian matrix of U.)

2.5. Galerkin approximation for non-quadratic potentials. Let $W(x)$ be a given (non-quadratic) potential. We consider the potential equation (2.9) with $V = W$. We let the Gauss parameters q, p, Q, P be fixed and consider the linear space

$$\mathcal{M}[q,p,Q,P] = \{\, v \in L^2(\mathbb{R}^N) \;:\; v(x) = \sum_{k\in\mathcal{K}} c_k\,\varphi_k^\varepsilon[q,p,Q,P](x), \quad c_k \in \mathbb{C}\,\}$$

where $\varphi_k = \varphi_k^\varepsilon[q,p,Q,P]$ are the Hagedorn functions (2.5) associated with the fixed Gaussian parameters q, p, Q, P. The variational approximation on $\mathcal{M}[q,p,Q,P]$ can be written:

At every time t, determine $\partial_t u \in \mathcal{M}[q,p,Q,P]$ such that

$$\forall\, k \in \mathcal{K}, \quad \langle \varphi_k\,,\, \varepsilon i \partial_t u - W u \rangle = 0. \tag{2.10}$$

The following is then straightforward:

PROPOSITION 2.3. *The Galerkin approximation* (2.10) *is equivalent to the linear system of ordinary differential equations*

$$i\varepsilon \frac{dc_k}{dt} = \sum_{\ell\in\mathcal{K}} f_{k\ell}\,c_\ell, \qquad k \in \mathcal{K},$$

where

$$f_{k\ell} = \langle \varphi_k | W | \varphi_\ell \rangle = \int_{\mathbb{R}^N} \overline{\varphi_k(x)}\, W(x) \varphi_\ell(x)\, dx\,. \tag{2.11}$$

If $c(t)$ denotes the vector with components $c_k(t)$, $k \in \mathcal{K}$, the solution of this problem is thus given by the action of the exponential of the Hermitian matrix $F = (f_{k\ell})$ on the coefficient vector at the initial time:

$$c(t) = \exp\Bigl(-\frac{it}{\varepsilon}\,F\Bigr)c(0).$$

We note that $F = O(\varepsilon^{3/2})$ if the quadratic Taylor polynomial of W at q vanishes. The computation of the matrix exponential times a vector can then be done efficiently using just a few Lanczos iterations with F [13]. The efficient computation of the multi–dimensional integrals in (2.11) is discussed in Section 4.

3. The time-stepping algorithm.

3.1. Abstract formulation. For given parameters $\Gamma^0 = (q^0, p^0, Q^0, P^0, S^0)$ and coefficients $c^0 = (c_k^0)_{k\in\mathcal{K}}$, we denote

- by $\mathcal{T}_t(\Gamma^0, c^0)$ the solution to the free Schrödinger equation given by Proposition 2.1,
- by $\mathcal{U}_t(\Gamma^0, c^0)$ the solution of the quadratic-potential equation given by Proposition 2.2,
- and by $\mathcal{W}_t(\Gamma^0, c^0)$ the propagator given by Proposition 2.3.

A noteworthy fact is that with both propagators $\mathcal{U}_t$ and $\mathcal{W}_t$, the parameters q and Q remain constant. Moreover, the propagators $\mathcal{U}_t$ and $\mathcal{W}_t$ commute. This can be straightforwardly seen from the expressions in Propositions 2.2 and 2.3.

The algorithm is based first on the splitting between the kinetic and potential operators, and secondly on a splitting of the potential into its quadratic part at the current position and the remainder.

For a given stepsize Δt, the time-stepping algorithm is described briefly as follows:

1. *Half-step of kinetic part.* We define the parameters $(\Gamma^{1/2,-}, c^0)$ by applying the propagator $\mathcal{T}_{\Delta t/2}$ starting from (Γ^0, c^0). This yields updates $q^{1/2}$, $Q^{1/2}$, and $S^{1/2,-}$.
2. *Full step of potential part.* We split the potential $V(x)$ into its quadratic Taylor expansion around $q^{1/2}$ and the corresponding remainder term: We define the potentials
$$U^{1/2}(x) = V(q^{1/2}) + \nabla V(q^{1/2})\,(x - q^{1/2}) + \frac{1}{2}(x - q^{1/2})^T \nabla^2 V(q^{1/2})(x - q^{1/2})$$
as the local quadratic approximation to $V(x)$, and the remainder
$$W^{1/2}(x) = V(x) - U^{1/2}(x).$$
   - We determine the parameters $(\Gamma^{1/2,+}, c^0)$ by applying the propagator $\mathcal{U}_{\Delta t}$ associated with the quadratic potential $U^{1/2}$ starting from $(\Gamma^{1/2,-}, c^0)$. This yields updates p^1, P^1 and $S^{1/2,+}$.
   - We determine the coefficients c^1 using the propagator $\mathcal{W}_{\Delta t}$ associated with the non-quadratic remainder $W^{1/2}$ starting from c^0.
3. *Half-step of kinetic part.* We define the parameters (Γ^1, c^1) by applying the propagator $\mathcal{T}_{\Delta t/2}$ starting from $(\Gamma^{1/2,+}, c^1)$. This yields updates q^1, Q^1, and S^1.

3.2. Properties. The algorithm is of second order accuracy in the parameters q, p, Q, P, S and c_k and enjoys a number of attractive conservation and limit properties:

- The algorithm preserves the symplecticity relations (2.2) and (2.3) between the matrices Q and P, since it is a composition of exact flows with no or a quadratic potential, and Q and P are not modified in the step with the non-quadratic remainder.
- The algorithm preserves the L^2 norm of the wavepacket, since the Hagedorn functions φ_k are orthonormal and the propagation of the coefficients (c_k) is unitary.
- The algorithm is time-reversible, which is known to be a favourable property of numerical integrators for conservative problems (see, e.g., [10, Chap. V]). The time reversibility is due to the fact that $\mathcal{U}_t$ and $\mathcal{W}_t$ commute, and that in the potential stage, the positions q and the width matrix $\operatorname{Im} PQ^{-1} = (QQ^*)^{-1}$ remain unchanged.

- For the position and momentum parameters q and p, the algorithm coincides with the Störmer-Verlet algorithm [9] applied to the corresponding classical equations of motion: in the kinetic part, we have the momentum p^0 unchanged and $q^{1/2} = q^0 + \frac{\Delta t}{2} p^0$, whereas in the quadratic potential part, $q^{1/2}$ is constant while $p^1 = p^0 - \Delta t \nabla V(q^{1/2})$.
- In the limit of taking the full basis set φ_k with all $k \in \mathbb{N}^N$, the variational approximation used in the remainder propagator becomes exact. Since $\mathcal{U}_t$ and $\mathcal{W}_t$ commute, the second step in the previous algorithm asymptotically tends to the solution of the potential equation (2.9) in L^2. Hence, in this limit the algorithm converges towards the Strang splitting (or symmetric Lie-Trotter splitting) $\exp(-\frac{i}{\varepsilon}\Delta t H) \approx \exp(-\frac{i}{\varepsilon}\frac{\Delta t}{2}T)\exp(-\frac{i}{\varepsilon}\Delta t V)\exp(-\frac{i}{\varepsilon}\frac{\Delta t}{2}T)$ of the Schrödinger equation.
- The algorithm is robust in the classical limit $\varepsilon \to 0$: The propagator of the non-quadratic remainder, $\mathcal{W}_{\Delta t}$, is $O(\varepsilon^{1/2}\Delta t)$ close to the identity operator, since $W^{1/2}$ is at least cubic in $(x - q^{1/2})$. Hence the approximation in the potential part becomes exact for $\varepsilon \to 0$, while the kinetic part is solved exactly for all ε.

We note that these properties (with the exception of the fifth one) remain valid with an approximate computation of the integrals in (2.11).

3.3. The practical time-stepping algorithm. We now give a full algorithmic description. Assume that the stepsize Δt is given, and let the real N-vectors q^n, p^n, the complex $N \times N$ matrices Q^n, P^n, the real scalar S^n, and the complex coefficient vector $c^n = \left(c_k^n\right)_{k\in\mathcal{K}}$ be such that

$$u^n = e^{iS^n/\varepsilon} \sum_{k\in\mathcal{K}} c_k^n \, \varphi_k^\varepsilon[q^n, p^n, Q^n, P^n]$$

is an approximation to the solution of the Schrödinger equation (1.1) at time $t^n = n\Delta t$. To compute the approximation u^{n+1} at time t^{n+1} we proceed as follows:

1. Compute $q^{n+1/2}$, $Q^{n+1/2}$, and $S^{n+1/2,-}$ via

$$\begin{aligned} q^{n+1/2} &= q^n + \frac{\Delta t}{2} M^{-1} p^n \\ Q^{n+1/2} &= Q^n + \frac{\Delta t}{2} M^{-1} P^n \\ S^{n+1/2,-} &= S^n + \frac{\Delta t}{4} p^{nT} M^{-1} p^n \,. \end{aligned} \tag{3.1}$$

2. Compute p^{n+1}, P^{n+1}, and $S^{n+1/2,+}$ via

$$\begin{aligned} p^{n+1} &= p^n - \Delta t \, \nabla V(q^{n+1/2}) \\ P^{n+1} &= P^n - \Delta t \, \nabla^2 V(q^{n+1/2}) Q^{n+1/2} \\ S^{n+1/2,+} &= S^{n+1/2,-} - \Delta t \, V(q^{n+1/2}) \,. \end{aligned} \tag{3.2}$$

3. Update the coefficient vector $c^{n+1} = (c_k^{n+1})_{k\in\mathcal{K}}$ as

$$c^{n+1} = \exp(-\Delta t \frac{i}{\varepsilon} F^{n+1/2}) \, c^n. \tag{3.3}$$

Here, $F^{n+1/2} = (f_{k\ell})_{k,\ell\in\mathcal{K}}$ is the Hermitian matrix with entries

$$f_{k\ell} = \langle \varphi_k^{n+1/2} \,|\, W^{n+1/2} \,|\, \varphi_\ell^{n+1/2} \rangle, \tag{3.4}$$

where $\varphi_k^{n+1/2} = \varphi_k^\varepsilon[q^{n+1/2}, p^{n+1}, Q^{n+1/2}, P^{n+1}]$ are the Hagedorn basis functions and

$$W^{n+1/2}(x) = V(x) - U^{n+1/2}(x)$$

is the remainder in the local quadratic approximation to V, given at $q = q^{n+1/2}$ by $U^{n+1/2}(x) = V(q) + \nabla V(q)\,(x-q) + \frac{1}{2}(x-q)^T\nabla^2 V(q)\,(x-q)$. Note that $f_{k\ell}$ actually depends only on $q^{n+1/2}$ and $Q^{n+1/2}$, but not on p^{n+1} and P^{n+1}, since the imaginary parts in the arguments of the Gaussian cancel out in (3.4).

4. Compute q^{n+1}, Q^{n+1}, and S^{n+1} via

$$\begin{aligned} q^{n+1} &= q^{n+1/2} + \frac{\Delta t}{2} M^{-1} p^{n+1} \\ Q^{n+1} &= Q^{n+1/2} + \frac{\Delta t}{2} M^{-1} P^{n+1} \\ S^{n+1} &= S^{n+1/2,+} + \frac{\Delta t}{4} p^{n+1,T} M^{-1} p^{n+1}. \end{aligned} \tag{3.5}$$

4. Computing the contribution of the non-quadratic remainder. In the above algorithm, Step 3 treating the non-quadratic remainder of the potential is the computationally most expensive part of the algorithm, since it requires the computation of the multi-dimensional integrals (3.4) and of the action of the matrix exponential in (3.3). The latter can be done efficiently by Lanczos iterations as studied in [13] and first proposed in [18]. Since $\|F\| = O(\varepsilon^{3/2})$, only few Lanczos iterations are needed (see [14, page 96]).

Here we describe two approaches to compute the Galerkin matrix for the non-quadratic remainder, $F = (\langle \varphi_k \,|\, W \,|\, \varphi_\ell \rangle)_{k,\ell\in\mathcal{K}}$, of (3.4) or its action on a coefficient vector $c = (c_k)_{k\in\mathcal{K}}$, as is needed in the Lanczos iteration. The second approach appears particularly promising for higher-dimensional problems.

4.1. Computing the matrix elements by Gauss-Hermite quadrature. Formulas (2.1) and (2.6) show that with the change of variables $x = q + \sqrt{\varepsilon}|Q|y$ with $|Q| = (QQ^*)^{1/2}$ we can write

$$\overline{\varphi_k(x)}\,\varphi_\ell(x) = |\det Q|\,\varepsilon^{N/2}\,\overline{\phi_k(y)}\,\phi_\ell(y)$$

with the ε-independent functions ϕ_k given recursively by

$$\phi_0(y) = \pi^{-N/4} e^{-|y|^2/2} \tag{4.1}$$

$$Q\big(\sqrt{k_j+1}\,\phi_{k+\langle j\rangle}(y)\big)_{j=1}^N = \sqrt{2}\,|Q|y\,\phi_k(y) - \overline{Q}\big(\sqrt{k_j}\,\phi_{k-\langle j\rangle}(y)\big)_{j=1}^N. \tag{4.2}$$

We note that $\phi_k(y) = p_k(y)e^{-|y|^2/2}$ with some polynomial $p_k(y)$ of degree $k_1 + \cdots + k_N$. With these functions we thus have

$$\int_{\mathbb{R}^N} \overline{\varphi_k(x)}\,W(x)\,\varphi_\ell(x)\,dx = \int_{\mathbb{R}^N} \overline{\phi_k(y)}\,W(q + \sqrt{\varepsilon}|Q|y)\,\phi_\ell(y)\,dy\,.$$

We use multi-dimensional Gauss-Hermite quadrature for the latter integral; cf. [1, p. 174]. We denote by γ_m and ω_m the quadrature points and weights, respectively, for multi-indices $m = (m_1, \dots, m_N)$ with $m_j = 1, \dots, M$, and approximate

$$\int_{\mathbb{R}^N} f(y)\,dy \approx \sum_m \omega_m\, f(\gamma_m).$$

We have $\gamma_m = (\xi_{m_1}, \dots, \xi_{m_N})$, where the ξ_i are the zeros of the Mth degree Hermite polynomial H_M. The weights are

$$\omega_m = w_{m_1} \cdot \ldots \cdot w_{m_N} \qquad \text{with} \qquad w_i = \frac{\sqrt{\pi}}{M \cdot h_{M-1}(\xi_i)^2}\,.$$

Here h_{M-1} is the $(M-1)$-th Hermite function, which is computed in a stable way via the recurrence relation (4.1)–(4.2) with $N = 1$ and $Q = 1$. Then, the quadrature formula is exact for all $f(y) = e^{-|y|^2} p(y)$ with $p(y)$ a polynomial of degree up to $2M-1$ with respect to every coordinate. In this way we compute the matrix elements of (3.4):

$$\langle \varphi_k \,|\, W \,|\, \varphi_\ell \rangle \approx \langle \varphi_k \,|\, W \,|\, \varphi_\ell \rangle_{\text{GH}} = \sum_m \omega_m \, \overline{\phi_k(\gamma_m)}\, W(q + \sqrt{\varepsilon}|Q|\gamma_m)\, \phi_\ell(\gamma_m)\,.$$

In higher dimensions, instead of the full tensor grid with M^N quadrature points, we use sparse Gauss-Hermite quadrature with $O(M^2(\log M)^{N-1})$ evaluations of the potential; cf. [14, Sect. III.1.2]. Note that the integrands are not highly oscillatory, though the Hagedorn functions φ_k are.

4.2. Computing the action of the Galerkin matrix. An alternative approach to computing the product of the matrix $F = (\langle \varphi_k \,|\, W \,|\, \varphi_\ell \rangle)$ with a vector c, as is required in the Krylov subspace approximation to $\exp(-\Delta t \frac{i}{\varepsilon} F)c$ in (3.3), is now described. This does not require building up the matrix F and is applicable if the potential W is given as (or approximated by)

$$W(x_1, \dots, x_N) = \sum_{r=1}^{R} \alpha_r \, p_r^1(x_1) \dots p_r^N(x_N)$$

with univariate polynomials p_r^j.

We consider the matrix of the jth coordinate function (for $j = 1, \dots, N$),

$$X_j = (x_{k\ell}^j) \qquad \text{with} \qquad x_{k\ell}^j = \langle \varphi_k \,|\, x_j \,|\, \varphi_\ell \rangle \quad \text{for} \quad k, \ell \in \mathcal{K}.$$

Using the recurrence relation (2.6) and the orthogonality of the Hagedorn functions, it turns out that the action of X_j on a vector $c = (c_k)_{k \in \mathcal{K}}$ is readily computed:

$$\big((X_j c)_k\big)_{j=1}^N = c_k \, q + \sqrt{\frac{\epsilon}{2}}\, Q\big(\sqrt{k_j}\, c_{k-<j>}\big)_{j=1}^N + \sqrt{\frac{\epsilon}{2}}\, \overline{Q}\big(\sqrt{k_j+1}\, c_{k+<j>}\big)_{j=1}^N \tag{4.3}$$

for all $k \in \mathcal{K}$. This remarkably efficient formula enables us to compute $X_j c$ in $O(N \cdot \#\mathcal{K})$ operations with $O(\#\mathcal{K})$ memory. We now approximate

$$F = \big(\langle \varphi_k \,|\, W \,|\, \varphi_\ell \rangle\big)_{k,\ell \in \mathcal{K}} \approx \widetilde{F} := \sum_{r=1}^{R} \alpha_r \, p_r^1(X_1) \dots p_r^N(X_N)\,. \tag{4.4}$$

With (4.3), we can

compute $\widetilde{F}c$ in $O(dN \cdot \#\mathcal{K})$ operations with $O(\#\mathcal{K})$ memory,

where $d = \sum_{r=1}^R \sum_{j=1}^N \deg p_r^j$ is the sum of all polynomial degrees.

5. Numerical experiments. We consider the following settings for the simulations:

(a) The torsional potential is

$$V(x) = \sum_{j=1}^{N} (1 - \cos(x_j)) .$$

As initial value we take the normed Gaussian wavepacket in Hagedorn's parametrization (2.1) with the identity matrix $Q = I_N$ and $P = iI_N$, localized around $q = (1, 0, \dots, 0)^T$ and with $p = 0$.

(b) The modified Henon-Heiles potential as in [16] and [19] is defined as:

$$V(x) = \frac{1}{2}\sum_{j=1}^{N} \sigma_j x_j^2 + \sum_{j=1}^{N-1} \left(\sigma_*(x_j x_{j+1}^2 - \frac{1}{3}x_j^3) + \frac{1}{16}\sigma_*^2(x_j^2 + x_{j+1}^2)^2 \right)$$

with harmonic part coefficients $\sigma_j = 1$ and the mixing coefficient $\sigma_* = 0.2$. In the case $N = 2$, we take the initial conditions from [16], i.e., a normed Gaussian wavepacket with $q = (1.8, 0)^T$, $p = (0, 1.2)^T$, and with the matrices $Q = \sqrt{2} \cdot \mathrm{diag}(\sqrt{0.56}, \sqrt{0.24})$, $P = iQ^{-1}$ in Hagedorn's parametrization (2.1).

5.1. Comparison with the Fourier method. The periodicity and the smoothness of the torsional potential make it ideal for comparison tests with the Fourier method. We take a uniform grid with $(2^r)^2$ points on $[-\pi, \pi]^2$. In the cases $\epsilon = 0.1, 0.01$ the choice $r = 11$ together with the Strang-splitting in time gives a good approximation to the solution of the time dependent Schrödinger equation, whereas $\epsilon = 10^{-3}$ needs a Fourier resolution of $r = 12$. We take the Fourier solution as reference for a comparison with the solution obtained by the Hagedorn wavepackets. We fix the time-step $\Delta t = 0.01$ and approximate by sparse Hagedorn wavepackets (2.7) with $\mathcal{K} = \{(k_1, k_2) : (1 + k_1)(1 + k_2) \leq K\}$ with $K = 8$, i.e., we use 20 basis functions. We compute the values of the solutions at the Fourier grid points. If we compare these values directly, we expect a phase-error of order $(\Delta t)^2/\epsilon$. If we compare the absolute values of the two solutions, we have only the error of order $(\Delta t)^2$, and this is the quantity we plot against time in Figure 5.1. On the right side of this figure we compare the absolute values of the Fourier solutions at levels $r = 10$ and $r = 12$ at $\epsilon = 0.1, 0.01, 0.001$. We note that contrary to the Fourier method, the method based on the Hagedorn wavepacket improves for small ϵ.

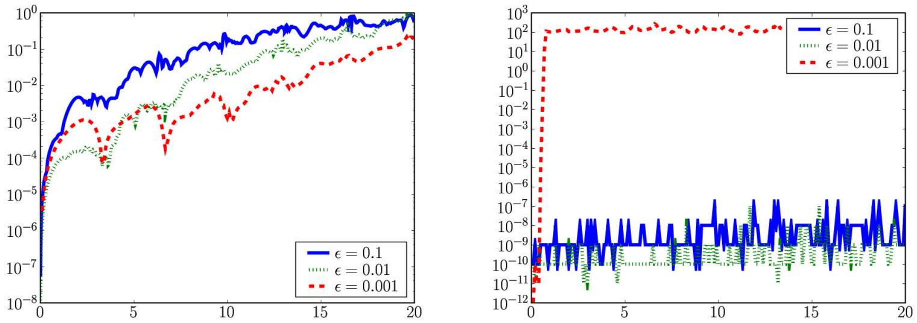

Fig. 5.1. *Time evolution of the maximum error in the absolute values of the wave function: Hagedorn with 20 basis functions (left) and Fourier with* $(2^r)^2$ *basis functions for* $r = 10$ *(right).*

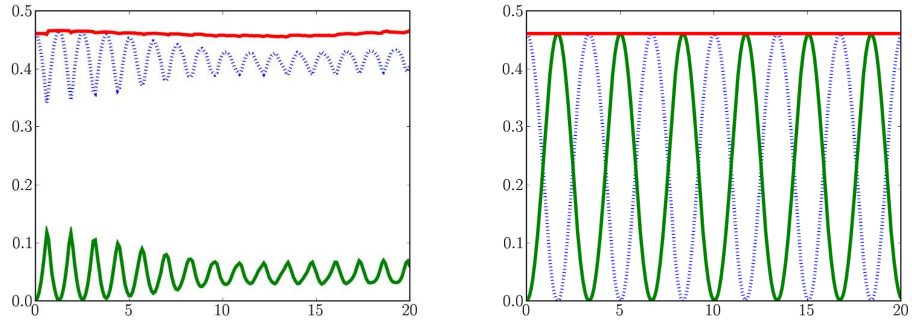

Fig. 5.2. *Fourier: total energy conservation for $\epsilon = 10^{-3}$ at resolutions $r = 10$ (left) and $r = 12$ (right); the kinetic and the potential energy (dotted line) oscillate.*

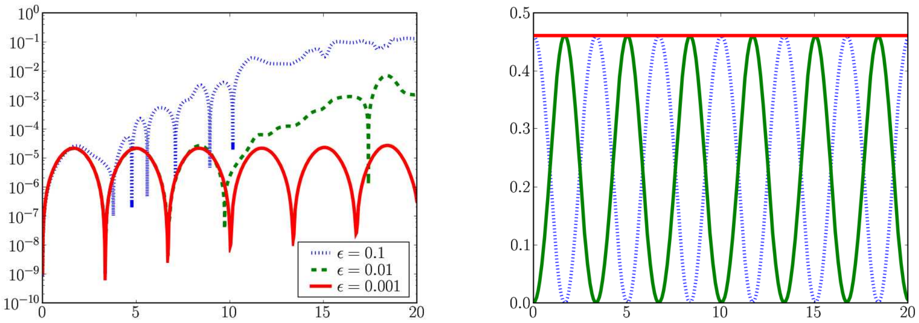

Fig. 5.3. *Hagedorn: energy deviation and evolution (right at $\epsilon = 10^{-3}$).*

We observe from Figure 5.2 that even though the total energy is well conserved at the level $r = 10$, the resolution of the full grid Fourier method must be increased to $r = 12$ in the case $\epsilon = 0.001$ in order to get a correct energy exchange. Moreover, at resolution $r = 11$, the picture of the energies looks very much like the right part of Figure 5.2, even if the maximum error is about 0.7. On the contrary, the Hagedorn wavepacket with 20 basis functions gives a better approximation of the energies as Figure 5.3 shows.

Raising the space dimension N makes the costs of the full-grid Fourier method prohibitive. An alternative is to consider the Fourier method on sparse grids [4]. However, as is explained in detail in [5], this alternative suffers from an aliasing problem that makes it incompatible with small ϵ and hence is not well suited for a semi-classical approximation. We illustrate this fact in Figure 5.4. Here, we take as reference the full grid Fourier method at level $r = 10$ using $(2^{10})^2 = 1048576$ points. We display the maximum error at the grid points that are common to the full grid and to sparse grids at levels 13, 14, and 15, having, respectively 77 825, 163 841, and 344 065 points. We see that for smaller ϵ, we need to increase the resolution of the grid. Hence, even if it lessens the curse of dimensionality, the sparse grid Fourier method is not well suited for small ϵ [4].

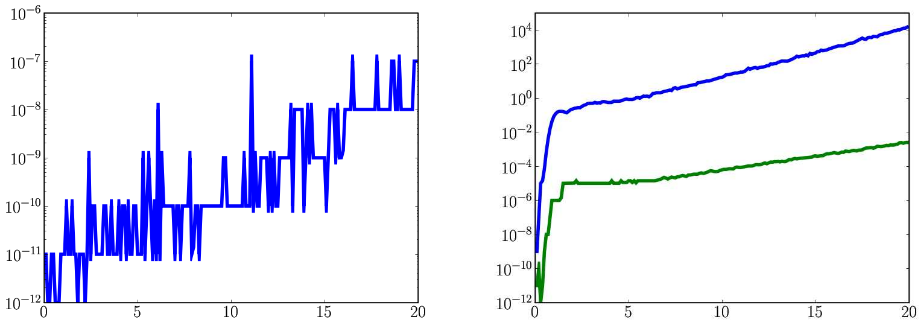

Fig. 5.4. *Maximum error in the sparse Fourier method: level* 13 *for* $\epsilon = 0.1$ *(left) and levels* 14 *and* 15 *for* $\epsilon = 0.01$ *(right).*

We focus on the modified Henon-Heiles potential for the rest of the numerical experiments. Similar results arise in the case of the torsional potential.

5.2. Convergence in time. A sparse Hagedorn wavepacket is now propagated with different time-steps. The solution computed with the smallest time-step $0.1 \cdot 2^{-11} = 9.765625 \cdot 10^{-5}$ serves as reference solution and is compared to the solutions obtained with other time-steps. Given enough observation points in the space domain, we look at the maximum error in the absolute values of the wave function at time $t = 1$ and $t = 5$. The plots for different ϵ are indistinguishable in Figure 5.5. The temporal convergence is of order 2, uniformly in ϵ.

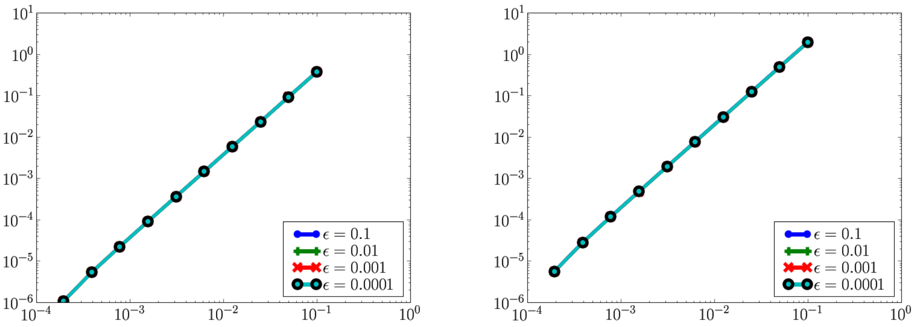

Fig. 5.5. *Convergence in time at* $t = 1$ *and* $t = 5$*: indistiguishable error curves for different* ϵ*.*

5.3. Variable number of basis functions. Sparse Hagedorn wavepackets with different numbers of basis functions are now propagated with the fixed time-step 10^{-2}. The solution computed with the largest set of basis functions (i.e., 645, hyperbolic cross with $K = 2^7$) serves as reference solution and is compared to the solutions obtained at other resolutions. Given enough obervation points in the space domain, we look at the maximum error of the absolute values at time $t = 1$ and $t = 5$. We notice that the convergence improves with smaller ϵ. However, we do not observe satisfactory convergence for long times, unless ϵ is very small.

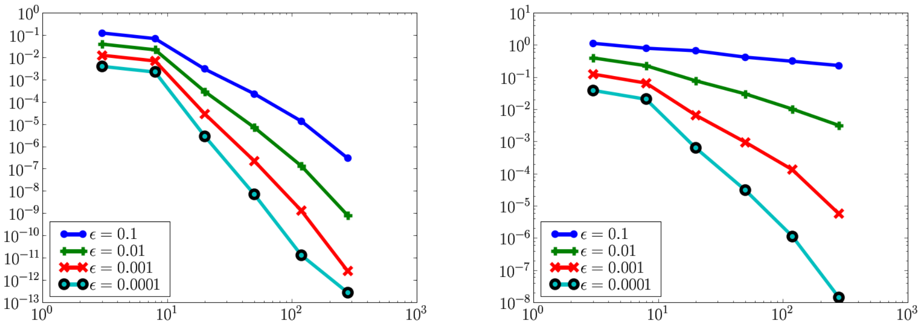

FIG. 5.6. *Maximum error versus number of basis funtions at $t = 1$ and $t = 5$.*

5.4. Computations in higher dimensions. Let us end with plots of the evolution of the energies in dimension $N = 6$ and of the computational time up to dimension $N = 12$. First, we extend our considered two dimensional Henon-Heiles model. We take initial values $q = (1.8, 0, \dots 0)^T$, $p = (0, 1.2, 0, \dots, 0)^T$, $Q = \sqrt{2} \cdot \mathrm{diag}(\sqrt{0.56}, \sqrt{0.24}, 1, \dots, 1)$, $P = iQ^{-1}$. In this case, we plot the total, the potential and the kinetic energy (dotted line) on the left of the Figure 5.7. The right side of Figure 5.7 has parameters similar to [17]: here we have $\sigma_* = 1/\sqrt{80}$ and the initial data given by $q = (2, \dots, 2)^T$, $p = 0$ and identity matrix Q. Both computations are done with $\epsilon = 10^{-2}$, a time-step $\Delta t = 10^{-2}$ and a sparse Hagedorn wavepacket with hyperbolically reduced multi-index set $\mathcal{K} = \{(k_1, \dots, k_N) : \prod_{j=1}^{N}(1 + k_j) \le K\}$ for $K = 8$, hence with 138 basis functions in $N = 6$ dimensions.

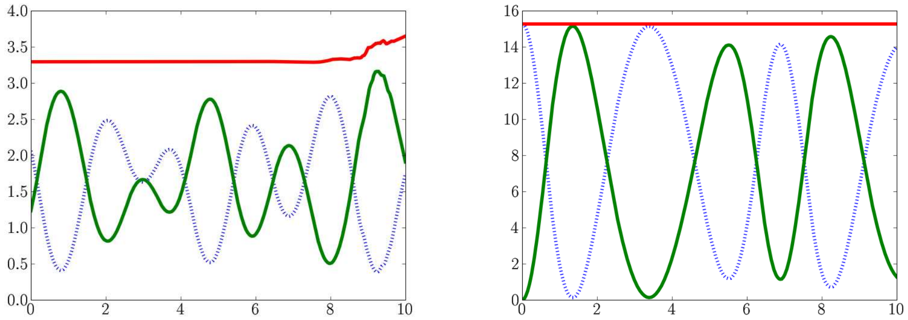

FIG. 5.7. *Energy evolution for the two test examples in dimension $N = 6$ at $\epsilon = 10^{-2}$.*

With the second choice of the model and parameters, we propagated a sparse Hagedorn wavepacket at fixed $K = 8$ for dimensions from $N = 2$ to $N = 12$. The results for the energy propagation in higher dimensions are very similar to the previous ones. In Figure 5.8, we show the user time spent by a dual-core AMD Opteron(tm) processor 2218 for the whole propagation (left) and the average user time for a time-step (right). The dashed line corresponds to the case of a full Gauss-Hermite quadrature as described in Section 4.1 while the full line corresponds to the case of the grid-free algorithm from Section 4.2. The results are produced by a pure Python code with a few calls of some array-manipulation routines of the numpy-library.

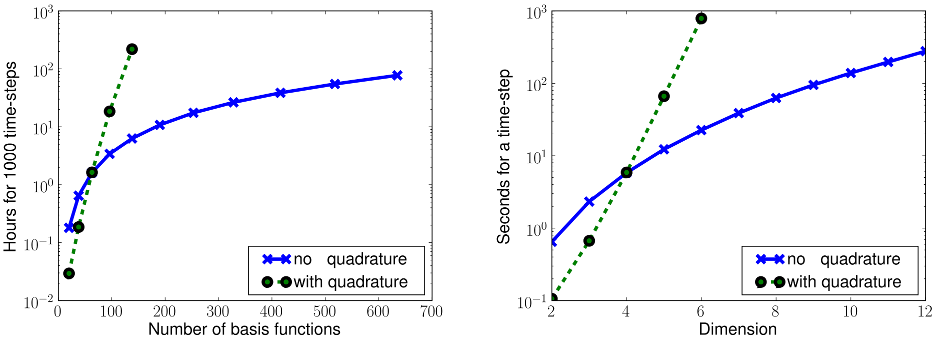

Fig. 5.8. *CPU time with a sparse Hagedorn wavepacket at fixed $K = 8$ in dimensions 2 to 12.*

Acknowledgement. This work was partially supported by DFG, SPP 1324.

REFERENCES

[1] P. Davis, P. Rabinowitz, *Methods of Numerical Integration.* Academic Press, 1975.

[2] E. Faou, V. Gradinaru, *Gauss-Hermite wavepacket dynamics: Convergence of the spectral and pseudo-spectral approximation.* IMA J. Numer. Anal., to appear.

[3] E. Faou, C. Lubich, *A Poisson integrator for Gaussian wavepacket dynamics.* Comput. Vis. Sci. 9 (2006), 45–55.

[4] V. Gradinaru, *Strang splitting for the time dependent Schrödinger equation on sparse grids.* SIAM J. Numer. Anal. 46 (2007), 103–123.

[5] V. Gradinaru, *Fourier transform on sparse grids: code design and application to the time dependent Schrödinger equation on sparse grids.* Computing 80 (2007), 1–22.

[6] G.A. Hagedorn, *Semi-classical quantum mechanics I: the $\hbar \to 0$ limit for coherent states.* Comm. Math. Phys. 71 (1980), 77–93.

[7] G.A. Hagedorn, *Semi-classical quantum mechanics. IV. Large order asymptotics and more general states in more than one dimension.* Ann. Inst. H. Poincaré Phys. Théor. 42 (1985), 363–374.

[8] G.A. Hagedorn, *Raising and lowering operators for semi-classical wave packets.* Ann. Physics 269 (1998), 77–104.

[9] E. Hairer, C. Lubich, G. Wanner, *Geometric numerical integration illustrated by the Störmer-Verlet method.* Acta Numerica 12 (2003), 399–450.

[10] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations.* Second edition, Springer, Berlin, 2006.

[11] E.J. Heller, *Time dependent approach to semi-classical dynamics.* J. Chem. Phys. 62 (1975), 1544–1555.

[12] E.J. Heller, *Time dependent variational approach to semi-classical dynamics.* J. Chem. Phys. 64 (1976), 63–73.

[13] M. Hochbruck, C. Lubich, *On Krylov subspace approximations to the matrix exponential operator.* SIAM J. Numer. Anal. 34 (1997), 1911–1925.

[14] C. Lubich, *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis.* Europ. Math. Soc., Zürich, 2008.

[15] P.A. Markowich, P. Pietra, C. Pohl, *Numerical approximation of quadratic observables of Schrödinger-type equations in the semi-classical limit.* Numer. Math. 81 (1991), 595–630.

[16] H.-D. Meyer, U. Manthe, L.-S. Cederbaum, *The Multiconfigurational Time-Dependent Hartree Approach.* Chem. Phys. Lett. 165 (1990), 73–78.

[17] M. Nest, H.-D. Meyer, *Benchmark calculations on high-dimensional Henon–Heiles potentials with the multi-configuration time dependent Hartree (MCTDH) method.* J. Chem. Phys. 117 (2002), 10499–10505.

[18] T.J. Park, J.C. Light, *Unitary quantum time evolution by iterative Lanczos reduction.* J. Chem. Phys. 85 (1986), 5870–5876.

[19] A. Raab, H.-D. Meyer, *A numerical study on the performance of the multiconfiguration time-dependent Hartree method for density operators.* J. Chem. Phys. 112 (2000), 10718–10729.

[20] D.J. Tannor, *Introduction to Quantum Mechanics: A Time-Dependent Perspective.* University Science Books, Sausalito, 2007.

[21] S. Teufel, *Adiabatic Perturbation Theory in Quantum Dynamics.* Lecture Notes in Mathematics 1821, Springer-Verlag, Berlin, 2003.