Calculating Floquet states of large quantum systems: A parallelization strategy and its cluster implementation

T.V. Laptyeva [a], E.A. Kozinov [b], I.B. Meyerov [b], M.V. Ivanchenko [c], S.V. Denisov [c,d,*], P. Hänggi [c,d]

[a] *Theory of Control and Systems Dynamics Department, Lobachevsky State University of Nizhny Novgorod, Russia*
[b] *Mathematical Software and Supercomputing Technologies Department, Lobachevsky State University of Nizhny Novgorod, Russia*
[c] *Department of Applied Mathematics, Lobachevsky State University of Nizhny Novgorod, Russia*
[d] *Institut für Physik, Universität Augsburg, Universitätsstraße 1, D-86135 Augsburg, Germany*

1. Introduction

Fast progress in manipulations with cold and ultra-cold atoms, quantum optics and nanoscale fabrication techniques has brought quantum physics in touch with technology [1–3]. It is then natural that computational quantum physics plays an ever increasing role in explaining and guiding current experiments and suggesting new ones [4]. From the computational point of view, the complete resolution of a coherent, i.e., an isolated from the environment, quantum system means the solution of the eigenvalue problem for the system Hamiltonian H. When the Hamiltonian is time-independent, this task can be executed by performing full diagonalization of the Hamiltonian matrix. When the system becomes too large the size of the matrix may not allow any longer for its full diagonalization. The task, however, could be restricted to finding lowest energy eigenstate(s) which can be accomplished by using the Lanczos algorithm [5] or more sophisticated tools, such as the Density-Matrix Renormalization Group (DMRG) methods [6]. In cases that the system is periodically modulated in time, its Hamiltonian becomes a time-periodic operator $H(t+T) = H(t + 2\pi/\omega) = H(t)$. The dynamics of the system is accordingly then governed by the set of so termed *Floquet states* [7,8]. These states are not eigenvectors of the Hamiltonian $H(t)$ but instead of the unitary Floquet operator

$$U_T = \mathcal{T} \exp\left[-\frac{i}{\hbar}\int_0^T H(t')dt'\right], \tag{1}$$

where $\mathcal{T}$ is Dyson's time-ordering operator. This operator propagates the system over the period T of modulations, while the corresponding time-periodic Floquet states form a time-periodic orthogonal basis spanning the system Hilbert space [9,10]. The structure of the unitary Floquet matrix, and thus the properties of the Floquet states, depend on the modulation protocols and parameters. This is a key feature of periodically driven quantum systems

* Corresponding author at: Institut für Physik, Universität Augsburg, Universitätsstraße 1, D-86135 Augsburg, Germany.
E-mail address: sergey.denisov@physik.uni-augsburg.de (S.V. Denisov).

which makes them so attractive to the theoreticians and experimentalists working in the field of quantum optics, optomechanics and solid state physics [9–13]. Strong modulations can sculpt a set of non-equilibrium eigenstates which may drastically differ from the states exhibited by the system in the unmodulated, stationary limit. Thus, modulations allow to grasp novel phenomena and effects which are out of reach within time-independent Hamiltonians; they can be used to create topological insulators in semiconductor wells [14], synthesize Majorana fermions in quantum wires [15], and engineer gauge fields for spinless neutral atoms [16].

The calculation of Floquet states of a large quantum system constitutes a challenge. The key step is a construction of the unitary Floquet matrix, Eq. (1) (its final diagonalization computationally similar to the diagonalization of stationary Hamiltonian matrices). The most straightforward way to obtain U_T is to numerically propagate the identity matrix over the time period T. However, the propagation with a time-dependent Hamiltonian operator presents an issue of its own. There are two ways to do so.

The first option is to use piecewise-constant modulation functions. This allows to reduce the computational task to the diagonalization of time-independent Hamiltonians, one for every time interval, and the expansion of eigenvectors of a preceding Hamiltonian in the basis of the consecutive one. Such modulations were used to investigate connections between integrability and thermalization [17–19], and to explore disorder-induced localization [20] in periodically driven many-body systems. With respect to the thermalization it was found that the temporal modulations heat the system to infinite temperature so that the system Floquet states are near uniformly smeared over the eigenbasis of the system in the absence of driving [17–19]. An important question that immediately arises is whether this is a universal phenomenon or it is related to the non-differentiability of the modulation function (whose property induces the presence of all multiple frequencies $k\omega$, $k = 1, 2, \ldots$, in the spectrum of the modulations function). Evidently, this question cannot be answered without going beyond the piecewise setup. In addition, in view of possible experimental realizations, smooth continuous modulations are also more preferable.

An alternative option is to expand the time-dependent Hamiltonian into a Fourier series and, and then truncating it, by keeping $2F + 1$ harmonics $k\omega$, $k = -F, \ldots, 0, \ldots, F$ only, to reduce the problem to the diagonalization of a time-*independent* super-Hamiltonian [8,21]. This is a reliable method to obtain Floquet spectrum of a system of a size up to a hundred of states. For larger systems, this strategy leads to a computational problem: the size of the super-Hamiltonian scales as $N \times (2F + 1)$, where N is the dimension of the system's Hilbert space. Computational diagonalization efforts increase as $[N \times (2F+1)]^3$, while the known diagonalization algorithms are poorly scalable. For a system of the size $N = 10^4$, already $F = 50$ harmonics is far too much; a full diagonalization of a $10^6 \times 10^6$ matrix becomes unfeasible. At the same time, this large number of harmonics is not enough to resolve faithfully the Floquet spectrum of the system.[1]

Therefore, in order to calculate the Floquet state of a system with $N \geqslant 10^3$ states, the propagation stage has to be included into an algorithm. A propagation method should guarantee a high accuracy with respect not only to the unitary time evolution, but as well with respect to the phases of complex vectors. That is because Floquet states appear as superpositions of basis vectors used to write system's Hamiltonian. Accumulated phase errors will destroy the interference and lead to an incorrect set of Floquet states. As we show in Section 7, quantum interference effects, together with some results from quantum chaos theory [22], can be used to benchmark the accuracy of an algorithm.

Because of the trade-off between the accuracy and system size, the time of sequential vector propagation grows superlinearly with N. Faithful calculations of Floquet spectra of non-integrable systems (whose Hilbert space cannot be decomposed into several non-interacting low-dimensional manifolds [23]), with tens of thousands of states, can only be performed with scalable algorithms.

This paper presents an algorithm to calculate the Floquet spectra of strongly-modulated quantum systems with $N \geqslant 10^4$ quantum states and its implementation on a parallel supercomputer. The propagation part of the algorithm is based on the combination of the Magnus expansion of time-dependent linear operators [24] and the Chebyshev expansion of operator exponents [25]. This combination has been proposed in [26], where its particular numerical realization, implementing a commutator-free Magnus scheme, was tested. We demonstrate here the accuracy and scalability of the algorithm by using two quantum models, with a synthesized random-matrix Hamiltonian and a many-body non-integrable bosonic dimer. The size of model system is limited by the diagonalization routine only, so the algorithm can be used to calculate Floquet states of systems of the size up to $N \sim 50\,000$ states.

The paper is organized as follows: Section 2 outlines the theoretical background and introduces the Magnus and Chebyshev expansions; Section 3 describes the algorithm; in Section 4 we introduce model systems, apply the cluster implementation to calculate their Floquet states in Section 5, and analyze the results in Section 7. Finally we summarize our findings and outline further perspectives in Section 8.

2. Theoretical background

Floquet states. We consider quantum systems whose dynamics is determined by the time-dependent Schrödinger equation

$$i\hbar\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle, \tag{2}$$

where the Hamiltonian $H(t)$ denotes a time-periodic Hermitian operator, $H(t + T) = H(t)$. We assume that the system evolves in a finite-dimensional Hilbert space spanned by N basis vectors. The time evolution of the system is fully determined by a unitary operator $U(t_0, t)$, being the solution of the equation

$$i\hbar\partial_t U(t_0, t) = H(t)U(t_0, t) \tag{3}$$

for the initial condition in the form of the identity matrix, $U(t_0, t_0) = \mathbb{1}$. This provides the *propagator* of the system, i.e. a unitary operator, which evolves any system state from a time t_0 to time $t_0 + t$, $U(t_0, t)|\psi(t_0)\rangle = |\psi(t_0 + t)\rangle$. A time $t_0 \in [0, T]$ specifies the state of the Hamiltonian operator at the initial time, when, for example, the driving is switched on. This starting time can be absorbed into the Hamiltonian as a parameter, $H(t, t_0) = H(t + t_0)$ (the propagator $U(t_0, t)$ can be obtained from $U(0, t)$ as $U(t_0, t) = U^\dagger(0, t_0)U(0, t + t_0)$), so for later convenience, we set $t_0 = 0$ in Eq. (3) and denote $U(0, t)$ by U_t. Eigenvectors $\{|\phi_\mu(0)\rangle\}$ of the unitary matrix U_T,

$$U_T|\phi_\mu(0)\rangle = e^{-i\theta_\mu}|\phi_\mu(0)\rangle, \quad \mu = 1, \ldots, N, \tag{4}$$

form a time-periodic full orthonormal basis in the system Hilbert space[2] [7–9,15,16,18,20]

$$|\phi_\mu(t + T)\rangle = |\phi_\mu(t)\rangle. \tag{5}$$

[1] The eigenvalue spectrum of the super-Hamiltonian can be resolved with the accuracy $2\pi/(2F + 1)$ at best. This is not enough taking into account that the actual mean spacing between the eigenvalues is π/N.

[2] When the notion of the Floquet states was introduced in quantum physics, the following convention has originally been employed: The set $\{\varphi_\mu(t) =$

These vectors are simply related to the stroboscopic snapshots, $|\psi(nT)\rangle$, when $|\psi(0)\rangle = |\phi_\mu(0)\rangle$, namely $|\psi(nT)\rangle = e^{-i\epsilon_\mu nT/\hbar}|\phi_\mu(t)\rangle$ with $\epsilon_\mu = \hbar\theta_\mu/T$, $n \in \mathbb{N}$. The exponents ϵ_μ have the dimension of energy and are termed *quasienergies*. Quasienergies can be determined up to multiples of $\hbar\omega$ so they are conventionally restricted to the interval $[-\hbar\omega/2, \hbar\omega/2]$.

Upon knowing the Floquet spectrum of a system, $\{\epsilon_\mu, |\phi_\mu(t)\rangle\}$, and the initial system state $|\psi(0)\rangle$, one can evaluate the state of the system at any later instant of time $t > 0$, i.e.

$$|\psi(t)\rangle = \sum_\mu c_\mu e^{-i\epsilon_\mu t/\hbar}|\phi_\mu(t)\rangle, \quad c_\mu = \langle\phi_\mu(0) \mid \psi(0)\rangle. \tag{6}$$

Magnus expansion. The idea of the Magnus expansion [27] is to construct a time independent Hamiltonian operator $\Omega(t_1, t_2)$, parameterized by the two times, t_1 and t_2, such that

$$U(t_1, t_2) = \exp\left[-\frac{i}{\hbar}\Omega(t_1, t_2)\right]. \tag{7}$$

The operator is given by an infinite series involving nested commutators [24], i.e.

$$\begin{aligned}\Omega(t_1, t_2) = &\int_{t_1}^{t_2} H(\tau_1)d\tau_1 \\ &+ \frac{1}{2}\int_{t_1}^{t_2} d\tau_1 \int_{t_1}^{\tau_1} [H(\tau_1), H(\tau_2)]d\tau_2 \\ &+ \frac{1}{6}\int_{t_1}^{t_2} d\tau_1 \int_{t_1}^{\tau_1} d\tau_2 \int_{t_1}^{\tau_2} \big([H(\tau_1), [H(\tau_2), H(\tau_3)]] \\ &+ [H(\tau_3), [H(\tau_2), H(\tau_1)]]\big)d\tau_3 + \cdots .\end{aligned} \tag{8}$$

An implementation of the expansion (8) assumes truncation of the infinite series, summation of the obtained finite series to obtain operator $\Omega(t_1, t_2)$, and use of the latter for the propagator $U(t_1, t_2)$ [27]. The Floquet operator U_T can then be approximated as a chain

$$\begin{aligned}U_T &= U(0, t_1)U(t_1, t_2)\ldots U(t_{M-1}, t_M) \\ &\approx e^{-i\Omega(0,t_1)/\hbar}e^{-i\Omega(t_1,t_2)/\hbar}\cdots e^{-i\Omega(t_{M-1},t_M)/\hbar},\end{aligned} \tag{9}$$

where $t_k = kh = kT/M$, $k = 0, \ldots, M$. Because all terms on the rhs of Eq. (8) are Hermitian, the truncated operator $\Omega(t_1, t_2)$ is Hermitian, and an approximation of any order preserves the unitary time evolution. The truncated operator in the form (8) is not very suitable for computations. It is more convenient to approximate $\Omega(t_1, t_2)$ with lower-order commutator series, calculated by using values of $H(t_{1/2})$ at the corresponding midpoints $t_{1/2} = (t_1 + t_2)/2$ (this is our choice, see Section 3 for more details), or with a commutator-free linear combination of $H(t_j)$, calculated at different times $t_j \in [t_1, t_2]$ [24,26].

Chebyshev expansion. The exponentiation of an operator is a computationally expensive operation [28]. In order to propagate vector $|\psi(t_1)\rangle$ to time t_2, the knowledge of the unitary operator $\exp(-i\Omega(t_1, t_2)/\hbar)$ is redundant: we need only the result of its action on the vector, $|\psi(t_2)\rangle = \exp(-i\Omega(t_1, t_2)/\hbar)|\psi(t_1)\rangle$. This can be calculated by implementing the Chebyshev polynomial expansion of the operator exponent, which is based on a recursive iteration scheme [25],

$$|\psi_{l+1}(t_2)\rangle = -2i\tilde{\Omega}(t_1, t_2)|\psi_l(t_2)\rangle + |\psi_{l-1}(t_2)\rangle \tag{10}$$

with the initial conditions $|\psi_0(t_2)\rangle = |\psi(t_1)\rangle$ and $|\psi_1(t_2)\rangle = -i\tilde{\Omega}(t_1, t_2)|\psi_0(t_2)\rangle$. Here $\tilde{\Omega}(t_1, t_2)$ is a shifted and rescaled operator,

$$\tilde{\Omega}(t_1, t_2) = \frac{\Omega(t_1, t_2) - \mathbb{1}\cdot(\Delta\Omega + \Omega_{\min})}{\Delta\Omega}, \tag{11}$$

which has all its eigenvalues restricted to the interval $[-1, 1]$ [25]. The spectral half-span $\Delta\Omega = (\Omega_{\max} - \Omega_{\min})/2$ should be estimated from the extreme eigenvalues $\Omega_{\min}$ and $\Omega_{\max}$ of $\Omega(t_1, t_2)$ operator beforehand.

Finally, the new vector can be obtained as

$$|\psi(t_2)\rangle = e^{-i\beta h/\hbar}\sum_{l=0}^{L} a_l|\psi_l(t_2)\rangle, \tag{12}$$

where $\beta = \Delta\Omega + \Omega_{\min}$ and $h = t_2 - t_1$. The expansion coefficients $a_l = 2J_l(r)$ and $a_0 = J_0(r)$, where $J_l(r)$ denote the Bessel functions of the first kind and $r = h\Delta\Omega/\hbar$. The parameter L sets the order of the Chebyshev approximation by truncating the series (12). Strictly speaking, this scheme does not preserve the unitary time evolution. However, its convergence upon an increase of L is fast so that L can be chosen such that the deviation from unitarity is dominated by the round-off error [25]. We have found that it is sufficient to take $L \leq 80$ for $N \lesssim 10^4$ and the further increase of L does not improve the accuracy of calculations.

3. The algorithm

We restrict the consideration to Hamiltonians of the form

$$H(t) = H_0 + f(t)H_{\text{mod}}, \qquad f(t+T) = f(t), \tag{13}$$

where $f(t)$ is a scalar function and H_0, H_{mod} are time-independent Hermitian operators. Most of the currently used models, including the ones discussed in Section 4, belongs to this class. Eq. (13) is the simplest nontrivial case of a general situation, $H(t) = H_0 + \sum_s f_s(t)\cdot H^{(s)}_{\text{mod}}$, with $s \leqslant N^2$. Our results can be generalized to the case $s > 1$ in a straightforward manner.

Next we specify the method to approximate $\Omega(t_1, t_2)$. As we discussed in the previous section, there exist a variety of schemes [24]. Our choice is conditioned by the form of the Hamiltonian, Eq. (13), and the intention to implement the algorithm on a parallel cluster. More specific on the last point, we are not concerned about the number of commutators needed to be calculated (and then stored) as long as they are all time-*in*dependent and do not have to be re-calculated in course of the propagation. Here, at each k-step of time propagation (where $k = 1, \ldots, M$) we use the following approximation of the Magnus expansion [24,29]:

$$\begin{aligned}\Omega_k &:= \Omega(t_{k-1}, t_k) \\ &= \alpha_1 + \frac{1}{12}\alpha_3 + \frac{1}{240}[-20\alpha_1 - \alpha_3 + C_1, \alpha_2 + C_2]\end{aligned} \tag{14}$$

so that $\Omega(t_{k-1}, t_k)$ is accurate up to order $\mathcal{O}(h^7)$. Specifically we have

$$\alpha_j = \frac{h^j}{(j-1)!}\frac{d^{j-1}H(t_{1/2})}{dt^{j-1}}, \qquad C_1 = [\alpha_1, \alpha_2], \tag{15}$$

$$C_2 = -\frac{1}{60}[\alpha_1, 2\alpha_3 + C_1],$$

where the midpoint $t_{1/2} = t_{k-1} + h/2$.

The original formulation demands the calculation of α_j on every time step. For the specific choice given by Eq. (13) this task reduces to calculations of the midpoint values of the scalar functions $f(t)$, $f'(t)$, and $f''(t)$ only. These values have to be weighted with time-independent nested commutators of the forms $[H_0, H_{\text{mod}}]$, $[H_0, [H_0, H_{\text{mod}}]]$, etc. There are nine commutators for the chosen

$\exp(-i\epsilon_\mu t/\hbar)\,\phi_\mu(t)\}$ was termed "Floquet states", while the set $\{\phi_\mu(t)\}$ was termed "Floquet modes", see Refs. [7–9]. In the context of the idea of "effective Hamiltonian", which is now being actively studied, the convention has been changed so that the states $\phi_\mu(t)$ are now addressed as Floquet states in most of the publications, see Refs. [15,16,18,20]. Here we stick to the latter convention.

scheme, Eq. (14), but they have to be calculated only once, when initiating the algorithm.

The choice of the operational basis to write operators H_0 and $H_{\rm mod}$ is important. We use of the eigenbasis of the operator H_0 *without* going into the interaction picture; see a relevant discussion in Ref. [26]. In this basis Eq. (13) assumes the form

$$i\hbar\partial_t|\psi(t)\rangle = [\mathrm{diag}(E_i) + f(t)\tilde{H}_{\rm mod}]|\psi(t)\rangle, \tag{16}$$

where $\mathrm{diag}(E_i)$ is a diagonal matrix consisting of the eigenvalues $\{E_i\}$ of H_0, and $\tilde{H}_{\rm mod}$ is the matrix representation of the operator in the eigenbasis of H_0. In numerical experiments with different periodically-driven nonlinear potentials, we found that this choice of the basis guarantees stable performance for $N > 10^3$. Because of the diagonal form of the matrix H_0, it also simplifies calculation of the nested commutators.

The algorithm can be presented as a sum of three constituents; these are

- construction of the time evolution operator $U(T)$ by propagating all basis states of a time-dependent Hamiltonian over the time interval $[0, T]$;
- use of the Magnus expansion for the propagation of a state with an explicitly time-dependent Hamiltonian;
- numerical diagonalization of the matrix $U(T)$.

Note that neither of these three is specific to the Floquet problem; they are often used, separately and in pairs, in computational quantum physics.

Operationally, the first constituent can be described as the propagation of the $N \times N$ identity (in the eigenbasis of H_0) matrix over the time interval T. The second one is implemented by means of a chain of M single-step Magnus propagators over the time interval $h = T/M$ by performing L Chebyshev iterations, Eqs. (10) and (12), with the rescaled operator $\tilde{\Omega}_k := \tilde{\Omega}(t_{k-1}, t_k)$, Eq. (11). In order to apply the rescaling procedure (11), $\Omega_k \mapsto \tilde{\Omega}_k$, which is a part of second component, one has to estimate the extreme eigenvalues $\Omega_{\rm min}$ and $\Omega_{\rm max}$ beforehand. We diagonalize the matrix $\tilde{\Omega}_k$ at five equidistant time instants within $[0, T]$ interval, and use the maximal and minimal values from the collected eigenvalues set as $\Omega_{\rm min}$ and $\Omega_{\rm max}$.

Once the propagation stage is completed, the result, i.e. a $N \times N$ unitary matrix U_T is diagonalized and its eigenvalues $\{\epsilon_\mu\}$ and eigenvectors $\{|\phi_\mu(0)\rangle\}$, are written into the output file. An additional propagation round can be performed, now with eigenvectors $\{|\phi_\mu(0)\}$ as initial vectors, to calculate characteristics of the Floquet states. For example, it can be the expectation value of a relevant operator $A(t)$, averaged over the one period

$$\langle A_\mu\rangle_T = \frac{1}{T}\int_0^T \langle\phi_\mu(t)|A(t)|\phi_\mu(t)\rangle dt. \tag{17}$$

4. Models

To test the algorithm, we employed two Hamiltonian models. The first is an abstract synthesized model, with the Hamiltonians H_0 and $H_{\rm mod}$ in Eq. (13) being members of a Gaussian orthogonal ensemble GOE(N) [30] of a variance σ. Random matrix theory and the corresponding models remain at the center of research in many areas of quantum physics [31], but it is only very recently that two hitherto disentangled research fields, random matrix and Floquet theories, started to interact [18,19].

Our second test model consists of a driven N-particle Bose–Hubbard dimer [32], with the Hamiltonians

$$\begin{aligned} H_0 &= -\upsilon(\hat{a}_1^\dagger\hat{a}_2 + \hat{a}_1\hat{a}_2^\dagger) + \frac{U}{2}(\hat{n}_1 - \hat{n}_2)^2, \\ H_{\rm mod} &= (\hat{n}_2 - \hat{n}_1), \end{aligned} \tag{18}$$

where $\hat{a}_j^\dagger$ ($\hat{a}_j$) and $\hat{n}_1 = \hat{a}_j^\dagger\hat{a}_j$ are the bosonic creation (annihilation) and particle number operators for the jth site, respectively. Parameters υ and U are the hopping rate and one-site interaction strength. In the Fock basis the Hamiltonian H_0 acquires tridiagonal structure, while $H_{\rm mod}$ becomes a diagonal matrix. This model is extensively used in many-body quantum physics, both in theoretical and experimental domains; e.g., see Ref. [33].

5. Implementation of the algorithm on a cluster

We now describe the program implementation of the algorithm on a high-performance cluster. Our C code employs Intel® Parallel Studio XE package [34]. The main data structures are complex double-precision matrices. Computational load is distributed among cluster nodes by the standard Message Passing Interface (MPI). On each node computationally intensive operations are implemented by calling BLAS functions from Intel® Math Kernel Library (Intel MKL), in the shared-memory parallel mode [35].

The code consists of three main steps, summarized in the pseudocode presented in Algorithm 1. Firstly, the program initializes MPI, allocates memory, reads the seed data and parameters from configuration files, and makes necessary pre-calculations before launching the main cycle: calculates the eigenbasis of the Hamiltonian H_0 and auxiliary matrices $\mathrm{diag}(E_i)$, $\tilde{H}_{\rm mod}$ (see Eq. (16)), the Bessel functions[3] $J_l(r)$ needed for the Chebyshev series (see Eq. (12)), and nine commutators needed for the Magnus expansion (see Eq. (14)). These computations are performed on each cluster node. It is important to choose appropriate operational presentation of the $N \times N$ matrices, starting from the initial identity matrix $\mathbb{1}$. The most straightforward solution is to split $\mathbb{1}$ into N vectors, store the vectors as independent arrays, and then propagate them independently and in parallel. A more efficient solution is to form sub-matrices of initial vectors that allows for parallel propagation and then make use of the third-level BLAS operations, in particular, matrix–matrix product, instead of a series of matrix–vector products. As a result, the memory hierarchy is used in a more efficient way and a substantial decrease of the computation time is achieved.

The second step involves propagation of the initial matrix $\mathbb{1}$ over the period T. Because the process is iterative, parallelization in time is not possible. However, data parallelization is feasible. The initial identity matrix can be split into P sub-matrices X_j, $j = 1, \ldots, P$, each consisting of N/P basis vectors, which are then distributed among P cluster nodes. Therefore, the first N/P rows are propagated on the first node, the next N/P rows on the second node, etc. (according to the C row-major order, initial vectors are written as rows). This idea is sketched in Fig. 1. The scheme possesses a minor drawback that could be encountered in the case of a large number of processing units, when splitting could cause a strong imbalance between the number of columns and rows. This might affect the performance of those mathematical kernels, which were not developed to handle "thin" matrices consisting of a few rows and thus limits the number of processing units that could be used to accelerate the propagation. The major advantage of the scheme, however, is a near uniform distribution of the workload among the nodes. Together with a constant number of operations on each step, this allows to estimate the scaling of the overall computing time with P.

A single propagation step implements the recipe given at the end of Section 3. By employing MKL functions, we calculate the matrix $\Omega(t_{k-1}, t_k)$, Eq. (14). It is computed independently on each

[3] The Bessel functions were numerically computed using Fortran intrinsic function *bessel_jn*.

Algorithm 1

1: initialization, memory allocation
2: **upload** system & method parameters: the number of steps M per period T, the number of Chebyshev iterations L
3: calculate basis of the stationary Hamiltonian H_0, auxiliary matrices diag(E_i) and $\tilde{H}_{\rm mod}$
4: calculate Bessel functions $J_l(r)$, $r = r(h)$, $l = 0, \ldots, L$
5: distribute the initial sub-matrices X_j, $j = 1, \ldots, P$ and precalculated data between P nodes
6: **for** $k = 1$ to M **do** ▷ *MPI distributed computational loop*
7: calculate $\Omega_k := \Omega[t_{k-1}, t_k]$ for the current $t_k = kh = kT/M$
8: rescale $\Omega_k \mapsto \tilde{\Omega}_k$
9: perform L Chebyshev iterations with $\tilde{\Omega}_k$ and update X_j
10: **end for**
11: combine X_j into U_T
12: diagonalize U_T
13: **save** eigenvectors and eigenvalues of U_T
14: **release** memory

Table 1
Single-node performance: Execution times (in sec) as a function of system size N. Multi-threaded version of the code employs all 16 node's cores on shared memory. Columns ii–iv and vi present data obtained for $M = 10^2$ time steps per period and $L = 50$ Chebyshev iterations on every step. To get an estimate for $M = 10^4$, the times needed to calculate Ω_k and perform Chebyshev iterations were extrapolated (last column), see text for more details.

System size, N	Auxiliary computations time	Time of Ω_k calculation	Chebyshev iterations time	Diagonalization time	Total time $M = 10^2$	Totaltime $M = 10^4$, extrapolation
256	0.2	0.4	4.3	0.2	5.1	470.4
512	0.4	1.8	25.3	0.6	28.1	2 711.0
768	1.3	3.2	79.9	1.4	85.8	8 312.7
1 024	1.8	8.3	177.3	2.6	190.0	18 564.4
1 536	4.3	18.9	559.1	7.3	589.6	57 811.6
2 048	8.6	33.2	1 296.6	16.0	1 354.4	133 004.6
3 072	23.5	72.1	4 179.2	51.0	4 325.8	425 204.5
4 096	49.2	126.0	9 730.5	117.5	10 023.2	985 816.7
5 120	100.5	184.3	18 667.2	242.1	19 194.1	1 885 492.6
10 240	755.3	919.7	181 722.3	1 857.7	150 0175.1	18 266 809.4

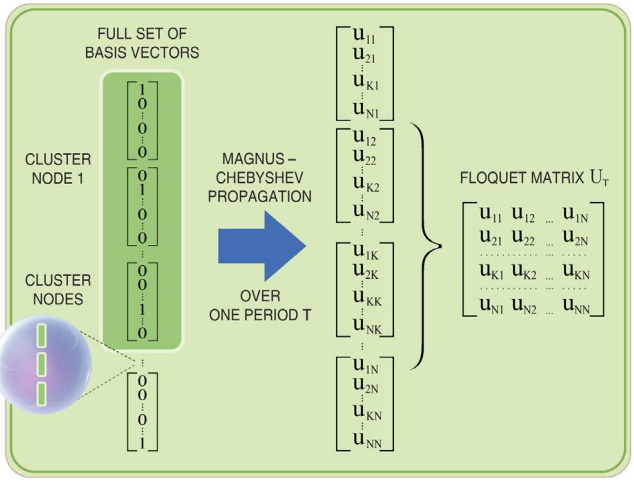

Fig. 1. (Color online) Parallel computation of a Floquet operator U_T. The initial $N \times N$ identity matrix $\mathbb{1}$ is sliced into P rectangular sub-matrices X_j, $j = 1, \ldots, P$, each consisting of N/P basis vectors. The sub-matrices are then independently propagated on P cluster nodes, by using the Magnus–Chebyshev propagation algorithm. The output vectors form the corresponding columns of the Floquet matrix.

cluster node, as the small computing time does not justify its calculation on the distributed memory.

The computationally intensive part of the algorithm is the approximation of the action of the matrix exponent by Chebyshev's iterations, Eqs. (10), (12), and the further updating of propagated sub-matrices on each cluster node. The mathematical core of this step is the multiplication of complex double precision dense matrices (it is implemented with *zgemm* routine [36]). This part of the algorithm is fully parallel.

During the final, third step the program assembles sub-matrices into the Floquet matrix and diagonalizes the latter by using a multi-threaded Intel MKL implementation (we use *zgeev* routine [36]). For the matrix size $N \sim 10^4$, a multi-core implementation is sufficient. Finally, the results of the diagonalization are written to the output files, the memory is deallocated, and MPI is finalized.

6. Program performance and scalability analysis

In this section we present the performance analysis of the code. Test runs were performed on the "Lobachevsky" supercomputer at the Lobachevsky State University of Nizhny Novgorod [37]. We employed up to 64 computational nodes, with the following configuration per node: 2× Intel Xeon E5-2660 CPU (8 cores, 2.2 GHz), 64 GB RAM, OS Windows HPC Server 2008. We use Intel MKL, Intel C/C++ Compiler, and Intel MPI from Intel Parallel Studio XE [34]. All parallel versions of computationally intensive routines from MKL utilized 16 cores on each node.

To test the performance of the program we use the synthesized random-matrix model as a benchmark (see Section 4). In this case, Hamiltonians H_0 and $\tilde{H}_{\rm mod}$ in Eq. (16) are generated randomly from the GOE(N) ensemble of the unit variance $\sigma = 1$ [38]. The driving function is $f(t) = \cos(\omega t)$ with $\omega = \pi$ [39].

Single-node performance. To test a single-mode performance, we use $M = 10^2$ steps per period and $L = 50$ Chebyshev iterations per every time step. The execution time for larger values of M can be easily extrapolated: due to the linear increase of operations number with iterations in time, it is sufficient to find and appropriately scale execution time of the core part of the code, and add execution time of the other parts, which are independent of M. Table 1 presents the dependence of the execution time for different stages on the size of the model system. The last column of Table 1 presents estimates for the case when the number of

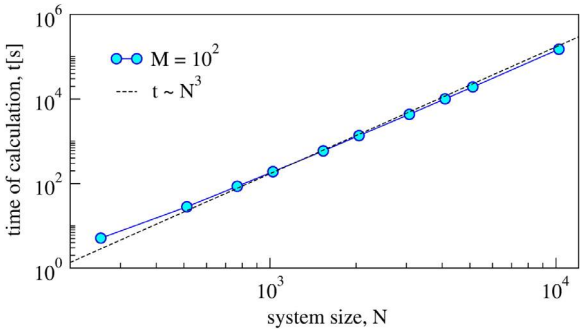

Fig. 2. (Color online) Total computation time on a single node as a function of the system size N. The number of steps M per period T is $M = 10^2$ with $L = 50$ Chebyshev iterations per step. Dashed line corresponds to the cubic scaling.

Table 2
Single-node performance. Computational intensity and efficiency measures, R_N and R'_N, as functions of the model system size N. Multi-threaded implementation of the algorithm on a single cluster node (16 cores on shared memory) was used. The number of steps per period is $M = 10^2$ with $L = 50$ Chebyshev iterations on every step.

System size, N	Total time, TIME_N, in sec	Operations count, OC_N, in mln	R_N	R'_N
256	5.1	173 612	1.00	1.00
512	28.1	1 365 642	1.43	1.45
768	85.8	4 594 916	1.57	1.60
1 024	190.0	10 875 688	1.68	1.72
1 536	589.6	36 655 208	1.83	1.87
2 048	1 354.4	86 961 706	1.89	1.93
3 072	4 325.8	292 683 840	1.99	2.04
4 096	10 023.2	693 473 512	2.03	2.08
5 120	19 194.1	1 354 053 950	2.07	2.13

steps is increased 100-fold, i.e. for $M = 10^4$. The total calculation time scales near cubically with N (see Fig. 2). The origin of this scaling is evident. Namely, as it can be seen from the table, the most time consuming operation is performing Chebyshev iterations. The time of a single iteration is determined by the matrix–vector multiplication, Eq. (10), whose time scales as N^2. Their total number on the propagation round is N (one for every basis vector). Multiplied, these two factors yield cubic scaling.

To gain a further insight, we analyze single-node performance in more detail. We consider two metrics, both essentially being the number of operations per execution time, required to make calculations for a system of a size N. First, we introduce the operation rate R and take the value obtained for $N = 256$ as a unit measure: $R_N = (\text{OC}_N/\text{OC}_{256})/(\text{TIME}_N/\text{TIME}_{256})$, where OC_N is the number of operations and TIME_N is the execution time of the code for the system of size N.

The number of operations is calculated by Intel® VTune™ Amplifier profiler [40] and returned to CPU performance counter SIMD_FP_256.PACKED_DOUBLE [41]. This variable contains the number of issued Advanced Vector Extensions (AVX) instructions for processing double precision values [42]. The choice of this counter is based on the fact that almost all computations in our code occur in Intel MKL BLAS routines. Note, however, that this estimate of the number of operations is not exact. It is well known that for the current architectures the profiler tends to overestimate this number, since it counts the number of instructions issued but retired. Nevertheless, this estimate is reliable for CPU-bound processes.

Still, keeping the above said in mind, we introduce the second quantity, R'_N, substituting the number of operations OC_N in the expression for R_N with its estimate N^3, according to the scaling of the most computationally intensive and most frequently called MKL subroutine *zgemm*.

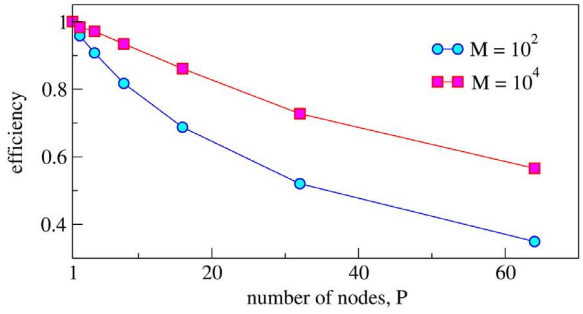

Fig. 3. (Color online) Computational efficiency as function of number of cluster nodes, P, for two values of number of steps per period, $M = 10^2$ and $M = 10^4$. The parameters are $N = 5120$ and $L = 50$.

The behavior of R and R' as functions of N are presented in two last columns of Table 2. The efficiency increases with the size of the model system, doubling for $N = 5120$ as compared to $N = 256$. That is because of the increasing efficiency in evaluation of larger matrices of the BLAS computational kernels in the parallel regime. While the efficiency grows with the system size, the execution time also increases, mainly because the increase of the number of steps per period needed, and for $N = 5120$ the estimated calculation time is about 22 days. Therefore, a multi-node parallelization is required to decrease the calculation time to more realistic time scales.

Strong scalability of the algorithm. We next analyze the performance of the algorithm on a cluster. To benchmark the code, we use the random-matrix model of the size $N = 5120$ and launch the code on $P = \{2^0, 2^1, \ldots, 2^6\}$ cluster nodes, using the multi-threaded implementation on each node as before. The results are summarized in Table 3. Let us note that the time needed for the diagonalization of $N = 5120$ matrix is 242.1 sec and does not depend on P (see column vi in Table 1). Therefore, it is omitted from the further analysis.

For $M = 10^2$ the code accelerates as the number of nodes increases to 64 (1024 computational cores in total), though the efficiency of parallelization, defined as the ratio between the speed up and the number of nodes, drops to 35%. That is because for this, relatively small, number of integration steps the execution time of serial parts of the code becomes comparable to that of the parallel code (Chebyshev iterations), which scaling efficiency, in turn, is about 86%. Taking into account that practically reasonable computations require much larger numbers of integration steps, it is expected that the efficiency of the code will increase with M. It is confirmed by the results of test runs, see last two columns of Table 3. In particular, for $N = 5120$ and $M = 10^4$ the code is executed on 64 cluster nodes in 14.5 h, demonstrating the efficiency of parallelizing about 57%. Fig. 3 depicts the dependence of the efficiency on the number of employed cluster nodes.

7. Applications

In this section we test the accuracy of the algorithm employing two physical model systems which we described in Section 4 above.

The random-matrix model was already specified in Section 6. For the dimer model, Eq. (18), we use parameters $\upsilon = 1$ and $U = U' \cdot N = 2$. The one-site interaction is scaled with the number of bosons, $N - 1$, to match in the limit $N \to \infty$ the classical mean-field Hamiltonian [33,43],

$$H_{\text{cl}}(z, \nu) = \frac{U'}{2} z^2 - 2\upsilon\sqrt{1 - z^2}\cos(\nu) + 2z \cdot f(t). \tag{19}$$

The mean-field variables z and ν measure the population imbalance and relative phase between the dimer sites, respectively. The

Table 3
Scalability analysis: execution times (in sec) of different steps of the algorithm and speed-up factors (in %) are shown as functions of the number of employed cluster nodes. Each node runs multi-threaded version on 16 cores on shared memory. The parameters are $N = 5120$ and $L = 50$.

Number of nodes, P	Auxiliary computations time	Time of Ω_k calculation	Chebyshev iterations time	Chebyshev iterations speed up	Total time $M = 10^2$	Speed up $M = 10^2$	Total time $M = 10^4$	Speed up $M = 10^4$
1	100.5	184.3	18 667.2	1.0	19 194.1	1.0	1 885 457.9	1.0
2	186.4	181.5	9 406.9	2.0	10 017.8	1.9	959 150.8	2.0
4	195.3	180.2	4 670.7	4.0	5 288.8	3.6	485 400.8	3.9
8	173.6	178.7	2 341.2	8.0	2 935.8	6.5	252 305.2	7.5
16	137.1	178.4	1 187.2	15.7	1 744.7	11.0	136 882.6	13.8
32	103.9	178.7	628.0	29.7	1 152.6	16.7	81 006.1	23.3
64	98.9	178.6	338.3	55.2	858.6	22.4	52 053.6	36.2

driving function $f(t) = f_{\rm dc} + f_{\rm ac}\cos(\omega t)$ consists of two components, a constant dc-bias $f_{\rm dc} = 2.7$ and single-harmonic term $f_{\rm ac}\cos(\omega t)$, with the amplitude $f_{\rm ac} = 2.5$ and frequency $\omega = 3$. We use the phase space of the mean-field system, Eq. (19) together with the semi-classical eigenfunction hypothesis [44] and the concept of "hierarchical eigenstates", as originally introduced in Ref. [45], to benchmark the program.

Once the diagonalization of U_T is completed, the program initiates an additional round of the T-propagation to calculate the average energies of the Floquet states, with $A(t) = H(t)$ in Eq. (17). Finally, the Floquet states are sorted in ascending order according to their average energies. To quantify the accuracy, we adapt the idea of the overlap phase error [25] and modify it to account for the periodicity of the Floquet states,

$$\Sigma_\mu = |1 - \langle\phi_\mu(0)|U_T|\tilde{\phi}_\mu(0)\rangle|. \tag{20}$$

where $\tilde{\phi}_\mu(0)$ is the Floquet state calculated with the time step $\tilde{h} = h/2$. Note that the error is state-specific.

Fig. 4 presents the error Σ_μ as a function of number of steps per period M and number of system states N. A linear dependence of $\log_{10}\Sigma_\mu$ on $\log_{10} M$ observed for the random-matrix model is typical for stepwise integrators. The error convergence with the number of steps does not saturate up to largest value $M = 10\,240$. In the case of the dimer, however, the error does not reveal the power-law scaling and demonstrate a noticeable saturation. The difference in the scalings can be attributed to the differences in the spectral properties of the matrices H_0 and $H_{\rm mod}$: while in case of the random-matrix model the level spacings of the Hamiltonians are characterized by a probability density function (pdf) with a gap near zero, the level spacings in the energy spectrum of the integrable dimer Hamiltonian are characterized by a gapless Poisson pdf [22]. The Floquet ground state $|\phi_1\rangle$ turns out to be the most sensitive one to the discretization of the unitary evolution, see in Fig. 4(a). We use this state to test the dependence of the error on the size of the model system. Fig. 4(b) shows that the scaling of Σ_μ with N is qualitatively similar for the both models.

Before analyzing the performance of the algorithm with the dimer model, we discuss relations between the accuracy and the efficiency (scalability) of the algorithm. We have not noticed any improvement of the accuracy on increasing the number of Chebyshev iterations per step beyond the limit $L_{\max} = 80$. This allows us to assume that for the used approximation of the Magnus expansion, Eq. (14), this parameter can be fixed, $L = L_{\max}$, for models with $N \leqslant 10^4$. The only parameter left to tune is the number of integration steps per period, M. From the data presented with Fig. 4(a), it follows that higher accuracy can be gained for the random-matrix model by increasing M further. The increase of the number of integration steps will increase the overall computation time linearly (for the scalable version of the algorithm).

For the Bose–Hubbard model (cf. Fig. 4(b)) the situation is different: the accuracy saturates with the increase of M. The only possibility to increase it further, while remaining within the algorithm bounds, is to use higher-order approximations of the Magnus expansion [29]. This would increase the number of commutators in the expression for Ω_k; see the discussion after Eq. (14). However, as before, the commutators have to be calculated once and then stored. The increase of the computation time will stem form the summation of a larger number of $N \times N$ matrices when calculating operator Ω_k. Although, within our scheme, this operation is not scalable, see Table 3, it is of a sub-linear order with respect to N. Therefore, when calculating the Floquet spectrum of a generic quantum model of the type (13), the increase of the approximation order seems to be the best default strategy to gain more accuracy.

Now we turn to the quantum benchmarking of the algorithm with the dimer model. Following the semi-classical eigenfunction hypothesis [44,22], the Floquet states of the model in the limit $N \gg 1$ can be sorted according to the location of the state's Husimi (or Wigner) distributions [22] in the mean-field phase space. The latter is *mixed* in the case of the driven Hamiltonian (19), so that regular and chaotic regions coexists [46] (see dots on the Poincaré sections on Fig. 5). If the distribution of a Floquet state is localized inside a regular region the state is labeled "regular". When the distribution is located in the bulk of the chaotic sea, the corresponding Floquet states is "chaotic". The regular regions, called "islands", are often organized in a hierarchical way, forming fine island-near-island structures. By increasing the number of bosons in the dimer, it is possible to resolve these structures with a higher level of detail. It is important to understand, however, that a gradual shift towards the semi-classical limit does not simplify the quantum problem. On the contrary, the increase of N increases the size of the Hamiltonian matrices, the number of the basis vectors, and, what is most dramatic, decreases the mean spacing between quasienergies ϵ_μ (which for any N remain restricted to the fundamental band $[-\hbar\omega/2, \hbar\omega/2]$). Therefore, the accuracy of calculations has to be increased in parallel, otherwise the numerically obtained Floquet states would represent incoherent mixtures of actual Floquet states (while the deviation form the unitarity may remain reasonably small). A theoretical interpretation of so obtained numerical results might lead to wrong conclusions.

The accuracy of the scheme can be checked by means of the chaotic–regular dichotomy. Figs. 5(e) and (f) show Husimi distributions for two regular Floquet states, while Fig. 5(c) presents the distribution for a chaotic state. The accuracy can be tested even more carefully with a third class of quantum states, located on the interface between chaotic and regular ones. On increasing the number of states N and approaching the semiclassical limit, the chaotic Floquet states are expected to fit Berry's conjecture [44] and have their Husimi distributions smeared over classically chaotic regions, while the distributions of the regular Floquet states become localized on the classical regular tori [22]. These *hierarchical* [45] states are supported by the classical phase-space structures located on the chaotic sea's offshore around regular islands. The Husimi distribution of a hierarchical state

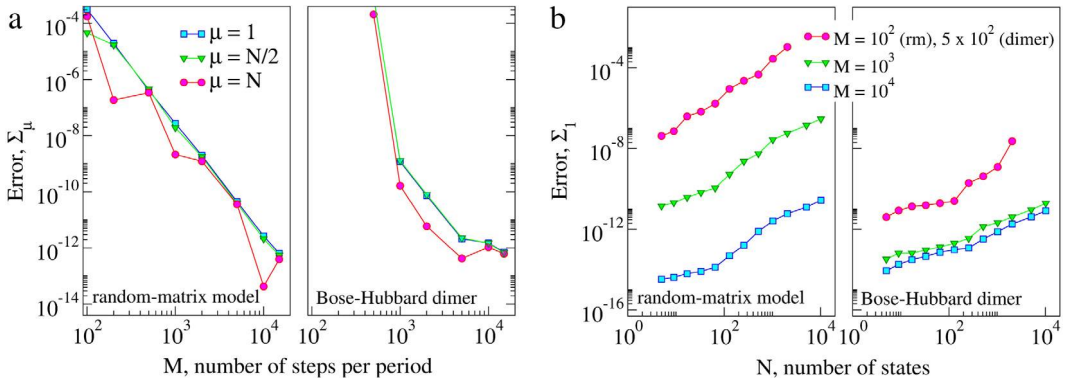

Fig. 4. (Color online) Phase error Σ_μ, Eq. (20), as a function of (a) number of steps per period M (for $N = 1024$) and (b) size N of the model system. (a) The dependence Σ_μ for the three Floquet states $|\phi_\mu\rangle$: (□) groundstate, $\mu = 1$; (▽) the state from the center of the energy spectrum, $\mu = N/2$; (○) the highest-energy state, $\mu = N$. (b) The dependence of Σ_μ on system size N for the Floquet groundstate $|\phi_1\rangle$ for different number M of steps per period: (○) $M = 10^2$ for the random-matrix model (rm) and $M = 5 \cdot 10^2$ for the Bose–Hubbard dimer (dimer); (▽) $M = 10^3$; (□) $M = 10^4$.

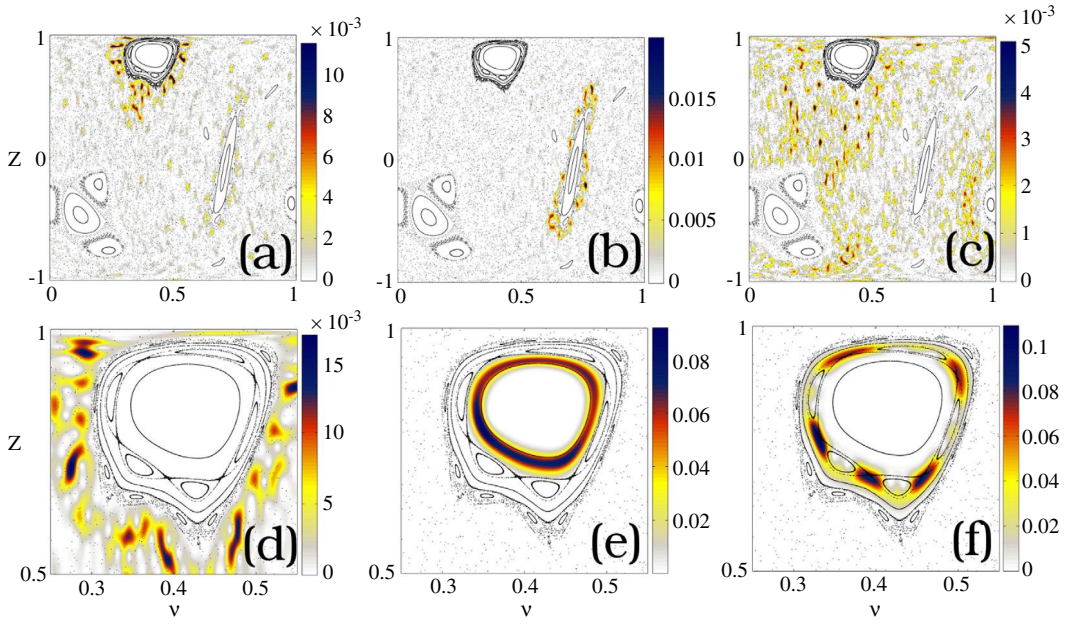

Fig. 5. (Color online) Husimi distributions of Floquet states $|\phi_\mu\rangle$ of the dimer model with $N = 2499$ bosons. Dots show the Poincaré section for the mean-field Hamiltonian, Eq. (19). Circle-like formations correspond to the KAM tori [46]. (a), (b) and (d) Hierarchical Floquet states of the quantum system: $\mu = 118$ (a), $\mu = 1024$ (b), $\mu = 101$ (d, zoom). (c) Chaotic Floquet state, $\mu = 1002$. (f) and (g) Two regular Floquet states: $\mu = 30$ (f) and $\mu = 55$ (g).

is concentrated in the immediate exterior of the corresponding island. These regions are structured by "semi-broken" tori of fractal geometrical structure, which are called *cantori* (from "*Can*tor" + "*tori*") and are responsible for several nontrivial phenomena such as anomalous diffusion and non-exponential relaxation [47].

Hierarchical states are exceptional in the sense that their absolute number increases sub-linearly with the number of states, $N_{hier} \sim N^\chi$, $\chi < 1$, so that their relative fraction N_{hier}/N goes to zero in the limit $N \to \infty$ [45]. These states must be carefully selected from the complete set of N Floquet states.

Hierarchical states are coherent superpositions of many basis vectors and therefore sensitive to the phase errors. Even a small mismatch in vector phases blurs the interference pattern and causes the flooding of the state's Husimi distribution into the island. The high coherence of the superpositions is also a trait of the regular Floquet states but there is an important difference: quasienergies ϵ_μ of the hierarchical states are randomly distributed over the interval $[-\hbar\omega/2, \hbar\omega/2]$, while those of regular and chaotic states tend to cluster in different regions [45]. Because of that, phases of hierarchical states are most vulnerable to the error produced by the numerical propagation. We selected several hierarchical states for the dimer model with $N = 2499$ bosons and inspect their Husimi distributions[4] [48], see Figs. 5(a), (b) and (d). The offshore localization and absence of the flooding [see zoomed distribution on Fig. 5(d)] are clearly visible.

8. Summary and outlook

We have put forward a method to calculate Floquet states of periodically-modulated quantum system with $N \geq 10^4$ states. Our method is scalable and therefore well suited for its implementation

[4] We use the definition of the Husimi distribution for the many-particle dimer presented in Refs. [48]. The expression involves summation over the series of square roots of binomial coefficients of the order N. We did not find an alternative expression which allows to avoid the term-by-term summation. Therefore, although we were able to calculate Floquet states for the periodically-modulated dimer with 10 239 bosons, we were not able to go beyond the limit $N = 2500$ when plotting the Husimi distributions.

on parallel computers. Our study uses massively parallel clusters as efficient devices to explore complex quantum systems far from equilibrium, thus answering the need of several, actively developing, research fields involving quantum physics [12–20].

The method particularly allows for improvements, such as the increase of the order of the Magnus expansion [29] or the use of commutator-free approximations [26]. With respect to further acceleration of the code for systems with $N \leq 10^4$ states, there is a promising perspective related to the fact that the main contribution to the computation time stems from the BLAS operations. These operations fit GPU and Intel Xeon Phi architectures very well. By our estimates, even a straightforward implementation of the most computationally intensive Chebyshev iteration stage on a heterogeneous CPU+GPU configuration will result in a three-fold speedup. A yet further speed-up can be obtained by using multiple accelerators.

There are several interesting research directions for which the proposed algorithm may serve as a useful starting point. For example, there is the perspective to resolve Floquet states of even larger systems by applying the spectral transformation Lanczos algorithm [49] to the corresponding time-*independent* super-Hamiltonians, to name but a few. Because the super-Hamiltonian elements can be generated on the fly, this idea potentially would allow to calculate Floquet states of a system with $N \sim 10^5$ states for $F \sim 10^4$ Fourier harmonics, by employing massively parallel exact diagonalization schemes[5] [51]. Note, however, that the eigenvalues of the Hamiltonian super-matrix (as well as the respective quasienergies) are merely phase factors and are not directly related with the properties of the corresponding Floquet states (average energy, etc.). Therefore, some targeting of the algorithm to the low-energy states is required. Our method can be used to locate the relevant Floquet eigenvectors in the quasi-energy spectrum of a system with a smaller number of states; combining this with a knowledge of the spectrum scaling with N, one can target the Lanczos algorithm.

Another direction relates to the computational physics of *open* periodically-modulated quantum systems that interact with a large environment (heat bath). Asymptotic states of such systems are affected by the combined effects of modulation and the decoherence induced by the environment [52]. Due to linearity of the model equations describing the evolution of the density matrices of the systems, the corresponding asymptotic states are specified by time-periodic density matrices, which can be called "quantum attractors" [53–56]. There is presently limited knowledge about the theme of quantum attractors beyond the limit of the rotating-wave approximation [57,58]. In the Floquet framework, the attractor's density matrix is the eigenvector, with the eigenvalue one, of the corresponding non-unitary Floquet super-operator acting in the space of $N \times N$ Hermitian matrices. This super-operator can be constructed by propagating the identity operator—but now in the space of $N \times N$ matrices. The propagation stage can be realized by using Padé approximation [28] or with Newton or Faber polynomial schemes [59,60], while the question whether there exists a possibility to generalize the Magnus expansion to dissipative quantum evolution equations remains open. As the number of the basis matrices scales as N^2, the scalability of non-unitary propagation algorithms is even more required.

Acknowledgments

The authors acknowledge support of Ministry of Education and Science of the Russian Federation, Research Assignment No. 1.115.2014/K (Sections 2–6) and the Russian Science Foundation Grant No. 15-12-20029 (Section 7).

[5] A similar idea was recently proposed in Ref. [50].

References

[1] I. Bloch, J. Dalibard, W. Zwerger, Rev. Modern Phys. 80 (2008) 885.
[2] M. Poot, H.S.J. van der Zant, Phys. Rep. 511 (2012) 273.
[3] M. Aspelmeyer, T.J. Kippenberg, F. Marquardt, Rev. Modern Phys. 86 (2014) 1391.
[4] M. Dutta, M.A Stroscio, Quantum-Based Electronic Devices and Systems, World Scientific, 1998.
[5] J.C. Cullum, R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Birkhauser, Boston, 1985.
[6] U. Schollwoeck, Rev. Modern Phys. 77 (2005) 259.
[7] J.H. Shirley, Phys. Rev. 138 (1965) B979.
[8] H. Sambe, Phys. Rev. A 7 (1973) 2203.
[9] M. Grifoni, P. Hänggi, Phys. Rep. 304 (1998) 229.
[10] S. Kohler, J. Lehmann, P. Hänggi, Phys. Rep. 406 (2005) 379.
[11] E. Arimondo, D. Ciampini, A. Eckardt, M. Holthaus, O. Morsch, Adv. At. Mol. Opt. Phys. 61 (2012) 515.
[12] M. Bukov, L. D'Alessio, A. Polkovnikov, Adv. Phys. 64 (2015) 139.
[13] J. Eisert, M. Friesdorf, C. Gogolin, Nat. Phys. 11 (2015) 124.
[14] N.H. Lindner, G. Refael, V. Galitski, Nat. Phys. 7 (2011) 490.
[15] D.E. Liu, A. Levchenko, H.U. Baranger, Phys. Rev. Lett. 111 (2013) 047002; A. Kundu, B. Seradjeh, Phys. Rev. Lett. 111 (2013) 136402.
[16] N. Goldman, J. Dalibard, Phys. Rev. X 4 (2014) 031027.
[17] A. Lazarides, A. Das, R. Moessner, Phys. Rev. E 90 (2014) 012110.
[18] A. Lazarides, A. Das, R. Moessner, Phys. Rev. Lett. 115 (2015) 030402.
[19] L. D'Alessio, M. Rigol, Phys. Rev. X 4 (2014) 041048.
[20] P. Ponte, Z. Papić, F. Huveneers, D. Abanin, Phys. Rev. Lett. 114 (2015) 140401.
[21] A. Eckardt, E. Anisimovas, High-frequency approximation for periodically driven quantum systems from a Floquet-space perspective. arXiv:1502.06477.
[22] F. Haake, Quantum Signatures of Chaos, Springer, Berlin, 2000.
[23] J.-S. Caux, J. Mossel, J. Stat. Mech. (2011) P02023.
[24] S. Blanes, F. Casaa, J.A. Oteo, J. Ros, Phys. Rep. 470 (2009) 151.
[25] H. Tal-Ezer, R. Kosloff, J. Chem. Phys. 81 (1984) 3967; C. Leforestier, et al., J. Comput. Phys. 94 (2001) 59.
[26] A. Alvermann, H. Fehske, J. Comput. Phys. 230 (2011) 5930.
[27] W. Magnus, Comm. Pure Appl. Math. VII (1954) 649.
[28] C. Moler, C.F. Van Loan, SIAM Rev. 45 (2003) 3.
[29] S. Blanes, F. Casas, J. Ros, BIT 42 (2002) 262.
[30] G. Akemann, J. Baik, P. Di Francesco, The Oxford Handbook of Random Matrix Theory, Oxford University Press, 2011.
[31] T. Guhr, A. Mueller-Groeling, H.A. Weidenmueller, Phys. Rep. 299 (1998) 189.
[32] A.J. Leggett, Rev. Modern Phys. 73 (2001) 307.
[33] T. Zibold, E. Nicklas, C. Gross, M.K. Oberthaler, Phys. Rev. Lett. 105 (2010) 204101; D. Midtvedt, A. Isacsson, A. Croy, Nat. Comm. 5 (2014) 4838.
[34] Intel® Parallel Studio XE 2015. https://software.intel.com/en-us/intel-parallel-studio-xe.
[35] Intel® Math Kernel Library. https://software.intel.com/en-us/intel-mkl.
[36] Reference Manual for Intel® Math Kernel Library 11.2. https://software.intel.com/en-us/mkl_11.2_ref.
[37] http://www.top500.org/system/178472.
[38] A. Edelman, N.R. Rao, Acta Numer. (2005) 1.
[39] We set the effective Planck constant to $\hbar = 1$ for both models.
[40] Intel® VTune™ Amplifier. https://software.intel.com/en-us/intel-vtune-amplifier-xe/.
[41] Intel® VTune™ Amplifier User's Guide. https://software.intel.com/ru-ru/node/529797.
[42] Intel® 64 and IA-32 Architectures Software Developers Manual. Volume 1: Basic Architecture.
[43] C. Weiss, N. Teichmann, Phys. Rev. Lett. 100 (2008) 140408.
[44] M.V. Berry, J. Phys. A 10 (1977) 2083.
[45] R. Ketzmerick, L. Hufnagel, F. Steinbach, M. Weiss, Phys. Rev. Lett. 85 (2000) 1214.
[46] A.J. Lichtenberg, M.A. Lieberman, Regular and Chaotic Dynamics, Springer-Verlag, New York, 1992.
[47] R.S MacKay, J.D Meiss, I.C. Percival, Physica D 13 (1984) 55.
[48] F.T. Arecchi, E. Courtens, R. Gilmore, H. Thomas, Phys. Rev. A 6 (1972) 2211; A.P. Hines, R.H. McKenzie, G.J. Milburn, Phys. Rev. A 71 (2005) 042303.
[49] T. Ericsson, A. Ruhe, Math. Comp. 34 (1980) 1251.
[50] C.A. Parra-Murillo, J. Madronero, S. Wimberger, Comp. Phys. Comm. 186 (2015) 19.
[51] A. Weisse, H. Fehske, Exact diagonalization techniques, Lecture Notes in Phys. 739 (2008) 529; S. Isakov, et al. High-Performance Exact Diagonalization Techniques, in: CUG2012 Final Proceedings, 2012.
[52] H.-P. Breuer, F. Petruccione, The Theory of Open Quantum Systems, Oxford University Press, 2007.
[53] T. Dittrich, B. Oelschlägel, P. Hänggi, Europhys. Lett. 22 (1993) 5; B. Oelschlägel, T. Dittrich, P. Hänggi, Acta Physica Polonica B 24 (1993) 845.
[54] T. Dittrich, F. Grossmann, P. Hänggi, B. Oelschlägel, R. Utermann, in: Z. Haba, W. Ceglar, L. Jakobczyky (Eds.), Stochasticity and Quantum Chaos, Kluwer Academic Publishers, Dordrecht, Boston, London, 1995, pp. 39–55.

[55] H.-P. Breuer, W. Huber, F. Petruccione, Phys. Rev. E 61 (2000) 4883.
[56] R. Ketzmerick, W. Wustmann, Phys. Rev. E 82 (2010) 021114.
[57] R. Utermann, T. Dittrich, P. Hänggi, Phys. Rev. E 49 (1994) 273.
[58] S. Kohler, R. Utermann, P. Hänggi, T. Dittrich, Phys. Rev. E 58 (1998) 7219.
[59] M. Berman, R. Kosloff, H. Tal-Ezer, J. Phys. A: Math. Gen. 25 (1992) 1283.
[60] W. Huisinga, L. Pesce, R. Kosloff, P. Saalfrank, J. Chem. Phys. 110 (1999) 5538.